

Kernsysteme im Spannungsfeld von Legacy-Modernisierung und KI-fähigen Architekturen

Die meisten Versicherer stecken mitten in ihrer IT-Modernisierung (fest)

Ein Teil der Anwendungen ist modernisiert, aber weiterhin zahlreiche Legacy-Anwendungen

Typische Situation von Versicherungsunternehmen



→ Ein Teil der Bestands- und Querschnittssysteme ist auf **Standard-Software** umgestellt



→ Die Umsetzung und Migration war **deutlich teurer** und **deutlich länger** als geplant



→ Die Standard-Software basiert oft auf 10 bis 20 Jahre alten **Architekturen**, die **monolithisch** und wenig flexibel sind



→ Vendoren erhöhen den Druck, On-Prem-Lösungen durch **SaaS-Lösungen** abzulösen



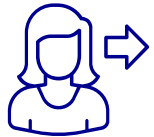
→ Die Vendoren treiben die **Lizenz-Kosten** nach oben bei gleichzeitigem **Vendor-lock-in**

Umgesetzte
Modernisierungen
meist **nicht**
zufriedenstellend

Gleichzeitig steigt
der Druck, die
verbleibenden
Mainframe-Systeme
abzulösen

Drei Haupttreiber forcieren die Diskussion um den Mainframe-Exit

Auslöser



Skills & Kapazitäten



Modernisierungs-Stau



Kostensteigerungen

- ➔ **Großer Anteil an Mainframe-Entwicklerinnen und -Entwicklern** mit einem Durchschnittsalter von über 50 Jahren
- ➔ Ausbildung und externes Sourcing reduzieren **Risiken des Skill-Verlustes** nicht dauerhaft
- ➔ Umbau oder Neubau **großer Monolithen** kostenintensiv und komplex und wird so häufig aufgeschoben
- ➔ Diskrepanz zu neu entwickelten Systemen wird immer größer und **hemmt „End-to-end“-Modernisierungsansätze**
- ➔ Betrieb der Mainframe-Infrastruktur unterliegt durch starken Vendor Lock-in **absehbar hohen Kostensteigerungen**
- ➔ **Externes Mainframe-Know-how** wird sukzessive immer **teurer**

Fünf klassische Ansätze zur Modernisierung werden diskutiert

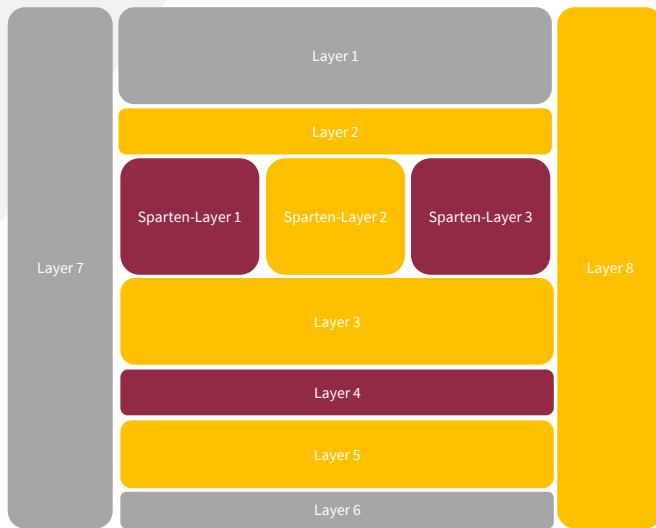
Risiken und Chancen der Mainframe-Exit-Optionen

Retire 	Rehost 	Replatform 	Rebuild 	Replace 
Abschaltung der Anwendung	Emulation des Mainframe in der Cloud	Automatisierte Umstellung von Cobol nach Java	Neubau auf anderer Plattform	Ersatz durch Kaufsoftware / SaaS
Geringster Aufwand	Skill-Problem bleibt	Generierter Code (immer noch) nicht gut wartbar	Hoher Aufwand	Sehr hoher Aufwand
Nur selten umsetzbar	Hoher Vendor-Lock-In	Cloud-Kosten-Risiko	Gute Kenntnis der Systeme nötig	Meist komplexe Integration
	Komplex in Bezug auf Schnittstellen	Für Systeme mit wenig Änderungen	Risiko, Nutzenpotenziale zu übersehen	Basis für Business-Transformation

Mainframe-Anwendungen finden sich in fast allen fachlichen Domänen

Die Gesamtkomplexität ist die Herausforderung

Mainframe-Footprint



Legende

low

medium

high

Herausforderungen

- Nahezu **in allen Domänen der IT-Landschaft** noch Mainframe-Anwendungen
- Enge Kopplung bedingt **Kette von technischen Abhängigkeiten**
- Mainframe ist nicht nur Cobol – weitere **Alttechnologien** (z.B. Assembler, IMS-DB) erschweren automatisierte Umstellungen
- **Mainframebasierte zentrale Jobsteuerung** für alle Anwendungen (ca. 50.000 Jobs an einem typischen Arbeitstag)
- **Veraltete Dokumentationen** oft nicht ausreichend für externe Wartung

Transversale Systeme stellen eine besondere Herausforderung dar

Interaktion mit zahlreichen anderen Systemen

Herausforderungen transversaler Systeme

- ➔ **Schnittstellen mit zahlreichen Systemen**, da Querschnitts-Systeme spartenübergreifende Funktionen zur Verfügung stellen, z.B. Partner, Zahlungsverkehr, IAM, Output-Management, Archiv
- ➔ **Platzierung in der Ablösungs-Roadmap ist anspruchsvoll**, da ihre Abschaltung von vielen Systemen abhängt

Abschaltung
am **Ende**



Abschaltung
am **Anfang**



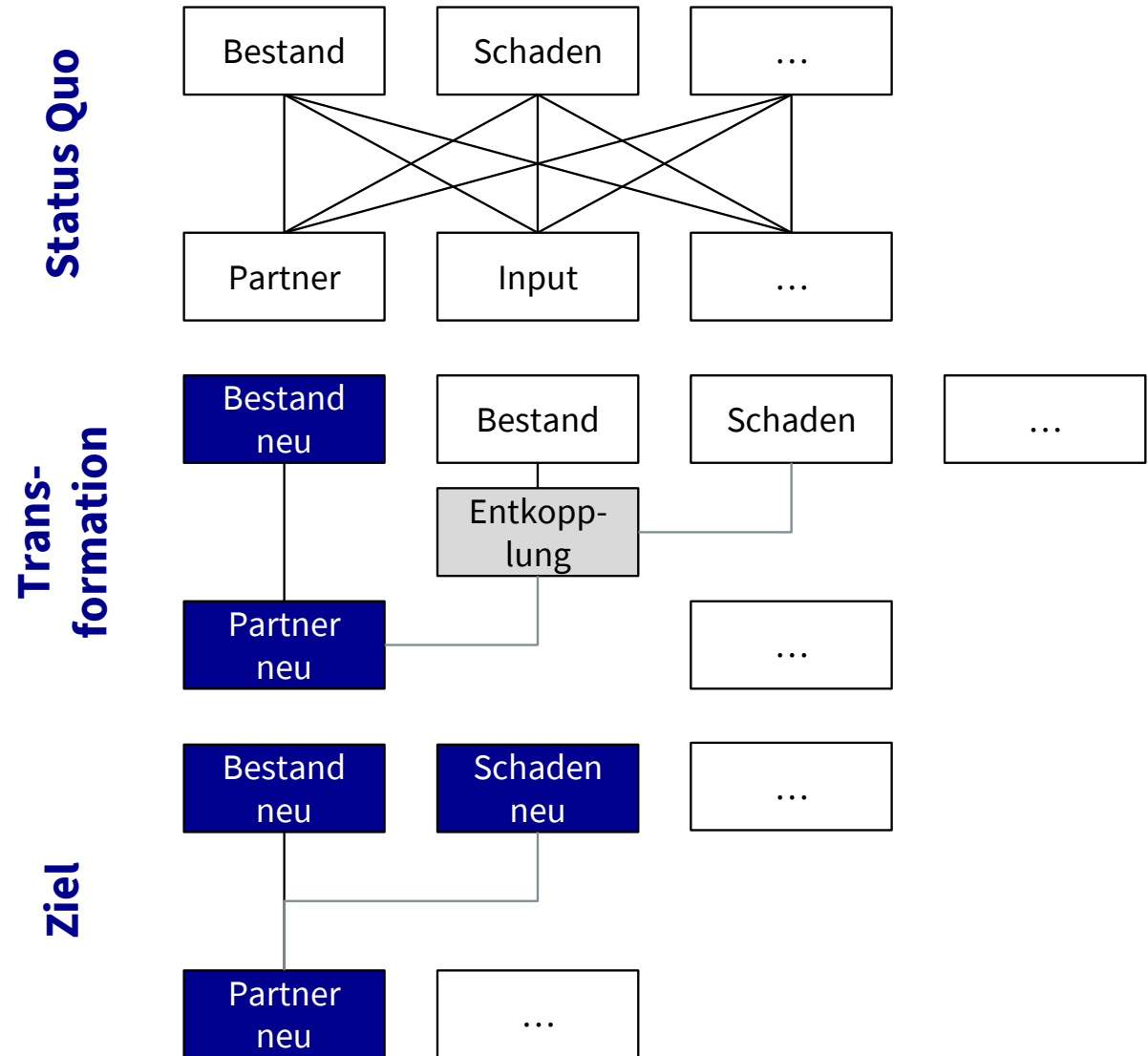
- ➔ Neue Fachsysteme nutzen zunächst alte transversale Systeme und damit ggf. veraltete Business-Logik
- ➔ Double-Run-Kosten, falls neue transversale Systeme parallel
- ➔ Risiko des weiteren Hinauszögerns, weil Nutzeneffekte bei Fachsystemen bereits geholt
- ➔ Alte Mainframe-Fachsysteme müssen vor Ablösung noch an neue transversale Systeme angebunden werden

Entkopplungsschicht löst das Problem der transversalen Systeme

- ➔ Reduktion der Abhängigkeiten ermöglicht eine parallele Transformation
- ➔ Mapping und Monitoring werden in der Entkopplungsschicht gebündelt

Aber

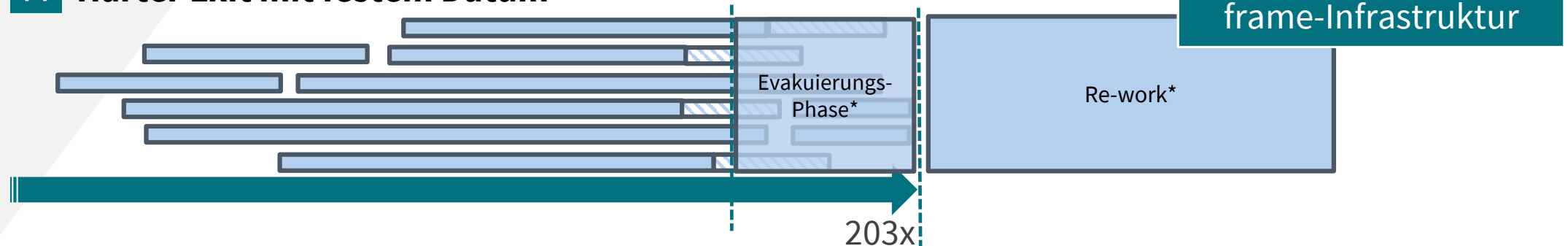
- ➔ Zusätzliche Plattform-/ Middleware-Kosten



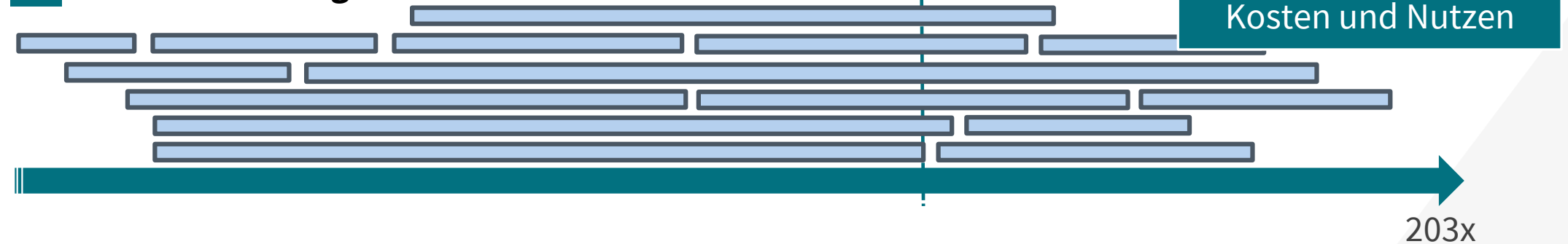
Zwei verschiedene Ansätze als Basis für eine Roadmap-Entwicklung

Geschwindigkeit versus Nutzen

A Harter Exit mit festem Datum



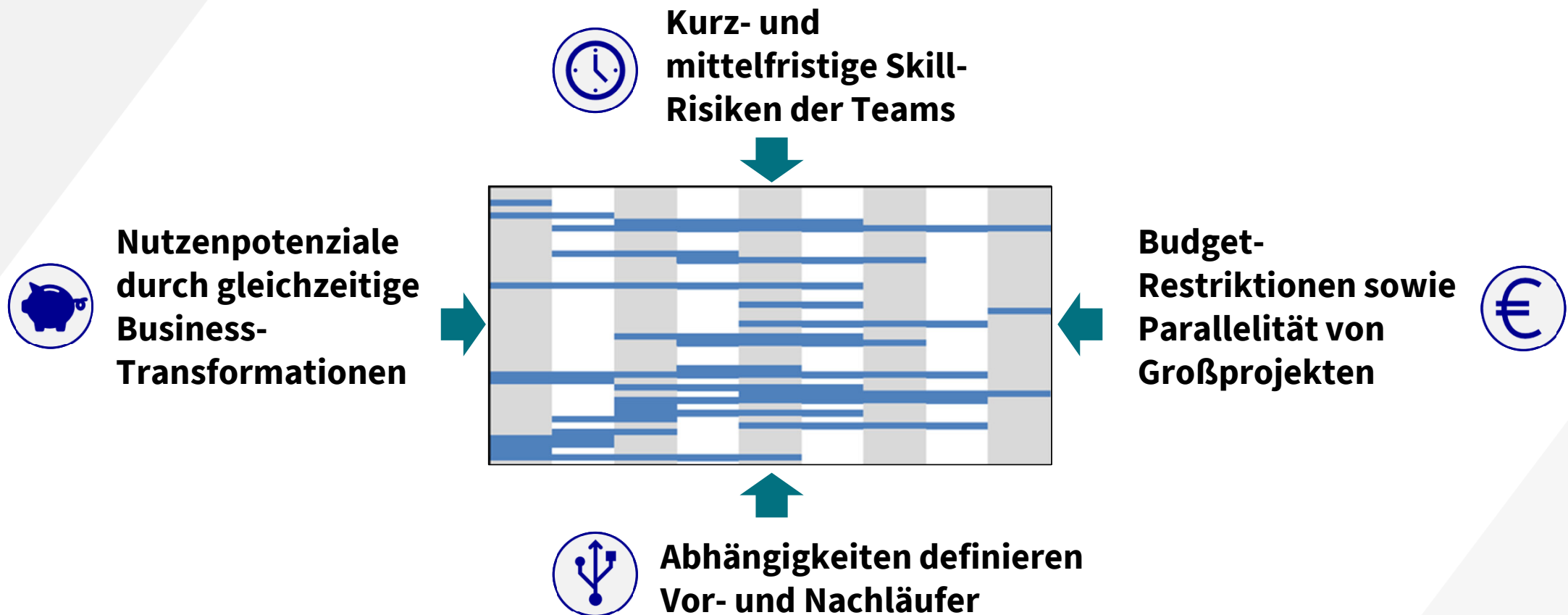
B "Natürliche" Migration



*) wo notwendig

Verschiedene Faktoren beeinflussen die Entwicklung der Exit-Roadmap

Kriterien für die Erstellung der Mainframe-Roadmap



Der Mainframe-Exit darf nicht als IT-Übung, sondern muss als Business-Transformation verstanden werden

Langfristige Planung des Mainframe-Exit führt zu Paradigmen-Wechsel in Richtung Business-Transformation

- ➔ Umstellung der großen Kernsysteme setzt **strategische Priorisierung auf Unternehmensebene** aus Business-Perspektive voraus
- ➔ IT gibt Rahmenbedingungen vor (techn. Abhängigkeiten, Umstellungspattern) und **steuert die Diskussion zur Reihenfolge** der Business-Transformationen
- ➔ Damit einhergehend ist ein **kultureller Change im Top-Management**: Tech-Themen, Architekturprinzipien, Capability-Landkarten etc. werden zunehmend natürlicher Teil von Diskussionen
- ➔ Die **Umsetzung** der Transformations-Roadmap erfordert einen **langen Atem**. Dies kann unter folgenden Voraussetzungen gelingen:
 - ➔ **Commitment auf Exit** und konsequente Modernisierung/Vereinfachung
 - ➔ **Langfristige Disziplin** hinsichtlich der Ziele und des Vorgehens

KI wirkt auf zweifache Weise auf Modernisierungsvorhaben

Software-Entwicklung und KI-Integration



KI erleichtert die Software-Entwicklung

- ➔ KI optimiert und vereinfacht den **Softwareentwicklungsprozess**, z.B. durch Unterstützung bei der Spezifikation, Definition von Test Cases, durch automatisierte Codegenerierung oder durch Testautomatisierung

Eigenentwicklung ist wieder eine realistische Option für die Ablösung von Alt-Systemen



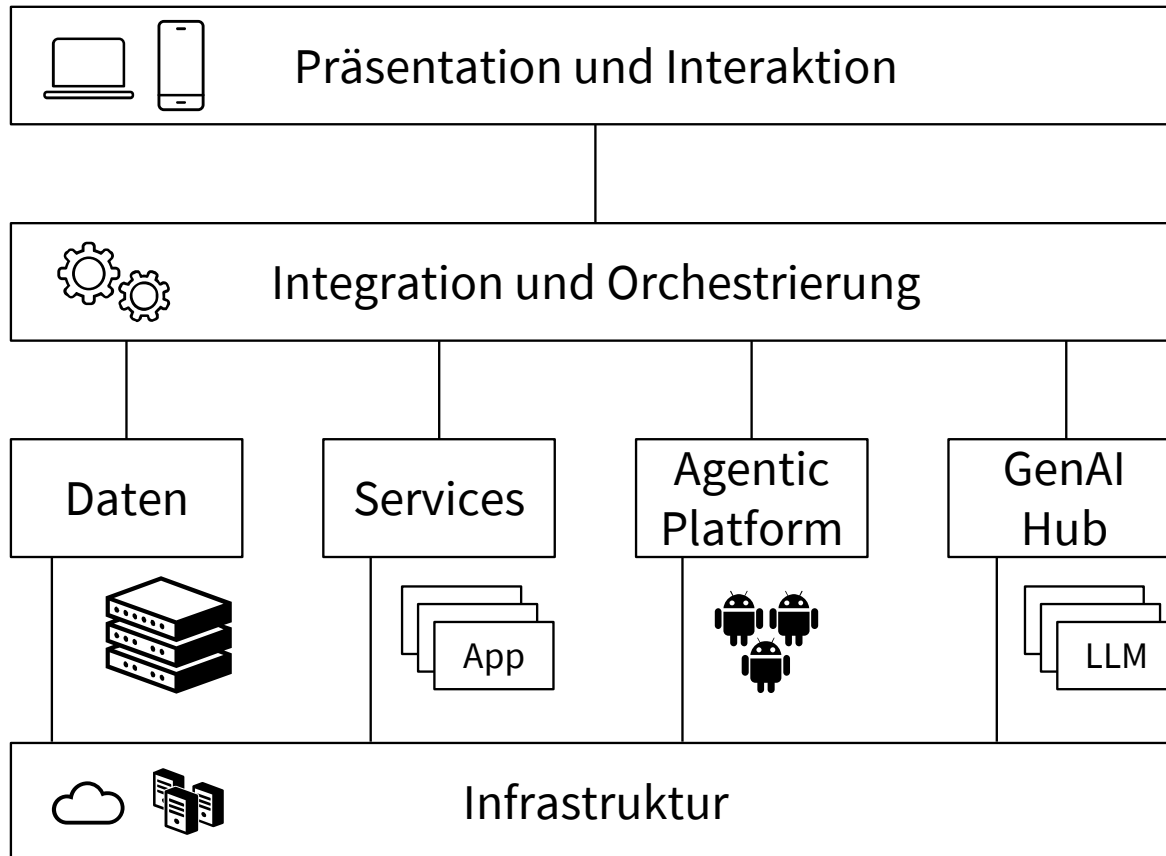
KI ermöglicht es, Prozesse zu optimieren oder neu zu denken

- ➔ KI-Lösungen **automatisieren Prozesse** oder verbessern die **Qualität der Ergebnisse** entlang von (veränderten) Prozessketten
- ➔ KI(-Agenten) können im Rahmen von **klassischen Workflows** eingesetzt werden oder können Prozesse selbst steuern (z.B. **Multi-Agenten-Systeme**)

Architekturen müssen eine **flexible Integration** verschiedener Komponenten ermöglichen

KI erfordert SOA-Ansatz

Generische Darstellung einer KI-fähigen Architektur



- Der Einsatz von KI (AI, GenAI, Agentic AI) forciert die Umsetzung von **serviceorientierten Architekturen**
- Business-Logik wird von Präsentation und Interaktion entkoppelt („**Headless Architecture**“)
- Monolithische Anwendungen werden in **Services** aufgeteilt, die **selektiv aufrufbar** sind
- **KI-Lösungen** müssen (wie auch andere Services) **flexibel** in Workflows **integrierbar** sein
- Eine **Orchestrierung** verschiedener **(KI-)Services** wird ermöglicht
- Verschiedene **(domänenspezifische) LLMs** werden bereitgestellt

Durch KI gewinnt die (eigene) IT bei Versicherern wieder an Bedeutung

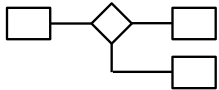
Hypothesen zur IT-Modernisierung



Mehr Eigenentwicklung



Kauflösungen selektiv



Klass. Workflows bleiben



On-Prem als Option

Eigenentwicklung gewinnt an Bedeutung durch KI-Unterstützung und weil Software-Entwickler wieder leichter am Markt verfügbar sind

Anbieter müssen ihre Standard-Software in (kleinen) Komponenten / Services zur Verfügung stellen, die leicht in bestehende Architekturen integrierbar sind

Der Einsatz von „echten“ KI-Agenten in Versicherungs-Prozessen wird vorerst begrenzt bleiben, da klassische Workflows aufgrund klarer Prozessabläufe effizienter sind

Aufgrund der hohen (Token-)Kosten für den Einsatz der großen Foundation Models und um bessere Ergebnisse zu erzielen, werden kleine / spezifische LLMs eingesetzt und diese ggf. lokal betrieben



Danke

**Vielen Dank
für Ihre
Aufmerk-
samkeit!**



**Dr. Matthias
Gröbner**

- ➔ Der Informatiker Matthias Gröbner begann nach der Promotion seine berufliche Laufbahn bei den Strategieberatungen BCG und Roland Berger, wo er für verschiedene Versicherungsunternehmen u.a. in den Bereichen PMI, Restrukturierung, CRM und Transformation der IT beratend tätig war.
- ➔ Ab 2013 leitete er bei der Digitalisierungs-Beratung Detecon als Managing Partner das Financial-Services-Geschäft und beriet Versicherungen und Banken bei Fragen der IT- und Facharchitektur, IT Governance sowie bei der digitalen Transformation.
- ➔ Seit 2021 ist er verantwortlich für das Enterprise Architecture Management und die IT-Strategie des AXA-Konzerns in Köln.