

Architekturansatz für dynamische Anwendungen



Markus Schärtel
JavaForum 2010

About...

Markus Schärtel

< 40 Jahre

1 Kind

Bodensee

markus.schaertel@innovations.de

Software-Architektur

Business Rules

Produktleitung



Was ist eine dynamische (Business-)Anwendung?

„A software system that embodies a business process and is built for change, adaptable to business context, and information rich.“

Quelle: „The Dynamic Business Applications Imperative“, Forrester Research Inc., 2007

Was ist eine dynamische (Business-)Anwendung?

Komponenten-orientiert
statt monolithische
Strukturen

Automatisierte und
manuelle Aktivitäten,
Aufgaben und Entscheidungen
mit Wertschöpfung

„A **software system** that embodies a
business process and is built for
change, adaptable to **business context**,
and **information rich**.“

Quelle: „The Dynamic Business Applications Imperative“, Forrester Research Inc., 2007

Daten, Inhalte, Visualisierung,
Bilder, Ton, Video, „physical
Feedback“, etc.

Abläufe und Informationen
mit Bezug zu Personen und
Rollen, Zeit, Ort, Geschäfts-
zielen, etc.

Aspekte einer typischen Anwendung

Transaktionssicherheit?

Persistenz?

Umsysteme?

Externe
Datenquellen?

Batch?

Security?

Skalierbarkeit?

Simulation?

Auditierbarkeit?

Reports?

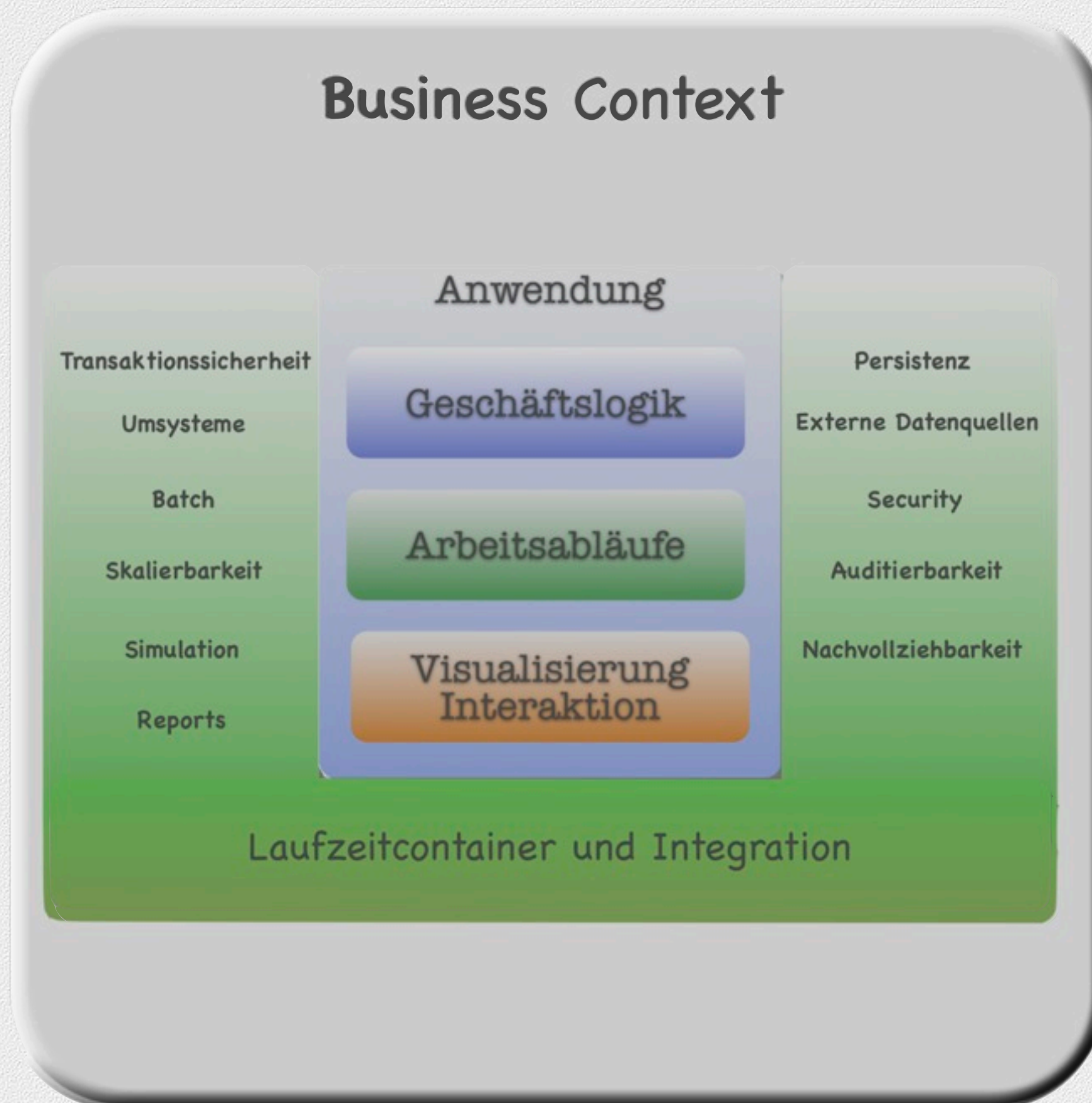
Nachvollziehbarkeit?



Anwendungen benötigen Container...



Anwendungen laufen im Kontext...



...den der Markt definiert

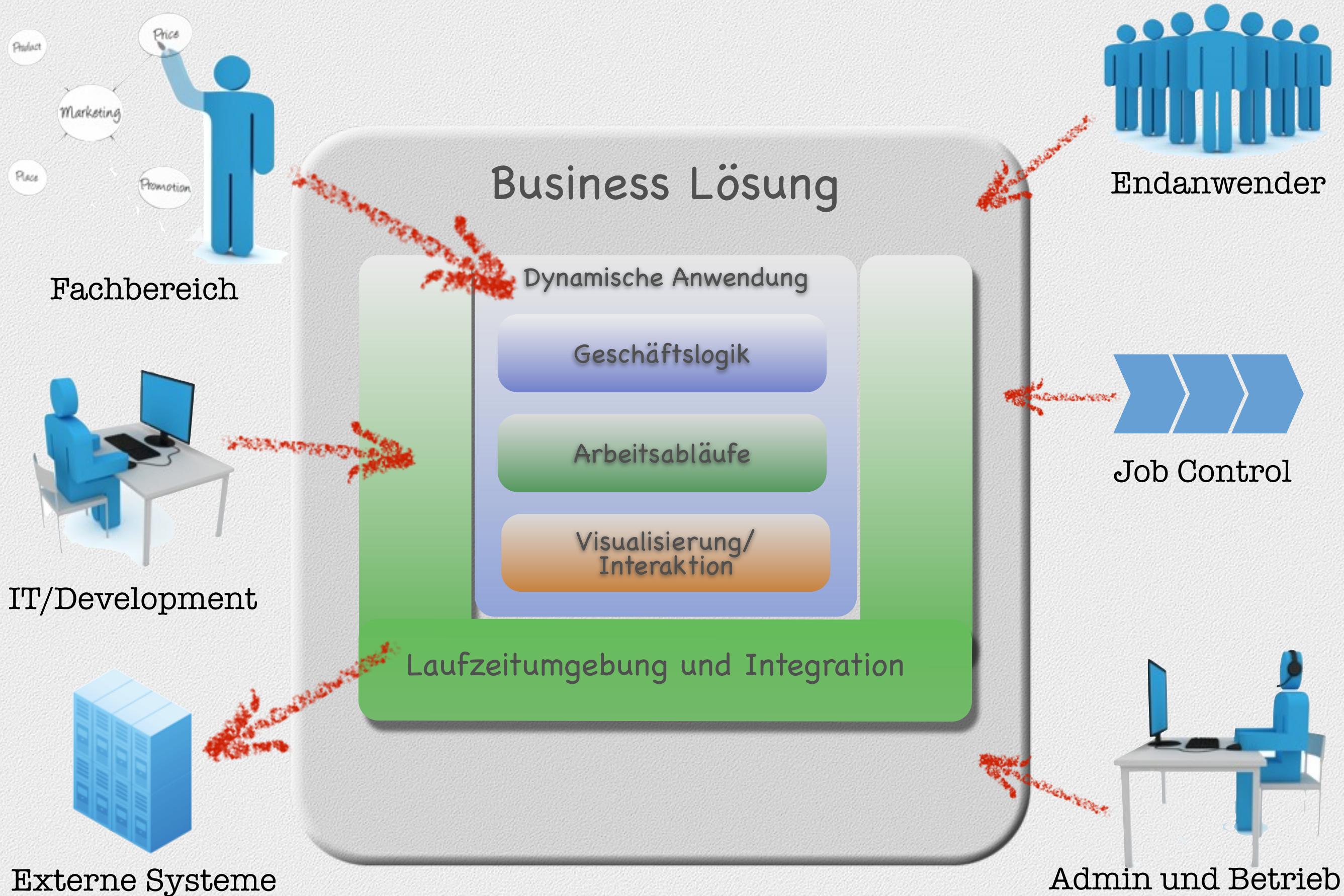
Markt



Business Context



Dynamische Anwendung im Kontext



Ideale Business-Welt

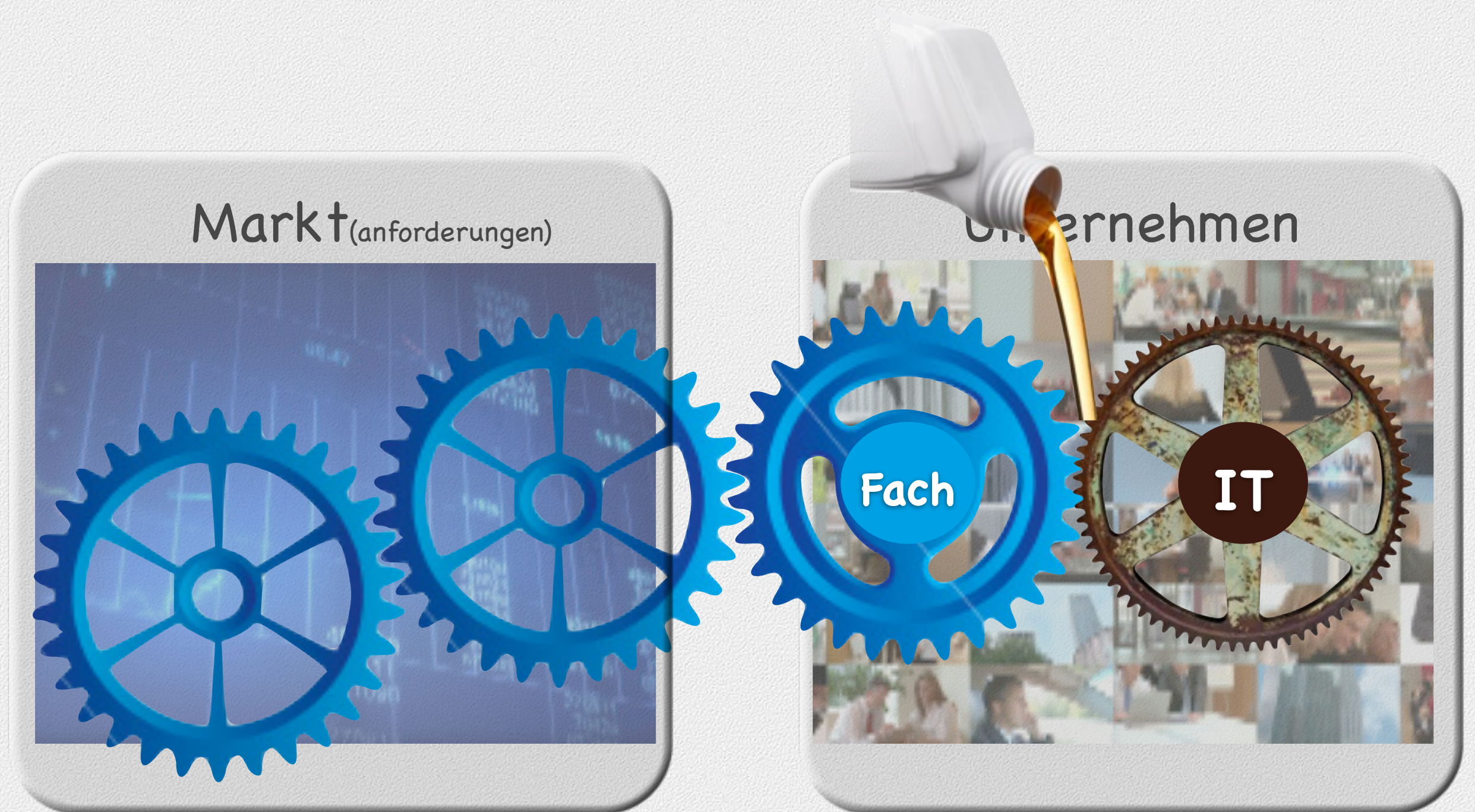
Markt_(anforderungen)



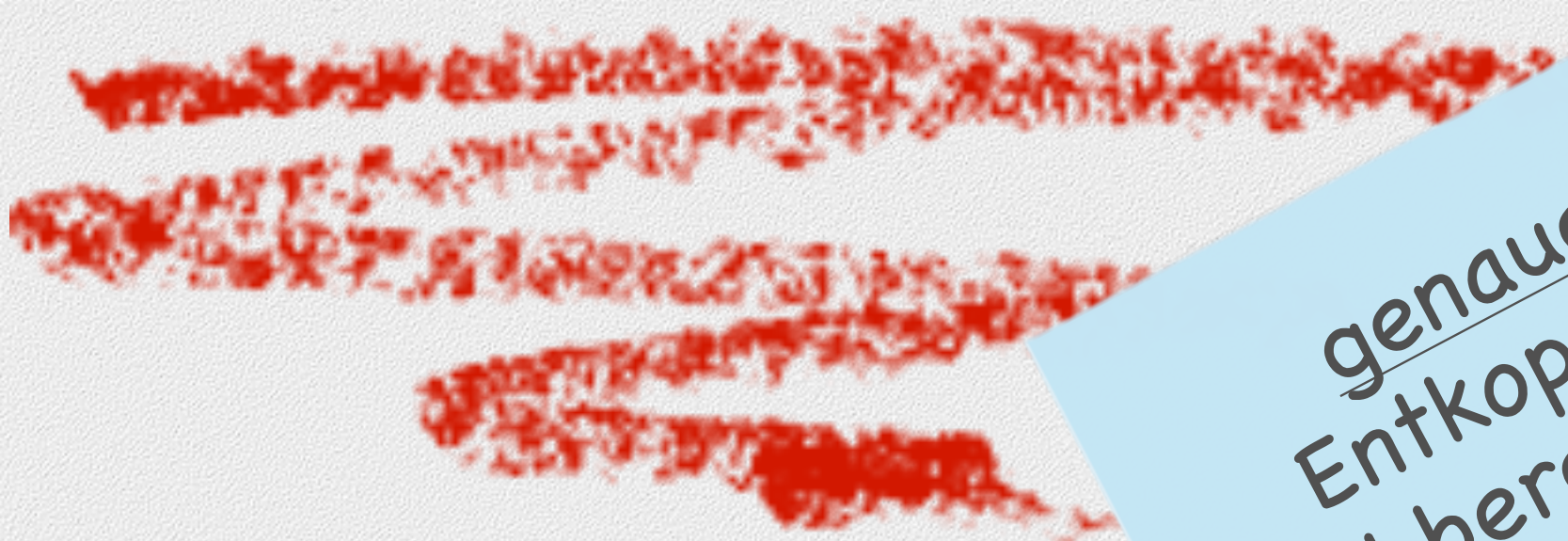
Unternehmen



Gefühlte Realität



Dynamische Anwendungen == Entkopplung

Three horizontal red scribbles, resembling paint strokes or ink marks, are located in the middle-left portion of the slide.

genauer:
Entkopplung
Fachbereiche und
IT-Abteilungen

Herausforderung



Domäne

IT



```
... = connection.getConnection()  
... = connection.createStatement()  
... = statement.executeQuery()  
... = resultSet.next()
```



Notwendig: Aufgabenteilung Fachbereich und IT



Aufgabenteilung Fachbereich/IT



Business process

Anwendung

Geschäftslogik

Arbeitsabläufe

Visualisierung
Interaktion



Business process

Laufzeitumgebung und Integration

(Neue) Aufgaben des Fachbereichs

Definition
Business Objektmodell

Modellierung
der Fachlogik



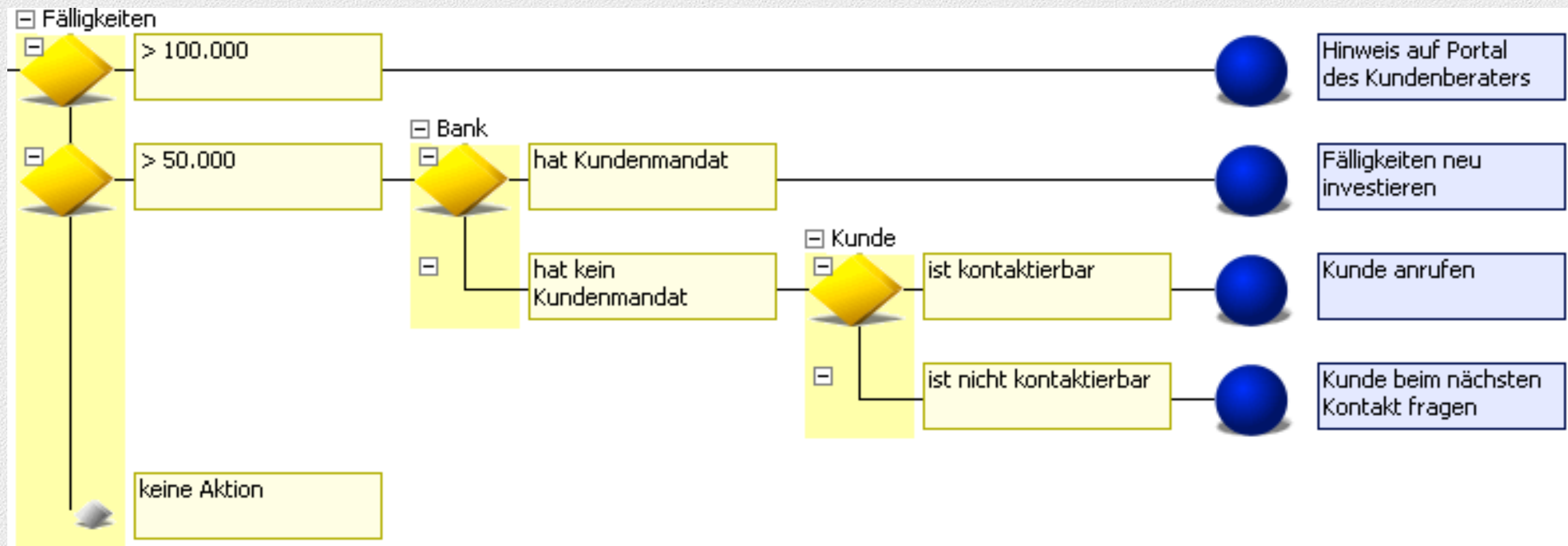
Modellierung der
Ablauflogik

Modellierung von
Layouts

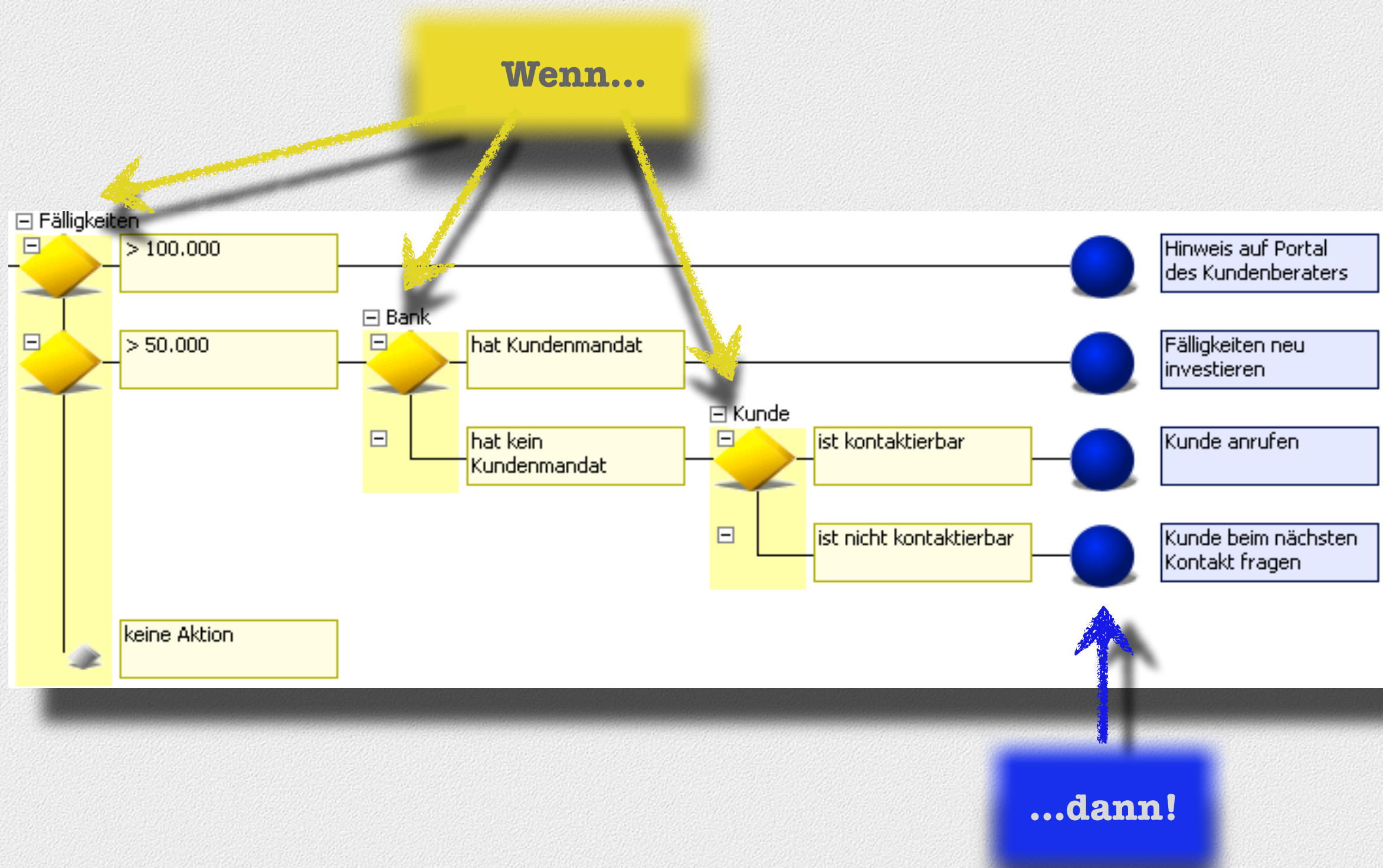
Tooling für den Fachbereich



Kurzer Ausflug: Business Rules



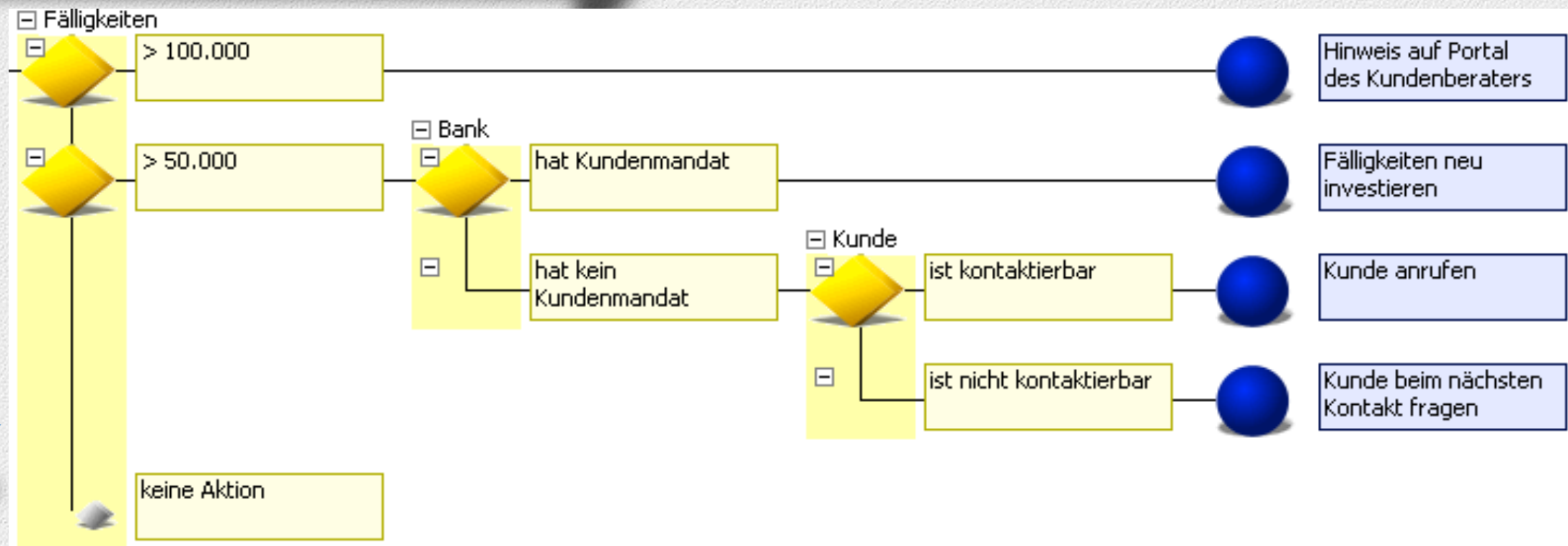
Kurzer Ausflug: Business Rules



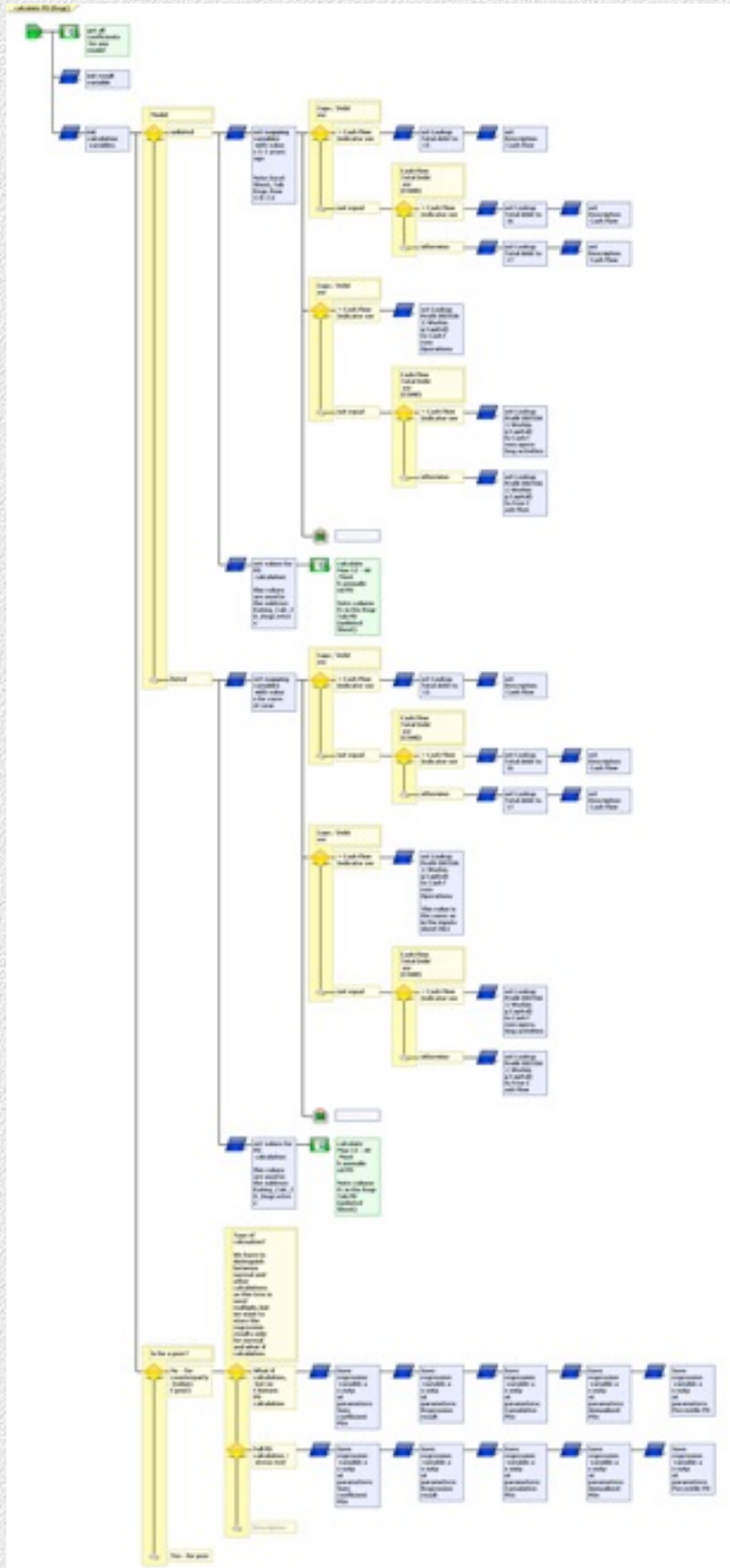
Kurzer Ausflug: Business Rules

von links nach rechts

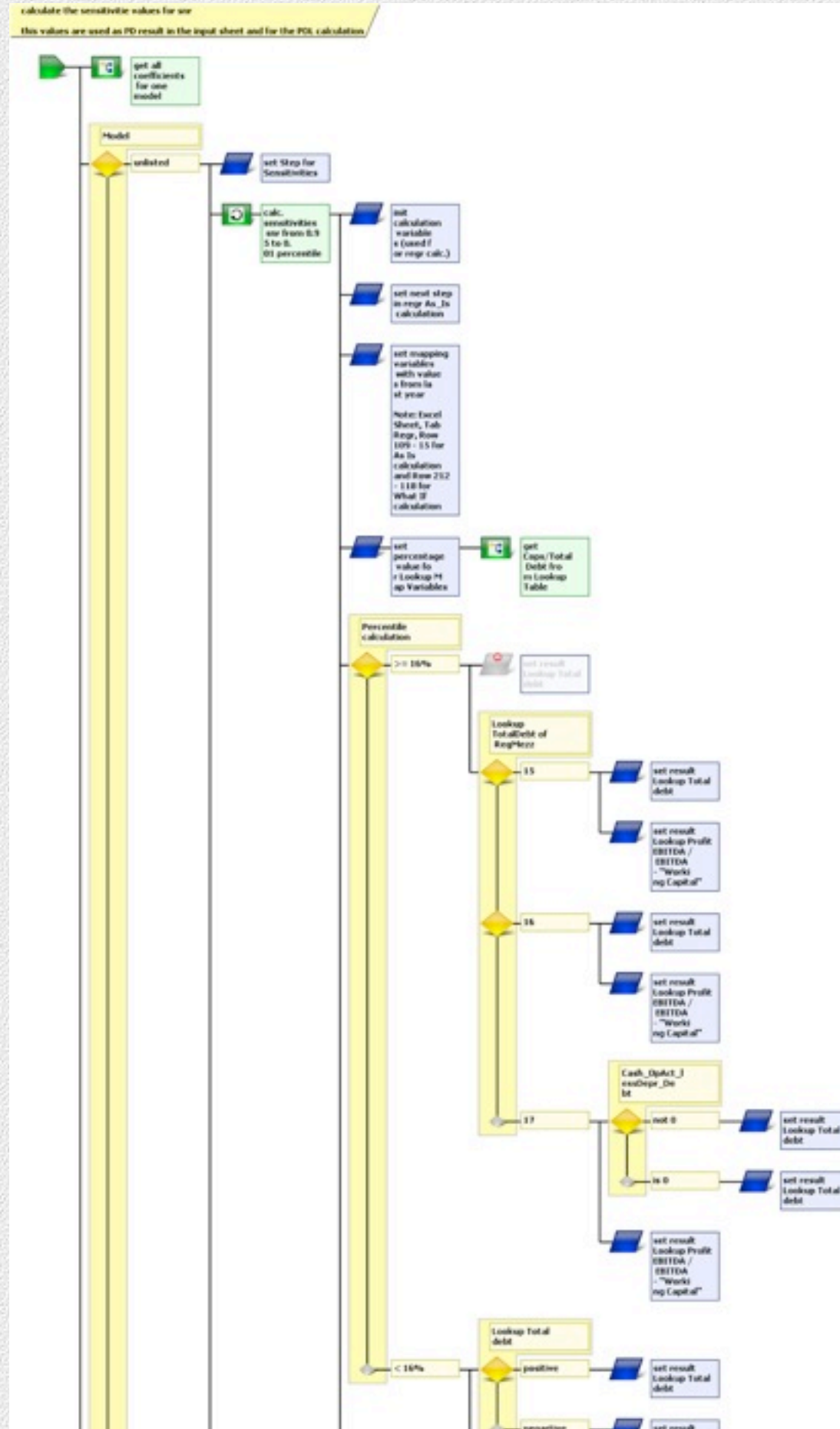
unten nach oben von

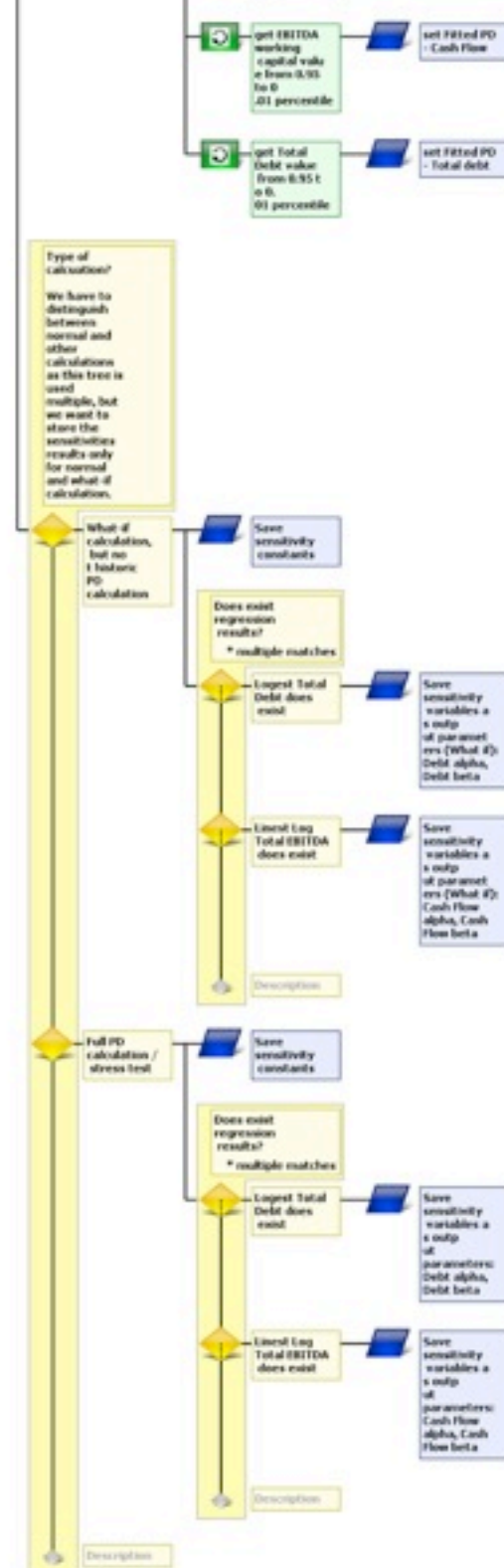


Regelmodelle werden groß...



- Hinweis auf Portal des Kundenberaters
- Fälligkeiten neu investieren
- Kunde anrufen
- Kunde beim nächsten Kontakt fragen





40 Seiten Spezifikation !!

Regelgesteuerte Ablauflogik

○ Zustände



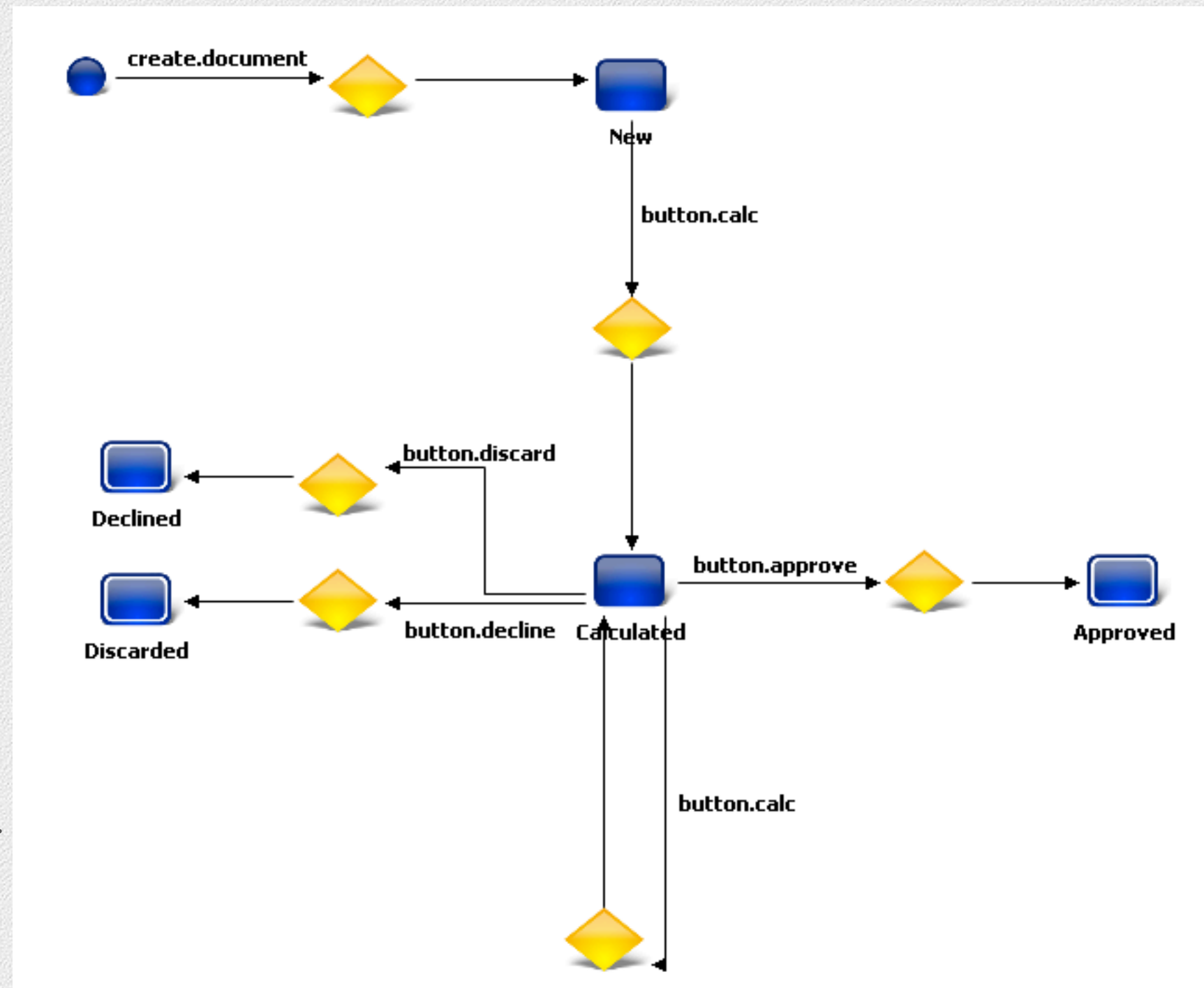
○ Events



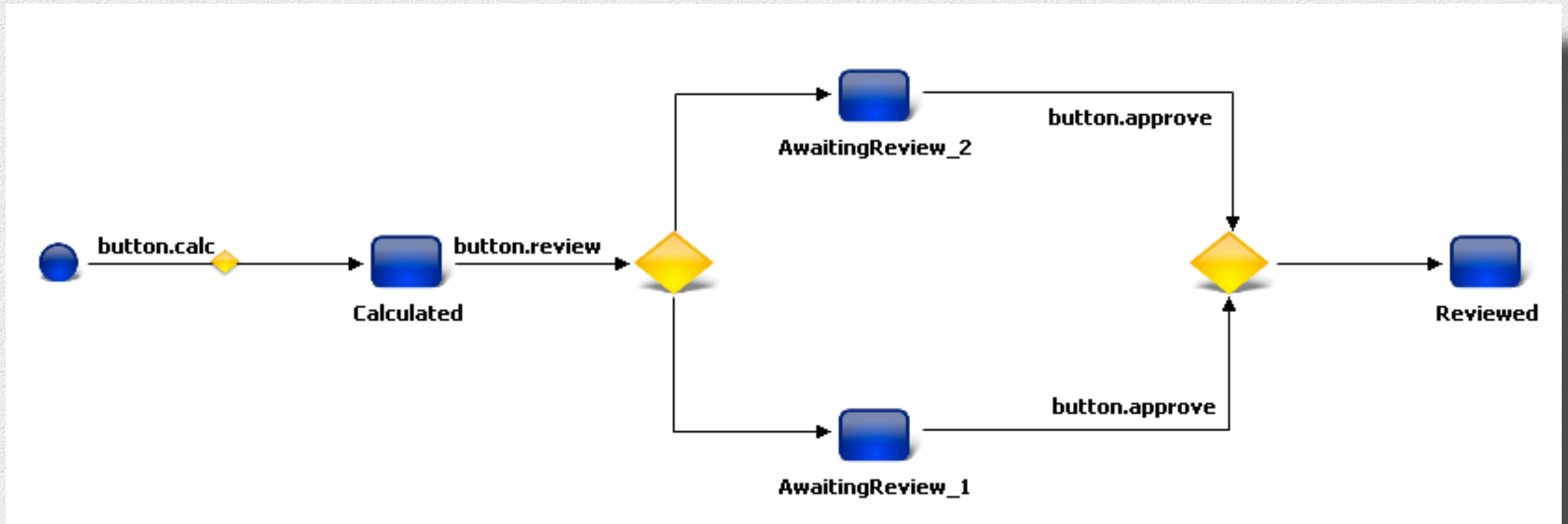
○ Transitions



○ Mediator Regeln



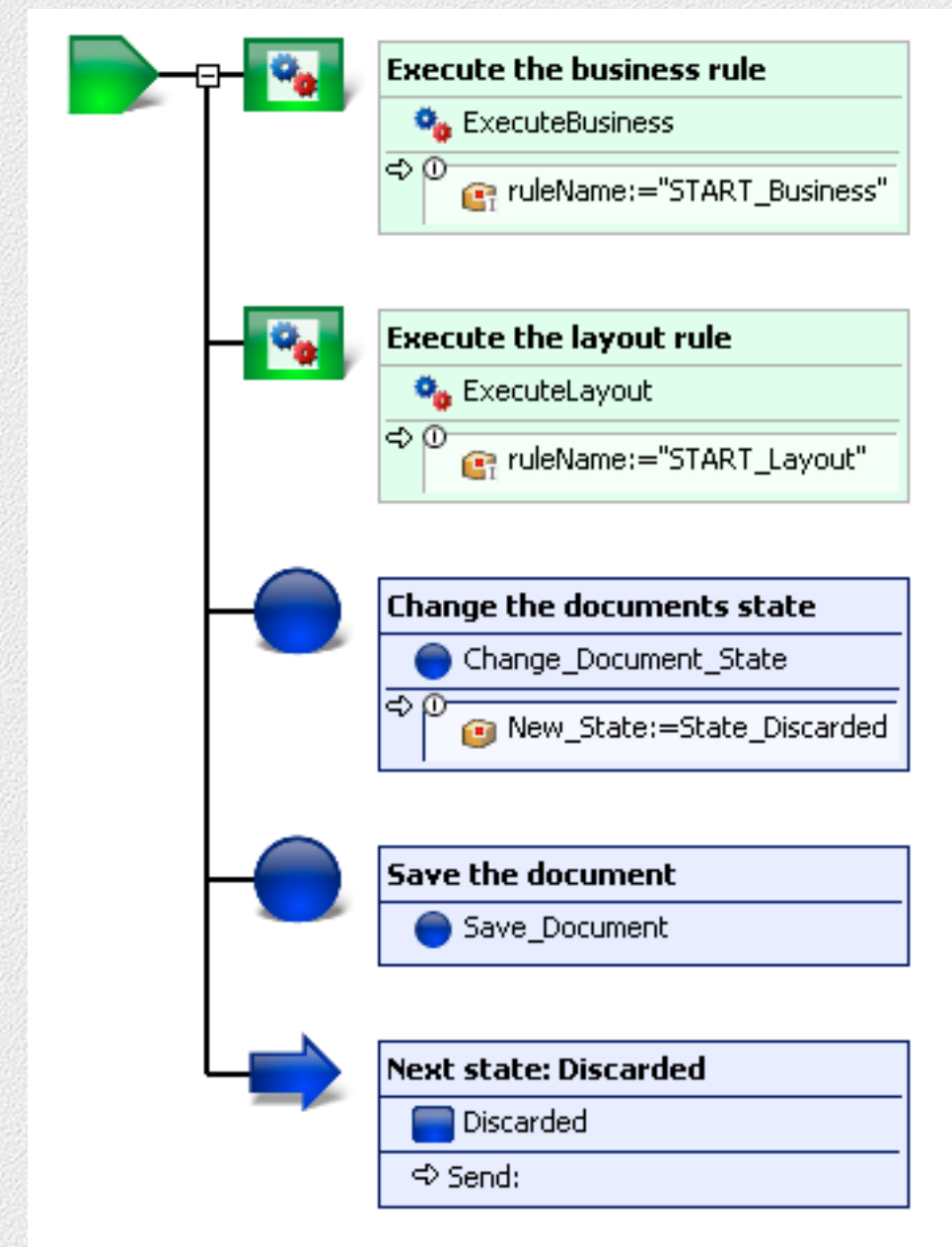
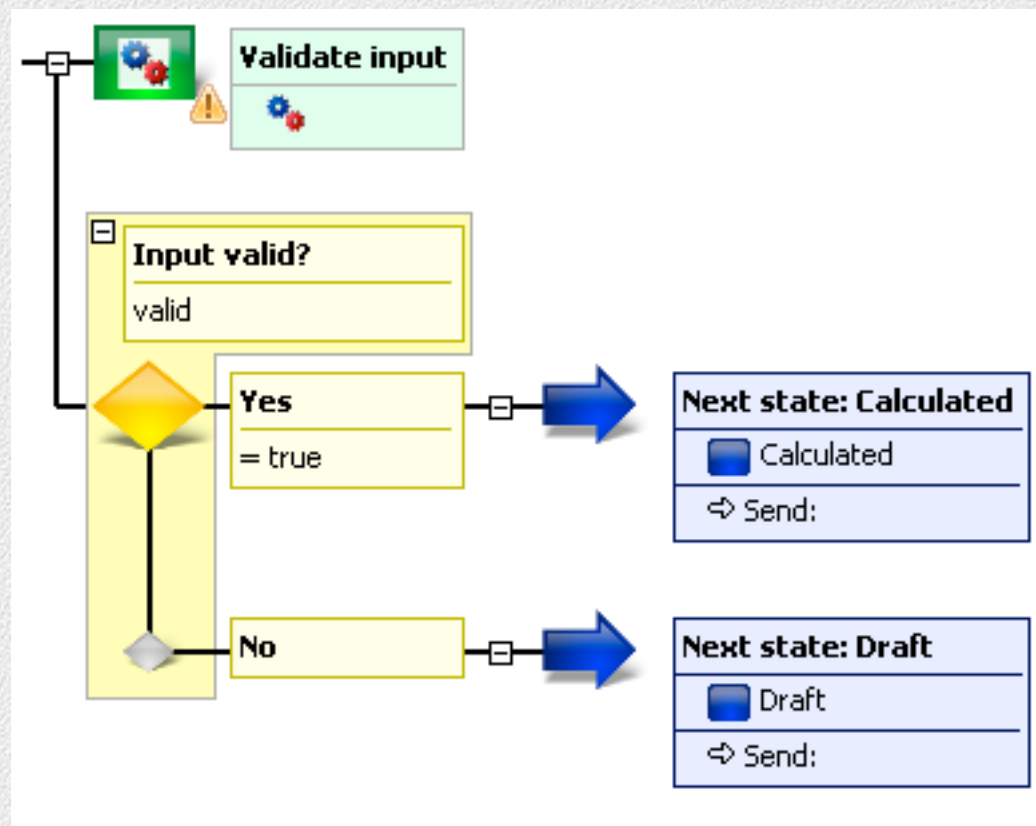
Parallele Abläufe



Beispiel: 4-Augen-Prinzip

Mediator Regeln

Ablauflogik und Bestimmung
der Zielzustände ➡



Modellierung des Layouts



a Section as the low-level container to hold two string and two date elements

sec_basic.labelKey:="sec_basic.labelKey"

sec_basic.nameOfReferencedContainer:=NAMEOF tab_webform

sec_basic.index:=30

sec_basic.numberOfLayoutColumns:=2

a StringElement to fetch pick-up destination

ui_pickupLoc.labelKey:="ui_pickupLocation.labelKey"

ui_pickupLoc.nameOfReferencedContainer:=NAMEOF sec_basic

ui_pickupLoc.nameOfReferencedDataElement:=NAMEOF pickupLoc

ui_pickupLoc.index:=31

a StringElement to fetch return destination

ui_returnLoc.labelKey:="ui_returnLocation.labelKey"

ui_returnLoc.nameOfReferencedContainer:=NAMEOF sec_basic

ui_returnLoc.nameOfReferencedDataElement:=NAMEOF returnLoc

ui_returnLoc.index:=32

a DateElement to fetch pick-up date

ui_pickupDate.labelKey:="ui_pickupDate.labelKey"

ui_pickupDate.nameOfReferencedContainer:=NAMEOF sec_basic

ui_pickupDate.nameOfReferencedDataElement:=NAMEOF pickupDate

ui_pickupDate.index:=33

a DateElement to fetch return date

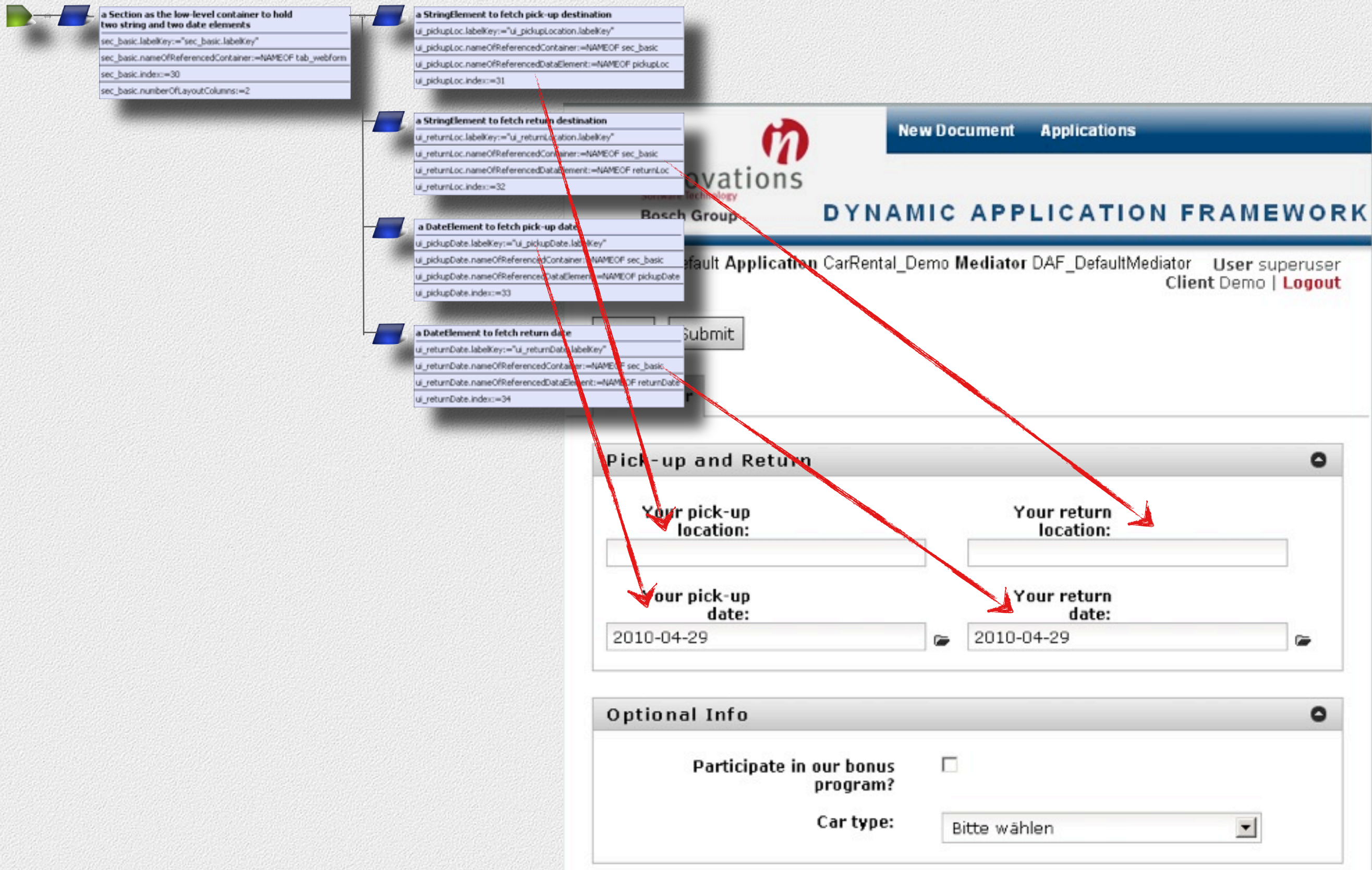
ui_returnDate.labelKey:="ui_returnDate.labelKey"

ui_returnDate.nameOfReferencedContainer:=NAMEOF sec_basic

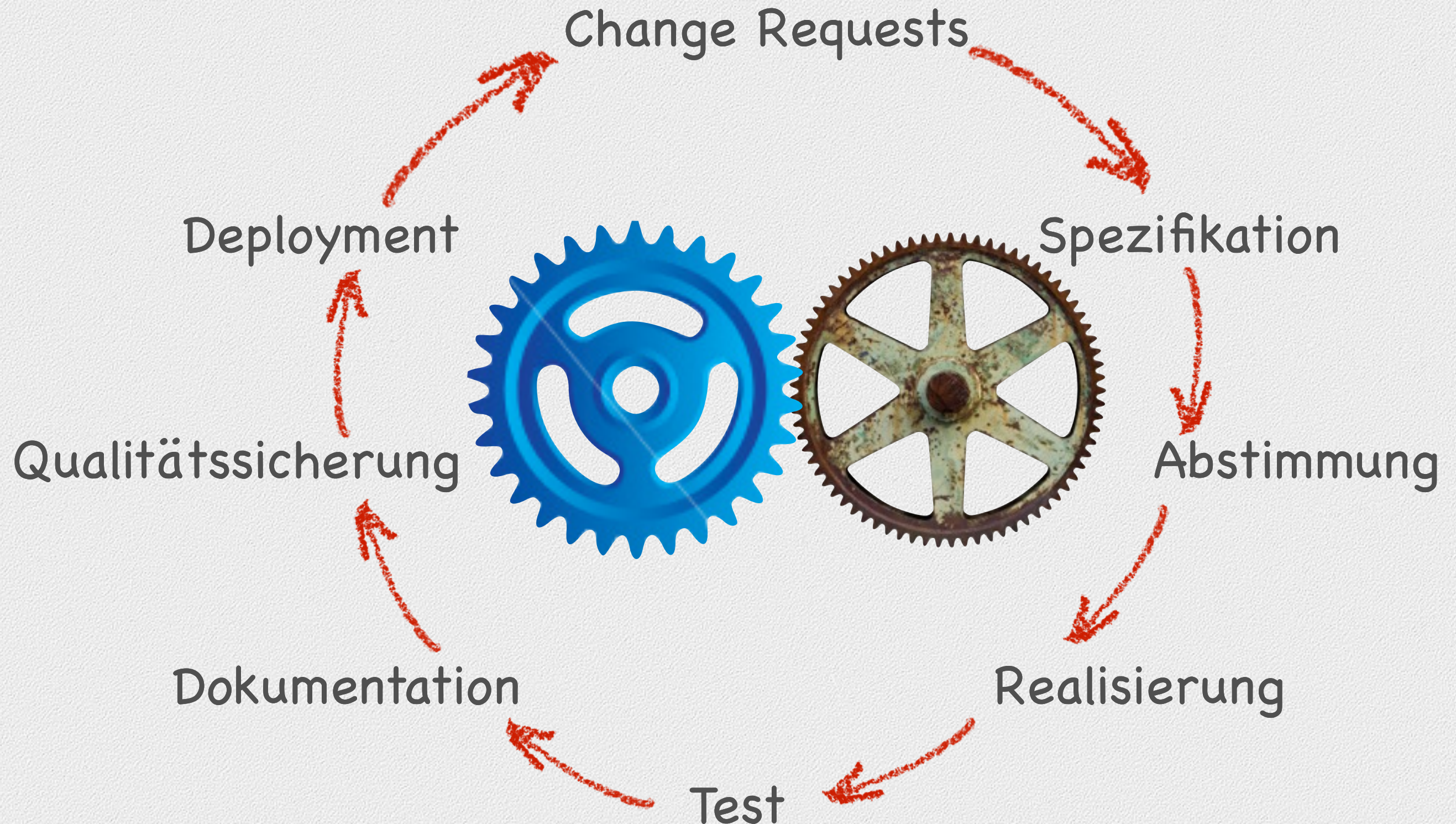
ui_returnDate.nameOfReferencedDataElement:=NAMEOF returnDate

ui_returnDate.index:=34

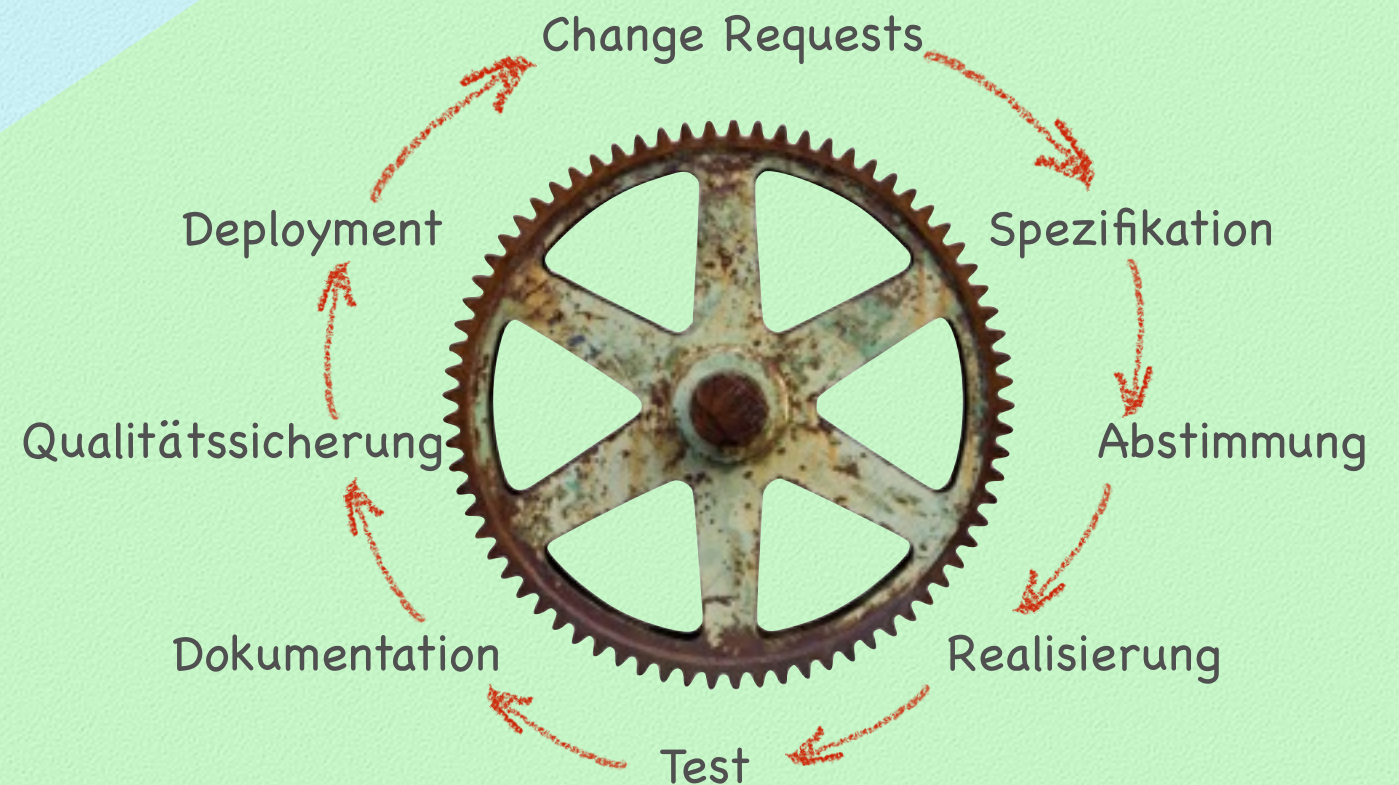
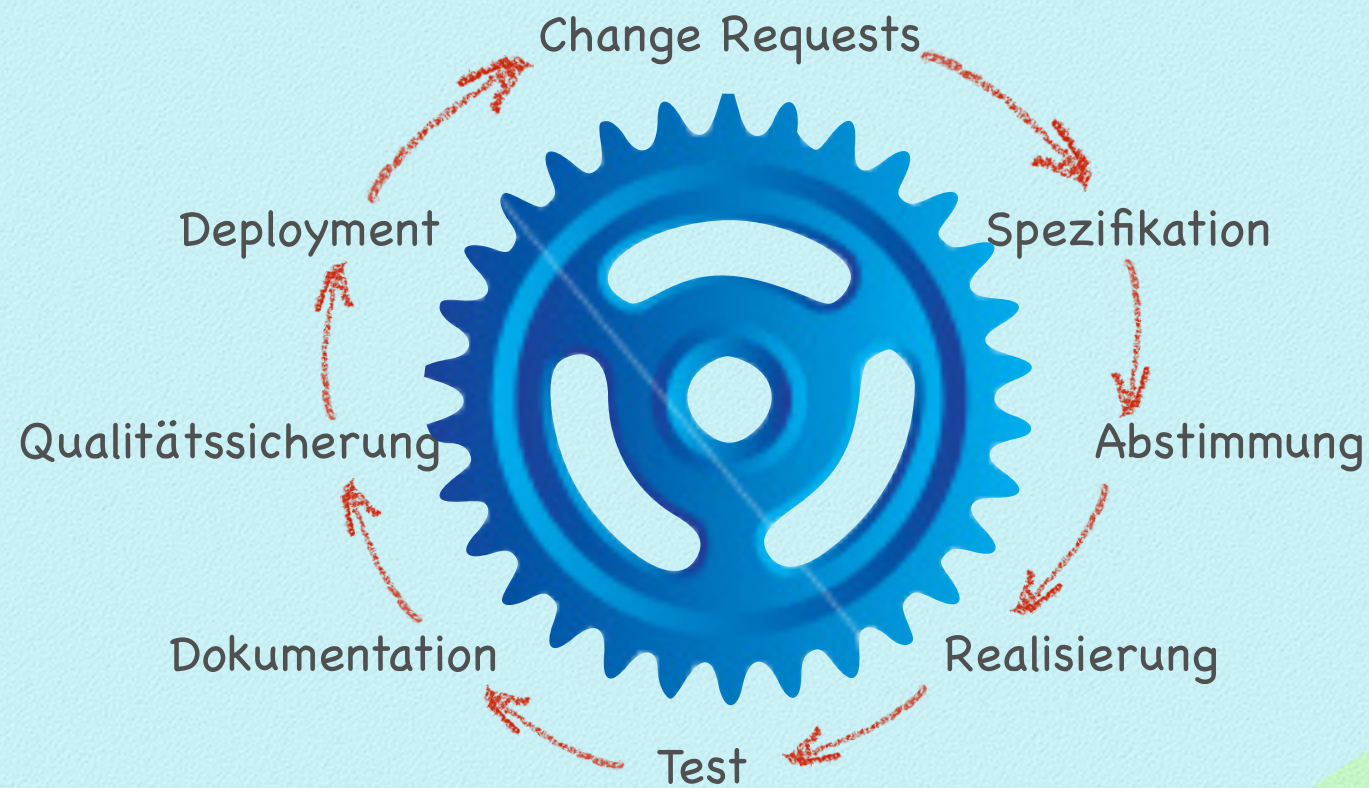
Modellierung des Layouts



Problem Agilität (a.k.a. Prozessentkopplung)



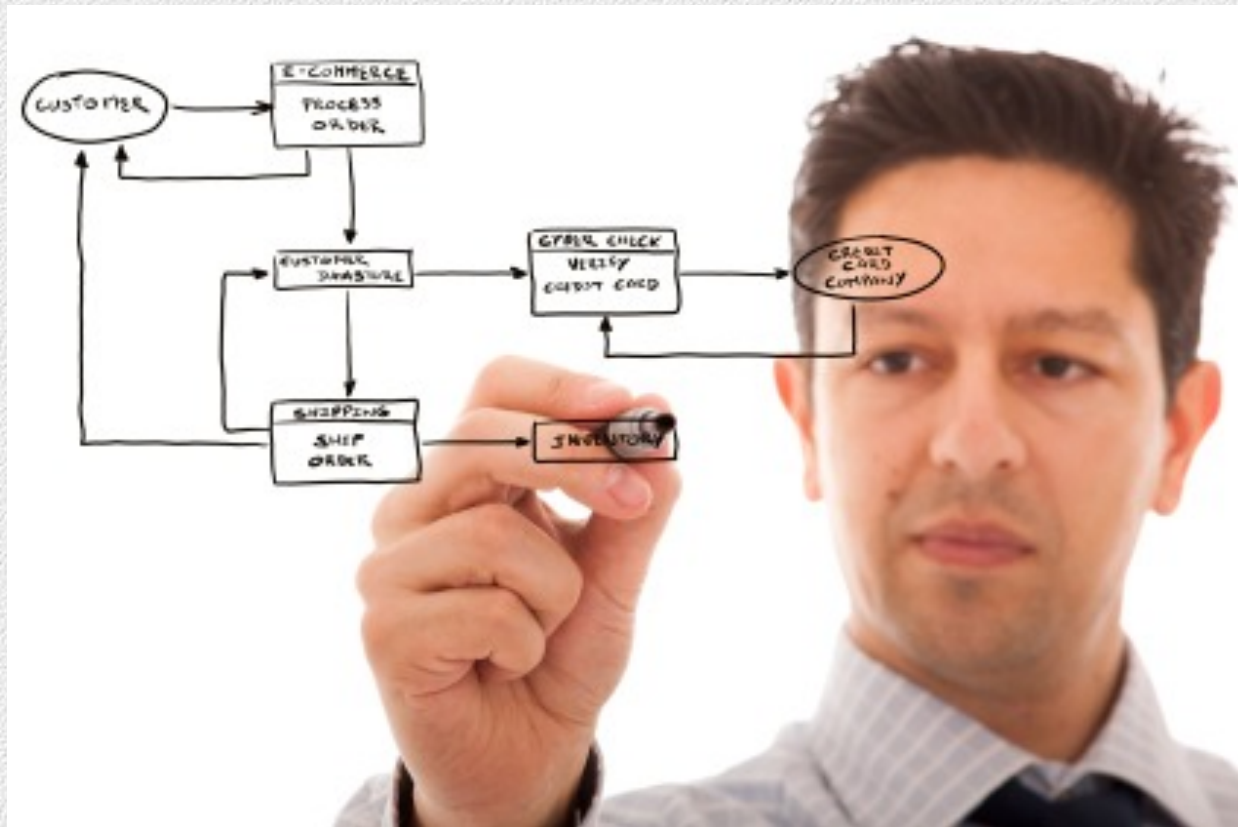
Separate Prozesse für Fachbereich und IT



Achtung: Fachbereich muss jetzt „entwickeln“

- Neue (IT-)Denkweise in Fachbereichen notwendig
 - Lifecycle wie Code (Model - Test - Deploy)
 - Versionierung, Branching, Merging,...
 - Übernahme der Verantwortung
- Modularisierungskonzept zwingend
- (noch) kein Modellierungsstandard

Abstimmung Fachbereich und IT...



Abstimmung Fachbereich und IT...



Contract

- Service/
Schnittstellen
(Infrastruktur und Umsysteme)
- Verfügbarkeit
- Antwortzeiten
- Datendurchsatz
- Mengengerüste
- Security
- Transaktionalität
- Verteilung
- Release-Zeitpunkte



Neue Freiheiten

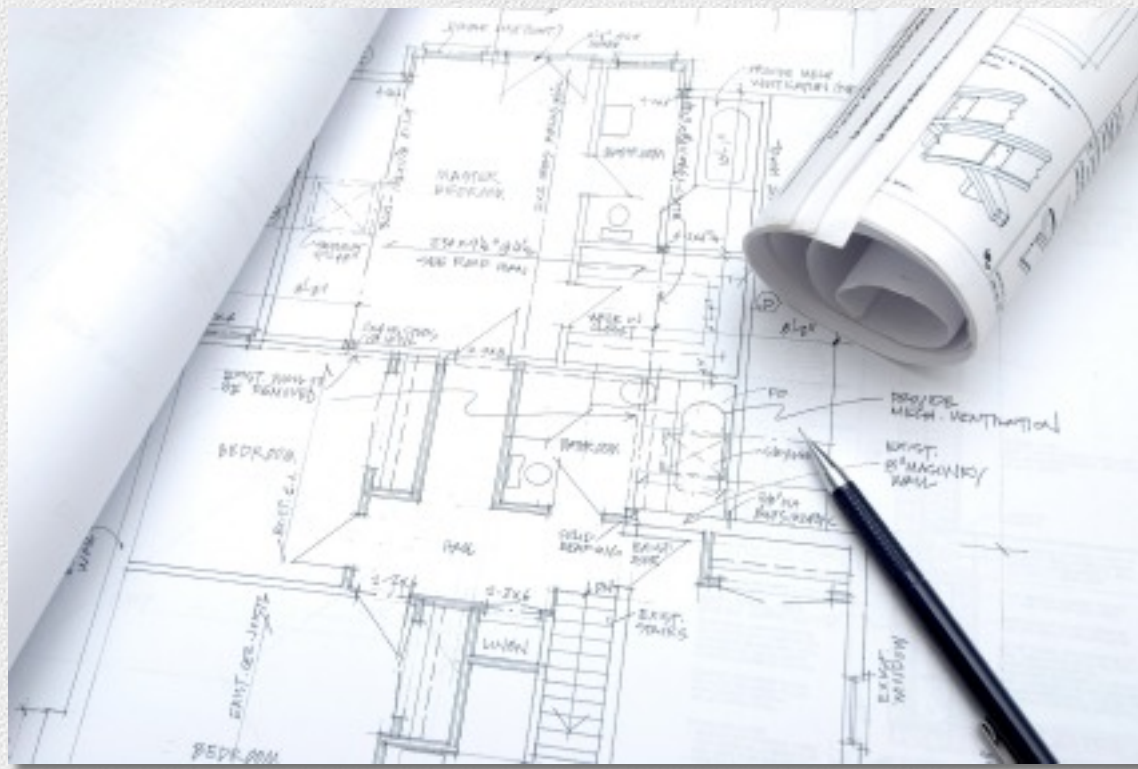


- Business (Objekt-) Modelle
- Autorisierung
- Konfiguration
- fachliche Plausibilisierung und Validierungen
- Eigene Werkzeuge
- fachliches Testing
- eigener „Rhythmus“

Neue Freiheiten

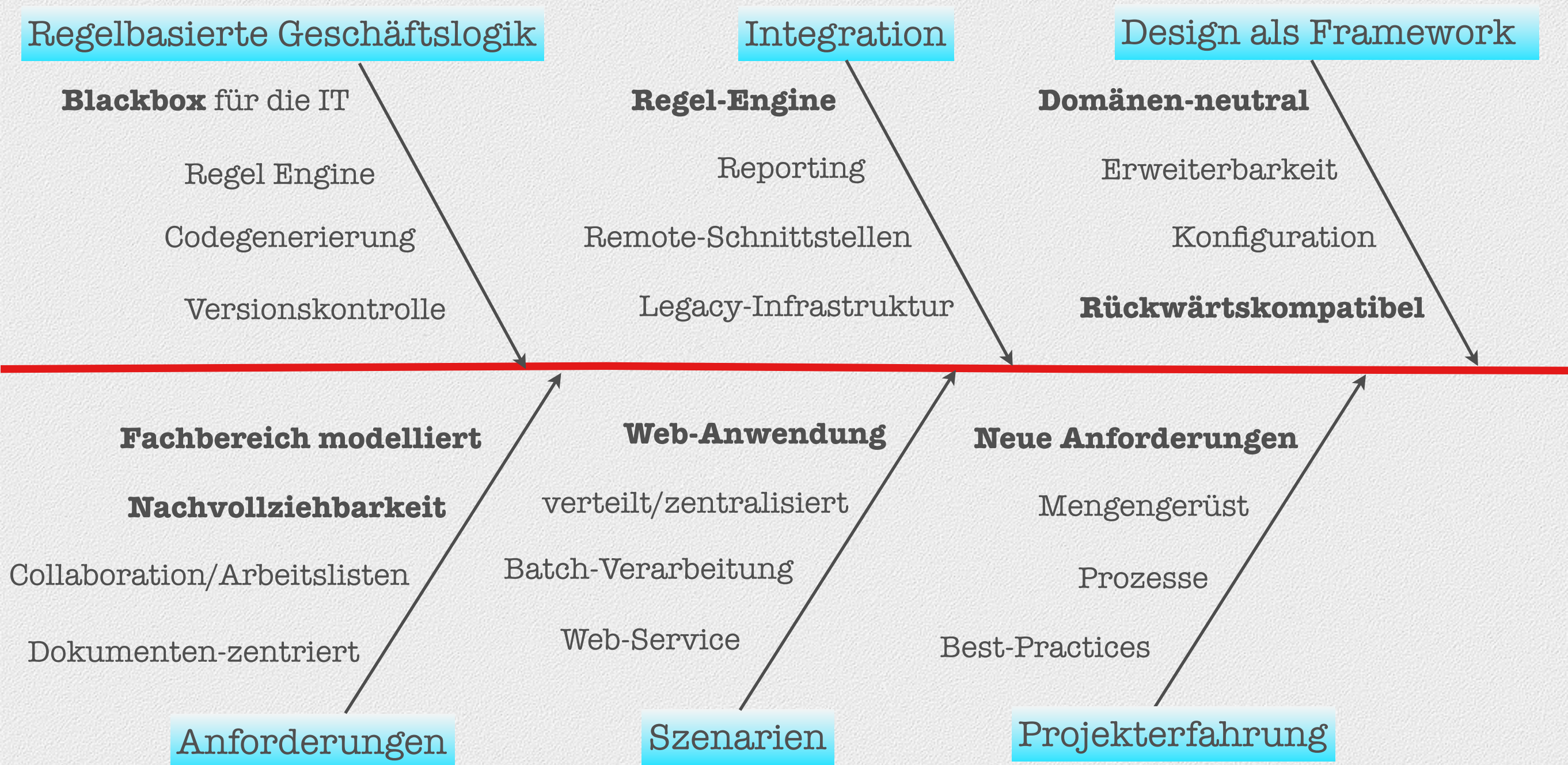


- Wahl von Algorithmen
- Wahl von Technologien
- Wahl der Hersteller
- Wahl der (Entwicklungs-) Werkzeuge
- Wahl von Vorgehensmodellen
- „nur noch“ syntaktische Validierungen
- Integrations- und Lasttests
- eigener „Rhythmus“

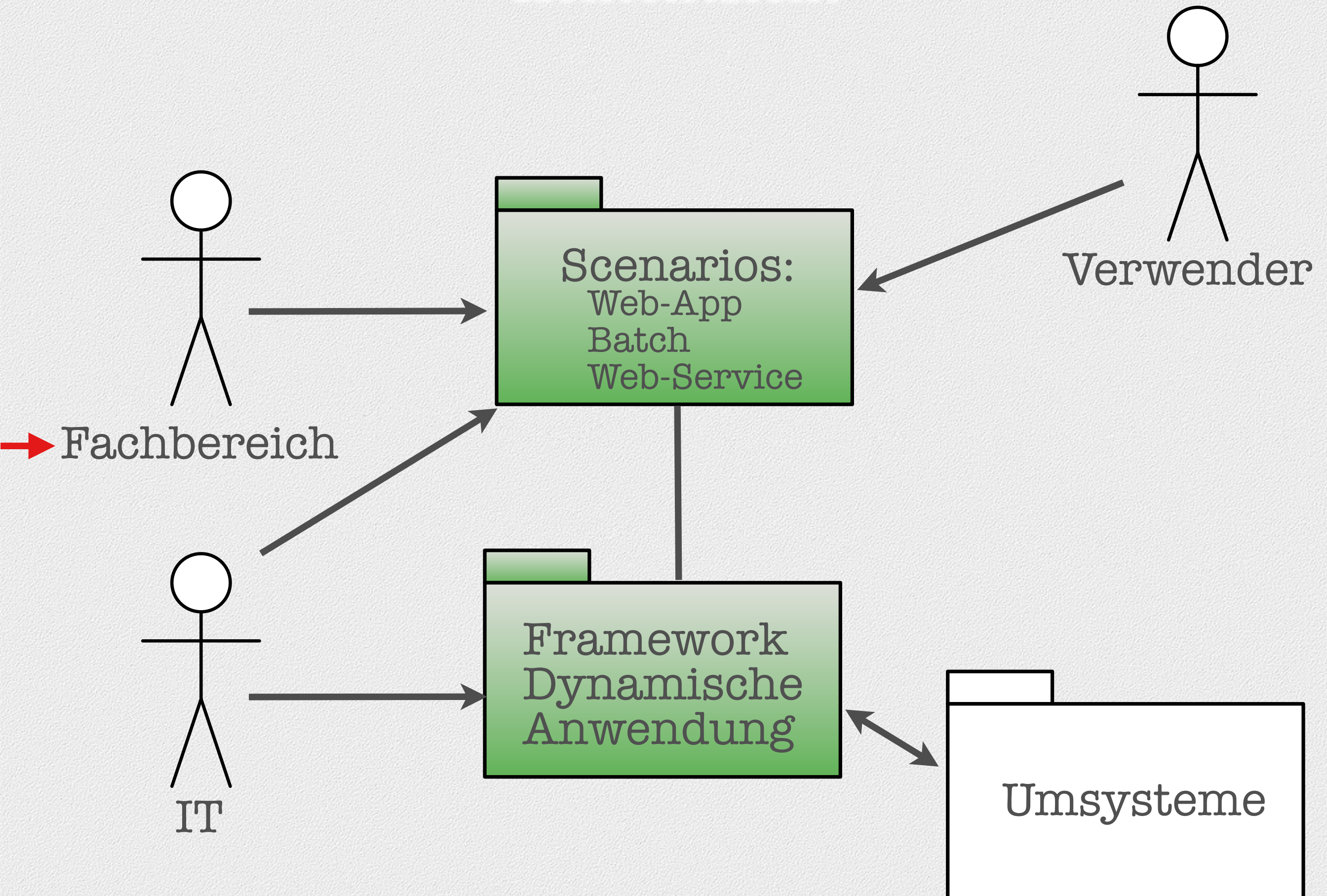


Die Architektur

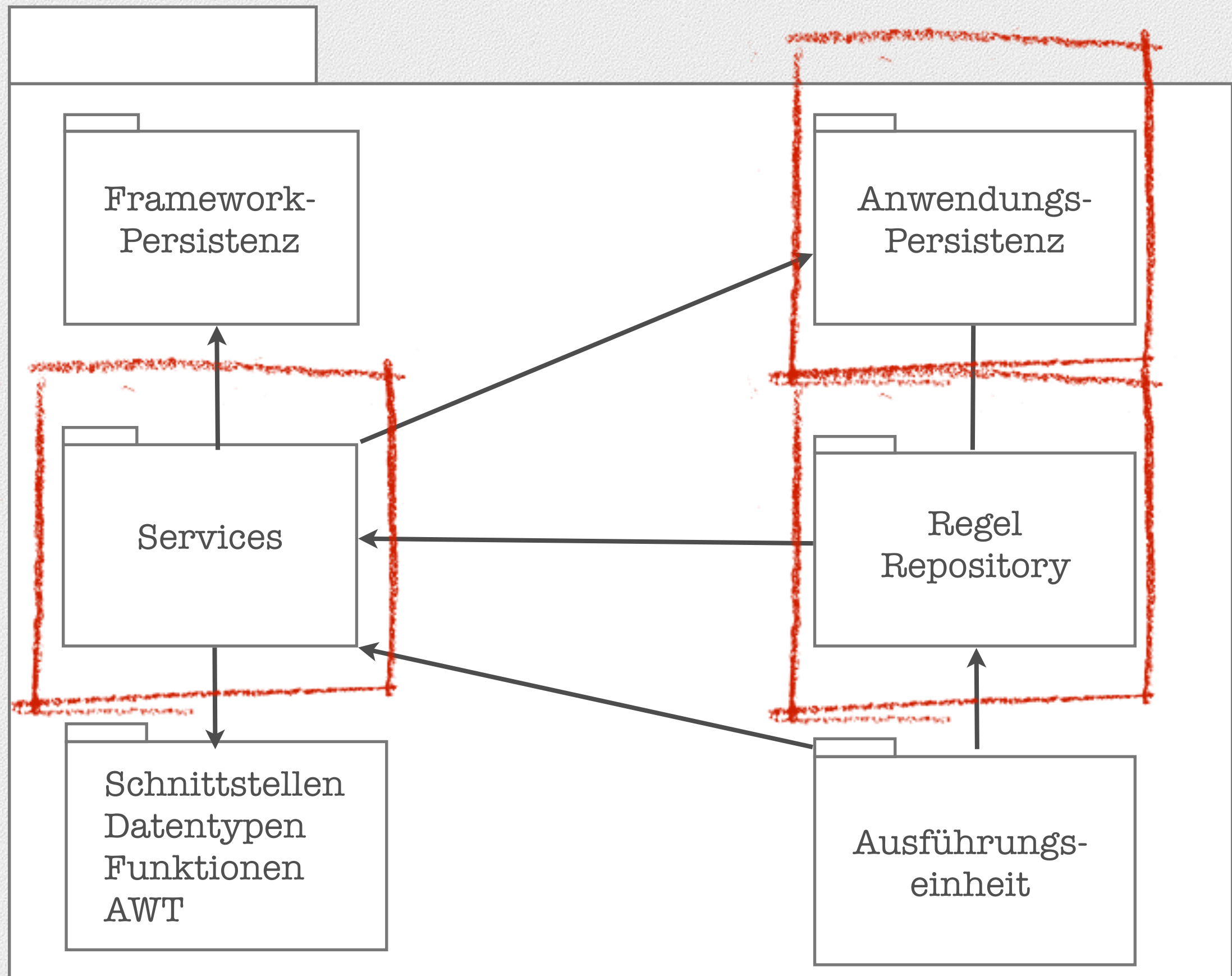
Treiber für die Architektur



Kontextsicht



Whitebox „Framework Dynamische Anwendung“



Regelrepository



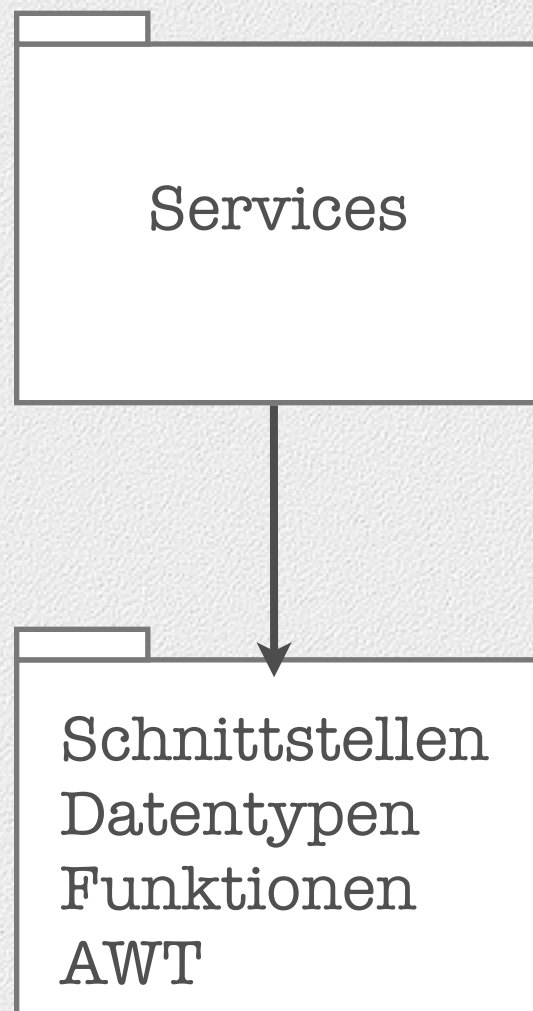
Typ	Name	Version	Valid from	Changed by	Active
 Business	Rating	v1.4	01.05.2010	User A	True
 Business	Rating	v1.2	17.02.2010	User B	True
 Ablauf	4-Augen	v1.0	01.05.2010	User C	True
 Layout	2-spaltig	v2.1	03.02.2010	User A	False
 Layout	3-spaltig	v.2.1	03.02.2010	User C	True

„Wer hat wann welche Logik in welcher Version aktiviert?“

v.a. für die
Modellierung

Wichtige Services

- Regelmodell Handling (“Hot Deployment“)
- „Dokumenten“-Management & Suche
- Locking
- Security & Identity Management
- Audit Logs
- Datentransformation & Migration
- I18n
- Rendering & Visualisierung
- Adapter für externe Systeme
- Cluster Support



Anwendungspersistenz

Anwendungs-
Persistenz

Fachbereich modelliert

beliebige Regelmodelle

Objektmodell unbekannt

Flexibilität

Iterative Entwicklung

Rahmenbedingungen

Wiederverwendung
bestehender DBMS

Datenmenge

Performance

Datentypen

Primitive

Strukturiert

Binärdaten/Attachments

Listen / Tabellen

Versionierung

Regelausführung

Beliebige Regelmodelle

Kompatibilität

Nachvollziehbarkeit

Bestehendes Tooling

„90-Grad-Drehung“

Reporting

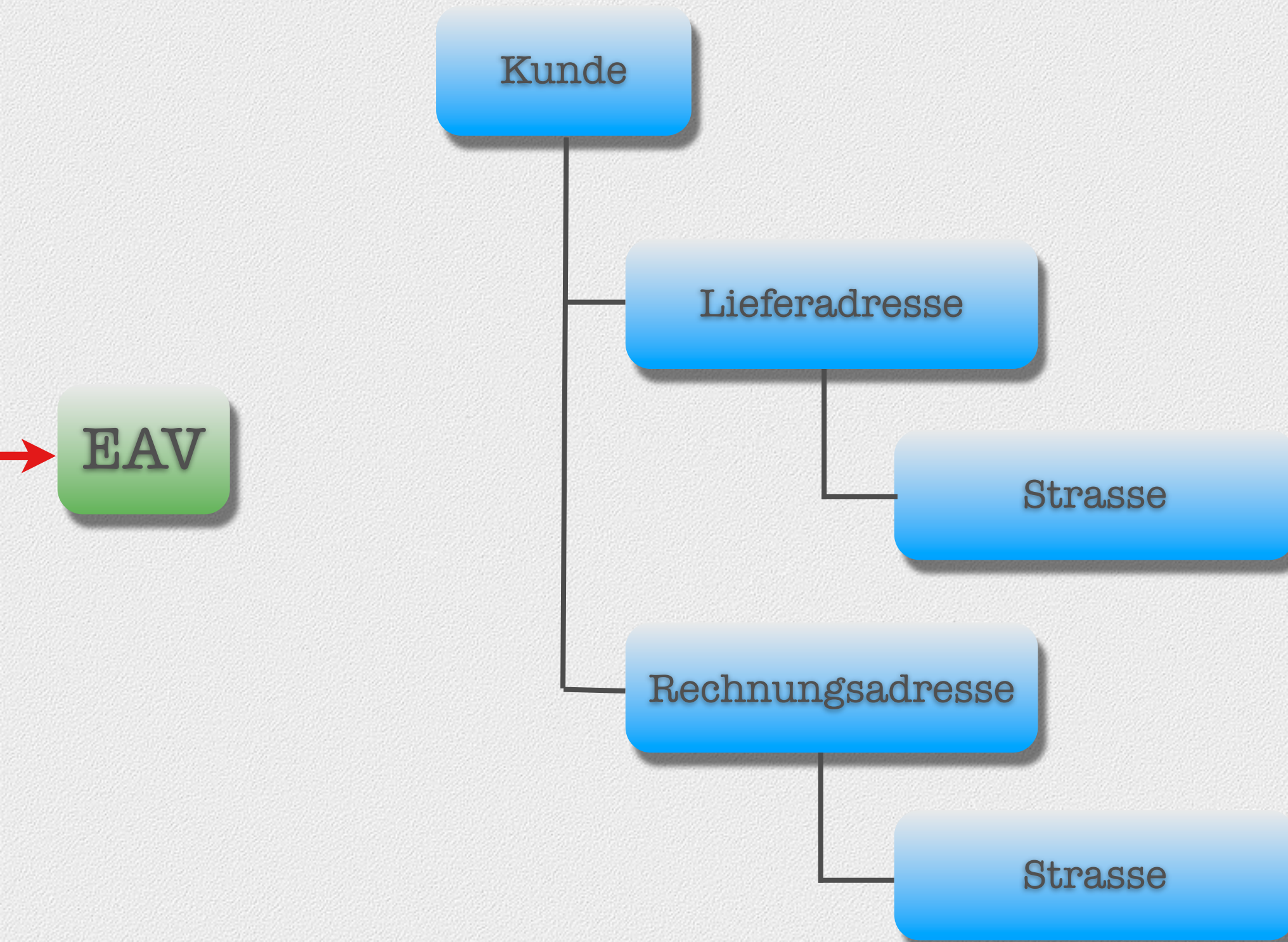
Optimierung

Bestehende DB-Tools

Zugriff auf Einzeldaten

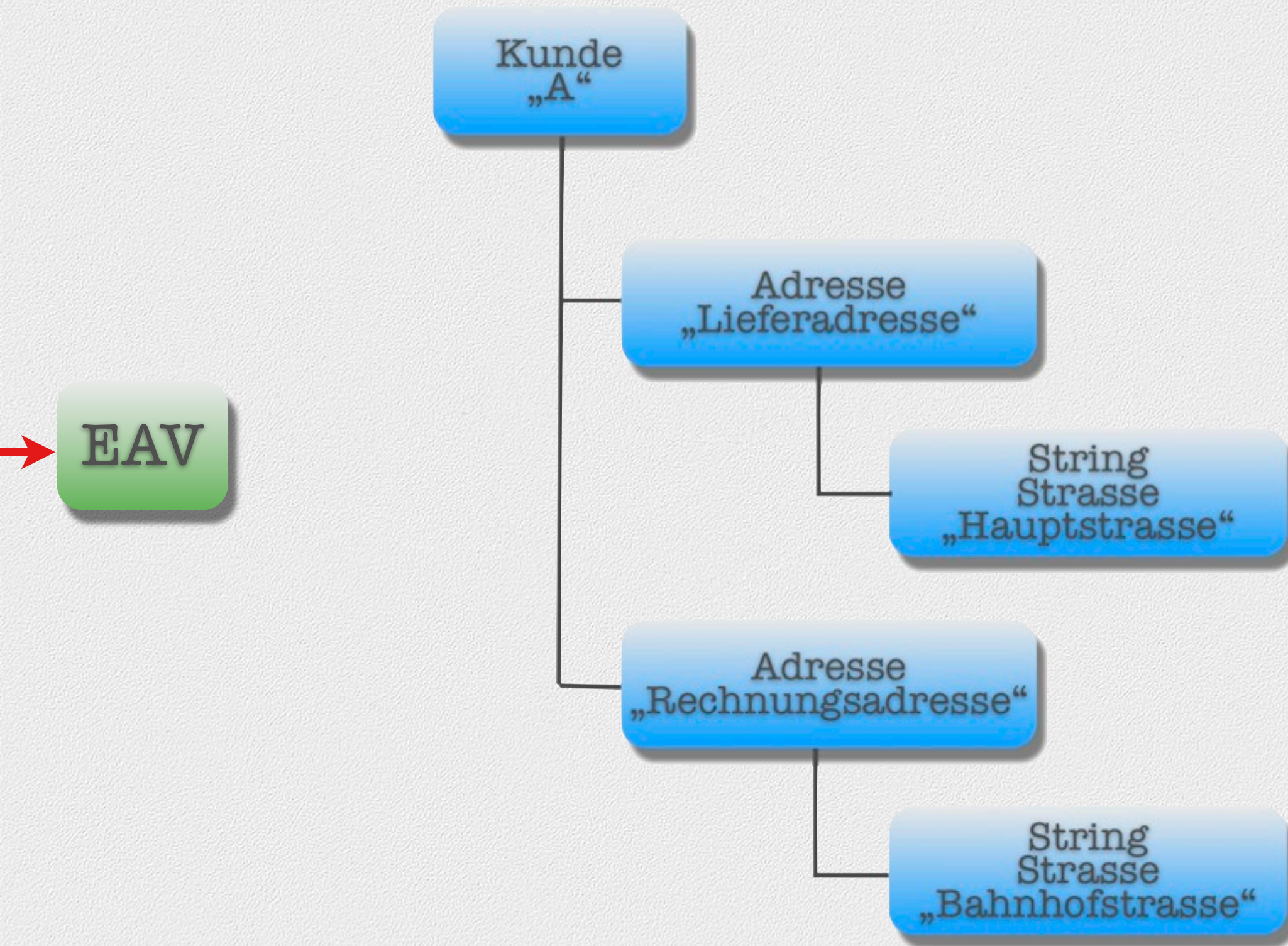
Wartung

Business Objektmodell

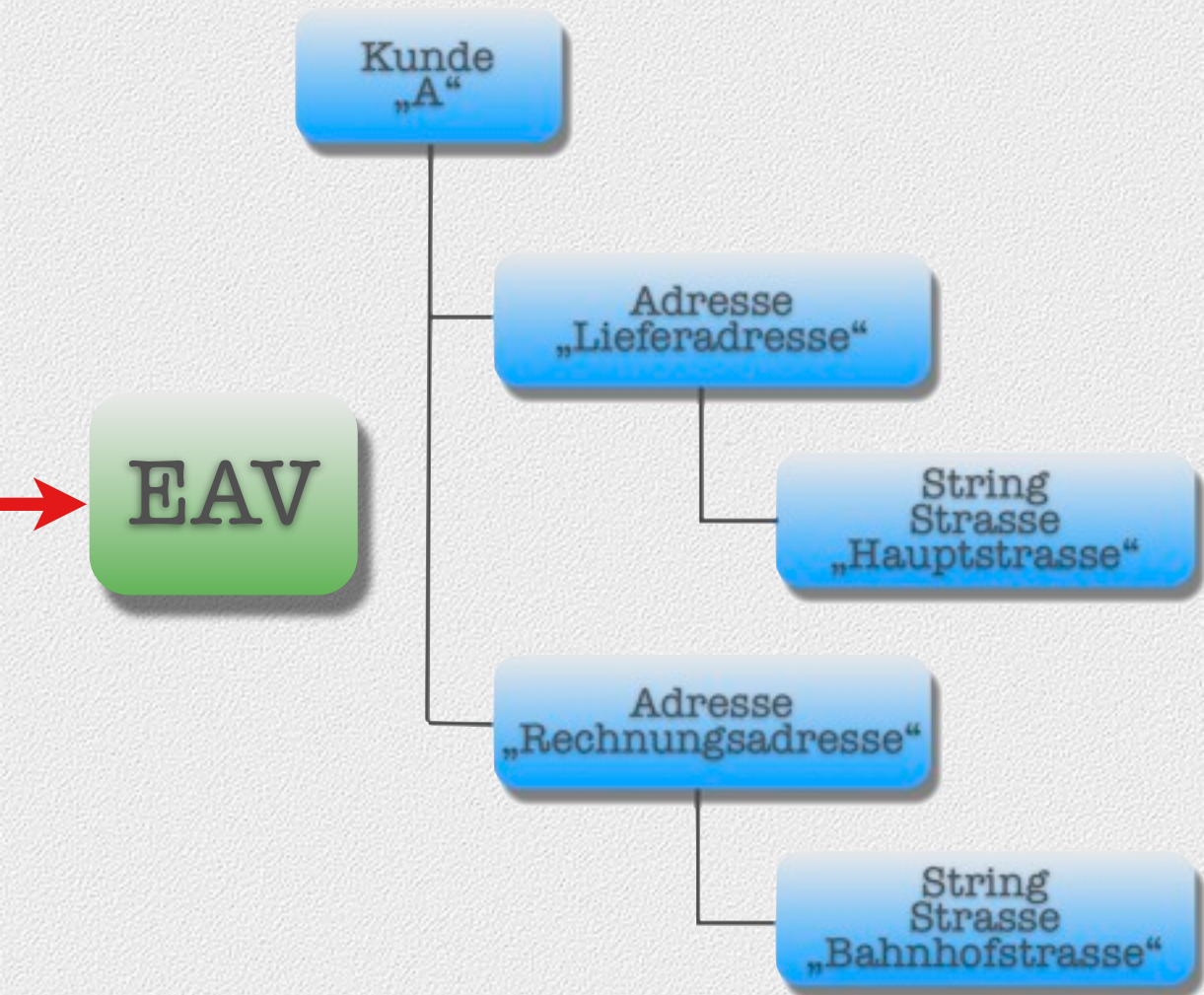


EAV = Entity Attribute Value

Objektgraph: Pfad, Typ und Wert



Objektgraph Flattening



Elementpfad	Datentyp	Wert
A	Kunde	
A.lieferadresse	Adresse	
A.lieferadresse.strasse	String	Hauptstrasse
A.rechnungsadresse	Adresse	
A.rechnungsadresse.strasse	String	Bahnhofstrasse

Nachvollziehbarkeit → Revisionssicherheit

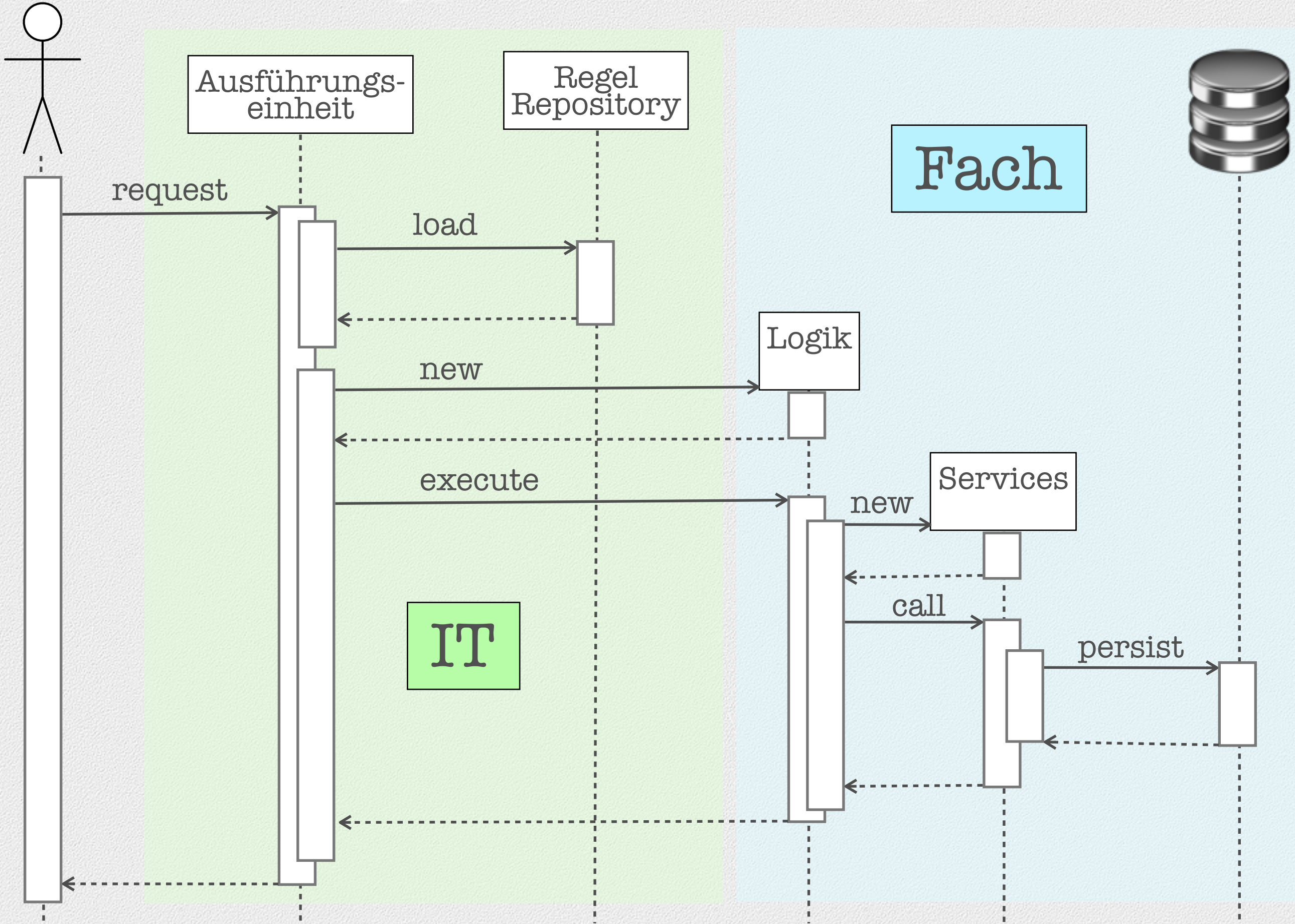
Typ	Name	Version
 Business	Rating	v1.4
 Business	Rating	v1.2
 Ablauf	4-Augen	v1.0
 Layout	2-spaltig	v2.1

Regelrepository

Elementpfad	Datentyp	Wert
A	Kunde	
A.lieferadresse	Adresse	
A.lieferadresse.strasse	String	Hauptstrasse
A.rechnungsadresse	Adresse	
A.rechnungsadresse.strasse	String	Bahnhofstrasse

Anwendungsdaten

Typische Verwendung





Nebeneffekte

Vergrabenes Wissen



Fachbereich



Sourcecode

Vergrabenes Wissen



Fachbereich



„Programmcode“

Stabilität

Geschäftsprozess A



Bugfix

Code

Geschäftsprozess A'



Geschäftsprozess B



Betrieb
Daten
Use Case

Stabilität

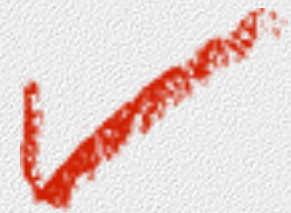
Geschäftsprozess A



Bugfix

Code
A

Geschäftsprozess A'

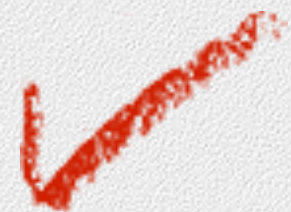


Geschäftsprozess B

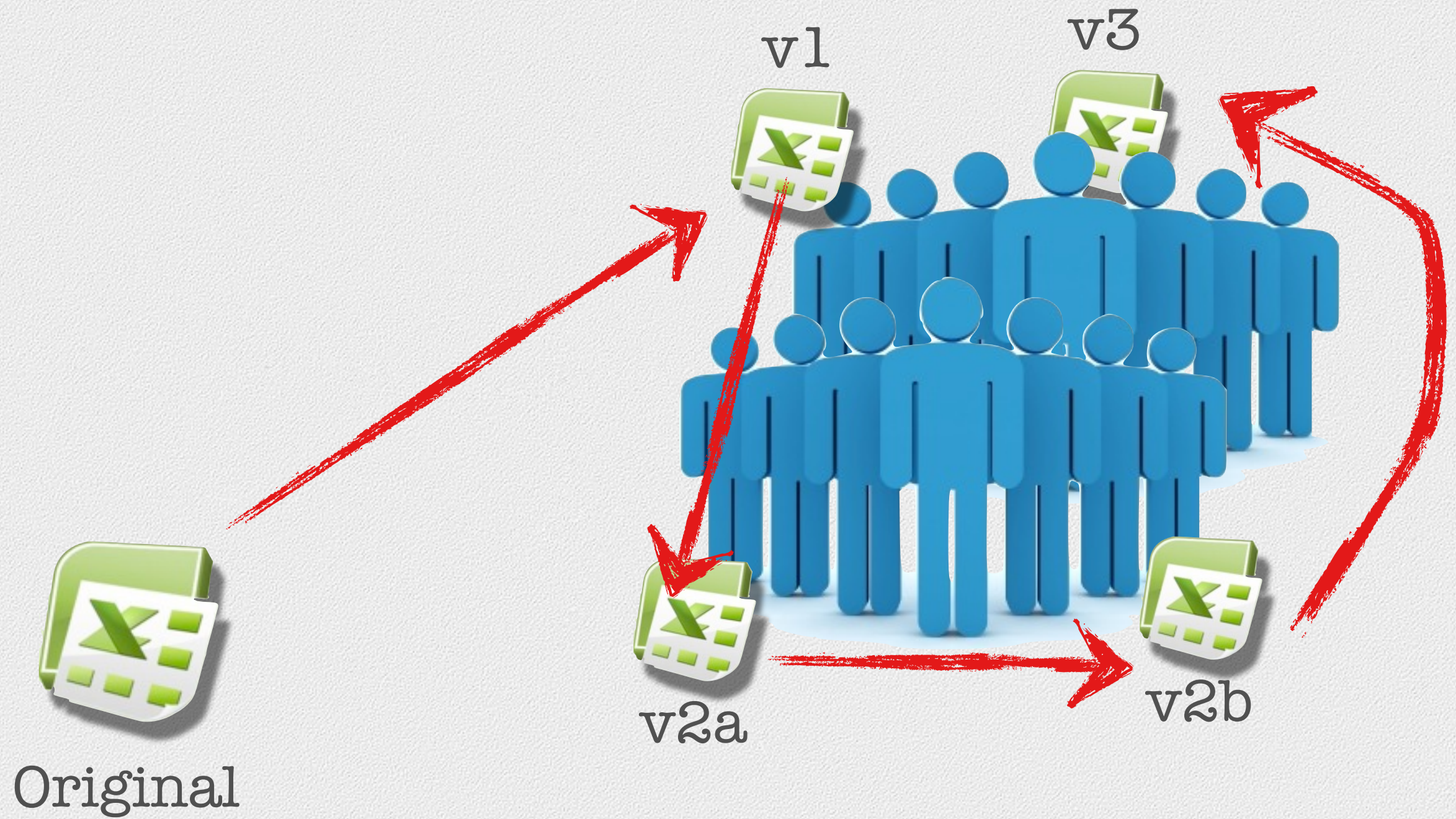


Code
B

Geschäftsprozess B



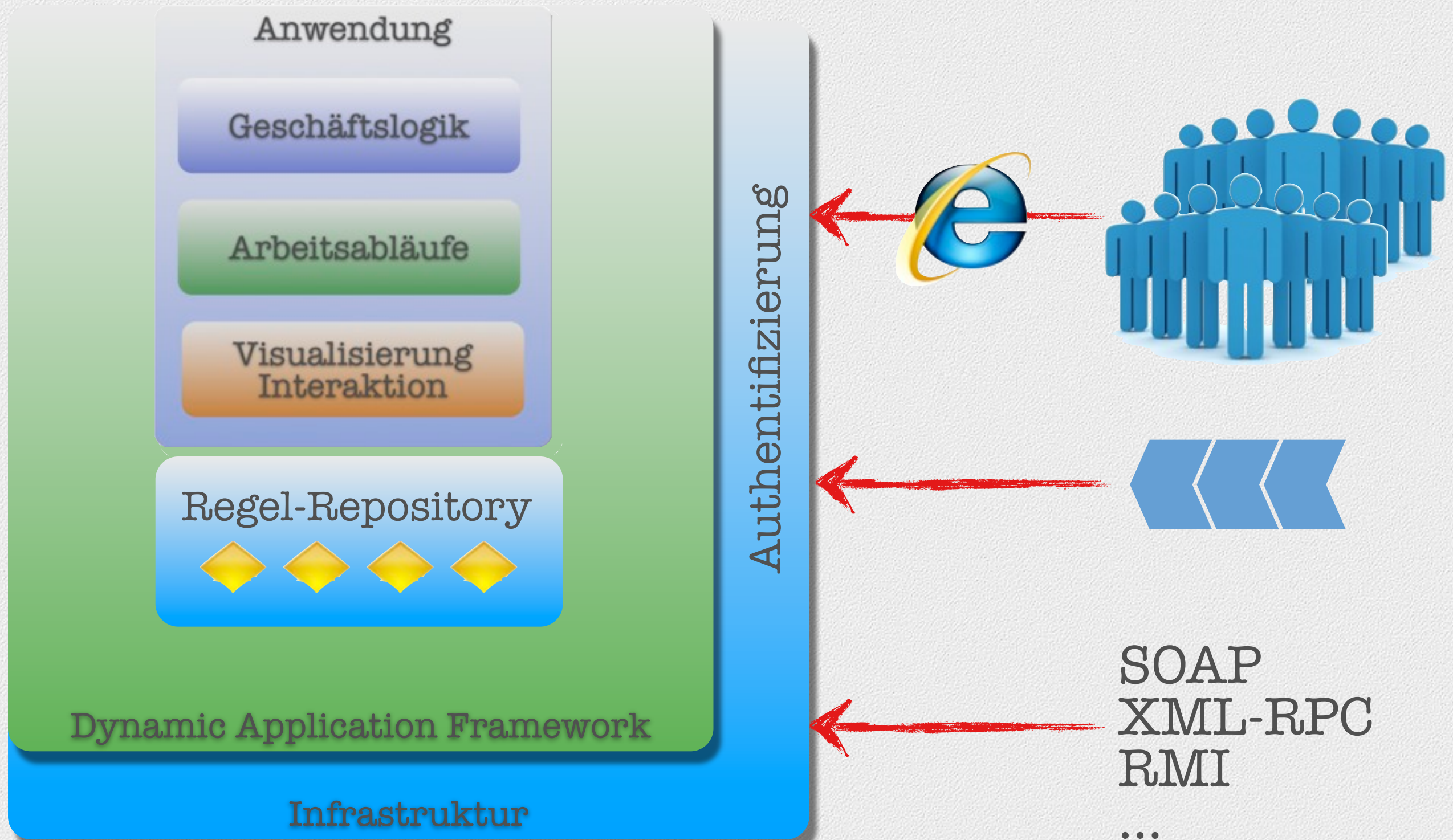
Skalierbarkeit...



...und Nachvollziehbarkeit



Dynamische Anwendungen, stabiles Fundament





Lessons learned

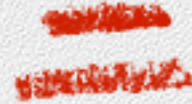
Gut geeignet für...

- ✓ Formular-basierte Dateneingabe
- ✓ Komplexe Geschäftslogik
- ✓ Häufige Änderungen in Geschäftslogik
- ✓ Komplexe Validierungen und Policies
- ✓ Dokumenten-orientierte Geschäftsprozesse
- ✓ Fachbereiche, die die Geschäftslogik selbständig verwalten wollen und können

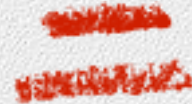
Weniger gut geeignet für...

- extrem große Datenmengen
- high performance Anwendungen
- Anforderungen, die Geschäftsprozess-übergreifende Logik benötigen
- spezialisierte Anwendungen für Power-User

Akzeptanz beim Fach



Geschäftsregeln



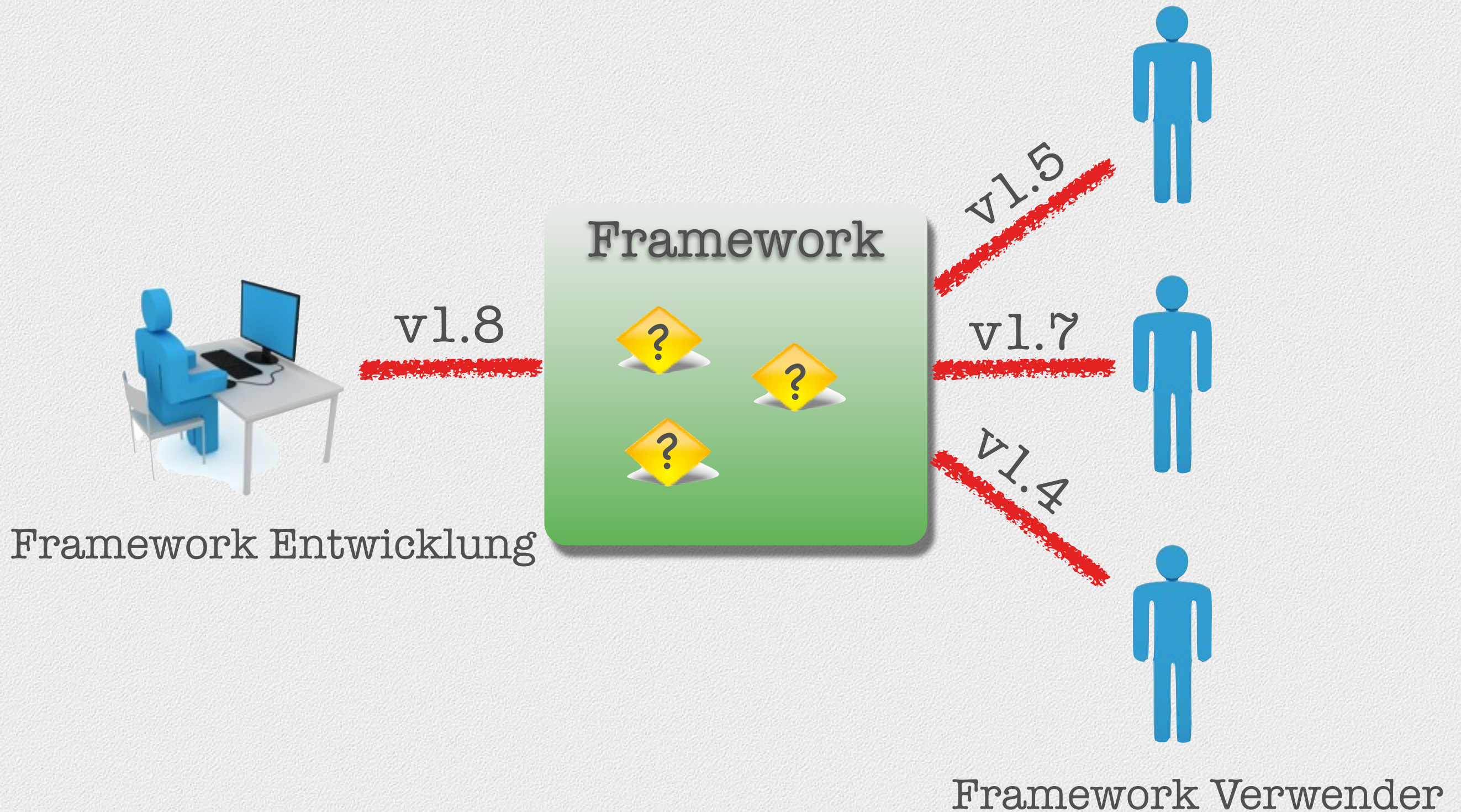
Ablaufregeln



Layoutregeln



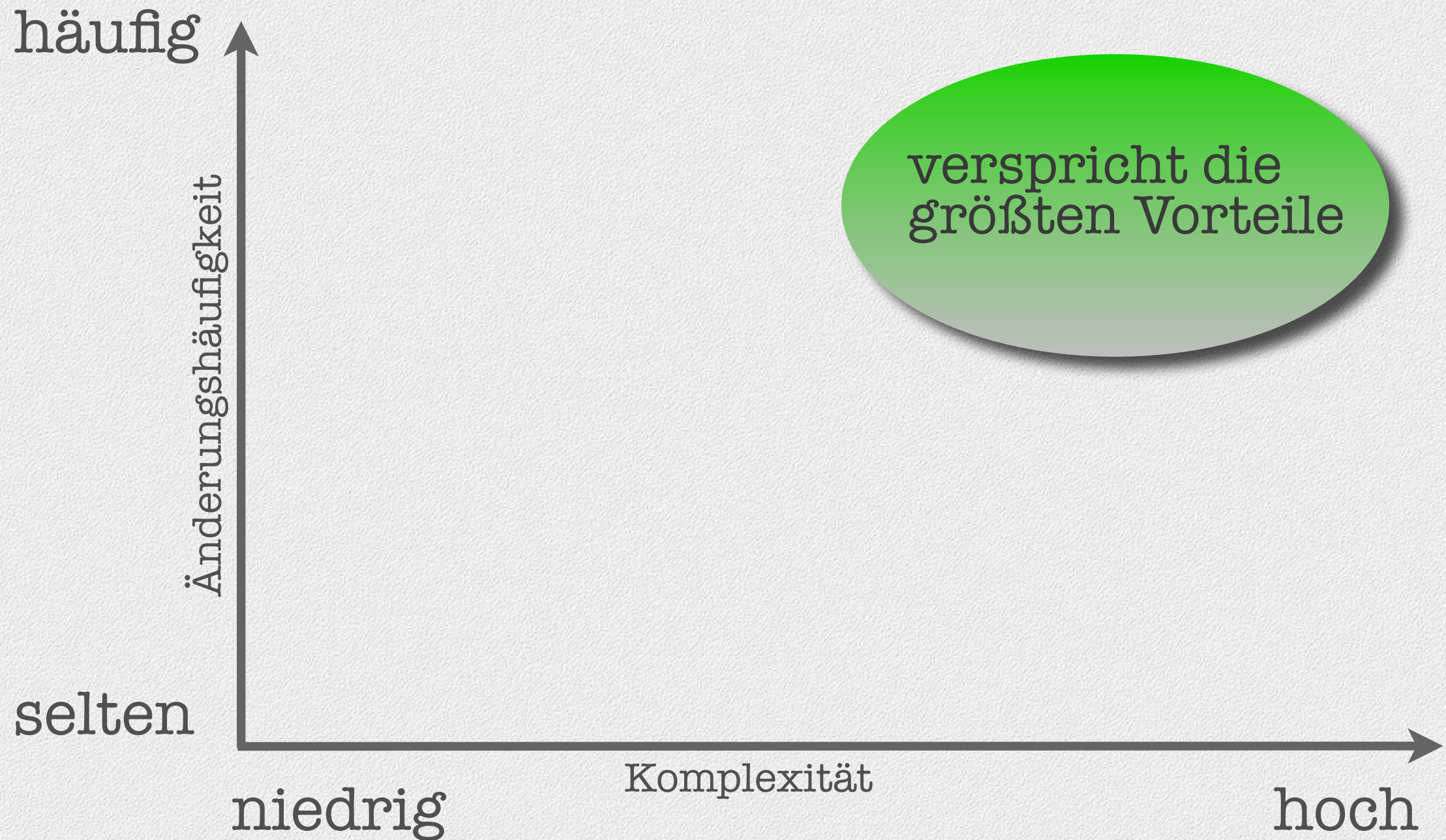
Framework-Entwicklung aufwändiger



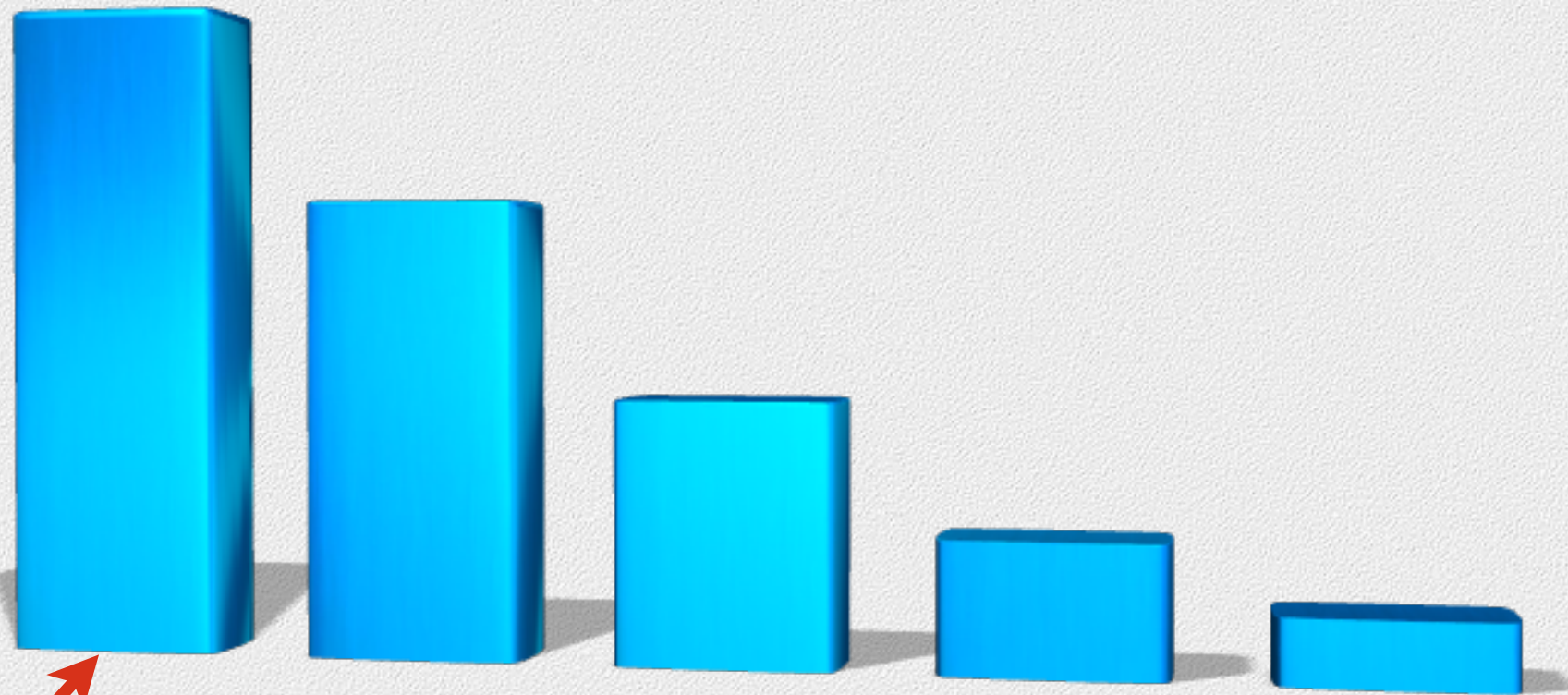
Dokumentation



Komplexe, sich häufig ändernde Geschäftslogik...



Time to Market



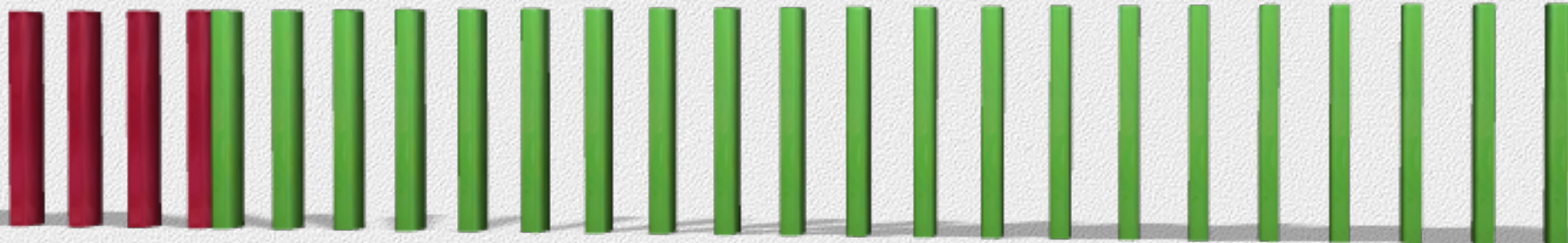
niedriger
Regelanteil

hoher
Regelanteil

Selbständige Fachbereiche

betreut

selbständig



Projektlaufzeit

Happy People...



Fachexperten

...weil sie **eigene** Werkzeuge bekommen

Regulierer

...weil Anwendungen **nachvollziehbar** sind

Endanwender

...weil Anwendungen **aktuell und dokumentiert** sind

Die IT

...weil sie nicht mehr der **Flaschenhals** ist

Einkäufer

...weil die Kosten **kalkulierbarer** sind



Nicht jede Geschäftsanwendung
muss als **dynamische Anwendung**
implementiert werden!

Vielen Dank für's
Zuhören!



Markus Schärtel
markus.schaertel@innovations.de