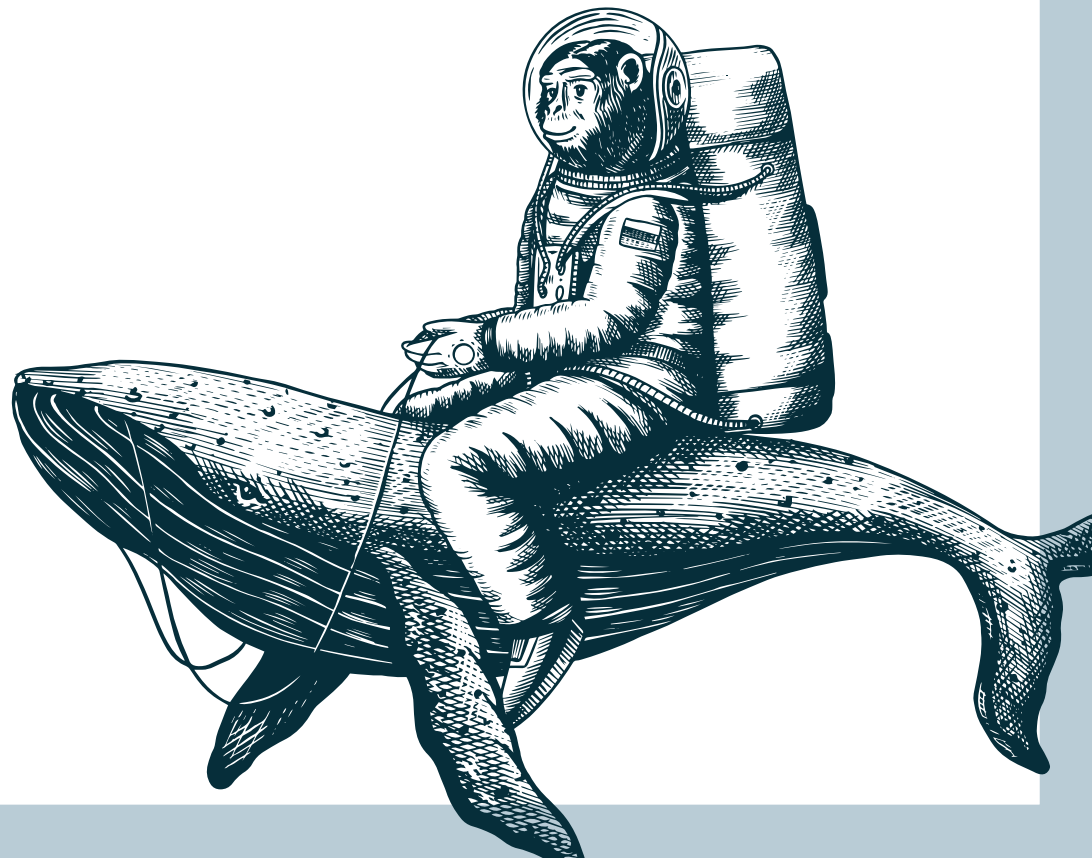
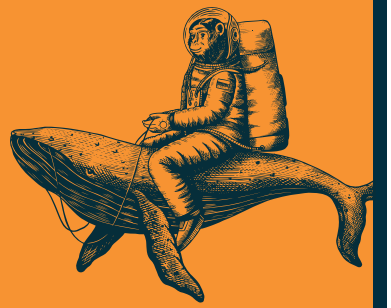
INFRALOVERS-ROADMAP:
DER WEG ZUR AGENTIC INFRASTRUCTURE

Standortbestimmung für Infrastructure-Leads, die verstehen wollen, warum IaC, Security und Platform Engineering die Voraussetzung für AI-gestützte Infrastruktur sind — und warum jetzt der Zeitpunkt ist, die Fundamente zu legen.



INFRALOVERS

INHALT



In zwei Jahren managen Agents Ihre Infrastruktur

Nicht als Vision. Als Notwendigkeit.

Drei Unternehmen zeigen, was kommt:

StrongDM betreibt eine Software Factory mit drei Personen — Agents schreiben Code, Agents testen Code, Menschen schreiben Specs.

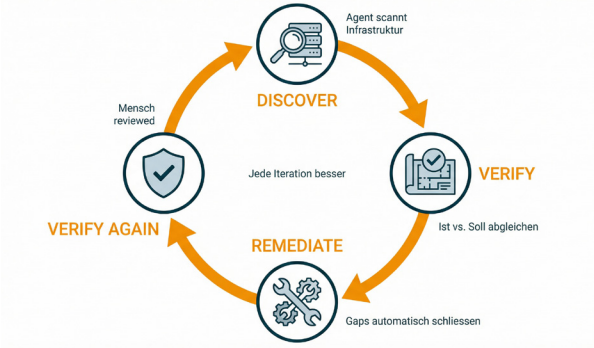
Anthropic generiert 70-90% seines Codes mit Claude Code.

Cursor, Anysphere — 300 Leute, über eine Milliarde Dollar Umsatz.

Vier Unternehmen — Cursor, Anthropic, Google, OpenAI — sind unabhängig voneinander auf dasselbe Pattern konvergiert: Decompose, Parallelize, Verify, Iterate.

Für Infrastruktur übersetzt: Agents, die Ihre Umgebung scannen, den Ist-Zustand gegen die Soll-Architektur abgleichen, und die Lücken schließen. Validiert gegen Compliance-Policies, verifiziert durch Observability-Daten, reviewed von einem Menschen.

Discovery → Architecture Verification → Remediation → Verification. Ein Loop, der sich mit jeder Iteration verbessert.



Ob das Zielbild IaC (Infrastructure as Code) mit Terraform ist oder etwas ganz anderes — das weiß heute niemand.

Was wir wissen: Agents brauchen strukturierte, maschinenlesbare Daten. Terraform State, Mondoo-Scans, Vault-Identity-Daten, OpenTelemetry Traces — das sind die Rohstoffe, ohne die kein Agent sinnvoll arbeiten kann. IaC ist heute das beste Werkzeug, diese Daten zu erzeugen. Kein Endzustand, aber der notwendige Schritt.

Das klingt weit weg, wenn Ihr Release-Zyklus in Wochen gemessen wird und die Deployment-Pipeline ein Jenkins-Job ist, den eine Person geschrieben hat.

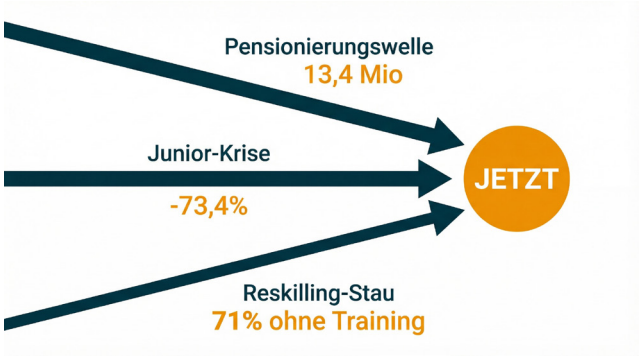
Genau das ist der Punkt.

Man kann nicht von manuellen Releases zu Agentic Infrastructure springen. Agents auf undokumentierte, manuell betriebene Infrastruktur loszulassen produziert nur schnelleres Chaos — Dr. Carsten Linz nennt es „AI Icing on the Cake“.

Die Stufen dazwischen — IaC, Vault, Mondoo, Platform Engineering — sind nicht das Zielbild. Sie sind die Daten-Infrastruktur, die Agents brauchen, um arbeiten zu können. Jede Stufe hat eigenständigen Wert. Jede Stufe macht die nächste möglich.

Und an jeder Stufe hilft AI — nur anders als gedacht.

Gleichzeitig schließt sich das Zeitfenster, in dem dieser Weg geordnet machbar ist. Drei Entwicklungen konvergieren gerade, die diesen Weg dringend machen.



Das Wissen geht — und es ist das Wissen, das Agents brauchen

In den nächsten 15 Jahren erreichen 13,4 Millionen Erwerbspersonen in Deutschland das Rentenalter (Destatis). In der Schweiz liegt die Ersatzquote für IT-Fachkräfte bei 38% (ICT-Berufsbildung, 2025).

In Österreich fehlen 28.000 IT-Spezialist:innen, projiziert 39.000 bis 2030 (WKO/UBIT). Das ist bekannt.

Was weniger diskutiert wird: Das Wissen, das mit diesen Menschen geht, ist genau das Wissen, das Agentic Infrastructure braucht.

Wenn Ihr:e Senior-Infrastruktur-Architekt:in geht, verlieren Sie nicht eine Person. Sie verlieren das Wissen darüber, warum das Netzwerk so segmentiert ist. Warum die Firewall-Regeln Ausnahmen haben. Warum das Monitoring genau diese Metriken zeigt. Warum das Deployment-Fenster da liegt, wo es liegt.

Dieses Wissen existiert in Köpfen, Shell-Histories und Slack-Nachrichten. In einer Form, die weder ein Junior lesen noch ein Agent verarbeiten kann.

Der neue Wissenstransfer sieht anders aus als Confluence-Seiten und Übergabe-Meetings:

Was Agents brauchen, damit sie funktionieren: Terraform-Module, die beschreiben, was gebaut wurde und warum. Mondoo-Policies, die kodifizieren, was compliant ist. Architektur-Specs als JSON-Schema, gegen die Discovery-Daten automatisch abgeglichen werden können. Szenarien, die testen, ob die Infrastruktur sich unter Last so verhält, wie erwartet.

Das sind keine abstrakten Formate. Das sind die Inputs für den Discovery-Verify-Generate-Loop. Und die einzigen Menschen, die dieses Wissen in diese Formate überführen können — weil sie wissen, warum die Infrastruktur so aussieht, wie sie aussieht — gehen gerade in Pension.

Jeder Monat, in dem dieses Wissen nicht kodifiziert wird, ist ein Monat, in dem es unwiederbringlich verloren geht. Nicht „schwer wiederzubeschaffen". Unwiederbringlich. Weil niemand sonst weiß, warum VLAN 42 existiert.

Der Wissenstransfer, der hier nötig ist, passiert nicht durch ein Seminar. Er passiert durch strukturierte Zusammenarbeit — Team Topologies nennt es „Enabling": Jemand arbeitet embedded mit Ihrem Team, nicht um die Infrastruktur zu bauen, sondern um die Fähigkeit zu transferieren, sie zu kodifizieren.

Seniors schreiben Specs, Policies, Szenarien — angeleitet, strukturiert, mit dem Ziel, dass am Ende das Wissen im Code steht und nicht nur im Kopf.

Die Frage für Ihre Organisation: *Wenn Ihre erfahrenste Infrastructure-Fachkraft morgen geht — wie viel ihres Wissens existiert in einer Form, die Nachfolgende UND ein Agent verarbeiten könnten?*

Die nächste Generation braucht andere Skills — und es gibt keine Ausbildung dafür

Tech-Stellenausschreibungen für Einsteiger:innen: Schweiz -46%, Österreich -34%, Deutschland -30% (Indeed/Euronews, 27-Länder-Analyse). Entry-Level-Einstellungen in Europa: -73,4% (Ravio, 350.000+ Mitarbeitende). Harvard (Hosseini Maasoum & Lichtinger, 2025): In AI-adoptierenden US-Firmen sank die Junior-Beschäftigung um 7,7% innerhalb von sechs Quartalen.

In der Infrastruktur trifft das besonders: Der klassische Einstieg — Junior-Admin, der Server aufsetzt und Backup-Prozeduren lernt — verschwindet. Wenn Terraform und Cloud-Konsolen die Basis-Tasks übernehmen, wo lernt ein Junior, was „unter der Haube" passiert?

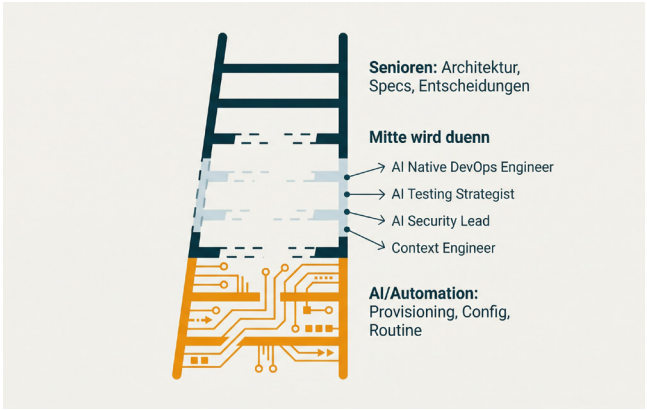
Aber die eigentliche Frage ist eine andere: Was müssen Ihre Leute in drei Jahren können?

Server-Provisioning wird Commodity — das übernehmen Agents.

Architektur-Verständnis, Domain-Wissen über Ihre spezifische Infrastruktur, die Fähigkeit einem Agent den richtigen Kontext zu geben — das bleibt beim Menschen. Und genau da liegen die neuen Rollen.

Vier Kompetenzcluster entstehen gerade — erkennbar an DORA 2024/2025, OWASP LLM Top 10 und dem MCP-Ökosystem:

- **AI Native DevOps Engineer:** Betreibt LLM-Gateway-Infrastruktur, MCP-Server, Agent-Sandboxing, AI Observability. Nicht ein neuer Hire — sondern das, was Ihr Senior DevOps Engineer werden muss.
- **AI Testing Strategist:** Baut Eval-Pipelines, die Agent-Output gegen deterministische Kriterien validieren. Scenario-based Validation statt Unit Tests. Nicht ein neuer Hire — sondern das, was Ihr QA-Lead oder Monitoring-Spezialist werden muss.
- **AI Security & Governance Lead:** NIS2, EU AI Act, AI-SBOM in CI/CD, Agent-Berechtigungen, Compliance-Automation. Die einzige Rolle mit mehr als 14.000 LinkedIn-Stellen weltweit — regulatorisch erzwungen, nicht absorbierbar.
- **Context Engineer:** Entscheidet, welchen Kontext ein Agent sieht — Architektur-Hypothese, Scan-Daten, Policy-Definitionen. MCP als Governance-Layer. Wird voraussichtlich in Platform Engineering absorbiert, muss aber als eigenständige Kompetenz erlernt werden.



Kein Major-Anbieter hat ein integriertes Curriculum dafür. Microsoft startet 2026 mit drei Beta-Zertifizierungen (AI-200, AI-300, SC-500). Das Zeitfenster, in dem diese Kompetenzen aufgebaut werden können, bevor Standardcurricula existieren: 12-18 Monate.

Manche Fähigkeiten werden durch AI entwertet — Server manuell konfigurieren zum Beispiel. Andere werden knapper denn je: Specs für Agents schreiben, Agent-Output bewerten, Architektur-Entscheidungen treffen.

Wer Agents steuern und ihre Arbeit bewerten kann, wird mehr gebraucht als je zuvor.

Die Frage für Ihre Organisation: *Wer in Ihrem Team lernt gerade die Skills, die in drei Jahren Kernkompetenz sein werden — und wer bildet diese Personen aus?*

ChatGPT kaufen löst es nicht — aber ohne AI geht es auch nicht

Hier wird es unangenehm. Mit Zahlen.

METR-Studie (arXiv:2507.09089): 16 erfahrene Entwickler:innen, 246 reale Tasks, randomisiert kontrolliert. Mit AI-Tools 19% langsamer. Konfidenzintervall: -2% bis -40%. Und: Selbst nach dem Experiment glaubten sie, 20% schneller gewesen zu sein.

DORA 2025: AI-Adoption korreliert mit 98% mehr Pull Requests. Lead Time unverändert. Mehr Output, nicht mehr Outcome. Faros AI (10.000+ Entwickler): 21% mehr Tasks, Delivery-Metriken flach, 9% mehr Bugs.

Das sind Software-Zahlen. Aber der Mechanismus ist derselbe: Ein Tool auf einen undokumentierten Prozess legen macht den Prozess nicht besser. AI auf manuelle Infrastruktur ist Stufe-1-Nutzung — und Stufe 1 bringt messbar negative Ergebnisse.

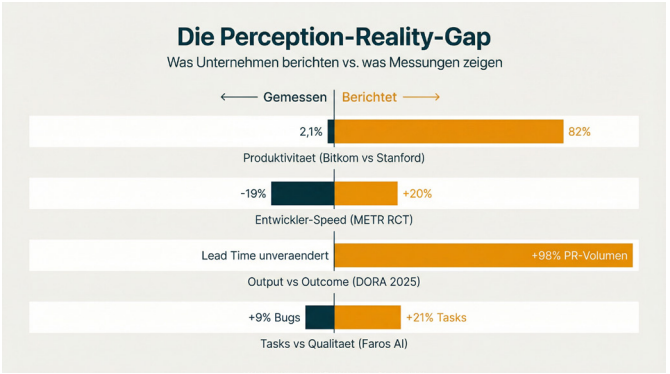
Die Gegenseite: TerminalBench 2.0 zeigt, dass der Rahmen allein — wie Tasks zerlegt, parallelisiert und validiert werden — 13,7 Prozentpunkte Leistungsunterschied erklärt. Mehr als ein Modellwechsel. Feldstudien: Deep Integration (angepasste Workflows, Spec-Driven Development, deterministische Validierung) bringt 25%+ Produktivitätsgewinn.

Der Unterschied ist nicht das Tool. Der Unterschied ist der Rahmen drumherum.

Und dieser Rahmen beginnt bei den Grundlagen.

Zwei Beispiele:

Config Management — ob Ansible, Puppet oder Chef — wächst in jeder Organisation über die Jahre zu einem Geflecht aus historisch gewachsenen Patterns: Source-Kompilierung statt Paketmanager, Custom-Logik, die niemand mehr versteht, hunderte Zeilen Code für etwas, das ein package-Statement löst.



Die Lösung ist radikal simpel: Config-Schicht so dünn wie möglich. Paketmanager statt Custom-Install. Golden Images mit Packer oder Container mit definierten Base Images — je nachdem, was in Ihrer Umgebung Sinn macht. Das reduziert Config-Code um 50-80% und Deployment-Zeiten um Größenordnungen.

Gleichzeitig schafft es die Basis für Supply-Chain-Security: Zentrale Paket-Repositories, signierte Artefakte, SBOM-Generierung, Mondoo/cnspec-Validierung. Ohne diese Basis ist Compliance-Automation ein leeres Versprechen.

Observability ist das zweite Fundament.

OpenTelemetry Traces können als unabhängige Validierungsschicht dienen: Hat der Agent die Infrastruktur tatsächlich in den gewünschten Zustand gebracht? Die Traces zeigen es — unabhängig davon, was der Agent behauptet.

Wer heute keine Observability hat, kann morgen keinen Agent-Output verifizieren.

Der Reskilling-Stau in Zahlen

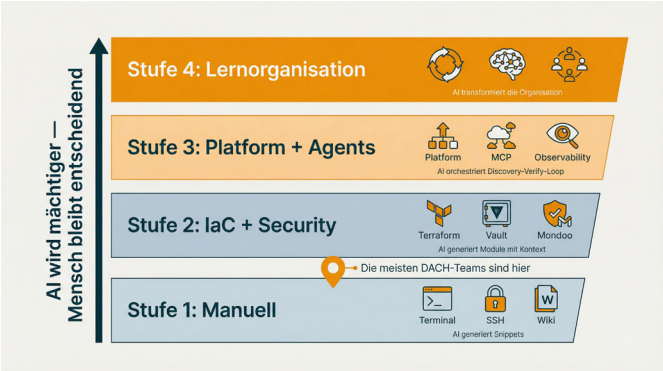
- 71% führen keine AI-Trainings durch.
  Quelle: TÜV, 2024, n=500
- 86% schöpfen AI-Potenzial zu maximal 25% aus.
  Quelle: Stifterverband/McKinsey, n=1.003
- 70% haben Rollen/Prozesse trotz AI nicht verändert.
  Quelle: dev-sync, 2026
- 78% nutzen AI auf eigene Faust (BYOAI).
  Quelle: Microsoft/LinkedIn, 2024
- Nur 7% haben GenAI unternehmensweit implementiert.
  Quelle: VDMA/Strategy&, n=247 DACH

Die Boston Consulting Group beschreibt, was AI-Transformation tatsächlich ausmacht — 10% Algorithmen, 20% Technologie, 70% Menschen und Prozesse. In der Praxis wird es systematisch invertiert: Budget fließt in Tool-Lizenzen. Nicht in die Befähigung, die Tools sinnvoll zu nutzen.

Aber: AI hilft an jeder Stufe — nur anders als erwartet

Die Lösung ist nicht „AI ignorieren bis die Grundlagen stehen".

Die Lösung ist: AI an jeder Stufe richtig einsetzen.



Stufe 1 → 2: AI generiert Terraform-Module, Ansible-Playbooks, Dockerfiles.

Ihr Team reviewt und lernt dabei. Copilot beschleunigt das Schreiben, ChatGPT erklärt, warum ein Modul so aussieht.

Aber: Die Kompetenz, den Output zu beurteilen, muss im Team wachsen. AI-generiertes Terraform ohne Review ist gefährlicher als manuell geschriebenes.

Stufe 2 → 3: AI leitet Mondoo-Policies aus Compliance-Dokumenten ab. AI generiert Vault-Konfigurationen aus Security-Specs. AI findet Architecture Gaps in Discovery-Daten.

Aber: Das Team muss verstehen, was eine gute Policy ist und was ein Gap bedeutet. Und das Team muss eine bewusste Entscheidung treffen: Werden wir das Enabling Team, das andere Teams befähigt — oder das Platform Team, das Self-Service bereitstellt (Team Topologies)?

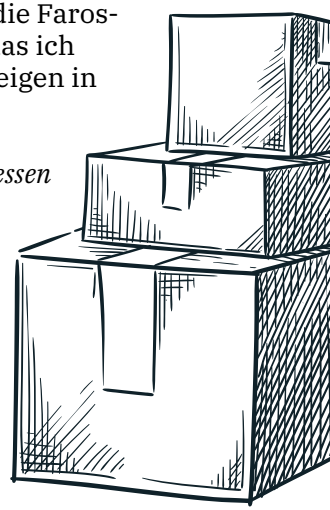
Stufe 3 → 4: AI orchestriert den Discovery-Verify-Generate-Loop. Agents scannen, vergleichen, generieren. Die Validierung läuft über Observability-Daten — OpenTelemetry Traces als externe Wahrheitsquelle, die der Agent nicht manipulieren kann.

Aber: Die Architektur-Hypothese schreibt ein Mensch. Die Szenarien definiert ein Mensch. Die Entscheidung, ob der Agent-Output in Produktion geht, trifft ein Mensch.

Was Stufensprünge blockiert:
Die häufigste Blockade ist keine technische. Teams bleiben stecken, weil niemand explizit Ownership für den nächsten Schritt übernimmt. IaC ist eingeführt — aber wer ist verantwortlich, dass es auch genutzt wird? Vault läuft — aber wer reviewed die Policies? Ohne bewusste Entscheidung, wer den nächsten Schritt führt, bleibt die Organisation auf der aktuellen Stufe. Nicht weil es zu schwer wäre, sondern weil es niemand als seine Aufgabe betrachtet.

Ab hier wird es spekulativ: Ob der Perception-Reality-Gap im DACH-Infrastructure-Bereich genauso massiv ist wie in der METR-Studie, weiß niemand. Aber die DORA-Daten, die Faros-Telemetrie und jedes Gespräch, das ich mit Infrastructure-Leads führe, zeigen in dieselbe Richtung.

Die Frage für Ihre Organisation: *Messen Sie die Wirkung Ihrer AI-Tools auf Release Frequency, Lead Time und Change Failure Rate? Wenn nein: Sie wissen nicht, ob AI Ihnen hilft oder schadet.*



Vier Stufen zur Agentic Infrastructure Organisation

Nicht als Roadmap. Als Standortbestimmung.

14 AI-Maturity-Modelle gibt es im DACH-Raum — keines davon misst, wie schnell Ihre Infrastructure-Organisation lernt (Schuster et al., 2021: 0 von 15 Modellen mit vollständiger Lern-/Wissensachse).

Unser Ansatz schließt die Lücke: Er misst nicht, welche Tools Sie haben, sondern ob Ihre Organisation bereit ist für den nächsten Schritt — und zeigt, wie AI diesen Schritt beschleunigt.

Stufe 1: Manuelle Infrastruktur → IaC-Grundlagen

„Wir machen das per Hand — und fangen an, es zu kodifizieren."

Wo AI schon hilft: AI generiert Terraform-Snippets, Ansible-Playbooks, Dockerfile-Entwürfe. 78% BYO-AI (Bring your Own-AI). Aber: 29,5% der AI-generierten Code-Snippets enthalten Sicherheitslücken (ACM TOSEM).

Was Sie brauchen: IaC als Team-Kompetenz. Config-Management so dünn wie möglich — Packer, Container. Vault für Secrets. Mondoo/cnspec für Security-Baselines.

Diagnose: *Könnte Ihr Team ein identisches Testsystem in unter einer Stunde provisionieren — reproduzierbar, ohne Login?*

Stufe 2: Workflow-Integration → Platform Foundations

„IaC funktioniert. Jetzt brauchen wir den organisatorischen Rahmen."

Wo AI stärker hilft: Ganze Module mit Kontext, Mondoo-Policies aus Compliance-Anforderungen, Vault-Konfigurationen aus Security-Specs. Feldstudien: 25%+ Produktivitätsgewinn bei Deep Integration.

Was Sie brauchen: Policy as Code (Mondoo, OPA/Sentinel). Vault mit dynamischen Credentials. GitOps (ArgoCD, Flux). NIS2-Compliance automatisch. Enabling Team oder Platform Team (Team Topologies)?

Diagnose: *Haben Sie Rollen und Prozesse wegen IaC verändert — oder nur das Werkzeug gewechselt?*

Stufe 3: Harness Engineering → Platform + Agent-Orchestrierung

„Wir bauen Rahmenwerke, die Agents nutzen können."

Wo AI zum Gamechanger wird: Discovery Agents scannen automatisch. Architecture Verification vergleicht Ist gegen Soll. Kontext-getriebene Generierung bringt 13,7 Prozentpunkte mehr als promptbasierte (TerminalBench 2.0). OpenTelemetry Traces als externe Validierung.

Was Sie brauchen: Internal Developer Platform. Agent-Pipelines. Observability für Agents. Security-by-Design: AI-SBOM, Agent Sandboxing, MCP-Governance.

Diagnose: *Können Sie einen Agent sicher auf Ihre Infrastruktur loslassen — mit Sandboxing und Output-Validierung?*

Stufe 4: Lernorganisation → Agentic Infrastructure

„Wir lernen schneller, als sich unsere Infrastruktur verändert."

Agents machen den Loop eigenständig — mit Human-in-the-Loop-Validierung. Senior-Wissen ist kodifiziert in Specs, Policies, Szenarien. Lerngeschwindigkeit ist KPI.

Diagnose: *Haben Sie einen Prozess, wie Infrastruktur-Wissen in Formate überführt wird, die Menschen und Agents nutzen können?*

Wo stehen Sie? Schnell-Diagnose in vier Fragen.

Beantworten Sie jede Frage ehrlich. Die erste, die Sie mit Nein beantworten, zeigt Ihren nächsten Schritt.

- *Kann Ihr Team ein Testsystem in unter einer Stunde neu aufsetzen — reproduzierbar, ohne manuelle Schritte?*
  Wenn nein: Stufe 1 ist Ihr nächster Schritt
- *Hat IaC die Rollen und Prozesse in Ihrem Team verändert — nicht nur die Tools?*
  Wenn nein: Stufe 2 ist Ihr nächster Schritt
- *Können Sie einen Agent sicher auf Ihre Infrastruktur loslassen — mit klarer Validierung des Outputs?*
  Wenn nein: Stufe 3 ist Ihr nächster Schritt
- *Ist Lerngeschwindigkeit ein KPI in Ihrem Team?*
  Wenn nein: Stufe 4 ist Ihr nächster Schritt

Die meisten DACH-Infrastruktur-Teams landen hier:
Stufe 1 — mit einzelnen Bereichen auf Stufe 2.

Was AI an jeder Stufe kann — und was nicht

Stufe 1 → 2
AI KANN: Terraform/Ansible generieren, Dockerfiles entwerfen

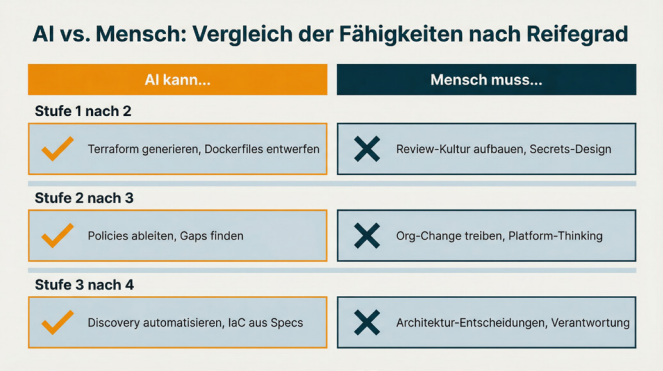
AI KANN NICHT: IaC-Kompetenz ins Team bringen, Review-Kultur aufbauen

Stufe 2 → 3
AI KANN: Policies ableiten, Vault-Configs generieren, Gaps finden

AI KANN NICHT: Org-Change treiben, Platform-Thinking einführen

Stufe 3 → 4
AI KANN: Discovery automatisieren, Remediation aus Specs generieren

AI KANN NICHT: Architektur-Entscheidungen treffen, Verantwortung übernehmen



AI macht an jeder Stufe die technische Arbeit schneller. Den organisatorischen Sprung muss der Mensch machen.

Der Weg, den ich bei jedem zweiten Kunden sehe

Ich leite Infralovers seit über 11 Jahren. Wir beraten, trainieren und begleiten Infrastructure-Teams im DACH-Raum mit HashiCorp-Tooling, Cloud Native, Security, und seit zwei Jahren: AI im Engineering-Kontext.

Das Muster ist fast immer dasselbe: Ein Infrastructure-Team — fünf, sechs Leute — betreut hunderte Systeme. Release-Zyklen in Wochen. Jenkins-Pipeline, die eine Person versteht. Compliance-Nachweis per Excel. Seniors, die alles im Kopf haben, gehen in zwei Jahren in Pension.

Der Weg heraus folgt einer klaren Logik.

Stufe 1 → 2: Infrastruktur wird Code
Der erste Schritt ist nie „AI einführen". Er ist: Infrastruktur wird Code. Terraform für Provisioning, Ansible für Konfiguration, Packer für Images, Container, wo es Sinn macht. Das Team lernt das gemeinsam. Vault für Secrets. Mondoo für Security-Baselines. Config-Management-Schulden aufräumen: so dünn wie möglich.

Allein dieser Schritt verkürzt Release-Zyklen typischerweise von Wochen auf Tage — nicht durch AI, sondern durch Reproduzierbarkeit. Und er schafft dabei die Datenbasis, die Agents auf den nächsten Stufen brauchen.

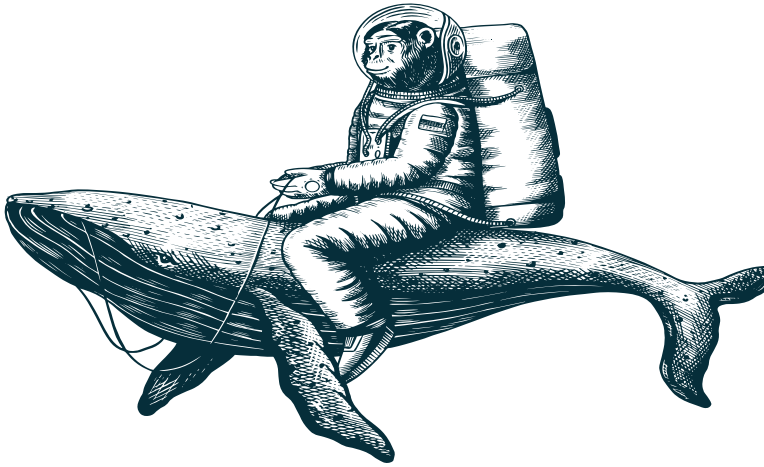
Stufe 2 → 3: Der organisatorische Rahmen
Jetzt kommen die schwierigeren Fragen: Wie geben wir den Entwicklern Self-Service, ohne dass Security auf der Strecke bleibt? Policy as Code, Guardrails, Platform-Denken. Welches Team verantwortet welche Infrastruktur-Domäne? Was ist Commodity und wird automatisiert, was bleibt beim Menschen? Werkzeuge wie Wardley Mapping und Domain-Driven Design helfen, diese Fragen zu beantworten. Aber der Sprung selbst ist kein Tool-Problem — es ist eine Entscheidung.

Stufe 3 → 4: AI übernimmt den Loop
Erst hier wird AI zum Gamechanger: AI übernimmt den Discovery-Scan, findet Drift, generiert IaC-Entwürfe aus dem Gap-Report. Der Mensch reviewed, korrigiert, entscheidet. Die Zeit und die Kompetenz verschiebt sich — weg von Server-Konfiguration, hin zu Architektur-Entscheidungen.

Die Quintessenz:
Der 10x Produktivitätszugewinn kommt nicht vom AI-Modell. Er kommt von der Fähigkeit, dem Modell zu sagen, was es tun soll — und zu überprüfen, ob es das getan hat. Das kann nur, wer die Grundlagen beherrscht.

Aber nicht jedes Team schafft den ganzen Weg. Manchmal fehlt das Management-Commitment. Und ob die StrongDM-Zahlen auf ein gewachsenes DACH-Enterprise übertragbar sind — StrongDM ist ein US-Startup, das bei null anfangen konnte, mit einem dreiköpfigen Team — bleibt eine offene Frage.

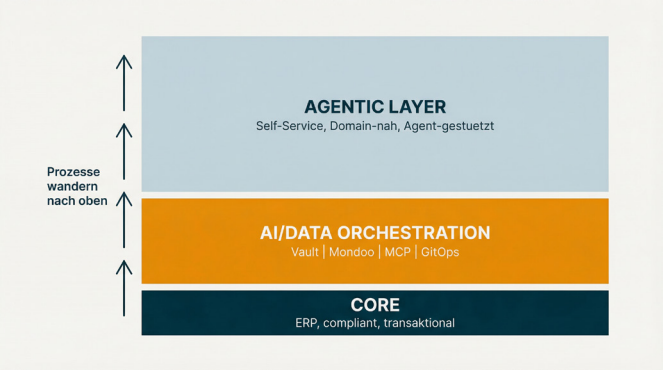
Aber die Richtung stimmt. Und jeder Schritt hat eigenständigen Wert.



Drei Schichten: Wo Agents in Ihre IT-Landschaft passen

Eine Architekturverschiebung definiert, wo die Infrastruktur-Arbeit der nächsten Jahre stattfindet. Sie verteilt sich auf drei Schichten — mit jeweils unterschiedlichen Anforderungen:

- **Der Core wird schlanker.** ERP, System of Record, hochgradig compliant, transaktional. Hier brauchen Agents die engsten Guardrails.
- **Der Agentic Layer wächst.** Self-Service-Plattformen, Agent-gestützte Workflows, Citizen Development nah am Domänenwissen.
- **Dazwischen: AI/Data Orchestration.** Internal Developer Platform. Vault als Identity-Layer. Mondoo als Compliance-Layer. MCP als Governance-Layer für Agent-Zugriffe.



Die Frage ist nicht „Welches Tool kaufe ich?“ Sondern: Welcher Prozess gehört in welche Schicht?

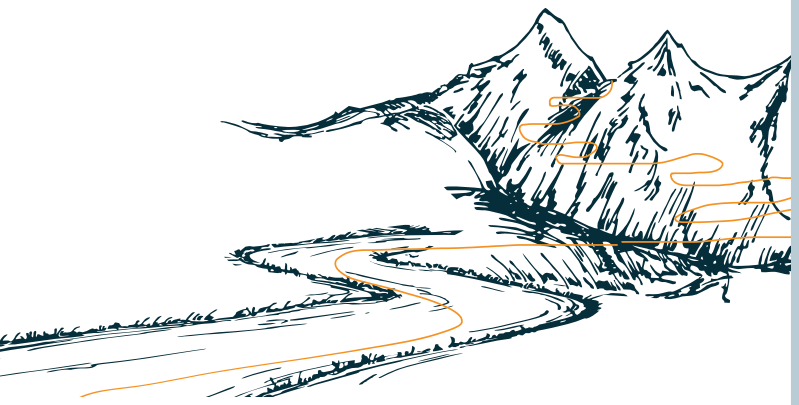
Organisationen auf Stufe 3 und 4 gestalten diese Schichtung bewusst. Organisationen auf Stufe 1 legen AI auf den bestehenden Stack. NIS2 (Deadline August 2026, ca. 29.500 Unternehmen betroffen, nur 12,1% compliant) macht diesen Zustand unhaltbar.

Und jetzt?

Das alles klingt nach viel. Und es ist viel — wenn man versucht, alles auf einmal anzugehen. Die meisten Teams, die scheitern, scheitern nicht an der Technologie. Sie scheitern daran, dass sie keinen konkreten nächsten Schritt definieren.

Die gute Nachricht: Sie müssen nicht bei Stufe 4 anfangen. Sie müssen nur wissen, wo Sie heute stehen — und was der eine Schritt ist, der Sie eine Stufe weiterbringt.

Das Modell ist klar. Die Daten sind eindeutig. Was fehlt, ist der erste konkrete Schritt — und jemand, der weiß, wo man anfängt. ▸



Wenn Sie sich wiederfinden

Vielleicht erkennen Sie sich hier wieder.

Ihr Release-Zyklus wird in Wochen gemessen. Der Weg zu Agents führt über Fundamente, die noch nicht stehen. Und ihre erfahrenste Person in Ihrem Infrastructure-Team geht in absehbarer Zeit — ihr Wissen in keiner Form kodifiziert, die Nachfolgende oder ein Agent nutzen könnten.

Sie wissen, dass „ChatGPT kaufen“ das Problem nicht löst. Und der Weg von Stufe 1 zu Stufe 4 sieht weit aus.

Er ist es nicht — wenn man weiß, wo man anfängt.

Ich beschäftige mich seit über 11 Jahren damit, wie Infrastructure-Teams lernen und wachsen. Der Weg zur Agentic Infrastructure ist real und machbar. Er beginnt nicht mit dem größten Tool, sondern mit dem ersten konkreten Schritt: Infrastruktur wird Code. Security wird Code. Platform wird Service. Und an jeder Stufe hilft AI — mehr und mehr, aber nie als Ersatz für die Grundlagen.

Die Transformation beginnt nicht mit einem Tool-Kauf. Sie beginnt mit einer ehrlichen Antwort auf die Frage: Wo stehen wir wirklich — und was ist unser nächster Schritt?

Lassen Sie uns das gemeinsam herausfinden.

ENABLEMENT + AI MAKES YOU STRONGER.

APE-ISH BUT IT WORKS

INFRALOVERS

Infralovers GmbH

Edmund Haselwanter
Janneckweg 3/3
8042 Graz
Österreich

ehaselwanter@infralovers.com
+43 676 3282554

www.infralovers.com