Tun wir das Richtige, tun wir das richtig?
Integrationsmanagement – Zukünftige Anforderungen an Technik und Management

1. Ziele der Integration
2. Integrationsformen
3. Strukturverbessernde Wartung
4. Management für Integrationsaufgaben
5. Zusammenfassung: Thesen

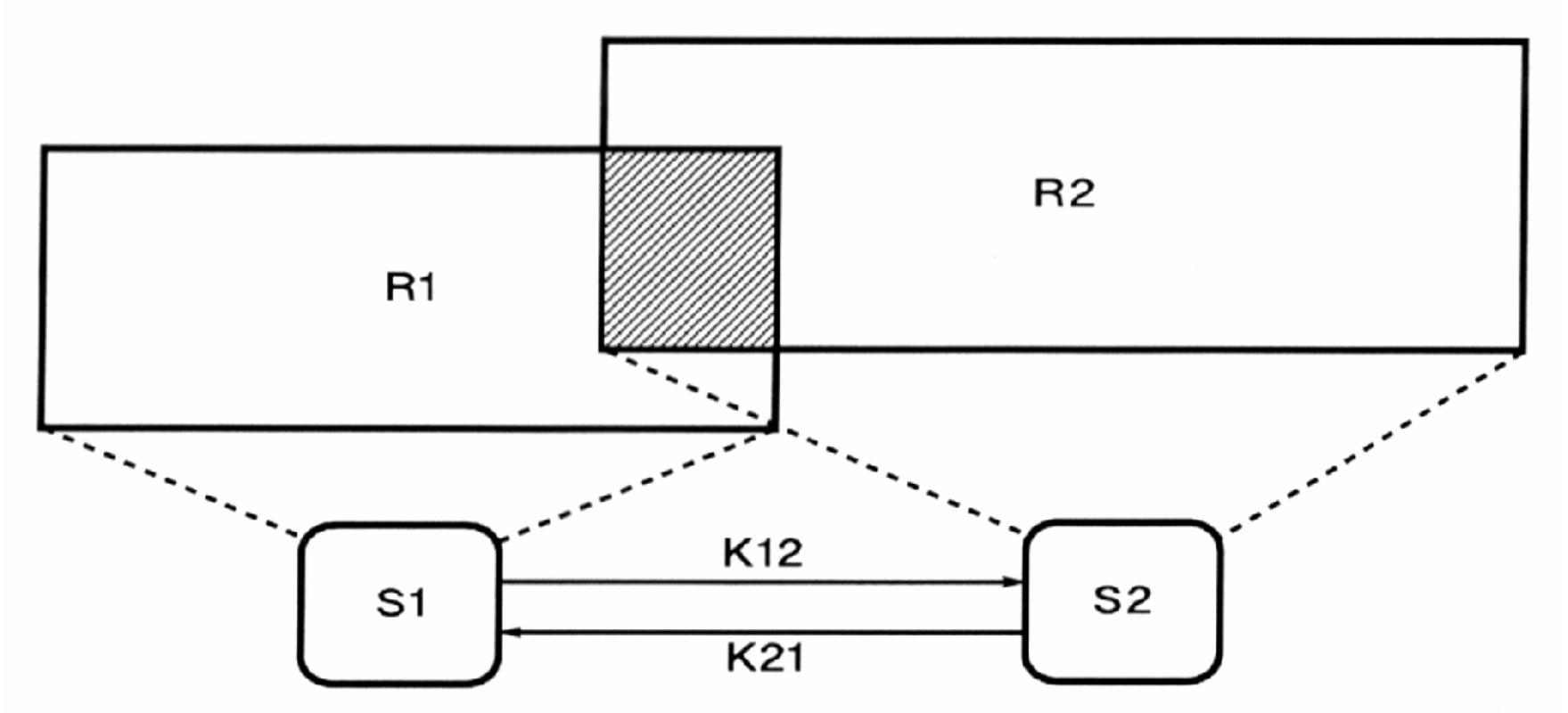
© Prof. Dr. Jörg Raasch

raasch@informatik.haw-hamburg.de
www.informatik.haw-hamburg.de/~raasch

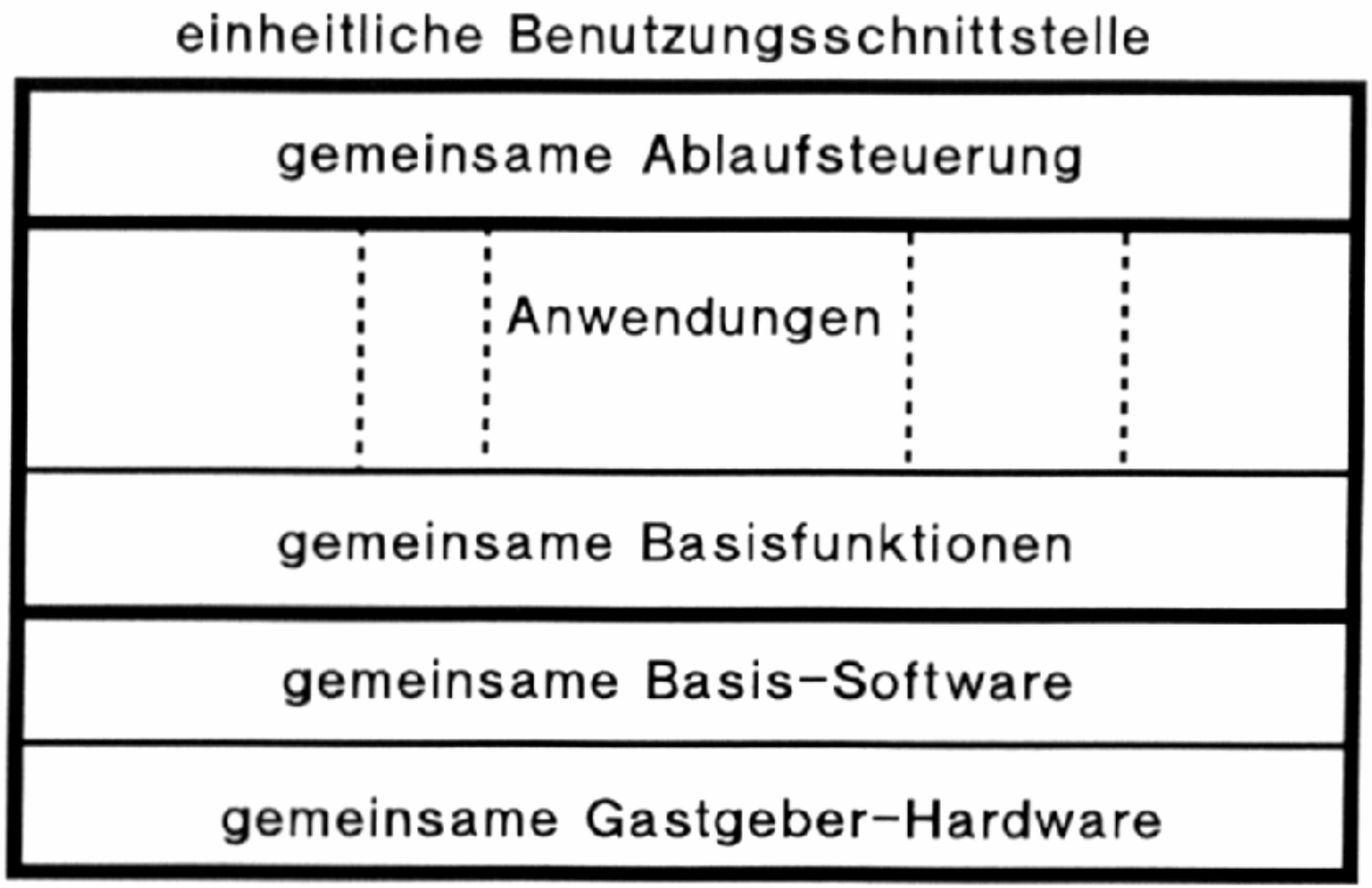
1. Ziele der Integration

- Wirtschaftlichkeit
- Vermeidung von Insellösungen
- Einheitliche Workflows, einheitliche Außenwirkung
- Schnellstmögliche Sachbearbeitung, Datenaktualität
- Daten- und Methoden-Konsistenz
- Vereinfachte Geschäftsprozesse
- Redundanzvermeidung
- Fehlervermeidung

Datenintegration



Funktionale Integration



Integration von Anwendungen

- **Integration** = Einbindung bestehender oder zu entwickelnder Systeme in einen neuen Kontext
- **Migration** = Weiterentwicklung bestehender Systeme in Richtung auf eine Zielvorstellung (Vision)
- Integration von
  - Legacy-Systemen
  - kommerziellen Komponenten, Fremdsysteme (z.B. SAP)
  - anderen Verfahrensteilen
- Wie sollte die SW-Landschaft aufgebaut sein?
  - Nicht monolithisch!
  - Sondern Komponentenarchitektur.

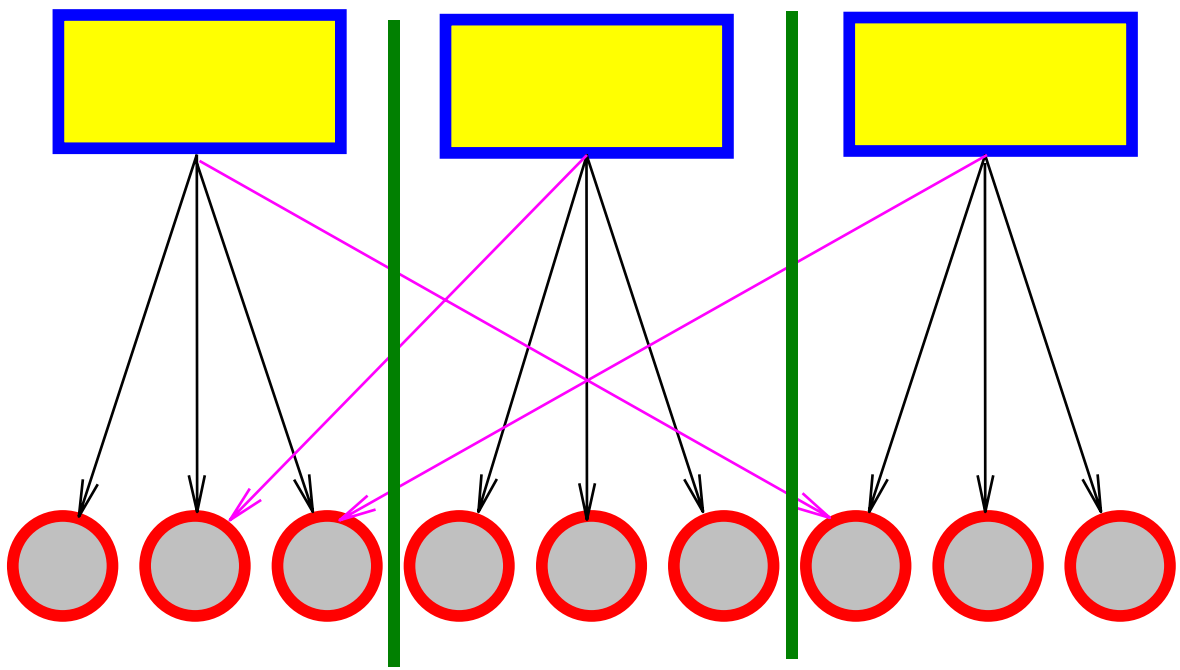
2. Integrationsformen

1. Disjunkt
2. Überlappend
3. Nutzung von Fremdsystemen

4. Probleme der Integration?

2.1 Disjunkte Integration

- Subsysteme haben disjunkte, jedoch angrenzende Aufgaben
- Klare, explizite Schnittstellen erforderlich
- Komponentenarchitektur
- Problem: in Legacy-Systemen diffuse Schnittstellen

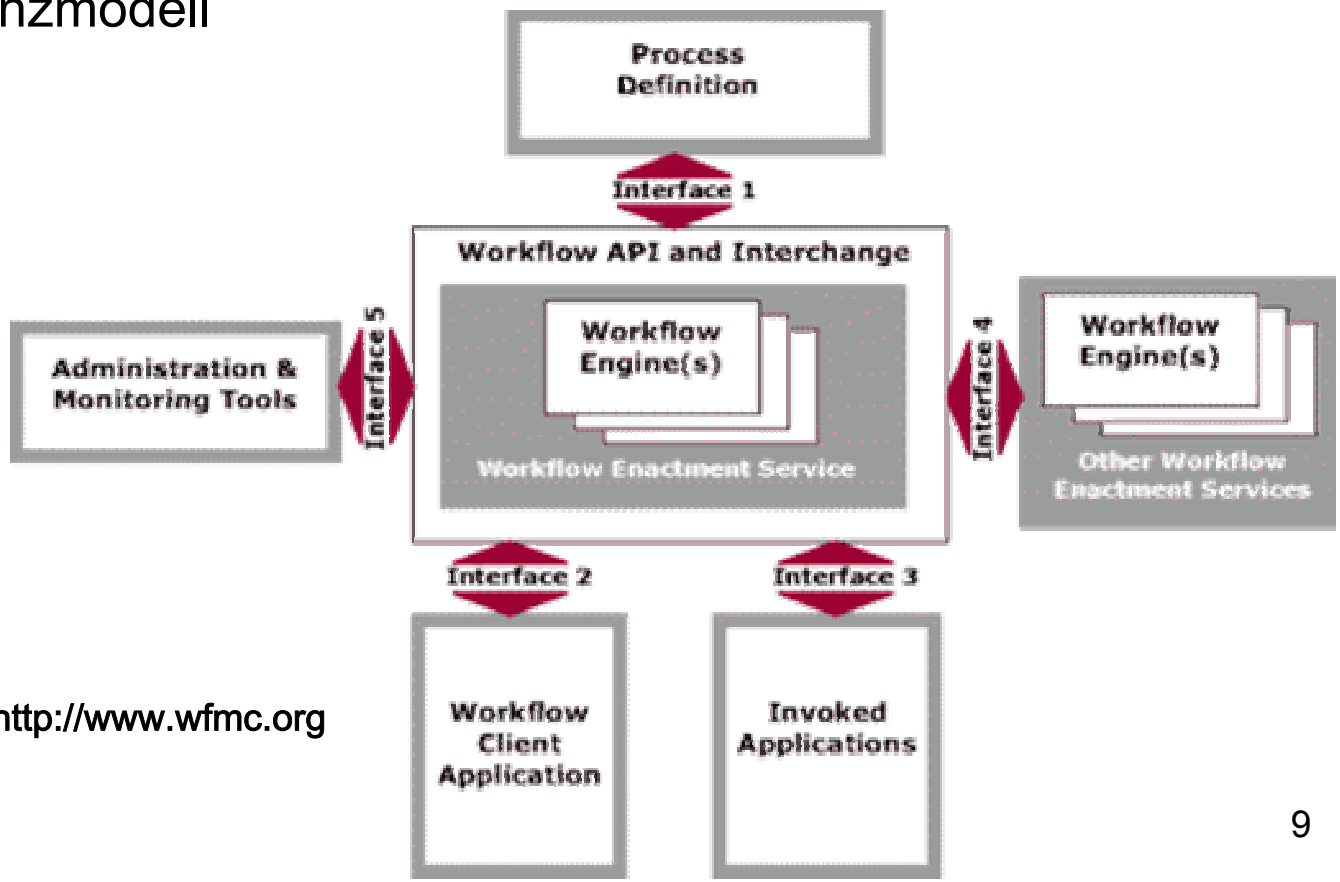


2.2 Überlappende Integration

- Die Aufgabenaufteilung der Subsysteme ist nicht disjunkt, Kooperation mehrerer Unternehmen oder Abteilungen in einem Geschäftsprozess.
- Beispiele:
  - B2B (über Unternehmensgrenzen hinweg)
  - In einem Unternehmen werden mehrere unterschiedlich spezialisierte WFMS eingesetzt, die aber in manchen Geschäftsprozessen zusammenwirken müssen
  - Zwei Organisationseinheiten machen (fast) dasselbe, benutzen dazu eigene Anwendungen mit (fast) gleicher Semantik (Fusionsproblematik).

Koordination zwischen WFM (WF-Interoperabilität)

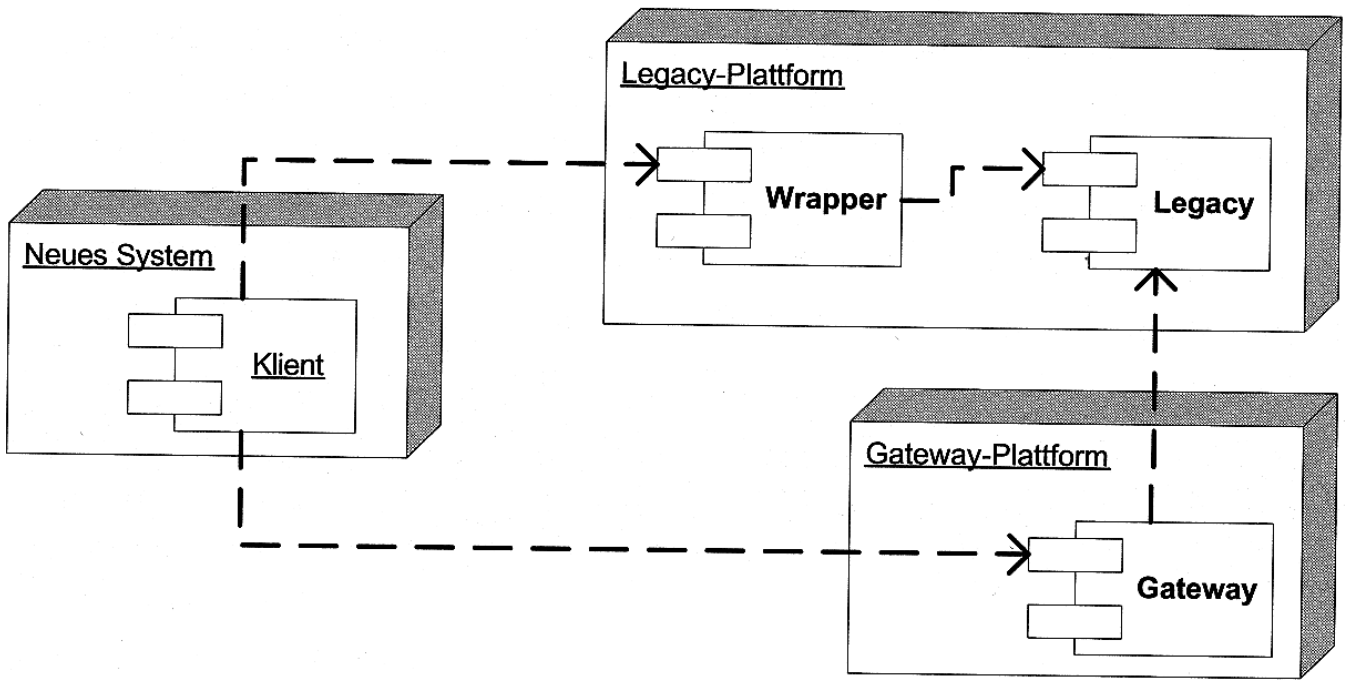
- Mehrere WFMS arbeiten im Peer-to-Peer-Modus,
- erscheinen aber nach außen wie *ein* System
- und sind bestmöglich entkoppelt
- Im WfMC-Referenzmodell
  Interface 4



Quelle: WfMC-Website http://www.wfmc.org

2.3 Nutzung von Fremdsystemen

- Ein Problem: Abhängigkeit von fremden Weiterentwicklungszyklen
- Zumeist andere Benutzungsschnittstelle, unpassende Schnittstellen etc.
- Daher ist „Verstecken“ erforderlich:



2.4 Probleme der Integration: Legacy-Systeme

- Trennung Interaktion / Funktion / Datenhaltung ?
- Optimierung von Speicherplatz und Performance auf Kosten der Wartbarkeit?
- Quellcode oder Dokumentation liegt nicht mehr vor
- Implementierte Plausibilitätsprüfungen verbinden oft mehrere Komponenten und führen zu starken Abhängigkeiten
- Fehler- und Ausnahmebehandlung ...

Fachliche und organisatorische Probleme der Integration

- Mitarbeiter überblicken nur ihren Bereich (wg. Taylorisierung)- **Gesamtsicht erforderlich**
- Integration erfordert oft **Leistungen für andere**, über die eigene Aufgabe hinaus
- **Standardisierung** ist Voraussetzung für Integration, wird aber abgelehnt (begrenzte Flexibilität)
- **NIH**-Syndrom.

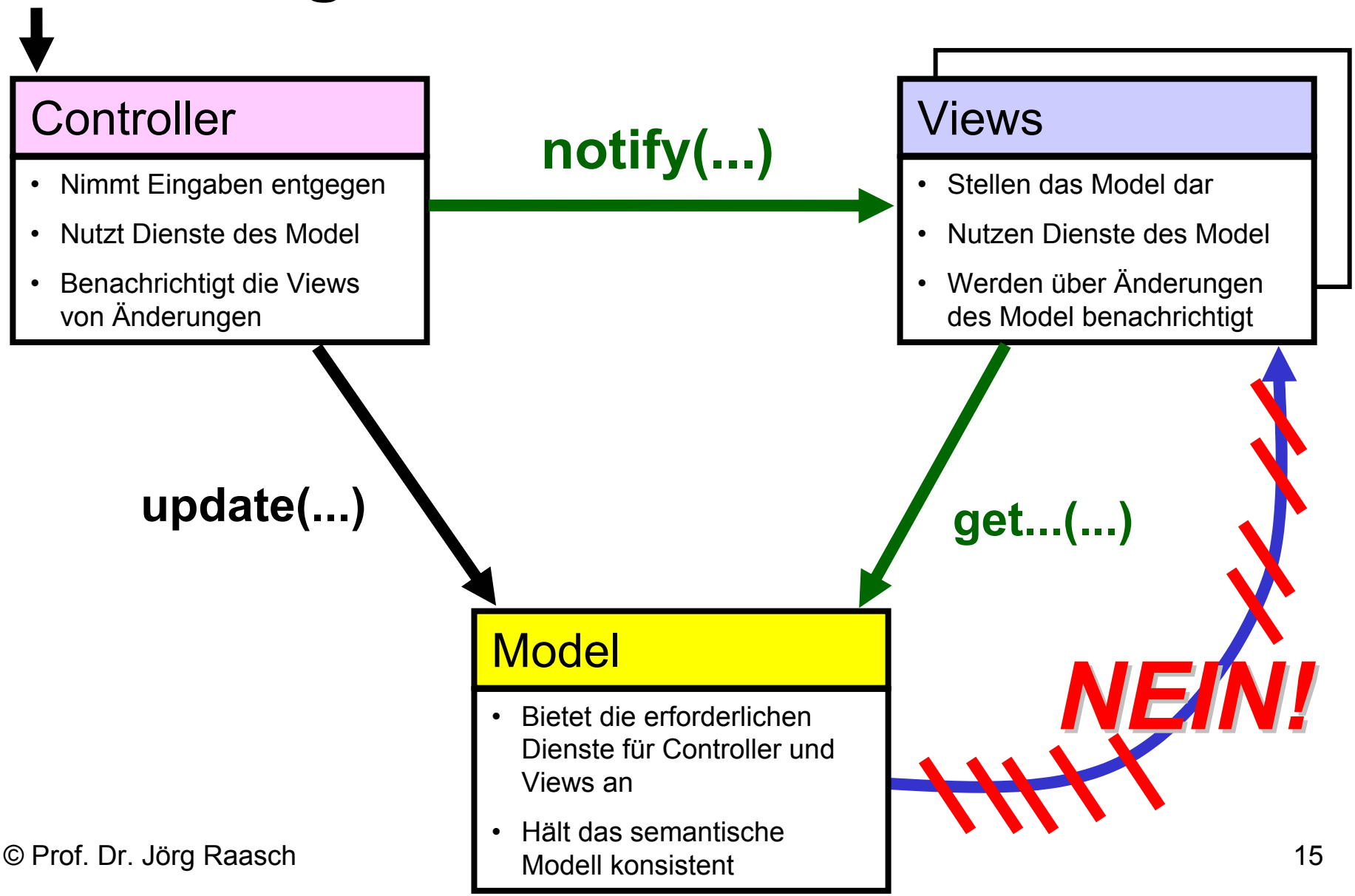
Technische Probleme der Integration

- Single Login, übergreifende Rechteverwaltung
- Harmonisierung der Geschäftsprozesse, anwendungsübergreifendes Workflow-Management
- Einheitliche Benutzungsschnittstellen
- Schnittstellen zwischen den Systemen
- Historisierung, Ordnungsmäßigkeit, Revisionierbarkeit
- Einheitliche Fehler- und Ausnahmebehandlung
- Teilsystemübergreifende Transaktionsverarbeitung
- Datenschutz und Systemverfügbarkeit
- Unterschiedliche Sicherheitsmechanismen
- Administration

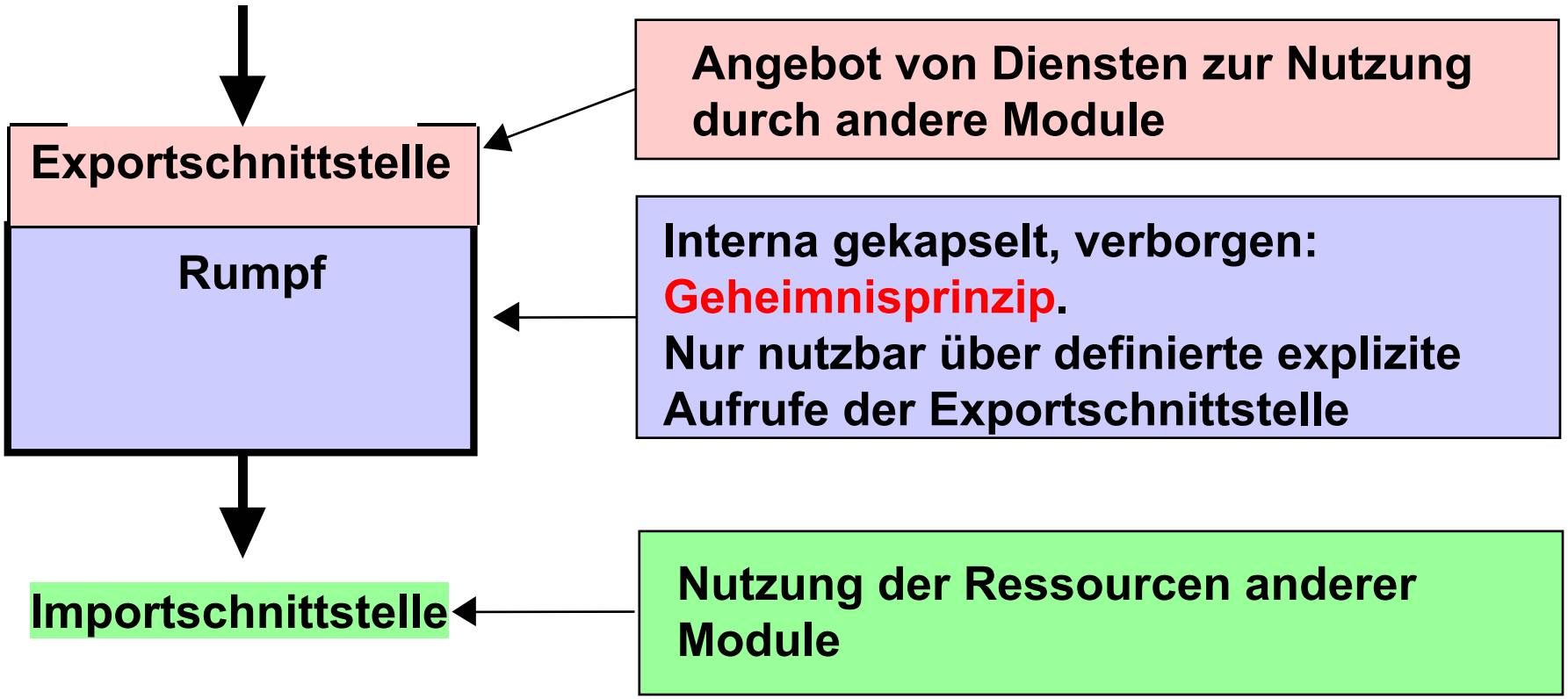
4. Technische Lösungsansätze

- Trennung und Rückkopplung Interaktion-Funktion
  - Migration auf andere Plattform ermöglichen
  - Benutzungsschnittstellen erneuern ohne die Funktionalität neu zu fassen
  - Funktionalität in neue Geschäftsprozesse einbinden

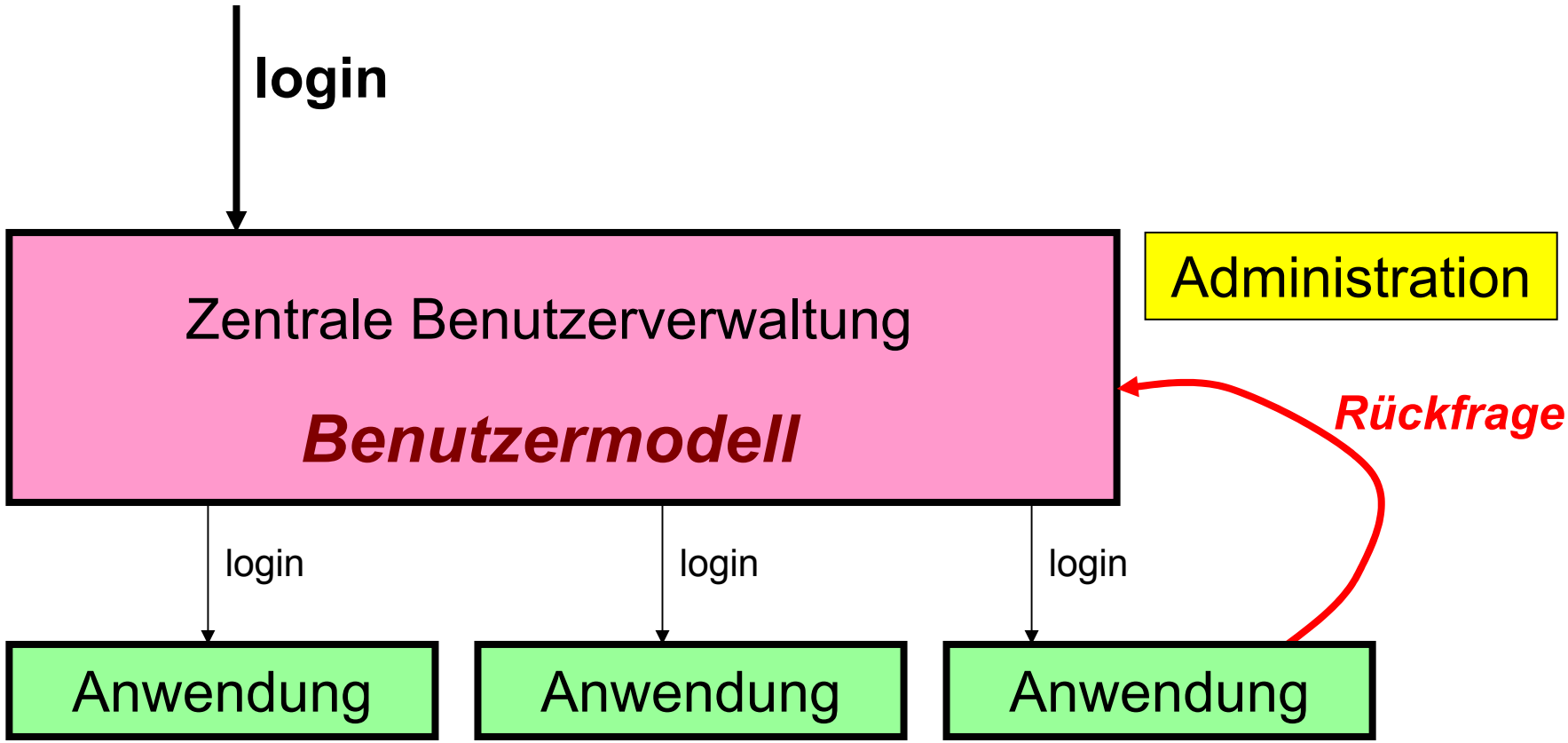
Trennung Interaktion – Funktion: MVC



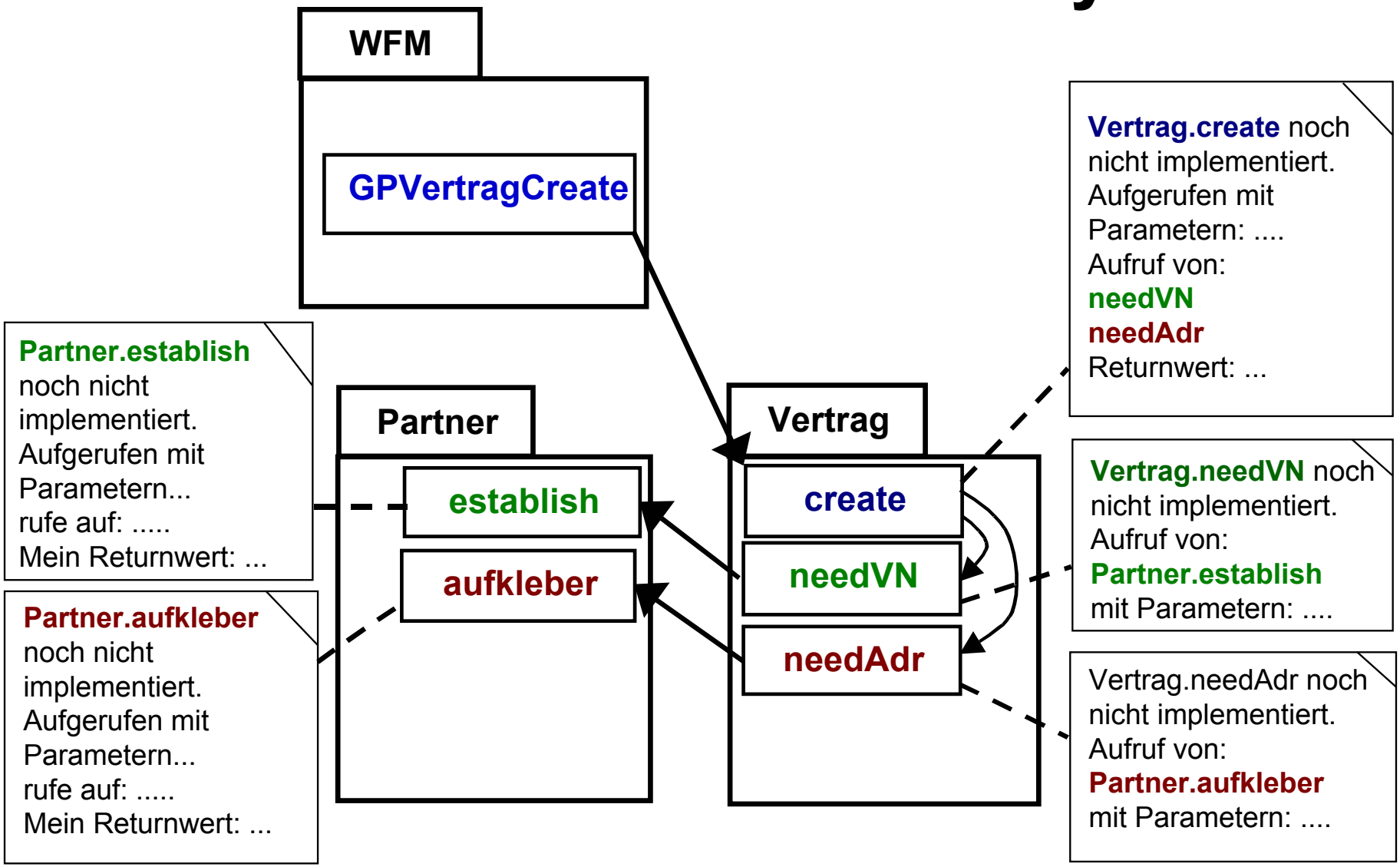
Modulkonzept, Information Hiding, Import- und Exportschnittstelle



Rechtesystem für Single-Login



Schnittstellen zwischen Subsystemen



Beispiel: Styleguide-Nutzung

PartnerIF

```
establish: aClassName owner:
aPersistentRef
"post: returns a reference ...
"pre: "
self ASSERT:
        (self catalogue
        includes: aClassName).
"code:" .....
```

```
catalogue
"returns a set of names which
can be used by clients"

^Partner subclasses asSet
        collect:
        [:each | each name]!
```

VertragIF

```
needVN: aVVertragRef
"IMPORT: ein VN soll
eingerichtet werden, seine
Referenz soll zurueckgegeben
werden"

(PartnerIF catalogue
        includes: 'Versicherungsnehmer')
        ifFalse: [ ... Fehlerbehandlung ... ].
^PartnerIF
        establish:
        'Versicherungsnehmer'
        owner: aVVertragRef
```

4. Strukturverbessernde Wartung

- Eingrenzung der Aufgaben im Wartungsprojekt auf Requirements und Fehler bedeutet:
  das System wird ständig verändert, ohne daß seine Struktur dabei im Hinblick auf eine Zielvorstellung *gestaltet* wird
- Der erste Entwurf hat immer eine durchdachte Struktur.
  Entwicklung = Einführung von Ausnahmebehandlungen?
- System wird ohne Strukturverbesserungen „tot-gepflegt“
- Streß $\Rightarrow$ Dokumentation verliert an Qualität.

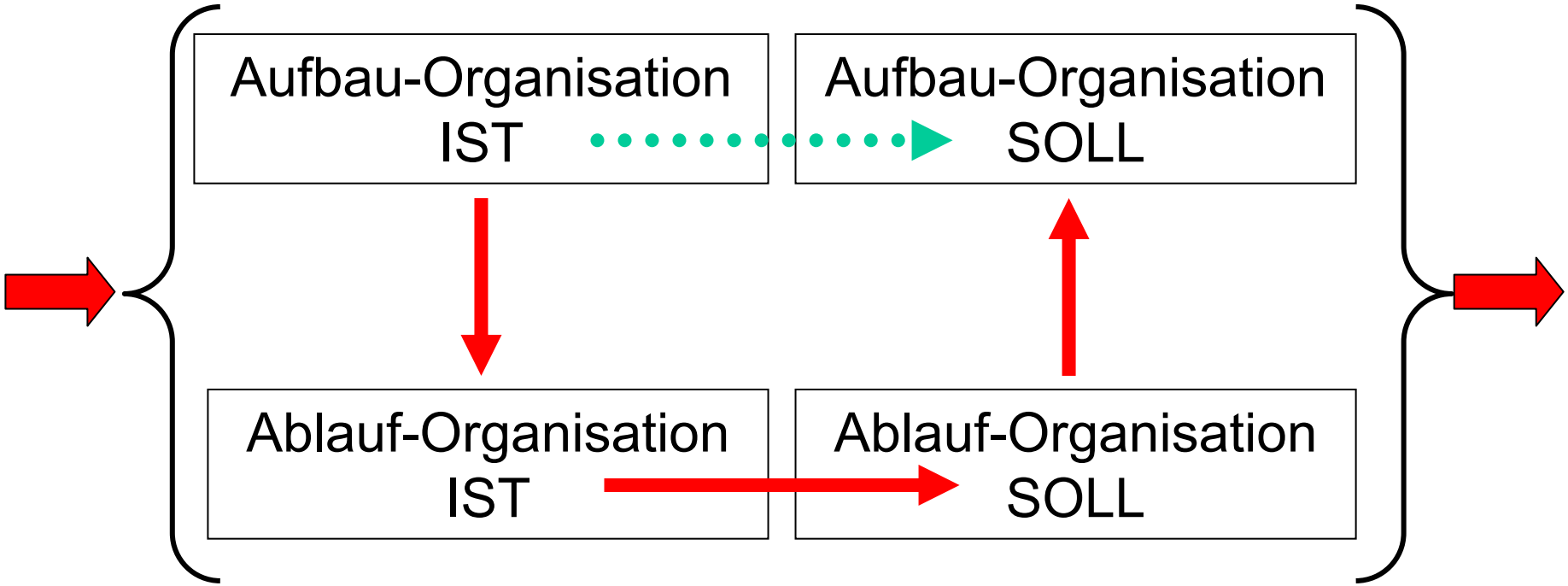
Was ist strukturverbessernde Wartung?

- **eine Vision entwickeln, wie das System künftig aussehen soll**
- jede Maßnahme im Hinblick auf die Vision bewerten, akzeptieren oder ggf. ablehnen
- ≥ **20 %** der Ressourcen nutzen für interne Strukturverbesserungen (von denen der Anwender/Kunde nichts sieht !) und Konvergenz zur Vision (Migration)
- Wartung ist dann gleichzeitig „schleichende Migration“
- Einbettung in das IT-Gesamtkonzept (z.B. Prioritätensetzung nach Renovierungsbedarf).

5. Management für Integrationsaufgaben

- Versionen und Konfigurationen bilden einen wichtigen Maßnahmen-Kontext
- Keine Neuentwicklung möglich: dann eben strukturverbessernde Weiterentwicklung
  - Visionen erarbeiten (lassen)
  - X % des Budgets für Strukturverbesserung (die der User nicht sieht)
- Beziehungen zwischen Organisation und Architektur?

Aufbau- und Ablauf-Organisation



Kein Wasserfall!

- Neue Organisation erfordert neue Werkzeuge
- Neue Werkzeuge erfordern neue Organisation

- Daher ist die Organisationsuntersuchung und –Verbesserung nicht aus dem SW-Werkzeug-Projekt herauszulösen, sondern intensiv verknüpft

Conway‘s Law

- Organisation des Fachbereichs prägt sich durch auf die SW-Architektur

[Cockburn 2002]
Alistair Cockburn: On the Interaction of Social Issues and Software Architecture. Webseite (über Suchmaschinen finden)

- daher: Aufbau und Abläufe integriert gestalten
  - Arbeitsplätze entwerfen, aber mit Blick auf die technischen *Möglichkeiten und Randbedingungen*
  - Werkzeuge entwickeln
  - Unternehmensmodellierung
    - Zielsetzungen und Verantwortlichkeiten
    - Reengineering und Modellierung der Geschäftsprozesse
    - Unterstützende statt ablaufsteuernde Sichtweise.

Was ist „kgV“?

- kgV= der „kleinste gemeinsame Vorgesetzte“
  - muß existieren ☺
  - muß wissen, dass er existiert ☺☺
  - muß Verbindlichkeit erzeugen
  - muß daher Weisungskompetenz haben
  - muß Probleme lösen
    - nicht die technischen
    - sondern Kommunikation, Kooperation
  - muß die Zusammenarbeit fordern und fördern
  - hat letztlich die Verantwortung, auch im Sinne der Aufgabendelegation
  - behält die Verantwortung auch, wenn er delegiert (z.B. Steuerungsgremium)
  - die wichtigste Person für Entwicklung, Integration, Weiterentwicklung.

6. Zusammenfassung: Thesen

1. Integrationsprobleme erfordern den kgV, der aber die technischen Aspekte delegieren sollte
2. Die engen Beziehungen zwischen Organisation und Architektur machen Neugestaltung schwierig
3. Disjunkte Integration ist verbreitet, daneben kommt überlappende Integration vor, neues Thema mit anderen Lösungsansätzen
4. Integration erfordert einige anspruchsvolle technische Lösungsansätze
5. Da Neuentwicklungen seltener werden: strukturverbessernde Weiterentwicklung. Entsprechende Budgetgestaltung und Projektverantwortung
6. Wartungsteams (Weiterentwicklung per Zuständigkeit) eher vermeiden.