



Architekturprinzipien zukünftiger Bestandsführungssysteme

Dr. Gerhard Friedrich

M+I Management und Informatik Unternehmensberatung GmbH
Heute: www.act-mc.at

Dipl.Ing. Dr. Raimund Binder
Generali Office-Service und Consulting AG
Heute: www.biac.at

Themenübersicht

1. Bestandsführungssysteme sind überladen
- was ist der wesentliche Kern?
2. Step by Step Reengineering durch
Einsatz eines Produktservers
– Alternativen zum Big Bang?
3. Vertriebs- und Serviceprozesse erfordern
Cross-Plattform-Verfügbarkeit- Wege zum Ziel
4. Modulare, objektorientierte Anwendungen bauen mit
ausgereiften Technologien
– pragmatische Lösungsansätze

Bestandsführungssysteme
sind überladen -
was ist der wesentliche Kern?

Bestandsführungssysteme

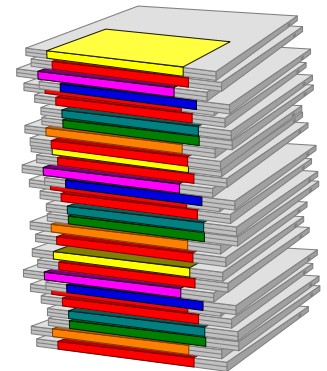
- Definitionsversuch (1) -

- Organisatorische Sicht:
Back Office-Anwendungen



- Technische Sicht:
Anwendungen am HOST bzw. Server

- Funktionale Sicht:
Management der Geschäftsdaten

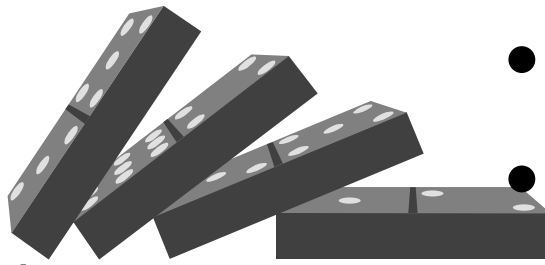


Bestandsführungssysteme

- Definitionsversuch (2) -

Fachliche Komponenten

- Partner
- Produkt
- Vertrag
- Schaden/Leistung
- Inkasso/Exkasso
- Vergütung
- Rechnungswesen

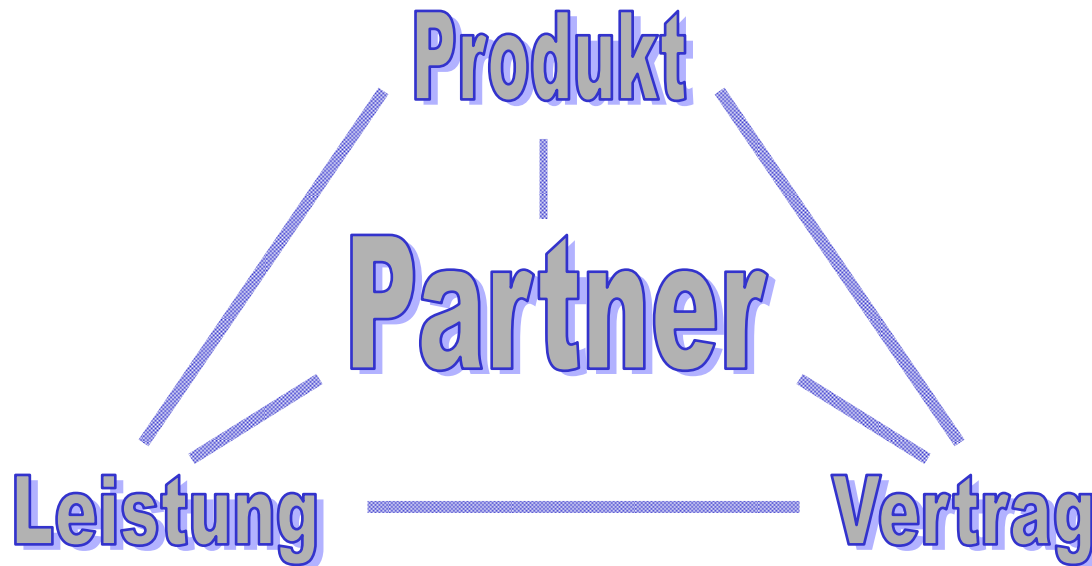


Technische Komponenten

- User Interfaces
- Datenbank
/Data Warehouse
- Text/Druck
- Kommunikation
- Workflow

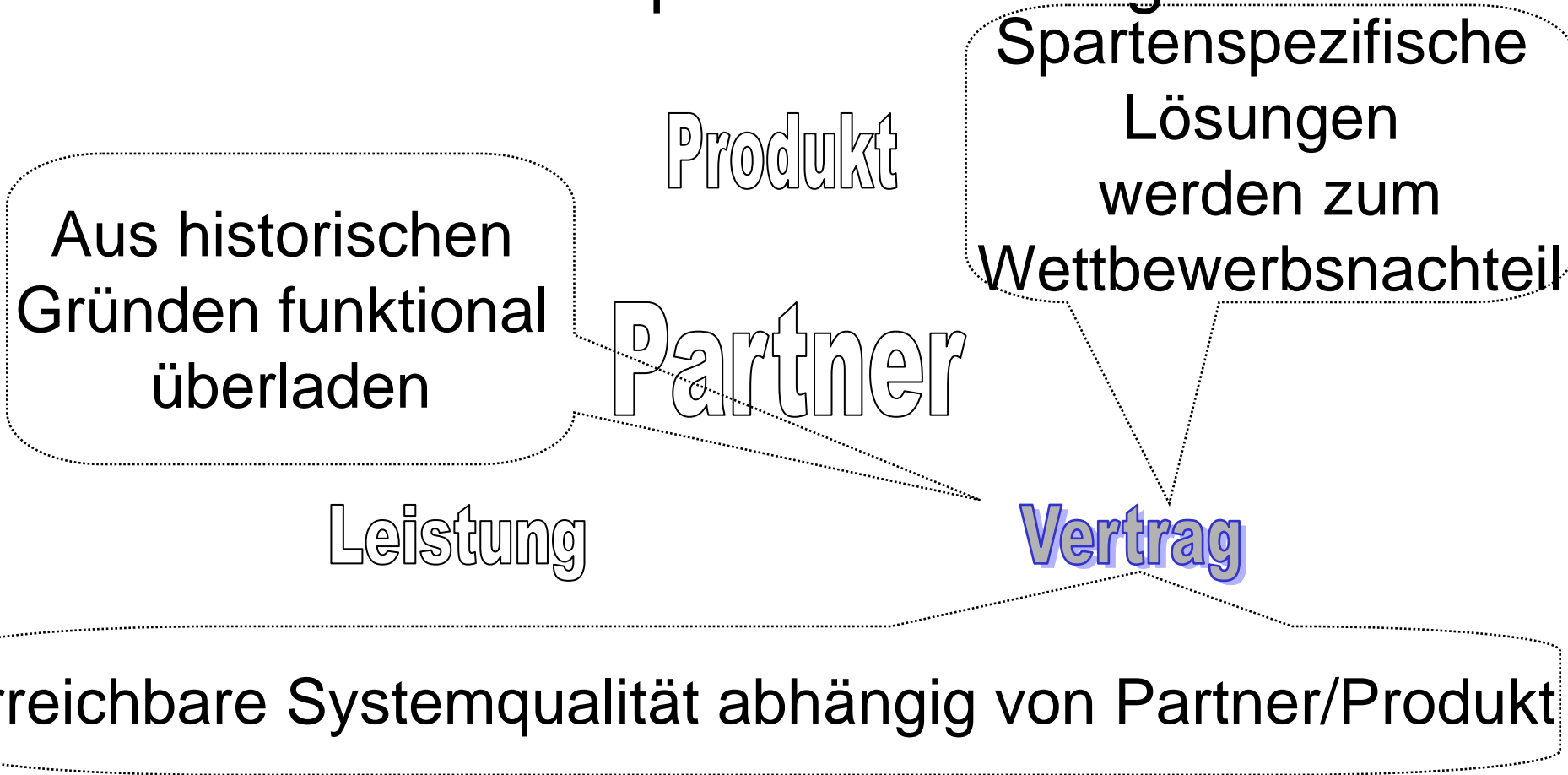
Bestandsführungssysteme

- Die fachlichen Kernkomponenten -



Bestandsführungssysteme

- Kernkomponente Vertrag -



Bestandsführungssysteme

- Kernkomponente Leistung -

Entscheidender
Wettbewerbsfaktor
(Kosten, Service)

Produkt

Vom
Zahlen und Buchen
zum

Partner Prozessmanagement

Leistung

Vertrag

Erfolgskritische Schnittstellen zu „Fremdsystemen“
(Bewertung, Assistance, Vertrieb, Text, E-Mail, Telefon, ...)

Bestandsführungssysteme

- Kernkomponente Produkt -

Zentrales Regelsystem
(incl. Mathematik)

Produkt

Auf absehbare Zeit
nur branchenspezifisch
lösbar

Partner

Leistung

Vertrag

Derzeit in Vertrag und Leistung „integriert“

Bestandsführungssysteme

- Kernkomponente Partner -

Partner, Adressen
und Objekte
sind kein
Branchenspezifikum

Produkt

Andere Branchen
haben mehr
Wettbewerbs Erfahrung

Partner

Leistung

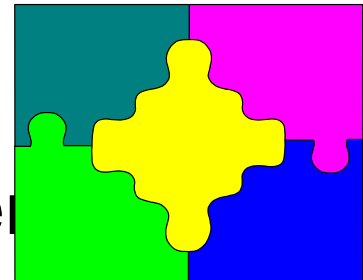
Vertrag

Branchenübergreifender Systemkern wäre Wettbewerbsvorteil

Bestandsführungssysteme

- Komponente Inkasso/Exkasso -

- Meist Eigenentwicklungen mit komplexen „Schnittstellen“ zu Vertrag und Leistung oder Branchenlösungen als Teil eines Gesamtpaketes
- SAP bietet nun mit IS/CD eine branchenspezifische Standardlösung als einzeln integrierbare Komponente
- Der Einsatz von branchenübergreifenden Lösungen (Banken, Versicherungen, Kreditkarten) brächte Wettbewerbsvorteile
- Einsatz branchenübergreifender Lösungen erfordert neue Qualität der Kernkomponenten



Bestandsführungssysteme

M+I

- Komponente Vergütung -

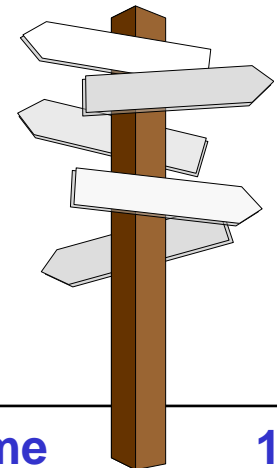
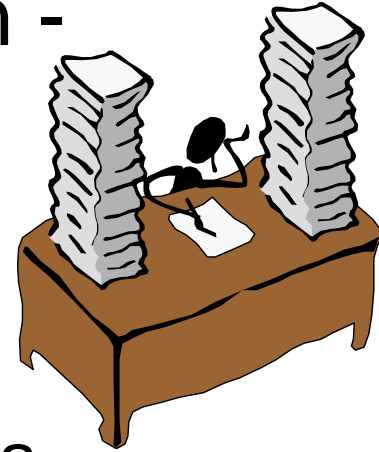
- Derzeit dominieren Eigenentwicklungen mit komplexen „Schnittstellen“ zu Vertrag, kaum zu Leistung, Controlling oder Produkt
- SAP plant eine branchenspezifische Standardlösung
- Der Einsatz von branchenübergreifenden Lösungen (Handel, Versicherungen) brächte Kosten- und Flexibilitätsvorteile



- Einsatz branchenübergreifender Lösungen erfordert neue Qualität der Kernkomponenten

Bestandsführungssysteme - Komponente Rechnungswesen -

- SAP dominiert den Markt
- Eigenentwicklungen sind mittlerweile die Ausnahme („Auslaufmodelle“)
- Branchenspezifische Lösungen sind kein Thema
- Entscheidungsunterstützung wird zum Erfolgsfaktor (Controlling, Data Warehouse, Data Mining)
- Hindernis: Defizite der Datennormierung bzw. -modellierung im versicherungsspezifischen Bereich



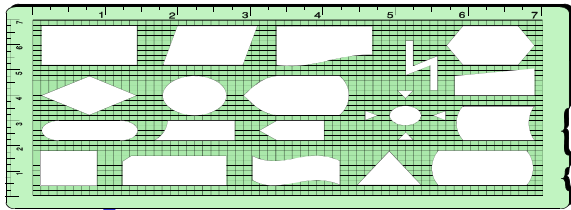
Bestandsführungssysteme

- Schwächen der heutigen Systeme -

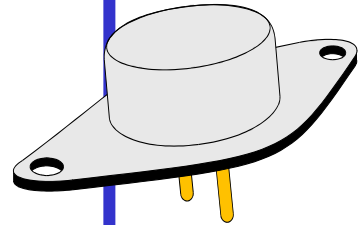
- Konzeptionelle Schnittführung zwischen den fachlichen sowie zwischen diesen und den technischen Komponenten uneinheitlich
- Applikationen wurden unter der Prämisse eines applikationsübergreifenden Datenmodells konzipiert, dieses ist aber nicht implementiert
- Datenmigration von Vorgängersystemen wurde nur teilweise gelöst
- Keine Cross-Plattform-Verfügbarkeit im Sinne von „write once - run everywhere“

Bestandsführungssysteme

- Zielarchitektur und Kernfunktionalität -



• Definition:
Spezifikation der Produkte und Geschäftsregeln

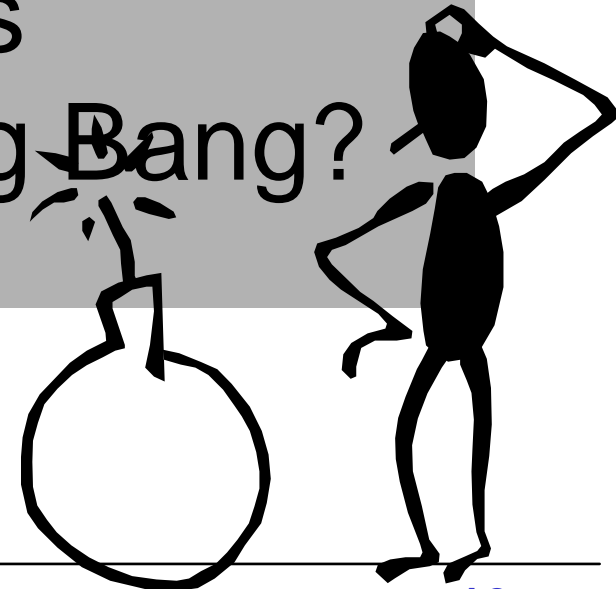


• Operation:
Definitionsbasierte Verarbeitung der Daten

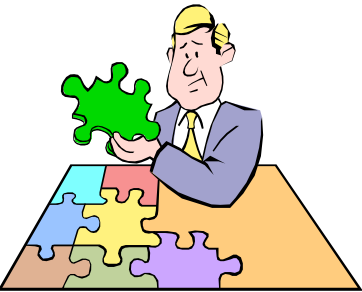
• Storage:
Archivierung/Retrieval der Geschäftsdaten



Step by Step Reengineering durch Einsatz eines Produktserver – Alternativen zum Big Bang?



Reengineering Basisstrategien „Entflechten“ und „Zudecken“



- **Entflechten:**
Trennung von Definition, Operation
und (falls nicht schon erfolgt) Storage

Zudecken:

GUI als Vorschaltssystem, möglichst als
“intelligente” Dialogschicht, die auch die
Dialogstruktur optimieren kann



Entflechten mit einem Produktserver (1)

Beispiel VP/MS *)

- VP/MS kapselt die gesamte Prüf- und Rechenlogik der Produkte
- Logische Schnittstelle zu Vertrag und Leistung ist die Liste der Datenfelder
 - Verarbeitungslogik in Vertrag und Leistung ist produktneutral
- Dateninhalte werden ausschließlich von VP/MS geprüft bzw. ermittelt



*) alle Aussagen
gelten auch für
PROMOS
(www.sbi.ch)

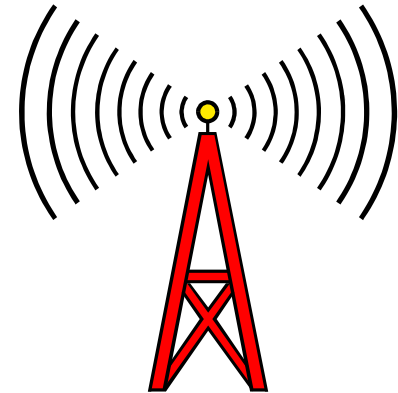
Anm.: G. Friedrich, 2007

Entflechten mit einem Produktserver (2)

Beispiel VP/MS

Schnittstellenfunktionen z.B. im Vertrag

- SETVAR (Eingabefeld): Wertübergabe
- COMPUTE (Ausgabefeld): Wertermittlung
- CHOICE (Eingabefeld): Wertebereich
- AVAILABLE (Ausgabefeld, Eingabefeld):
Verwendungsnachweis Eingabefeld
für die Ermittlung des Ausgabefeldes



Entflechten mit einem Produktserver

Resümée am Beispiel VP/MS

- Definitionsschicht bietet u.a. folgende Services
 - Prüfung aller Eingabedaten (incl. Wertebereiche)
 - Berechnung aller Ausgabedaten
 - Kontextspezifische Vollständigkeitsprüfung
 - Hinweise, Fehlermeldungen, Hilfe, Textbausteine

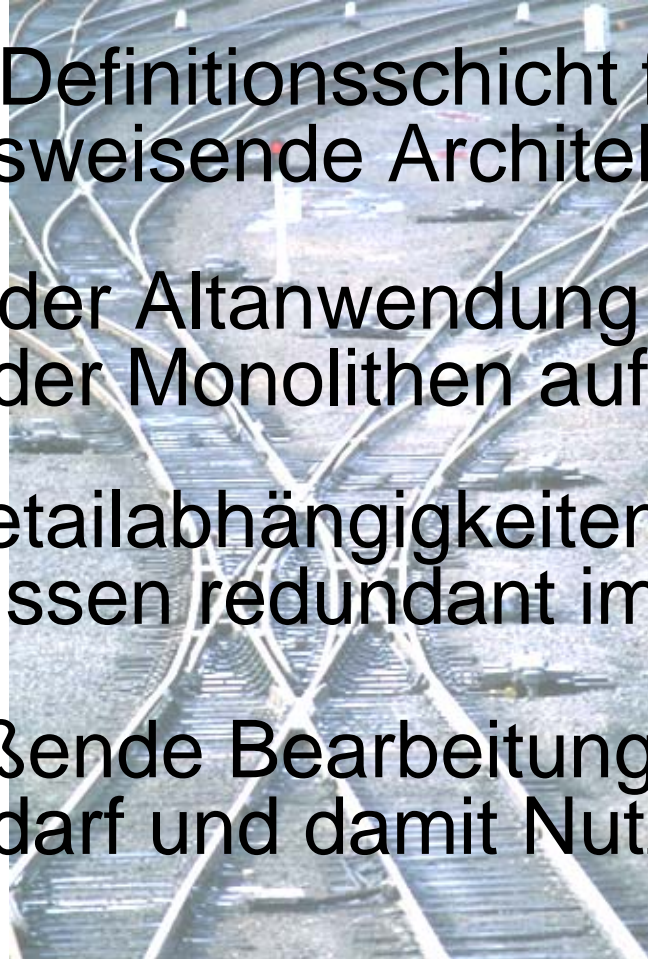


- Operation ist Information Broker
- Storage verwaltet kontextfreie, zeitgestempelte Datenstrukturen

Entflechten mit Workflow

Chancen und Probleme

- Gesonderte Definitionsschicht für Workflow ist eine zukunftsweisende Architektur
- Granularität der Altanwendung selten passend, Aufbrechen der Monolithen aufwendig und riskant
- Fachliche Detailabhängigkeiten zwischen den Schritten müssen redundant implementiert werden
- Fallabschließende Bearbeitung an einer Stelle reduziert Bedarf und damit Nutzen



Schnittstellen-Reengineering

Die Alternativen

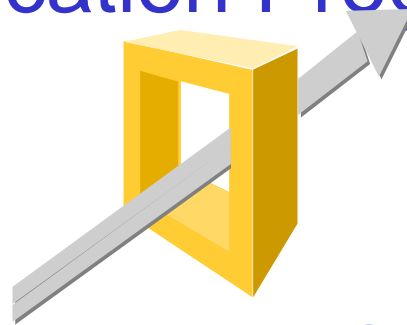
- Gemeinsame Datenbank gilt oft noch als Ideal, ist jedoch nicht machbar



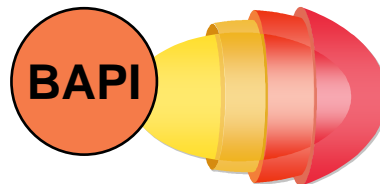
- Periodischer Datenaustausch (eventuell mit Transformationen) ist oft ausreichend
- Message-Schnittstellen („Object Wrapping“) sind zukunftsweisend und realistisch – z.B. BAPI-Konzept der SAP

Was ist ein BAPI?

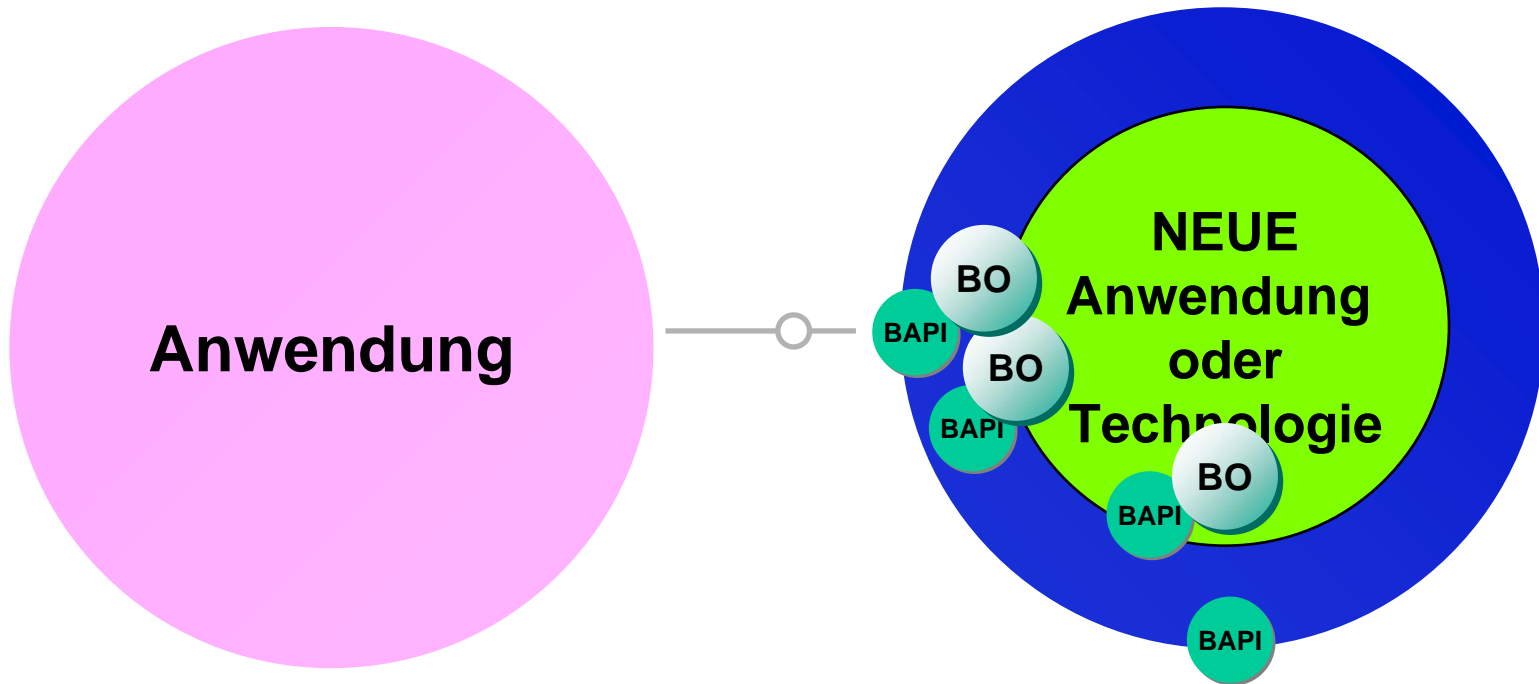
- Business Application Programming Interface



- Ein BAPI ist eine klar definierte **Schnittstelle** zu Prozessen und Daten von **Anwendungskomponenten**, implementiert als **Methode** eines **Geschäftsobjektes** im Business Object Repository (BOR).

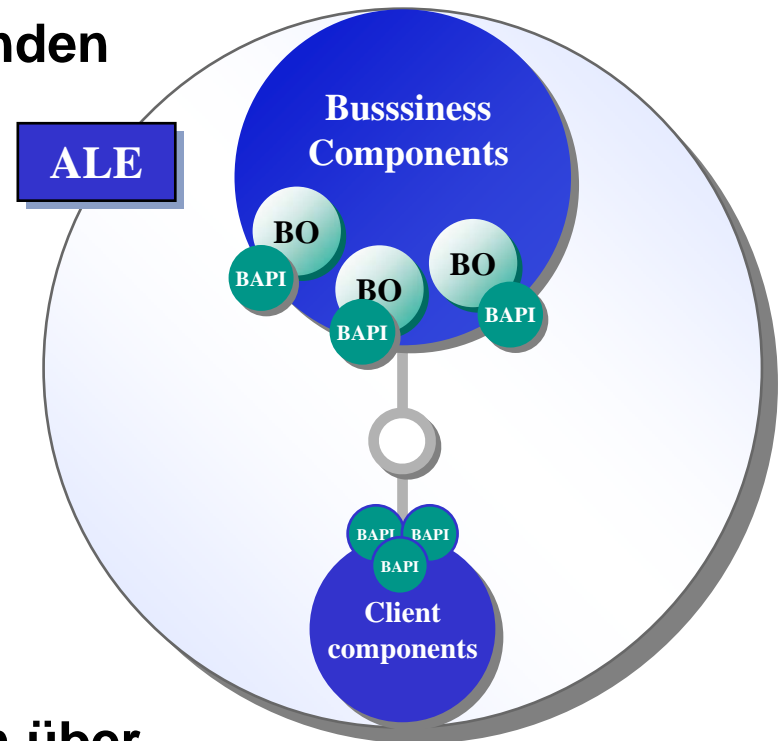


B APIs ermöglichen den Einsatz neuer Technologien ohne Änderungen der bisherigen Anwendungskomponenten



Das SAP-Business Framework

- **Integrierte Familie von Produkten, zusammengesetzt aus kooperierenden Komponenten**
- **Strategische Produktarchitektur**
- **Bestandteile**
 - Business Components (z.B. Human Resources)
 - Business Objects (z.B. Employee, Applicant, ...)
 - BAPI interfaces (z.B. Employee.ChangeAddress)
- **ALE sichert die Prozessintegration über unterschiedliche Komponenten hinweg**



Der Big Bang ist vermeidbar



- Das Internet hat bewiesen, daß heterogene Systeme zusammenarbeiten können

Das Geheimnis:

- Pull-Architektur: Server als Supplier (z.B. Browser oder Remote Function Call)
- Verzicht auf gemeinsames ER-Datenmodell („nur“ semantische Synchronisation)
- Messaging statt funktionaler Koppelung



Vertriebs- und Serviceprozesse
erfordern
Cross-Plattform-Verfügbarkeit
– Wege zum Ziel -

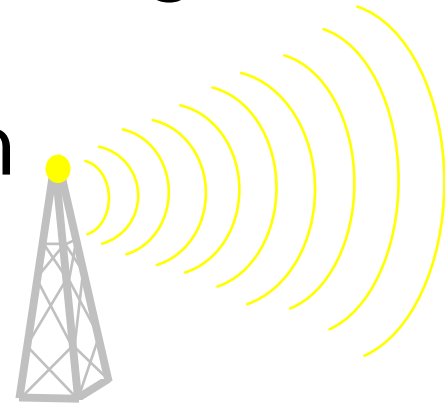
Cross-Plattform-Verfügbarkeit Warum in Zukunft zwingend?

- Versicherungsvertrieb über unterschiedlichste Kanäle (Agenten, Banken, Werkstätten, Kaufhäuser, Internet, Interaktives TV, ...)
- Leistungserbringung durch externe Partner (Assisteure, Werkstätten, Krankenhäuser, ...)
- Gleichzeitige Verfügbarkeit neuer Produkte und geänderter Geschäftsregeln bei allen Partnern

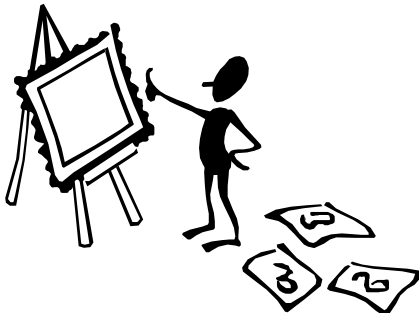
Cross-Plattform-Verfügbarkeit

Die technischen Herausforderungen

- Business-Daten und Logik müssen bei allen beteiligten Partnern 1:1 verfügbar sein



- Die richtige Oberflächen für jeden Anwender



☞ Heterogene Technologie
wird Normalfall

Cross-Plattform-Verfügbarkeit

Wie erreichbar?

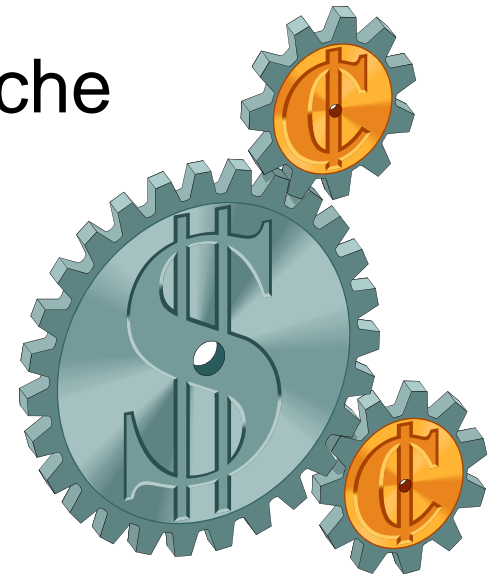
- Die JAVA-Architektur und das Internet sind die Paradigmen der nächsten Softwaregeneration
- Implementierung der Business Logik nach dem Prinzip „write once, run anywhere“ wird zum Erfolgsfaktor

Im Prinzip die „alte“ Tabellensteuerung plus Prozeduren?

Cross-Plattform-Verfügbarkeit

Beispiel: Es geht nicht nur mit JAVA (1)

- Anwendungssystem greift auf VP/MS-RunTime zu (produktneutrale, plattformspezifische „virtuelle Maschine“)
- RunTime serviciert Anwendung durch Interpretieren der Produktdatei (produktspezifischer, plattformneutraler „Byte-Code“)

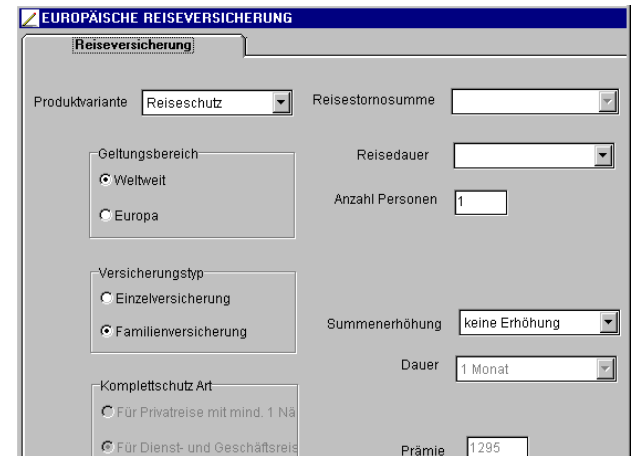


Cross-Plattform-Verfügbarkeit

Beispiel: Es geht nicht nur mit JAVA (2)

- ◆ Produkte werden mit VP/MS einmal definiert und den Anwendungen zur Verfügung gestellt, z.B.

- ◆ Windows 3.x, 95 und NT
- ◆ Unix
- ◆ Internet
- ◆ MVS
- ◆ VM/ESA



HOST	
BEARBEITUNG / VERTRAGSDATEN Neugeschäft (2111) AM: 17.10.97/11.	

Polizzen-Nr.: 8/71/82021087-01<	
NEUANTRAG REISESCHUTZ	

Versicherungsart	: 11+
Anzahl der Personen	: 4*

VERS.	Reisedauer : 5+
	Geltungsber. : 01000000-
	Versich.typ : 00000500-
	Stornoschutz : 00010000-
	Prämie : 00000500

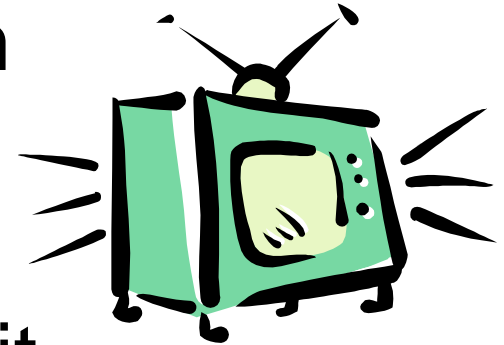
Nachname	:
Adresse	: +
Kontonummer	: +

Nächstes Bild (Fortsetzung Spartenenteil):	

Cross-Plattform-Verfügbarkeit

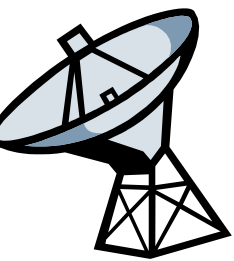
Alle sind immer Online?

- Netze werden leistungsfähiger, aber die Anforderungen steigen auch (z.B. Grafik, Sound, Video)



Remote-Anwendungen mit

- periodischem Datenaustausch (Replikation)
- temporärer bzw. asynchroner Koppelung (RFC) bleiben trotz Internet, GSM etc. relevant



Cross-Plattform-Verfügbarkeit ^{M+I}

Neue Herausforderung: Logistik

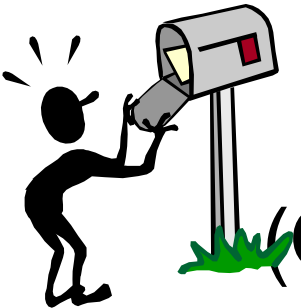
- Dateigrößen und segmentierte Updatemöglichkeit werden zum Thema



(Moore'sches Gesetz auch für Software?)

- Management der Daten- und Programmlogistik wird zum Erfolgsfaktor

(Qualitätsspektrum: gelbe Post ↔ FedEx)



Modulare, objektorientierte
Anwendungen bauen mit
ausgereiften Technologien –
pragmatische Lösungsansätze

Technologie ist ein Hygienefaktor

- Schlechte Technologien machen Unternehmen krank

aber ...

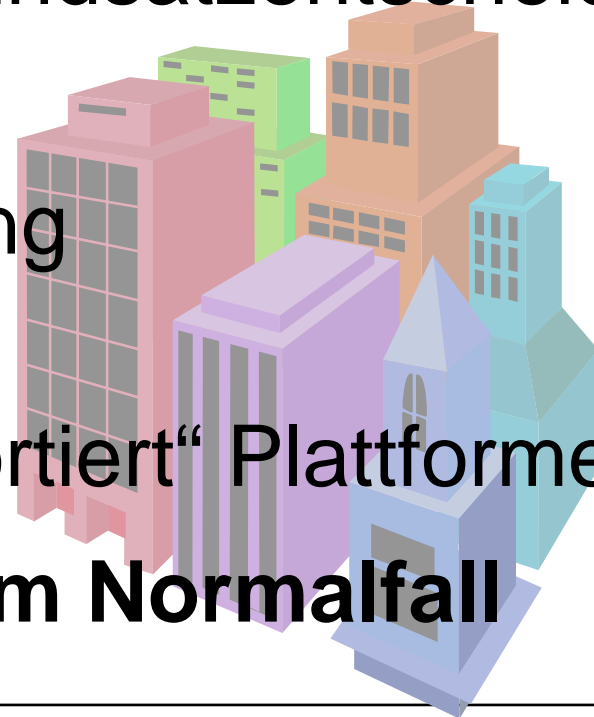


- von guten Technologieentscheidungen allein wird ein Unternehmen nicht gesund und leistungsfähig



Auswahlkriterien für Technologien (1)

- Die Auswahl der Softwaretechnologie ist keine global gültige Grundsatzentscheidung mehr
- Je nach Aufgabenstellung unterschiedlich
- Standardsoftware „importiert“ Plattformen
- ☞ **Heterogenität wird zum Normalfall**



Auswahlkriterien für Technologien (2)

- Internettechnologie ist (wo immer verfügbar) vorzuziehen (z.B. TCP/IP)
- Taktisch bedingte Abweichungen sollten klar gekapselt und damit substituierbar gemacht werden
- Die Qualifikation der Entwickler wird zu einer wesentlichen Determinante

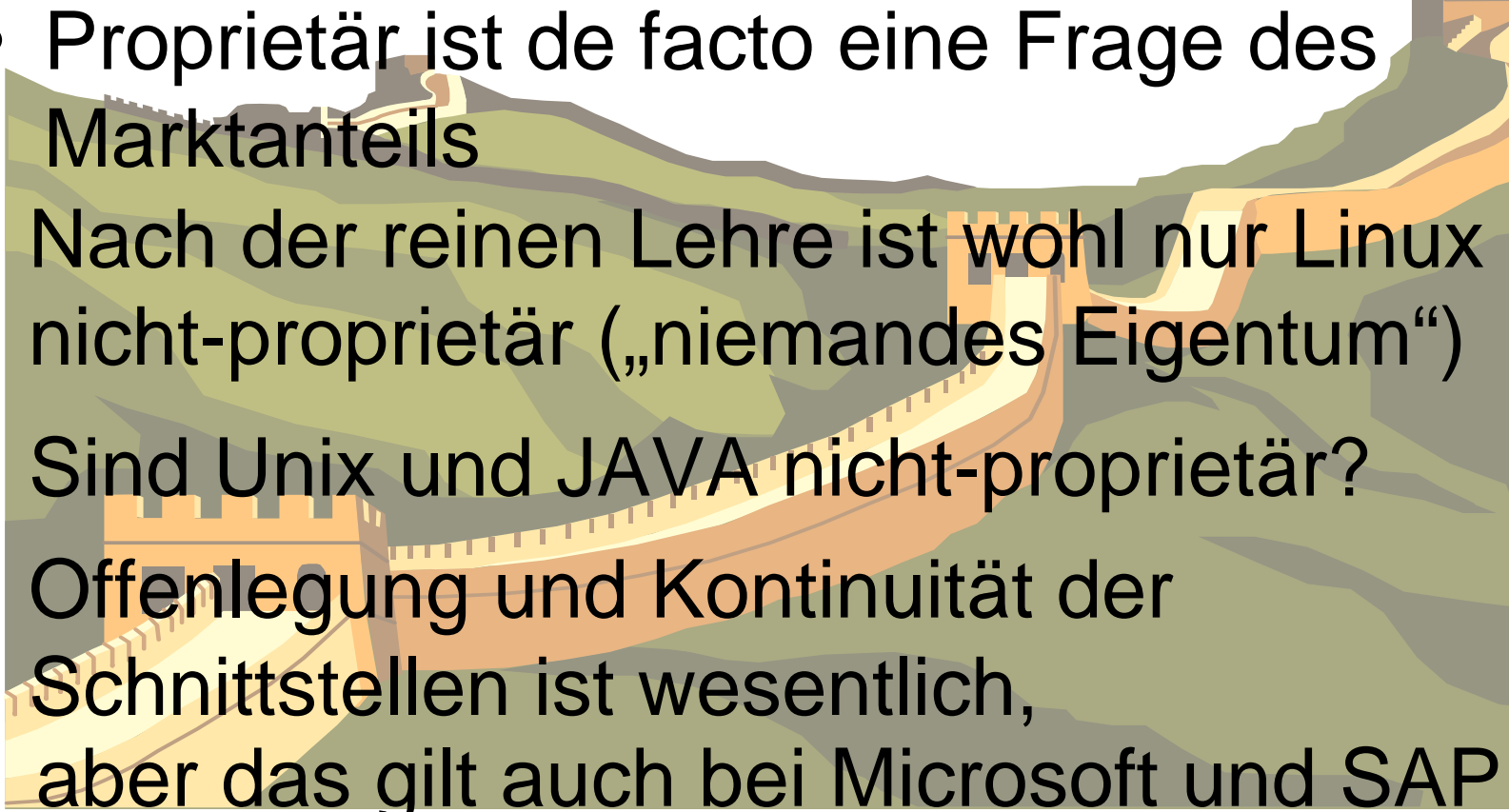


Neue Schichtenarchitektur der Geschäftslogik

- Logische Schichtung/Modularisierung:
 - Definition
 - Operation und
 - Storage
- Definitionen können nicht auf Daten reduziert/flachgewalzt werden

Objektorientierung daher unabdingbar

Proprietär - sinnvolles Kriterium?

- 
- Proprietär ist de facto eine Frage des Marktanteils
 - Nach der reinen Lehre ist wohl nur Linux nicht-proprietär („niemandes Eigentum“)
 - Sind Unix und JAVA nicht-proprietär?
 - Offenlegung und Kontinuität der Schnittstellen ist wesentlich, aber das gilt auch bei Microsoft und SAP

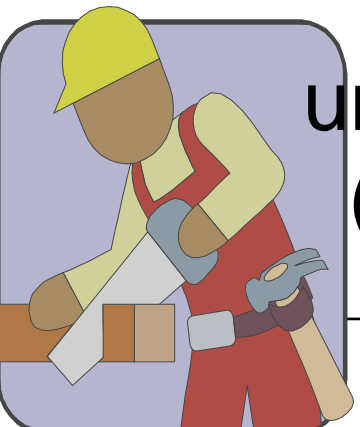
Make and buy

- Technische Komponenten grundsätzlich kaufen



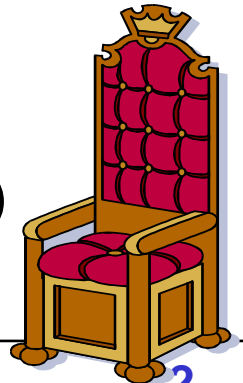
- Marktgängige Anwendungsbausteine kaufen (wo immer vorhanden)

- Eigenentwicklungen nur noch als unternehmensspezifische Integrationsmittel (Wettbewerbsvorteil, Differenzierungsmittel)



Objektorientierung?

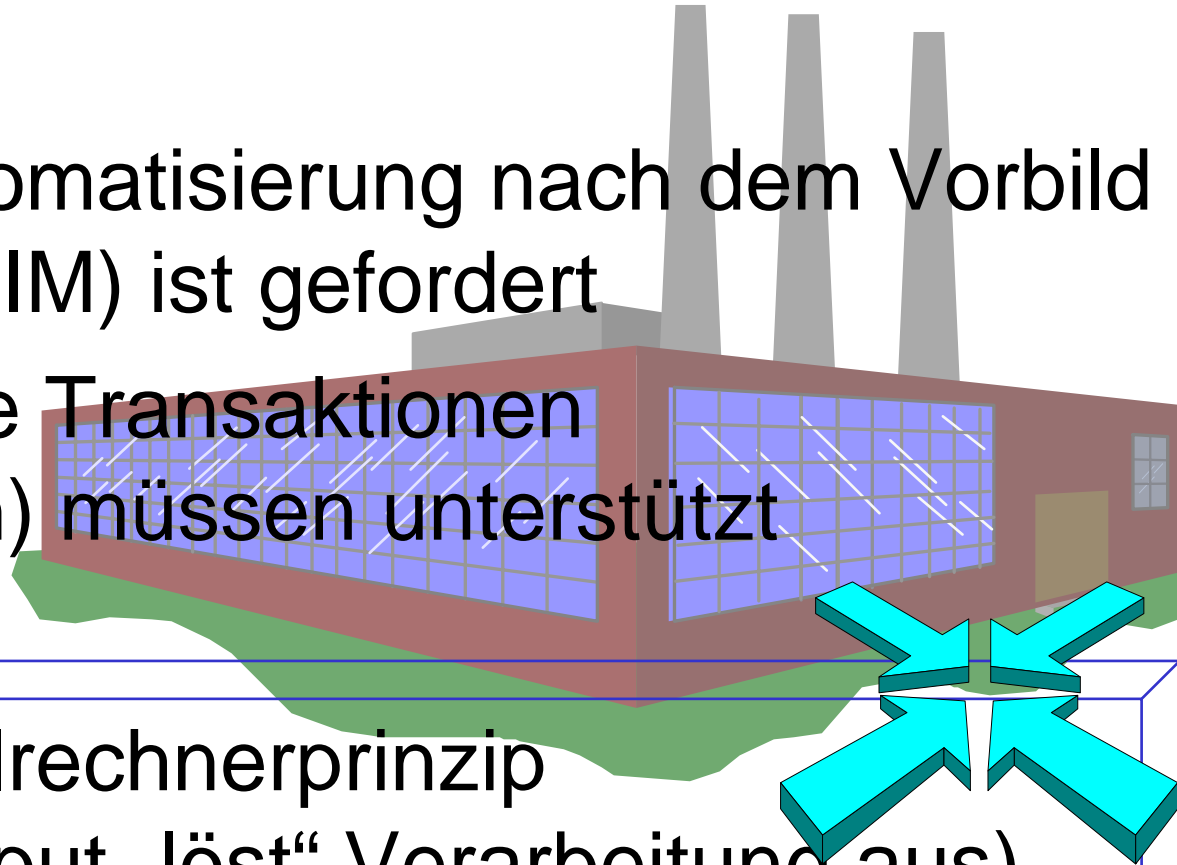
- OO als Architektur steht außer Streit
- OO als spezifische Technologie ist noch selektiv zu handhaben
- OO-Datenbanken wegen Massendatenverarbeitung (noch?) kritisch zu sehen
- Datenhoheit für Applikationen ist die pragmatische Umsetzung von OO in die relationale Welt



Transaktionsmanagement

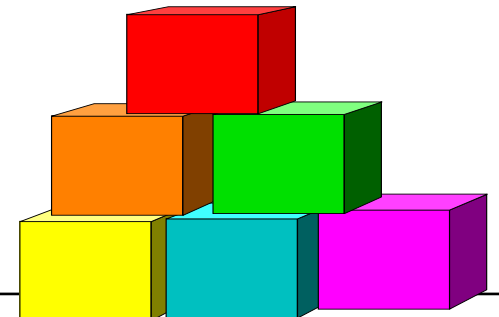
- Hintergrundautomatisierung nach dem Vorbild der Industrie (CIM) ist gefordert
- Lange, parallele Transaktionen (Schwebedaten) müssen unterstützt werden

☞ Lösung: Parallelrechnerprinzip (vollständiger Input „löst“ Verarbeitung aus)



IAA

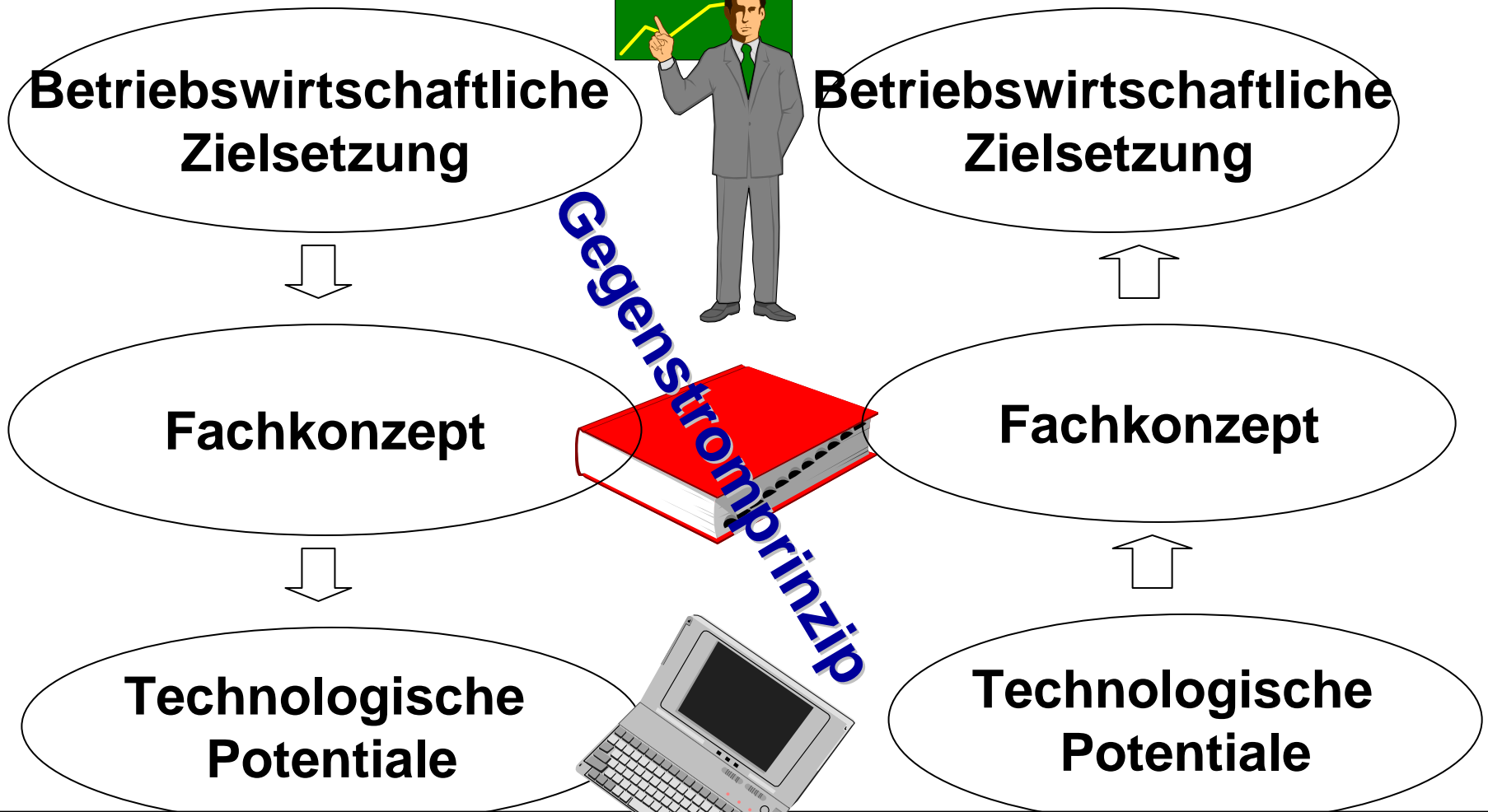
- IAA hat nicht funktioniert, weil
 - keine Lösung damit verbunden war
 - egoistische Anbieterinteressen verfolgt wurden
- Inhaltliche Schwächen
 - unrealistisches Ideal der durchgängigen Daten- und Funktionsmodellierung
 - keine Komponentenphilosophie



VAA

- VAA wird nur funktionieren, wenn Lösungen dahinter stehen
- ☞ Man müßte Lösungen mit hinreichender Marktdurchdringung und Qualität zum Standard erheben
- Fundamentalkritik von Prof. Raasch: (auch) VAA folgt einem veralteten Paradigma (v.a. keine Datenhoheit von Komponenten)

Architekturentwicklung als Prozess (1)



Architekturentwicklung als Prozess (2)

- Kreative Auswahl, Kombination und Ergänzung von Technologien/Zukaufkomponenten ist der Erfolgsfaktor
- Das Wasserfallmodell in allen Varianten ist tot bzw. tödlich!
- Die Entwicklung von Anwendungen muss nach dem Gegenstromprinzip erfolgen, sonst bleiben die Chancen neuer Technologien ungenutzt