

# Bestandsführung leicht gemacht

qx-Club Köln

Köln, 12. April 2016

Frank Richter

- ▶ Eigenschaften einer Bestandsführung
- ▶ Konzepte effizienter Produktmodellierung
- ▶ Strukturiertes Klassenmodell und spartenübergreifende Synergien
- ▶ Flexibilität im Rechenkern mit Markov-Modellen

- ▶ Maschinelle Unterstützung aller erforderlichen Geschäftsvorfälle
  - > Korrekte und vollständige Funktionalität, revisionssicher
- ▶ Automatische Versorgung aller Nachbarsysteme
  - > Richtige und vollständige Daten, revisionssicher
- ▶ Unterstützung aller relevanten Zugangswege
  - > GUI, Web, Batch, Dunkerverarbeitung
- ▶ Kosteneffiziente Wartung und Betrieb
  - > Skalierbarkeit, Sicherheit, Verfügbarkeit, Performance
- ▶ Zukunftsfähige Architektur
  - > Flexible Erweiterbarkeit und Anpassbarkeit, Time-to-Market
- ▶ Benutzerfreundliche Ergonomie und Anwendereffizienz
  - > Workflowanbindung, Benutzerführung

- ▶ Time-to-Market zu Lasten vollständiger Bearbeitungsfunktionen
  - > Lücken- oder fehlerhafte Verarbeitungsfunktionalität
  - > Viele Workarounds, Bestandsfehler als Ergebnis
- ▶ Schwerfällige, nur sehr aufwendig anpassbare Rechenkerne
  - > Schlechte Architektur, veraltete Entwicklungsmethodik
  - > Gleichzeitige Pflege mehrere Rechenkerne: für Bestandsführung, Angebotssystem, Disposition, ...
- ▶ Keine klaren Schnittstellen von Bestandsführung zu Nachbarsystemen
  - > Keine einfache Austauschbarkeit der „Komponenten“
  - > Selten Synergien durch gemeinsame Nutzung von „Querschnittsservices“
- ▶ Unzureichende Zeitmodellsteuerung
  - > Probleme mit Rückdatierungen in Bestandsführung und Schnittstellen sind nicht selten
- ▶ Erlernte Anwendereffizienz häufig eine (vermeintliche) Stärke
  - > Workflowanbindung / Benutzerführung sind „maßgeschneidert“ programmiert

- ▶ Die fachliche Architektur einer Bestandsführung ist eine Komponentenarchitektur.



- ▶ Die Modellierungssprache einer modernen Bestandsführung ist **durchgängig** objektorientiert.
- ⇒ Alle Komponenten der Facharchitektur werden auf Basis eines **einheitlichen Klassenmodells** erstellt.
- ⇒ Intelligente Strukturierung des Klassenmodells schafft spartenübergreifende Synergien.

- ▶ Produktmodellierung = Abbildung der produktabhängigen Verarbeitungslogik.  
⇒ **Hohe Anforderungen an Flexibilität und Anpassbarkeit!**

- ▶ Eigenschaften einer Bestandsführung
- ▶ Konzepte effizienter Produktmodellierung
- ▶ Strukturiertes Klassenmodell und spartenübergreifende Synergien
- ▶ Flexibilität im Rechenkern mit Markov-Modellen

## ► Gedankenexperiment:

- > Wenn ein neues Produkt eingeführt wird und man beim Programmieren einer Bearbeitung feststellt, dass jedes Code-Fragment schon einmal an irgendeinem anderen Produkt programmiert wurde, so will man dieses Fragment einfach wiederverwenden können (**ohne es zu kopieren!!!**).
- ⇒ Wenn dies für alle Bearbeitungen des neuen Produktes zutrifft, müssen die Bearbeitungen für das neue Produkt modellierbar sein **ohne eine einzige Zeile Code implementieren zu müssen**.

## ► Ableitbare Prinzipien:

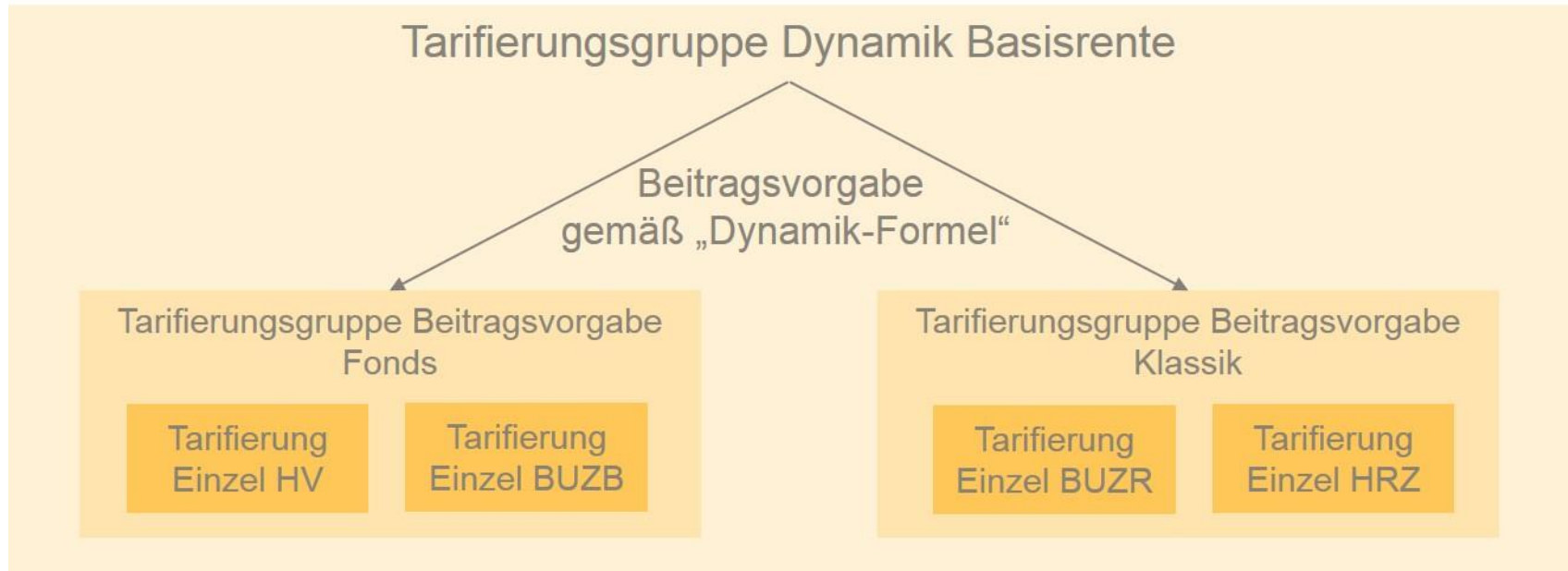
- > Gleiches wird nur einmal implementiert
- > Unterschiedliches ist leicht austauschbar
- > Versicherungsmathematische Fragestellungen werden für ähnliche Bearbeitungen einheitlich beantwortet (Abschlusskosten, Stornoabschläge, Gebühren, ...)

## ► Vorgehen:

- > Zerlegen einer Tarifierungs-Aufgabe in bereits gelöste Teilaufgaben
- > Zerlegen einer Teilaufgabe in wiederverwendbare Arbeitsschritte (einer Tarifierung)

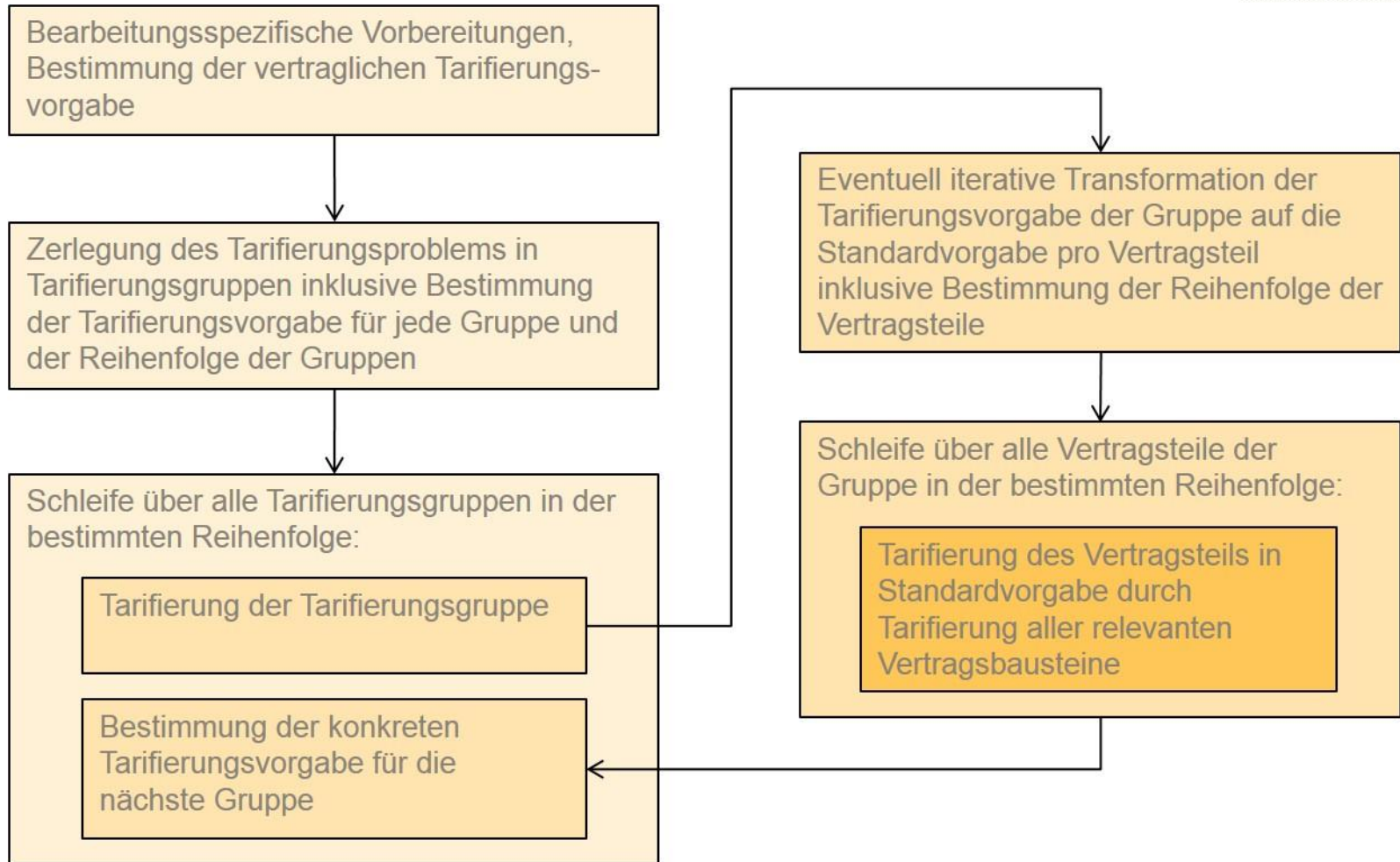


- ▶ Beispiel: Dynamik fondsgebundene Basisrente mit BUZ und HRV
  - ⇒ Auch nach Dynamik muss gelten:  $B_{HV+BUZB} > B_{BUZR+HRZ}$
  - ⇒ „Dynamik-Formel“ liefert Beitragsvorgaben für HV+BUZB und BUZR+HRZ



- ⇒ Die Dynamik einer fondsgebundenen Basisrente wird in zwei bereits bekannte Tarifierungsaufgaben zerlegt





- ▶ Enthält Tarifierungsgruppen für alle

## Tarifierungskontexte

und alle

## Tarifierungsarten

(Beispiel)

(Beispiel)

- Beitragsvorgabe brutto/netto, gesamtvertraglich/pro VT-Gruppe
- Leistungsvorgabe brutto/netto, relativ/absolut/implizit
- Weiterverwendung Beiträge der Zusatzversicherungen
- „BUZ-Optimierer“ bei Basisrenten
- Dynamik
- Dynamik Basisrente
- Beibehaltung Leistungsverhältnisse
- ...

- Änderung Beitrag/Leistung
- Änderung Dauern
- Änderung Preisstufe
- Änderung Risikodaten
- Änderung Zahlweise
- Ausschluss Zusatzversicherung
- Dynamik
- Einschluss Zusatzversicherung
- Erhöhung Gesamtbeitrag
- Fortsetzung Dynamik
- Policierung
- Reduktion Gesamtbeitrag
- Tarifierung
- WIK nach Beitragsfreistellung
- ...

- ▶ Zerlegt jede „einfache“ Tarifierungsaufgabe in Tarifierungsschritte, abhängig von der Tarifierungsart – Beispiel:

## *Policierung*

Tarifierung Vertrag AbgleichAblaufTermine  
Tarifierung Vertrag AbgleichAblaufTermineDynamik  
Tarifierung Vertrag JahrestagAblaufVersicherung  
Tarifierung Vertrag BerechnenStandard  
Tarifierung Vertrag NachbearbeitungVertragsteileStandard  
Tarifierung Vertrag ErzeugenDynamikvereinbarungEnde67J11M  
Tarifierung Vertrag EinrichtenSparplan  
Tarifierung Vertrag EinrichtenPortfeuilleStandardFRV  
Tarifierung Vertrag EinmalbeitragZerlegen  
Tarifierung Vertrag InitialenInhaltBerechnen  
Tarifierung Vertrag ErzeugenRueckversicherungsverhaeltnisse  
Tarifierung Vertrag BerechnenNaechsteDynamikStandard

## *Dynamik*

Tarifierung Vertrag AkkumulierenLeistungBausteineRente  
Tarifierung Vertrag BerechnenDynamikStandard  
Tarifierung Vertrag NachbearbeitungVertragsteileStandard  
Tarifierung Vertrag InitialenInhaltBerechnen  
Tarifierung Vertrag BerechnenNaechsteDynamikStandard

## *Vertragsübernahme*

Tarifierung Vertrag AbgleichAblaufTermine  
Tarifierung Vertrag AbgleichAblaufTermineDynamik  
Tarifierung Vertrag JahrestagAblaufVersicherung  
Tarifierung Vertrag VertragExternAusVorgabe  
Tarifierung Vertrag VertragsuebernahmeStandard  
Tarifierung Vertrag NachbearbeitungVertragsteileStandard  
Tarifierung Vertrag ErzeugenDynamikvereinbarungEnde67J11M  
Tarifierung Vertrag EinrichtenSparplan  
Tarifierung Vertrag EinrichtenPortfeuilleStandardFRV  
Tarifierung Vertrag EinmalbeitragZerlegenVertragsuebernahme  
Tarifierung Vertrag InitialenInhaltBerechnen

## *Änderung Beitrag/Leistung*

Tarifierung Vertrag BerechnenStandard  
Tarifierung Vertrag NachbearbeitungVertragsteileStandard  
Tarifierung Vertrag InitialenInhaltBerechnen

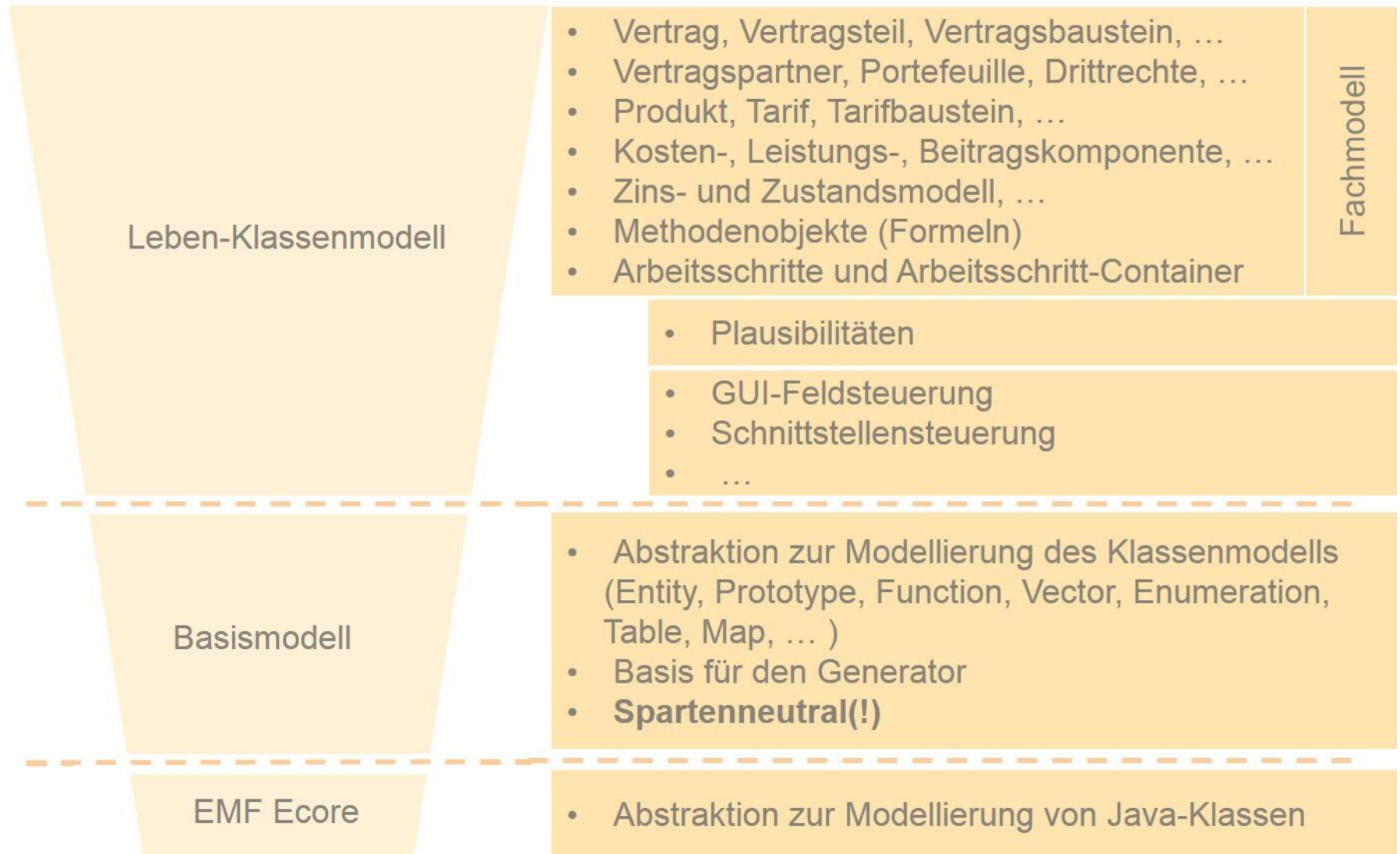
## *Switch*

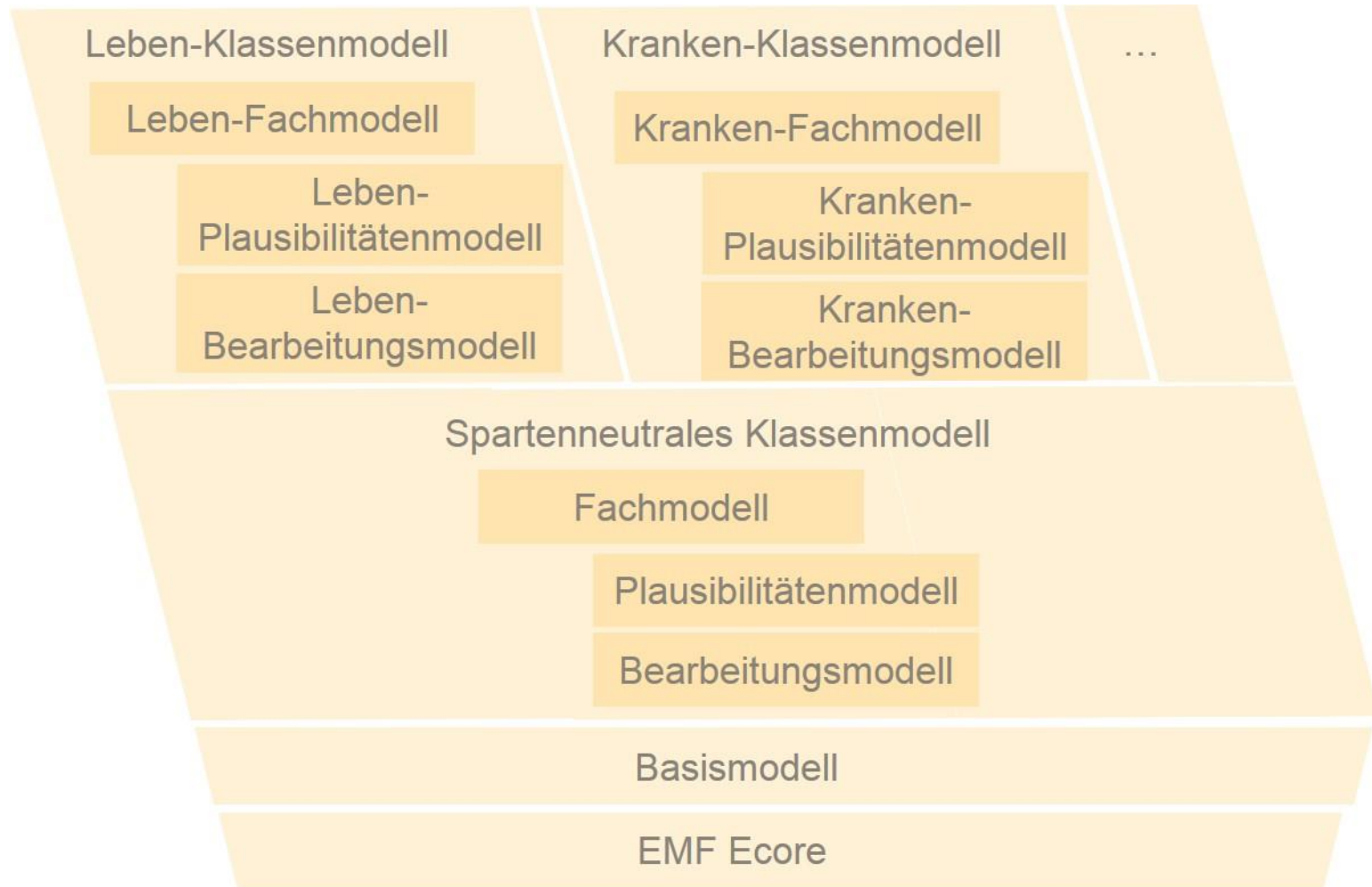
Tarifierung Vertrag EinrichtenSparplan  
Tarifierung Vertrag EinrichtenPortfeuilleStandardFRV  
Tarifierung Vertrag BereinigenPortefeuilleStandard

- ▶ Eigenschaften einer Bestandsführung
- ▶ Konzepte effizienter Produktmodellierung
- ▶ **Strukturiertes Klassenmodell und spartenübergreifende Synergien**
- ▶ Flexibilität im Rechenkern mit Markov-Modellen

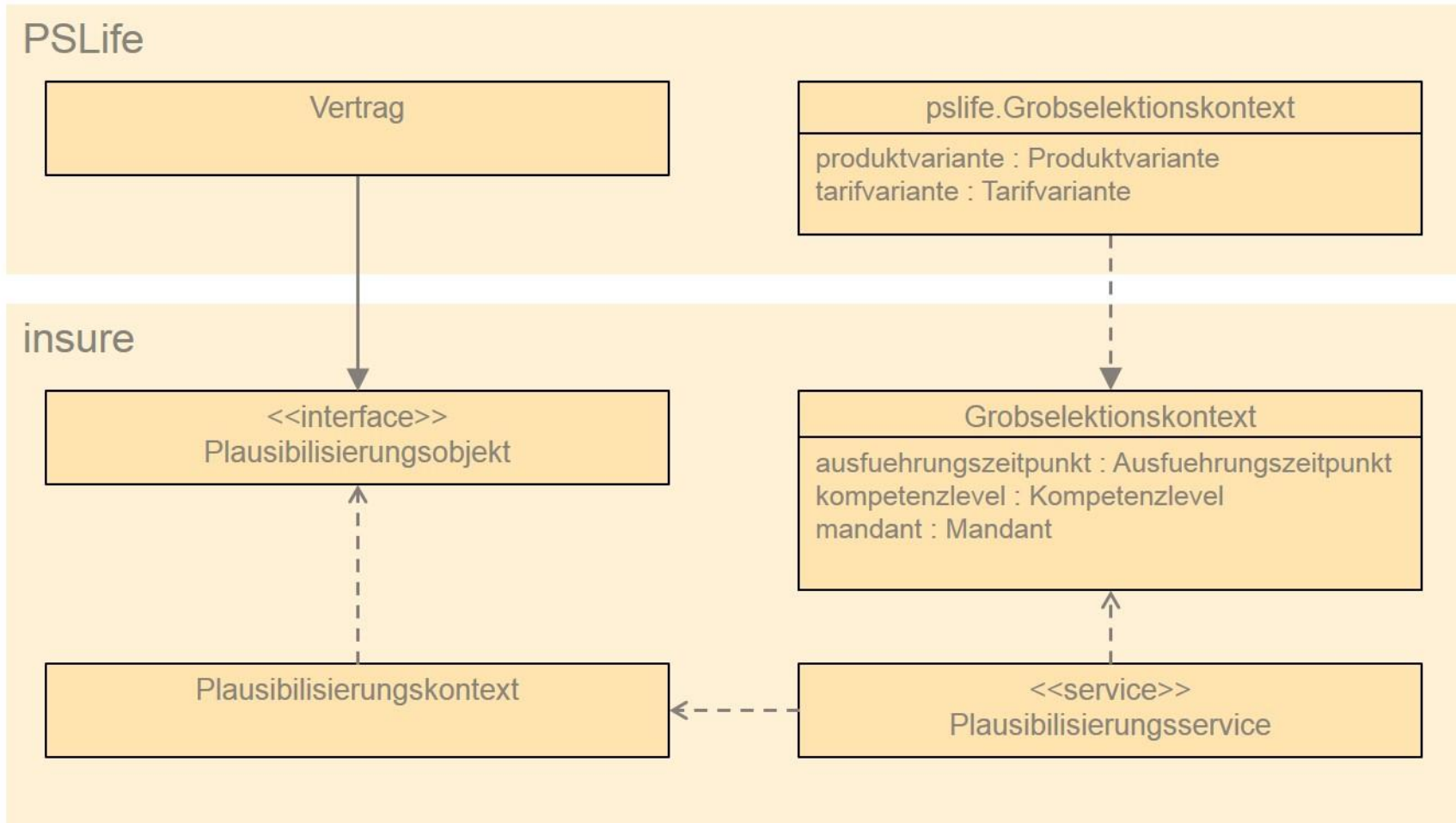
- ▶ Modellierung der Vertragsbearbeitungen => sehr viele Klassen (Methodenobjekte)
- ▶ Weitere (relativ viele) Produktbausteine durch strukturelle Elemente
  - > Kosten-, Zins- und Zustandsmodelle; Beitrags-, Leistungs-, Kosten- und Überschusskomponenten; Abschlags- und Rabattsysteme; Kalkulationsgrundlagen; ...
- ⇒ Verwaltungswerkzeug wird benötigt, das aus Klassen-Metamodell **generierbar** ist
- ⇒ Eclipse als IDE und Eclipse Modeling Framework (EMF) für die Modellierung
  - ⇒ „Modellierungskomponente“ für Produktdefinition wird mit EMF-Bordmitteln generiert
- ▶ Aber: Zur Laufzeit keine Abhängigkeit vom Modellierungswerkzeug!
- ⇒ Ein Generator wird benötigt, der alle vom Modell und von den Modellinstanzen abhängigen Java-Klassen erzeugen muss!
- ▶ Generator muss aber unabhängig sein von den konkreten Klassen des Modells!
- ⇒ Klassenmodell muss strukturiert werden!



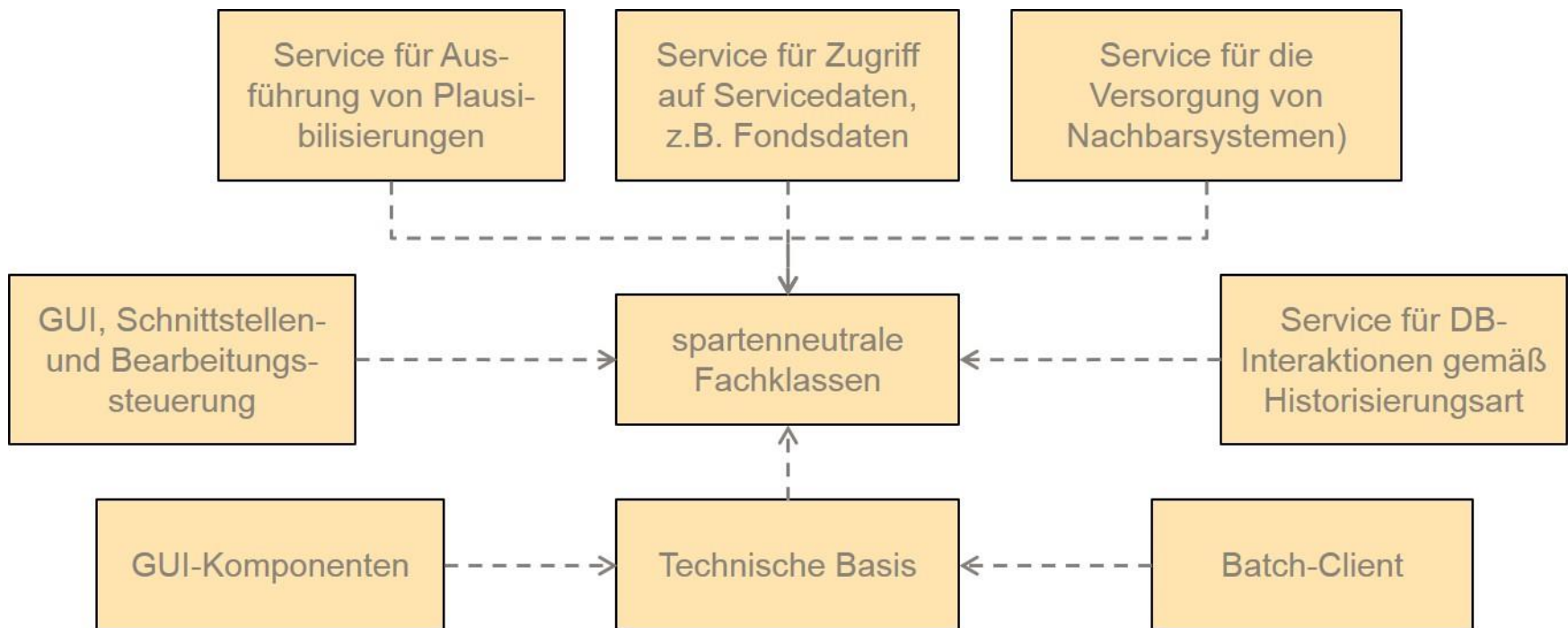








- ▶ Einheitlicher, hierarchischer Aufbau der Klassenmodelle auf **gleicher Basis**
  - > Gleicher Generator, gleiche Modellierungstechnik, identischer Entwicklungsprozess
- ⇒ Erlaubt die Implementierung gemeinsam nutzbarer Services in spartenneutralen „Querschnittskomponenten“

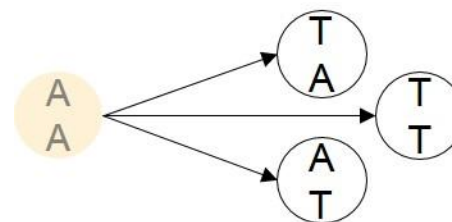


- ▶ Eigenschaften einer Bestandsführung
- ▶ Konzepte effizienter Produktmodellierung
- ▶ Strukturiertes Klassenmodell und spartenübergreifende Synergien
- ▶ Flexibilität im Rechenkern mit Markov-Modellen

- ▶ Ein Lebensversicherungstarif wird abstrahiert durch eine endliche Menge versicherungstechnischer Zustände  $Z = \{\alpha, \beta, \gamma, \dots\}$  (Hoem, 1969)

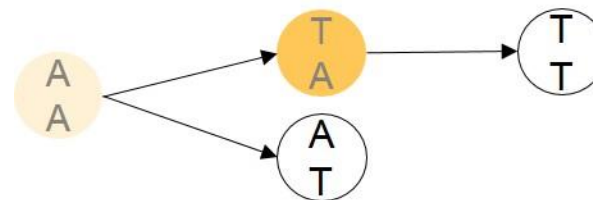
- ▶ Risikoversicherung auf zwei Leben:

>  $Z = \{AA, AT, TA, TT\}$



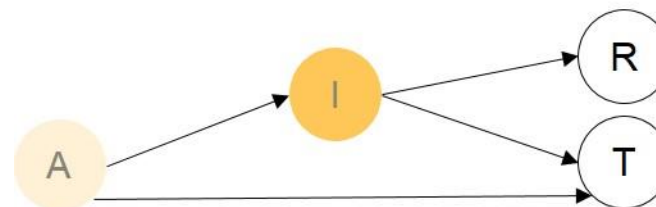
- ▶ Hinterbliebenenrente:

>  $Z = \{AA, AT, TA, TT\}$



- ▶ Berufsunfähigkeit (Standardmodell):

>  $Z = \{A, I, T, R\}$



- Modell führt zu einer Rekursionsformel für die Reserve:

$$V_{\alpha}^{+}(t) = b_{\alpha}(t) + v(t, t^{+}) \cdot \sum_{\beta \neq \alpha} p_{\alpha\beta}(t^{+}) \cdot [b_{\alpha\beta}(t^{+}) + V_{\beta}^{+}(t^{+})] \\ + v(t, t^{+}) \cdot \left( 1 - \sum_{\beta \neq \alpha} p_{\alpha\beta}(t^{+}) \right) \cdot V_{\alpha}^{+}(t^{+})$$

$$b_{\alpha}(t) = b_{\alpha}^{K}(t) + b_{\alpha}^{R}(t) - \pi_{\alpha}(t)$$

$b_{\alpha}(t)$  ... Zahlung (= Kosten + Rente – Beitrag) im Zustand  $\alpha$

$b_{\alpha\beta}(t)$  ... Zahlung (= Einmalleistung) bei Übergang von  $\alpha$  nach  $\beta$

$v(t, t^{+})$  ... Diskontierung mit dem Rechnungszins

$p_{\alpha\beta}$  ... Übergangswahrscheinlichkeit von  $\alpha$  nach  $\beta$

- ▶ Es genügt **eine(!!!)** Formel zu implementieren statt **vieler Barwerte**
  - ⇒ Markov-Ansatz ist nur auf Ebene „Reserve“ effizient, nicht auf Barwert-Ebene
  - ⇒ Kosteneinsparung bei Einführung von Produkten mit (teilweiser) klassischer Mathematik
  
- ▶ Modellierung ist pragmatisch und intuitiv:
  - > Kosten- und Leistungen müssen vorgegeben werden ...
  - > ... sowie das konkrete Zustandsmodell und das zu verwendende Zinsmodell
  - > Für Kosten und Leistungen kann man vorgeben:
    - Höhe und Fälligkeit der Zahlung
    - Beginn und Ende einer periodischen Zahlung
  
- ▶ Völlig neue Produkte möglich:
  - > Zum Beispiel Homogenisierung Klassik und Fonds:
    - Gleichartige Todesfallleistungen für klassische und fondsgebundene Versicherungen
    - Bedarfsgerechte Erlebensfallgarantien auch für fondsgebundene Versicherungen

- ▶ Eine fundamentale Anwendung der Rekursionsformel:

$$H_t(\text{Beitrag, Leistung}) := V_\alpha^+(t)$$

$$F_t(\text{Beitrag, Leistung, Reserve}) := H_t(\text{Beitrag, Leistung}) - V_\alpha^+(t)$$

⇒ Tarifierung zum Zeitpunkt t bedeutet, die Nullstelle der Funktion  $F_t$  zu finden

⇒ Genau ein Verfahren **für alle(!!!)** Tarifierungsprobleme

- ▶ Früher mussten viele Beitragsberechnungsformeln implementiert werden:

$$B_{x,n,t}^{(\zeta)} = (1 + \zeta) \cdot \frac{E_{x,n,t} + \Gamma_{x,n,t}}{(1 - \beta) \cdot \ddot{a}_{x,t}^{aa} - \frac{\alpha_0 \cdot t}{t^{AK}} \cdot \ddot{a}_{x,t}^{aa}}$$

$$B = \frac{P + \alpha_1 + \gamma_1 + \gamma_2 \cdot \frac{a_{x,n}}{a_{x,t}}}{1 - \beta_1 - \alpha_2}$$

$${}_K B_{x,n} = \frac{{}_K P_{x,n}^N + \alpha^{\gamma_2}}{1 - \beta - \alpha^{\gamma_1} - \frac{\alpha^z \cdot t_{\max}}{a_{x,n}} - \frac{\alpha^\alpha \cdot t_{\max}}{a_{x,n}}} \quad \text{mit } {}_K P_{x,n}^N = \frac{|_n A_x}{a_{x,n}}$$

$$B_{x,t,k} = \begin{cases} \frac{B(E)_{x,t} \cdot (1 + \gamma_2) \cdot (a_{k|}^{(12)} + {}_k a_{x+n}^{(12)})}{1 - B(T_1)_{x,t}} & \text{für } k > 0 \\ \frac{B(E)_{x,t} \cdot (1 + \gamma_2) \cdot a_{x+n}^{(12)}}{1 - B(T_1)_{x,t}} & \text{für } k = 0 \end{cases}$$

und

$$B(E)_{x,t} = \frac{{}_n E_x + \gamma_1' \cdot a_{x,t} + \gamma_2' \cdot ({}_t a_x - {}_n a_x)}{(1 - \alpha_\beta - \beta) \cdot a_{x,t} - \alpha_t \cdot a_{x,t+1,A}}$$

$$B(T_1)_{x,t} = \frac{{}_{/t,n} \tilde{A}_x + t \cdot \gamma_1'' \cdot a_{x,t} + t \cdot \gamma_2'' \cdot ({}_t a_x - {}_n a_x)}{(1 - \alpha_\beta - \beta) \cdot a_{x,t} - \alpha_t \cdot a_{x,t+1,A}}$$