



Embedded Architecture & Change Economics

Hohe Produkt-, Partner- und Betriebsvarianz braucht eine dedizierte Variations- und Capability-Schicht

KASKO & VIG Platform Partners

16. September 2026

Ausgangspunkt | Wir sind für Vertrieb angetreten und bei der IT Architektur gelandet



Situation/Complication VIG Platform Partners

1



Neuer Kundenzugang

- Versicherung dort verkaufen, wo der Kunde ohnehin ist
- Z.B. Telco, Bank, Airline, Loyalty, Plattform

2



Neue Geschwindigkeit

- Partner denken nicht in Versicherungs-Releasezyklen
- Ein Deal muss in Monaten umsetzbar sein, teilweise in Wochen

3



Hohe Unsicherheit

- Anfangs offen, welches Produkt, welcher Partner und welches Modell skaliert
- Ausprobieren und verändern muss möglich bleiben

4



Konzernrealität

- Viele Risikoträger, Märkte und gewachsene Systeme
- Die bestehende IT nicht bei jedem Deal umbauen



Unsere erste IT-Frage war deshalb nicht: Welches Policy-System brauchen wir?
Sondern: Wie schaffen wir maximale geschäftliche Optionalität bei vertretbaren Grenzkosten?

Folge | Daraus entstand ein anderes Anforderungsprofil an die IT



Fünf Geschäftsanforderungen, bevor über Architektur gesprochen wird



Geschwindigkeit

Von der Partneridee zum Produkt in Monaten statt in klassischen Projektzyklen



Variabilität

Produkte, Preise, Journeys, Rollen und Prozesse ohne Großprojekt verändern



Wiederverwendung

Was einmal gebaut ist, beim nächsten Partner möglichst wiederverwenden



Konnektivität

Mit unterschiedlichen Versicherern und deren vorhandener IT arbeiten können



Optionalität

Keine große Wette auf eine Zielarchitektur, bevor klar ist, was skaliert

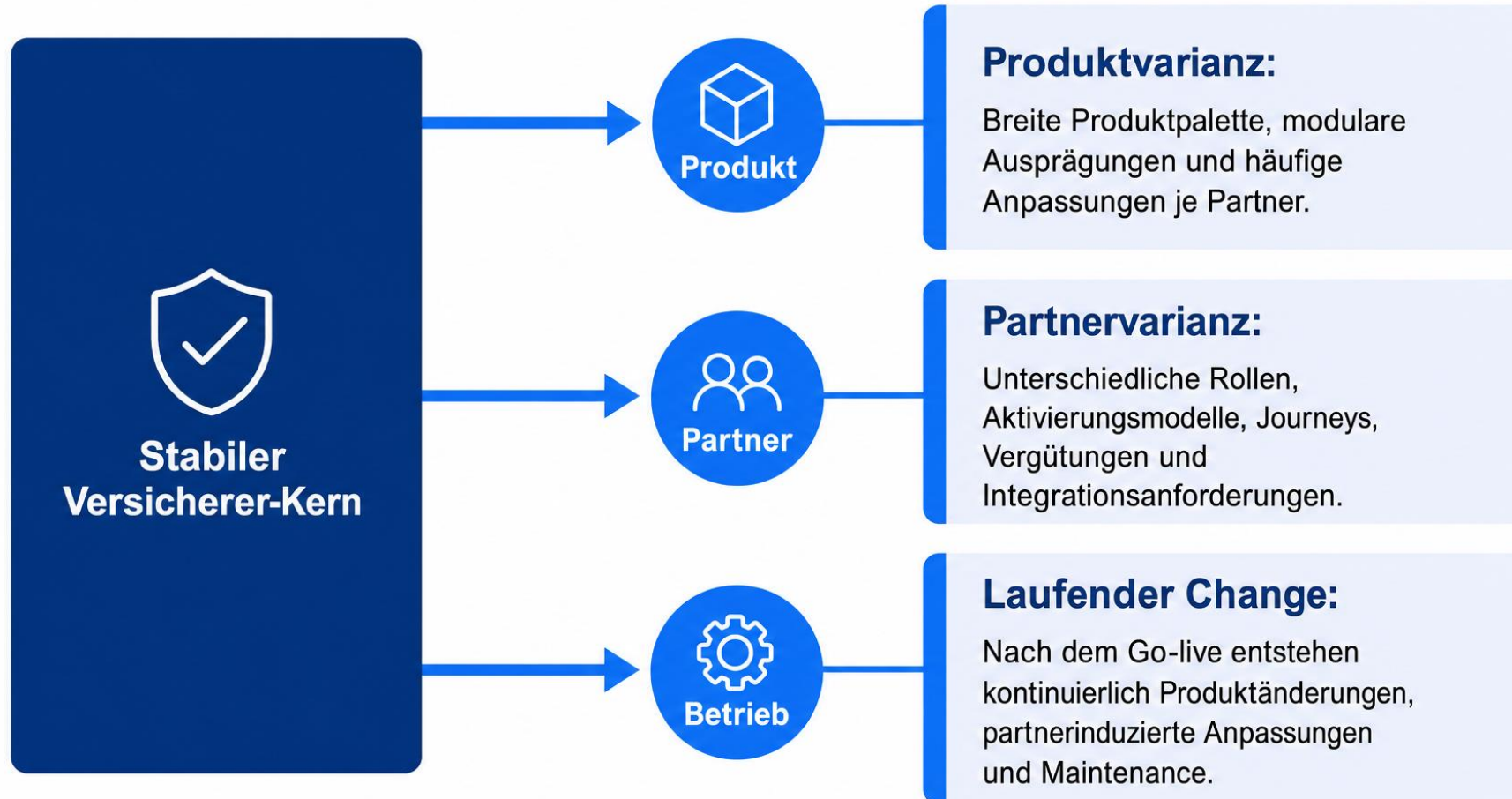


**Aus diesen fünf Geschäftsanforderungen folgt unmittelbar die Architekturfrage:
Was bleibt im Carrier-Kern und was gehört in eine variable Enabler-Schicht?**

Hinter diesem Anforderungsprofil stehen andere Change Economics als im klassischen Versicherungsvertrieb



Produkt-, Partner- und Betriebsvarianz



Wenige gekoppelte Geschäftsentscheidungen erzeugen schnell hohe Produkt- und Partnervarianz



Illustrativer Entscheidungsraum



Wenn jede Variante zum Versicherer-IT-Change wird, steigen Setup-, Integrations- und Folgekosten



Varianz im Core



Produktänderungen:

Neue oder geänderte Deckungen, Pricing-/UW-Regeln, Dokumente und Workflows erfordern Anpassungen in bestehenden Systemen.



Partneränderungen:

Individuelle Journeys, Datenmodelle, Datenaustausch und Reporting-Anforderungen erzeugen zusätzliche Integrationsarbeit je Partner.



Laufender Betrieb:

Maintenance, Regressionstests und spätere Änderungen vervielfachen den Aufwand über Produkte, Partner und Systeme hinweg.



Variieren

Produkt & Wertschöpfung variieren:
Unterschiedliche Produkte, Policenkonstrukte und Wertschöpfungstiefen schnell kombinieren.



Verbinden

Systeme flexibel verbinden:
Versicherer-Funktionalität unterschiedlich tief einbinden sowie heterogene Datenmodelle, Validierungen, Imports, Exports und APIs beherrschen.



Betreiben

Varianz kontrolliert betreiben:
Viele Varianten sicher, skalierbar, versioniert und auditierbar betreiben - ohne für jede Änderung neue Sonderlösungen aufzubauen.

Eine modulare Enabler-Architektur trennt häufige Varianz von stabileren Versicherer-Funktionen



Drei Architekturbausteine



AI-enabled Product Design macht aus verstreutem Kontext einen belastbaren Umsetzungsplan



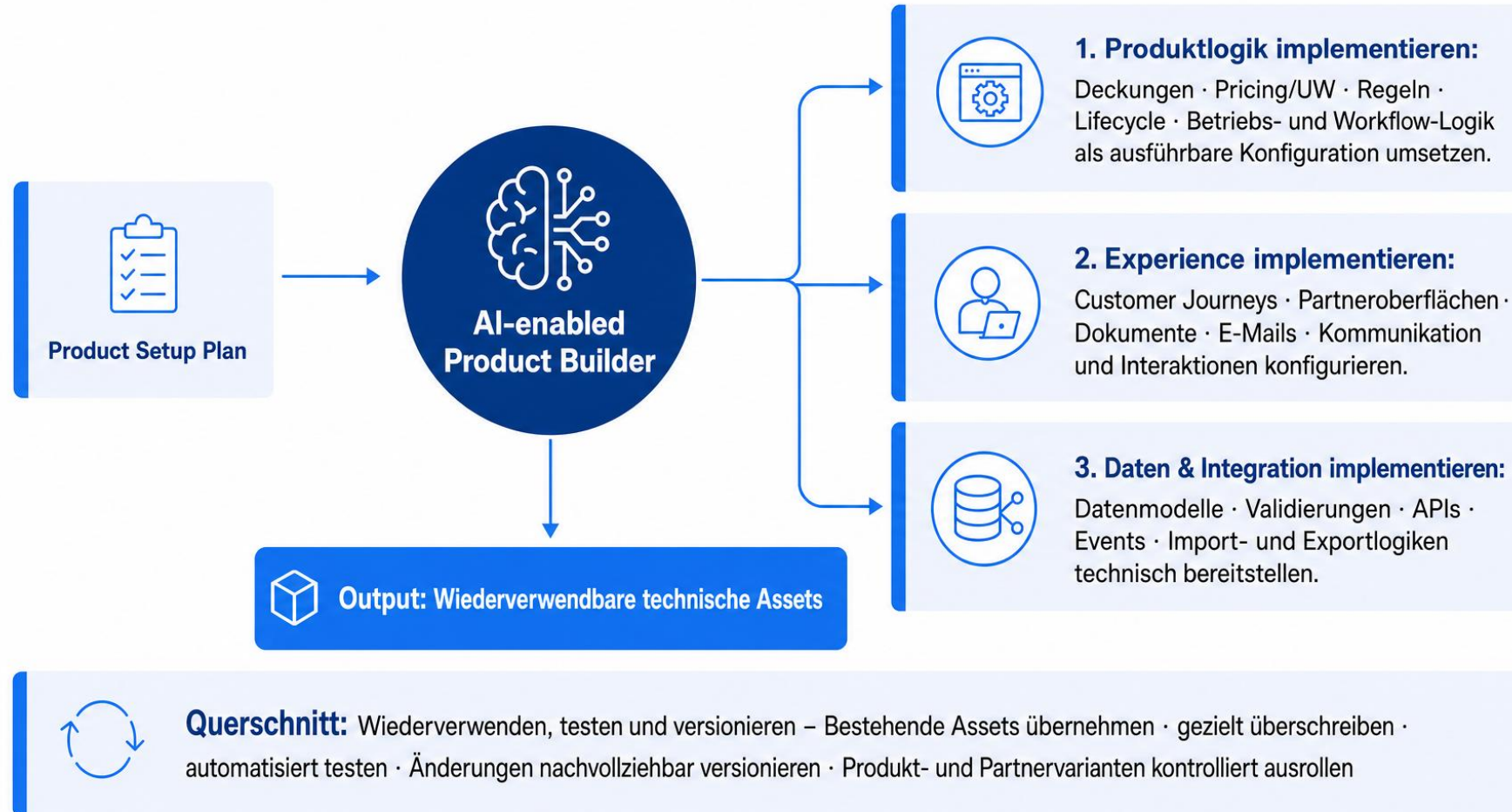
Vom Kontext zur Product Baseline



Der AI-enabled Product Builder setzt den Plan direkt in wiederverwendbare technische Assets um



Vom Plan zum Produkt



Ein explizites Mapping trennt Produkt- und Partnervarianz vom stabilen Kernvertrag



Variationsschicht, Mapping, Kernvertrag



Wo eine Funktion läuft, entscheidet sich fallweise aus Wiederverwendung, Kopplung und Varianzkosten



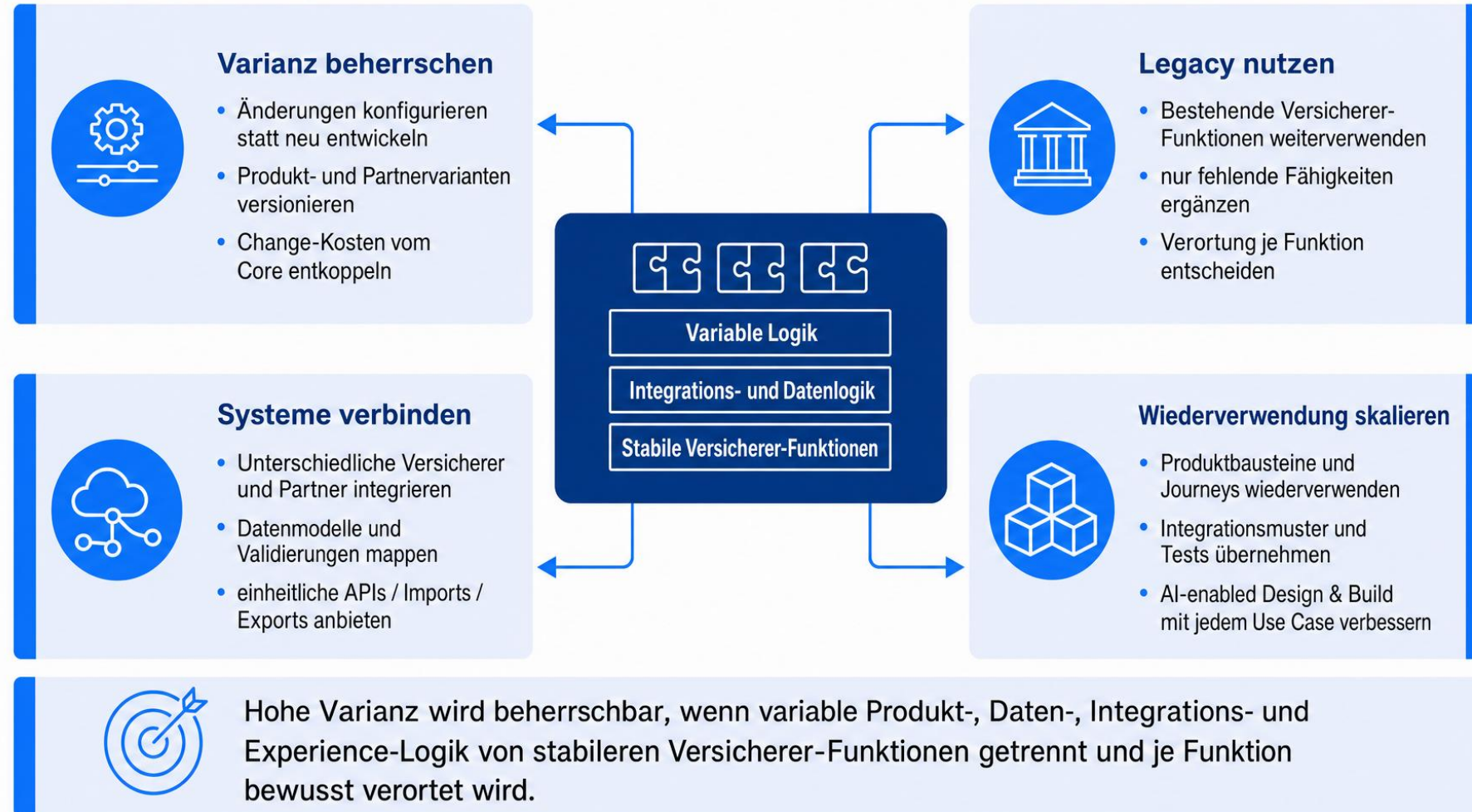
Verortung je Funktion



Die dedizierte Embedded-Architektur optimiert auf Varianz - ohne bestehende Versicherer-IT zu ersetzen



Warum der eigene Architekturpfad



Je mehr Interaktion digitalisiert wird, desto mehr wird Varianz zum Architekturproblem - nicht nur in Embedded



Digitale Varianz skaliert



Ein gut gewählter Use Case zeigt, ob Varianz schneller, günstiger und kontrollierter beherrschbar wird



Kontrollierter Test



Optionaler Proof: Live-Demo



Product Design → Implementierung



[Live Demo Öffnen](#)

Was Versicherer von InsurTechs lernen können, ist weniger Technologie als eine andere Logik des Veränderens



Vier strategische Learnings – unabhängig vom Zielbild



Vom Use Case nicht der Zielarchitektur starten

Nicht erst das perfekte Zielbild bauen und danach das Geschäft dazu suchen



Time-to-change statt Time-to-market

Der zweite Partner und die dritte Produktänderung entscheiden ökonomisch mehr als der erste MVP



Standardisieren und variieren

Standardisieren, was stabil ist. Variieren, was differenziert. Nicht für jeden Partner neu bauen



End-to-End-Verantwortung schaffen

Produkt, Distribution, Ops & IT auf ein Geschäftserg.. Partner braucht eine verantwortl. Ebene, kein Konsortium

Nicht übernehmen:

Core ersetzen · Greenfield-Programme · Technologie ohne Bestandsanbindung · Varianz, die niemand kommerziell verantwortet



Der eigentliche strategische Asset ist nicht die Plattform. Es ist die Fähigkeit, neue Versicherungsangebote schnell und zu niedrigen Grenzkosten zu starten und zu verändern.