WHITEPAPER

Enterprise KI-Architektur 2026

Der Praxisleitfaden für deutsche Unternehmen

Für Vorstand und Geschäftsführung, CIO, CTO und CDO
sowie Enterprise Architekten
Version 1.1 · Oktober 2026

RLC
ROMAN LANGFELD CONSULTING

„Wer versteht, entscheidet.
Wer entscheidet, gestaltet."

Inhalt

Executive Summary 3

1 Warum 95 Prozent nicht reichen 5
1.1 Der Status quo: KI ist überall, Wirkung ist selten 5
1.2 Fünf Architekturfehler, die immer wiederkommen 6
1.3 Architektur als strategischer Hebel 8

2 Grundlagen: Enterprise AI Architecture verstehen 9
2.1 Was ist Enterprise AI Architecture? 9
2.2 Architekturprinzipien für KI-Systeme 11
2.3 Der TOGAF-Zyklus für KI-Projekte 14

3 Das Fundament: Daten und KI-Infrastruktur 16
3.1 Datenstrategie als KI-Strategie 16
3.2 Datenarchitektur-Muster für KI 18
3.3 KI-Infrastruktur: Cloud, Hybrid, On-Premise 20

4 Die Säulen: Architektur-Muster 2026 22
4.1 Generative KI: RAG und das, was danach kommt 22
4.2 Agentische KI: Vom Werkzeug zum Akteur 24
4.3 Die Enterprise AI Platform 27
4.4 Multi-Cloud, Hybrid und die Frage der Souveränität 28

5 Der Betrieb: MLOps und LLMOps im Produktionsalltag 31
5.1 MLOps: Von Ad-hoc zu industrialisierter KI 31
5.2 ML Lifecycle Management 35
5.3 GenAI in Produktion: Die besonderen Herausforderungen 38

6 Governance und Compliance: Architektur unter regulatorischem Druck 42
6.1 EU AI Act: Architektur-Implikationen 42
6.2 Das Governance-Rahmenwerk im Unternehmen 47
6.3 Sicherheit in der KI-Ära: neue Angriffsflächen und NIS-2 50
6.4 Datenschutz und DSGVO im KI-Kontext 53

7 Der Weg: Roadmap für deutsche Unternehmen 56
7.1 Der RLC AI Architecture Readiness Check 56
7.2 Drei Architektur-Pfade nach Unternehmensgröße 58
7.3 Vier Meilensteine bis zum produktionsreifen KI-Betrieb 61
7.4 Die häufigsten Fallstricke und wie man sie vermeidet 64

8 Fazit und Ausblick 67
8.1 Fünf Prinzipien für 2026 67
8.2 Was nach 2026 kommt 68
8.3 Vom Whitepaper zur Umsetzung 69

Quellen 70

Executive Summary

Eine ehrliche Lage

Mehr als die Hälfte der Unternehmen in Deutschland setzt heute Künstliche Intelligenz ein: 57 Prozent, fast dreimal so viele wie vor zwei Jahren.[1] Eine Zahl, die nach Aufbruch klingt. Dieselbe Erhebung klingt an anderer Stelle nüchterner. Kein einziges der befragten Unternehmen mit KI im Einsatz gibt an, deren Potenzial vollständig zu nutzen, 59 Prozent nutzen es nach eigener Einschätzung überhaupt nicht.[1] Eine breit referenzierte Erhebung von MIT Project NANDA kommt zu dem Ergebnis, dass rund 95 Prozent der untersuchten Organisationen aus ihren Investitionen in generative KI keinen messbaren Ertrag ziehen.[2] Der AI Index der Stanford University findet messbare Produktivitätsgewinne in eng zugeschnittenen Aufgaben und nennt die Belege auf gesamtwirtschaftlicher Ebene noch früh und uneinheitlich.[3] Adoption und Wirkung sind, kurz gesagt, zwei verschiedene Dinge.

Warum ist das so? Auf diese Frage gibt es drei seriöse Antworten, und sie wirken in den meisten Unternehmen gleichzeitig. Die erste hat mit Wirtschaftlichkeit zu tun: Viele Projekte starten ohne klaren Wertbeitrag, werden nach Eindruck priorisiert, ohne Geschäftsfall, und scheitern still, weil niemand definiert hat, woran Erfolg eigentlich zu erkennen wäre. Die zweite hat mit Menschen zu tun: KI verändert Arbeit, und ein Unternehmen, das Belegschaft, Prozesse und Führung nicht mitnimmt, hat am Ende ein technisches Werkzeug, das niemand bedient. Die dritte ist die unterschätzte: KI verlangt eine Umgebung, in der sie funktionieren kann. Daten, Schnittstellen, Betrieb, Governance. Wir nennen das **KI-Architektur**.

Dieses Whitepaper widmet sich der dritten Dimension. Die beiden anderen bekommen im öffentlichen Diskurs viel Aufmerksamkeit, die Architekturfrage fällt typischerweise unter den Tisch. Das bleibt so bis zu dem Moment, in dem ein Pilot in Produktion gehen soll und niemand verstanden hat, was das eigentlich heißt.

57 %	59 %	95 %	26 %
der Unternehmen in Deutschland setzen KI ein	davon schöpfen das Potenzial nach eigener Einschätzung nicht aus	der untersuchten Organisationen ohne messbaren Ertrag aus GenAI	Produktivitätsgewinn in der Softwareentwicklung
Bitkom	Bitkom	MIT Project NANDA	Stanford AI Index

Abbildung 1 · Ausgangslage in vier Kennzahlen. Belege siehe Quellen 1 bis 3.

Drei Fragen, eine Perspektive

Das Whitepaper beantwortet drei Fragen. Erstens: Was meinen wir, wenn wir von Enterprise KI-Architektur sprechen, und an welcher Stelle wirkt sie? Zweitens: Welche Architekturmuster sind 2026 für deutsche Unternehmen relevant, von RAG über agentische Systeme bis zu MLOps? Drittens: Wie führt der Weg von einer Architektur-Vision zu einer produktionsreifen KI-Plattform, pragmatisch und anschlussfähig an das, was im Unternehmen schon läuft?

Architektur macht dabei die wirtschaftliche Bewertung erst belastbar und gibt der organisatorischen Befähigung ein Ziel. Drei Dimensionen, eine Antwort: Eine KI-Einführung braucht alle drei.

Fünf Thesen, die diese Arbeit trägt

These 1. KI scheitert an drei Dingen gleichzeitig: an unklaren Geschäftsfällen, an unterschätztem Wandel, an fehlender Architektur. Die Technik ist selten die Ursache. Ein Programm, das nur an einem dieser Hebel ansetzt, bewegt nichts.

These 2. Der Mittelstand braucht andere Architekturen als der Konzern. Referenzarchitekturen großer Hyperscaler sind für Organisationen mit dediziertem Data-Science-Apparat gedacht. Mittelständische Unternehmen brauchen modulare, schlanke Architekturen, die mit dem bestehenden IT-Team umsetzbar sind, ohne Verzicht auf Skalierbarkeit.

These 3. Agentische KI verändert die Architekturgrundlagen. 2026 ist das Jahr, in dem autonome KI-Agenten beginnen, den Experimentierstatus zu verlassen. Das verlangt ein neues Muster: ein Geflecht spezialisierter Agenten, mit zentraler Governance und dezentraler Ausführung.

These 4. Governance ist ein Architekturprinzip. EU AI Act, NIS-2 und DSGVO sind Designvorgaben, die von Beginn an in die Architektur gehören. Ein Unternehmen, das sie so behandelt, baut einmal und erfüllt mehrere Regelwerke zugleich.

These 5. Die Datenstrategie ist die KI-Strategie. Kein Modell ist besser als die Daten, auf denen es arbeitet. Data Mesh, Feature Stores und Datensouveränität gehören zum Fundament. Fehlende Datenbereitschaft hält Unternehmen häufig davon ab, ihre Piloten zu skalieren.[4]

Für wen ist dieses Whitepaper geschrieben?

Für C-Level- und Board-Verantwortliche reichen Executive Summary sowie die Kapitel 1, 6, 7 und 8. Für CTO, CIO und Enterprise Architekten ist das Dokument vollständig gedacht. CDOs und Data-Verantwortliche finden ihre Schwerpunkte in den Kapiteln 1, 3, 4 und 6. IT-Projektleitung in 3, 4, 5 und 7. Jede Lesergruppe findet einen Einstieg, der sie ernst nimmt, und keinen, der sie überfordert.

KAPITEL 1

1 Warum 95 Prozent nicht reichen

1.1 Der Status quo: KI ist überall, Wirkung ist selten

Im September 2026 meldete der Bitkom, dass erstmals die Mehrheit der deutschen Unternehmen Künstliche Intelligenz einsetzt: 57 Prozent, nach 36 Prozent im Vorjahr und 20 Prozent zwei Jahre zuvor.[1] Weitere 38 Prozent planen oder diskutieren den Einsatz. Die Zahl wurde als Durchbruch gelesen: endlich produktiv, endlich angekommen.

Und dann liest man weiter. Kein einziges Unternehmen mit KI im Einsatz gibt in derselben Erhebung an, deren Potenzial vollständig zu nutzen, 59 Prozent schöpfen es nach eigener Einschätzung überhaupt nicht aus.[1] Dazu passt eine zweite Studie.

Eine im Sommer 2025 veröffentlichte Erhebung von MIT Project NANDA, breit zitiert auch in der überregionalen Presse, kommt zu einem ernüchternden Befund: Rund 95 Prozent der untersuchten Organisationen ziehen aus ihren Investitionen in generative KI keinen messbaren Ertrag, die große Mehrheit der Piloten bleibt ohne messbaren Beitrag zum Geschäftsergebnis.[2] Die Studie stützt sich auf über 300 öffentlich bekannte Initiativen, Interviews mit Vertretern von 52 Organisationen und Antworten von 153 Führungskräften und ist für diese schmale Basis kritisiert worden. Als Größenordnung bleibt sie ein ernstzunehmendes Signal. Ein im Umfeld der Harvard Business Review veröffentlichter Beitrag nennt die mangelnde Datenbereitschaft als häufigen Grund, warum Unternehmen ihre Vorhaben mit generativer KI nicht skalieren.[4] Der Stanford AI Index 2026 findet messbare Produktivitätsgewinne in strukturierten, gut messbaren Aufgaben, etwa 14 bis 15 Prozent im Kundenservice und 26 Prozent in der Softwareentwicklung. Die Belege auf gesamtwirtschaftlicher Ebene nennt er noch früh und uneinheitlich.[3]

Die Befunde ergeben ein Bild: Adoption steigt schnell, Wirkung folgt langsamer. McKinsey hat dafür den Begriff *Gen AI Paradox* geprägt: Fast acht von zehn Unternehmen setzen generative KI ein, und fast ebenso viele berichten von keiner wesentlichen Wirkung auf ihr Ergebnis.[5] Ob „Paradox" das richtige Wort ist, sei dahingestellt, denn was hier geschieht, lässt sich erklären. Ein genauer Blick zeigt drei wiederkehrende Ursachen, die sich in den meisten Unternehmen überlagern.

Die erste ist wirtschaftlicher Natur. Viele Initiativen starten, weil ein Tool spannend wirkt, ein Wettbewerber es einsetzt oder ein Vorstand die Etage angeschoben hat. Was fehlt, ist ein nüchterner Geschäftsfall. Wofür genau wollen wir KI einsetzen? Welcher Prozess wird wie viel schneller, günstiger, besser? Wie messen wir das? Ohne diese Klärung verschwindet ein Pilot still in der Bilanz für „Innovationsausgaben" und niemand kann am Ende sagen, ob er sich gelohnt hat. Der Stanford AI Index 2026 hält fest, dass die Produktivitätsgewinne dort am größten sind, wo die Arbeit strukturiert und ihr Ergebnis messbar ist, und dass sie gering oder sogar negativ ausfallen, wo Werkzeug und Aufgabe schlecht zusammenpassen.[3] Eine gekaufte Lizenz und eine Townhall stellen diese Passung nicht her.

Die zweite ist organisatorischer Natur. KI verändert Arbeit, und Arbeit verändert sich nicht, weil eine Software installiert wurde. Sie verändert sich, wenn Menschen verstehen, wie sich ihre Rolle verschiebt, wenn Führungskräfte den Wandel aktiv gestalten und wenn Prozesse entsprechend angepasst werden. Unter den Unternehmen, die noch keine KI einsetzen, nennen 85 Prozent fehlendes technisches Know-how zur Implementie-

rung als Grund, deutlich vor rechtlichen Unklarheiten (66 Prozent) und schwer kalkulierbaren Kosten (63 Prozent).[1] Das überrascht wenig. Überraschend wäre eher, wenn ein Werkzeug, dessen Bedienung Übung verlangt, ohne Übung gute Ergebnisse liefern würde.

Die dritte ist architektonischer Natur. KI braucht eine Umgebung, in der sie funktionieren kann. Daten, die verfügbar und vertrauenswürdig sind. Schnittstellen, die das Ergebnis dorthin tragen, wo eine Entscheidung getroffen wird. Monitoring, das erkennt, wenn das Modell sich von der Wirklichkeit entfernt. Governance, die regelt, wer was darf, und das im Code festhält, über das Protokoll hinaus. Eine Plattform, auf der nicht jeder Anwendungsfall sein eigenes Improvisationstheater aufführt. Das ist Architektur, und sie ist der Faktor, der in der öffentlichen Debatte am stillsten geblieben ist.

Diese drei Ursachen sind nicht gleich häufig, aber sie sind selten allein anzutreffen. Ein Projekt mit unklarem Geschäftsfall scheitert oft auch an der Befähigung der Anwender. Ein Projekt mit guter Architektur, aber ohne Wertbeitragsklarheit, läuft technisch sauber und betriebswirtschaftlich ins Leere. Eine KI-Einführung kommt um alle drei Dimensionen nicht herum. Dieses Whitepaper konzentriert sich auf die dritte, weil über sie im Moment am wenigsten nachgedacht wird. Für die Wirtschaftlichkeit hat sie nach unserer Einschätzung den größten Einfluss auf Erfolg oder Misserfolg.

Es lohnt sich, an dieser Stelle eine Unterscheidung einzuziehen, die im laufenden Diskurs gern verschwimmt und die McKinsey in derselben Veröffentlichung schärft.[5] **Horizontale Anwendungsfälle** sind die Copiloten, Chatbots und Schreibassistenten, die mittlerweile in den meisten Unternehmen im Einsatz sind. Sie sind günstig zu integrieren, sie produzieren spürbare Effizienzgewinne im Arbeitsalltag des Einzelnen, und sie sind aus exakt diesem Grund die niedrigste Stufe eines KI-Reifegrads. Sie kosten Lizenz und etwas Schulung. Sie verändern aber keine Wertschöpfungslogik.

Vertikale Anwendungsfälle sind die andere Geschichte. Eine KI, die im Vertriebsprozess Angebote priorisiert. Eine, die in der Produktion Anomalien erkennt, bevor sie zu Ausfällen werden. Eine, die im Schadenfall Versicherungsfälle vorqualifiziert. Diese Anwendungen versprechen den eigentlichen Hebel, und sie scheitern am häufigsten. Sie verlangen Datenzugriff über Systemgrenzen, Integration in operative Prozesse, regulatorische Absicherung, kontinuierlichen Betrieb. Sie verlangen alle drei Dimensionen: Geschäftsfall, Befähigung, Architektur. Und sie sind der Grund, warum dieses Whitepaper existiert.

Ein einfacher Test: Lassen Sie sich von Ihrem Team die letzten fünf KI-Initiativen aufschreiben. Markieren Sie jene, die im Tagesgeschäft eines Sachbearbeiters wirken, und jene, die produktiv mit dem Kernsystem des Unternehmens verbunden sind. In den meisten Unternehmen ist die zweite Spalte deutlich kürzer als die erste. Sie zeigt, wo der Hebel liegt. Sie zeigt auch, wo die eigentliche Arbeit noch zu tun ist.

Im folgenden Abschnitt sehen wir uns an, woran das im Detail liegt. Welche fünf Architekturfehler sich in deutschen Unternehmen mit beklagenswerter Regelmäßigkeit wiederholen, und warum ein weiteres Pilotprojekt sie nicht behebt. Behoben werden sie durch eine andere Art zu denken.

1.2 Fünf Architekturfehler, die immer wiederkommen

Gespräche mit IT-Leitungen und Enterprise Architekten fördern eine erstaunliche Wiederholung zutage. Unabhängig von Branche und Größe tauchen dieselben fünf Probleme auf, sobald offen über KI-Vorhaben gesprochen wird. Keines davon ist neu, keines ist exotisch, und keines lässt sich durch ein weiteres Modell oder einen weiteren Anbieter beheben.

Erstens: KI als Aufsatz. Viele Unternehmen behandeln Künstliche Intelligenz wie eine zusätzliche Anwendung, die man neben ERP, CRM und HR-System installiert. Eine Insel, betreut von einem kleinen Team, das den Rest des Unternehmens nicht erreicht. Das funktioniert für horizontale Werkzeuge, also Copiloten und Schreibassistenten, durchaus. Es funktioniert nicht, sobald KI etwas tun soll, das in den Kernprozessen zählt. Eine Insel kann keinen Lieferprozess verändern. Sie kann ihn allenfalls kommentieren.

Zweitens: Daten als Nachgedanke. In den meisten Architektur-Diskussionen über KI taucht das Wort „Daten" überraschend spät auf. Erst wird über Modelle gesprochen, dann über Plattformen, dann irgendwann über die Daten, die da hineinfließen sollen. In der Praxis kehrt sich die Reihenfolge um: Datenqualität, Datenverfügbarkeit und Datengovernance entscheiden zuerst, was überhaupt möglich ist. Die Modellwahl kommt später. Ein im Umfeld der Harvard Business Review veröffentlichter Beitrag nennt die Datenbereitschaft als häufigen Grund, warum generative KI nicht über den Piloten hinauskommt.[4] Eine Investition in der falschen Reihenfolge baut Schichten, die später nicht tragen.

Drittens: Einkauf vor Architektur. Eine erstaunlich häufige Konstellation: Ein Anbieter zeigt eine beeindruckende Demo, der Einkauf verhandelt einen Vertrag, das Projekt startet. Niemand hat geprüft, wie die Lösung mit der bestehenden Landschaft zusammenarbeitet. Im Lauf des Projekts stellt sich heraus, dass das gekaufte System Datenformate erwartet, die im Unternehmen nicht existieren, oder eine Cloud-Region nutzt, die regulatorisch problematisch ist, oder eine Lizenzlogik hat, die mit dem geplanten Skalierungspfad kollidiert. Architektur kommt nicht nach dem Einkauf. Sie kommt davor.

Viertens: Governance als Anhang. EU AI Act, NIS-2, DSGVO und sektorspezifische Regularien werden in vielen Unternehmen als Pflichtprogramm behandelt, das nach der eigentlichen Entwicklung kommt. Ein Projekt läuft, die Compliance-Abteilung schaut hinterher drauf, irgendetwas wird angepasst. Das Verfahren ist teuer und es ist riskant. Teuer, weil nachträgliche Eingriffe in laufende Systeme aufwendiger sind als saubere Designentscheidungen am Anfang. Riskant, weil die Verstöße, die der EU AI Act bei verbotenen Praktiken mit Bußgeldern von bis zu 35 Millionen Euro oder sieben Prozent des weltweiten Jahresumsatzes belegt, bei anderen Pflichtverletzungen mit bis zu 15 Millionen Euro oder drei Prozent, in der gelebten Praxis entstehen.[6] Eine Governance, die erst am Ende eingebaut wird, ist meistens auch erst am Ende verstanden worden.

Fünftens: Pilot ohne Plan. Das letzte Muster ist das verbreitetste. Ein Pilotprojekt wird aufgesetzt, läuft technisch sauber, wird intern als Erfolg vermeldet. Dann passiert nichts mehr. Es gibt keinen Plan, wie aus dem Piloten Produktion wird, weil niemand bei der Konzeption darüber nachgedacht hat. Welche Schnittstellen müssten gebaut werden? Welche Last muss die Plattform tragen, wenn statt 50 Nutzern 5.000 darauf zugreifen? Wer betreibt das System dauerhaft, mit welchen SLAs, zu welchen Kosten? Diese Fragen werden im Piloten elegant umgangen. In der Produktion lassen sie sich nicht mehr umgehen. Ein Pilot, der vom ersten Tag an so gebaut wird, als müsse er morgen in Produktion gehen, ist teurer in der Erstellung und sehr viel billiger in der Skalierung.

Diese fünf Muster wirken zusammen, und sie verstärken sich gegenseitig. Eine KI als Aufsatz, die mit Daten arbeitet, an die niemand gedacht hat. Beschafft bei einem Anbieter, dessen Lösung mit der Governance nicht harmoniert. Im Piloten gebaut, als müsse niemand morgen damit produktiv arbeiten. Das ist der Regelfall.

Die gute Nachricht: Alle fünf Fehler sind vermeidbar, und zwar durch ein anderes Denken. Genau das ist Aufgabe einer Enterprise KI-Architektur.

1.3 Architektur als strategischer Hebel

Was ist Enterprise KI-Architektur eigentlich? Eine kurze Antwort: die Summe der Entscheidungen, die bestimmen, ob KI in einem Unternehmen funktionieren kann. Und ob sie dauerhaft funktionieren kann, auch wenn sich Anbieter, Modelle, Vorschriften und Anwendungsfälle ändern. Das klingt abstrakt. Es lässt sich konkretisieren.

Architektur in diesem Sinne ist ein Bündel von Festlegungen, die in Schichten denken. Diagramme und Dokumente halten sie nur fest. Welche Daten sind die Grundlage, woher kommen sie, wie werden sie gepflegt, wie werden sie geschützt? Welche Plattform trägt die KI-Anwendungen, wer betreibt sie, wer entwickelt darauf? Welche Muster wenden wir an, wenn ein neuer Anwendungsfall hinzukommt? Bauen wir alles neu, oder gibt es eine Vorlage? Welche Regeln gelten für Modelle, für Daten, für Nutzungen, und wie sind diese Regeln im System verankert, über ein PDF hinaus? Wer baut die Brücke zwischen einer KI-Idee und dem Tag, an dem die Anwendung im Einsatz ist und Mehrwert liefert?

Architektur entscheidet darüber, ob der zweite, der fünfte und der zwanzigste Anwendungsfall mit vertretbarem Aufwand möglich sind. Der einzelne Anwendungsfall braucht sie kaum. Genau hier liegt der ökonomische Hebel. Ein erstes Pilotprojekt ist mit jeder beliebigen Konstellation machbar. Der zehnte produktive Anwendungsfall ist es nur mit einer Architektur, die diesen Skalierungspfad mitdenkt. Eine frühe Investition zahlt sich in niedrigeren Stückkosten pro Anwendungsfall aus. Eine späte baut für jeden Fall die halbe Plattform neu.

Es gibt drei Eigenschaften, die eine gute KI-Architektur von einer mittelmäßigen unterscheiden. Sie ist **modular**, also so geschnitten, dass einzelne Bestandteile ausgetauscht werden können, ohne das Ganze umzubauen. Sie ist **anbieterneutral**, also nicht so tief in eine Plattform eingelassen, dass ein Wechsel undenkbar wird. Sie ist **beobachtbar**, also so gebaut, dass man sieht, was sie tut, wenn sie etwas tut, und merkt, wenn sie etwas tut, was sie nicht tun sollte. Diese drei Eigenschaften sind im Pilotmodus verzichtbar. Im Produktivbetrieb sind sie der Unterschied zwischen einem System, das fünf Jahre trägt, und einem, das nach zwei Jahren neu gemacht werden muss.

Damit ist auch die Grenze gezogen. Strategie, Geschäftsfall und die Befähigung der Menschen bleiben eigene Aufgaben. Architektur ist das Fundament, auf dem sie stehen. Auf einem schlechten Fundament kann auch eine gute Strategie nicht halten. Aber ein gutes Fundament ohne Strategie steht leer.

In den folgenden Kapiteln gehen wir diesen Stoff systematisch durch. Kapitel 2 klärt die Grundlagen, was Enterprise AI Architecture im Kern bedeutet und welche Prinzipien tragen. Kapitel 3 zoomt auf das Fundament, die Daten- und Infrastrukturfragen. Kapitel 4 stellt die wichtigsten Architektur-Muster für 2026 vor, von RAG über agentische Systeme bis zur Enterprise AI Platform. Kapitel 5 widmet sich dem Betrieb, MLOps und LLMOps. Kapitel 6 behandelt Governance und Compliance als Architekturthemen. Kapitel 7 zeigt die konkreten Wege je nach Unternehmensgröße. Kapitel 8 fasst zusammen und blickt voraus.

Eilige Leser können nach diesem Kapitel direkt zu Kapitel 6 oder 7 springen. Für ein tiefes Verständnis der Architektur empfiehlt sich die Reihenfolge der Kapitel; sie bauen aufeinander auf.

KAPITEL 2

2 Grundlagen: Enterprise AI Architecture verstehen

Bevor wir in den folgenden Kapiteln über Daten, Architektur-Muster, Betrieb und Governance sprechen, lohnt sich ein Schritt zurück. Was meinen wir eigentlich, wenn wir von Enterprise AI Architecture sprechen? Welche Begriffe tragen, welche tragen nur scheinbar, und an welcher Stelle hilft die etablierte Architekturpraxis weiter, an welcher muss sie für KI nachjustiert werden?

Dieses Kapitel klärt drei Dinge. Erstens den Begriff selbst, einschließlich der Frage, wie sich Enterprise AI Architecture in das größere Feld der Unternehmensarchitektur einordnet. Zweitens die Prinzipien, die eine tragfähige KI-Architektur von einer gebauten unterscheiden. Drittens das methodische Vorgehen: wie ein etablierter Architekturansatz wie TOGAF auf KI-Vorhaben angewendet wird und an welchen Stellen KI ihre eigenen Regeln mitbringt.

Kapitel 1 hat drei Architektur-Qualitäten als Mindestmaß genannt: modular, anbieterneutral, beobachtbar. Sie kehren in diesem Kapitel als Designprinzipien wieder, ergänzt um zwei weitere, die in Kapitel 1 nur gestreift wurden. Das Kapitel lässt sich auch ohne Kapitel 1 lesen.

2.1 Was ist Enterprise AI Architecture?

Eine Definition vorab, so kompakt wie möglich: **Enterprise AI Architecture ist die Disziplin, die festlegt, wie Künstliche Intelligenz in die Unternehmensarchitektur eingebettet wird, damit sie wiederholbar, sicher und wirtschaftlich Wert erzeugen kann.** Wiederholbar, weil ein einzelner Anwendungsfall keine Architektur verlangt. Sicher, weil KI in Prozesse greift, die reguliert, sensibel oder beides sind. Wirtschaftlich, weil der Hebel von KI in der Skalierung liegt.

Diese Definition unterscheidet sich von dem, was im Markt häufig mit dem Wort gemeint ist. Mit „AI Architecture" ist nicht selten ein Diagramm mit einer Vektordatenbank, einem Sprachmodell und einem Pfeil dazwischen gemeint. Das ist eine **Lösungsarchitektur** für eine einzelne KI-Anwendung. Enterprise AI Architecture liegt eine Ebene darüber. Sie beschreibt die Umgebung, in der zehn, zwanzig oder fünfzig solcher Anwendungen entstehen, betrieben und kontrolliert werden können.

Einordnung in die Unternehmensarchitektur

Die Architektur-Community hat für diese Ebene seit den 1990er Jahren ein Vokabular entwickelt. TOGAF, das *Open Group Architecture Framework*, ist der verbreitetste Bezugsrahmen, ArchiMate seine grafische Sprache.[7] In der Zusammenarbeit mit Enterprise Architekten führt an diesen Begriffen kein Weg vorbei, und es lohnt, sie kurz aufzurufen, weil sie auch für KI gelten.

TOGAF unterscheidet vier Architektur-Domänen. Die **Business Architecture** beschreibt, wie das Unternehmen Wert schafft: Prozesse, Rollen, Fähigkeiten. Die **Data Architecture** beschreibt, welche Informationen das Unternehmen nutzt und wie sie organisiert sind. Die **Application Architecture** beschreibt die Anwendungslandschaft, die diese Prozesse trägt. Die **Technology Architecture** beschreibt die technische Grundlage darunter: Server, Netzwerke, Plattformen.

KI zieht quer durch alle vier Domänen, eine fünfte braucht es dafür nicht. Eine KI-Anwendung verändert einen Geschäftsprozess (Business), arbeitet auf bestehenden oder neuen Datenbeständen (Data), wird als Anwendung gebaut und integriert (Application) und läuft auf einer Infrastruktur, die andere Anforderungen stellt als klassische Geschäftsanwendungen (Technology). KI ist ein Querschnittsthema, und genau deshalb fällt sie ohne dedizierte Architekturarbeit zwischen die Stühle.

Der AI Architecture Stack

Damit zur praktischen Frage: Wie lässt sich diese Querschnittsarbeit ordnen, ohne das gesamte TOGAF-Inventar aufrufen zu müssen? In der Praxis hat sich ein Schichtenmodell bewährt, das die KI-spezifischen Bestandteile in eine nachvollziehbare Reihenfolge bringt. Wir nennen es den AI Architecture Stack. Sechs Schichten, von unten nach oben gelesen, vom Boden bis zum Geschäftsnutzen.

Ganz unten liegt die **Infrastrukturschicht**. Hier wohnen Rechenleistung, Speicher und Netzwerk, in einer Form, die KI-Lasten trägt: GPUs oder spezialisierte Beschleuniger für Training und Inferenz, Speicherklassen für große Datenmengen, Netzanbindungen, die nicht zum Engpass werden. Für die meisten Unternehmen ist diese Schicht Cloud, für einige hybrid, für wenige rein on-premise. Architekturentscheidungen auf dieser Schicht sind selten reversibel ohne Schmerzen, weshalb sie früh und mit Bedacht getroffen werden müssen.

Darüber liegt die **Datenschicht**. Sie umfasst die Systeme, in denen Daten gespeichert, aufbereitet und bereitgestellt werden: Data Warehouses, Data Lakes, Lakehouses, Feature Stores, Vektordatenbanken. Die Datenschicht ist nicht KI-spezifisch, sie wird es aber durch zwei Anforderungen, die andere Anwendungen seltener stellen: kontinuierliche Verfügbarkeit großer Mengen für Training, niedrige Latenz für Inferenz. Kapitel 3 vertieft, was hier zu beachten ist.

Auf der Datenschicht steht die **Modell- und Plattformschicht**. Hier liegen die Werkzeuge, mit denen Modelle entwickelt, trainiert, versioniert und bereitgestellt werden: Notebooks und Entwicklungsumgebungen, Trainings-Frameworks, Modell-Registries, Inferenz-Endpunkte. In der GenAI-Welt kommt eine zweite Familie hinzu, die zunehmend gleichgewichtig wird: Foundation Models über APIs, Prompt-Management, Vektor-Index-Pflege, Embedding-Pipelines. Diese Schicht entscheidet darüber, ob Data Scientists und Anwendungsentwickler produktiv arbeiten oder ihre Zeit darauf verwenden, Werkzeuge, Umgebungen und Schnittstellen immer wieder neu zusammenzusetzen.

Darüber kommt die **Orchestrierungsschicht**. Sie verbindet, was sonst nebeneinander stehen würde: Pipelines, die Daten aus dem Quellsystem über die Aufbereitung bis ins Modell führen; Workflows, die mehrere Modelle hintereinanderschalten; Agentenlogiken, die Werkzeuge in einer geplanten Reihenfolge aufrufen. Die Orchestrierungsschicht ist im Pilotbetrieb verzichtbar, weil ein Pilot meist ein einzelner Pfad ist. Im Mehrfach-Use-Case-Betrieb ist sie das Stück, das aus einer Sammlung Einzelanwendungen ein System macht.

Darüber liegt die **Anwendungs- und Erlebnisschicht**. Hier sitzt das, was Nutzer sehen: Chatbots, Suchassistenten, eingebettete Funktionen in bestehenden Anwendungen, eigenständige Oberflächen. Diese Schicht ist die sichtbarste, sie wird in der internen Kommunikation am häufigsten genannt, und sie ist die, in der die wenigsten Architekturentscheidungen fallen. Ein Programm, das hier die meiste Aufmerksamkeit investiert, baut Pilotprojekte. Eines, das die Schichten darunter ernst nimmt, baut Architektur.

Quer durch alle fünf Schichten zieht sich die **Governance- und Sicherheitsschicht**. Sie hält alle Etagen zusammen: Identitäten und Berechtigungen, Verschlüsselung, Audit-Spuren, Risikoklassifikationen, technische Dokumentation, Richtlinien, die im Code stehen. In den meisten Pilot-Architekturen fehlt diese Schicht. In jeder produktiven Architektur ist sie unverzichtbar. Kapitel 6 widmet sich ihr ausführlich.

Das Schichtenmodell dient dazu, eine Architekturdiskussion zu ordnen. Es hilft zu klären, auf welcher Ebene eine Entscheidung gerade getroffen wird, und welche anderen Ebenen davon berührt sind. Eine Wahl auf der Infrastrukturschicht hat Folgen für die Plattformschicht. Eine Entscheidung in der Modellschicht hat Folgen für die Daten- und für die Governance-Schicht. Eine Architektur, die das Modell isoliert betrachtet, passt im zweiten Jahr nicht mehr.



Von unten nach oben gelesen: vom Fundament zum Geschäftsnutzen

Abbildung 2.1 · Der AI Architecture Stack. Fünf Schichten von der Infrastruktur bis zur Anwendung, Governance und Sicherheit als Querschnitt.

Was Enterprise AI Architecture nicht ist

Eine Definition gewinnt an Schärfe, wenn man sagt, was sie ausschließt. Drei Abgrenzungen sind hier nützlich.

Enterprise AI Architecture ist erstens **kein Modellkatalog**. Die Frage, welches Foundation Model das beste ist, gehört in die Modellauswahl für einen konkreten Anwendungsfall. Architektur sorgt dafür, dass Modelle ausgetauscht werden können.

Sie ist zweitens **keine Werkzeugliste**. Eine Aufzählung der Lieblings-Tools eines Teams ist eine Inventarliste. Werkzeuge sind die Materialisierung von Entscheidungen, die auf einer abstrakteren Ebene getroffen wurden. Ein Vorgehen, das mit Werkzeugen beginnt, hat die Reihenfolge umgedreht.

Sie ist drittens **kein einmaliges Artefakt**. Eine Architekturdokumentation, die 2025 erstellt und 2027 unverändert auf dem SharePoint liegt, ist 2027 ein historisches Dokument. Architektur ist eine fortgesetzte Tätigkeit, weil sich Anbieter, Modelle, Vorschriften und Anwendungsfälle weiterentwickeln. Eine Architektur, die nicht atmet, lebt auch nicht.

2.2 Architekturprinzipien für KI-Systeme

Architekturprinzipien sind die Festlegungen, die alle weiteren Entscheidungen rahmen. Sie wirken als Leitplanken. Eine Architektur, die sie missachtet, funktioniert im Einzelfall und versagt als System.

In Kapitel 1 haben wir drei Qualitäten als Mindestmaß genannt: modular, anbieterneutral, beobachtbar. Sie kehren in diesem Abschnitt wieder, ergänzt um zwei weitere Prinzipien, die für KI besonders zählen. Fünf Prinzipien, jedes mit einem klaren Inhalt, jedes mit einer klaren Konsequenz.

Modularität und Separation of Concerns

Eine modulare Architektur ist so geschnitten, dass sich Bestandteile ändern lassen, ohne das Ganze umzubauen. In der KI-Welt heißt das konkret: Das Modell ist austauschbar, ohne dass die Anwendung neu gebaut werden muss. Die Vektordatenbank ist austauschbar, ohne dass die Indizierungspipeline neu geschrieben werden muss. Der Foundation-Model-Anbieter ist austauschbar, ohne dass jeder Prompt von Hand übersetzt werden muss.

Das gelingt durch konsequente *Separation of Concerns*: Jede Komponente tut eine Sache und hat klare, schmale Schnittstellen zu den anderen. Was banal klingt, ist im Bau seltener umgesetzt, als es sein müsste. Eine typische Verfallsform: Das Modell ist so tief in die Anwendung eingelassen, dass ein Wechsel des Anbieters einen erheblichen Umbau bedeutet, der dann doch nicht stattfindet. Modularität ist die Versicherung gegen Entscheidungen, die heute richtig sind und morgen falsch.

Vendor-Neutralität und Exit-Strategien

Anbieterneutralität bedeutet, sich nur so tief zu binden, dass ein Wechsel möglich bleibt. Jede ernsthafte KI-Architektur enthält Anbieterprodukte, denn der gegenteilige Anspruch, alles selbst zu bauen, ist für die wenigsten Unternehmen ökonomisch sinnvoll. Die Frage ist, an welcher Stelle die Bindung eingegangen wird und an welcher Stelle Spielraum bleibt.

Drei konkrete Stellschrauben haben sich in der Praxis bewährt. Erstens offene Datenformate für die Daten, die das Modell ein- und ausgeben, etwa Parquet oder Iceberg für strukturierte Daten und ONNX für portable Modelle. Zweitens offene Standards und verbreitete De-facto-Standards für die Anwendungslogik, etwa OpenAI-kompatible APIs für LLMs, MLflow-konforme Modellregister und für die Anbindung von Werkzeugen an Agenten das Model Context Protocol (Kapitel 4.2). Drittens vertragliche und technische Exit-Pläne, die dokumentiert sind und regelmäßig geprobt werden. Vendor Lock-in ist akzeptabel, solange er bewusst, dokumentiert und überprüft ist. Wenn er sich heimlich aufbaut, beginnt das Problem.

Observability by Design

Eine KI-Anwendung, die nicht beobachtet wird, wird mit der Zeit messbar schlechter, oft ohne dass es jemandem auffällt. Eine in *Scientific Reports* veröffentlichte Studie hat 128 Kombinationen aus Modell und Datensatz aus vier Branchen untersucht und in 91 Prozent der Fälle eine Verschlechterung über die Zeit festgestellt.[8] Das gilt für klassische Machine-Learning-Modelle durch Drift in den Eingangsdaten. Für GenAI-Anwendungen kommen Veränderungen am Modell, an der Quellenstruktur und an der Nutzungsweise hinzu.

Observability by Design heißt: Die Beobachtung wird von Anfang an mitgebaut. Telemetrie gehört in die Modellschicht, sobald sie steht; Evaluierungs-Sets entstehen mit der Anwendung, lange bevor der erste Kunde reklamiert. Drift-Erkennung und Alarmierung gehören in die Plattform. Im Pilotmodus kann man auf all das verzichten. Im Produktivbetrieb entscheidet es darüber, ob ein Unternehmen sein System steuert oder ihm bei der Arbeit zusieht und hofft.

Security und Privacy by Design

Sicherheit und Datenschutz sind in der klassischen Anwendungsentwicklung lange als nachgelagerte Disziplinen behandelt worden. In KI-Anwendungen ist das nicht mehr tragfähig, aus zwei Gründen. Erstens sind die Angriffsvektoren neu: *Prompt Injection*, bei der Nutzereingaben das Modell zu unerwünschtem Verhalten verleiten, *Data Poisoning*, bei dem Trainings- oder Retrieval-Daten manipuliert werden, *Model Inversion*, bei der aus den Antworten auf die Trainingsdaten zurückgeschlossen wird. Die Taxonomie des US-amerikanischen National Institute of Standards and Technology (NIST) für Angriffe auf maschinelles Lernen ordnet diese Klassen systematisch; die ENISA hatte die KI-spezifische Bedrohungslage für Europa bereits 2020 kartiert.[9] Zweitens fließen in KI-Anwendungen häufig personenbezogene oder geschäftskritische Daten ein. Wie sie verarbeitet werden, muss vor der Inbetriebnahme entschieden und dokumentiert sein.

Privacy by Design ist im Übrigen seit Mai 2018 als Pflicht in der DSGVO verankert (Art. 25).[10] Was in der Verordnung „Datenschutz durch Technikgestaltung" heißt, beschreibt genau das, was hier gemeint ist: Schutzmaßnahmen werden mit der Architektur entworfen und mit ihr gebaut. Ein späteres Nachziehen bedeutet, zweimal zu bauen. Für KI-Architektur heißt das konkret: Datenminimierung in der Aufbereitung, Pseudonymisierung in der Trainingspipeline, Zugriffskontrollen auf der Retrieval-Ebene, dokumentierte Verarbeitungszwecke pro Anwendung. Ein Versäumnis an dieser Stelle wird im Audit zum Architekturproblem, das sich mit Dokumentation nicht reparieren lässt.

Skalierbarkeit von Anfang an

Das fünfte Prinzip wird am häufigsten missverstanden. Falsch verstanden führt es am ersten Tag zu einer Architektur für eine Million Nutzer. Richtig verstanden führt es zu einer Architektur, die für die Million offen ist, sollte sie kommen.

Es geht um **Designentscheidungen**, weniger um Dimensionierung. Eine Anwendung, die auf einem einzelnen Server läuft, kann dort vollkommen passend sein. Erst die Frage, ob sie auch wachsen kann, ohne von Grund auf neu gebaut zu werden, entscheidet, ob ihr Entwurf trägt. Zustandslose Komponenten, asynchrone Verarbeitung wo möglich, klare Trennung von Training und Inferenz, getrennte Skalierung von Datenpfad und Modellpfad: das sind Entscheidungen, die früh getroffen werden und später den Unterschied machen.

Skalierbarkeit ist auch eine ökonomische Frage. Eine Anwendung, deren Inferenzkosten linear mit der Nutzung steigen, ist im Pilotbetrieb günstig und im Produktivbetrieb ein Problem. Ein Programm, das den Kostenpfad nicht von Beginn an mitdenkt, wird spätestens auf dem Weg zu den Meilensteinen M3 und M4 der in Kapitel 7 beschriebenen Roadmap unangenehm überrascht.

Wie die fünf Prinzipien zusammenwirken

Die fünf Prinzipien wirken als Bündel. Modularität ohne Unabhängigkeit vom Anbieter verschafft nur einen Teil der Bewegungsfreiheit, die sie verspricht. Skalierbarkeit ohne Privacy führt in eine Compliance-Lage, die sich mit einem früheren Blick hätte vermeiden lassen. Die Beispiele lassen sich beliebig fortsetzen, und genau das ist der Punkt: Jedes Prinzip steht für sich, aber tragfähig wird die Architektur erst in ihrer Kombination.

Was die fünf Prinzipien verbindet, ist eine Haltung zur Zukunft. Architektur trifft Entscheidungen über etwas, das man nicht kennt. Sie tut das, indem sie Optionen offen hält, wo Veränderung wahrscheinlich ist, und Festlegungen trifft, wo Verbindlichkeit Sicherheit gibt. Das Schwere an dieser Disziplin ist die Bereitschaft, beim Entwerfen ehrlich zu beantworten, an welcher Stelle man heute schon weiß, was morgen noch gelten wird. Das Wissen über die Prinzipien ist der leichtere Teil.

2.3 Der TOGAF-Zyklus für KI-Projekte

Bis hierher haben wir über Begriffe und Prinzipien gesprochen. Bleibt die Frage nach dem Vorgehen. Wie geht man als Enterprise Architekt ein KI-Vorhaben an, und an welcher Stelle unterscheidet sich das von einem klassischen IT-Projekt?

Die Architektur-Community arbeitet seit Jahrzehnten mit einem methodischen Vorgehen, das sich auch für KI bewährt: der **Architecture Development Method**, kurz ADM, dem Herzstück von TOGAF.[11] Der ADM-Zyklus beschreibt acht Phasen (A bis H), von der Architekturvision über die Entwicklung der vier Architektur-Domänen bis zur Steuerung von Implementierung und Architekturänderungen, eingerahmt von einer vorbereitenden Phase und einem fortlaufenden Anforderungsmanagement. Er ist als iterativer Rahmen angelegt, mit bewussten Rückkopplungen zwischen den Phasen.

Wir verzichten an dieser Stelle auf eine Phase-für-Phase-Wiedergabe; sie steht im Standard selbst. Interessanter ist die Frage, was KI an dem etablierten Vorgehen verschiebt. Drei Verschiebungen sind aus unserer Sicht zentral.

Iterativität als Normalfall

Klassische Architektur-Projekte durchlaufen den ADM-Zyklus oft in großen Bögen: eine Architecture Vision wird formuliert, die Architekturen entwickelt, die Implementierung gesteuert, dann beginnt der nächste Zyklus. Bei KI funktioniert das nicht. Die Halbwertszeit der Annahmen ist zu kurz. Ein Foundation Model, das im Februar die richtige Wahl war, kann im August von einer leistungsfähigeren oder günstigeren Variante abgelöst sein. Eine Regulierung, die in der Vision noch als Aussicht behandelt wurde, kann noch vor Abschluss des Zyklus als Pflicht gelten.

Der Methodenrahmen bleibt. Was sich ändert, ist die Größe der Zyklen. KI-Vorhaben durchlaufen den ADM-Zyklus in kürzeren Schleifen als klassische Architekturprojekte. Eine angemessene Größe für eine Iteration ist ein einzelner Anwendungsfall: Die Iteration endet mit seinem produktiven Betrieb und mit der Anpassung der Architektur an die Erkenntnisse daraus.

Datenabhängigkeit verändert die Reihenfolge

In klassischen Projekten beginnt die Architekturarbeit häufig mit der Business und der Application Architecture. Die Data Architecture folgt, oft als Konsequenz aus den Anwendungen, die gebraucht werden. Bei KI funktioniert diese Reihenfolge nur eingeschränkt. Datenqualität, Datenverfügbarkeit und Datengovernance entscheiden zuerst, was überhaupt möglich ist. Eine schöne Anwendungsarchitektur auf einem schlechten Datenfundament ist die häufigste Form, in der KI-Vorhaben in den Sand laufen.

Damit verlieren Business und Application keine Bedeutung. Die Data Architecture rückt nur früher und gewichtiger in den Zyklus. In manchen Unternehmen beginnt der Zyklus faktisch mit der Datenarchitektur und entwickelt von dort die anderen Domänen. Ein Unternehmen, das den klassischen TOGAF-Pfad strikt einhält, sollte zumindest eine substanzielle Daten-Iteration vor die ersten produktiven Anwendungen ziehen.

Nicht-Determinismus verlangt neue Akzeptanzkriterien

Die dritte Verschiebung ist die schwierigste, weil sie an der Wurzel der ingenieurtechnischen Tradition rührt. Klassische Software ist deterministisch: Bei gleicher Eingabe kommt das gleiche Ergebnis heraus. Das macht Akzeptanztests einfach: Soll-Werte definieren, Ist-Werte messen, vergleichen, fertig.

KI-Anwendungen, besonders generative, verhalten sich anders. Die gleiche Eingabe kann zu Antworten führen, die in Wortwahl, Reihenfolge oder Detailtiefe variieren. Diese Variabilität ist eine Eigenschaft des Systems. Architektur und Methodenrahmen müssen darauf reagieren, und zwar an drei Stellen. Erstens bei der **Akzeptanzdefinition**: An die Stelle einer Soll-Ausgabe tritt ein Spektrum akzeptabler Antworten, oft über ein Eval-Set mit Referenzen und einer Bewertung der Abweichungen. Zweitens beim **Testen**: Die einmalige Prüfung weicht einer kontinuierlichen Evaluierung gegen ein Set, das sich mit den Anwendungsfällen weiterentwickelt. Drittens bei der **Verantwortung**: Eine Architektur muss klar regeln, an welchen Stellen ein Mensch im Loop bleibt, weil die Toleranz für Variabilität dort gleich null ist, und an welchen Stellen Variabilität akzeptabel oder sogar erwünscht ist.

Das ist eine kulturelle Verschiebung in einer Disziplin, die sich über Jahrzehnte darauf trainiert hat, Verhalten festzulegen. KI-Architektur ohne diese Verschiebung führt entweder zu Sicherheitskäfigen, die jeden Nutzen ersticken, oder zu Systemen, die im Audit niemand verantworten kann.

Architecture Vision für KI-Plattformen

Eine letzte Bemerkung zum Beginn des Zyklus, der **Architecture Vision**. In klassischen Projekten beschreibt sie das gewünschte Zielbild und die strategische Begründung dafür. Bei KI-Vorhaben kommt eine Anforderung hinzu, die unterschätzt wird: Die Vision muss eine Aussage darüber treffen, **welche Klasse von Anwendungen** die Architektur tragen soll. Eine Architektur für drei Copilot-Anwendungen sieht anders aus als eine für drei agentische Systeme im Vertrieb. Eine Architektur, die beides bedienen soll, ist teurer und braucht andere Trennlinien.

In der Praxis hat sich bewährt, in der Architecture Vision eine **Anwendungs-Typologie** zu formulieren: Welche zwei oder drei Anwendungsklassen soll die Architektur bis zum Regelbetrieb tragen, also bis zum Meilenstein M4 in Kapitel 7.3? Daraus leiten sich die Anforderungen an die Plattform ab, daraus die Auswahl der Werkzeuge, daraus die Investitionspfade. Eine abstrakte KI-Vision ohne diese Typologie führt entweder zu wenig oder zu viel Plattform. Beides ist teuer.

Wohin Kapitel 2 trägt

Drei Klärungen, drei Werkzeuge. Der **Begriff** Enterprise AI Architecture, der das einzelne Anwendungs-Diagramm hinter sich lässt und die Umgebung für viele Anwendungen meint. Die **fünf Prinzipien**, die jede Architekturentscheidung rahmen: Modularität, Anbieterneutralität, Observability, Security und Privacy by Design, Skalierbarkeit von Anfang an. Und der **ADM-Zyklus**, in seiner für KI angepassten Form: kürzer in den Iterationen, datenseitig früher gewichtet, mit Akzeptanzkriterien, die mit Variabilität umgehen können.

Diese drei Klärungen sind die Voraussetzung für die folgenden Kapitel. Kapitel 3 vertieft die Datenschicht, das Fundament jeder KI-Architektur. Kapitel 4 stellt die Architektur-Muster vor, mit denen die hier skizzierten Prinzipien in konkrete Anwendungen übersetzt werden. Kapitel 5 zeigt, wie aus einer Architektur ein Betrieb wird. Kapitel 6 nimmt sich die Governance- und Sicherheitsschicht vor, die in diesem Kapitel als Querschnitt eingeführt wurde. Kapitel 7 schließt mit den konkreten Pfaden für die unterschiedlichen Unternehmensgrößen.

Damit ist der Begriffsraum sortiert. Was folgt, ist die Arbeit am Stoff.

KAPITEL 3

3 Das Fundament: Daten und KI-Infrastruktur

Es gibt einen Satz, den Unternehmen in jeder KI-Diskussion irgendwann sagen, meistens, wenn das erste Pilotprojekt enttäuscht hat. Er lautet: „Wir müssen erst unsere Daten in Ordnung bringen." Der Satz stimmt. Ernst gemeint ist er selten. In Ordnung bringen klingt nach einem Frühjahrsputz, nach einem abgeschlossenen Projekt für ein motiviertes Team. Datenarbeit für KI ist eine Dauerleistung, die in die Architektur der Organisation eingebaut werden muss. Ein Ticket reicht dafür nicht.

Dieses Kapitel beschreibt die Schicht, auf der jede KI-Anwendung steht. Es geht um drei nüchterne Fragen. Wie wird aus einer Sammlung von Tabellen und Dokumenten ein Material, das eine KI-Anwendung tragen kann? Welche Muster haben sich für die Datenschicht großer Unternehmen bewährt, und welche überfordern den Mittelstand? Und wo steht die Infrastruktur in einem Markt, in dem Hyperscaler, europäische Anbieter und eigenes Rechenzentrum jeweils gute Argumente und ehrliche Schwächen haben?

3.1 Datenstrategie als KI-Strategie

Warum Datenqualität über den Erfolg jedes KI-Vorhabens entscheidet

Die Stärke eines KI-Systems ist nach oben durch die Qualität seiner Daten begrenzt. Das ist eine alte Beobachtung aus dem klassischen maschinellen Lernen, und sie gilt für generative KI in einer leicht verschobenen Form weiter. Wenn ein Sprachmodell halluziniert, liegt die Ursache in Unternehmensanwendungen meist im Kontext, der dem Modell zur Verfügung steht, seltener in einem zu schwachen Modell: in Dokumenten, deren Versionen nebeneinander existieren, in Tabellen, deren Spaltenbedeutung nirgendwo dokumentiert ist, in Wissensbeständen, deren Pflege niemand verantwortet. Die Forschung zu produktiven ML-Systemen beschreibt Daten und ihre Abhängigkeiten seit Langem als die größte Quelle versteckter technischer Schulden.[12]

Datenqualität ist dabei ein Bündel aus mehreren Eigenschaften, die selten gleichzeitig hoch sind. Die einschlägigen Standards des Daten-Managements kennen je nach Zählung sechs bis zwölf Dimensionen; fünf davon sind für KI-Anwendungen die operativ entscheidenden.[13] **Vollständigkeit** beschreibt, ob die Daten überhaupt da sind, die für die Frage gebraucht werden. **Aktualität** beschreibt, ob sie den heutigen Stand abbilden oder einen, den niemand mehr meint. **Konsistenz** beschreibt, ob dasselbe Geschäftsobjekt in zwei Systemen dasselbe heißt und das Gleiche bedeutet. **Korrektheit** beschreibt, ob das, was im Datensatz steht, der Wirklichkeit entspricht, die er abbilden soll. **Eindeutigkeit** beschreibt, ob ein Sachverhalt einmal oder ungewollt mehrfach im Bestand liegt. Erst aus diesen fünf zusammen entsteht das, was ein KI-System verwerten kann, ohne dass die Anwendung zur Glückssache wird.

Eine sechste Dimension verdient für KI eine eigene Erwähnung: die **Nachvollziehbarkeit der Herkunft**. Sie beschreibt, ob für einen Datensatz dokumentiert ist, woher er stammt und welche Transformationen er bis zur produktiven Nutzung durchlaufen hat. Für KI-Systeme wird sie zur regulatorischen Pflicht, weil der EU AI Act für Hochrisikoanwendungen genau diese Dokumentation verlangt.[14] Ein Hochrisikosystem muss die Herkunft seiner Daten deshalb von Anfang an als Architekturanforderung mitführen.

Der praktische Punkt für die Architektur: Diese Eigenschaften lassen sich nicht nachträglich für einen Datensatz ermitteln, indem man eine Pipeline davorschaltet. Sie müssen an der Quelle entstehen oder dort, wo die Daten in einen produktiven Bestand übernommen werden. Eine KI-Architektur, die auf einer Datenschicht aufsetzt, in der diese fünf Eigenschaften nicht gemessen werden, baut auf einem Boden, der bei der ersten Last nachgibt.

Data Governance entscheidet, was die KI sagt

Datenqualität ohne klare Verantwortung verfällt. Das ist die unspektakuläre Erkenntnis aus zwei Jahrzehnten Data-Warehouse-Geschichte, und unter generativer KI gilt sie umso mehr. Ein Sprachmodell, das aus dem Wissensbestand des Unternehmens antwortet, verstärkt jeden Fehler in diesem Bestand. Eine veraltete Preisliste in einem PDF wird zu einer veralteten Preisauskunft in einem Vertriebs-Chat. Ein widersprüchlicher Vertragsstandard wird zu einer widersprüchlichen Vertragsempfehlung. Die Anwendung ist nur so gut wie die Quelle, aus der sie schöpft.

Data Governance bezeichnet die Antwort darauf: Wer ist verantwortlich für welchen Datenbestand, mit welchen Befugnissen und welchen Pflichten? In der Form, in der die DAMA-Vereinigung sie seit Jahren beschreibt, kennt Data Governance vier wiederkehrende Rollen: einen **Data Owner**, der für die fachliche Definition und Qualität eines Bestands geradesteht, meist auf Bereichsleitungs-Ebene; einen **Data Steward**, der diese Definition operativ durchsetzt; einen **Data Custodian**, der die technische Pflege übernimmt; und ein **Data Council** oder **Governance Board**, das übergreifende Konflikte entscheidet.[15] Eine KI-Architektur, die ohne diese Rollen aufgesetzt wird, ist eine technische Lösung für ein organisatorisches Problem.

Es ist deshalb ein konkretes Architekturprinzip dieses Kapitels: Bevor eine KI-Anwendung an einen Datenbestand angeschlossen wird, muss für diesen Bestand ein Owner benannt sein, der unterschreibt, dass die Pflege dauerhaft gesichert ist. Das klingt bürokratisch und ist es teilweise. Es ist aber die einzige Stelle in der ganzen Architektur, an der Verantwortung nicht in Code übersetzbar ist. Das Ergebnis dieser Auslassung zeigt sich, sobald die produktive Anwendung schlechter antwortet als am ersten Tag und niemand weiß, wer es richten soll.

Daten-Souveränität im europäischen Kontext

Die dritte Dimension der Datenstrategie ist eine Frage des Rechtsrahmens und der Verarbeitungslage, in dem die Daten verwendet werden. Für deutsche Unternehmen ist das keine theoretische Überlegung. Die DSGVO regelt seit 2018, unter welchen Bedingungen personenbezogene Daten verarbeitet werden dürfen, und sie regelt das mit erheblichem extraterritorialem Anspruch.[16] Der EU AI Act überlagert das mit zusätzlichen Anforderungen an Datenqualität und Dokumentation für Hochrisikosysteme. Der Digital Omnibus, seit dem 27. Juli 2026 als Verordnung (EU) 2026/1744 in Kraft, hat diese Pflichten verschoben: für eigenständige Hochrisikosysteme nach Anhang III vom 2. August 2026 auf den 2. Dezember 2027, für in regulierte Produkte nach Anhang I eingebettete Systeme vom 2. August 2027 auf den 2. August 2028.[17] Die Verschiebung verschafft Vorlaufzeit und lässt die Anforderungen selbst unverändert. Ein System, das heute entworfen und nach diesen Stichtagen in Verkehr gebracht wird, muss sie von Anfang an erfüllen. Der Data Governance Act und der Data Act ergänzen den Rahmen für Datenräume und Datenteilung in Europa und bilden die Grundlage für branchenübergreifende Datennutzung.[18]

Für die Architektur folgen daraus drei konkrete Festlegungen, die früh getroffen werden müssen. Erstens die Frage des **Speicher- und Verarbeitungsorts**: Welche Daten dürfen in welchen Regionen verarbeitet werden, und welche Modelldienste sind unter dieser Festlegung überhaupt nutzbar? Zweitens die Frage der **Vertrags-**

lage mit Anbietern: Welche Auftragsverarbeitung gilt, welche Unterauftragnehmer sind erlaubt, welche Trainingsnutzung der eigenen Daten durch den Anbieter ist ausgeschlossen? Drittens die Frage der **Nachvollziehbarkeit**: Wer kann zu welchem Zeitpunkt belegen, welches Datum in welche Antwort eingeflossen ist? Diese drei Festlegungen sind Architekturparameter. Sie bestimmen mit, welche Muster aus dem nächsten Kapitel überhaupt in Frage kommen, und ein Legal-Team kann sie im Nachgang nicht mehr abhaken.

Eine pragmatische Konsequenz daraus, die schnell übersehen wird: Die Arbeit mit personenbezogenen oder strategisch sensiblen Daten verlangt eine bewusste Entscheidung über Verarbeitungsorte. Ein eigenes Rechenzentrum ist dafür eine Möglichkeit unter mehreren. Eine EU-Region eines Hyperscalers mit klar dokumentierter Datenresidenz, ein europäischer Anbieter oder eine Gaia-X-konforme Kombination kommen ebenso in Frage.[19]

3.2 Datenarchitektur-Muster für KI

Lake, Warehouse, Lakehouse: drei Antworten auf eine alte Frage

Bevor man über KI-spezifische Muster redet, lohnt ein nüchterner Blick auf die Datenhaltung. Die drei gängigen Bauformen sind seit Jahren beschrieben, und sie haben jede ihre Daseinsberechtigung. Ein **Data Warehouse** ist eine strukturierte Sammlung kuratierter Daten, modelliert für bekannte Auswertungsfragen, mit klarer Schema-Disziplin und hoher Antwortqualität auf das, was geplant wurde. Ein **Data Lake** ist das Gegenteil davon: Rohdaten in beliebigem Format, mit minimaler Vorab-Verdichtung, als Ressource für Anwendungen, deren Anforderungen noch nicht feststehen. Ein **Data Lakehouse** versucht beides, also Roh-Verfügbarkeit mit Schema-Disziplin und Transaktionsfähigkeit auf einer gemeinsamen Datenebene, prominent vertreten durch Tabellenformate wie Delta Lake, Apache Iceberg und Apache Hudi.[20]

Für KI-Vorhaben hat das Lakehouse-Muster einen handfesten Vorteil: Trainings- und Inferenzdaten müssen oft sowohl roh, also für explorative Modellentwicklung, als auch kuratiert, also für reproduzierbares Training und für Reporting, zur Verfügung stehen. Zwei getrennte Plattformen mit eigener ETL-Strecke dafür zu betreiben, ist teuer und führt zu Abweichungen zwischen den Beständen, die später in widersprüchlichen Auswertungen sichtbar werden. Eine gemeinsame Tabellenformat-Schicht löst das auf der Speicherebene, ohne dass die fachlichen Konsumenten gemeinsame Werkzeuge nutzen müssten.

Warehouse und Lake bleiben im Mittelstand trotzdem nützlich. Ein bestehendes Warehouse mit funktionierender Auswertung sollte nicht abgelöst werden, weil eine KI-Initiative anrollt. Die realistische Antwort ist meistens eine Ergänzung: Das Warehouse bleibt der Ort für vertraute Kennzahlen, ein neuer Bereich, oft auf einem Objektspeicher mit Tabellenformat, wird für die KI-relevanten Daten aufgebaut, und beide Bereiche werden über klar dokumentierte Schnittstellen verbunden. Das ist weniger ambitioniert als ein vollständiger Lakehouse-Umbau, dafür risikoärmer und bezahlbarer.

Data Mesh: dezentrale Verantwortung, zentrale Disziplin

Mit wachsender Größe trifft jede zentrale Datenplattform auf eine Grenze, die nicht technischer Natur ist. Es ist die Grenze des zentralen Teams, das alle fachlichen Bedeutungen zugleich kennen müsste, um die Plattform sinnvoll zu pflegen. **Data Mesh**, in der Beschreibung von Zhamak Dehghani, dreht das Modell um: Die Verantwortung für einen Datenbestand wandert zu jenen Bereichen, die die fachliche Sprache dafür besitzen. Vertrieb verantwortet Vertriebsdaten, Service verantwortet Servicedaten, Finance verantwortet Finanzdaten, jeweils als „Data Product", das die anderen Bereiche über vertraglich klare Schnittstellen konsumieren.[21] Dehghani fasst das Konzept in vier Prinzipien: fachliche Verantwortung für Daten in den Domänen, Daten als Pro-

dukt, eine Self-Service-Datenplattform und eine föderierte Governance, deren Regeln automatisiert durchgesetzt werden. Data Mesh ist damit ein Organisations- und Architekturprinzip für die Verantwortung über Daten; ein Reifegradmodell ist es nicht.

Die Idee ist seit 2019 in Umlauf und hat in größeren Unternehmen eine fundierte Anhängerschaft gefunden, im Mittelstand aber häufiger Enttäuschungen als Erfolge produziert. Der Grund liegt in einer stillen Voraussetzung des Konzepts. Data Mesh setzt voraus, dass jeder Bereich über genügend Datenkompetenz verfügt, ein eigenes Data Product mit Versionierung, Schema-Kontrakten, Qualitätsmessung und Pflegeprozess zu führen. In einem Konzern mit dedizierten Daten-Teams je Domäne ist das machbar. In einem mittelständischen Unternehmen, in dem die Datenkompetenz oft in einer einzigen Person sitzt, ist es ein Strukturversprechen, das die Organisation nicht halten kann.

Die pragmatische Lesart für die meisten Unternehmen lautet deshalb: Die Prinzipien des Data Mesh übernehmen, ohne den vollen organisatorischen Umbau zu erzwingen. Konkret heißt das: Datenbestände werden als Produkte gedacht, mit benannten Ownern und dokumentierten Schnittstellen, auch wenn die technische Pflege vorerst zentral bleibt. Erst wenn ein Bereich tatsächlich die Kompetenz und das Volumen für ein eigenes Datenprodukt entwickelt, übernimmt er auch die technische Verantwortung. Dehghani selbst versteht die vier Prinzipien als zusammengehörig. Die schrittweise Einführung, erst Daten als Produkt und fachliche Verantwortung, die technische Dezentralisierung später, ist eine Anpassung an die Lage im Mittelstand und nach unserer Einschätzung die einzige Form, in der das Muster dort belastbar funktioniert.

Feature Stores: das stille Bindeglied

Eine letzte Komponente verdient Aufmerksamkeit, weil sie häufig unterschätzt wird und gleichzeitig viele konkrete Probleme löst: der **Feature Store**. Ein Feature ist im maschinellen Lernen ein abgeleiteter Eingangswert, also etwa „durchschnittlicher Umsatz pro Kunde in den letzten 90 Tagen", den ein Modell als Eingabe nutzt. Wird dieselbe Größe einmal im Training und einmal im produktiven Betrieb berechnet, schleichen sich an beiden Stellen leicht unterschiedliche Werte ein. Das Modell, das im Test brilliert, liefert dann in Produktion seltsam schwache Ergebnisse, und niemand findet sofort heraus, warum. Das Phänomen ist als *training-serving skew* bekannt und gehört zu den teuersten Fehlerquellen produktiver ML-Systeme.[22]

Ein Feature Store ist eine Komponente, die Features zentral definiert, berechnet, speichert und sowohl für Training als auch für Inferenz gleich ausliefert. Damit wird der Skew an seiner Quelle abgeschnitten. Open-Source-Implementierungen wie Feast oder die in den großen Plattformen verbauten Varianten von Databricks, Tecton, Vertex AI und SageMaker bedienen dasselbe Muster, mit Unterschieden in der Tiefe der Integration und der Anzahl der unterstützten Quellen.

Für die Architekturentscheidung im Mittelstand zählt zuerst der Schwellwert, das konkrete Werkzeug kommt danach. Die Antwort hängt an einer einzigen Größe: Wie viele Modelle teilen sich wie viele Features? Bei einem oder zwei isolierten Modellen ist ein Feature Store eine teure Antwort auf ein Problem, das man noch nicht hat. Ab fünf bis zehn produktiven Modellen, die auf überlappenden Feature-Mengen arbeiten, ist die Frage umgekehrt: Ein Unternehmen ohne Feature Store baut sich dann einen, ohne es zu merken, und schlechter als die Spezialwerkzeuge. Für die Datenschicht der Architektur folgt daraus: Den Feature Store explizit als Option in der Vision vermerken, mit einer Schwelle, ab der er eingeführt wird. So wird aus einer Bauchentscheidung später eine vorbereitete.

3.3 KI-Infrastruktur: Cloud, Hybrid, On-Premise

Eine Entscheidung mit drei guten Antworten

Die Frage, wo KI-Workloads laufen sollen, hat keine universell richtige Antwort, und ein Whitepaper, das so tut, sollte man wegwerfen. Was es gibt, sind drei Konstellationen, die jeweils für bestimmte Unternehmen tragen und für andere nicht.

Die erste Konstellation ist die **Hyperscaler-Variante**: AWS, Microsoft Azure oder Google Cloud Platform als primäre Plattform für KI-Workloads, idealerweise in einer europäischen Region mit dokumentierter Datenresidenz. Sie bringt den größten Werkzeugkasten mit, die kürzeste Strecke zur produktiven Anwendung und die niedrigste Einstiegshürde für ein Unternehmen ohne tiefe KI-Infrastruktur-Erfahrung. Sie bringt aber auch eine Abhängigkeit mit, die über das Kommerzielle hinausgeht. Eine Pipeline, die auf SageMaker, Vertex AI oder Azure ML aufgesetzt ist, lässt sich nicht ohne Reibung in ein anderes Ökosystem umziehen. Nach unserer Beobachtung wird diese Abhängigkeit in vielen Unternehmen als Sorge genannt, ohne dass sie die Nutzung der Hyperscaler bislang merklich gebremst hätte.

Die zweite Konstellation ist die **Hybrid-Variante**. Sensible Daten und besonders regulierte Workloads bleiben in einem eigenen oder einem klar europäisch verankerten Rechenzentrum, weniger sensible Workloads laufen in einer Hyperscaler-Region. Das ist die Form, in die viele Banken, Versicherungen und Industrieunternehmen mit komplexer Rechtslage hineinwachsen. Sie ist teurer als die reine Hyperscaler-Variante, weil zwei Welten betrieben und integriert werden müssen, und sie verlangt eine Architekturdisziplin, die Workload-Klassen klar zuordnet. Für Unternehmen mit echter Datensensitivität ist sie nach unserer Einschätzung meist die tragfähigste Antwort.

Die dritte Konstellation ist die **on-premise Variante**. Sie hat (mal wieder) ein Comeback, getrieben von zwei Bewegungen: dem Wunsch nach digitaler Souveränität, der politisch und kommerziell zugleich wirkt, und einer veränderten ökonomischen Rechnung bei großen, dauerhaft hohen Inferenz-Lasten. Für den kontinuierlichen Betrieb von Sprachmodellen in größerer Stückzahl kann dedizierte Hardware bei passenden Modellgrößen günstiger sein als gebuchte GPU-Stunden bei einem Hyperscaler. Diese Rechnung gilt nicht für jeden Anwendungsfall, und sie verlangt Betriebsreife, die ein durchschnittlicher Mittelständler nicht hat. Sie ist aber für eine wachsende Zahl industrieller und öffentlicher Organisationen wieder ernsthaft im Gespräch.

Eine Entscheidungsmatrix, die man sich aufschreiben sollte

Eine kleine Matrix hilft, diese Entscheidung zu disziplinieren. Vier Achsen werden geprüft: **Datensensitivität** (welche Klassifikation tragen die genutzten Daten?), **Lastprofil** (sporadisch, regelmäßig, dauerhaft hoch?), **Reifegrad der eigenen IT** (welche Betriebs- und KI-Kompetenz steht zur Verfügung?) und **strategische Souveränität** (wie wichtig ist Unabhängigkeit vom außereuropäischen Anbieter?). Hyperscaler in einer EU-Region trägt fast jede Konstellation, in der die ersten drei Achsen nicht extrem ausschlagen. Hybrid wird eine ernsthafte Option, sobald Datensensitivität hoch und Lastprofil regelmäßig sind. On-Premise wird zur Option, sobald Lastprofil dauerhaft hoch und Reifegrad eindeutig vorhanden ist. Ohne aufgeschriebene Matrix wird die Entscheidung trotzdem getroffen, nur unbewusst und je nach Tagesform.

GPUs, Inferenz, FinOps

Eine kurze Bemerkung zur Kostenseite, weil sie 2026 anders aussieht als noch vor zwei Jahren. KI-Kosten teilen sich in drei Gruppen. **Trainingskosten** entstehen, wenn ein Modell selbst trainiert oder fine-getuned wird, und sind für die meisten deutschen Unternehmen nicht die Hauptlast, weil sie selten eigene Foundation Mo-

dels trainieren. **Inferenzkosten** entstehen bei jeder Anwendung des Modells, also bei jeder Antwort, jedem Embedding, jeder Klassifikation. Sie sind die dominante laufende Last, sie skalieren mit der Nutzung, und sie sind die häufigste Stelle, an der KI-Vorhaben am Ende ein Budget reißen, das in der Pilotphase nicht thematisiert wurde. **Plattformkosten** sind die Werkzeuge drumherum: Vektordatenbanken, Pipeline-Orchestrierung, Monitoring, Logging.

Für die Architektur folgt daraus eine Pflicht: Jede KI-Anwendung braucht von Anfang an eine Inferenzkosten-Modellierung, die zeigt, wie sich die laufenden Kosten bei steigender Nutzung entwickeln. Eine sinnvolle Modellierung berechnet Kosten pro Anfrage, multipliziert mit erwartetem Anfragevolumen, und hält daneben den jeweiligen Geschäftswert pro Anfrage. Erst dieses Verhältnis zeigt, ob die Anwendung skaliert, sobald sie wirkt. Eine GenAI-Anwendung, die pro Antwort einige Cent verursacht, ist im Pilotbetrieb unauffällig und in der Produktion ein Posten, der jeden Monat sichtbar wird. Werkzeuge des FinOps-Frameworks, ursprünglich für Cloud-Kosten entwickelt, lassen sich auf KI-Lasten übertragen und bilden die operative Grundlage dieser Steuerung.[23]

Edge AI: ein Hinweis, kein Hauptmuster

Edge AI bezeichnet die Verlagerung von KI-Inferenz auf Geräte oder Knoten in der Nähe der Datenquelle, statt sie in ein zentrales Rechenzentrum oder eine Cloud-Region zu schicken. Für produzierende Industrie, Logistik, Energieverteilung und Verkehr ist das ein ernsthaftes Muster, weil Latenz, Verfügbarkeit und Datenmenge sonst die Anwendung unbrauchbar machen. Für die meisten Unternehmen, die mit GenAI in Verwaltung, Vertrieb oder Service starten, ist Edge AI 2026 noch kein Pflichtthema. Im industriellen Kontext gehört Edge AI trotzdem in die Architecture Vision, weil die Entscheidung über die Datenschicht und die Konnektivität spätere Edge-Strategien stark einschränkt, wenn sie ohne diesen Blick getroffen wird.

Wohin Kapitel 3 trägt

Datenstrategie, Datenarchitektur, Infrastruktur. Drei Schichten, die in vielen Diskussionen verschmelzen und dabei voneinander wegrutschen. Die Datenstrategie klärt, was ein Datenbestand sein muss, damit eine KI-Anwendung ihn verwerten kann: gepflegt, beschrieben, verantwortet, rechtlich sauber verarbeitet. Die Datenarchitektur stellt die Muster bereit, in denen diese Bestände gehalten werden: Lake, Warehouse, Lakehouse als drei Bauformen, Data Mesh als Organisationsprinzip für dezentrale Datenverantwortung, Feature Store als technische Antwort auf wiederkehrende Skew-Probleme. Die Infrastrukturentscheidung legt fest, wo die Workloads laufen und unter welchen Kosten.

Nach diesem Kapitel lautet die Frage, welche der vier Architekturmuster, die das nächste Kapitel beschreibt, auf das gerade beschriebene Fundament passen. RAG ohne Datenqualität ist eine Halluzinationsmaschine. Eine Plattform ohne Daten-Governance ist ein Werkzeugkasten ohne Werkstattordnung. Ein Agentensystem ohne Feature-Disziplin produziert Entscheidungen, die niemand mehr nachvollzieht. Kapitel 4 baut auf diesem Fundament auf.

KAPITEL 4

4 Die Säulen: Architektur-Muster 2026

Eine Vorstellung von Architekturmustern läuft Gefahr, zum Bilderbuch zu werden. Hier ein Diagramm, dort ein Akronym, irgendwo eine Box, die „Orchestrator" heißt. Dieses Kapitel hat einen anderen Anspruch. Architektur-Muster sind interessant, weil sie Entscheidungen abnehmen. Sie sind Antworten auf wiederkehrende Probleme, in einer Form, die andere Unternehmen schon bezahlt haben. Mit ihnen muss nicht jeder Anwendungsfall von Grund auf neu gedacht werden.

Die folgenden vier Muster sind 2026 die relevanten. Neu sind sie nicht: RAG existiert seit 2020,[24] Plattform-Patterns sind so alt wie das Wort Cloud, Multi-Cloud-Strategien werden seit zehn Jahren beschrieben. Relevant sind sie, weil sie 2026 in einer Form verfügbar sind, die für deutsche Unternehmen tragfähig ist, und weil sie sich gegenseitig brauchen. Eine RAG-Anwendung ohne Plattform skaliert nicht. Eine Plattform ohne Governance ist eine offene Tür. Ein Agentensystem ohne saubere Datenanbindung ist ein Schauspielhaus mit unbespielten Bühnen.

Was hier folgt, ist eine Auswahl. Federated Learning, Edge AI und Quantum-ML-Hybride sind ausgespart, weil sie in den meisten Unternehmen dieses Jahres noch keine Investitionsentscheidung erzwingen. Ein Unternehmen mit gutem Fundament kann sie später ergänzen, ohne neu anzufangen.

4.1 Generative KI: RAG und das, was danach kommt

Es gibt eine Beobachtung, die in fast jedem Workshop zur Sprache kommt, in dem ein erstes GenAI-Projekt evaluiert wird. Ein Mitarbeiter sagt sinngemäß: „Wir haben das Sprachmodell mit unseren Dokumenten gefüttert, aber es erfindet trotzdem Dinge." Die Antwort fällt dann meist in zwei Varianten. Die eine: „Dann nehmen wir ein besseres Modell." Die andere: „Wir müssen es fine-tunen." Beide Antworten greifen daneben. Die richtige Antwort heißt: Sie haben dem Modell die Dokumente vermutlich nicht gegeben. Sie haben sie ihm zur Verfügung gestellt.

RAG-Architektur: Unternehmenskontext für LLMs

Das ist der Kern von **Retrieval-Augmented Generation**, kurz RAG. Ein Sprachmodell, so groß und gut es auch sei, kennt die Welt nur bis zum Tag, an dem sein Training endete, und es kennt das Unternehmen, das es einsetzt, in den meisten Fällen gar nicht. RAG schließt diese Lücke, indem es zwischen Anfrage und Antwort eine Stufe einzieht: Wenn ein Nutzer eine Frage stellt, wird sie nicht direkt an das Modell weitergereicht. Sie wird zunächst genutzt, um in einem Index relevante Textstellen aus den eigenen Quellen zu finden. Diese Textstellen werden mit der Frage zusammen an das Modell geschickt, mit der Anweisung, die Antwort auf Basis genau dieser Stellen zu formulieren. Das Modell wird damit vom Allwissenden zum Bibliothekar.

Klingt einfach. Ist in der Theorie auch einfach. In der Praxis liegt die Schwierigkeit in einem halben Dutzend Detailfragen, die sich erst bei der Implementierung stellen. Welche Dokumente kommen in den Index, und in welcher Granularität? Wie werden sie zerlegt, eingebettet, gesucht? Reicht eine Vektorsuche, oder braucht es eine Kombination aus semantischer und klassischer Volltextsuche, eine sogenannte hybride Suche? Wie wird sichergestellt, dass die Treffer aktuell sind, wenn sich die Quelldokumente ändern? Wie geht das System mit

Zugriffsrechten um, damit der Mitarbeiter aus dem Vertrieb keine Personalakten gezeigt bekommt? Bleiben diese Fragen unbeantwortet, entsteht eine RAG-Anwendung, die in der Demo brilliert und im Pilotbetrieb peinlich wird.

Eine Referenzarchitektur in sechs Schichten

Eine tragfähige RAG-Architektur besteht aus sechs erkennbaren Schichten. **Quellen und Aufnahme** verbinden die Systeme, in denen das Unternehmenswissen tatsächlich liegt: Dateiablagen, Wikis, Ticketsysteme, ERP-Felder. **Aufbereitung und Indexierung** zerlegen die Inhalte in sinnvolle Einheiten, errechnen Einbettungen und legen sie in einer Vektordatenbank ab. **Retrieval** findet auf eine Anfrage die passenden Stellen, idealerweise hybrid und mit Re-Ranking, also einer zweiten, präziseren Bewertung der ersten Treffer. **Generation** ist das Sprachmodell selbst, das mit den gefundenen Stellen die Antwort formuliert. **Leitplanken** prüfen Eingaben und Ausgaben auf das, was nicht passieren soll: keine Antworten auf Themen außerhalb des Anwendungsfalls, keine Preisgabe interner Inhalte an Unbefugte, keine Halluzinationen, die wie Zitate aussehen, aber keine sind. Und **Telemetrie und Evaluation** beobachten kontinuierlich, was das System tut, wo es daneben liegt und wo seine Antworten von der Wirklichkeit driften.

Die sechste Schicht ist die, die in Pilotprojekten am häufigsten fehlt, und sie ist die teuerste Lücke. Ein GenAI-System, das nicht ständig gegen Referenzantworten gemessen wird, altert schneller, als man hinsehen kann. Ein neues Modell, eine veränderte Quellenstruktur, ein Update am Retrieval, und plötzlich liefert das System andere Antworten als zuvor, ohne dass jemand es merkt. Eine GenAI-Anwendung ohne Eval-Setup in einem produktiven Prozess überlässt einer Maschine die Stimme des Unternehmens, und niemand sieht hin.

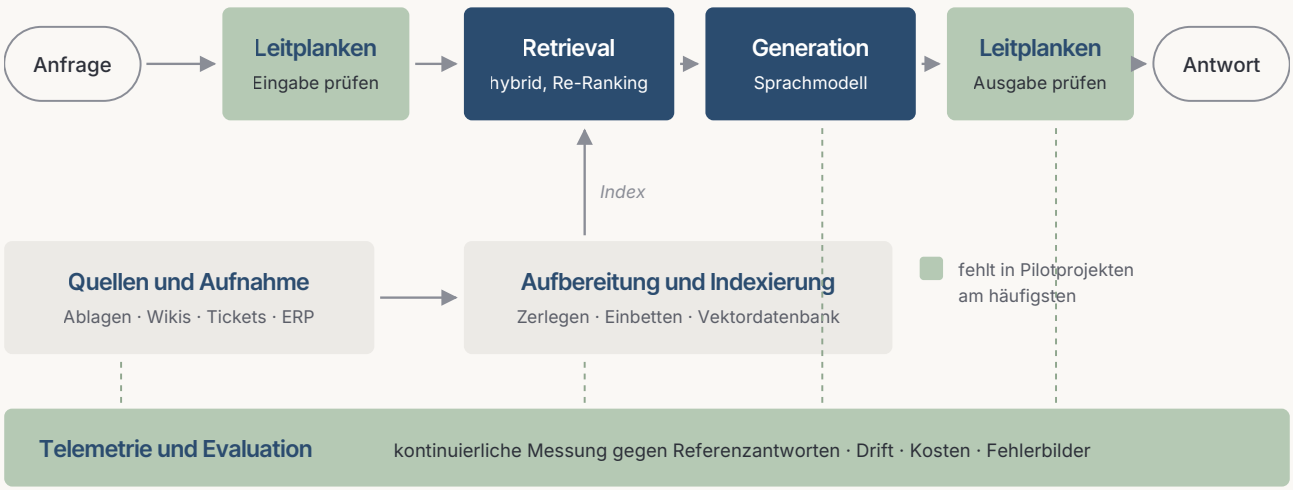


Abbildung 4.1 · Referenzarchitektur einer RAG-Anwendung in sechs Schichten. Leitplanken sowie Telemetrie und Evaluation fehlen in Pilotprojekten am häufigsten.

Fine-Tuning, RAG oder Prompt Engineering: die Entscheidungsmatrix

Damit zur Entscheidung, die in fast jedem Workshop irgendwann ansteht: **RAG, Fine-Tuning oder Prompt Engineering?** Sie wird gern als Glaubensfrage geführt und ist in Wirklichkeit eine Frage der Aufgabe. Prompt Engineering, also die geschickte Formulierung der Anweisung an das Modell, ist die billigste Antwort. Sie funktioniert, wenn die Aufgabe stilistisch oder strukturell ist: einen Text zusammenfassen, ein Format einhalten, einen Ton treffen. Sie versagt, sobald Fachwissen aus dem Unternehmen ins Spiel kommt. RAG ist die Antwort, wann

immer das Modell auf aktuelles, nicht-öffentliches oder unternehmensspezifisches Wissen zugreifen muss. Das ist der Regelfall im Enterprise-Einsatz. Fine-Tuning, also die Anpassung des Modells selbst, ist die teuerste und seltenste der drei Antworten. Sie lohnt sich, wenn ein sehr spezifisches Verhalten erforderlich ist, etwa eine bestimmte Ausgabestruktur, eine Fachsprache, eine domänentypische Argumentationslogik, die sich auch mit vielen Beispielen im Prompt nicht stabil herstellen lässt. Die meisten produktiven Anwendungen sind eine Kombination: ein moderat geprompteter Generator, mit RAG für das Wissen, und sehr selten ein Fine-Tuning für ein klar abgegrenztes Verhaltensmerkmal.

Halluzinationen, Nicht-Determinismus und Architektur-Antworten

Ein Wort zu Halluzinationen. Sprachmodelle erfinden, weil ihre Aufgabe statistischer Natur ist: Sie sagen voraus, was wahrscheinlich auf eine Eingabe folgt. Wo sie nichts wissen, füllen sie die Lücke mit dem, was plausibel klingt. Eine Architekturantwort auf Halluzinationen besteht aus drei Bausteinen. Erstens: Begrenzen, worüber das Modell überhaupt sprechen soll, durch klare Aufgabendefinitionen und durch Leitplanken, die thematische Ausreißer abfangen. Zweitens: Daten dazwischenschalten, die das Modell zwingen, mit echten Belegen zu arbeiten. Das tut RAG. Drittens: Antworten überprüfen, wo es kritisch ist. Bei rechtlichen, medizinischen oder finanziellen Fragen reicht ein Confidence Score nicht. Es braucht eine zweite Instanz, die die Antwort gegen die genannten Quellen prüft, gegebenenfalls automatisiert durch ein zweites Modell, gegebenenfalls durch einen Menschen im Loop. Halluzinationen vollständig auszuschließen, verlangt deterministische Software. Der Umgang mit ihnen verlangt Architektur.

4.2 Agentische KI: Vom Werkzeug zum Akteur

Von reaktiven Tools zu autonomen Agenten

Bis Mitte 2024 war eine KI-Anwendung im Enterprise-Kontext fast immer dasselbe: ein Nutzer stellte eine Frage oder gab eine Eingabe, das System antwortete einmal, der Vorgang war abgeschlossen. Ein Tool, in der ehrlichen Bedeutung des Wortes. Etwas, das man in die Hand nimmt, benutzt und wieder weglegt.

Seit 2025 verschiebt sich dieses Bild, und 2026 ist das Jahr, in dem die Verschiebung in den ersten produktiven Anwendungen ankommt. Wie früh diese Bewegung noch ist, zeigt der Stanford AI Index 2026: Der Einsatz von KI-Agenten lag 2025 in fast allen Unternehmensfunktionen im einstelligen Prozentbereich.[25] Ein **Agent** ist eine KI, die handelt. Sie zerlegt eine Aufgabe in Teilschritte, plant eine Reihenfolge, ruft Werkzeuge auf, prüft Zwischenergebnisse, korrigiert sich und liefert am Ende ein Resultat. Werkzeuge sind dabei nicht metaphorisch zu verstehen: Es sind echte Schnittstellen, die der Agent ansteuert, vom Datenbank-Query über einen Webservice bis zum Versand einer Bestellung. Der Unterschied zur RAG-Anwendung des vorigen Abschnitts liegt in der Architektur um das Modell herum; das Modell selbst ist oft dasselbe. RAG bedient eine Anfrage. Ein Agent bedient ein Ziel.

McKinsey hat diese Verschiebung im Sommer 2025 unter dem Titel *Seizing the agentic AI advantage* beschrieben und sieht in Agenten den Weg aus dem Gen AI Paradox: Sie sollen funktionsspezifische, vertikale Anwendungsfälle aus dem Pilotstatus in die Wirkung bringen.[26] Das ist die gute Nachricht. Die andere: Agentensysteme verändern die Architekturanforderungen grundlegend. Ein Team, das sie wie eine etwas größere RAG-Anwendung behandelt, bemerkt die Probleme erst im Betrieb.

Drei Probleme, die sich erst im Betrieb zeigen

Drei dieser Probleme sind so verbreitet, dass es sich lohnt, sie zu benennen. Das erste ist die **Verkettungsfrage.** Ein Agent, der mehrere Werkzeuge nacheinander aufruft, multipliziert Fehler. Wenn jeder Schritt zu 95 Prozent richtig ist, sind fünf verkettete Schritte nur noch zu 77 Prozent richtig. Eine Architektur, die das nicht abfängt, durch Zwischenprüfungen, durch klare Abbruchkriterien, durch eine Rückkehr zum Menschen an den richtigen Stellen, produziert eine erstaunliche Menge präzise formulierter Fehlleistungen. Das zweite ist die **Berechtigungsfrage.** Ein Werkzeug ist eine Tür. Wenn der Agent es aufrufen darf, kann er hindurch. Welche Türen darf er aufmachen, in welcher Reihenfolge, unter welchen Bedingungen, in wessen Namen? Diese Fragen sind Designentscheidungen, die vor dem ersten produktiven Werkzeugaufruf getroffen sein müssen. Das dritte ist die **Beobachtbarkeitsfrage.** Ein Agent, der zehn Schritte selbständig ausführt, hinterlässt zehn Entscheidungen, die nachvollzogen werden müssen. Ohne Spuren, die von Anfang an mitgeschrieben werden, lässt sich später nicht erklären, warum das System getan hat, was es getan hat. Das ist regulatorisch, betrieblich und kulturell ein Problem.

Der Agentic AI Mesh: Architektur für Multi-Agent-Systeme

Die Architekturantwort auf diese drei Probleme heißt **Agentic AI Mesh**, ein Begriff aus derselben McKinsey-Veröffentlichung.[26] Er klingt nach Marketing und beschreibt eine Idee, die sich in der Praxis bewährt. Das Muster setzt auf ein Geflecht spezialisierter Agenten, von denen jeder eine klare Aufgabe hat, einen klaren Werkzeugkasten und klare Grenzen. Ein großer, allwissender Agent, der alles kann, ist darin nicht vorgesehen. Ein Orchestrierungsagent, der nichts selbst tut, sondern Aufgaben verteilt. Ein Recherche-Agent, der Quellen durchsucht. Ein Buchungs-Agent, der eine Transaktion vorbereitet. Ein Prüfagent, der ein Ergebnis kontrolliert, bevor es freigegeben wird. Über allen liegt eine zentrale Steuerungsebene, die Identitäten, Berechtigungen, Protokolle und Eingriffsrechte verwaltet. Die Ausführung ist dezentral, die Aufsicht ist zentral. Diese Trennung ist strukturell notwendig.

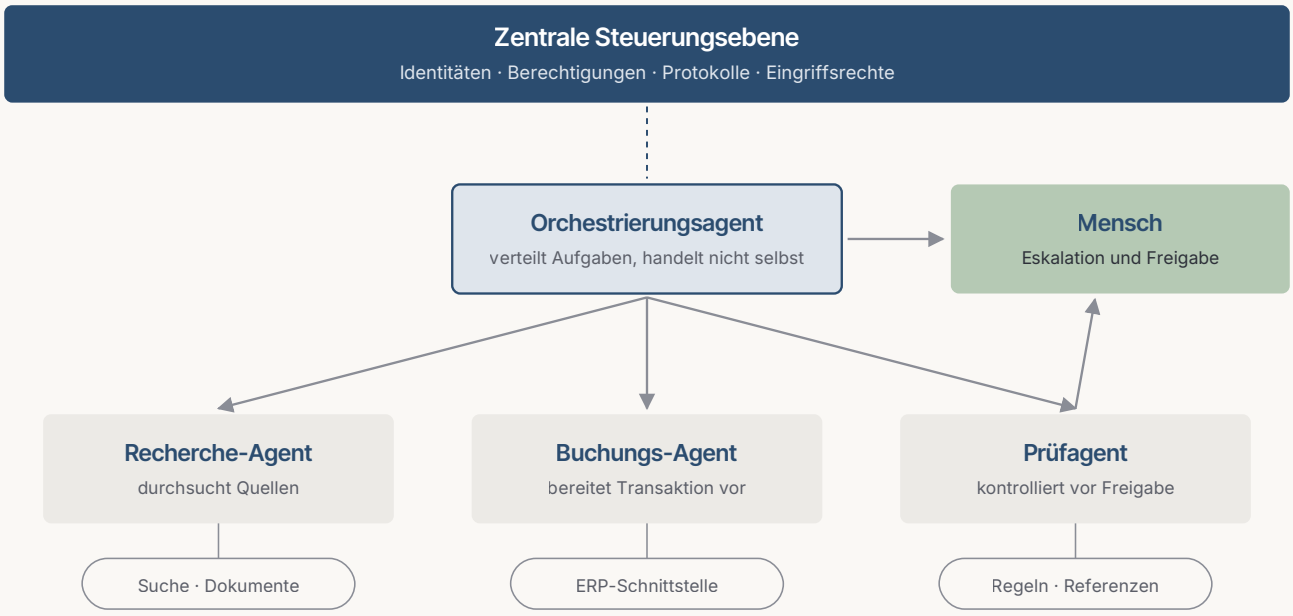


Abbildung 4.2 · Der Agentic AI Mesh. Spezialisierte Agenten mit eigenem Werkzeugkasten, eine zentrale Steuerungsebene und der Mensch als Eskalations- und Freigabestelle.

Orchestrierung oder Choreographie

Damit verbunden ist eine Entwurfsentscheidung, die früher selten gestellt wurde und in agentischen Systemen unausweichlich ist: **Orchestrierung oder Choreographie?** Bei der Orchestrierung gibt es einen Dirigenten, der weiß, wer wann was tut, und die Anweisungen verteilt. Das ist klar nachvollziehbar, leicht zu debuggen, aber starr. Bei der Choreographie folgen die Agenten Regeln und reagieren auf Ereignisse, ohne dass eine zentrale Stelle alles steuert. Das ist flexibel, robust gegen Ausfälle, aber schwerer zu durchschauen. Für die meisten Unternehmensanwendungen im Jahr 2026 ist Orchestrierung der pragmatischere Einstieg. Sie kostet etwas Flexibilität und liefert dafür Auditierbarkeit, die in regulierten Branchen ohnehin verlangt wird. Choreographische Muster lohnen sich dort, wo viele unabhängige Agenten in einem Ökosystem agieren, etwa zwischen Unternehmen, oder in sehr großen internen Plattformen.

Interoperabilität: MCP und A2A

Ein Geflecht spezialisierter Agenten funktioniert nur, wenn seine Teile eine gemeinsame Sprache sprechen. Dafür haben sich zwei offene Protokolle etabliert, und sie gehören in jede Architektur, die mehr als einen Agenten vorsieht.

Das **Model Context Protocol (MCP)** regelt, wie ein Agent Werkzeuge, Datenquellen und Prompts anbindet. Anthropic hat es im Dezember 2025 an die Agentic AI Foundation übergeben, einen Fonds unter dem Dach der Linux Foundation.[27] Die Spezifikationsrevision vom 28. Juli 2026 hat den Protokollkern zustandslos gemacht: Jede Anfrage trägt ihre Protokollversion und die Fähigkeiten des Clients selbst mit, Listenergebnisse sind zwischenspeicherbar, die Autorisierung ist gehärtet, und für abgekündigte Funktionen gilt ein Zeitraum von mindestens zwölf Monaten bis zur frühesten Entfernung.[27] Für die Architektur zählt vor allem der zustandslose Kern. Er macht MCP-Server auf gewöhnlicher HTTP-Infrastruktur betreibbar, mit Lastverteilung und Zwischenspeichern, wie sie für jede andere Schnittstelle üblich sind.

Das **Agent2Agent Protocol (A2A)** regelt die Zusammenarbeit zwischen Agenten, auch über Anbieter- und Framework-Grenzen hinweg. Es liegt in Version 1.0 vor und wird unter dem Dach der Linux Foundation von einem Gremium gesteuert, in dem unter anderem AWS, Cisco, Google, IBM Research, Microsoft, Salesforce, SAP und ServiceNow vertreten sind.[28] Die Arbeitsteilung beschreibt das Projekt selbst knapp: MCP verbindet einen Agenten mit seinen Werkzeugen, A2A verbindet Agenten miteinander.

Für die Architektur folgt daraus eine Festlegung, die früh getroffen werden sollte. Werkzeuganbindungen laufen über MCP, die Kommunikation zwischen Agenten über A2A, und die zentrale Steuerungsebene des Agentic AI Mesh setzt an genau diesen beiden Schnittstellen an. Dort lassen sich Identitäten prüfen, Berechtigungen durchsetzen und Spuren schreiben, gleich welches Modell und welches Framework hinter einem einzelnen Agenten steht. Das ist die Anbieterneutralität aus Kapitel 2, übertragen auf agentische Systeme.

Governance-Anforderungen für autonome KI-Agenten

Eine letzte Bemerkung zur Governance autonomer Systeme, die noch reichlich zu Diskussionen führen wird. Ein agentisches System handelt in einem Rahmen, den Menschen gesetzt haben, auch wenn „autonom" an seiner Tür steht. Der EU AI Act ordnet es deshalb wie jedes andere KI-System nach seinem Einsatzzweck ein: Fällt dieser unter Anhang III, etwa in Personalauswahl oder Kreditvergabe, gilt es als Hochrisiko-System, gleich wie autonom es arbeitet.[29] Das heißt, jeder Agent braucht einen Verantwortlichen, jede Aktion braucht eine Spur, jede Eskalation braucht eine definierte Rückgabestelle an den Menschen. Als Prüfliste für die Sicherheitsseite dient die OWASP Top 10 für agentische Anwendungen vom Dezember 2025. Sie reicht von der Übernahme der Agentenziele (ASI01) über den Missbrauch von Werkzeugen (ASI02) und von Identitäten und Rech-

ten (ASI03) bis zu sich ausbreitenden Folgefehlern (ASI08) und abtrünnigen Agenten (ASI10).[30] Ein System, das all das von Anfang an einbaut, erspart sich später den Compliance-Kraftakt. Ein System ohne diese Vorkehrungen muss seinen Mehrwert im Audit erst einmal verteidigen.

4.3 Die Enterprise AI Platform

Das AI Factory Pattern: eine Plattform für 20+ Anwendungsfälle

Die ersten beiden Abschnitte dieses Kapitels beschreiben Anwendungstypen. Eine RAG-Anwendung ist ein Anwendungsfall. Ein Agent ist ein Anwendungsfall. Das nächste Muster ist von anderer Art: Es ist die Grundlage, auf der zehn, zwanzig oder fünfzig solcher Anwendungen entstehen können, ohne dass jede für sich eine eigene Welt aufbaut.

Eine **Enterprise AI Platform**, manchmal auch als *AI Factory* bezeichnet, ist eine interne Entwicklungsplattform, wie sie das klassische Software-Engineering kennt, übersetzt in die Bedürfnisse von KI-Anwendungen. Sie stellt Bausteine bereit, die jeder Anwendungsfall braucht, damit ihn nicht jeder Anwendungsfall neu bauen muss. Das klingt unspektakulär. Es ist der Hebel, der über Stückkosten entscheidet. Ein Unternehmen, das ohne Plattform arbeitet, baut beim zehnten Use Case zum zehnten Mal das Modellmanagement, das Logging, die Zugriffskontrolle und das Eval-Setup. Ein Unternehmen mit Plattform baut diese Dinge einmal und nutzt sie überall.

Das Layer-Modell: Daten, Modelle, Betrieb, Beobachtung, Governance

Welche Schichten gehören in eine solche Plattform? Im Kern fünf, und sie sind in fast jeder seriösen Referenzarchitektur dieselben, auch wenn die Namen variieren. Eine **Datenschicht** stellt Zugriffe, Aufbereitungen und Feature-Definitionen bereit. Sie ist die Brücke zum Datenfundament, von dem in Kapitel 3 ausführlicher die Rede ist. Eine **Entwicklungs- und Modellschicht** bietet Werkzeuge zum Experimentieren, Trainieren, Versionieren und Registrieren von Modellen, einschließlich der Prompts, die in der GenAI-Welt zu eigenständigen Artefakten geworden sind. Eine **Betriebs- und Bereitstellungsschicht** kümmert sich darum, dass Modelle in Produktion gehen und dort bleiben: Inferenz, Skalierung, Routing, A/B-Tests, Rollbacks. Eine **Beobachtungs- und Evaluierungsschicht** misst kontinuierlich, was das System tut, vergleicht es mit Referenzen, schlägt Alarm, wenn etwas driftet. Und eine **Governance- und Zugangsschicht**, die quer durch alle anderen liegt, regelt, wer was darf, dokumentiert Entscheidungen, hält Audit-Spuren und macht Richtlinien im Code verbindlich.

Make vs. Buy: Databricks, Azure ML, Vertex AI, SageMaker im Vergleich

Diese Schichten ergeben zusammen die Antwort auf eine Frage, die in jedem mittelgroßen Unternehmen früher oder später gestellt wird: **Selbst bauen oder einkaufen?** Die populären Plattformangebote, Databricks, Azure Machine Learning und Microsoft Foundry, Vertex AI auf Google Cloud und SageMaker auf AWS, sowie eine wachsende Zahl spezialisierter Anbieter decken die Schichten in unterschiedlichem Reifegrad ab. Keiner deckt alle gleich gut. Databricks hat seine Wurzeln in Daten und Modellentwicklung. Die Microsoft-Dienste, Azure Machine Learning für klassische Modelle und Microsoft Foundry für generative Anwendungen, sind nahtlos in die Microsoft-Welt integriert, was Vorteil und Risiko zugleich ist. Vertex AI bringt Googles GenAI-Stack mit. SageMaker hat einen großen Werkzeugkasten und eine entsprechende Lernkurve.

Eine pragmatische Antwort: Nutzen Sie Komponenten der genannten Plattformen und bauen Sie die **Verbindungslogik** selbst. Die Abstraktionsschicht, die Ihre Anwendungen von der konkreten Plattform unabhängig hält, ist die Investition, die den Unterschied macht. Sie wirkt am Anfang wie unnötige Mehrarbeit. Sie wird Gold

wert, wenn später ein Plattformanbieter seine Preise verdoppelt, eine Region schließt oder ein neues, besseres Werkzeug auf dem Markt auftaucht. Das ist der Unterschied zwischen einem Mieter und einem Eigentümer. Der Mieter trägt jede Preiserhöhung mit oder muss einen teuren Umzug in Kauf nehmen. Der Eigentümer entscheidet, ob er investiert oder ob das, was er sich gebaut hat, weiterhin reicht.

Self-Service AI: wie Fachbereiche und Data Scientists autonom arbeiten

Ein häufig unterschätzter Aspekt einer reifen KI-Plattform ist der **Self-Service** für Fachbereiche und Data Scientists. In den meisten Unternehmen entstehen die ersten Anwendungsfälle in der zentralen IT oder in einem KI-Kompetenzzentrum. Das ist sinnvoll, solange die Anzahl der Anwendungen klein ist. Sobald sie wächst, wird die zentrale Stelle zum Flaschenhals. Eine gute Plattform öffnet sich dann nach außen: Fachbereiche können eigene Anwendungen auf vorgegebenen Mustern bauen, Data Scientists können eigene Modelle deployen, ohne Tickets bei der IT zu schreiben. Aber, und das ist der Punkt, der dieses Muster oft scheitern lässt, sie tun das in einem Rahmen, der vorgegeben ist. Welche Daten zugänglich sind, welche Modelle erlaubt sind, welche Schnittstellen offenstehen, welche Protokolle anfallen, all das ist in der Plattform geregelt. Self-Service ohne diesen Rahmen ist Schatten-IT mit einem schöneren Namen. Self-Service in einem klaren Rahmen bedeutet Skalierung.

4.4 Multi-Cloud, Hybrid und die Frage der Souveränität

Vendor Lock-in als Architekturrisiko

Es gibt eine Frage, die in jedem Architektur-Workshop früher oder später aufgeworfen wird, oft mit gesenkter Stimme, als sei sie unanständig: „Können wir das nicht auch ohne die Amerikaner?" Die Antwort ist nuancierter als die beiden Reflexe, die sie üblicherweise auslöst: der vollständige Rückzug aus US-Clouds oder das Weitermachen wie bisher.

Die Frage ist berechtigt, und sie ist 2026 lauter, als sie es noch vor drei Jahren war. Sie hat drei Treiber. Erstens die regulatorische Lage: DSGVO, EU AI Act, branchenspezifische Vorgaben im Finanzwesen, im Gesundheitsbereich und in der öffentlichen Verwaltung machen für bestimmte Datenarten Verarbeitungen außerhalb Europas zumindest erklärungsbedürftig, in manchen Fällen unzulässig. Zweitens die geopolitische Lage, die hier nicht aufgefächert werden muss, deren Wirkung auf Cloud-Strategien sich aber inzwischen in jedem Vorstand niederschlägt. Drittens die wirtschaftliche Lage: Die Preise der großen Anbieter steigen, und die Verhandlungsmacht eines Kunden, der seine ganze KI-Last auf einem Anbieter konzentriert hat, ist begrenzt.

Multi-Cloud und Hybrid: zwei Antworten auf drei Treiber

Multi-Cloud und Hybrid sind die beiden Architekturmuster, die auf diese Treiber antworten. **Multi-Cloud** bedeutet, dass Anwendungen über mehrere Cloud-Anbieter verteilt sind. **Hybrid** bedeutet, dass eine Anwendung Cloud und eigene Rechenzentren kombiniert. Beide Muster erhöhen die Komplexität, sie erhöhen die Betriebskosten, sie erfordern Fähigkeiten, die nicht jedes mittelständische Unternehmen hat. Sie sind die richtige Antwort, wenn die genannten Treiber sie tatsächlich verlangen, und die falsche, wenn sie aus diffusen Bauchgefühlen heraus eingesetzt werden.

Ein nüchterner Blick auf die Optionen. Für viele KI-Workloads ist eine **bewusste Single-Cloud-Strategie mit europäischer Region und sauberen Exit-Vorbereitungen** die ökonomisch sinnvollste Lösung. Sie kostet weniger, sie verlangt weniger Personal, sie liefert dieselben Funktionen. Der Schlüssel liegt in den Exit-Vorbereitun-

gen: portable Datenformate, abstrahierte Schnittstellen, Verträge, die einen Wechsel in einem planbaren Rahmen möglich machen, regelmäßig geprobte Migrationspfade. Das ergibt die Fähigkeit, Multi-Cloud zu werden, sobald die Lage es verlangt.

Für regulierte Branchen ist **Hybrid** das gängigere Muster. Die sensitiven Daten und Prozesse bleiben im eigenen Rechenzentrum oder bei einem souveränen Anbieter, die rechenintensiven Trainings- und Inferenzlasten laufen in der Public Cloud, beide Welten sind über klar definierte Schnittstellen verbunden. Das funktioniert, wenn die Schnittstellen sauber sind und die Datenflüsse dokumentiert. Es scheitert, wenn die Trennung weich wird und plötzlich nicht mehr nachvollziehbar ist, welche Daten wo verarbeitet werden.

Echtes Multi-Cloud, also die parallele Nutzung mehrerer Anbieter für dieselben Arten von Workloads, ist die teuerste und seltenste Variante. Sie lohnt sich für Unternehmen mit hoher Verhandlungsmacht, sehr großen Lasten und der Personalstärke, sie zu betreiben. Für den deutschen Mittelstand ist sie in den allermeisten Fällen die falsche Antwort auf die richtige Frage.

Abstraktionsschichten und portierbare ML-Pipelines

Was alle drei Varianten verbindet, ist die Frage, an welcher Stelle man sich an einen Anbieter bindet und an welcher man sich Spielraum lässt. Die Antwort entscheidet weniger das Marketing der Hyperscaler als die eigene Architektur. Eine portierbare KI-Landschaft zeichnet sich an vier konkreten Stellen ab.

Erstens an den **Datenformaten**. Trainings- und Inferenzdaten in offenen, herstellerneutralen Formaten (Parquet, Iceberg, Delta in der offenen Variante, ONNX für Modelle) lassen sich morgen umziehen. Daten in proprietären Tabellenformaten oder geschlossenen Modell-Containern lassen sich das nicht. Zweitens an den **Schnittstellen**. Anwendungen, die gegen offene oder verbreitete De-facto-Standards entwickelt sind, etwa gegen OpenAI-kompatible APIs für LLMs, gegen MLflow für Modelle, gegen Kubernetes für Deployment und gegen MCP für die Werkzeuganbindung von Agenten, haben eine Trennschicht zwischen Anwendung und Plattform. Eine Anwendung, die direkt gegen das proprietäre SDK eines Anbieters gebaut ist, ist mit ihm verschweißt. Drittens an den **Pipelines**. Trainings- und Inferenzpipelines, die in herstellerneutralen Orchestrierungswerkzeugen beschrieben sind, lassen sich auf jedem Anbieter ausführen, der die Werkzeuge unterstützt. Pipelines, die in einer proprietären Pipeline-Sprache geschrieben sind, müssen für jeden Wechsel neu gebaut werden. Viertens an den **Geheimnissen und Identitäten**: Zugangsdaten und Berechtigungen in einer eigenen, anbieterunabhängigen Schicht erlauben einen Cloud-Wechsel, ohne die Identitätenlandschaft zu zerschießen.

Diese vier Punkte klingen technisch und sind in der Sache strategisch. Sie sind der Unterschied zwischen einer Architektur, die einen Anbieterwechsel überlebt, und einer, die ihn nicht überlebt.

Digitale Souveränität: Gaia-X und europäische Cloud-Alternativen

Ein Wort zu **Gaia-X** und den europäischen Souveränitäts-Initiativen, die in den letzten Jahren mit hohen Erwartungen gestartet und mit gemischten Ergebnissen vorangekommen sind. Das Versprechen, eine europäische Cloud-Alternative zu schaffen, ist nicht eingelöst worden, jedenfalls nicht in dem Maße, dass ein Unternehmen heute seine KI-Strategie darauf aufbauen könnte. Es gibt nutzbare Bausteine, es gibt seriöse Anbieter, es gibt einzelne Branchen, in denen sich Souveräne-Cloud-Lösungen etablieren. Die Behauptung, 2026 stehe eine voll funktionsfähige, mit den Hyperscalern auf Augenhöhe stehende europäische Alternative zur Verfügung, ist übertrieben. Was verfügbar ist, sind **vertrauenswürdige Datenräume**, in denen Daten zwischen Unternehmen geteilt werden können, ohne dass eine zentrale Plattform sie aggregiert. Das ist für bestimmte Anwendungsfälle wertvoll. Es ersetzt keine Cloud-Strategie.

Die ehrliche Empfehlung an dieser Stelle lautet weniger spektakulär, als das Thema es nahelegt: Wählen Sie eine Cloud bewusst, bevorzugen Sie europäische Regionen, achten Sie auf vertragliche und technische Exit-Optionen, vermeiden Sie tiefe Verflechtungen mit proprietären Diensten, dort wo Alternativen bestehen, und prüfen Sie für sensitive Bereiche eine hybride Konstellation. Beobachten Sie die europäischen Initiativen, ohne auf sie zu warten. Und führen Sie eine Liste der Funktionen, deren Wegfall Ihr Geschäft ernsthaft beeinträchtigen würde. Diese Liste ist die ehrlichste Form, über Vendor Lock-in zu sprechen.

Was diese vier Muster gemeinsam haben

Auf den ersten Blick wirken die vier Muster wie unverbundene Stichworte: Eine Methode für Wissenszugriff, eine für autonomes Handeln, eine Plattform und eine Cloud-Frage. Auf den zweiten Blick teilen sie drei Eigenschaften, die nicht zufällig sind.

Erstens: Sie sind alle **modular gedacht.** Keines der vier verlangt eine vollständige Umstellung, alle vier lassen sich schrittweise einführen. Das ist die Voraussetzung, um in einem laufenden Geschäft zu arbeiten, ohne es anzuhalten. Zweitens: Sie setzen alle **Governance als integralen Bestandteil voraus**. Ohne Governance-Schicht sind alle vier Muster falsch gebaut. Drittens: Sie geben **Skalierung Vorrang vor Eleganz.** Die erste Anwendung soll laufen. Die zwanzigste muss laufen. Eine Architektur, die nur an die erste denkt, wird zwanzig Mal gebaut.

Diese drei Eigenschaften klingen wie das Echo der drei Architektur-Qualitäten aus Kapitel 1: modular, anbieterneutral, beobachtbar. Das ist kein Zufall. Die Muster sind die Umsetzung der Prinzipien, und die Prinzipien sind die Verallgemeinerung der Muster. Mit den einen sind die anderen mitverstanden.

Im nächsten Kapitel wenden wir uns dem Tag zu, an dem die Architektur steht und das System läuft. Dem Betrieb. MLOps und LLMOps entscheiden, ob aus einem Pilotprojekt eine produktive Anwendung wird.

KAPITEL 5

5 Der Betrieb: MLOps und LLMOps im Produktionsalltag

Der Bitkom zählt im September 2026 erstmals eine Mehrheit der deutschen Unternehmen, die KI einsetzen: 57 Prozent, nach 36 Prozent im Vorjahr und 20 Prozent zwei Jahre zuvor.[31] Hinter dieser Zahl steht eine zweite, weniger gefeierte. Eine in *Scientific Reports* veröffentlichte Studie hat 32 Datensätze aus vier Branchen mit vier verbreiteten Modelltypen untersucht und in 91 Prozent der 128 Kombinationen festgestellt, dass die Modellqualität mit wachsendem Abstand zum letzten Training sinkt.[32] Was in der Demo funktioniert, funktioniert nach längerer Laufzeit in einem produktiven Prozess nicht zwangsläufig noch.

Die Disziplin, die diese Lücke schließt, heißt **MLOps**, *Machine Learning Operations*. Der Begriff ist eine bewusste Anlehnung an *DevOps*, übertragen auf die Eigenheiten von Modellen. Der Vergleich ist nützlich und hat seine Grenzen. Nützlich, weil sich Werkzeuge, Disziplinen und Kultur überlappen: Versionierung, automatisierte Tests, Continuous Delivery, Beobachtbarkeit. Begrenzt, weil ein Modell kein Programm im klassischen Sinn ist. Ein Programm tut, was im Code steht. Ein Modell tut, was die Daten ihm beigebracht haben, und tut es probabilistisch. MLOps als DevOps zu behandeln baut die Werkzeuge richtig auf und verfehlt die eigentliche Aufgabe.

Mit den großen Sprachmodellen ist eine zweite Disziplin hinzugekommen, die in der Praxis 2026 weniger etabliert, dafür dringender ist: **LLMOps**. Sie behandelt die Besonderheiten generativer Modelle in Produktion, von der Versionierung von Prompts bis zur kontinuierlichen Bewertung nicht-deterministischer Ausgaben. Dieses Kapitel behandelt beide Disziplinen, weil sie in den meisten Unternehmen parallel aufgebaut werden müssen, oft mit denselben Teams und überlappenden Werkzeugen.

Zur Einordnung innerhalb dieses Whitepapers: Kapitel 4 hat an mehreren Stellen von Observability, Eval-Setups und Modellregistern gesprochen. Diese Themen tauchen hier wieder auf, weil sie zur Architektur gehören, in ihrer Umsetzung aber zum Betrieb. Was Kapitel 4 als Anforderung formuliert, beschreibt dieses Kapitel als Praxis. Das heißt konkret: Idealerweise arbeiten Architekten und Betrieb an denselben Artefakten, mit unterschiedlicher Perspektive.

5.1 MLOps: Von Ad-hoc zu industrialisierter KI

Was MLOps ist und was es nicht ist

MLOps beschreibt die Praxis, mit der Modelle reproduzierbar trainiert, kontrolliert in Produktion gebracht, dort beobachtet und bei Bedarf zurückgezogen werden. Die Disziplin hat sich seit 2018 herausgebildet und ist 2025 in einer empirischen Studie der Malmö University und der Chalmers University of Technology systematisch beschrieben worden. Die Autoren haben in einer Multi-Case-Untersuchung über vierzehn Unternehmen ein Framework und ein Reifegradmodell für die Einführung von MLOps entwickelt und validiert. Sie fanden viele Gemeinsamkeiten in der Datenpraxis und große Unterschiede bei fortgeschrittenen Praktiken wie Feature Stores und automatisierten Deployments.[33]

Daraus lässt sich ableiten, was MLOps ausmacht: eine Praxis. Es gibt eine wachsende Auswahl nützlicher Werkzeuge, MLflow, Kubeflow, Weights & Biases, Vertex AI Pipelines, SageMaker Pipelines und ein gutes Dutzend mehr. Keines davon verschafft einem Unternehmen diese Praxis, und eine Plattform-Initiative, die unabhängig von Anwendungsfällen vorangetrieben wird, ebenso wenig. Die Frage „Wie reif ist unser MLOps?" ist eine schlechtere als die Frage „Welcher unserer produktiv genutzten Use Cases würde an einer fehlenden MLOps-Praxis scheitern, wenn wir nichts tun?". Die erste Frage führt zu Plattformen, die niemand auslastet. Sie deckt sich gut mit Roadmaps und schlecht mit Wirkung. Die zweite verlangt, einen konkreten Use Case zu Ende zu denken, und führt zu Investitionen, die unmittelbar tragen.

MLOps braucht in den ersten Jahren auch keine eigenständige Organisationseinheit. Die Disziplin liegt zwischen Data Science und IT-Betrieb. Ein natürlicher Eigentümer findet sich dort selten von allein. Was sich beobachten lässt, wenn Unternehmen den Schritt überspringen: Eine Disziplin, die als reines Plattform-Team aufgezogen wird, ohne dass Data Scientists und Betrieb in denselben Artefakten arbeiten, erzeugt eine dritte Silostruktur, wo eine Brücke gebraucht würde.

Ein Reifegradmodell zur ersten Verortung

Zur ersten Verortung haben sich Stufenmodelle etabliert. Das **MLOps-Reifegradmodell von Microsoft** beschreibt fünf Stufen vom Ad-hoc-Betrieb bis zur vollautomatisierten Plattform.[34] Auf Stufe null entsteht das Modell im Notebook und wird als Datei in die Anwendung gereicht. Stufe eins bringt klassische DevOps-Praxis ins Spiel: Versionskontrolle, automatisierte Builds, das Training bleibt manuell. Auf Stufe zwei läuft das Training automatisiert, Stufe drei automatisiert zusätzlich die Bereitstellung, Stufe vier schließt den Kreis aus Beobachtung und automatischem Re-Training. Google beschreibt dieselbe Entwicklung in drei Stufen.[35] Stufe vier wird nach unserer Beobachtung häufiger behauptet als praktiziert.

Das Modell ist nützlich zur ersten Verortung. Es hat zwei Schwächen, die man kennen sollte, bevor man sich auf eine Zahl festlegt. Es suggeriert Linearität, wo in der Praxis Sprünge stattfinden. Und es verleitet Unternehmen dazu, sich eine Stufe höher einzuordnen, als die tägliche Arbeit hergibt. Microsoft selbst weist darauf hin, dass eine Organisation oft Merkmale mehrerer Stufen zugleich zeigt.[34] Reife ist mehrdimensional, und ein Unternehmen scheitert selten am Durchschnitt, meistens am schwächsten Punkt. Damit wird aus der Frage „Welche Stufe haben wir erreicht?" die nützlichere Frage „In welcher Dimension stehen wir wo?".

Fünf Dimensionen erfolgreicher MLOps-Adoption

Die bereits zitierte Studie von John, Holmström Olsson und Bosch arbeitet diese Mehrdimensionalität empirisch heraus. Sie beschreibt fünf Dimensionen, entlang derer Adoption gelingt oder scheitert, **Daten, Modell, Auslieferung, Betrieb und Infrastruktur, Organisation**, und ordnet jeder Dimension fünf Reifestufen zu, von *Ad hoc* über *DataOps*, *Manual MLOps* und *Automated MLOps* bis *Kaizen MLOps*.[33] Wir greifen diese Systematik auf, weil sie den Reifegrad lesbar macht, ohne ihn zu ersetzen. Ein Unternehmen kann in einer Dimension auf Stufe drei stehen und in einer anderen auf Stufe null. Erst die fünf Dimensionen zeigen, wo investiert werden muss.

Die **Datendimension** umfasst alles, was vor dem Modell entscheidet, ob es überhaupt eine Chance hat. Standardisierte Prozesse für Datenverwaltung, Versionierung, Qualitätsprüfung, Umgang mit sensiblen Daten, Governance. In der Fallstudie zeigten die untersuchten Unternehmen eine starke Datenpraxis; Feature Stores nutzten dagegen nur drei von sieben.[33] Datenarbeit als bloße Vorstufe zu behandeln, engt den Spielraum in den anderen vier Dimensionen empfindlich ein.

Die **Modelldimension** beschreibt die Praxis um das Training herum. Standardisierter Entwicklungsprozess, Versionierung von Code, reproduzierbare Experimente, Verwaltung der Modell-Metadaten. Reproduzierbare Experimente sind in der Untersuchung die größte Lücke: Nur zwei der sieben Unternehmen hatten dafür Vorkehrungen getroffen.[33] Reproduzierbarkeit ist das, was ein Modell vor dem Vergessen bewahrt: Sobald die ursprünglichen Entwickler das Team verlassen, ist ein nicht reproduzierbares Modell ein Stück Code ohne Geschichte. Modell-Metadaten wiederum werden in dem Moment wichtig, in dem eine regulatorische Rückfrage kommt. Ohne sie wird die Antwort zur Spurensuche.

Die **Auslieferungsdimension** trennt produktive Unternehmen von experimentierfreudigen. Automatisierte Deployments, eine CI/CD-Strecke für Modelle, die Fähigkeit, mehrere Versionen parallel auszuspielen. In der Fallstudie hatte zum Erhebungszeitpunkt nur eines der sieben Unternehmen automatisierte Deployments für Modelle eingeführt, die anderen lieferten ad hoc oder manuell aus.[33] Die Zahl ist instruktiv, auch wenn sie sich seitdem verschoben haben mag. Auslieferung ist die Dimension, in der der Übergang vom Prototyp zum Produkt sichtbar wird.

Die **Dimension Betrieb und Infrastruktur** umfasst, was nach dem Go-live geschieht. Automatisiertes Monitoring und Re-Training, Performance-Alerting, eine ausreichend stabile Infrastruktur zum Hosten der Modelle. Hier setzt der Befund zur Modellalterung an, mit dem dieses Kapitel begann. Unternehmen ohne Alerting bemerken eine Verschlechterung erst, wenn ein Fachbereich sich beschwert oder eine Auswertung den Schaden offenlegt.

Die **Organisationsdimension** ist die unauffälligste und die am häufigsten unterschätzte. Definierte Rollen für die Verantwortung produktiver Modelle, regelmäßige Kommunikation zwischen Data Science, IT-Betrieb und Fachbereich. Die Autoren weisen die Aufnahme dieser Dimension ausdrücklich als Validierungsergebnis aus; die befragten Praktiker sahen darin eine Ergänzung gegenüber verbreiteten MLOps-Modellen. Definierte Rollen und regelmäßige Kommunikation fanden sich bei allen Unternehmen mit reifer MLOps-Praxis.[33] Ein Modell, für das niemand zuständig ist, verfällt geräuschlos. Architekturarbeit allein löst das Problem der Eigentümerschaft nicht. Sie schafft den Rahmen, in dem es überhaupt gestellt werden kann.

Für die Praxis ergibt sich aus den fünf Dimensionen eine einfache Lese-Anweisung. Eine Investition, die die schwächste Dimension stärkt, bringt fast immer mehr Wirkung als eine, die die stärkste verfeinert. Ein typisches Muster in der Praxis: Es geschieht das Gegenteil. Die stärkste Dimension hat einen Eigentümer, der weiter investieren will. Die schwächste hat keinen, der das Defizit benennt.

CI/CD für ML: Besonderheiten und Best Practices

Eine CI/CD-Strecke für klassische Software folgt einer bekannten Choreografie: ein Commit, ein Build, ein Test, eine Auslieferung. Für ML gilt dasselbe Grundmuster, mit einer Erweiterung, die so wichtig ist, dass sie in der MLOps-Literatur einen eigenen Buchstaben bekommen hat: **CT, Continuous Training**.[35] Code, Daten und Modell sind drei bewegliche Teile, und eine reife Pipeline beherrscht alle drei. Ausgeliefert wird deshalb die Pipeline, die das Modell trainiert.[36] Solange nur das fertige Modell wandert, hängt jede Aktualisierung am Notebook, in dem es entstanden ist. Sobald die trainierende Pipeline selbst das Artefakt ist, kann sie auf neue Daten neu angesetzt werden, ohne dass jemand ihre Bauanleitung rekonstruiert. Was sie auslöst, ist ebenfalls breiter angelegt als in der klassischen Welt: ein Code-Commit, ein veränderter Trainingsdatensatz, ein Zeitplan oder ein Drift-Signal aus der Produktion. Eine Strecke, die nur auf Commits hört, lässt das Modell altern, bis jemand merkt, dass es altert.

Die zweite Verschiebung betrifft das, **was in der Pipeline getestet und versioniert wird**. In klassischer Software prüft ein Test, ob eine Funktion bei definierter Eingabe eine definierte Ausgabe liefert. In einer ML-Pipeline kommen zwei Klassen hinzu, die es in der klassischen Welt nicht gibt: Datentests prüfen Schema, Wertebereiche, Verteilungen und Skew zwischen Trainings- und Inferenzdaten, bevor überhaupt trainiert wird, Modelltests prüfen Metriken auf einem Referenz-Datensatz und vergleichen das neue Modell mit dem produktiven Vorgänger.[37] Heikel sind die Schwellenwerte. Werden sie nach dem Training festgelegt, orientieren sie sich am aktuellen Modell statt am geforderten Qualitätsniveau. Vor dem Training festgelegt, übernehmen sie die Funktion, die in der klassischen Entwicklung Akzeptanzkriterien haben. Das versionierte Artefakt selbst ist breiter: Code, Konfiguration, Modellgewichte, Referenzen auf Trainings- und Validierungsdaten, Hyperparameter, Evaluierungs-Ergebnisse, hinterlegt in einem **Model Registry**, das jede Version mit Metadaten, Freigabe-Status und Roll-back-Pfad führt.[38] Eine Versionierung, die nur den Code erfasst, lässt später nicht mehr nachvollziehen, welche Version warum welche Antworten gegeben hat. Annex IV des EU AI Act verlangt für Hochrisikosysteme genau diese Nachvollziehbarkeit, vom Datenherkunfts-Nachweis bis zum Post-Market-Monitoring-Plan.[39] Wir schätzen den Aufwand für eine nachträglich erstellte Dokumentation auf das Zwei- bis Dreifache dessen, was anfällt, wenn sie im Pipeline-Lauf von Anfang an mitentsteht.

Die dritte Verschiebung ist die **Auslieferung selbst**. Bei klassischer Software haben sich Muster für riskante Veränderungen etabliert, Blue-Green-Deployments, Canary-Releases, Feature-Flags. Auf ML übertragen werden sie zur Pflicht, und ein drittes Muster kommt hinzu, das in der ML-Welt häufiger gebraucht wird: das **Shadow Deployment**.[40] Das neue Modell läuft parallel zum alten und bekommt eine Kopie des echten Verkehrs. Seine Antworten erreichen den Anwender nicht; sie werden protokolliert und mit den Antworten des produktiven Modells verglichen. Erst wenn die Vergleichsmetriken über einen definierten Zeitraum tragen, übernimmt das neue Modell den Verkehr, zuerst anteilig, dann vollständig. Das verlangt eine Bereitstellungs-Schicht, die mehrere Versionen gleichzeitig kennt. Ohne sie wird jede Aktualisierung zur Vollumstellung im produktiven Betrieb. Zusammengenommen lassen sich diese Besonderheiten auf einen Satz verdichten: Eine CI/CD-Strecke für ML versioniert Code, Daten und Modell, sie testet Funktionen, Daten und Modelle, und sie lässt zwei Versionen nebeneinander laufen, bis die Zahlen sprechen.

Was der Aufschub im EU AI Act bedeutet

An dieser Stelle eine Bemerkung zur Regulierung, weil sie unmittelbar in die MLOps-Praxis hineinwirkt. Mit dem **Digital Omnibus on AI**, seit dem 27. Juli 2026 als Verordnung (EU) 2026/1744 in Kraft, hat die EU die Hochrisiko-Anforderungen des AI Act verschoben. Für Anhang-III-Systeme gelten sie ab dem 2. Dezember 2027 statt ab dem 2. August 2026, für in Produkte eingebettete KI-Systeme nach Anhang I ab dem 2. August 2028 statt ab dem 2. August 2027.[41] Die Transparenzpflichten nach Artikel 50 gelten dagegen unverändert seit dem 2. August 2026. Nur für die maschinenlesbare Kennzeichnung synthetischer Inhalte bei Systemen, die schon vorher auf dem Markt waren, läuft eine Schonfrist bis zum 2. Dezember 2026.[41]

Ein Aufschub bedeutet mehr Zeit für dieselbe Arbeit. Aus der Verschiebung den Schluss zu ziehen, die Dokumentations- und Logging-Anforderungen seien weniger dringend, übersieht zwei Dinge. Erstens: Die Anforderungen selbst sind unverändert, verschoben ist allein die Frist. Zweitens: Die Anforderungen sind so beschaffen, dass sie sich nur dann ohne disproportionalen Aufwand erfüllen lassen, wenn sie von Anfang an in die MLOps-Pipeline integriert werden. Ein Modell-Inventar, das Klassifikationen, Datenherkunft, Evaluierungs-Ergebnisse und Risikobewertung jeder produktiven Version mitführt, ist eine Investition, die sich technisch und regulatorisch trägt. Eine retroaktive Dokumentation Ende 2027 ist es nicht. Die zusätzlichen Monate sind Vorbereitungszeit.

5.2 ML Lifecycle Management

Abschnitt 5.1 hat MLOps als Disziplin beschrieben, ihre Reifegrade verortet und die fünf Dimensionen benannt, an denen Adoption gelingt oder scheitert. Dieser Abschnitt geht eine Ebene tiefer und betrachtet den Lebenszyklus eines Modells als das, was er im Betrieb tatsächlich ist: ein Kreislauf, der mit dem ersten Experiment beginnt und mit dem Außerdienststellen des letzten produktiven Modells endet. Dazwischen liegen Versionierung, Auslieferung, Beobachtung und, wenn die Beobachtung es verlangt, ein erneutes Training. Drei Bausteine tragen diesen Kreislauf, und ein vierter wacht über ihn. Die ersten drei sind Experiment Tracking, Model Registry und Feature Store. Der vierte ist das Monitoring.

Eine Vorbemerkung zur Begriffslage. „Lifecycle Management" klingt nach Software-Verwaltung, nach Lizenzen und Versionsständen. Gemeint ist etwas Anspruchsvolleres. Ein Modell altert, während die Software, in der es läuft, unverändert bleibt. Der Code ist derselbe, die Welt, über die das Modell Aussagen trifft, verschiebt sich. Lifecycle Management für ML ist deshalb die Disziplin, die ein Modell über seine gesamte produktive Lebensdauer nachvollziehbar, reproduzierbar und beobachtbar hält, von der Idee bis zur kontrollierten Abschaltung.

Drei Bausteine: Experiment Tracking, Model Registry, Feature Store

Am Anfang jedes Modells steht eine Phase, in der mehr verworfen als behalten wird. Data Scientists probieren Merkmale aus, variieren Hyperparameter, tauschen Algorithmen, messen Ergebnisse. Diese Phase ist von Natur aus chaotisch, und genau deshalb braucht sie ein Gedächtnis. **Experiment Tracking** ist dieses Gedächtnis. Jeder Trainingslauf wird mit seinen Eingaben, seiner Konfiguration und seinen Ergebnissen festgehalten: welche Daten, welche Hyperparameter, welche Metriken, welche Code-Version. Werkzeuge wie MLflow oder Weights & Biases haben sich dafür etabliert.[42] Ohne diese Protokollierung lässt sich später nicht mehr rekonstruieren, warum ein bestimmter Lauf besser war als die anderen, und das beste Ergebnis ist genau das, das man in Produktion bringen will. Ein nicht protokolliertes Experiment ist ein Ergebnis, das man einmal hatte und nicht wiederfindet.

Was im Experiment überzeugt, wandert in die **Model Registry**. Sie ist das zentrale Verzeichnis, das jede Modellversion mit ihren Metadaten führt: Herkunft, Trainingsdaten, Evaluierungs-Ergebnisse, Freigabe-Status, Roll-back-Pfad. Kapitel 5.1 hat die Registry bereits als Versionierungs-Artefakt der CI/CD-Strecke erwähnt. Im Lebenszyklus betrachtet hat sie eine zweite Funktion, die über die Technik hinausreicht. Sie ist der Ort, an dem ein Modell einen Status bekommt, der eine Entscheidung dokumentiert. Ein Modell ist in Erprobung, ist freigegeben, ist produktiv, ist abgelöst. Jeder Übergang ist eine Freigabe mit einem Verantwortlichen und einem Datum. Damit wird die Registry zum Bindeglied zwischen technischer Versionierung und der Governance, die Kapitel 6 behandelt. Annex IV des EU AI Act verlangt für Hochrisikosysteme eine lückenlose technische Dokumentation, und die Registry ist der Ort, an dem diese Dokumentation im laufenden Betrieb entsteht.[43]

Der dritte Baustein wurde in Kapitel 3 bereits eingeführt und kehrt hier mit einer betrieblichen Aufgabe zurück: der **Feature Store**. In der Datenarchitektur ist er das Mittel gegen *training-serving skew*, also gegen die Abweichung zwischen den Merkmalswerten, die ein Modell im Training sieht, und denen, die es in Produktion bekommt. Im Lebenszyklus betrachtet leistet er zwei Dinge zugleich. Er stellt sicher, dass ein Merkmal an beiden Stellen identisch berechnet wird, und er macht einmal definierte Merkmale für andere Modelle wiederverwendbar. Ein Unternehmen, das den durchschnittlichen Kundenumsatz der letzten 90 Tage einmal sauber definiert hat, muss diese Definition nicht für jedes neue Modell erneut schreiben und erneut falsch machen. In der be-

reits zitierten Fallstudie gehörten Feature Stores zu den am wenigsten verbreiteten Praktiken.[44] Das ist insofern bemerkenswert, als der Feature Store von allen drei Bausteinen derjenige ist, der einen produktiven Fehler am direktesten verhindert.

Diese drei Bausteine greifen ineinander. Das Experiment Tracking dokumentiert, wie ein Modell entstanden ist, die Registry, welche Version produktiv ist und warum. Im Feature Store liegt, mit welchen Eingaben sie rechnet. Fehlt einer der drei, reißt die Kette der Nachvollziehbarkeit an genau der Stelle, an der eine regulatorische Rückfrage oder ein Produktionsfehler sie auf die Probe stellt.

Continuous Training und automatisches Retraining

Ein Modell, das einmal trainiert und nie wieder angefasst wird, ist ein Modell auf Zeit. Die Welt, aus der seine Trainingsdaten stammen, verändert sich: Kunden verhalten sich anders, Preise verschieben sich, ein neues Produkt taucht im Sortiment auf, eine Pandemie ändert über Nacht jedes Nachfragemuster. Der Befund zur Modellalterung vom Anfang dieses Kapitels, eine Verschlechterung in 91 Prozent der untersuchten Kombinationen aus Modell und Datensatz, ist die empirische Kehrseite dieser Beobachtung.[45] **Continuous Training**, in Kapitel 5.1 als dritter Bestandteil der ML-spezifischen CI/CD-Strecke eingeführt, ist die betriebliche Antwort darauf.

Der Reflex, daraus „dann trainieren wir eben dauernd neu" zu machen, führt in die Irre. Häufiges Retraining ist teuer, es verbraucht Rechenzeit, und jedes neue Modell ist ein neues Risiko, das getestet und freigegeben werden will. Die eigentliche Entwurfsentscheidung liegt in der Frage, *wann* neu trainiert wird. Drei Auslöser haben sich etabliert. Der erste ist zeitbasiert: ein fester Takt, etwa wöchentlich oder monatlich, einfach umzusetzen und blind gegenüber der Frage, ob das Modell überhaupt schlechter geworden ist. Der zweite ist datenbasiert: Sobald eine definierte Menge neuer Daten vorliegt, läuft die Pipeline an. Der dritte ist der anspruchsvollste und der wirksamste: ein Signal aus dem Monitoring. Erst wenn die Beobachtung des produktiven Modells eine messbare Verschlechterung oder eine Verschiebung der Eingangsdaten meldet, wird neu trainiert.[46] Dieser dritte Auslöser schließt den Kreis, der das Lifecycle Management ausmacht. Das Monitoring stößt das Retraining an, das Retraining erzeugt eine neue Version in der Registry, die neue Version wird beobachtet, und der Kreis beginnt von vorn.

Ein Wort zur Vorsicht, das in der Begeisterung über Automatisierung leicht untergeht. Ein automatisiertes Retraining ohne automatisierte Qualitätskontrolle ist eine Maschine, die zuverlässig auch schlechtere Modelle in Produktion bringt. Die in Kapitel 5.1 beschriebenen Modelltests gegen einen Referenz-Datensatz und das Shadow Deployment sind hier keine Kür. Sie sind die Bedingung, unter der automatisches Retraining überhaupt verantwortbar ist. Ein frisch trainiertes Modell wird erst dann produktiv, wenn es gegen seinen Vorgänger gemessen hat und die Zahlen tragen. Die Pflege der Trainingsdaten verdient dieselbe Sorgfalt: Lernt ein Modell kontinuierlich aus produktiven Daten, die es teilweise selbst erzeugt hat, kann es seine eigenen Fehler verstärken. Eine durchdachte Continuous-Training-Strecke trennt deshalb sauber, welche Daten als Wahrheit gelten und welche das Modell nur produziert hat.

Model Monitoring: Drift, Performance, Bias

Software-Monitoring fragt, ob ein System läuft. Verfügbarkeit, Antwortzeit, Fehlerrate. Ein Modell kann diese Prüfungen alle bestehen und trotzdem falsch liegen. Es antwortet schnell, es antwortet zuverlässig, und es liegt dabei zunehmend daneben. Model Monitoring beobachtet deshalb eine Ebene, die das klassische Monitoring nicht kennt: die inhaltliche Qualität der Vorhersagen über die Zeit. Drei Phänomene stehen dabei im Vordergrund.

Das erste ist **Drift**, und es lohnt, zwei Formen zu unterscheiden. Von *Data Drift* spricht man, wenn sich die Verteilung der Eingangsdaten verschiebt. Das Modell bekommt Eingaben, wie es sie im Training so nicht gesehen hat, etwa weil eine neue Kundengruppe hinzukommt. Von *Concept Drift* spricht man, wenn sich der Zusammenhang zwischen Eingabe und korrekter Ausgabe selbst verändert. Die Eingaben sehen aus wie immer, aber die richtige Antwort ist eine andere geworden, weil sich die Welt darunter gedreht hat.[47] Beide Formen führen zur selben Folge, einem still sinkenden Modell, und beide verlangen unterschiedliche Antworten. Data Drift lässt sich messen, indem man die Verteilung der aktuellen Eingaben mit der Trainingsverteilung vergleicht, und das ist auch ohne sofortiges Wissen über die korrekte Antwort möglich. Concept Drift zeigt sich erst, wenn die tatsächlichen Ergebnisse vorliegen, und ist deshalb schwerer und langsamer zu erkennen.

Das zweite ist die **Performance** im engeren Sinn, also die Frage, wie gut die Vorhersagen tatsächlich treffen. Hier liegt die eigentliche Schwierigkeit produktiven Monitorings. Die Güte einer Vorhersage lässt sich erst beurteilen, wenn die Wirklichkeit sie eingeholt hat. Ob die Kreditausfall-Prognose richtig war, weiß man, wenn der Kredit bedient oder nicht bedient wurde, also Monate später. Diese Verzögerung, die *ground truth latency*, bedeutet, dass ein rein performance-basiertes Monitoring den Schaden immer erst im Rückblick sieht. Genau deshalb ist die Drift-Messung der Eingangsdaten so wertvoll, denn sie ist ein Frühindikator, der nicht auf die verspätete Wahrheit warten muss. Ein Unternehmen ohne Alerting bemerkt eine Verschlechterung erst, wenn ein Fachbereich sich beschwert oder eine Auswertung den Schaden offenlegt, und zu diesem Zeitpunkt sind die fehlerhaften Entscheidungen längst getroffen.

Das dritte ist **Bias**, und es ist die Dimension, in der Monitoring und Governance unmittelbar zusammentreffen. Ein Modell, das sich über die Zeit verschiebt, kann sich auch in eine Richtung verschieben, die bestimmte Gruppen systematisch benachteiligt. Das kann von Anfang an im Modell stecken, etwa weil die Trainingsdaten eine historische Schieflage abbilden, oder es kann durch Drift entstehen, wenn sich die Datenlage einer Gruppe anders entwickelt als die einer anderen. Bias-Monitoring beobachtet die Vorhersagequalität getrennt nach relevanten Gruppen und schlägt an, wenn sie auseinanderläuft. Für Hochrisikosysteme ist das Pflicht. Der EU AI Act verlangt für solche Systeme Maßnahmen zur Erkennung und Minderung von Verzerrungen über den gesamten Lebenszyklus, und ohne ein Monitoring, das nach Gruppen aufschlüsselt, lässt sich diese Pflicht nicht erfüllen.[48] Kapitel 6 vertieft die regulatorische Seite. Im Lebenszyklus betrachtet gilt: Fairness ist eine Eigenschaft, die ein Modell im Betrieb verlieren kann, und deshalb muss sie im Betrieb fortlaufend gemessen werden, über die einmalige Prüfung vor der Freigabe hinaus.

Damit schließt sich der Kreislauf dieses Kapitels. Die drei Bausteine halten das Modell nachvollziehbar, das Continuous Training hält es aktuell, und das Monitoring entscheidet, wann beides nötig wird. Ein Lebenszyklus ohne Monitoring ist ein Kreislauf ohne Steuerung. Er läuft, aber niemand weiß, in welche Richtung. Was diese betriebliche Disziplin von der Architektur unterscheidet, ist allein die Perspektive. Kapitel 4 hat Observability, Registry und Feature Store als Anforderungen formuliert. Hier sind sie zur täglichen Praxis geworden, an der sich entscheidet, ob ein Modell nach längerer Laufzeit noch das tut, wofür es gebaut wurde.

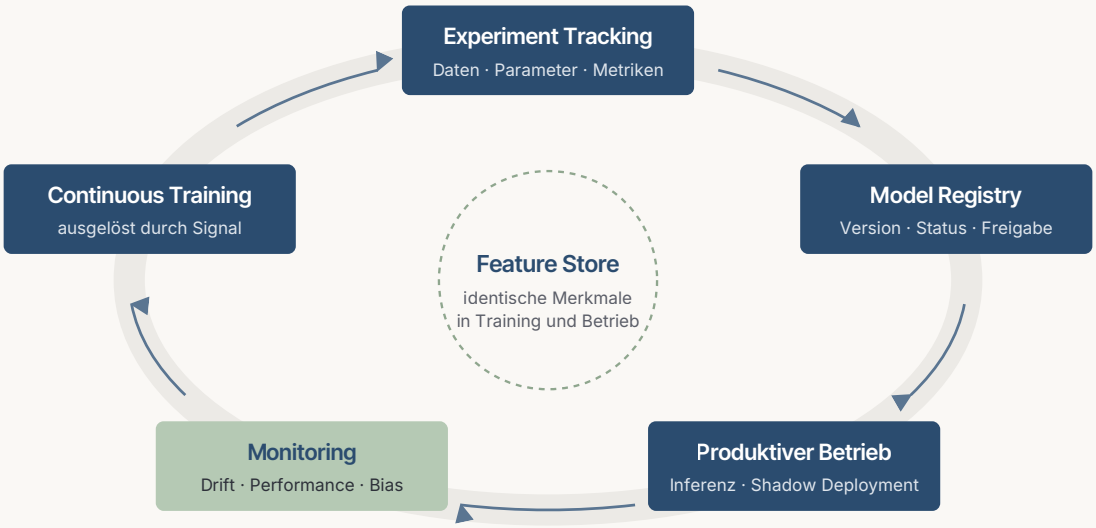


Abbildung 5.1 · Der ML-Lebenszyklus als Kreislauf. Experiment Tracking, Model Registry und Feature Store halten das Modell nachvollziehbar, das Monitoring stößt das Retraining an.

5.3 GenAI in Produktion: Die besonderen Herausforderungen

Die beiden vorigen Abschnitte haben den Betrieb klassischer Modelle beschrieben, von der Reifegradverortung bis zum Lebenszyklus. Generative Modelle erben diese Disziplin und stellen sie zugleich vor Aufgaben, die sie so nicht kannte. Ein klassisches Modell trifft eine eng umrissene Vorhersage, eine Zahl, eine Klasse, eine Wahrscheinlichkeit. Ein großes Sprachmodell erzeugt freien Text, und damit verschieben sich drei Dinge auf einmal. Das Modell selbst kommt in den meisten Fällen als Dienst eines Anbieters ins Unternehmen, sein Training liegt außerhalb. Die Qualität der Ausgabe lässt sich nicht mehr an einer einzelnen Metrik ablesen. Und die laufenden Kosten entstehen pro Anfrage, statt einmalig beim Training.

Diese Verschiebungen haben einer eigenen Disziplin den Namen gegeben: **LLMOps**, am Anfang dieses Kapitels als jüngere Schwester von MLOps eingeführt. Sie ist 2026 deutlich weniger erprobt als MLOps und in den meisten Unternehmen, die generative Anwendungen produktiv betreiben, die akutere Baustelle. Die Werkzeuge überlappen sich, die Teams sind oft dieselben, aber die Fragen sind andere. Dieser Abschnitt behandelt die vier, die im Produktionsbetrieb am häufigsten unterschätzt werden: den veränderten Zuschnitt der Disziplin, das Management von Prompts, die Bewertung freier Ausgaben und die Steuerung der Inference-Kosten.

LLMOps: MLOps für große Sprachmodelle

Der erste Unterschied betrifft die Frage, woran überhaupt gearbeitet wird. In klassischem MLOps ist das Modell das zentrale Artefakt, das trainiert, versioniert und ausgeliefert wird. Bei generativen Anwendungen liegt das Modell meistens beim Anbieter, erreichbar über eine Schnittstelle. Was im eigenen Unternehmen entsteht und gepflegt wird, ist das, was um das Modell herum liegt: der Prompt, der die Aufgabe beschreibt, der Kontext, der aus den eigenen Daten beigesteuert wird, die Kette von Aufrufen, die eine komplexe Anfrage in Teilschritte zerlegt, und die Leitplanken, die unerwünschte Ausgaben abfangen.[49] Die Iteration verschiebt sich entsprechend. Statt eines Trainingslaufs über Stunden verändert ein Entwickler einen Prompt in Sekunden und sieht das Ergebnis sofort. Diese Geschwindigkeit ist ein Gewinn und eine Gefahr zugleich. Sie verführt dazu, am produktiven System zu drehen, ohne festzuhalten, was gedreht wurde.

Der zweite Unterschied ist die Abhängigkeit von einem fremden Modell, das sich ohne Zutun des Unternehmens verändern kann. Ein Anbieter aktualisiert sein Modell, und ein Prompt, der gestern verlässlich funktioniert hat, liefert heute eine andere Ausgabe. Eine wissenschaftliche Untersuchung dieser Frage hat gezeigt, dass sich das Verhalten kommerzieller Sprachmodell-Schnittstellen über aufeinanderfolgende Versionen messbar verschiebt, ohne dass die aufrufende Anwendung etwas verändert hätte.[50] Für den Betrieb folgt daraus eine Konsequenz, die MLOps in dieser Form nicht kennt. Das Unternehmen muss eine Versionsänderung beobachten können, die außerhalb seiner Kontrolle stattfindet. Eine Anwendung, die ohne festgepinnte Modellversion und ohne regelmäßige Nachprüfung läuft, altert auf eine Weise, die im Code nicht sichtbar ist.

Der dritte Unterschied liegt in der Regulierung. Mit den Pflichten für Anbieter von KI-Modellen mit allgemeinem Verwendungszweck, die seit dem 2. August 2025 gelten, hat der EU AI Act eine Informationskette etabliert.[51] Der Anbieter eines Basismodells muss nachgelagerten Anbietern, die das Modell in eigene Systeme einbauen, Informationen über Fähigkeiten und Grenzen bereitstellen, damit diese ihre eigenen Pflichten erfüllen können. Ein Unternehmen, das ein fremdes Modell produktiv einsetzt, ist ein solcher nachgelagerter Anbieter. Es übernimmt die Verantwortung für das, was seine Anwendung tut, und kann sich dabei auf die Modelldokumentation des Anbieters stützen, aber nicht hinter ihr verstecken.

Prompt-Management und Versionierung

In klassischer Software liegt die Logik im Code. In einer generativen Anwendung liegt ein erheblicher Teil davon im Prompt, in den Anweisungen, die dem Modell sagen, was es tun soll, in welchem Ton, mit welchen Grenzen. Eine kleine Umformulierung kann die Ausgabe spürbar verändern. Solange dieser Prompt als Zeichenkette im Quellcode steht, fällt jede Änderung durch das Raster, das für Modelle und Daten längst etabliert ist. Niemand weiß hinterher, welche Fassung welche Antwort erzeugt hat. Die Praxis, die sich daraus ableitet, lässt sich auf eine Formel bringen, die in der Disziplin zur Standardformulierung geworden ist: Prompts sind Code.[52] Sie werden versioniert, getestet und freigegeben wie jedes andere Artefakt, mit einer Versionsgeschichte, die jede produktive Antwort auf die Anweisung zurückführt, die sie erzeugt hat.

Aus dieser Behandlung folgt eine zweite Anforderung, die unmittelbar an die CI/CD-Strecke aus Kapitel 5.1 anschließt: das Regressionsproblem. Eine neue Prompt-Fassung, die ein Problem behebt, kann an anderer Stelle ein neues schaffen. Klassische Regressionstests prüfen das, indem sie definierte Eingaben gegen definierte Ausgaben halten. Bei freiem Text greift dieser Mechanismus nicht, weil zwei unterschiedlich formulierte Antworten beide richtig sein können. Die bereits zitierte Untersuchung zur Verschlechterung von Prompts über Modellversionen hinweg leitet daraus ab, dass Regressionstests für Sprachmodelle grundlegend anders zugeschnitten werden müssen als für klassische Software: Korrektheit ist uneinheitlich definiert, Prompts reagieren empfindlich auf kleine Änderungen, und die Ausgaben sind nicht deterministisch.[53] In der Praxis läuft eine neue Prompt-Fassung gegen einen festen Satz von Testfällen, und ihre Ausgaben werden bewertet, bevor sie produktiv geht. Womit sie bewertet werden, ist die Frage des nächsten Abschnitts.

Evaluation von GenAI-Ausgaben in Produktion

Die Bewertung freier Ausgaben ist die schwierigste der vier Aufgaben und die, an der LLMOps sich am deutlichsten von MLOps entfernt. Ein klassisches Modell wird gegen Ground Truth gemessen, gegen eine bekannte richtige Antwort. Bei einem Sprachmodell gibt es diese eine richtige Antwort oft nicht. Eine Zusammenfassung kann auf zehn Arten gut sein und auf zwanzig Arten schlecht, und ob sie das eine oder das andere ist, hängt vom Kontext und vom Leser ab. Drei Wege haben sich etabliert, die Qualität dennoch messbar zu machen, und keiner von ihnen genügt allein.

Der erste ist die menschliche Bewertung. Sie ist die verlässlichste und die teuerste. Sie skaliert schlecht und eignet sich für Stichproben und für die Kalibrierung der beiden anderen Verfahren, nicht für die laufende Prüfung jeder produktiven Ausgabe. Der zweite Weg sind objektive Kennzahlen für die Fälle, in denen es sie gibt. Bei einer RAG-Anwendung etwa lässt sich messen, ob die Antwort durch die abgerufenen Quellen gedeckt ist, und eine ungedeckte Aussage ist ein belastbares Signal für eine Halluzination. Der dritte Weg hat sich 2025 als pragmatischer Mittelweg durchgesetzt: ein zweites Sprachmodell bewertet die Ausgabe des ersten, ein Verfahren, das unter dem Namen *LLM-as-a-Judge* bekannt geworden ist.

Dieser dritte Weg ist verführerisch, weil er skaliert, und gefährlich, weil seine Verlässlichkeit überschätzt wird. Eine empirische Untersuchung aus dem Jahr 2025 hat gezeigt, dass die Bewertungsqualität solcher Modell-Richter stark von Designentscheidungen abhängt, allen voran von klar formulierten Bewertungskriterien, aber auch von der Art der Stichprobenziehung.[54] Daraus folgt nach unserer Einschätzung: Ein Richter, der auf Chat-Dialogen validiert wurde, braucht eine erneute Kalibrierung, bevor er RAG-Antworten oder Agenten-Aufgaben verlässlich beurteilt. Die praktische Lehre daraus ist doppelt. Ein Modell-Richter braucht dieselbe Sorgfalt wie das System, das er bewertet, eigene Kriterien, eigene Validierung gegen menschliche Urteile, eine regelmäßige Nachprüfung. Und er streckt die menschliche Bewertung. Die menschliche Stichprobe bleibt der Anker, an dem sich der automatische Richter messen lassen muss.

Über alle drei Wege legt sich eine betriebliche Anforderung, die in Kapitel 5.2 für klassische Modelle schon galt und für generative ungleich dringender ist: Die Bewertung gehört auch in die Produktion. Ein Eval-Set, das einmal vor der Freigabe durchlief und danach nie wieder, übersieht die Verschiebung des fremden Modells ebenso wie die schleichende Veränderung der Nutzeranfragen. Evaluierungs-Sets werden mit der Anwendung gepflegt und laufen im Takt mit, als stehende Einrichtung der Produktion.

Kosten-Management für LLM-Inference

Die letzte Verschiebung betrifft das Geld. Ein klassisches Modell verursacht seine Hauptkosten beim Training und ist danach im Betrieb vergleichsweise billig. Eine generative Anwendung dreht dieses Verhältnis um. Das Training entfällt, weil das Modell vom Anbieter kommt, und stattdessen entstehen Kosten bei jeder einzelnen Anfrage, abgerechnet nach verarbeiteten Token. Diese Kosten sind über die Zeit dramatisch gefallen. Der AI Index Report des Stanford HAI weist aus, dass der Preis für eine Anfrage auf dem Leistungsniveau von GPT-3.5 zwischen November 2022 und Oktober 2024 von zwanzig Dollar auf sieben Cent je Million Token gesunken ist, ein Rückgang um mehr als das 280-Fache.[55] Diese Zahl verführt zu dem Schluss, Inference-Kosten seien ein Problem von gestern. Im produktiven Betrieb sind sie es selten.

Der Grund liegt in der Skalierung. Was pro Anfrage günstig ist, summiert sich, sobald eine Anwendung Tausende Anfragen am Tag bedient, jede mit langem Kontext, manche in Ketten aus mehreren Modellaufrufen. Ohne Kostensteuerung wachsen die Betriebskosten einer generativen Anwendung mit ihrer Nutzung, ohne dass jemand die Kurve im Blick hat. Die Steuerung folgt denselben FinOps-Prinzipien, die Kapitel 3 für die Infrastruktur beschrieben hat, übersetzt auf die Einheit, die hier zählt: das Token. Den Anfang macht die Sichtbarkeit. Kosten je Anfrage, je Anwendungsfall und je Nutzergruppe gehören in dieselbe Beobachtungsschicht, in der schon Qualität und Drift überwacht werden.

Auf die Sichtbarkeit folgen die Hebel, und die wirksamsten setzen an, bevor eine Anfrage überhaupt das Modell erreicht. Das Zwischenspeichern wiederkehrender Modellantworten erspart den Aufruf ganz, wenn dieselbe oder eine sehr ähnliche Anfrage erneut gestellt wird. Das Zwischenspeichern wiederkehrender Prompt-Anteile beim Anbieter senkt den Preis der Aufrufe, die trotzdem nötig sind. Die Wahl des Modells nach Schwere der Aufgabe verhindert, dass ein teures Spitzenmodell eine Aufgabe erledigt, für die ein kleineres genügt hät-

te. Das Bündeln nicht eiliger Anfragen in zeitversetzte Stapelverarbeitung nutzt die günstigeren Tarife, die Anbieter dafür gewähren. Und die Verdichtung des Kontexts, der mit jeder Anfrage mitgeschickt wird, senkt die Tokenzahl an der Stelle, an der sie am unauffälligsten wächst. Die Preislisten der Anbieter zeigen die Größenordnung einzelner Hebel. Anthropic und OpenAI gewähren für zeitversetzte Stapelverarbeitung 50 Prozent Rabatt; bei Anthropic kostet ein aus dem Zwischenspeicher gelesener Prompt-Anteil standardmäßig zehn Prozent des regulären Eingabepreises.[56] Wie viel davon auf eine konkrete Anwendung entfällt, hängt von ihrem Lastprofil ab. Für typische Produktivlasten schätzen wir das gemeinsame Einsparpotenzial von Zwischenspeichern, Modellwahl, Stapelverarbeitung und Kontextverdichtung auf 50 bis 90 Prozent der Inference-Kosten einer ungesteuerten Anwendung.

Damit schließt Kapitel 5 den Bogen vom klassischen Modell zum generativen Dienst. Was beide teilen, ist die Einsicht, dass der Betrieb über den Wert einer KI-Investition genauso entscheidet wie die Architektur, die ihr zugrunde liegt. Ein unbeobachtetes Modell verfällt. Ein nicht versionierter Prompt entzieht sich der Rückverfolgung in dem Moment, in dem eine Antwort erklärt werden muss. Eine ungeprüfte Ausgabe ist ein Versprechen ohne Deckung, und Inference-Kosten, die niemand verfolgt, werden teuer, lange bevor es auffällt. Wie aus diesen betrieblichen Einsichten ein Weg für das einzelne Unternehmen wird, behandelt Kapitel 7. Was vorher noch fehlt, ist die regulatorische Klammer um alles Bisherige, und die ist Gegenstand von Kapitel 6.

KAPITEL 6

6 Governance und Compliance: Architektur unter regulatorischem Druck

Die bisherigen Kapitel haben beschrieben, wie eine KI-Architektur entsteht, worauf sie steht und wie sie betrieben wird. Über allem lag eine Schicht, die in Kapitel 2 als Querschnitt eingeführt und seitdem aufgeschoben wurde: Governance und Sicherheit. Dieses Kapitel holt sie ein.

Der Aufschub war Absicht. Governance lässt sich erst sinnvoll behandeln, wenn klar ist, woran sie ansetzt: an Daten, an Modellen, an Plattformen, an dem Betrieb, der sie am Laufen hält. Diese Gegenstände stehen jetzt im Raum. Was folgt, ist die Frage, unter welchen Regeln sie betrieben werden müssen, und wie diese Regeln in die Architektur eingebaut werden.

Vier Abschnitte teilen den Stoff. Abschnitt 6.1 behandelt den EU AI Act, das Regelwerk, das die KI-Governance in Europa rahmt, seine Folgen für die Architektur und die deutsche Aufsicht nach dem KI-MIG. Abschnitt 6.2 beschreibt das Governance-Rahmenwerk innerhalb des Unternehmens: Gremien, Entscheidungspunkte, Verantwortlichkeiten. Abschnitt 6.3 nimmt die Sicherheit in den Blick, mit den neuen Angriffsklassen, die KI mitbringt, und den Anforderungen aus NIS-2. Abschnitt 6.4 schließt mit dem Datenschutz und seinen architektonischen Mustern.

Ein Wort zur Haltung vorab. Governance hat im Diskurs den Ruf des Bremsklotzes. Dieses Kapitel argumentiert in die andere Richtung. Die regulatorischen Anforderungen, richtig gelesen, sind Designvorgaben, die eine Architektur ohnehin braucht, um produktiv zu tragen. Als Architekturprinzip behandelt, kosten sie keine zusätzliche Zeit. Sie ersparen das zweite Bauen.

6.1 EU AI Act: Architektur-Implikationen

Am 1. August 2024 ist die Verordnung (EU) 2024/1689 in Kraft getreten, im allgemeinen Sprachgebrauch der *EU AI Act*.[57] Sie ist das erste umfassende Gesetz weltweit, das Künstliche Intelligenz horizontal reguliert, über Branchen und Anwendungen hinweg. Für jedes Unternehmen, das KI in Europa entwickelt, einkauft oder betreibt, ist sie der regulatorische Rahmen, in dem Architektur entworfen wird. Und sie greift, anders als viele Compliance-Themen, in die Bauentscheidung selbst.

Dieser Abschnitt klärt vier Dinge. Erstens, wie der AI Act KI-Systeme nach Risiko einteilt und was diese Einteilung für die Architektur bedeutet. Zweitens, welche technischen Anforderungen ein Hochrisiko-System erfüllen muss. Drittens, wie der Nachweis dieser Erfüllung organisiert ist, das sogenannte Konformitätsbewertungsverfahren. Viertens, warum all das in die Architektur gehört. Vorab aber ein Blick auf den Zeitplan, denn der hat sich 2026 verschoben.

Der Zeitplan nach dem Digital Omnibus

Die ursprüngliche Verordnung sah einen gestaffelten Fahrplan vor: verbotene Praktiken ab dem 2. Februar 2025, Pflichten für Modelle mit allgemeinem Verwendungszweck ab dem 2. August 2025, Transparenzpflichten und Pflichten für eigenständige Hochrisiko-Systeme nach Anhang III ab dem 2. August 2026, Pflichten für Hochrisiko-Systeme in regulierten Produkten nach Anhang I ab dem 2. August 2027. An den Hochrisiko-Terminen hängt für die meisten Unternehmen die eigentliche Last, und genau diese Termine haben sich verschoben.

Im Rahmen des sogenannten *Digital Omnibus*, eines Pakets gezielter Vereinfachungen der EU-Digitalgesetzgebung, haben Rat und Parlament sich am 7. Mai 2026 vorläufig geeinigt. Der Rat hat am 29. Juni 2026 endgültig zugestimmt, die Verordnung (EU) 2026/1744 ist am 24. Juli 2026 im Amtsblatt erschienen und seit dem 27. Juli 2026 in Kraft.[58] Für eigenständige Hochrisiko-Systeme nach Anhang III, also etwa KI in Personalauswahl, Bildung, Kreditvergabe oder Biometrie, gilt die Pflicht künftig ab dem 2. Dezember 2027. Für Hochrisiko-Systeme, die in regulierte Produkte nach Anhang I eingebettet sind, etwa in Medizinprodukte oder Maschinen, verschiebt sie sich auf den 2. August 2028. Der Hintergrund ist nüchtern: Die harmonisierten Normen, die den Unternehmen einen gangbaren Weg zur Erfüllung weisen sollen, sind noch nicht fertig, und ohne sie wäre die Pflicht in Kraft getreten, bevor der Weg zu ihrer Erfüllung ausgeschildert war.[59]

Mit der Veröffentlichung im Amtsblatt ist die Verschiebung geltendes Recht geworden, rechtzeitig vor dem ursprünglichen Stichtag des 2. August 2026. Für die Architekturarbeit ändert sie weniger, als die Schlagzeilen vermuten lassen, und das ist der eigentliche Punkt dieses Abschnitts. Sie verschafft Zeit für die Umsetzung. Die technischen Anforderungen, die ein Hochrisiko-System am Ende erfüllen muss, bleiben dieselben. Ein Aufschub der Frist lässt die Arbeit unverändert; er ist die Gelegenheit, sie ohne Hektik richtig zu machen.

Andere Pflichten wurden nicht verschoben. Die Transparenzpflichten nach Artikel 50 gelten wie ursprünglich vorgesehen seit dem 2. August 2026: Ein Chatbot muss erkennbar machen, dass er eine Maschine ist, und synthetisch erzeugte Inhalte müssen gekennzeichnet sein. Für die maschinenlesbare Kennzeichnung nach Artikel 50 Absatz 2 gewährt der Omnibus nur Systemen, die vor dem 2. August 2026 in Verkehr gebracht wurden, eine Schonfrist bis zum 2. Dezember 2026.[60] Neu hinzu kommen zwei verbotene Praktiken nach Artikel 5: KI-Systeme, die nicht einvernehmliche intime Darstellungen oder Darstellungen sexuellen Kindesmissbrauchs erzeugen oder manipulieren, sind ab dem 2. Dezember 2026 verboten.[60] Für Betreiber generativer Anwendungen, deren Ausgaben Bilder, Audio oder Video umfassen, ist die Kennzeichnung damit eine Architekturaufgabe, die bereits heute gilt.

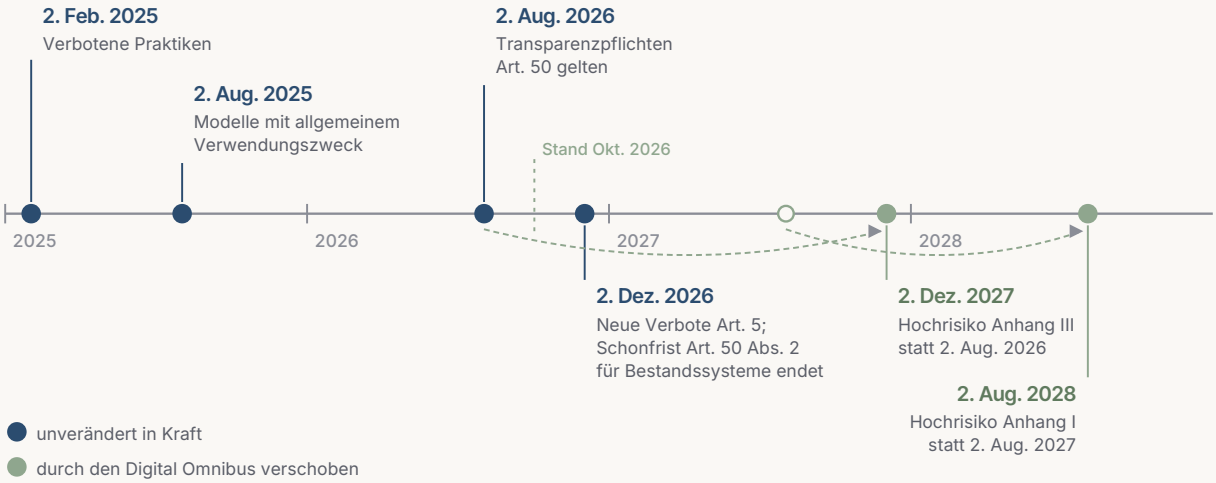


Abbildung 6.1 · Fristen der KI-Verordnung nach dem Digital Omnibus. Verschoben sind die Hochrisiko-Fristen, die Transparenzpflichten nach Artikel 50 gelten seit dem 2. August 2026.

KI-Kompetenz und die deutsche Aufsicht

Eine Pflicht gilt bereits seit dem 2. Februar 2025 für jedes Unternehmen, das KI-Systeme anbietet oder betreibt, gleich welcher Risikoklasse: die Förderung der KI-Kompetenz nach Artikel 4. Der Digital Omnibus hat sie neu gefasst. Anbieter und Betreiber müssen Maßnahmen ergreifen, die die KI-Kompetenz ihres Personals und der in ihrem Auftrag tätigen Personen fördern, abgestimmt auf Vorwissen, Einsatzkontext und die betroffenen Personen. Ein bestimmtes Kompetenzniveau müssen sie nicht garantieren.[61] Auch für die Architektur zählt diese Pflicht: Ein freigegebenes KI-Angebot mit Nutzungsregeln, Schulung und dokumentierten Einsatzzwecken ist zugleich der Nachweis für Artikel 4 und die wirksamste Antwort auf Schatten-KI (Kapitel 7.4).

In Deutschland ist seit dem 29. Juli 2026 geklärt, wer die Einhaltung überwacht. Das Gesetz zur Marktüberwachung und Innovationsförderung von künstlicher Intelligenz (KI-MIG) macht die Bundesnetzagentur zur zentralen Marktüberwachungsbehörde, Anlaufstelle und Beschwerdestelle für die KI-Verordnung. In ihre Zuständigkeit fallen unter anderem KI-Systeme in Funkanlagen und in sensiblen Bereichen wie Personalwesen, kritischer Infrastruktur und Bildung. Wo Produktbereiche bereits vollständig harmonisiert sind oder eine Fachaufsicht besteht, bleibt diese zuständig, im Finanzsektor etwa die BaFin.[62] Für kleine und mittlere Unternehmen bietet die Bundesnetzagentur einen KI-Service-Desk mit einem Online-Werkzeug zur Risikoeinstufung sowie ein KI-Reallabor, in dem innovative Systeme unter Aufsicht entwickelt und erprobt werden können.[62] Ein Unternehmen weiß damit, an wen es sich mit Einstufungsfragen wendet und wer im Ernstfall prüft.

Wie der AI Act Risiko einteilt, und was das für die Architektur heißt

Der AI Act reguliert KI nach ihrem Verwendungszweck und dem Risiko, das daraus folgt; die eingesetzte Technik spielt für die Einstufung keine Rolle. Das Gesetz kennt vier Stufen, und die Architektur wird von ihnen unterschiedlich stark berührt.

Auf der untersten Stufe steht die große Mehrheit der Anwendungen: KI mit **minimalem Risiko**. Ein Spamfilter, eine Produktempfehlung, eine Routenoptimierung. Für sie gelten keine besonderen Pflichten aus dem AI Act. Architektonisch sind sie unauffällig, und das bleiben sie.

Darüber steht KI mit **begrenztem Risiko**, für die Transparenzpflichten gelten. Ein Chatbot muss seinen Nutzern zu erkennen geben, dass sie mit einer Maschine sprechen. Synthetisch erzeugte Inhalte müssen als solche gekennzeichnet werden. Diese Stufe verlangt von der Architektur wenig, aber etwas Konkretes: eine Stelle im System, an der die Kennzeichnung erzeugt und ausgegeben wird, zuverlässig und nicht abschaltbar. Diese Pflichten gelten seit dem 2. August 2026 und sind damit für Betreiber generativer Anwendungen die ersten, die sie heute schon erfüllen müssen.

Die dritte Stufe trägt die Last des Gesetzes: KI mit **hohem Risiko**. Hierher gehören Systeme, die über Zugang zu Bildung, Beschäftigung oder Krediten mitentscheiden, die in kritischer Infrastruktur arbeiten, die biometrische Identifikation leisten oder in Medizinprodukten und Maschinen stecken. Für sie gilt der volle Pflichtenkatalog, den der nächste Abschnitt beschreibt. Hier entscheidet die Architektur über die Erfüllbarkeit. Die meisten dieser Pflichten lassen sich nur erfüllen, wenn das System von Anfang an so gebaut ist, dass es sie erfüllen kann.

Auf der obersten Stufe steht die **verbotene KI**. Praktiken, die der Gesetzgeber für unvereinbar mit europäischen Grundrechten hält und gar nicht erst zulässt: Sozialkredit-Systeme, manipulatives Verhaltens-Targeting, ungezielte Gesichtsdatensammlung, und ab dem 2. Dezember 2026 auch die beiden oben genannten Missbrauchsfälle. Architektonisch ist diese Stufe einfach: Was verboten ist, wird nicht gebaut. Die Schwierigkeit liegt früher, in der ehrlichen Prüfung, ob ein geplantes System in diese Kategorie fällt.

Für die Architekturarbeit folgt aus dieser Staffelung eine erste, oft übersehene Aufgabe. Bevor irgendetwas gebaut wird, muss die Risikoklasse eines geplanten Systems bestimmt werden, denn sie entscheidet über den gesamten weiteren Pflichtenumfang. Diese Einordnung ist im Hochrisiko-Bereich weniger automatisch, als die Anhänge vermuten lassen. Artikel 6 Absatz 3 enthält einen Mechanismus, den die Kommission selbst als Filter bezeichnet: Ein System, das unter eine der Anhang-III-Kategorien fällt, ist trotzdem nicht hochriskant, wenn es kein erhebliches Risiko für Gesundheit, Sicherheit oder Grundrechte birgt, etwa weil es nur eine eng umrissene Vorbereitungsaufgabe erledigt.[63] Die Inanspruchnahme dieses Filters verlangt eine dokumentierte Selbstbewertung, die die Einordnung begründet, bevor das System in Verkehr gebracht oder in Betrieb genommen wird. Der Digital Omnibus hat den Umfang der dabei einzureichenden Informationen gestrafft, die ursprünglich vorgesehene Streichung der Registrierungspflicht für solche Systeme aber wieder zurückgenommen.[63] Die Selbstbewertung bleibt eine Architekturaufgabe, weil sie verlangt, dass man Zweck, Datenfluss und Wirkung eines Systems vor dem Bau versteht. Eine Einordnung, die erst nach der Inbetriebnahme erfolgt, kommt zu spät, um die Architektur noch zu prägen.

Was ein Hochrisiko-System können muss

Der Pflichtenkatalog für Hochrisiko-Systeme steht in den Artikeln 8 bis 15 der Verordnung. Er liest sich auf den ersten Blick wie eine Compliance-Liste. Auf den zweiten Blick ist er eine Architektur-Spezifikation, denn jede einzelne Pflicht setzt voraus, dass das System auf eine bestimmte Weise gebaut wurde. Sechs Anforderungen prägen die Architektur am stärksten.

Die erste ist das **Risikomanagement** über den gesamten Lebenszyklus (Art. 9). Risiken müssen identifiziert, bewertet und gemindert werden, fortlaufend, nicht einmalig zur Abnahme. Das verlangt eine Architektur, in der Risiken überhaupt sichtbar werden, also Telemetrie, Protokollierung, ein definierter Prozess für den Umgang mit erkannten Risiken. Eine Architektur ohne Beobachtbarkeit, wie sie Kapitel 2 als Prinzip beschrieben hat, kann diese Pflicht nicht erfüllen.

Die zweite betrifft **Daten und Daten-Governance** (Art. 10). Trainings-, Validierungs- und Testdaten müssen relevant, repräsentativ und so weit wie möglich fehlerfrei sein, und mögliche Verzerrungen müssen untersucht und gemindert werden. Das ist keine Aufgabe, die sich an die Datenpipeline anhängen lässt. Sie gehört in die Datenschicht, die Kapitel 3 beschrieben hat, und sie verlangt, dass die Herkunft und Qualität der Daten dokumentiert wird, lange bevor ein Modell darauf trifft.

Die dritte ist die **technische Dokumentation** (Art. 11 in Verbindung mit Anhang IV). Sie muss vor der Inbetriebnahme vorliegen und über den Lebenszyklus aktuell gehalten werden. Anhang IV beschreibt neun Bestandteile, von der Systembeschreibung über die Entwicklungsmethode, die Datengrundlage und die Leistungskennzahlen bis zur Konformitätserklärung und dem Plan für die Marktbeobachtung.[64] Eine solche Dokumentation lässt sich am Ende eines Projekts nicht aus dem Gedächtnis rekonstruieren. Sie entsteht mit dem System oder gar nicht. Architektonisch heißt das: Die Artefakte, die in die Dokumentation gehören, müssen beim Bauen anfallen. Hier schließt sich der Bogen zu Kapitel 5, denn Experiment-Tracking, Modell-Registry und Versionierung, dort als Betriebshygiene beschrieben, sind zugleich die Maschine, die diese Dokumentation erzeugt.

Die vierte Pflicht ist die **Aufzeichnung durch Protokollierung** (Art. 12). Hochrisiko-Systeme müssen Ereignisse automatisch protokollieren, um ihre Funktion über die Zeit nachvollziehbar zu machen. Das ist eine Architekturentscheidung auf der Plattformebene.

Die fünfte verlangt **Transparenz und menschliche Aufsicht** (Art. 13 und 14). Das System muss so gebaut sein, dass ein Mensch seine Ausgaben verstehen, überwachen und im Zweifel übersteuern kann. Eine Architektur, die den Menschen erst nachträglich in die Schleife einbaut, erfüllt diese Pflicht schlecht. Die Aufsicht muss als Schnittstelle vorgesehen sein, bevor die erste Entscheidung automatisiert wird.

Die sechste schließlich fordert **Genauigkeit, Robustheit und Cybersicherheit** (Art. 15). Das System muss seine Leistung verlässlich erbringen, gegen Fehler und Manipulation gehärtet sein und seine Genauigkeit über die Zeit halten. Diese Pflicht verbindet den AI Act mit der Sicherheitsarchitektur, die Abschnitt 6.3 behandelt, und mit dem Modell-Monitoring aus Kapitel 5. Sie zeigt, wie die regulatorischen Anforderungen ineinandergreifen.

Sechs Pflichten, ein gemeinsamer Nenner. Keine von ihnen lässt sich nachträglich in ein fertiges System einsetzen, ohne es aufzumachen. Genau deshalb ist der AI Act für Hochrisiko-Anwendungen ein Architekturthema.

Wie der Nachweis organisiert ist: das Konformitätsbewertungsverfahren

Ein Hochrisiko-System darf in der EU erst in Verkehr gebracht werden, wenn seine Konformität mit den genannten Anforderungen bewertet und bestätigt wurde. Dieses Verfahren, das *Conformity Assessment*, kennt zwei Wege, und welcher gilt, hängt von der Art des Systems ab.[65]

Für die meisten eigenständigen Hochrisiko-Systeme nach Anhang III genügt eine **interne Kontrolle** nach Anhang VI. Der Anbieter bewertet selbst, dokumentiert die Bewertung, stellt eine EU-Konformitätserklärung aus und bringt die CE-Kennzeichnung an. Für Systeme, die in regulierte Produkte eingebettet sind, oder für bestimmte biometrische Anwendungen ist die Bewertung durch eine **benannte Stelle** nach Anhang VII vorgesehen, also durch einen externen, akkreditierten Prüfer. In beiden Fällen ist die technische Dokumentation aus Anhang IV das Beweismittel. Sie ist der Nachweis, an dem die Konformität hängt.

Hier wird die viel diskutierte Lücke relevant. Der AI Act sieht vor, dass die Erfüllung seiner Anforderungen über harmonisierte Normen nachgewiesen werden kann; erst die Zitierung einer solchen Norm im Amtsblatt begründet die Vermutung, konform zu sein. Die europäischen Normungsorganisationen CEN und CENELEC erarbeiten

diese Normen in ihrem gemeinsamen Komitee JTC 21. Als erste wurde im Juli 2026 die EN 18286 zum Qualitätsmanagement für Zwecke der KI-Verordnung angenommen. Die Entwürfe zum Risikomanagement (prEN 18228), zur Qualität von Datensätzen (prEN 18284), zur Cybersicherheit (prEN 18282) und zu einem Rahmen für vertrauenswürdige KI mit Protokollierung, Transparenz und menschlicher Aufsicht (prEN 18229) sind noch im Entwurfsstadium.[59] Im Amtsblatt zitiert ist zum Redaktionsschluss keine dieser Normen, und ihre Verzögerung war einer der Gründe für die Verschiebung der Hochrisiko-Fristen. Solange die Zitierung fehlt, müssen Anbieter ihre Konformität auf anderem Weg belegen und greifen auf internationale Normen zurück: ISO/IEC 42001 für KI-Managementsysteme, ISO/IEC 23894 für das KI-Risikomanagement und ISO/IEC 42005 für die Folgenabschätzung von KI-Systemen.[66] Diese Normen begründen keine automatische Konformitätsvermutung unter dem AI Act. Sie sind aber der nächstgelegene strukturierte Weg, und ISO/IEC 42001 ist zertifizierbar. Für die Architekturarbeit heißt das: Ein KI-Managementsystem, das entlang dieser Normen aufgebaut wird, legt das Fundament, auf dem der spätere formale Nachweis stehen wird. Diese Arbeit behält ihren Wert, auch solange die Zitierung der harmonisierten Normen aussteht.

Der AI Act als Architekturprinzip

Bleibt die Frage nach der Haltung, mit der ein Unternehmen dem AI Act begegnet. Zwei Reaktionen sind im Markt verbreitet, und beide führen in die Irre. Die erste behandelt den Aufschub der Fristen als Entwarnung und vertagt die Arbeit. Die zweite behandelt den AI Act als Compliance-Projekt, das eine Rechtsabteilung nach Fertigstellung der Technik abarbeitet. Die erste verkennt, dass die verschobene Frist die Arbeit nicht kleiner macht. Die zweite verkennt, dass sich die Anforderungen technisch gar nicht nachreichen lassen.

Die Pflichten des AI Act lesen sich, Punkt für Punkt, wie die Architekturprinzipien aus Kapitel 2. Das Risikomanagement über den Lebenszyklus setzt Beobachtbarkeit voraus. Datengovernance steht und fällt mit einer sauberen Datenschicht. Hinter der Pflicht zur technischen Dokumentation steckt dieselbe Versionierung und Nachverfolgbarkeit, die ein gut gebautes System ohnehin mitbringt, und die menschliche Aufsicht gelingt nur mit Schnittstellen, die von Beginn an dafür entworfen sind. Robustheit schließlich ist Sicherheit von Anfang an. Das ist kein Zufall. Der Gesetzgeber hat aufgeschrieben, was eine produktive, vertrauenswürdige KI-Architektur ohnehin braucht. Ein Unternehmen, das diese Prinzipien aus eigenem Interesse befolgt, hat den größten Teil der Compliance-Arbeit erledigt, bevor es das Gesetz aufschlägt.

Damit ist der Bogen dieses Abschnitts geschlossen, und der Übergang zum nächsten ist vorgezeichnet. Der AI Act sagt, was ein KI-System können muss. Er sagt nicht, wer im Unternehmen dafür sorgt, dass es das tut. Diese Frage, die Frage nach Gremien, Entscheidungspunkten und Verantwortlichkeiten, behandelt der folgende Abschnitt.

6.2 Das Governance-Rahmenwerk im Unternehmen

Der vorige Abschnitt hat beschrieben, was ein KI-System können muss, damit es dem AI Act genügt. Eine Frage hat er bewusst offengelassen: Wer im Unternehmen sorgt dafür, dass es das tut? Das Gesetz nennt Pflichten, es nennt keine Organigramme. Es verlangt eine technische Dokumentation, sagt aber nicht, wer sie freigibt. Es verlangt menschliche Aufsicht, sagt aber nicht, wer aufsichtsführend ist und an wen diese Person berichtet. Diese Lücke füllt kein Paragraph. Sie füllt das interne Governance-Rahmenwerk, und sie zu füllen ist eine Architekturaufgabe wie die technischen Pflichten auch, weil eine Verantwortung, die im System nicht verankert ist, im Ernstfall niemand trägt.

Governance hat dabei zwei Gesichter, die sich leicht verwechseln lassen. Das eine ist die Aufbauorganisation: Gremien, Rollen, Berichtswege. Das andere ist die Ablauforganisation: Entscheidungspunkte über den Lebenszyklus eines Vorhabens, von der Idee bis zur Abschaltung. Beide gehören zusammen, und beide müssen so geschnitten sein, dass sie zur Größe des Unternehmens passen. Eine Governance, die für einen Konzern entworfen wurde, erstickt einen Mittelständler an Sitzungen. Eine Governance, die für einen Mittelständler reicht, lässt einen Konzern im Blindflug. Dieser Abschnitt beschreibt beide Gesichter und ihre Abstufung nach Unternehmensgröße, bevor er sich zwei Spezialfällen zuwendet, die in regulierten Branchen über die allgemeine Praxis hinausgehen: dem Management von Modellrisiken und der technischen Verankerung von Fairness.

Die Aufbauorganisation: drei Funktionen in so wenigen Gremien wie möglich

In der Literatur zur KI-Governance kehren drei Funktionen wieder. Eine **strategische Steuerung**, die über Prioritäten und Mittel entscheidet, oft als KI-Lenkungsausschuss oder AI Steering Committee bezeichnet. Ein **Kompetenzzentrum**, das Methoden, Standards und wiederverwendbare Bausteine bereitstellt, das Center of Excellence. Und eine **ethisch-rechtliche Prüfinstanz**, die heikle Vorhaben gegen Werte und Recht hält, das Ethics Board oder Review Board. Die internationale Norm ISO/IEC 42001 für KI-Managementsysteme ordnet diese Funktionen einem Grundsatz unter, der aus dem Qualitätsmanagement vertraut ist: Die Verantwortung beginnt oben. Die Leitungsebene legt die KI-Politik fest, weist Rollen und Entscheidungsrechte zu und verankert die KI-Steuerung in den strategischen Zielen des Unternehmens.[67]

Wichtig ist die Unterscheidung zwischen Funktion und Gremium. Drei Funktionen lassen sich auch in einem einzigen Sitzungskörper ausüben. In einem mittelständischen Unternehmen kann ein einziger Kreis aus IT-Leitung, einem Fachbereichsvertreter und der Geschäftsführung alle drei Funktionen ausüben, sofern klar ist, in welcher Rolle er gerade tagt. Erst mit der Zahl der Vorhaben und der Höhe der Risiken lohnt sich die Trennung in eigene Gremien. Ein Konzern mit fünfzig parallelen KI-Initiativen über mehrere Geschäftsbereiche kommt um ein dediziertes Lenkungsgremium und ein eigenständiges Prüfgremium nicht herum, weil die Last sonst an einer einzigen Stelle hängt und dort zum Engpass wird. Die Faustregel ist schlicht: So viel Gremium wie nötig, so wenig wie möglich. Jede zusätzliche Instanz kostet Geschwindigkeit. Und Geschwindigkeit ist im KI-Umfeld die Bedingung dafür, dass die Architektur mit dem Markt Schritt hält.

Eine Rolle verdient besondere Aufmerksamkeit, weil ihr Fehlen der häufigste Grund ist, warum Governance auf dem Papier bleibt: die Eigentümerschaft. Für jedes produktive KI-System muss eine namentlich benannte Person verantwortlich sein. Diese Person beantwortet im Zweifel die Frage, warum das System eine bestimmte Entscheidung getroffen hat, und sie hat die Befugnis, es abzuschalten. Ohne diese Verankerung verteilt sich die Verantwortung auf alle und liegt damit bei niemandem. Kapitel 7 zählt diese fehlende Eigentümerschaft unter den häufigsten Stolperstellen, und der Grund steht hier: Eine Architektur kann technisch sauber sein und trotzdem unregierbar, solange offen ist, wer für sie geradesteht.

Die Ablauforganisation: Entscheidungspunkte über den Lebenszyklus

Die Aufbauorganisation klärt, wer entscheidet. Die Ablauforganisation klärt, wann. KI-Governance, die nur aus Gremien besteht, läuft Gefahr, entweder alles oder nichts zu prüfen. Alles zu prüfen lähmt; nichts zu prüfen ist keine Governance. Der Ausweg sind definierte Entscheidungspunkte, im Englischen Decision Gates, an denen ein Vorhaben eine Schwelle passiert und dafür eine Freigabe braucht.

Vier solcher Punkte haben sich als sinnvoll erwiesen. Der erste liegt am Anfang, bei der Idee: Hier wird die Risikoklasse nach AI Act bestimmt, von der Abschnitt 6.1 gezeigt hat, dass sie den gesamten weiteren Pflichtenumfang festlegt. Diese Einordnung gehört an den Beginn, weil sie über alles Weitere entscheidet. Der zweite

Punkt liegt vor dem Bau, bei der Konzeption: Hier wird geprüft, ob Datengrundlage, geplante Architektur und vorgesehene Aufsicht zur Risikoklasse passen. Der dritte liegt vor der Inbetriebnahme: Hier wird die Konformität bestätigt, die technische Dokumentation abgenommen, die Eigentümerschaft benannt. Der vierte liegt im Betrieb und wiederholt sich: Hier wird in festen Abständen geprüft, ob das System noch das tut, wofür es freigegeben wurde, oder ob Drift, veränderte Nutzung oder neue Vorschriften eine erneute Bewertung verlangen. Der letzte Punkt schließt den Kreis bis zur Abschaltung, denn auch das geordnete Außerbetriebnehmen eines Systems ist eine Governance-Entscheidung.

Diese vier Punkte sind die KI-Lesart des ADM-Zyklus, den Kapitel 2 beschrieben hat, übersetzt in Freigaben. Wo der ADM-Zyklus von Architekturvision über die Domänen bis zur Implementierungssteuerung führt, setzt die Lifecycle-Governance an denselben Übergängen die Tore. Die Verbindung ist beabsichtigt, denn eine Governance, die neben der Architekturmethodik herläuft, erzeugt doppelte Arbeit. Eine, die in sie eingebaut ist, erzeugt keine.

Die Wirkung dieser Tore hängt an einer Gestaltungsfrage, die über Erfolg oder Misserfolg des ganzen Rahmenwerks entscheidet. Ein Decision Gate kann absichern oder bremsen, und der Unterschied liegt im Zuschnitt. Ein Tor, das jedes Vorhaben gleich behandelt, bremst die kleinen, ohne die großen besser abzusichern. Ein Tor, das die Prüftiefe an der Risikoklasse staffelt, lässt das minimale Risiko fast ungehindert passieren und konzentriert die Aufmerksamkeit dort, wo sie zählt, beim hohen Risiko. Governance wird zum Bremsklotz, wenn sie undifferenziert prüft. Sie wird zur Absicherung, wenn sie ihre Strenge nach dem Risiko bemisst, das vor ihr liegt. Damit nimmt dieser Abschnitt die Haltung wieder auf, mit der das Kapitel eröffnet wurde.

Modellrisiko-Management: der regulierte Sonderfall

Für die meisten Unternehmen reicht das bisher Beschriebene. Für Banken und Versicherer reicht es nicht, und der Grund liegt in der Aufsicht. Der Einsatz von Modellen für Kreditentscheidungen, Risikoklassifizierung oder Bewertung steht seit jeher unter einer Modellaufsicht, die es lange vor der ersten KI-Anwendung gab. Diese Unternehmen bringen darum etwas mit, das andere Branchen erst aufbauen müssen: eine etablierte Disziplin der Modellvalidierung.

Diese Disziplin wird gerade auf KI ausgeweitet, und das Tempo hat zugenommen. Die BaFin hat am 30. Juni 2026 die neunte MaRisk-Novelle als Rundschreiben 06/2026 (BA) veröffentlicht. Ihr Modul AT 4.3.4 zur Verwendung von Modellen gilt ausdrücklich auch für automatisierte Modelle, technologiegestützte Innovation und Künstliche Intelligenz.[68] Es verlangt, Angemessenheit und Eignung eines Modells vor dem Einsatz zu bewerten und regelmäßig zu überprüfen, die Qualität der zugrundeliegenden Daten zu sichern, die Verwendung der Modellergebnisse einschließlich Überschreibungen zu regeln, die Modelle regelmäßig zu validieren und, als für KI heikelste Anforderung, auf hinreichende Erklärbarkeit zu achten: Ein Modell gilt als erklärbar, wenn Wirkungszusammenhänge zwischen Eingangs- und Ausgangsgrößen aufgezeigt werden können.[68] Das trifft neuronale Netze und große Sprachmodelle dort, wo sie am undurchsichtigsten sind. Unternehmen, die solche Modelle für risikorelevante Entscheidungen nutzen, müssen künftig zeigen, dass sie deren Wirkungsweise verstehen, etwa über etablierte Verfahren der Erklärbarkeit wie SHAP-Werte oder LIME.

Für die Architektur folgt daraus eine doppelte Botschaft. Die erste ist eine Warnung an die regulierten Unternehmen: Die Erklärbarkeitsanforderung schließt bestimmte Modellklassen für bestimmte Einsatzzwecke faktisch aus oder zwingt zu einem Erklärungsapparat, der mitgebaut werden muss. Diese Entscheidung fällt in der Architektur, beim Zuschnitt der Modellschicht, und sie fällt zu spät, wenn das Modell bereits produktiv ist. Die zweite Botschaft richtet sich an alle anderen: Das Modellrisiko-Management der Banken ist ein Vorgriff auf das, was der AI Act über Artikel 9 und 15 für Hochrisiko-Systeme ohnehin verlangt. Aus einer regulierten Bran-

che kommt man mit dem Vokabular bereits an. Den übrigen Unternehmen bietet der Bankenrahmen eine ausgearbeitete Vorlage, an der sich die eigene Modellvalidierung orientieren kann, lange bevor die harmonisierten Normen zum AI Act fertig sind.

Fairness als technische Komponente

Das letzte Element des Rahmenwerks ist das, bei dem die Versuchung am größten ist, es bei einer Absichtserklärung zu belassen. Fast jede KI-Richtlinie bekennt sich zu Fairness. Sehr viel seltener steht im System etwas, das Fairness prüfbar macht. Genau dort liegt der architektonische Punkt: Verzerrung wird dort kontrolliert, wo sie messbar ist, im Code und in den Daten. Eine Leitlinie kann sie nur bekräftigen.

Der AI Act verlangt in Artikel 10, dass Trainings-, Validierungs- und Testdaten von Hochrisiko-Systemen auf mögliche Verzerrungen untersucht und diese gemindert werden, wie Abschnitt 6.1 gezeigt hat. Aus einer Pflicht auf Papier wird eine Architekturkomponente erst, wenn drei Dinge im System verankert sind. Erstens definierte Fairness-Metriken, die zur Anwendung passen. Fairness ist dabei eine Familie von Maßen, die sich im Einzelfall sogar widersprechen können. Zweitens eine Prüfung dieser Metriken über geschützte Gruppen hinweg, eingebaut in die Pipeline, die ein Modell vor der Freigabe durchläuft. Drittens eine fortlaufende Beobachtung im Betrieb, weil sich Verzerrung mit den Eingangsdaten verschieben kann, lange nachdem die Freigabe erteilt wurde.

Das verbindet zwei der vorigen Kapitel. Das Monitoring, das Kapitel 5 als Betriebshygiene beschrieben hat, ist zugleich der Ort, an dem Fairness im Betrieb überwacht wird. Die Beobachtbarkeit, die Kapitel 2 als Architekturprinzip eingeführt hat, ist die Voraussetzung dafür, dass eine Verzerrung überhaupt sichtbar wird. Fairness ist damit eine Anforderung, die dieselben Komponenten erfüllen, die das System für den Produktivbetrieb ohnehin mitbringt.

Damit steht das interne Rahmenwerk. Es klärt, wer entscheidet, wann geprüft wird und wie Modellrisiko und Fairness im System verankert werden. Eine Gefahr hat es bisher ausgespart, obwohl sie jede dieser Strukturen unterlaufen kann: den gezielten Angriff. KI bringt Angriffsflächen mit, die klassische Anwendungssicherheit so nicht kannte, und der Gesetzgeber hat parallel über NIS-2 die Anforderungen an die Cybersicherheit verschärft. Beidem widmet sich der folgende Abschnitt.

6.3 Sicherheit in der KI-Ära: neue Angriffsflächen und NIS-2

Der vorige Abschnitt hat das interne Rahmenwerk beschrieben, mit dem ein Unternehmen seine KI steuert. Eine Gefahr lag dabei außerhalb des Bildes, weil sie von außen gegen das Unternehmen gerichtet ist: der Angriff. KI verschiebt die Angriffsfläche eines Unternehmens, und sie tut es auf eine Weise, die klassische Anwendungssicherheit nur teilweise abdeckt. Parallel hat der Gesetzgeber die Anforderungen an die Cybersicherheit verschärft, in Deutschland seit Dezember 2025 mit unmittelbarer Geltung. Dieser Abschnitt verbindet beides: die neuen Angriffsklassen, die architektonischen Antworten darauf und die regulatorische Pflichtenlage, in der diese Antworten gegeben werden müssen.

Ein Hinweis zur Einordnung vorab. Anwendungssicherheit im klassischen Sinn bleibt gültig. Eine KI-Anwendung ist eine Anwendung, sie braucht Zugriffskontrolle, Verschlüsselung und gepatchte Abhängigkeiten wie jede andere. Was hinzukommt, sind Angriffsflächen, die es ohne das Modell nicht gäbe. Sie stehen hier im Vordergrund, weil sie das Neue sind. Sie legen sich über die alten Sorgen.

Die neuen Angriffsklassen

Die NIST-Taxonomie für Angriffe auf maschinelles Lernen, von Kapitel 2 bereits als Bezugspunkt für das Prinzip Security by Design genannt, ordnet die KI-spezifischen Bedrohungen in wiederkehrende Klassen.[69] Vier davon prägen die produktive Praxis am stärksten, und sie greifen an verschiedenen Stellen des AI Architecture Stack an, den Kapitel 2 beschrieben hat.

Die bekannteste ist die **Prompt Injection**. Sie nutzt einen Konstruktionszug der Sprachmodelle aus: Anweisung und Inhalt laufen durch denselben Kanal, und das Modell kann beide oft nicht sauber trennen. Eine eingeschleuste Anweisung, versteckt in einem Dokument, einer Webseite oder einer Nutzereingabe, kann das Modell dazu bringen, seine eigentliche Aufgabe zu verlassen. Das OWASP-Projekt für die Sicherheit generativer KI führt Prompt Injection auch in seiner Ausgabe 2026 auf Platz eins der Risiken für LLM-Anwendungen; *Excessive Agency*, die übermäßige Handlungsmacht eines Modells, ist dort auf Platz drei gestiegen.[70] Besonders heikel wird die Angriffsklasse in agentischen Systemen, die Kapitel 4 behandelt hat: Sobald ein Modell Werkzeuge bedient, Code ausführt oder Datenbanken abfragt, vergrößert eine einzige erfolgreiche Injection den möglichen Schaden erheblich, weil die Anweisung sich in eine Handlung übersetzt.

Die zweite Klasse ist das **Data Poisoning**, die gezielte Manipulation der Daten, auf denen ein Modell trainiert oder aus denen es zur Laufzeit abruft. Vergiftete Trainingsdaten verschieben das Verhalten des Modells dauerhaft. Eine manipulierte Wissensbasis verfälscht die Antworten einer RAG-Anwendung, ohne das Modell selbst anzurühren. Diese Angriffsklasse setzt an der Datenschicht an, dem Fundament, dem Kapitel 3 ein eigenes Kapitel gewidmet hat, und sie ist der Grund, warum die Herkunft und Integrität von Daten eine Sicherheitsfrage ist und keine reine Qualitätsfrage.

Die dritte Klasse zielt auf die Vertraulichkeit. Unter Begriffen wie **Model Inversion** und Membership Inference werden Angriffe zusammengefasst, die aus den Antworten eines Modells auf seine Trainingsdaten zurückschließen. Bei einem Modell, das auf personenbezogenen oder geschäftskritischen Daten trainiert wurde, kann das ein Datenschutzvorfall sein, der ohne klassischen Datenabfluss entsteht. Hier berührt die Sicherheit unmittelbar den Datenschutz, dem sich Abschnitt 6.4 zuwendet.

Die vierte Klasse ist der **Diebstahl des Modells selbst**, durch unbefugten Zugriff auf die Gewichte oder durch systematisches Abfragen, mit dem sich ein Modell näherungsweise nachbauen lässt. Für ein Unternehmen, das in ein eigenes oder feinjustiertes Modell investiert hat, steht hier geistiges Eigentum auf dem Spiel. Der Angriff setzt an der Modell- und Plattformschicht an, an den Endpunkten, über die ein Modell erreichbar ist.

Diese vier Klassen haben eine Gemeinsamkeit, die für die Architektur entscheidend ist. Sie lassen sich nicht an einer einzigen Stelle abwehren, weil sie an verschiedenen Schichten ansetzen. Eine Firewall vor dem Modell hilft gegen den Diebstahl der Gewichte und nichts gegen vergiftete Trainingsdaten. Sicherheit in KI-Systemen ist darum eine Eigenschaft des gesamten Stack.

Die architektonische Antwort

Aus den vier Angriffsklassen folgt eine Sicherheitsarchitektur, die an denselben Schichten ansetzt, an denen die Angriffe ansetzen. Vier Bauelemente bilden den Kern.

An der Anwendungs- und Orchestrierungsschicht steht die **Behandlung von Ein- und Ausgaben**. Eingaben werden geprüft, bevor sie das Modell erreichen, Ausgaben, bevor sie eine Handlung auslösen. Gegen Prompt Injection gibt es keine vollständige Lösung, solange Anweisung und Inhalt denselben Kanal teilen. Es gibt aber wirksame Dämpfung: die Trennung von System- und Nutzeranweisungen, die Begrenzung der Rechte, die ein

Modell über Werkzeuge ausüben darf, und eine Bestätigungsschwelle vor folgenreichen Aktionen. Das letzte Element ist die menschliche Aufsicht aus Artikel 14 des AI Act, von Abschnitt 6.1 als Pflicht beschrieben, hier in ihrer sicherheitstechnischen Funktion. Für Systeme mit Agenten liefert die OWASP Top 10 für agentische Anwendungen die passende Prüfliste (Kapitel 4.2).

An der Datenschicht steht die **Absicherung des Abrufpfads**. Die Wissensbasis einer RAG-Anwendung wird wie ein produktives System behandelt, mit Zugriffskontrolle, Herkunftsnachweis und Integritätsprüfung der aufgenommenen Inhalte. Eine Quelle, die jeder ungeprüft befüllen kann, ist eine offene Tür für Data Poisoning.

An der Modellschicht steht die **Härtung der Endpunkte**. Modell-Endpunkte bekommen Authentifizierung, Ratenbegrenzung und Anomalieerkennung, die ungewöhnliche Abfragemuster bemerkt, bevor aus systematischem Abfragen ein nachgebautes Modell wird. Die Gewichte eigener Modelle werden wie ein Kronjuwel verwahrt, mit Verschlüsselung und engem Zugriff.

Quer durch alle Schichten steht die **Beobachtbarkeit**, die Kapitel 2 als Architekturprinzip eingeführt hat. Ein Angriff, der nicht sichtbar wird, wird nicht abgewehrt. Telemetrie, Protokollierung und Alarmierung sind damit auch eine Sicherheitsfrage, über die Betriebsfrage hinaus, und sie sind zugleich die Grundlage der Nachweispflichten, die der folgende Teil behandelt. An dieser Stelle treffen sich Security by Design aus Kapitel 2, das Modell-Monitoring aus Kapitel 5 und die Protokollierungspflicht aus Artikel 12 des AI Act in einer einzigen Architekturentscheidung.

NIS-2: aus der Aussicht ist Pflicht geworden

Während dieses Whitepaper entstand, hat sich die regulatorische Lage in einem Punkt entscheidend geändert. Die NIS-2-Richtlinie der EU, lange als kommende Verschärfung der Cybersicherheitspflichten behandelt, ist in Deutschland inzwischen geltendes Recht. Das Gesetz zur Umsetzung der NIS-2-Richtlinie, kurz NIS-2-Umsetzungsgesetz, wurde am 5. Dezember 2025 im Bundesgesetzblatt verkündet und ist am Tag darauf, dem 6. Dezember 2025, in Kraft getreten, ohne Übergangsfrist.[71] Aus einer Aussicht, die man in der Architekturplanung berücksichtigen sollte, ist eine Pflicht geworden, die seit dem 6. Dezember 2025 gilt.

Die Umsetzung erfolgt über eine Novellierung des BSI-Gesetzes und erweitert dessen Anwendungsbereich erheblich. Waren bisher etwa 4.500 Organisationen reguliert, vor allem Betreiber kritischer Infrastrukturen, fallen künftig rund 29.500 Einrichtungen unter die Aufsicht des BSI.[71] Maßgeblich sind die Zugehörigkeit zu einem der gesetzlich genannten Sektoren und das Überschreiten von Schwellenwerten bei Beschäftigtenzahl, Umsatz und Bilanzsumme. Das Gesetz unterscheidet zwei Kategorien, „wichtige" und „besonders wichtige Einrichtungen", an die unterschiedlich strenge Pflichten anknüpfen. Für die Geschäftsführung vieler mittelständischer Unternehmen ist die erste Aufgabe darum eine Prüfung der eigenen Betroffenheit, denn die Einordnung erfolgt in Selbstverantwortung, ohne behördlichen Bescheid.

Drei Kernpflichten treffen die betroffenen Einrichtungen. Sie müssen sich beim BSI registrieren, wofür die erste Frist bereits am 6. März 2026 ablief. Sie müssen erhebliche Sicherheitsvorfälle melden, über ein dafür eingerichtetes Portal des BSI. Und sie müssen Risikomanagementmaßnahmen für ihre IT-Sicherheit umsetzen und diese Umsetzung dokumentieren.[71] Die letzte Pflicht ist die mit der größten architektonischen Tragweite, weil sie fortlaufend wirkt und sich nicht mit einer einmaligen Maßnahme erledigen lässt.

Wie NIS-2 und der AI Act zusammenwirken

Damit liegen zwei Regelwerke nebeneinander, die denselben Gegenstand aus verschiedenen Richtungen berühren. Der AI Act fordert in Artikel 15 für Hochrisiko-Systeme Genauigkeit, Robustheit und Cybersicherheit, wie Abschnitt 6.1 gezeigt hat. NIS-2 fordert ein Risikomanagement für die Cybersicherheit der gesamten betroffenen Einrichtung. Das eine zielt auf das einzelne KI-System, das andere auf das Unternehmen, in dem es läuft. Ein Unternehmen, das beidem genügen muss, betreibt seine KI-Sicherheit zweckmäßigerweise als eine Architektur, die beide Anforderungen zugleich bedient.

Hier liegt der praktische Vorteil einer gemeinsamen Architektur. Die Bauelemente der KI-Sicherheit, von der Eingabeprüfung über die Absicherung des Abrufpfads bis zur durchgängigen Beobachtbarkeit, sind zugleich Bausteine des Risikomanagements, das NIS-2 verlangt, und Belege für die Cybersicherheit, die der AI Act fordert. Sie einmal sauber zu bauen, erfüllt beide Pflichten. Sie zweimal nachzurüsten, kostet doppelt.

Eine letzte Regelung fehlt noch im Bild, die älteste von allen und die, mit der die meisten Unternehmen am längsten Erfahrung haben. Der Datenschutz war schon Thema, als von Künstlicher Intelligenz im Unternehmen kaum die Rede war. Wie er sich mit KI verbindet und welche architektonischen Muster dabei tragen, behandelt der abschließende Abschnitt dieses Kapitels.

6.4 Datenschutz und DSGVO im KI-Kontext

Der Datenschutz ist die älteste der vier Regelungen dieses Kapitels und die, mit der die meisten Unternehmen seit Jahren leben. Die Datenschutz-Grundverordnung gilt seit Mai 2018, ihre Grundbegriffe sind eingespielt, ihre Verfahren etabliert. Gerade diese Vertrautheit birgt eine Gefahr: KI stellt den Datenschutz vor Fragen, auf die die eingeübte Praxis nicht immer eine fertige Antwort hat. Dieser Abschnitt beschreibt die architektonischen Muster, mit denen sich beides verbinden lässt, und schließt das Kapitel mit einer Zusammenführung der vier Regelwerke.

Privacy by Design, vom Prinzip zur Pipeline

Kapitel 2 hat Privacy by Design als eines der fünf Architekturprinzipien eingeführt und an Artikel 25 der DSGVO festgemacht, der den Datenschutz durch Technikgestaltung seit 2018 zur Pflicht erhebt. Was dort als Prinzip stand, wird hier konkret. Datenschutz in einer ML-Pipeline ist eine Folge von Entscheidungen an bestimmten Stellen des Datenpfads, und jede dieser Entscheidungen fällt früh oder gar nicht.

Drei Stellen sind besonders wirksam. Die erste ist die Datenminimierung bei der Aufbereitung: Was ein Modell für seinen Zweck nicht braucht, sollte den Trainingsdatensatz erst gar nicht erreichen. Die zweite ist die Pseudonymisierung in der Trainingspipeline, an der Schwelle, an der Rohdaten zu Trainingsdaten werden. Die dritte ist die Zweckbindung, festgehalten pro Anwendung, damit nachvollziehbar bleibt, wofür ein Datenbestand verarbeitet werden darf und wofür eben nicht. Diese drei Maßnahmen lassen sich an eine bestehende Pipeline nur schwer anflanschen, weil sie tief in deren Aufbau greifen. In eine neu entworfene Pipeline gehören sie als selbstverständliche Bestandteile. Damit gilt für den Datenschutz, was Abschnitt 6.1 für den AI Act gezeigt hat: Die nachträgliche Ergänzung ist teuer, die eingebaute Lösung kostet kaum mehr als die Entscheidung, sie vorzusehen.

Anonymisierung und synthetische Daten: ein Muster mit Grenzen

Ein verbreiteter Reflex lautet, das Datenschutzproblem durch Anonymisierung aufzulösen. Sind die Daten anonym, greift die DSGVO nicht. So weit die Theorie. Die Praxis ist heikler, und sie ist seit Ende 2024 deutlich klarer umrissen. Der Europäische Datenschutzausschuss hat in seiner Stellungnahme 28/2024 dargelegt, dass die Anonymität eines KI-Modells im Einzelfall zu prüfen ist und nicht pauschal angenommen werden kann.[72] Maßgeblich ist, wie wahrscheinlich es ist, dass sich mit angemessenen Mitteln personenbezogene Daten aus dem Modell gewinnen lassen, etwa durch gezielte Abfragen. Anonym ist ein Modell erst, wenn der Verantwortliche belegen kann, dass diese Wahrscheinlichkeit unerheblich ist.

Das führt unmittelbar zurück zu Abschnitt 6.3. Die dort beschriebenen Angriffe auf die Vertraulichkeit, Model Inversion und Membership Inference, sind genau die Mittel, mit denen sich Daten aus einem vermeintlich anonymen Modell extrahieren lassen. Datenschutz und Sicherheit sind hier dieselbe Frage aus zwei Blickwinkeln: Ein Modell ist nur so anonym, wie es gegen Extraktion gehärtet ist. Die Härtung der Endpunkte aus 6.3 ist damit zugleich eine Maßnahme des Datenschutzes.

Synthetische Daten gelten als der elegante Ausweg. Statt mit echten Personendaten wird mit künstlich erzeugten Daten trainiert, die deren statistische Eigenschaften nachbilden. Das Muster trägt in vielen Fällen, und es hat seinen festen Platz in der Datenschicht aus Kapitel 3. Seine Grenzen sind allerdings real, und sie bilden einen schmalen Korridor. Liegen die erzeugten Daten zu nah am Original, lassen sie Rückschlüsse auf reale Personen zu und sind dann nicht mehr anonym. Liegen sie zu weit entfernt, verlieren sie den Nutzen, der sie überhaupt erst interessant machte. Die Datenschutzprüfung bleibt damit nötig; sie verschiebt sich auf die Frage, wie synthetisch die synthetischen Daten wirklich sind.

Datensouveränität und das Recht auf Auskunft

Zwei weitere Anforderungen prägen die Architektur. Die erste ist die Datensouveränität, an die Abschnitt 4.4 mit Blick auf europäische Cloud-Alternativen herangeführt hat. Im Datenschutzkontext bekommt sie eine zusätzliche Schärfe, weil der Ort der Verarbeitung über die anwendbare Rechtslage mitentscheidet. Wo personenbezogene Daten durch ein Modell laufen, das auf Infrastruktur außerhalb des europäischen Rechtsraums betrieben wird, stellt sich die Frage des Drittlandtransfers, und sie stellt sich am besten beim Entwurf der Infrastrukturschicht, lange bevor die ersten Daten fließen.

Die zweite Anforderung ist die Auskunftsfähigkeit. Die DSGVO gibt betroffenen Personen das Recht zu erfahren, welche Daten über sie verarbeitet werden, und im Fall automatisierter Entscheidungen nach Artikel 22 auch ein Recht auf Information über die Logik dahinter. Für ein KI-System ist das eine Architekturanforderung mit Folgen. Ein System muss in der Lage sein, zu einer einzelnen Person Auskunft zu geben, was ein Datenmodell voraussetzt, in dem sich diese Auskunft überhaupt zusammentragen lässt. Und es muss, wo es Entscheidungen über Menschen trifft, erklärbar bleiben. Hier trifft sich die Datenschutzanforderung mit der Erklärbarkeit, die Abschnitt 6.2 für regulierte Modelle aus der MaRisk-Welt beschrieben hat, und mit der Transparenzpflicht aus Artikel 13 des AI Act. Drei Regelwerke verlangen, aus je eigenem Grund, dasselbe: dass ein KI-System nachvollziehbar bleibt.

Vier Regelwerke, eine Architekturschicht

Damit liegen die vier Regelungen vollständig auf dem Tisch, und ihr Zusammenhang lässt sich nun benennen. Der AI Act regelt das KI-System nach seinem Risiko. Eine Ebene daneben legt das interne Governance-Rahmenwerk fest, wer im Unternehmen dafür geradesteht. NIS-2 nimmt die Cybersicherheit der gesamten Einrichtung in den Blick, in der das System läuft, während die DSGVO beim Umgang mit den personenbezogenen Daten ansetzt, die hindurchfließen. Vier Regelwerke, vier Blickwinkel, ein Gegenstand.

Das auffällige Muster dieses Kapitels war, wie oft diese Regelwerke an derselben Architekturentscheidung ansetzen. Eine einzige Protokollierung erfüllt eine Pflicht des AI Act, liefert das Material für die Sicherheitsüberwachung nach NIS-2 und stützt die Auskunftsfähigkeit der DSGVO. Bei der Erklärbarkeit treffen sich die Transparenzpflicht des AI Act, die Modellvalidierung der MaRisk und das Auskunftsrecht der DSGVO. Und die Härtung gegen Extraktion zählt als Sicherheits- und als Datenschutzmaßnahme zugleich. Vier Regelwerke verlangen darum eine Architektur, die ihre gemeinsamen Anforderungen an einer Stelle erfüllt.

Das ist die Botschaft, mit der das Kapitel begann, und sie trägt nun das volle Gewicht der vier Abschnitte. Governance ist eine Schicht der Architektur, die Kapitel 2 als Querschnitt durch den AI Architecture Stack eingeführt hat. Sie wird eingebaut, solange das System entsteht, und ist später nur mit Mühe und doppelten Kosten nachzurüsten. Ein Unternehmen, das die fünf Architekturprinzipien aus Kapitel 2 ernst nimmt, hat den größten Teil dessen erfüllt, was diese vier Regelwerke fordern, bevor es das erste Gesetz aufschlägt. Genau darin liegt der Unterschied zwischen Governance als Bremsklotz und Governance als Fundament.

Damit ist das Bild der Governance vollständig. Die Architektur ist beschrieben, die Muster sind geordnet, die Regeln sind geklärt. Was bis hierher fehlt, ist die Umsetzung, abgestuft nach Größe und Reife des Unternehmens. Ihr widmet sich das folgende Kapitel.

KAPITEL 7

7 Der Weg: Roadmap für deutsche Unternehmen

Bis hierher haben wir das Problem beschrieben und die Muster geordnet. Was fehlt, ist die Frage, die in jedem Vorstandssaal nach der zweiten Folie gestellt wird: *Und wo fangen wir an?*

Sie ist die schwerste der drei Fragen, weil sie die anderen beiden voraussetzt. Ohne Wissen darüber, wo ein Use Case wirtschaftlich trägt, entsteht die falsche Plattform. Ohne die Belegschaft läuft sie ohne Nutzer. Und ohne gewählte Architektur bleibt offen, wofür die Roadmap geschrieben wird. KI-Einführung ist eine Arbeit an drei Hebeln gleichzeitig: an der wirtschaftlichen Bewertung, an der organisatorischen Befähigung, an der technischen Architektur. Dieses Kapitel beschreibt den dritten Hebel. Es beschreibt ihn so, dass die beiden anderen im Blick bleiben.

Was folgt, ist ein Raster, in das sich Unternehmen einordnen können, eine Beschreibung dreier Wege je nach Größe und Reifegrad, eine Abfolge von vier Meilensteinen und eine Liste der Stolperstellen, an denen KI-Programme am häufigsten scheitern. Ein Unternehmen, das dieses Raster gegen die eigene Lage hält, gewinnt einen belastbaren Ausgangspunkt für seinen eigenen Plan.

7.1 Der RLC AI Architecture Readiness Check

Eine Selbstbewertung am Anfang ist eine Schutzmaßnahme. Ohne Standortbestimmung lässt sich nicht entscheiden, wohin der nächste Schritt führt. Eine erste Runde der folgenden Bewertung kann eine einzelne Person ausfüllen. Sie wird gründlicher, sobald mehr Beteiligte mitwirken und mehr Belege auf den Tisch kommen.

Fünf Dimensionen, eine Skala

Der Readiness Check fragt nach fünf Dimensionen: Daten als Fundament, Strategie und Wertbeitrag, Architektur und Muster, Betrieb und MLOps, Governance und Compliance. Für jede Dimension gibt es eine Skala von eins bis fünf. Die Zahlen sind hier weniger wichtig als die Beschreibungen dahinter.

Daten und Infrastruktur. Stufe eins bedeutet: Daten liegen in operativen Systemen verteilt, keine konsistente Sicht. Stufe fünf bedeutet: Eine Datenarchitektur mit klaren Domänen, dokumentierten Verträgen zwischen Systemen, einem Feature Store oder funktionalen Äquivalent, einer dokumentierten Cloud-Strategie. Ein Unternehmen mit funktionierendem Data Warehouse, aber ohne dokumentierte Verträge zwischen den Systemen, steht bei zwei oder drei.

Strategie und Wertbeitrag. Stufe eins: KI-Projekte starten aus Begeisterung oder Druck, ein Geschäftsfall wird, wenn überhaupt, im Nachhinein konstruiert. Stufe fünf: Jeder produktiv genutzte Use Case hat einen dokumentierten Business Case, eine zugeordnete Wertschöpfung, eine Erfolgsmessung und eine Stelle, an der das Ergebnis verantwortet wird. Eine schnelle Prüfung: Ein Unternehmen, das nennen kann, welcher KI-Use-Case im letzten Quartal welchen messbaren Beitrag geleistet hat, liegt näher an drei. Kann es das nicht, liegt es bei zwei. Die Bitkom-Erhebung legt nahe, dass Letzteres der häufigere Fall ist: 59 Prozent der Unternehmen mit KI im Einsatz schöpfen deren Potenzial nach eigener Einschätzung überhaupt nicht aus.[73]

Architektur und Muster. Stufe eins: KI-Anwendungen sind Inseln, jede Lösung bringt ihren eigenen Datenzugriff, ihre eigene Modellverwaltung, ihre eigene Oberfläche mit. Stufe fünf: Eine Plattform trägt mehrere Anwendungen, die Muster aus Kapitel 4 sind erkennbar umgesetzt, Wiederverwendung ist die Regel.

Betrieb und MLOps. Stufe eins: Modelle und Anwendungen werden händisch deployed, Monitoring ist anekdotisch. Stufe fünf: Automatisierte Pipelines, dokumentiertes Modell-Inventar, kontinuierliche Beobachtung von Qualität, Drift und Kosten. Diese Dimension folgt typischerweise mit Verzögerung. Ein Unternehmen, das hier vorprescht, ohne die anderen Dimensionen mitzuziehen, baut industrielle Betriebsfähigkeit für Anwendungen, die niemand braucht.

Governance und Compliance. Stufe eins: Es gibt keine dokumentierte Verantwortung für KI-Risiken, der EU AI Act ist als Stichwort bekannt, mehr nicht. Stufe fünf: Eine Stelle verantwortet KI-Governance organisatorisch, jedes produktive System ist klassifiziert, technische Dokumentation und Risikoanalysen sind Teil des Lebenszyklus, regulatorische Anforderungen sind in der Architektur abgebildet. Eine verlässliche Erhebung zur deutschen Mittellage gibt es für diese Dimension nicht. Ein Unternehmen, das drei ehrlich erreicht, ist nach unserer Einschätzung gut aufgestellt.

Wie man die Ergebnisse liest

Die Summe der fünf Werte ist nützlich, aber die Verteilung ist entscheidend. Ein Unternehmen mit der Verteilung 4/2/4/3/1 hat ein Compliance-Problem; seine Datenarchitektur ist in Ordnung. Ein Unternehmen mit 2/4/2/2/2 weiß, was es will, hat aber nichts gebaut. Beide brauchen unterschiedliche nächste Schritte, obwohl ihre Gesamtpunktzahl ähnlich ist. Als bloßer Score gelesen, verschenkt der Check seinen diagnostischen Wert.

Vier Verteilungen lassen sich als Typen unterscheiden. Erstens das **datenstarke Unternehmen mit Use-Case-Lücke**: starke Datenarchitektur, schwache Verbindung zu Geschäftsfällen. Zweitens das **pilotgetriebene Unternehmen**: viele Use Cases auf niedrigem Reifegrad, ohne Plattform, ohne Governance, oft mit großer Begeisterung. Drittens das **Compliance-orientierte Unternehmen**, häufig in regulierten Branchen: hoher Anspruch an Governance, dafür Zurückhaltung im Aufbau. Viertens das **plattformlastige Unternehmen**, das eine Infrastruktur gebaut hat, die im Vergleich zur Use-Case-Nutzung überdimensioniert ist. Jedes dieser Profile braucht einen anderen ersten Schritt.

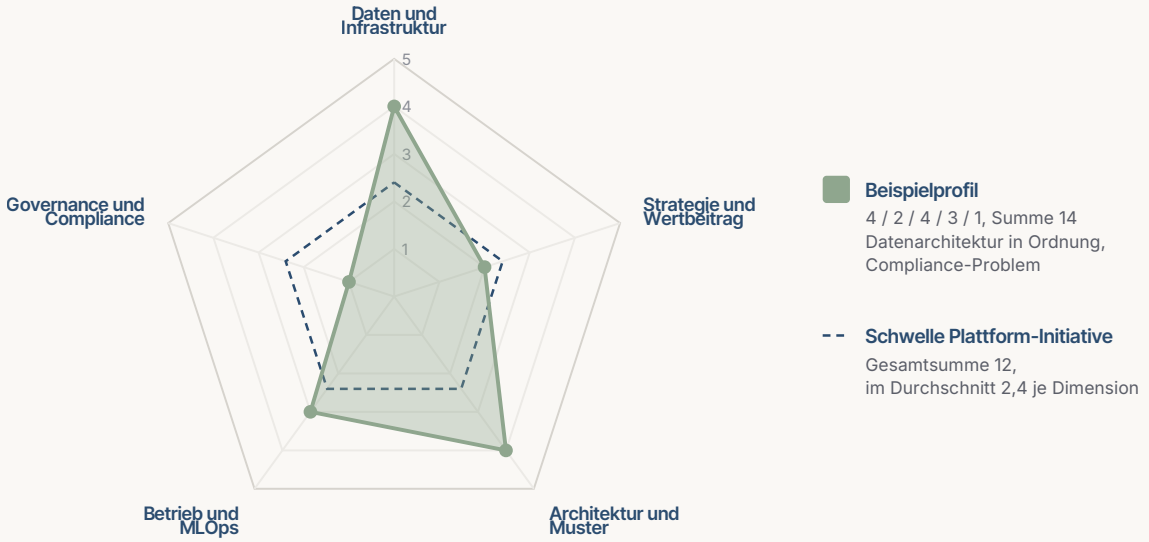


Abbildung 7.1 · Der RLC AI Architecture Readiness Check. Fünf Dimensionen auf einer Skala von eins bis fünf, mit einem Beispielprofil und der Schwelle für eine Plattform-Initiative.

Typische Reifegrade in deutschen Unternehmen

Belastbare Querschnittsdaten zur KI-Reife in Deutschland sind rar. Was es gibt, zeichnet ein konsistentes Bild: Adoption hoch, Reife niedrig. Die repräsentative Bitkom-Erhebung vom September 2026 zählt 57 Prozent der Unternehmen mit KI im Einsatz; von ihnen gibt kein einziges an, das Potenzial vollständig zu nutzen, und nur drei Prozent nutzen es nach eigener Einschätzung eher stark.[73] Die OECD zeigt für die EU, wie jung diese Bewegung ist: 2024 setzten 13,5 Prozent der Unternehmen ab zehn Beschäftigten KI ein, nach 8 Prozent im Jahr davor.[74] Wegen unterschiedlicher Abgrenzung und Unternehmensgröße sind die beiden Zahlen nicht direkt vergleichbar; beide zeigen eine schnell wachsende Nutzung bei geringer Reife.

Für den Readiness Check setzt dieser Leitfaden eine Schwelle: Ab einer Gesamtsumme von zwölf empfehlen wir eine KI-Plattform-Initiative. Die Schwelle ist eine Setzung, abgeleitet aus den Stufenbeschreibungen; gemessen ist sie nicht. Sie verlangt, dass die Dimensionen im Durchschnitt über die Stufe zwei hinausgekommen sind. Darunter ist die Reihenfolge der Arbeit eine andere: zuerst Datenarchitektur und Geschäftsfälle aufräumen, dann die Plattform.

7.2 Drei Architektur-Pfade nach Unternehmensgröße

Es gibt keine universelle KI-Architektur. Was für einen DAX-Konzern angemessen ist, überfordert ein Familienunternehmen mit dreihundert Mitarbeitenden. Was für den Mittelstand pragmatisch ist, unterfordert eine Versicherung mit zehntausend Mitarbeitenden, eigenem Rechenzentrum und regulatorischer Last. Die folgenden drei Pfade sind Idealtypen. Reale Unternehmen liegen oft dazwischen oder weisen Mischformen auf. Aber als Orientierung helfen sie, weil sie die typischen Festlegungen je nach Größe und Reifegrad sortieren.

Eine Vorbemerkung zu den Investitionsbandbreiten in diesem Abschnitt. Die Zahlen sind Richtwerte dieses Leitfadens. Sie beruhen auf Annahmen, die je Pfad genannt sind: Umfang von Software und Lizenzen, Aufbauleistung, externe Begleitung und, bei den Pfaden B und C, Cloud-Verbrauch. Sie gelten jeweils für den Weg bis zum Regelbetrieb, also bis zum Meilenstein M4 in Abschnitt 7.3. Sie verschieben sich, wenn das Unternehmen

eine ungewöhnlich gute Datenausgangslage mitbringt oder eine ungewöhnlich schwere. Eine seriöse Schätzung für den eigenen Fall verlangt eine eigene Aufwandsrechnung. Die Zahlen hier setzen eine Größenordnung, in der die Diskussion stattfinden kann.

Pfad A: KMU und Mittelstand bis etwa 500 Mitarbeitende

Für kleinere Unternehmen ist die wichtigste Architekturentscheidung die, möglichst wenige Architekturentscheidungen selbst treffen zu müssen. Das klingt paradox, ist aber die Konsequenz aus knappen IT-Teams und der Notwendigkeit, KI-Wertbeiträge schnell sichtbar zu machen.

Die typische Ausgangslage: Wenige Use Cases mit klarem Bezug zum Tagesgeschäft, IT-Team in der Größe einer Handvoll Personen, oft ohne dediziertes Data-Engineering. Eine eigene KI-Plattform zu bauen, ist hier in den allermeisten Fällen die falsche Antwort.

Die empfohlene Strategie ist **SaaS-first mit strukturierter Exit-Option**. Das bedeutet: Generative KI über die Unternehmensversionen etablierter Anbieter wie Microsoft 365 Copilot, Gemini in Google Workspace oder spezialisierte Branchen-SaaS. Eigene Datenflüsse über die offiziellen, vertraglich abgesicherten Anbindungen. RAG-Anwendungen, wenn sie nötig sind, über Managed Services aus der gewählten Cloud, ohne eigenes Infrastruktur-Setup.

Die Exit-Option besteht in zwei Disziplinen. Erstens in der Datenhaltung: Dokumente, Wissensbasen und Trainingsdaten liegen in offenen Formaten, außerhalb proprietärer Anbieter-Container. Zweitens in einer dokumentierten Liste der Funktionen, deren Wegfall das Geschäft beeinträchtigen würde. Diese Liste ist die ehrlichste Form, über Vendor Lock-in zu sprechen. Sie macht aus dem diffusen Bauchgefühl eine überprüfbare Größe.

Empfohlene Bausteine in dieser Größenordnung: Microsoft 365 Copilot oder ein vergleichbares Pendant aus dem Google- oder europäischen Lager als Produktivitätsschicht. Eine SaaS-Lösung für RAG auf interne Wissensbasen, etwa Azure AI Search mit einer schlanken Anwendungsschicht oder ein spezialisiertes Mittelstands-Produkt. Ein Identity-Provider, der über alle KI-Anwendungen hinweg konsistente Berechtigungen ermöglicht. Ein einfaches Monitoring der KI-Nutzung in Bezug auf Kosten und sensitive Inhalte.

Als Richtwert für den Weg bis zum Regelbetrieb setzen wir in dieser Größenklasse **150.000 bis 400.000 Euro** für Software-, Anbindungs- und Beratungsleistungen an. Hinzu kommen Lizenzkosten, die mit der Zahl der Nutzer skalieren. Microsoft 365 Copilot etwa kostet als Zusatzlizenz je nach Vertragsvariante 18,20 Euro (Copilot Business) oder 26,00 Euro je Nutzer und Monat, jeweils bei jährlicher Abrechnung, zuzüglich Mehrwertsteuer und zusätzlich zur Microsoft-365-Basislizenz (Listenpreise, Stand Oktober 2026).[75] Die Spanne setzt voraus, dass keine eigene KI-Plattform aufgebaut wird. Ein Unternehmen, das eine baut, ist nicht mehr auf Pfad A.

Pfad B: Großmittelstand zwischen 500 und 5.000 Mitarbeitenden

In dieser Größenklasse beginnt die Architekturfrage, ernsthaft Konsequenzen zu haben. Eigene Datenströme sind komplex genug, dass ihre Bewirtschaftung zur Disziplin wird. Use Cases häufen sich, sodass eine Plattform-Logik sinnvoll wird. Aber die Ressourcen reichen selten für eine vollständige eigene AI Factory.

Die empfohlene Strategie ist eine **hybride Architektur mit Cloud-Core**. Eine zentrale Cloud-Region trägt die Modell-Infrastruktur, die Pipelines und das Modell-Inventar. Datenhaltung und -aufbereitung sind dort konzentriert, wo sie ohnehin schon liegen, in der Regel in einer der großen Cloud-Datenplattformen oder in einer Kombination aus eigener Datenbank und Cloud-Lakehouse. Sensitive Lasten bleiben on-premise oder in einer dedizierten souveränen Umgebung, der Rest läuft in der Public Cloud.

Was hier neu hinzukommt, ist eine schlanke eigene **AI Platform**. Sie besteht aus einer zusammengeführten Sammlung von Diensten: ein Modell-Inventar, eine Pipeline-Orchestrierung, ein Feature Store, eine zentrale Beobachtungsschicht, eine Governance-Stelle. Die Komponenten sind in der Regel Managed Services aus der gewählten Cloud, ergänzt um offene Komponenten dort, wo Portierbarkeit zählt.

Empfohlene Bausteine in dieser Größenklasse: Eine Daten-Lakehouse-Lösung wie Microsoft Fabric oder Databricks auf Azure, AWS oder GCP. Eine Modell-Inventar- und MLOps-Schicht, etwa Azure Machine Learning, Vertex AI oder eine offene MLflow-Installation. Eine LLM-Schicht über die eigene Cloud, etwa Microsoft Foundry, Vertex AI oder Amazon Bedrock, mit Auswahl zwischen mehreren Foundation-Modellen. Eine Vector-Datenbank, in der Regel als Managed Service. Eine zentrale Identitäts- und Berechtigungslogik. Ein Beobachtungs-Stack für Modellqualität, Drift und Kosten.

Der Richtwert bis zum Regelbetrieb liegt in dieser Klasse bei **500.000 bis 1,5 Millionen Euro**, einschließlich Lizenzen, Cloud-Verbrauch, Aufbauleistungen und Beratung. Die Streuung ist groß, weil die Datenausgangslage stark variiert. Unternehmen mit einer schon konsolidierten Datenlandschaft sparen einen wesentlichen Teil. Unternehmen, die parallel ihr Data Warehouse aufräumen, schlagen ihn drauf.

Pfad C: Konzern ab 5.000 Mitarbeitende

In der Konzerngröße ist eine eigene KI-Plattform gesetzt; offen ist, wie sie geschnitten wird. Hier greift das **AI Factory Pattern**, das wir in Kapitel 4 beschrieben haben: eine Plattform, die mehrere Geschäftsbereiche bedient, mit klaren Schnittstellen, zentraler Governance und dezentraler Nutzung.

Die typische Konzern-Konstellation ist Multi-Cloud, ob bewusst gewählt oder historisch gewachsen. Das hat Konsequenzen für die Architektur: Die Trennschicht zwischen Anwendung und Plattform muss tiefer liegen, die Pipelines müssen portierbarer sein, die Identitäts- und Berechtigungslogik muss anbieterübergreifend funktionieren. Was im Mittelstand eine pragmatische Vereinfachung sein darf, ist im Konzern eine Bedingung.

Was zusätzlich hinzukommt, ist die organisatorische Dimension. Eine AI Factory ohne ein **AI Center of Excellence** ist ein technisches Konstrukt ohne Steuerung. Das CoE übersetzt die Plattform in Anwendungsmuster, schult die Fachbereiche, hält die Standards. Es ist die Stelle, an der die drei Dimensionen Wertbeitrag, Befähigung und Architektur operativ zusammenkommen.

Empfohlene Bausteine in dieser Größenklasse, in Stichworten: Eine produktiv betriebene Data-Lakehouse-Architektur, in der Regel Multi-Cloud-fähig, mit eigener Verantwortung für Datenmodelle und Schnittstellen. Eine Modell-Plattform mit eigenem Modell-Inventar, Trainings- und Inferenzpipelines, häufig auf Kubernetes oder einer vergleichbaren Orchestrierungsschicht. Eine LLM-Strategie, die mehrere Foundation-Modelle bedient, mit eigenem Routing zwischen Modellen je nach Anwendung und Kostenlage. Eine eigene Governance-Schicht mit Risikoklassifikation, technischer Dokumentation, Audit-Fähigkeit. Ein etabliertes AI CoE mit dauerhafter Personaldecke.

Der Richtwert bis zum Regelbetrieb liegt in dieser Klasse bei **2 bis 8 Millionen Euro**, mit erheblicher Streuung nach Branche, Datenausgangslage und Tiefe des Engagements. Der untere Wert entspricht einem Konzern, der eine bestehende Cloud-Landschaft konsolidiert und um KI-spezifische Komponenten ergänzt. Der obere Wert entspricht einer echten Neuaufstellung, häufig getrieben durch eine größere Umstrukturierung oder einen regulatorischen Anlass. Auch hier gilt: Lizenzen und Cloud-Verbrauch skalieren mit der Nutzung und sind in der Bandbreite enthalten, soweit sie bis zum Regelbetrieb anfallen.

Eine Bemerkung zu Zwischenformen

Drei Pfade beschreiben drei Typen, und reale Unternehmen sind selten Idealtypen. Ein hochregulierter Mittelständler mit 800 Mitarbeitenden in der Versicherungsbranche kann näher an Pfad C liegen als an Pfad B, weil seine Compliance-Last die Architekturlast nach oben treibt. Ein internationaler Konzern mit 20.000 Mitarbeitenden, aber dezentraler Struktur, kann sinnvoll wie eine Reihe paralleler Pfad-B-Unternehmen denken. Entscheidend ist die Kombination aus Größe, Regulierung, Datenausgangslage und vorhandenen Fähigkeiten; die Mitarbeiterzahl allein reicht als Maßstab nicht. Am meisten leisten die Pfade, wenn sie als Orientierung gelesen werden.

PFAD A

KMU und Mittelstand

bis etwa 500 Mitarbeitende

LEITSTRATEGIE

SaaS-first mit strukturierter Exit-Option

BAUSTEINE

- Produktivitätsschicht, etwa Microsoft 365 Copilot
- RAG über Managed Services oder Mittelstands-SaaS
- Identity-Provider für konsistente Berechtigungen
- Einfaches Monitoring von Kosten und sensiblen Inhalten

RICHTWERT BIS REGELBETRIEB

150.000 bis 400.000 €

zuzüglich nutzerabhängiger Lizenzen

PFAD B

Großmittelstand

500 bis 5.000 Mitarbeitende

LEITSTRATEGIE

Hybride Architektur mit Cloud-Core

BAUSTEINE

- Lakehouse, etwa Microsoft Fabric oder Databricks
- Modell-Inventar und MLOps-Schicht
- LLM-Schicht mit mehreren Foundation-Modellen
- Vektordatenbank, Identitätslogik, Beobachtungs-Stack

RICHTWERT BIS REGELBETRIEB

0,5 bis 1,5 Mio. €

inkl. Lizenzen, Cloud, Aufbau, Beratung

PFAD C

Konzern

ab 5.000 Mitarbeitende

LEITSTRATEGIE

AI Factory mit AI Center of Excellence

BAUSTEINE

- Multi-Cloud-fähiges Data Lakehouse
- Modell-Plattform, häufig auf Kubernetes
- LLM-Routing über mehrere Modelle
- Eigene Governance-Schicht, AI CoE mit fester Personaldecke

RICHTWERT BIS REGELBETRIEB

2 bis 8 Mio. €

starke Streuung nach Branche und Ausgangslage

Abbildung 7.2 · Drei Architektur-Pfade nach Unternehmensgröße, mit Leitstrategie, Bausteinen und Richtwert der Investition bis zum Regelbetrieb.

7.3 Vier Meilensteine bis zum produktionsreifen KI-Betrieb

Der Weg von einer Reihe von Piloten zu einer produktiven KI-Landschaft lässt sich in vier Phasen gliedern. Jede endet mit einem Meilenstein, der sich prüfen lässt, und jede setzt den Meilenstein der vorigen voraus: M1 blockt M2, M2 blockt M3, M3 blockt M4. Diese Abfolge bleibt wahr, gleich wie schnell ein Unternehmen arbeitet. Wie lange eine Phase dauert, hängt von Kapazität, Datenlage und Entscheidungswegen ab und lässt sich seriös erst mit einer eigenen Planung beziffern. Dieser Leitfaden verzichtet deshalb auf Zeitangaben.

Die Phasen laufen selten linear ab. In der Realität überlappen sie sich, weil Unternehmen nicht warten können, bis das Fundament fertig ist, bevor sie den ersten Use Case vorbereiten. Die Abhängigkeit gilt für die Meilensteine: Ein Use Case darf vorbereitet werden, bevor M1 erreicht ist; produktiv gehen sollte er erst danach.

Phase 1: Foundation, Meilenstein M1

In dieser Phase wird wenig gebaut, das nach KI aussieht, und viel, das Voraussetzung dafür ist. Es ist die Phase, in der die meisten Programme die schlechtesten Bilder produzieren, weil es nichts zu zeigen gibt. Die Versuchung, etwas vorzuziehen, ist groß. Ein Programm, das ihr nachgibt, baut den ersten Use Case auf Sand.

Was in dieser Phase geleistet wird: Eine dokumentierte Datenstrategie, die sich auf eine Liste von zwei oder drei Datenbereichen herunterbricht, die bis zum Meilenstein M3 KI-Anwendungen tragen sollen. Ein Cloud-Setup, das mehr ist als ein Konto: Regionen festgelegt, Identitätsverwaltung angebunden, Kostenstellen eingezogen, ein erstes Sicherheitsmodell verabschiedet. Eine Auswahl der ersten ein bis drei Use Cases, mit dokumentiertem Business Case und einer Person, die das Ergebnis verantwortet. Dazu die Risikoklasse jedes ausgewählten Use Case nach AI Act und ein erstes Angebot zur KI-Kompetenz nach Artikel 4.

Was hier bewusst noch nicht passiert: die große Plattform, die fertige Governance, das vollständige Tooling. Ein Programm, das in dieser Phase alles auf einmal aufstellen will, verschwendet die Aufmerksamkeit der Organisation für die wenigen Dinge, die wirklich am Anfang stehen müssen.

Meilenstein M1 ist erreicht, wenn Datenstrategie, Cloud-Grundaufbau und die ausgewählten Use Cases mit Business Case, Verantwortlichem und Risikoklasse freigegeben sind. Abbruchkriterium: Findet sich für keinen Use Case ein Business Case mit messbarer Zielgröße, geht das Programm zurück in die Use-Case-Auswahl, bevor Phase 2 beginnt.

Als Richtwert setzen wir für diese Phase **50.000 bis 150.000 Euro** in der Mittelstands-Größe an, deutlich mehr in der Konzern-Klasse. Er umfasst typischerweise externe Begleitung, erste Lizenzen, Cloud-Setup und die internen Aufwände für die Klärung von Verantwortung und Geschäftsfällen.

Phase 2: First Value, Meilenstein M2

In dieser Phase gehen die ersten Use Cases in Produktion. Zwei, vielleicht drei, in einer Form, die im Tagesgeschäft genutzt wird, gemessen und verantwortet ist. Das Programm muss hier zum ersten Mal liefern, was es versprochen hat.

Was dabei aufgebaut wird, reicht über die Use Cases hinaus. Es sind die Anfänge einer **MLOps-Grundlage**: ein erstes Modell-Inventar, automatisierte Deployment-Pipelines für die produktiv gehenden Anwendungen, ein Monitoring, das mehr leistet als die Selbstauskunft des Entwicklers. Es ist die erste **Iteration der Governance**: eine Stelle, die KI-Risiken bewertet, eine Klassifikation der produktiven Anwendungen nach AI-Act-Vorgaben, eine technische Dokumentation, die vor dem ersten Audit entsteht.

Die häufigste Falle dieser Phase ist ein Use Case, der ohne Rückmeldung aus der Praxis zu lange poliert wird. Eine Anwendung, die in der Produktion bei 80 Prozent läuft und sichtbar genutzt wird, ist wertvoller als eine, die 95 Prozent aller Features umgesetzt hat und noch gar nicht freigegeben ist. Architektur lebt vom Echtbetrieb. Was im Echtbetrieb läuft, lehrt mehr als jeder Pilot.

Meilenstein M2 ist erreicht, wenn zwei bis drei Use Cases produktiv laufen, für jeden eine vor dem Start vereinbarte Kennzahl gemessen wird und jede Anwendung im Modell-Inventar mit Risikoklasse und Verantwortlichem geführt ist. Abbruchkriterium: Erreicht ein Use Case seine Kennzahl nicht, wird er nachgebessert oder eingestellt, bevor er als Vorlage für die Plattform dient.

Als Richtwert setzen wir für diese Phase **100.000 bis 250.000 Euro** in der Mittelstands-Größe an. Er enthält die Aufbauleistungen für MLOps und Governance, die produktive Inbetriebnahme der ersten Use Cases, externe Begleitung und beginnenden Cloud-Verbrauch, der mit dem ersten produktiven Betrieb spürbar wird.

Phase 3: Scale, Meilenstein M3

Diese Phase ist die anspruchsvollste, weil sich hier entscheidet, ob aus drei Use Cases ein Programm wird oder eine Sammlung. Bauliche Antwort: die **Plattform**. Wo bisher Anwendungen einzeln betrieben wurden, übernimmt eine Plattform die wiederkehrenden Aufgaben: Datenzugriff, Modellverwaltung, Deployment, Beobachtung, Berechtigung.

Was in dieser Phase passiert, hängt vom Pfad ab. Auf Pfad A ist die Plattform eine Zusammenstellung von Managed Services mit klaren Anbindungen. Auf Pfad B beginnt sich die eigene AI Platform zu formen, mit eigener Modell-Schicht und eigenem Inventar. Auf Pfad C wird aus dem, was bisher in einem Bereich gebaut wurde, ein Plattform-Angebot für mehrere Bereiche, mit dem ersten Schritt zum AI Center of Excellence.

Parallel laufen fünf bis zehn Use Cases produktiv, in unterschiedlichen Reifegraden. Einige liefern bereits messbare Wirkung. Andere werden zurückgenommen, weil sie weniger geliefert haben als gedacht. Das Zurücknehmen gehört dazu und ist eine Disziplin für sich. Ein Programm, das nie einen Use Case einstellt, steuert nicht ehrlich.

Die Governance wird in dieser Phase erwachsen. Aus der ersten Klassifikation wird ein Lebenszyklus: Use Cases durchlaufen definierte Entscheidungsstufen, technische Dokumentation entsteht unterwegs, Risiken werden in der Architektur abgebildet. Was nach Bürokratie klingt, ist die Voraussetzung dafür, dass die Plattform in Phase 4 industriell betrieben werden kann.

Meilenstein M3 ist erreicht, wenn neue Use Cases die gemeinsamen Plattformdienste nutzen und jeder neue Use Case die Decision Gates aus Kapitel 6.2 durchläuft. Abbruchkriterium: Baut ein neuer Use Case dennoch eigene Komponenten für Datenzugriff, Modellverwaltung oder Beobachtung, wird die Plattform nachgeschärft, bevor weitere Use Cases folgen.

Als Richtwert setzen wir für diese Phase **200.000 bis 500.000 Euro** in der Mittelstands-Größe an; er steigt erkennbar mit der Anzahl der produktiven Use Cases. In der Konzern-Klasse verschiebt sich die Spanne in den niedrigen siebenstelligen Bereich, weil die Plattform tiefer geschnitten und mehrfach genutzt wird.

Phase 4: Industrialize, Meilenstein M4

Die letzte Phase hat einen anderen Charakter. Im Mittelpunkt steht die Überführung des Aufgebauten in den Regelbetrieb. Self-Service ist das Stichwort: Fachbereiche und Data-Science-Teams können auf der Plattform selbstständig arbeiten, ohne dass jede Anfrage durch das zentrale Team läuft.

Was hier sichtbar wird, ist eine Veränderung in der Rolle der zentralen KI-Mannschaft. Aus den Erbauern werden Plattformbetreiber. Aus Projekten werden Produkte. Aus einer Reihe einzeln verantworteter Anwendungen wird ein Portfolio, das auf Plattform-Standards aufsetzt und an Plattform-Standards gemessen wird.

Meilenstein M4 ist erreicht, wenn zehn oder mehr Use Cases in Produktion sind, ihre Beobachtung automatisiert ist, ihre Kosten je Use Case transparent sind und ihre Governance im Regelbetrieb läuft. Eine fertige KI-Strategie gibt es auch dann nicht; sie bleibt eine fortlaufende Aufgabe. Das Unternehmen hat aber eine industriell betriebene KI-Plattform, die es über mehrere Modell- und Anbietergenerationen trägt, ohne grundlegend neu aufgesetzt werden zu müssen.

Als Richtwert setzen wir für diese Phase **150.000 bis 400.000 Euro** in der Mittelstands-Größe an. Der Anteil sinkt im Verhältnis zur produktiven Nutzung, weil die großen Aufbaulasten hinter dem Programm liegen. Was bleibt, sind Betriebskosten, die in den Linienbereich gehören und nicht mehr als Programmkosten geführt werden.

Eine Zusammenfassung in Tabellenform

Tabelle 1 · Vier Phasen bis zum produktionsreifen KI-Betrieb, Richtwerte für den Mittelstand

Phase	Meilenstein	Was geleistet wird	Setzt voraus	Richtwert Investition (Mittelstand)
Foundation	M1	Datenstrategie, Cloud-Setup, erste Use Cases mit Business Case und Risikoklasse	–	50.000 bis 150.000 €
First Value	M2	Zwei bis drei Use Cases produktiv und gemessen, erste MLOps-Grundlage, erste Governance-Iteration	M1	100.000 bis 250.000 €
Scale	M3	Plattform-Aufbau, fünf bis zehn Use Cases produktiv, Decision Gates im Einsatz	M2	200.000 bis 500.000 €
Industrialize	M4	Self-Service auf der Plattform, zehn und mehr Use Cases, Regelbetrieb	M3	150.000 bis 400.000 €

Eine Wiederholung an dieser Stelle: Die Zahlen sind Richtwerte. Sie beziehen sich auf die Mittelstands-Größenklasse aus Pfad B. Für Pfad A liegen die unteren Werte der Bandbreite näher an der Wirklichkeit, für Pfad C die oberen, oft deutlich darüber. Eine seriöse Schätzung für den eigenen Fall verlangt eine eigene Aufwandsrechnung. Für die erste Diskussion in der Vorstandsetage ist die Tabelle ein redlicher Ausgangspunkt.

7.4 Die häufigsten Fallstricke und wie man sie vermeidet

Was in einem Projekt schiefgeht, lehrt mehr als das, was glattläuft. KI-Programme stolpern an einer Reihe wiederkehrender Muster. Sie sind selten dramatisch. Sie sind meistens unspektakulär, und genau das ist ihre Tücke. Ein Programm, das spektakulär scheitert, wird neu aufgesetzt. Eines, das unauffällig versandet, wird verschleppt.

Die folgenden zehn Fallstricke sortieren sich zu drei Familien. Die ersten gehen auf Architekturentscheidungen zurück. Die zweiten auf das Verhältnis zwischen Pilot und Produktion. Die dritten auf Governance und Steuerung. Jede Familie hat eine eigene Logik, in der sie zuschlägt, und eine eigene Logik, in der man ihr begegnet.

Vier Architekturentscheidungen, die man früh richtig trifft

1. Plattform vor Use Case. Foundation und Plattform sind zwei verschiedene Dinge, und sie werden häufig verwechselt. Foundation ist das, was bis zum Meilenstein M1 gelegt wird: Datenarchitektur, Cloud-Setup, Identitätsverwaltung. Plattform ist das, was aus dem produktiven Betrieb der ersten Anwendungen entsteht, wenn aus wiederholten Aufgaben gemeinsame Komponenten werden. Ein Programm, das die zweite mit der ersten erledigen will, baut für Anwendungen, die es noch nicht gibt.

2. Werkzeug-Inflation. Eine Inflation an MLOps-Werkzeugen, jedes für sich gut, in Summe nicht zusammenhängend, ist ein häufiges Bild in Unternehmen, in denen verschiedene Teams parallel beginnen. Die Antwort ist eine kleine, dokumentierte Liste der zentralen Komponenten und die Disziplin, sich darauf zu beschränken. Eine Liste mit fünf Bausteinen, die zusammenpassen, schlägt eine Liste mit fünfzehn, die es nicht tun.

3. Unbewusste Anbieterbindung. Die Frage nach Anbieterabhängigkeit wird in Workshops aufgeworfen und in der Architektur nicht beantwortet. Die saubere Antwort ist die bewusste Wahl, an welcher Stelle man sich bindet und an welcher man sich Spielraum lässt. Kapitel 2 und 4 beschreiben das ausführlich. An dieser Stelle nur die Erinnerung: Vendor Lock-in ist akzeptabel, wenn er erstens bewusst gewählt, zweitens dokumentiert und drittens regelmäßig überprüft wird. Er ist gefährlich, wenn er sich heimlich aufbaut.

4. Datenfrage vertagt. Programme, die mit der Modellseite beginnen und die Datenseite vertagen, kommen früher oder später an einen Punkt, an dem das Vertagen Konsequenzen hat. Modelle sind so gut wie die Daten, auf denen sie arbeiten. Ohne Datenstrategie ist eine KI-Strategie eine Modell-Spielwiese. Die Reihenfolge ist umkehrbar, die Konsequenz nicht.

Drei Phänomene zwischen Pilot und Produktion

5. Pilot ohne Anschluss. Das verbreitetste Muster und das, das den größten Schaden anrichtet. Eine Organisation hat fünfzehn oder zwanzig Piloten, von denen drei oder vier sehenswert sind, aber keiner produktiv läuft. Die Ursache ist selten ein Mangel an guten Ideen. Sie ist fast immer ein Mangel an Verantwortung dafür, einen Piloten in Produktion zu bringen. Die Antwort ist organisatorisch: eine Stelle, die die Verantwortung für die Inbetriebnahme trägt, eine Definition, ab wann ein Pilot endet, und eine Bereitschaft, Piloten ohne Produktions-Perspektive ehrlich einzustellen.

6. Wirkung nicht gemessen. Use Cases gehen produktiv, ohne dass jemand definiert hat, woran Erfolg gemessen wird. Später kann niemand sagen, ob die Anwendung wirkt. Beispiel: Microsoft Copilot wird ausgerollt. Im Vorstand ein Erfolg, weil er existiert. Im Alltag weiß niemand, ob er Zeit spart oder nur eine teure neue Schnittstelle ist. Die Lösung: eine Kennzahl, die vor dem Start vereinbart wird, zum Beispiel Bearbeitungszeit oder Akzeptanzrate. Dazu eine verantwortliche Stelle, die regelmäßig berichtet. Mehr braucht es nicht, um aus einer Lizenz ein Ergebnis zu machen.

7. Modell nicht beobachtet. Anwendungen gehen in Produktion und werden dann sich selbst überlassen. Modelle altern. Die Daten, mit denen ein Modell in Produktion arbeitet, weichen mit der Zeit von den Daten ab, auf denen es entstanden ist. Die Antwortqualität sinkt, ohne dass jemand hinsieht. Eine in *Scientific Reports* veröffentlichte Studie hat diesen Qualitätsverlust in 91 Prozent der untersuchten Kombinationen aus Modell und Datensatz gemessen.[76] Monitoring geht mit der Anwendung in Produktion, lange bevor ein Kunde sich beschwert oder ein Vorstand fragt.

Drei Schwächen in Governance und Steuerung

8. Compliance nachgereicht. Governance, die nach der Architektur kommt, ist Reparaturarbeit. Sie ist teurer, langsamer und schlechter als Governance, die in die Architektur eingebaut ist. Das gilt für den EU AI Act, der eine technische Dokumentation und Risikoklassifikation verlangt, die sich nicht im Nachgang erzeugen lässt, ohne dass das System auseinandergenommen wird. Es gilt für NIS-2 und für die DSGVO. Compliance als Designprinzip macht das Bauen günstiger.

9. Schatten-KI ignoriert. Während das Programm an der offiziellen Architektur arbeitet, nutzen Fachbereiche frei verfügbare KI-Werkzeuge mit Geschäftsdaten, oft ohne dass es jemandem auffällt. Verbote helfen hier wenig, weil sie ohnehin nicht durchsetzbar sind. Was wirkt, ist ein offizielles Angebot, das so gut ist, dass die Schatten-Nutzung an Attraktivität verliert. Eine zentral bereitgestellte, freigegebene KI-Schicht ist die wirksamste Maßnahme gegen unkontrollierten Werkzeug-Wildwuchs. Mit Nutzungsregeln und Schulung ergänzt, ist sie zugleich der Nachweis für die KI-Kompetenz nach Artikel 4 des AI Act (Kapitel 6.1).

10. Niemand verantwortet. Eine Architektur ohne Eigentümer ist eine Sammlung von Diagrammen, die niemand pflegt. Eine KI-Plattform ohne Eigentümer ist eine Plattform, die im zweiten Jahr nicht mehr aktuell ist und im dritten nicht mehr läuft. Die Antwort ist banal und wird trotzdem oft versäumt: eine namentlich benannte Stelle, die für die Architektur und für die Plattform verantwortet, was sie sagt. Im Mittelstand reicht oft eine halbe Stelle. Im Konzern braucht es ein eigenes Team. In keinem Unternehmen reicht es, die Verantwortung in einem Lenkungskreis aufzulösen.

Was diese zehn Punkte teilen

Die zehn Fallstricke verteilen sich auf drei Familien, aber sie haben eine gemeinsame Wurzel. Alle gehen darauf zurück, dass eine Entscheidung, die früh getroffen werden müsste, vertagt wird. Die verfrühte Plattform vertagt die Geduld. Die nachgereichte Compliance vertagt die Verantwortung. Der Pilot ohne Anschluss vertagt die Inbetriebnahme. Die ungeklärte Eigentümerschaft vertagt sich selbst. In allen Fällen ist die Reparatur teurer als die rechtzeitige Entscheidung.

Das ist die unspektakuläre Wahrheit über KI-Programme, die scheitern. Sie scheitern an Entscheidungen, die niemand getroffen hat; die Technik ist selten der Grund.

KAPITEL 8

8 Fazit und Ausblick

Sieben Kapitel sind ein langer Weg, und es lohnt sich, am Ende noch einmal kurz zu machen, was zwischendurch zwangsläufig breit werden musste. Dieses Whitepaper hatte eine einzige These im Rücken, von der ersten Seite an. KI scheitert in deutschen Unternehmen an drei Dingen, die zusammen entschieden werden müssen: an der wirtschaftlichen Bewertung der Anwendungsfälle, an der Befähigung der Menschen, die mit ihnen arbeiten, und an der Architektur, die beides trägt. Die ersten beiden Dimensionen bekommen im öffentlichen Diskurs viel Aufmerksamkeit. Der dritten war dieses Dokument gewidmet, weil über sie am wenigsten nachgedacht wird, obwohl sie über die Wirtschaftlichkeit des Ganzen mitentscheidet.

Was bleibt, sind zwei Dinge. Eine Verdichtung dessen, was über acht Kapitel auseinandergelegt wurde, in eine Handvoll Prinzipien, die ein Vorstand mitnehmen kann. Und ein nüchterner Blick über das Jahr 2026 hinaus, denn die Architektur, die heute gebaut wird, soll auch tragen, wenn sich Modelle, Anbieter und Regeln wieder verschoben haben.

8.1 Fünf Prinzipien für 2026

Die folgenden fünf Prinzipien sind die Summe der vorangegangenen Kapitel, eingedampft auf das, was übrig bleibt, wenn man das Tooling, die Diagramme und die Detailfragen einen Moment beiseitelegt. Diese eine Seite des Whitepapers genügt, um das Wesentliche behalten zu haben.

Erstens: Architektur kommt vor dem Einkauf. Die Reihenfolge entscheidet über die Kosten. Eine Lösung, die beschafft wird, bevor geklärt ist, wie sie mit der bestehenden Landschaft zusammenarbeitet, erzeugt Folgekosten, die in keinem Angebot stehen. Die Architekturfrage gehört an den Anfang, vor die Demo und vor den Vertrag. Das ist die unbequemste der fünf Empfehlungen, weil sie verlangt, einer attraktiven Anwendung eine langweilige Vorfrage voranzustellen. Sie ist auch die wirksamste, weil sich an dieser Stelle die meisten der in Kapitel 7 beschriebenen Fallstricke im Vorfeld auflösen.

Zweitens: Die Datenschicht ist das Fundament. Kein Modell ist besser als die Daten, auf denen es arbeitet. Programme, die mit der Modellseite beginnen und die Datenseite vertagen, kommen verlässlich an den Punkt, an dem das Vertagen seinen Preis verlangt. Datenqualität, Datenverfügbarkeit und Datengovernance entscheiden zuerst, was überhaupt möglich ist. Die Modellwahl kommt danach. Ein im Umfeld der Harvard Business Review veröffentlichter Beitrag nennt die Datenbereitschaft als häufigen Grund, warum generative KI nicht über den Piloten hinauskommt.[77] Die Reihenfolge ist umkehrbar, die Konsequenz nicht.

Drittens: Modular, anbieterneutral, beobachtbar. Diese drei Eigenschaften unterscheiden eine Architektur, die fünf Jahre trägt, von einer, die nach zwei Jahren neu gemacht werden muss. Modularität erlaubt, einzelne Bausteine auszutauschen, ohne das Ganze umzubauen. Anbieterneutralität hält den Wechsel denkbar, auch dort, wo man sich bewusst bindet. Beobachtbarkeit macht sichtbar, was ein System tut, und meldet, wenn es etwas tut, das es nicht tun sollte. Im Pilotmodus sind die drei verzichtbar. Im Produktivbetrieb sind sie der Unterschied zwischen Bestand und Neuanfang.

Viertens: Governance ist eine Schicht der Architektur. EU AI Act, NIS-2 und DSGVO greifen auffällig oft an derselben Architekturentscheidung an. Eine einzige saubere Protokollierung erfüllt eine Pflicht des AI Act, liefert das Material für die Sicherheitsüberwachung nach NIS-2 und stützt die Auskunftsfähigkeit der DSGVO. Drei Regelwerke verlangen darum eine Architektur, die ihre gemeinsamen Anforderungen an einer Stelle erfüllt, eingebaut, solange das System entsteht. Nachgerüstet kostet dasselbe Ergebnis ein Vielfaches.

Fünftens: Jedes produktive System hat einen Namen. Für jede produktive KI-Anwendung muss eine namentlich benannte Person verantwortlich sein. Diese Person beantwortet im Zweifel die Frage, warum ein System eine bestimmte Entscheidung getroffen hat, und sie hat die Befugnis, es abzuschalten. Das klingt banal und wird trotzdem am häufigsten versäumt. Eine Architektur ohne Eigentümer ist eine Sammlung von Diagrammen, die niemand pflegt. Im Mittelstand reicht oft eine halbe Stelle. Im Konzern braucht es ein Team. In keinem Unternehmen löst sich die Verantwortung in einem Lenkungskreis auf.

Diese fünf Prinzipien haben eine gemeinsame Eigenschaft. Keines davon ist eine Technologiefrage. Alle fünf sind Entscheidungen, die früh fallen müssen, und alle fünf werden teurer, je länger man sie vertagt. Das ist die unspektakuläre Summe dieses Whitepapers. Die anspruchsvolle Arbeit an einer KI-Architektur besteht weniger im Bauen als im rechtzeitigen Entscheiden.

8.2 Was nach 2026 kommt

Ein Whitepaper mit der Jahreszahl im Titel trägt eine Verpflichtung: ehrlich zu sein über das, was es nicht wissen kann. Die folgenden drei Entwicklungen zeichnen sich für die Jahre nach 2026 ab. Keine davon ist Spekulation, alle drei sind in Bewegung, und alle drei verschieben die Architekturfrage, ohne sie aufzuheben.

Agentische Systeme werden zur Regel. Kapitel 4 hat das Muster des agentischen Geflechts beschrieben, zu einem Zeitpunkt, an dem es in den meisten Unternehmen noch Experiment war. Diese Phase geht zu Ende. Mit der Verbreitung autonomer Agenten verschiebt sich die Governance-Frage allerdings an eine Stelle, an der die klassischen Modelle nicht greifen. Gartner warnt, dass ein einheitliches Regelwerk über alle Agenten hinweg eher zum Scheitern führt als zur Kontrolle: Ein Agent, der Dokumente nur lesend zusammenfasst, braucht andere Schranken als einer, der eigenständig Aktionen auslöst. Gartner erwartet, dass bis 2027 40 Prozent der Unternehmen autonome Agenten wegen Governance-Lücken zurückstufen oder abschalten.[78] Die plausible Antwort ist eine abgestufte Governance, die Agenten nach ihrer Autonomie und Reichweite klassifiziert und jeder Stufe eine eigene Vertrauensgrenze zuordnet. Das Prinzip aus Abschnitt 8.1 bleibt unberührt: Auch ein autonomer Agent braucht einen Namen, der für ihn geradesteht. Die Frage, wer einen Agenten abschaltet, wenn er aus dem Ruder läuft, wird in den nächsten Jahren häufiger gestellt werden als die Frage, was er kann.

Die Regulierung bleibt in Bewegung. Der EU AI Act hat 2026 seine erste große Änderung erfahren: Seit dem 27. Juli 2026 verschiebt der Digital Omnibus die Hochrisiko-Pflichten für eigenständige Systeme nach Anhang III auf den 2. Dezember 2027, für in regulierte Produkte eingebettete Systeme auf den 2. August 2028.[79] In Deutschland überwacht seit dem 29. Juli 2026 die Bundesnetzagentur die Einhaltung, und die erste harmonisierte Norm ist angenommen und wartet auf ihre Zitierung im Amtsblatt. Die Verschiebung ist eine Atempause. Entwarnung gibt sie nicht. Die Anforderungen an technische Dokumentation, Risikoklassifikation und Erklärbarkeit bleiben dieselben. Sie kommen später, und sie kommen sicher. Die Konsequenz für die Architektur ist gleichlautend mit der aus Kapitel 6: Was als Designprinzip eingebaut wird, ist auch dann günstig, wenn der Stichtag sich verschiebt. Was auf den Stichtag hin nachgerüstet wird, bleibt teuer, gleichgültig wann er fällt.

Die Rolle des Architekten verschiebt sich. Mit autonomen Systemen, die Entscheidungen treffen, verschiebt sich auch, was Enterprise Architektur eigentlich leistet. Die Arbeit verlagert sich vom statischen Blueprint zur Frage, wo Verantwortung und Risiko in einem System sitzen, das sich zur Laufzeit selbst verändert. Ein Risiko in diesem Bild verdient einen eigenen Namen: das verwaiste Modell, das in Produktion läuft, ohne dass eine Person dafür geradesteht. Die Aufgabe des Architekten verschiebt sich damit vom Entwerfen von Strukturen zum Verankern von Verantwortung. Es ist dieselbe Bewegung, die dieses ganze Whitepaper trägt, eine Ebene höher. Die Architektur entscheidet, ob KI funktioniert. Die benannte Verantwortung für die Architektur entscheidet, ob sie funktionsfähig bleibt.

Was über alle drei Entwicklungen hinweg stabil bleibt, ist die Diagnose, mit der dieses Dokument begann. Die Technik wird sich weiter ändern, schneller als jede Prognose. Die drei Dimensionen ändern sich nicht. Eine KI-Einführung wird auch 2028 an der wirtschaftlichen Bewertung, der organisatorischen Befähigung und der technischen Architektur gemessen werden. Die Modelle, die heute neu sind, werden bis dahin alt sein. Die Architekturfrage wird dieselbe geblieben sein.

8.3 Vom Whitepaper zur Umsetzung

Ein Whitepaper kann eine Lage ordnen und einen Weg beschreiben. Den Weg gehen muss jedes Unternehmen selbst, mit seiner eigenen Datenlage, seinen eigenen Anwendungsfällen, seinem eigenen Reifegrad. Roman Langfeld Consulting (RLC) begleitet diesen Weg an drei Stellen, die sich an den Kapiteln dieses Dokuments orientieren.

Am Anfang steht eine Standortbestimmung. Das **AI Architecture Assessment** nimmt den Readiness Check aus Kapitel 7 und führt ihn in der Tiefe durch, in einem kompakten Workshop-Format mit den Verantwortlichen aus IT, Daten und Fachbereich. Am Ende stehen eine ehrliche Verortung in den fünf Dimensionen und eine Liste der nächsten Schritte, die zur Lage des Unternehmens passen.

Darauf folgt, wo es gebraucht wird, das **Enterprise AI Architecture Design**. In einem abgegrenzten Auftrag mit festem Ergebnis entsteht die Zielarchitektur für die konkrete Organisation: Datenschicht, Plattform, Muster und Governance, abgestimmt auf einen der drei Pfade aus Kapitel 7 und auf die Anwendungsfälle, die wirtschaftlich tragen. Das Ergebnis ist eine Architektur, die mit dem bestehenden IT-Team umsetzbar ist und den Skalierungspfad von Anfang an mitdenkt.

Die dritte Stelle ist die Governance. Die **Implementierung eines AI Governance Framework** übersetzt die regulatorischen Anforderungen aus Kapitel 6 in eine Architektur und in benannte Verantwortung, eingebaut, solange die Systeme entstehen. Das ist die Arbeit, die sich am wenigsten aufschieben lässt und am teuersten nachzuholen ist.

Allen drei Angeboten liegt dieselbe Haltung zugrunde, die dieses Whitepaper trägt. Architektur macht die wirtschaftliche Bewertung belastbar und gibt der organisatorischen Befähigung ein Ziel. RLC denkt KI-Strategie und KI-Architektur als eine Einheit, anbieterneutral und mittelstandstauglich. Das Whitepaper war der Anfang dieses Gesprächs. Die Fortsetzung beginnt mit einer einzigen Frage, der schwersten der drei aus Kapitel 7, und der einzigen, die ein Vorstand stellen muss, um anzufangen: *Und wo fangen wir an?*

Roman Langfeld
Consulting

Roman Langfeld

consulting@romanlangfeld.de
+49 (0)174 3228558
www.romanlangfeld.de

www.romanlangfeld.de

Quellen

1. Bitkom e. V., *Erstmals nutzt die Mehrheit der Unternehmen KI*, Presseinformation, Berlin, 14. September 2026. Telefonische Befragung von 603 Unternehmen ab 20 Beschäftigten in Deutschland in den Kalenderwochen 28 bis 33 des Jahres 2026, repräsentativ für die Gesamtwirtschaft. Angaben zur Potenzialnutzung beziehen sich auf Unternehmen mit KI im Einsatz, Angaben zu Hemmnissen auf Unternehmen ohne KI.
2. Challapally, A., Pease, C., Raskar, R., Chari, P., *The GenAI Divide: State of AI in Business 2025*, MIT Project NANDA, Cambridge MA, Juli 2025, Executive Summary. Breit rezipiert u. a. in Fortune, 18. August 2025.
3. Stanford Institute for Human-Centered AI (HAI), *AI Index Report 2026*, Kapitel 4 (Economy), Stanford, April 2026, Überblick S. 171 und Chapter Highlights.
4. Harvard Business Review, *Scaling Generative AI for Value: Data Leader Agenda for 2025*, Sponsored Content von Amazon Web Services, 20. November 2024.
5. McKinsey & Company (QuantumBlack), *Seizing the agentic AI advantage*, 13. Juni 2025. Dort auch die Unterscheidung horizontaler und vertikaler Anwendungsfälle und der Begriff *agentic AI mesh*.
6. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Art. 99 Abs. 3 und 4. Die Fristen für Hochrisiko-Systeme hat die Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*), in Kraft seit 27. Juli 2026, verschoben: für eigenständige Hochrisiko-Systeme nach Anhang III vom 2. August 2026 auf den 2. Dezember 2027, für in regulierte Produkte nach Anhang I eingebettete Systeme vom 2. August 2027 auf den 2. August 2028. Einzelheiten in Kapitel 6.1.
7. The Open Group, *TOGAF Standard, 10th Edition*, Reading, 2022. ArchiMate als zugehörige Modellierungssprache: The Open Group, *ArchiMate 3.2 Specification*, Reading, 2023. Seit April 2026 liegt die *ArchiMate 4 Specification* vor, mit rund 30 Prozent weniger Konzepten. Die Unterstützung in Modellierungswerkzeugen ist uneinheitlich; das verbreitete Open-Source-Werkzeug Archi arbeitet weiterhin mit Version 3.2.
8. Vela, D. et al., *Temporal quality degradation in AI models*, Scientific Reports, Bd. 12, Art. 11654, 2022. Untersucht wurden 32 Datensätze aus vier Branchen (Gesundheitswesen, Verkehr, Finanzen, Wetter) mit vier verbreiteten Modelltypen. Vergleiche grundlegend Sculley, D. et al., *Hidden Technical Debt in Machine Learning Systems*, NeurIPS 2015.
9. National Institute of Standards and Technology (NIST), *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*, NIST AI 100-2 E2025, Gaithersburg, März 2025, mit Evasion, Poisoning, Angriffen auf die Privatsphäre und Prompt Injection bei generativer KI. Für Europa: ENISA, *Artificial Intelligence Cybersecurity Challenges: Threat Landscape for Artificial Intelligence*, Athen, 15. Dezember 2020, mit einer Taxonomie KI-spezifischer Bedrohungen, darunter Data Poisoning und die Manipulation von Modellen.
10. Verordnung (EU) 2016/679 (Datenschutz-Grundverordnung, *DSGVO*), Art. 25 („Datenschutz durch Technikgestaltung und durch datenschutzfreundliche Voreinstellungen"), in Kraft seit 25. Mai 2018.
11. The Open Group, *TOGAF Standard, 10th Edition*, Fundamental Content: *Architecture Development Method*, Reading, 2022.
12. Sculley, D. et al., *Hidden Technical Debt in Machine Learning Systems*, NeurIPS 2015, Abschnitt 3 (*Data Dependencies Cost More than Code Dependencies*). Vergleiche Polyzotis, N. et al., *Data Management Challenges in Production Machine Learning*, ACM SIGMOD 2017.
13. DAMA International, *DAMA-DMBOK: Data Management Body of Knowledge*, 2. Auflage, Technics Publications, Basking Ridge NJ, 2017, Kapitel 13 (Data Quality); ISO/IEC 25012:2008, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model*. Die Nachvollziehbarkeit der Herkunft wird in der englischsprachigen Fachliteratur als *Data Lineage* bzw. *Provenance* geführt und in den klassischen Qualitätskanons eher als eigene Daten-Management-Disziplin als als Qualitätsmerkmal behandelt.
14. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Art. 10 („Daten und Daten-Governance"), insbesondere Abs. 2 lit. b bis g zu Datenherkunft, Aufbereitung und Eignungsprüfung für Hochrisikosysteme.
15. DAMA International, *DAMA-DMBOK: Data Management Body of Knowledge*, 2. Auflage, Technics Publications, Basking Ridge NJ, 2017, Kapitel 3 (Data Governance).

16. Verordnung (EU) 2016/679 (Datenschutz-Grundverordnung, *DSGVO*), Art. 3 (räumlicher Anwendungsbereich), anwendbar seit 25. Mai 2018.
17. Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*) zur Änderung der Verordnung (EU) 2024/1689, Amtsblatt der Europäischen Union vom 24. Juli 2026, in Kraft seit 27. Juli 2026. Einzelheiten in Kapitel 6.1.
18. Verordnung (EU) 2022/868 (*Data Governance Act*), anwendbar seit 24. September 2023; Verordnung (EU) 2023/2854 (*Data Act*), anwendbar seit 12. September 2025.
19. Gaia-X European Association for Data and Cloud AISBL, *Gaia-X Architecture Document*, Release 24.04, Brüssel, 2024; die Nachfolgefassung 25.11 ist im Software-Release *Danube* umgesetzt. Zu den im deutschen und europäischen Markt 2026 relevanten Anbietern zählen unter anderem STACKIT (Schwarz Digits), Open Telekom Cloud (T-Systems), OVHcloud, IONOS, Scaleway und Nebius. Die Aufzählung ist nicht erschöpfend; die Eignung hängt von Workload, Region und benötigtem Reifegrad des Plattform-Stacks ab.
20. Armbrust, M. et al., *Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics*, CIDR 2021. Für die offenen Tabellenformate siehe die jeweiligen Projektspezifikationen Delta Lake (Linux Foundation), Apache Iceberg und Apache Hudi, Stand 2025.
21. Dehghani, Z., *Data Mesh: Delivering Data-Driven Value at Scale*, O'Reilly Media, Sebastopol CA, 2022. Erstbeschreibung des Konzepts: Dehghani, Z., *How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh*, martinfowler.com, Mai 2019.
22. Polyzotis, N. et al., *Data Management Challenges in Production Machine Learning*, Google Research, ACM SIGMOD 2017. Zum Begriff *training-serving skew* siehe insbesondere Abschnitt 4.
23. FinOps Foundation, *FinOps Framework*, Linux Foundation, San Francisco, laufend gepflegt. Zur Übertragung auf KI-spezifische Kostenstrukturen siehe die Arbeitspapiere der FinOps Foundation zum Thema *FinOps for AI*, etwa *GenAI FinOps: How Token Pricing Really Works* und *Optimizing GenAI Usage: A FinOps Perspective on Cost, Performance, and Efficiency*, https://www.finops.org/topic/finops-for-ai/, abgerufen am 2. Oktober 2026.
24. Lewis, P. et al., *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, Facebook AI Research / UCL, NeurIPS 2020. Erste systematische Beschreibung des Musters.
25. Stanford Institute for Human-Centered AI (HAI), *AI Index Report 2026*, Kapitel 4 (Economy), Stanford, April 2026, Chapter Highlights.
26. McKinsey & Company (QuantumBlack), *Seizing the agentic AI advantage*, 13. Juni 2025. Dort werden Agenten als Weg aus dem Gen AI Paradox beschrieben und das *agentic AI mesh* als Orchestrierungsschicht für viele Agenten eingeführt.
27. Linux Foundation, *Linux Foundation Announces the Formation of the Agentic AI Foundation*, Pressemitteilung, 9. Dezember 2025; Model Context Protocol, *Specification 2026-07-28, Key Changes*, https://modelcontextprotocol.io/specification/2026-07-28/changelog, abgerufen am 2. Oktober 2026.
28. Agent2Agent (A2A) Protocol, Projektseite, https://a2a-protocol.org/latest/, abgerufen am 2. Oktober 2026.
29. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Art. 6 Abs. 2 i. V. m. Anhang III. Die Einstufung als Hochrisiko-System folgt dem Einsatzbereich, unabhängig vom Grad der Autonomie.
30. OWASP GenAI Security Project, *OWASP Top 10 for Agentic Applications*, veröffentlicht am 9. Dezember 2025, Kennungen ASI01 bis ASI10.
31. Bitkom e. V., *Erstmals nutzt die Mehrheit der Unternehmen KI*, Presseinformation, Berlin, 14. September 2026. Repräsentative Befragung von 603 Unternehmen ab 20 Beschäftigten.
32. Vela, D. et al., *Temporal quality degradation in AI models*, Scientific Reports, Bd. 12, Art. 11654, 2022. Vier Modelltypen auf 32 Datensätzen aus Gesundheitswesen, Verkehr, Finanzen und Wetter; Verschlechterung über die Zeit in 91 Prozent der 128 Kombinationen.
33. John, M. M., Holmström Olsson, H., Bosch, J., *An empirical guide to MLOps adoption: Framework, maturity model and taxonomy*, Information and Software Technology, Bd. 183, Art. 107725, 2025. Fallstudie mit sieben Unternehmen, Validierung des Frameworks über insgesamt vierzehn Unternehmen.
34. Microsoft, *MLOps Maturity Model*, Azure Architecture Center, Stand 3. Juni 2026, https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-maturity-model, abgerufen am 2. Oktober 2026.
35. Google Cloud, *MLOps: Continuous delivery and automation pipelines in machine learning*, Cloud Architecture Center, fortlaufend aktualisiert, abgerufen am 2. Oktober 2026. Drei Stufen (Level 0 bis 2) und Continuous Training (CT) als dritter Bestandteil produktiver ML-Pipelines.
36. Ebenda, insbesondere zur Beschreibung von Level 2, in der die trainierende Pipeline selbst als Auslieferungs-Artefakt definiert wird.
37. Polyzotis, N. et al., *Data Validation for Machine Learning*, Proceedings of the 2nd SysML Conference, 2019. Konzeptioneller Ursprung der Trennung in Datentests und Modelltests in CI/CD-Pipelines.
38. MLflow, *Model Registry*, Projektdokumentation, https://mlflow.org/docs/latest/ml/model-registry/, abgerufen am 2. Oktober 2026: Herkunft, Versionierung, Aliase, Metadaten und Annotationen je Modellversion.
39. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Anhang IV i. V. m. Art. 11. Pflicht zur technischen Dokumentation für Hochrisiko-KI-Systeme.
40. Amazon Web Services, *Shadow tests*, Amazon SageMaker AI Developer Guide, https://docs.aws.amazon.com/sagemaker/latest/dg/shadow-tests.html, abgerufen am 2. Oktober 2026.
41. Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*), Amtsblatt der Europäischen Union vom 24. Juli 2026, in Kraft seit 27. Juli 2026. Zu den Fristen nach Anhang III und Anhang I sowie zur Schonfrist nach Art. 50 Abs. 2 bis zum 2. Dezember 2026 für Systeme, die vor dem 2. August 2026 in Verkehr gebracht wurden: Gibson Dunn, *EU AI Act Omnibus Agreement: Postponed High-Risk Deadlines and Other Key Changes*, abgerufen am 2. Oktober 2026.
42. Projektdokumentationen von MLflow (https://mlflow.org/docs/latest/) und Weights & Biases (https://docs.wandb.ai/), abgerufen am 2. Oktober 2026.

43. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Anhang IV i. V. m. Art. 11. Pflicht zur technischen Dokumentation für Hochrisiko-KI-Systeme über den gesamten Lebenszyklus.

44. John, M. M., Holmström Olsson, H., Bosch, J., *An empirical guide to MLOps adoption: Framework, maturity model and taxonomy*, Information and Software Technology, Bd. 183, Art. 107725, 2025, Auswertung der Datendimension.

45. Vela, D. et al., *Temporal quality degradation in AI models*, Scientific Reports, Bd. 12, Art. 11654, 2022.

46. Google Cloud, *MLOps: Continuous delivery and automation pipelines in machine learning*, Cloud Architecture Center, fortlaufend aktualisiert. Zu Trigger-Strategien für Continuous Training (Zeitplan, Datenverfügbarkeit, Drift-Signal) siehe die Beschreibung von Level 1 und Level 2.

47. Gama, J. et al., *A Survey on Concept Drift Adaptation*, ACM Computing Surveys, Vol. 46, No. 4, 2014. Grundlegende Unterscheidung von Data Drift und Concept Drift sowie ihrer Erkennungsverfahren.

48. Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Art. 10 (Daten und Daten-Governance) i. V. m. Art. 15 (Genauigkeit, Robustheit und Cybersicherheit). Pflicht zur Untersuchung möglicher Verzerrungen und zu Maßnahmen ihrer Minderung über den Lebenszyklus von Hochrisiko-KI-Systemen.

49. Microsoft, *Generative AI Operations for Organizations with MLOps Investments*, Azure Architecture Center, https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/genaiops-for-mlops, abgerufen am 2. Oktober 2026. Dort wird der Prompt samt Orchestrierung, Kontextdaten und Suchindex als zentraler, mit GenAIOps zu steuernder Bestandteil beschrieben.

50. Ma, W. et al., *(Why) Is My Prompt Getting Worse? Rethinking Regression Testing for Evolving LLM APIs*, arXiv:2311.11123, 2023/2024. Empirische Untersuchung der Verhaltensverschiebung kommerzieller LLM-Schnittstellen über aufeinanderfolgende Modellversionen bei unverändertem Aufruf.

51. Europäische Kommission, *Guidelines for providers of general-purpose AI models*, sowie Verordnung (EU) 2024/1689 (KI-Verordnung, *EU AI Act*), Art. 53 (Pflichten für Anbieter von KI-Modellen mit allgemeinem Verwendungszweck). Anwendung der GPAI-Pflichten ab dem 2. August 2025; für vor diesem Datum in Verkehr gebrachte Modelle Übergangsfrist bis zum 2. August 2027.

52. Microsoft, *Generative AI Operations for Organizations with MLOps Investments*, a. a. O., Abschnitte zu Experimentation und Deployment: Prompts und Orchestrierung werden wie andere Artefakte versioniert, getestet und über Rollout-Strategien ausgeliefert.

53. Ma, W., Yang, C., Kästner, C., *(Why) Is My Prompt Getting Worse? Rethinking Regression Testing for Evolving LLM APIs*, arXiv:2311.11123, 2023, überarbeitet 2024.

54. Yamauchi, Y., Yano, T., Oyamada, M., *An Empirical Study of LLM-as-a-Judge: How Design Choices Impact Evaluation Reliability*, arXiv: 2506.13639, 2025.

55. Stanford Institute for Human-Centered Artificial Intelligence, *Artificial Intelligence Index Report 2025*, Stanford University, April 2025, Kapitel 1 (Research and Development). Rückgang des Anfragepreises auf GPT-3.5-Leistungsniveau von 20,00 US-Dollar (November 2022) auf 0,07 US-Dollar je Million Token (Oktober 2024), mehr als das 280-Fache.

56. Anthropic, *Pricing*, Abschnitte Batch processing und Prompt caching, https://platform.claude.com/docs/en/about-claude/pricing, abgerufen am 2. Oktober 2026; OpenAI, *Batch API*, https://developers.openai.com/api/docs/guides/batch, abgerufen am 2. Oktober 2026.

57. Verordnung (EU) 2024/1689 des Europäischen Parlaments und des Rates vom 13. Juni 2024 zur Festlegung harmonisierter Vorschriften für Künstliche Intelligenz (*KI-Verordnung, EU AI Act*), Amtsblatt der Europäischen Union, L-Reihe, 12. Juli 2024. In Kraft getreten am 1. August 2024.

58. Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*) zur Änderung der Verordnung (EU) 2024/1689, Amtsblatt der Europäischen Union vom 24. Juli 2026, in Kraft seit 27. Juli 2026. Vorläufige Einigung von Rat und Parlament am 7. Mai 2026, endgültige Zustimmung des Rates am 29. Juni 2026. Verschiebung der Anwendbarkeit der Hochrisiko-Pflichten für eigenständige Systeme nach Anhang III vom 2. August 2026 auf den 2. Dezember 2027, für in regulierte Produkte nach Anhang I eingebettete Systeme vom 2. August 2027 auf den 2. August 2028. Amtlicher Titel: Verordnung (EU) 2026/1744 des Europäischen Parlaments und des Rates vom 8. Juli 2026 (*Digital-Omnibus-Verordnung zur KI*), ELI http://data.europa.eu/eli/reg/2026/1744/oj. Wortlaut der deutschen Fassung geprüft am 2. Oktober 2026, insbesondere Art. 1 Nr. 5 (Art. 4), Nr. 39 (Schonfrist für Art. 50 Abs. 2), Nr. 40 (Art. 113 Abs. 3) und Nr. 42 (Anhang VIII). Zum Verfahrensablauf vgl. NicFab, *Digital Omnibus on AI: Regulation (EU) 2026/1744 Is Published in the Official Journal*, abgerufen am 2. Oktober 2026.

59. CEN-CENELEC, *EN 18286 in the Spotlight: Supporting Compliance with the AI Act*, Brüssel, 30. Juli 2026, zur EN 18286:2026 (*Artificial intelligence – Quality management system for EU AI Act regulatory purposes*). Zum Stand der Entwürfe prEN 18228, prEN 18229, prEN 18282 und prEN 18284 und zur noch ausstehenden Zitierung im Amtsblatt: Modulos, *EU AI Act Harmonized Standards: Presumption of Conformity and the JTC 21 Pipeline*, abgerufen am 2. Oktober 2026.

60. Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*), vgl. Abschnitt 6.1. Neue Verbote nach Art. 5 für KI-Systeme, die nicht einvernehmliche intime Darstellungen oder Darstellungen sexuellen Kindesmissbrauchs erzeugen oder manipulieren, mit Geltung ab dem 2. Dezember 2026. Art. 50 gilt seit dem 2. August 2026; Schonfrist für die maschinenlesbare Kennzeichnung nach Art. 50 Abs. 2 bis zum 2. Dezember 2026 für Systeme, die vor dem 2. August 2026 in Verkehr gebracht wurden. Vgl. Gibson Dunn, *EU AI Act Omnibus Agreement: Postponed High-Risk Deadlines and Other Key Changes*, abgerufen am 2. Oktober 2026.

61. Verordnung (EU) 2024/1689, Art. 4 in der Fassung der Verordnung (EU) 2026/1744 (Art. 1 Nr. 5). Vgl. NicFab, a. a. O.

62. Bundesnetzagentur, *Bundesnetzagentur übernimmt zentrale Rolle bei der Umsetzung der KI-Verordnung*, Pressemitteilung, Bonn, 29. Juli 2026, zum Gesetz zur Marktüberwachung und Innovationsförderung von künstlicher Intelligenz (KI-MIG).

63. Verordnung (EU) 2024/1689, Art. 6 Abs. 3 (Filtermechanismus für Anhang-III-Systeme ohne erhebliches Risiko) i. V. m. Art. 49 (Registrierung). Zur Straffung der einzureichenden Informationen (Wegfall der Nummern 7 und 9 in Anhang VIII Abschnitt B) bei fortbestehender Registrierungspflicht siehe Verordnung (EU) 2026/1744 sowie NicFab, a. a. O.

64. Verordnung (EU) 2024/1689, Art. 11 i. V. m. Anhang IV. Pflicht zur technischen Dokumentation für Hochrisiko-KI-Systeme; die neun Bestandteile des Anhangs IV reichen von der Systembeschreibung über Entwicklungsmethode, Datengrundlage, Leistungskennzahlen und Risikomanagement bis zur EU-Konformitätserklärung und dem Plan zur Beobachtung nach dem Inverkehrbringen. Für KMU und Start-ups ist eine vereinfachte Form vorgesehen, für die die Kommission ein Formular bereitstellen soll.

65. Verordnung (EU) 2024/1689, Art. 43 i. V. m. Anhang VI (interne Kontrolle) und Anhang VII (Bewertung durch eine benannte Stelle). Zur Pflicht der EU-Konformitätserklärung und CE-Kennzeichnung siehe Art. 47 und Art. 48.

66. ISO/IEC 42001:2023 (*Information technology — Artificial intelligence — Management system*) sowie ISO/IEC 23894:2023 (*Information technology — Artificial intelligence — Guidance on risk management*). Ergänzend ISO/IEC 42005:2025 (*Information technology — Artificial intelligence — AI system impact assessment*), ein Leitfaden ohne Zertifizierung. Keine dieser Normen begründet eine formale Konformitätsvermutung unter dem AI Act; sie gelten als der nächstgelegene strukturierte Bezugsrahmen, solange harmonisierte Normen nicht im Amtsblatt zitiert sind.

67. ISO/IEC 42001:2023 (*Information technology — Artificial intelligence — Management system*), Abschnitt 5 (Führung). Die Norm verpflichtet die oberste Leitung, eine KI-Politik festzulegen, Rollen und Verantwortlichkeiten samt Entscheidungsrechten und Eskalationswegen zuzuweisen und das KI-Managementsystem in die strategischen Ziele der Organisation einzubinden.

68. Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin), Rundschreiben 06/2026 (BA), *Mindestanforderungen an das Risikomanagement (MaRisk)*, 9. Novelle, 30. Juni 2026, Modul AT 4.3.4 (Verwendung von Modellen), Tz. 1 bis 6. Modelle im Anwendungsbereich der CRR sind ausgenommen. Neue oder verschärfte Anforderungen gelten nach einer Übergangsfrist ab dem 1. Januar 2027.

69. National Institute of Standards and Technology (NIST), *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations*, NIST AI 100-2 E2025, Gaithersburg, März 2025. Klassifikation von Angriffen auf prädiktive und generative KI, darunter Evasion, Poisoning, Angriffe auf die Privatsphäre wie Model Inversion und Membership Inference sowie Prompt Injection.

70. OWASP GenAI Security Project, *OWASP Top 10 for LLM Applications 2026*, 3. August 2026: LLM01 Prompt Injection, LLM03 Excessive Agency. Zu Rangfolge und Änderungen gegenüber der Ausgabe 2025 vgl. Imperva, *OWASP LLM Top 10 2026: What Changed and Why*, abgerufen am 2. Oktober 2026.

71. Gesetz zur Umsetzung der NIS-2-Richtlinie und zur Regelung wesentlicher Grundzüge des Informationssicherheitsmanagements in der Bundesverwaltung (*NIS-2-Umsetzungsgesetz*), Bundesgesetzblatt 2025 I Nr. 301, verkündet am 5. Dezember 2025, in Kraft getreten am 6. Dezember 2025 ohne Übergangsfrist. Umsetzung über eine Novellierung des BSI-Gesetzes; Erweiterung des Anwendungsbereichs von rund 4.500 auf etwa 29.500 beaufsichtigte Einrichtungen, gegliedert in „wichtige" und „besonders wichtige Einrichtungen". Kernpflichten: Registrierung beim BSI (erste Frist 6. März 2026), Meldung erheblicher Sicherheitsvorfälle, Umsetzung und Dokumentation von Risikomanagementmaßnahmen.

72. Europäischer Datenschutzausschuss (EDSA/EDPB), *Opinion 28/2024 on certain data protection aspects related to the processing of personal data in the context of AI models*, angenommen am 17. Dezember 2024. Die Anonymität eines KI-Modells ist im Einzelfall zu prüfen; maßgeblich ist die Wahrscheinlichkeit, dass sich mit angemessenen Mitteln personenbezogene Daten aus dem Modell gewinnen lassen, etwa durch gezielte Abfragen. Ein Modell gilt erst als anonym, wenn der Verantwortliche belegt, dass diese Wahrscheinlichkeit unerheblich ist. Zur DSGVO selbst siehe Verordnung (EU) 2016/679, insbesondere Art. 25 (Datenschutz durch Technikgestaltung) und Art. 22 (automatisierte Entscheidungen im Einzelfall).

73. Bitkom e. V., *Erstmals nutzt die Mehrheit der Unternehmen KI*, Presseinformation, Berlin, 14. September 2026. Repräsentative Befragung von 603 Unternehmen ab 20 Beschäftigten; Angaben zur Potenzialnutzung beziehen sich auf Unternehmen mit KI im Einsatz.

74. OECD, *Emerging divides in the transition to artificial intelligence*, OECD Regional Development Papers Nr. 147, Paris, Juni 2025, S. 13, auf Basis von Eurostat-Daten für 2024.

75. Microsoft, *Microsoft 365 Copilot, Preise und Pläne*, https://www.microsoft.com/de-de/microsoft-365-copilot/business und https://www.microsoft.com/de-de/microsoft-365-copilot/pricing/enterprise, abgerufen am 2. Oktober 2026. Für Copilot Business gilt bis 31. Dezember 2026 ein Aktionspreis von 15,60 Euro.

76. Vela, D. et al., *Temporal quality degradation in AI models*, Scientific Reports, Bd. 12, Art. 11654, 2022. Siehe grundsätzlich auch Sculley, D. et al., *Hidden Technical Debt in Machine Learning Systems*, NeurIPS 2015.

77. Harvard Business Review, *Scaling Generative AI for Value: Data Leader Agenda for 2025*, Sponsored Content von Amazon Web Services, 20. November 2024. Siehe auch die Befunde von MIT Project NANDA in Kapitel 1.

78. Gartner, *Gartner Says Applying Uniform Governance Across AI Agents Will Lead to Enterprise AI Agent Failure*, Pressemitteilung, 26. Mai 2026. Empfehlung einer abgestuften Governance, die KI-Agenten nach Autonomiegrad klassifiziert und jeder Stufe angemessene Kontrollen zuordnet.

79. Verordnung (EU) 2026/1744 (*Digital Omnibus on AI*), Amtsblatt der Europäischen Union vom 24. Juli 2026, in Kraft seit 27. Juli 2026. Vgl. die ausführliche Darstellung in Abschnitt 6.1.