

---

# Enterprise Application Integration

Prof. Dr. Christian Pape

---

# Übersicht

---

- Überblick
  - Literatur
  - Was ist EAI?
  - Warum EAI?
- Anwendungsfelder

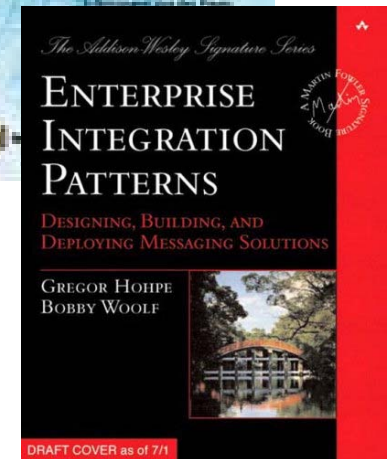
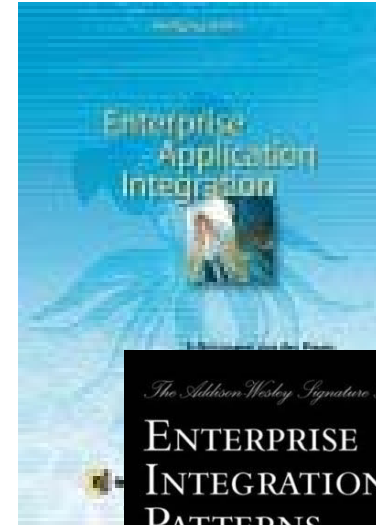
# Themen

---

- Übersicht, Beispiele EAI
- Wiederholung / Vertiefung Verteilte Systeme
  - Referenzarchitekturen
  - J2EE Entwurfsmuster
  - CORBA
- EAI Produkte
- EAI Entwurfsmuster
- Evtl. praktische Übung zum Vorlesungsende

# Literatur

- Wolfgang Keller (Einführung)  
„Enterprise Application Integration“  
dpunkt Verlag
- Gregor, Hohpe, Bobby Woolf  
„*Enterprise Integration Patterns*“  
Addison-Wesley
- Vorlesungsfolien, Übungsblätter:



<http://www.home.hs-karlsruhe.de/~pach0003/eai.html>

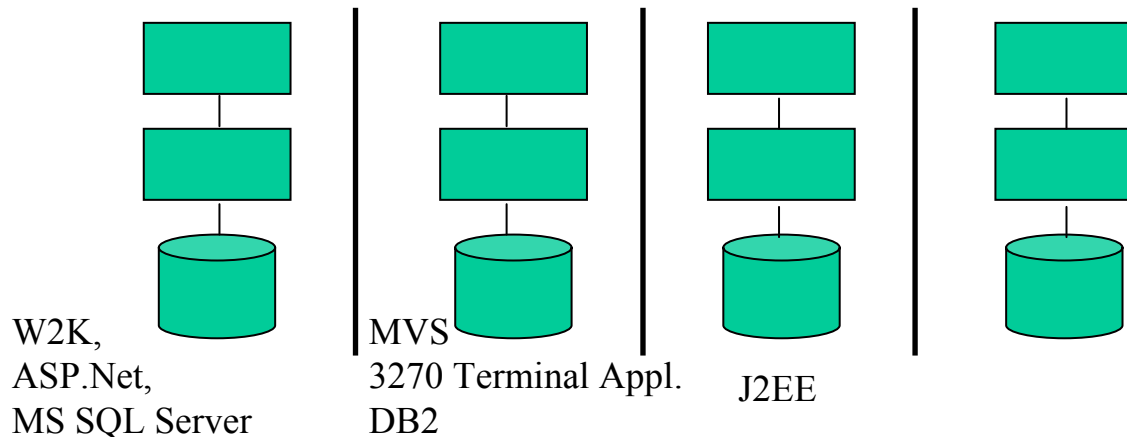
# Was ist EAI?

---

- Allgemeine Definition existiert nicht
- EAI: Enterprise Application Integration
  - Enterprise: Unternehmen
  - Application: Softwareanwendungen
  - Integration
- EAI
  - *Heterogene* Softwareanwendungen eines Unternehmens zusammenbringen, um unternehmensweite *Geschäftsprozesse* einheitlich abzubilden
- Focus Vorlesung
  - Heterogene Technik, nicht Prozesse

# Was ist EAI?

- *Heterogene Technologien*
  - Homogene Systeme einfach zu integrieren
  - Alles J2EE Anwendungen mit EJBs, dann kann EJB einer Anwendung problemlos von anderer verwendet werden
- *Ändernde Geschäftsprozesse*
  - Systeme sind oft Insellösungen zu einem geänderten Geschäftsprozess
  - Marketingabteilung lässt Datenbank für Pflege von Kundendaten hinsichtlich Werbezwecke entwickeln (ohne Verbindung zu bestehenden Kundeninformationssystemen)
- **Resultat Menge von Stove-pipe-systems**
  - nicht vernetzte isolierte Systeme unterschiedlichster Technologie



# Was ist EAI?

---

- Fiktives Beispiel Bank
  - Bank verwaltet Konten von Privatkunden
  - Prozesse: Konto anlegen, Geld abheben, Zinsen verbuchen, ...
  - Softwaresystem dazu, vor 25 Jahren eingeführt
  - Technik: COBOL, 3270 Terminal-Emulation und zentrale Server.
  - Software wird im Intranet z.B. an Kassenschalter verwendet

# Was ist EAI?

---

- *Geschäftsprozesse* ändern sich
  - 15 Jahre später: Geld Automaten ersetzen zunehmend Kassenpersonal
  - 20 Jahre später: Internetbanking
- *Technik* hat sich in 20 Jahren stark gewandelt
  - Geldautomat und Internetanwendung muss Zugriff auf Konten haben
  - Zugriff auf alte Bankwendung problematisch
- Lösungsalternativen
  1. Alte Anwendung durch neue ersetzen
  2. Wege suchen, alte Anwendung weiter zu verwenden

# Was ist EAI?

---

## 1. Alte Anwendung ersetzen

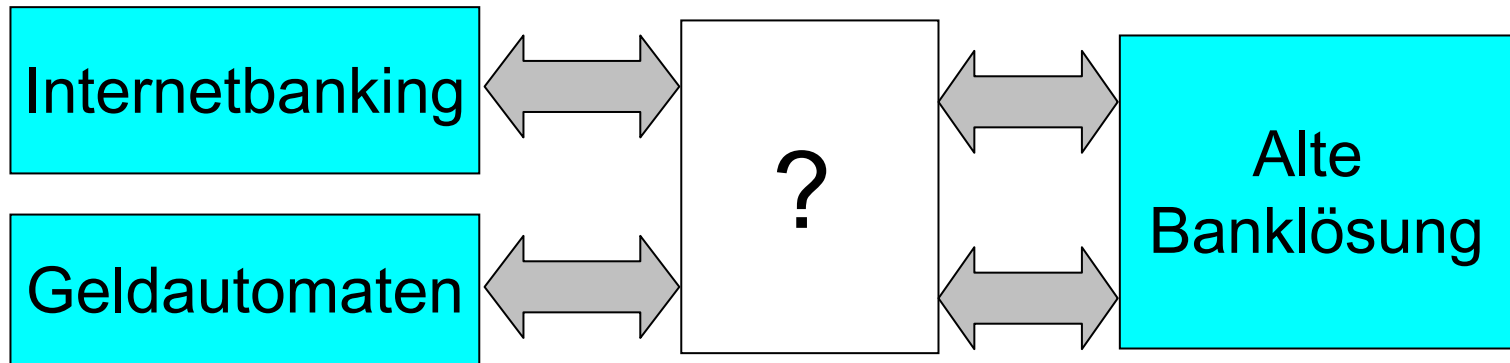
- Teuer (50 Millionen €)
- Risikoreich (Projekt könnte scheitern)
- Entwicklung und Einführung dauert 2-3 Jahre (Internet Banking soll aber in 1.5 Jahren eingeführt sein)
- Branchenwissen der COBOL Entwickler geht verloren
- ...

## ■ Fazit

- Die Investition alter Anwendungen ist oft zu hoch, um sie zu ersetzen

# Was ist EAI?

2. Wege suchen, alte Anwendung weiter zu verwenden
- EAI Produkte und Methoden einsetzen, um Technologie Hürden zu überwinden
  - Alte Anwendung schrittweise modernisieren
  - Heterogene Systeme integrieren

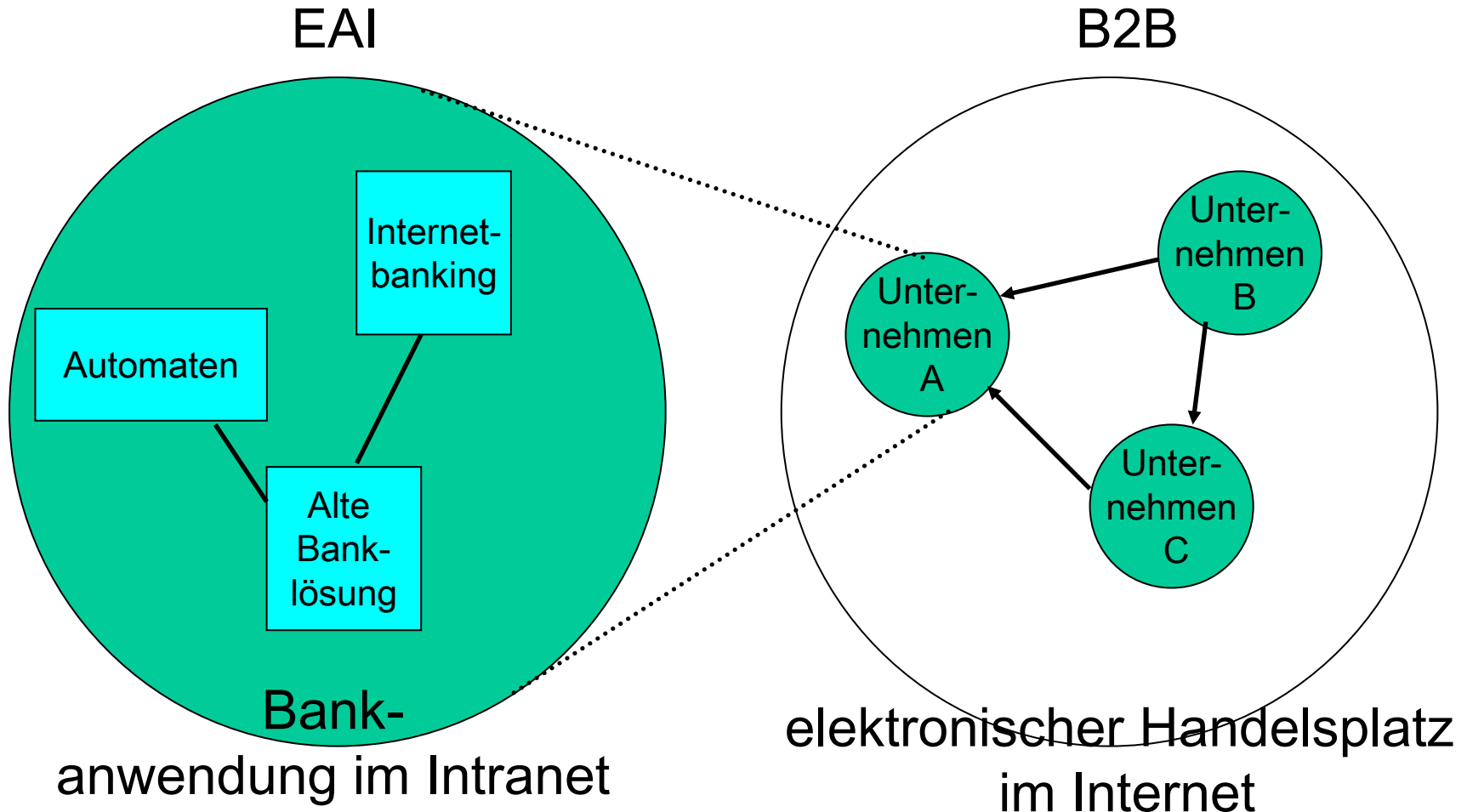


# Was ist EAI?

---

- Abgrenzung zu B2B (B2C)
  - Business to Business (Customer)
  - Integration von System und Geschäftsprozessen *verschiedener* Unternehmen
  - Meist über das Internet oder WWW
  - Elektronische Handelsplätze (für Geschäftskunden)

# Was ist EAI?



# Was ist EAI?

---

- B2B und EAI
  - Unterschiede geringfügig
  - Werkzeuge für B2B ähnlich zu EAI
  - Sicherheitsanforderungen bei B2B zu EAI restriktiver
- Ansonsten: Probleme und Herausforderungen ähnlich
  - Technologien überbrücken
  - Prozesse neu gestalten

# Übersicht

---

- Überblick
  - Literatur
  - Was ist EAI?
  - Warum EAI?
- Anwendungsfelder

# Warum EAI?

---

- Trends, die zur Integration von IT-Systemen führen:
  1. Time-to-market
  2. Zurechtschneidern halbfertiger Anwendungen, statt Eigenentwicklung
  3. Unternehmensfusionen und –aufspaltungen

# Warum EAI?

---

- Time-to-market
  - Produktzyklen werden immer geringer (Mobiltelefon- oder DSL-Abos)
  - Neue Produkte müssen immer schneller eingeführt werden (Voice over IP, TV über Internet)
  - Hat oft neue Prozesse zur Folge (TV bei Telekommunikationsunternehmen bestellen)
- Neue Prozesse → IT-Anwendungen müssen angepasst werden

# Warum EAI?

---

- Zurechtschneidern halbfertiger Anwendungen
  - Statt teure Eigenentwicklung, Einführung von „Standard“ Software wie SAP R/3.
  - Anwendung im Quelltext ausgeliefert
    - ➔ *Anbindung* z.B. für eigenes Logistiksystem, CRM System, Webshop nötig
  - MOTS (Modifiable off the shelf): Anwendung halbfertig, muss – teilweise im Quelltext – angepasst werden
  - COTS (Custom off the shelf): Anwendung fix und fertig, leicht installierbar (z.B. Textverarbeitung)

# Warum EAI?

---

- Unternehmensfusionen und Aufspaltungen
  - Aufspaltung: Abspaltung von T-Online von T-Com
  - Merger & Akquisition (MA)
    - Verschmelzen von T-Com und T-Online (2005)
    - Übernahme Debis von T-Systems (2000)
  - Verschmelzung
    - ➔ Unterschiedliche Systeme verschmolzener Unternehmen integrieren, doppelte abschaffen (z.B. nur ein Lohnbuchhaltungssystem)
  - Aufspaltung
    - ➔ „verlorene“ Systeme ersetzen und integrieren (z.B. neues Lohnbuchhaltungssystem)

# Überblick

---

- Zusammenfassung
  - EAI beschäftigt sich mit der Integration heterogener Anwendungen und der zugrunde liegenden Prozesse in einem Unternehmen
  - EAI wird durch technologischen Fortschritt, schnelleren Produktzyklen und sich stetig ändernde Unternehmen bedeutender

# Übersicht

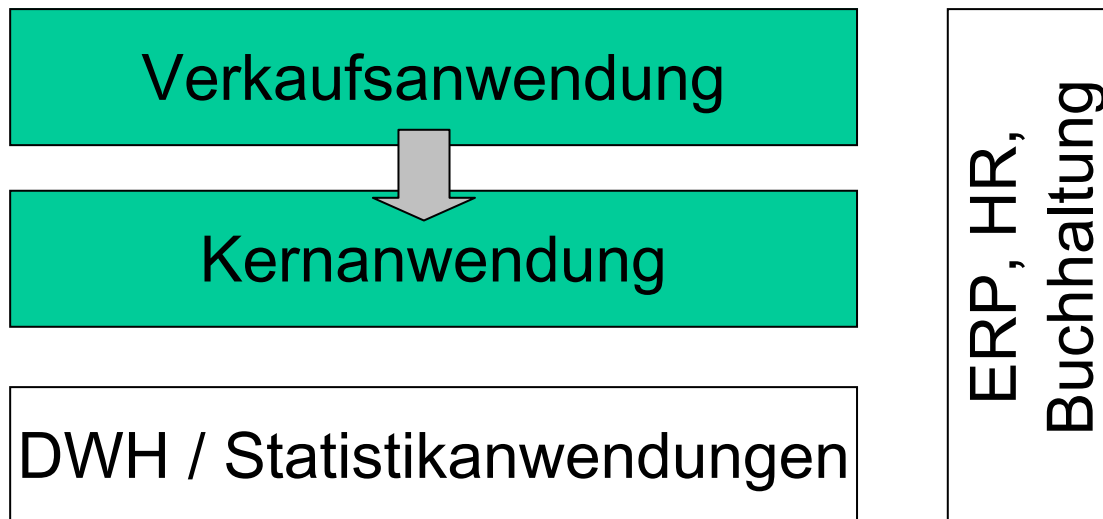
---

- Überblick
  - Literatur
  - Was ist EAI?
  - Warum EAI?
- Anwendungsfelder
  - Multi-Channel-Architekturen
  - Applikation-zu-Applikation Kommunikation
  - Geschäftsprozessintegration

# Multi-Channel-Architekturen

---

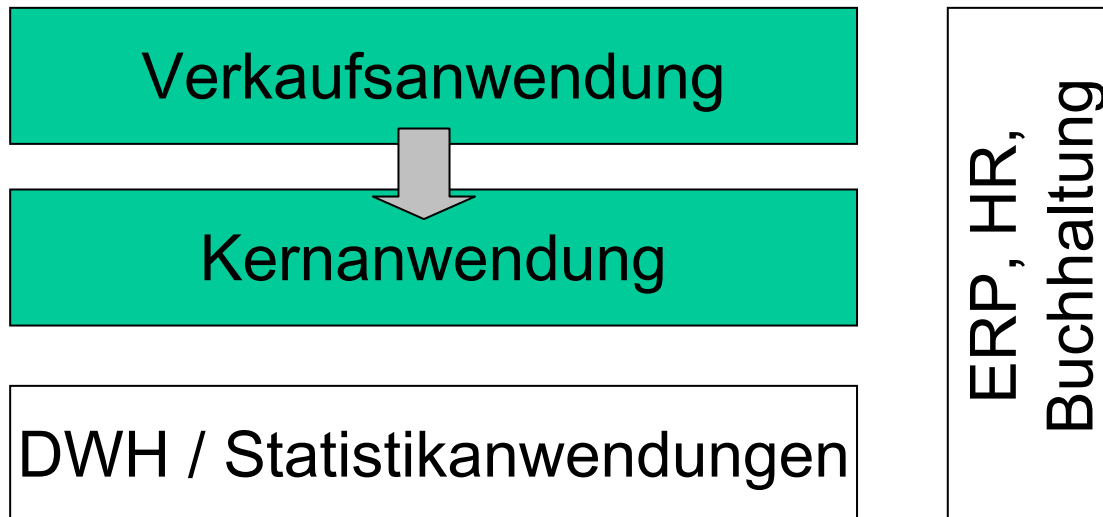
- Mehrkanal-Architektur
- Kanal: Vertriebskanal
  - Früher: Meist einige wenige Vertriebskanäle



- Heute: Viele verschiedene Vertriebskanäle

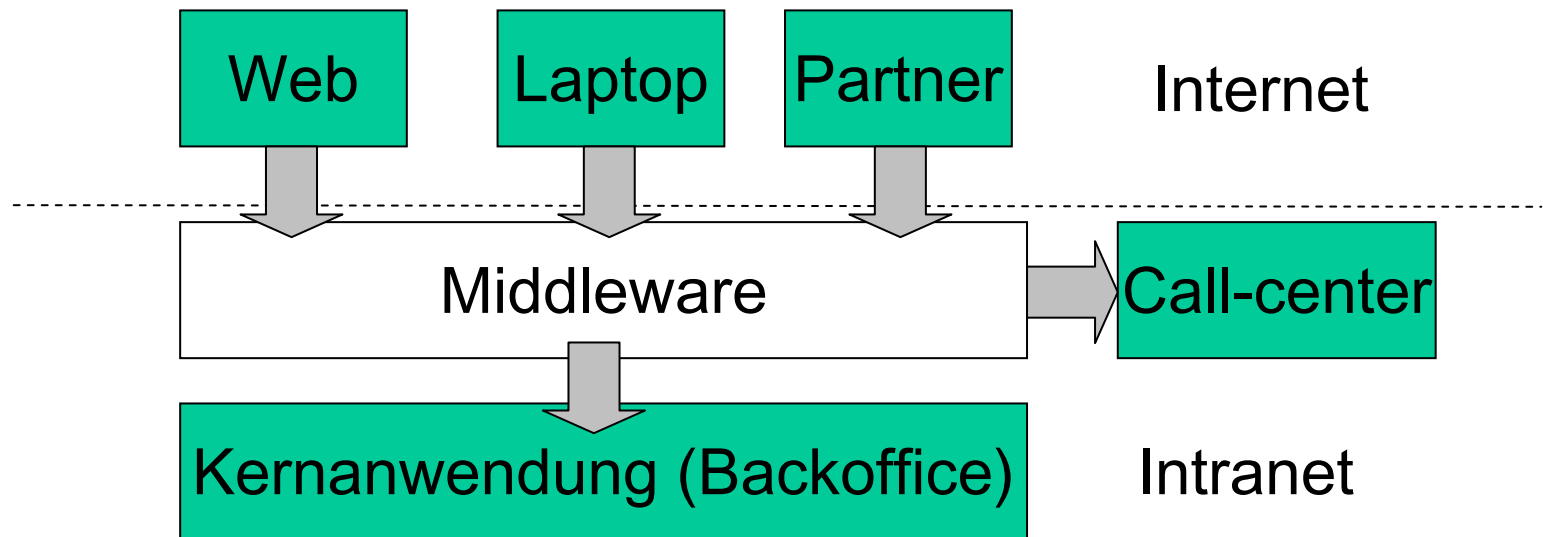
# Multi-Channel-Architekturen

- Beispiel Versicherung
  - Kanäle: Versicherungsvertreter, Partnerorganisationen, Agenten
  - Informationsfluss vom Kunden zur Versicherung hauptsächlich Paper
  - Eingabe Daten in einer Verkaufsanwendung (Frontoffice) oder direkt im Kernsystem (Backoffice)



# Multi-Channel-Architekturen

- Neue Verkaufskanäle kommen hinzu
  - Webanwendung (Prämienberechnung Versicherung)
  - Off-line Anwendung für Laptop
  - Wiederverkäufer (Partner)
  - Call-center



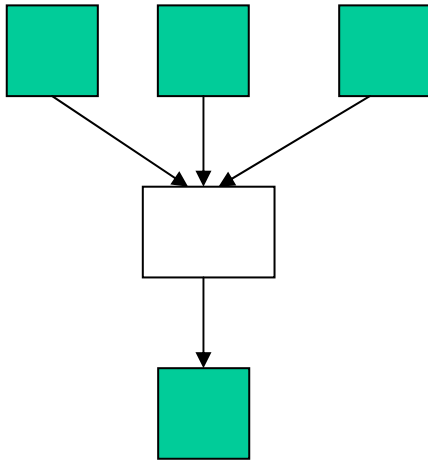
# Multi-Channel-Architekturen

---

- Middleware
  - Keine klare, fassbare Definition
  - Minimal: *Kommunikationsinfrastruktur zwischen mehreren Systemen* (oder Schichten)
- Beispiele
  - Common Object Request Broker (CORBA)
  - Message Queue (MQ)
  - Transaktionsmonitor
  - EAI Produkte (z.B. MS BizTalk)

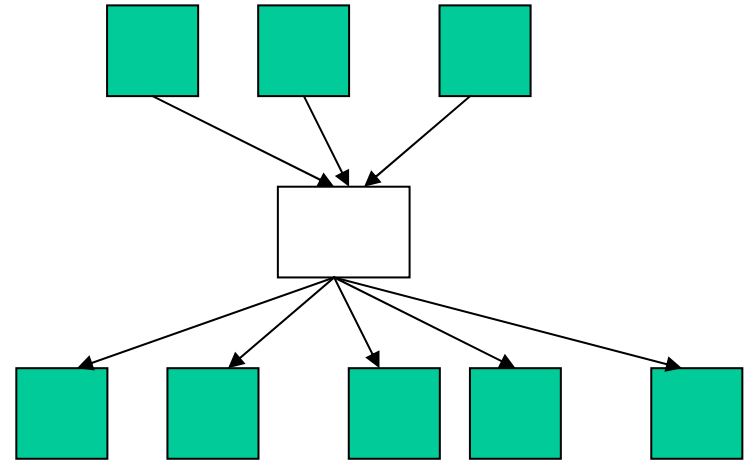
# Multi-Channel-Architektur

n:1 Kanäle



- Ein Backoffice für n Kanäle

n:m Kanäle



- m Hintergrundsysteme für n Kanäle

# Multi-Channel-Architektur

---

- Extrem abstrakte Architektursicht
- EAI in diesem Kontext
  - Wie soll Zielarchitektur sein?
  - (nicht-funktionale) Anforderungen
  - Übersicht der bestehenden Systeme
  - Einzusetzende Produkte, Middleware (MQ oder Web Services oder ???)
  - Zu entwickelnde Komponenten für diese Produkte/Middleware?

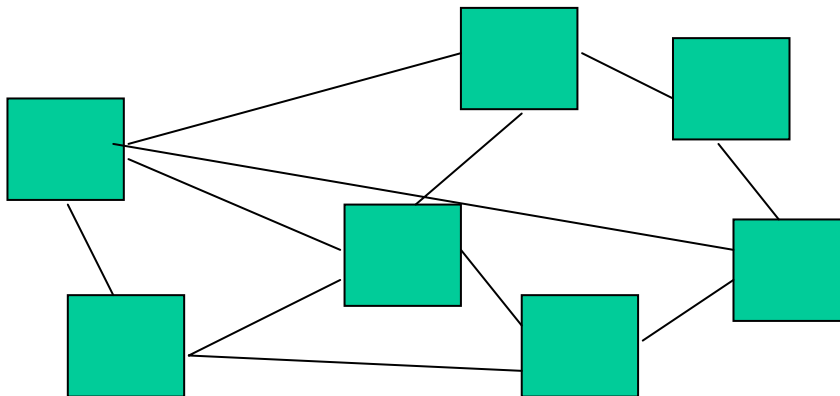
# Übersicht

---

- Überblick
  - Literatur
  - Was ist EAI?
  - Warum EAI?
- Anwendungsfelder
  - Multi-Channel-Architekturen
  - Applikation-zu-Applikation Kommunikation
  - Geschäftsprozessintegration

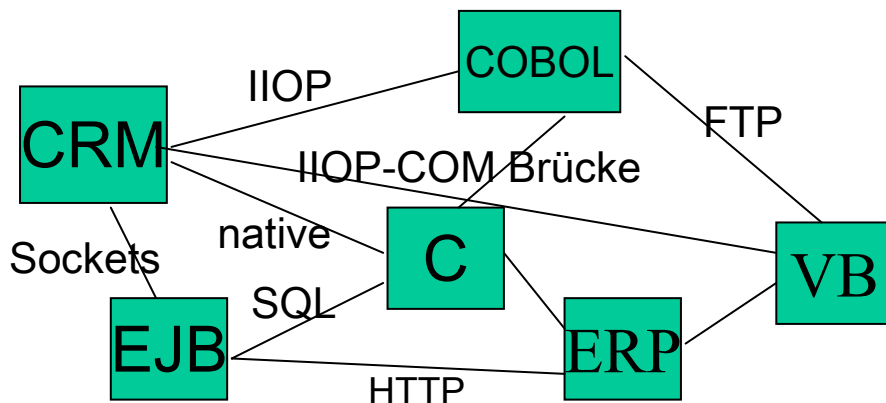
# A2A Integration

- Application to application (A2A) integration
  - Häufigstes Verkaufsargument für EAI
  - Dutzende von Client/Server Inselanwendungen unterschiedlichster Technologie
  - Anwendungen dieses Szenarios kommunizieren direkt miteinander
  - Schlimmster Fall:  $O(n^2)$  Verbindungen



# A2A Integration

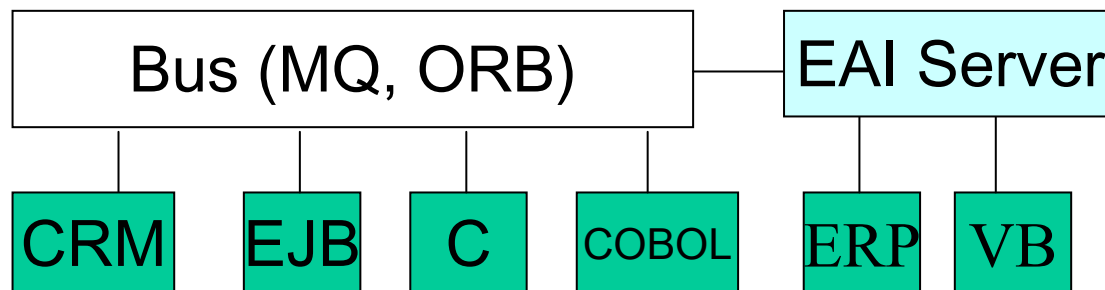
- Probleme bei A2A
  - Zu viele direkte Verbindungen (Komplexität hoch)
  - Viele verschiedene Technologien, Kommunikationsprotokolle (System können nicht vernünftig miteinander gekoppelt werden)
  - Nicht normalisierte Schnittstellen (RPC, OO, Datenorientiert, Granularität verschieden)
  - Jedes System von anderen Personen (jahrelang) gewartet (Wissen ist über gesamte Organisation verteilt)



Babylonische Sprachverwirrung:  
Kein System versteht das andere

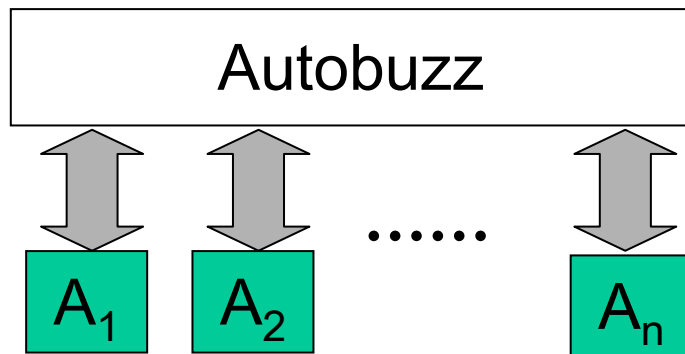
# A2A Integration

- Typische Lösungsarchitekturen
  - Softwarebus (CORBA, MQ)
  - Dienst-orientierung (CORBA, WS)
  - Hub and Spoke (EAI Produkten)
- Typische Versprechungen
  - O(n) Abhängigkeiten
  - Schnellere Entwicklung, dank existierender Adapter
  - Zukunftssicheres Produkt
  - Geringere Betriebskosten

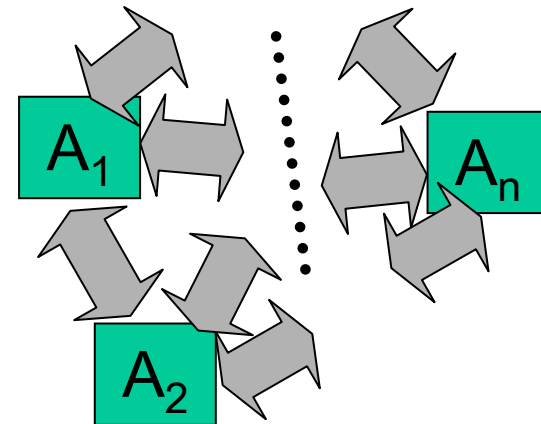


# A2A Integration / Versprechungen

- Mit HALs Autobuzz Technologie nur noch  $O(n)$  Abhängigkeiten



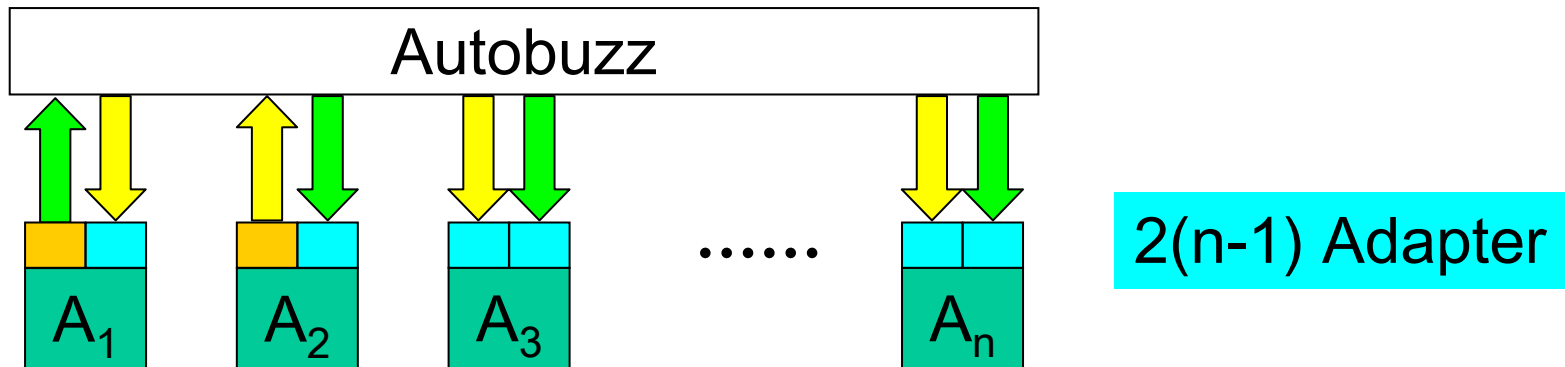
$O(n)$



$O(n^2)$

# A2A Integration / Versprechungen

- $A_1$  sendet **Produktdaten** an alle anderen
- $A_2$  sendet **Auftragsdaten** an alle anderen
- Abhängigkeiten:  $2(n-1)$  statt 2
  - Sendendes System muss Daten (einmal) aufbereiten (einheitliches Format)
  - Jedes empfangendes System muss Daten *spezifisch verarbeiten* (Adapter)



$O(n^2)$  Abhängigkeiten *eine Architekturstufe tiefer*

# A2A Integration / Versprechungen

---

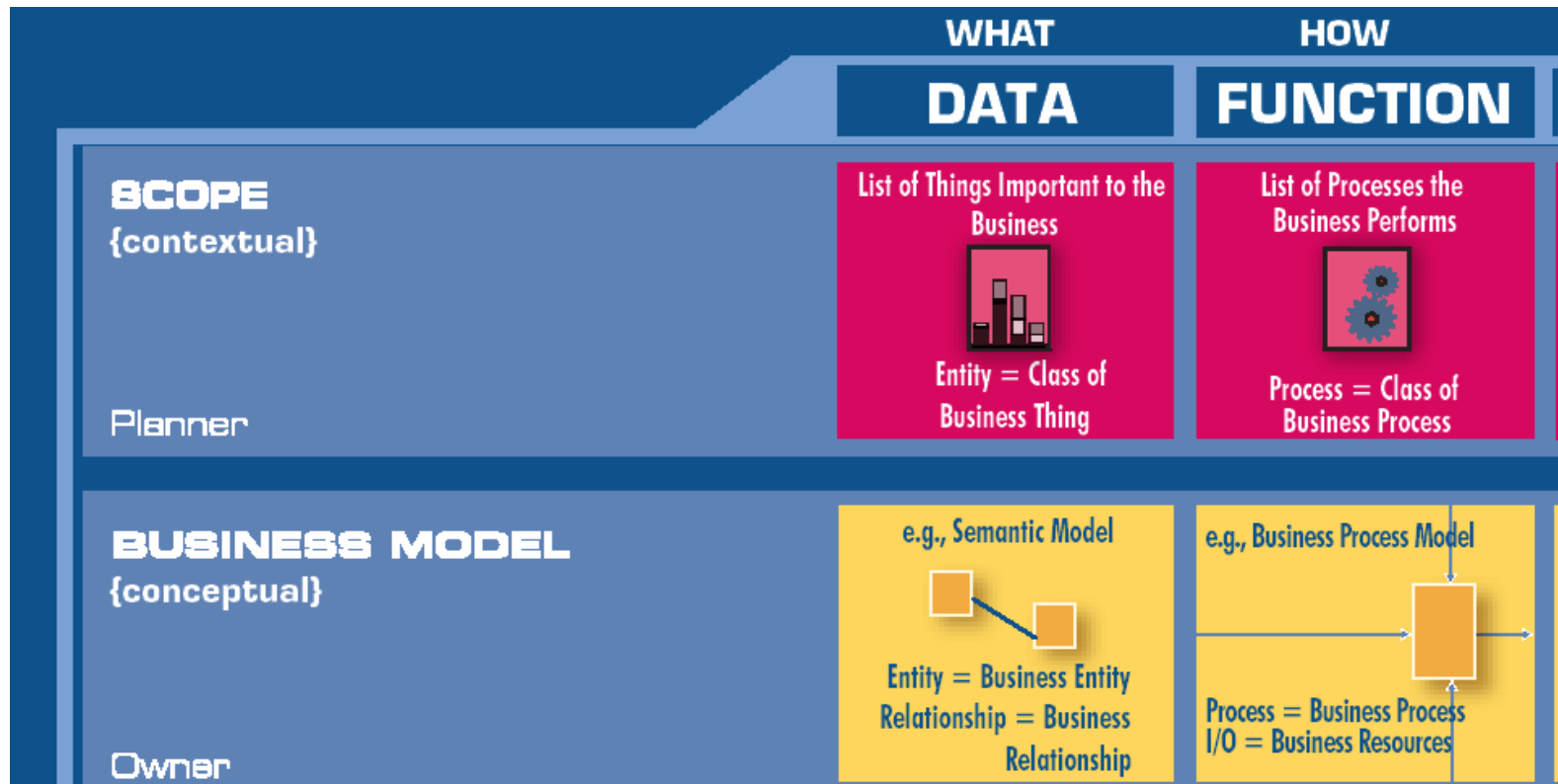
- Versprechungen kritisch gegenübertreten
  - Welcher Architektursicht „verdangt“ Verkäufer sein Argument?
    - Konzeptionelle Sicht auf Infrastruktur (Netzwerkverbindungen, Hardware)?
    - Funktionale Sicht auf Komponenten (Software)?
  - Abhängigkeiten genauer untersuchen:
    - Welche Aufgaben fallen bei A2A Integration an?
    - Welche neuen Komponenten sind notwendig?
- Bei A2A Integration
  - Im schlimmsten Fall (alle Anwendungen kommunizieren miteinander) immer  $O(n^2)$  funktionale Abhängigkeiten ( $n$  = Anzahl Systeme)
  - Schlimmer noch:  $n$  = Anzahl Schnittstellen (Ein System kann mehrere Schnittstellen haben)

# Architektursichten

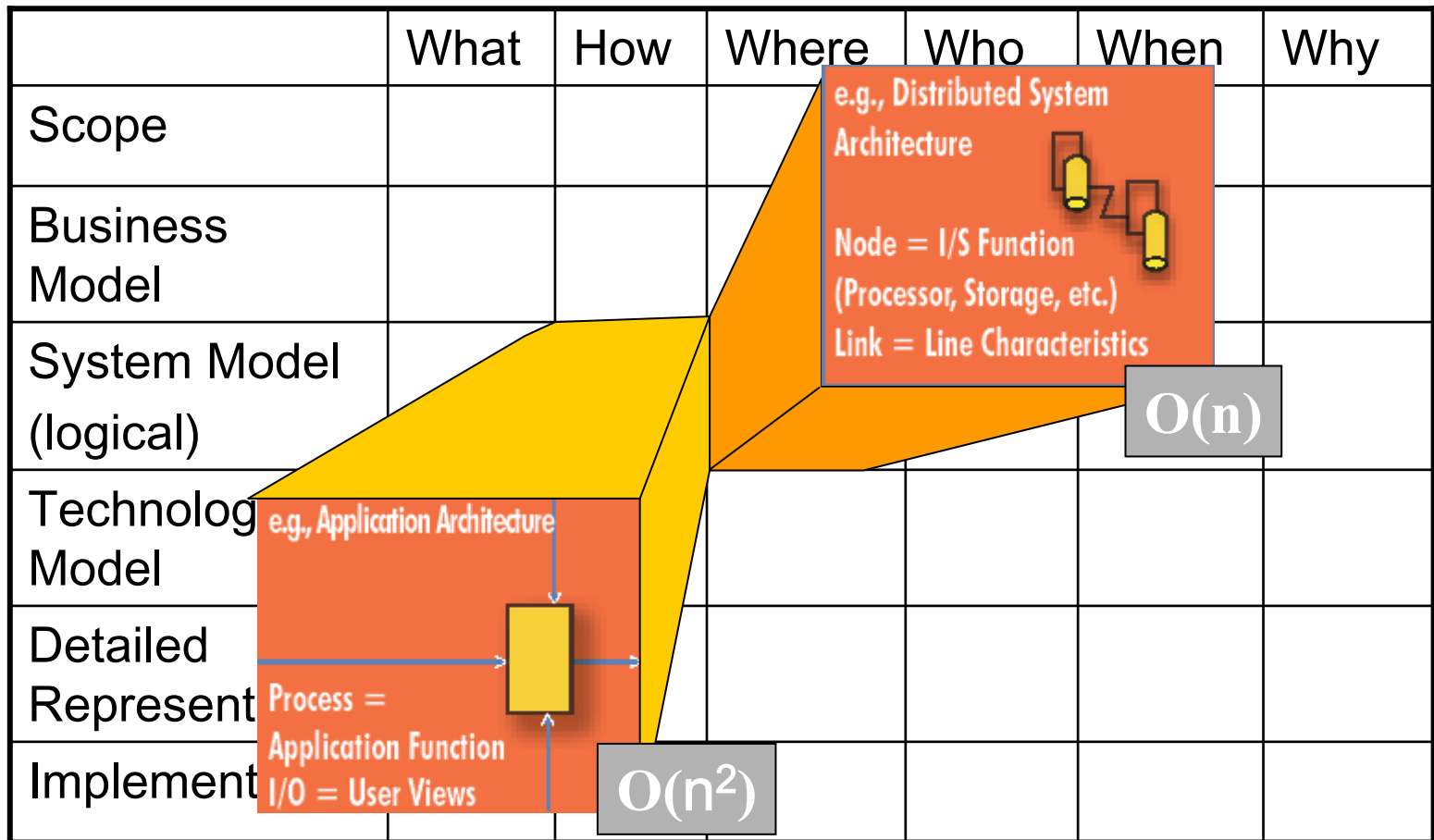
- Zachman Framework for Architecture
  - [www.zifa.com](http://www.zifa.com)

abstrakt		What	How	Where	Who	When	Why
Architekturebene ↓	Scope						
	Business Model						
	System Model						
	Technology Model						
	Detailed Representation						
	detailliert	Implementation					

# Zachmann Framework



# Architektursichten



# Architektursichten

---

- Verteilte Sicht (Where)
  - (vermutlich) Verbesserung auf  $O(n)$
- Funktionale Sicht (How)
  - Weiterhin  $O(n^2)$ , keine  $O(n)$
- Funktionale Sicht kann beträchtliche Kosten verursachen
  - Anforderungsanalyse pro Schnittstelle
  - Implementierung
  - Testen verteilter Komponenten

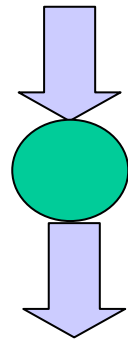
# Übersicht

---

- Überblick
  - Literatur
  - Was ist EAI?
  - Warum EAI?
- Anwendungsfelder
  - Multi-Channel-Architekturen
  - Applikation-zu-Applikation Kommunikation
  - Geschäftsprozessintegration

# Geschäftsprozessintegration

- Prozess
  - Eingabe (Antragsformular)
  - Verarbeitung (Entscheidung über Antrag)
  - Ausgabe (pos. Entscheid, Ablehnung)
- Prozesse werden angestoßen
  - Kundenbestellung (eShop, Anruf)
- Prozesse können verzweigen
  - Bestellung ausliefern und Inkasso betreiben
- Prozesse enden
  - Bestellung ausgeliefert
  - Bestellung bezahlt
- Prozesse können verschmelzen
  - Bestellung ausgeliefert und bezahlt, löst Teilnahme an einem Wettbewerb aus



# Geschäftsprozessintegration

---

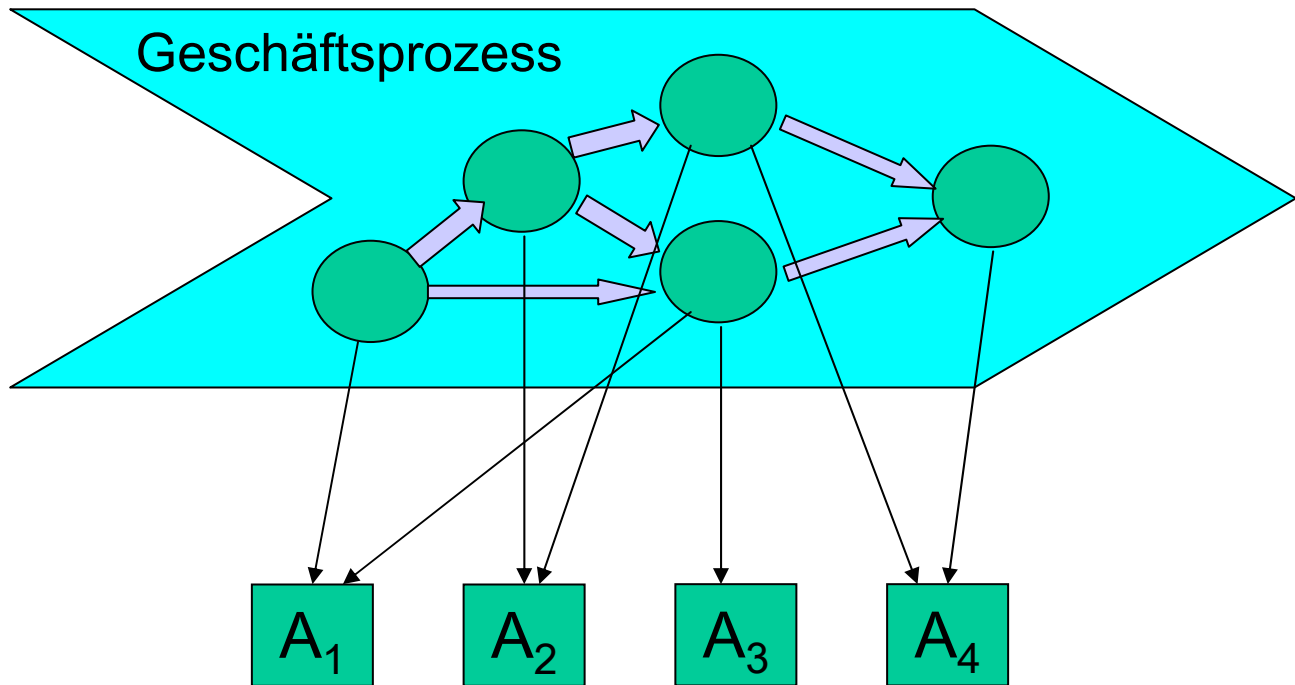
- Unternehmen „bestehen“ aus Vielzahl von Geschäftsprozessen
- Geschäftsprozesse regeln
  - abteilungsübergreifende Aufgabenteilung
  - Verantwortlichkeiten von Personen/Rollen
  - Entscheidungskompetenzen
- Geschäftsprozesse ohne IT (bis ca. 1950)
  - Eingabe: Papier (von einer anderen Abteilung)
  - Verarbeitung: Mensch
  - Ausgabe: Papier (geht an eine weitere Abteilung)
- Heute
  - Prozesse sind (teil)automatisiert, Eingabe, Verarbeitung, Ausgabe elektronisch

# Geschäftsprozessintegration

---

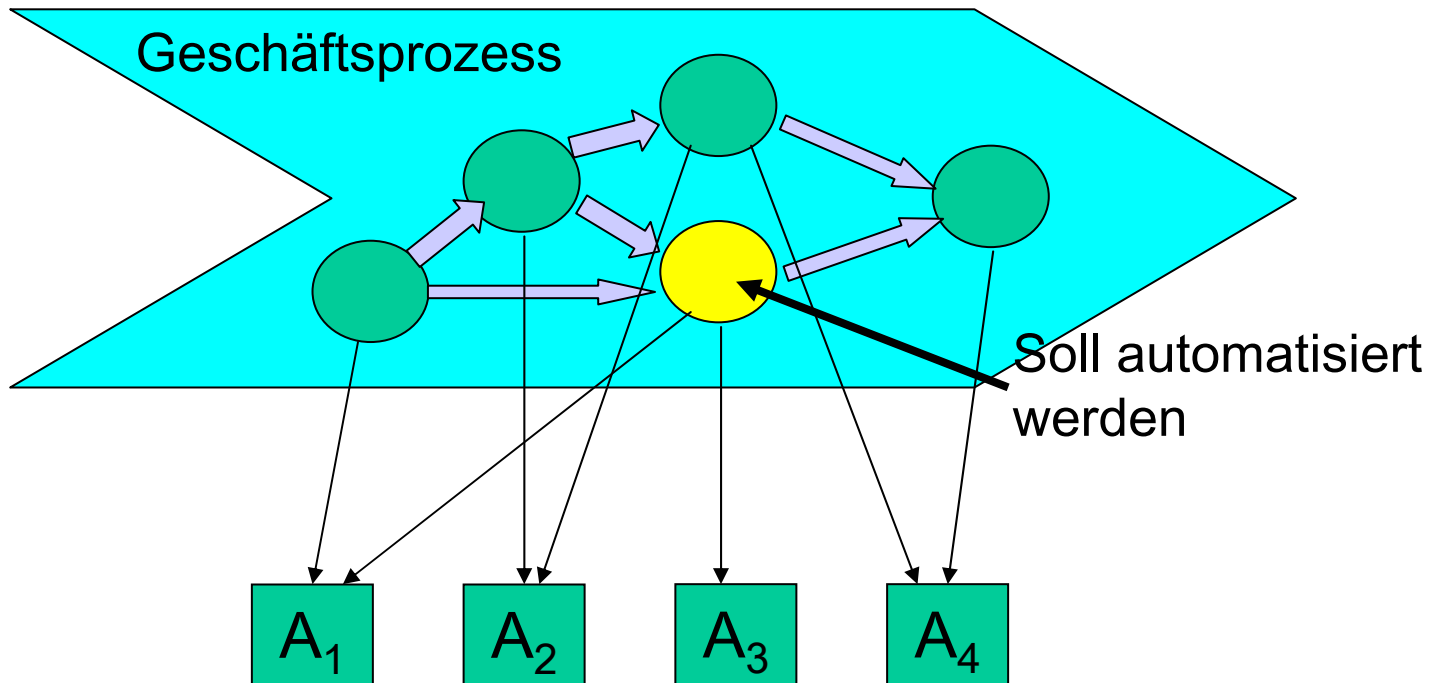
- 1950 – heute
- Ein noch manueller Prozess wird automatisiert oder bereits automatisierte Prozesse ändern sich. Folge:
  - Bestehendes Softwaresystem erweitern oder
  - Neues Softwaresystem dazu
  - Jeweils neuer Teil in bestehende IT-Landschaft integrieren

# Geschäftsprozessintegration



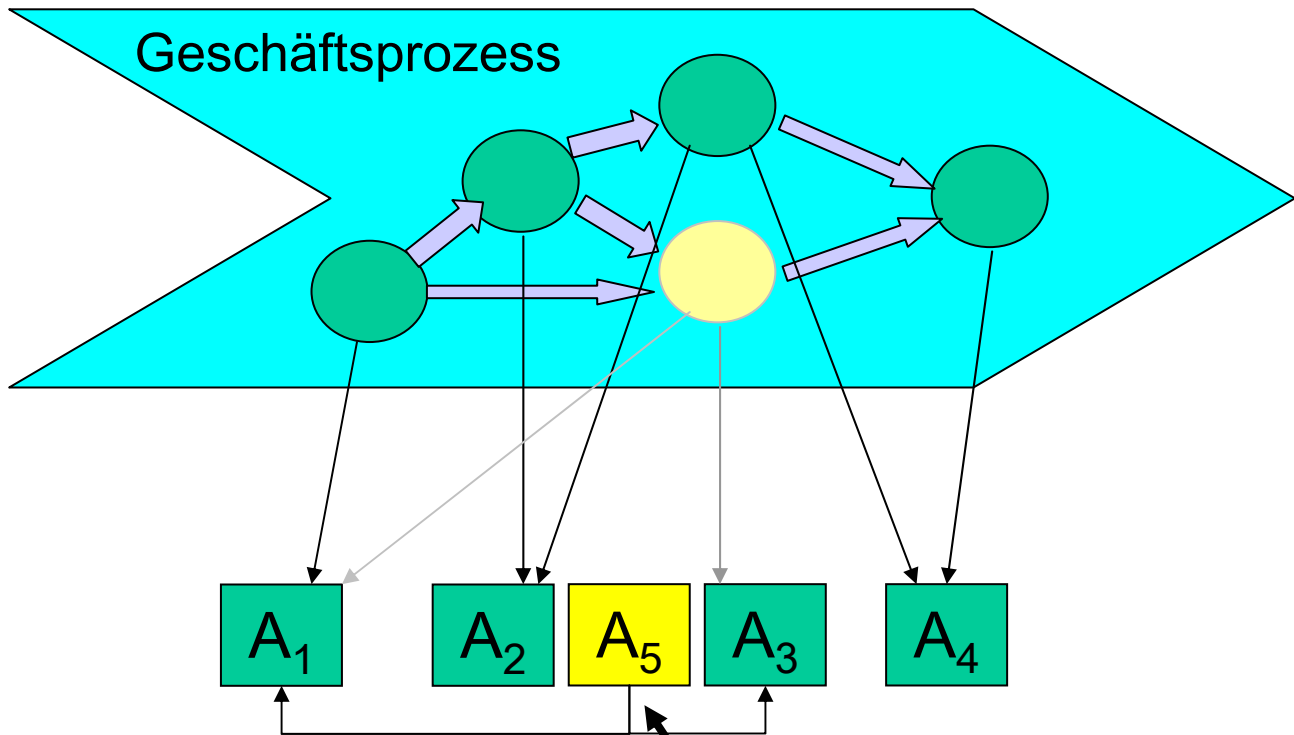
Automatisierten Prozesse mit bestehenden Systemen

# Geschäftsprozessintegration



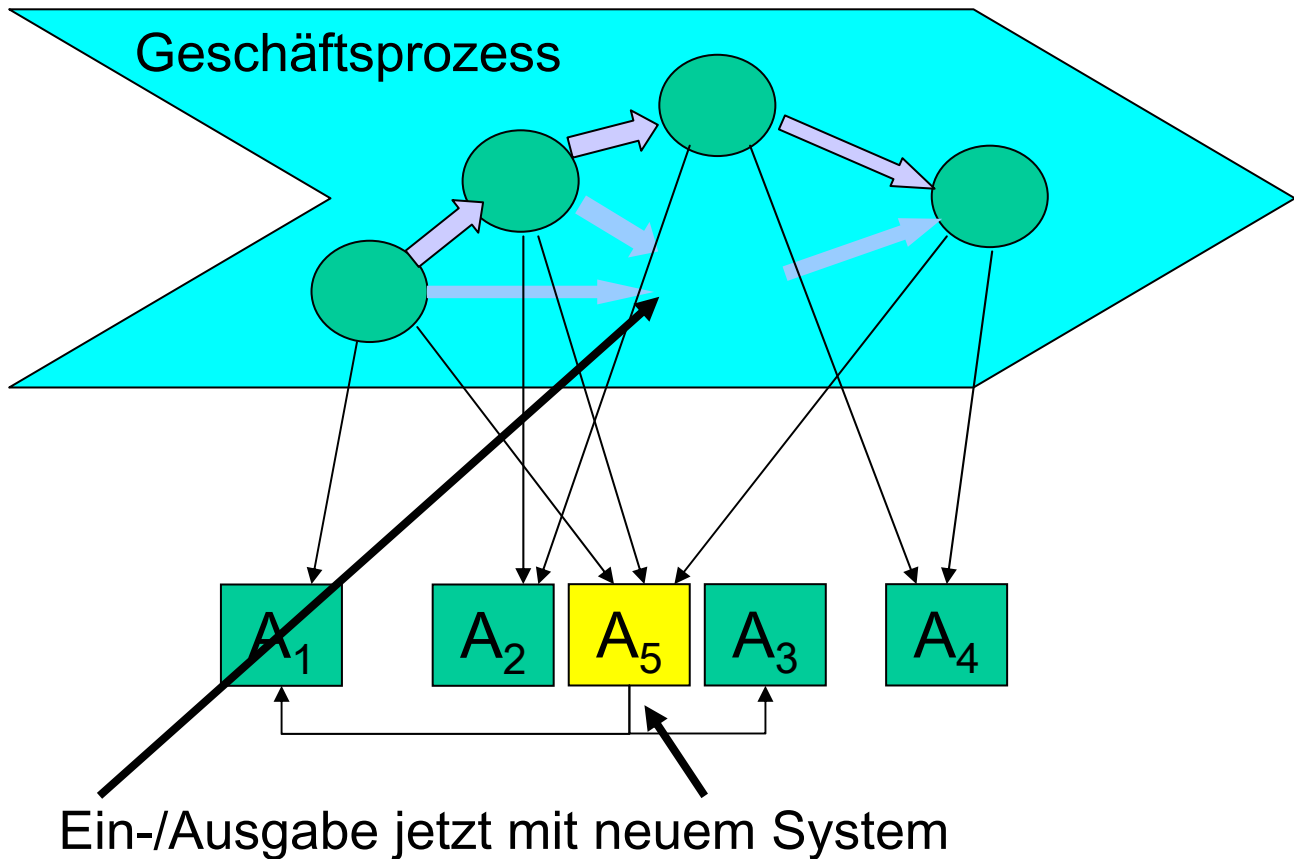
Automatisierten Prozesse mit bestehenden Systemen

# Geschäftsprozessintegration

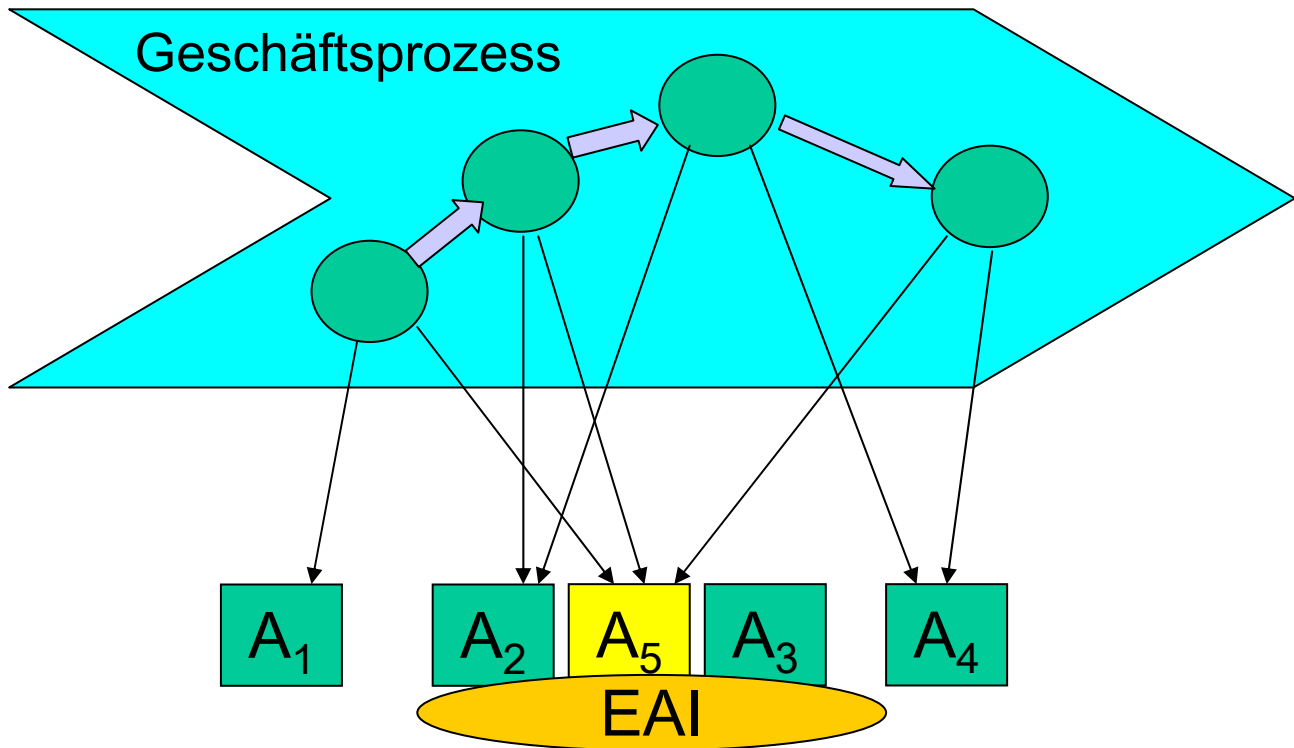


Neues System dazu  
Manuelle Kommunikation mit bestehenden Systemen  
wird automatisiert

# Geschäftsprozessintegration



# Geschäftsprozessintegration



Geschäftsprozess vereinfacht durch Automatisierung

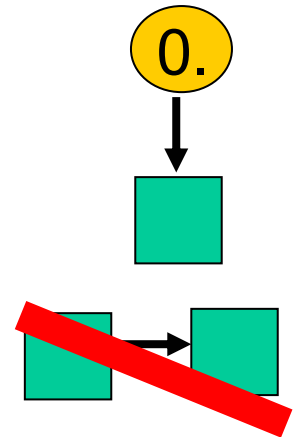
# Geschäftsprozessintegration

---

- Geschäftsprozesse ändern immer schneller
  - Kurze Produktzyklen
  - Fusionen
  - Outsourcing
  - Reorganisationen
- Integrationsaufgaben im Unternehmen nehmen zu

# Kontextdiagramm

- Beschreibung Schnittstellen System zu Umwelt
- Syntaktische Regeln
  - Nur ein **Prozess** (Gesamtsystem), erhält die Nummer 0.
  - Mindestens eine **Schnittstelle** (Teilsystem)
  - Zwischen Schnittstellen keine Datenflüsse (Pfeile)
  - Enthält keinen Speicher
  - Jede Schnittstelle nur einmal vorhanden  
Ausnahme: Übersichtlichkeit



# Kontextdiagramm

---

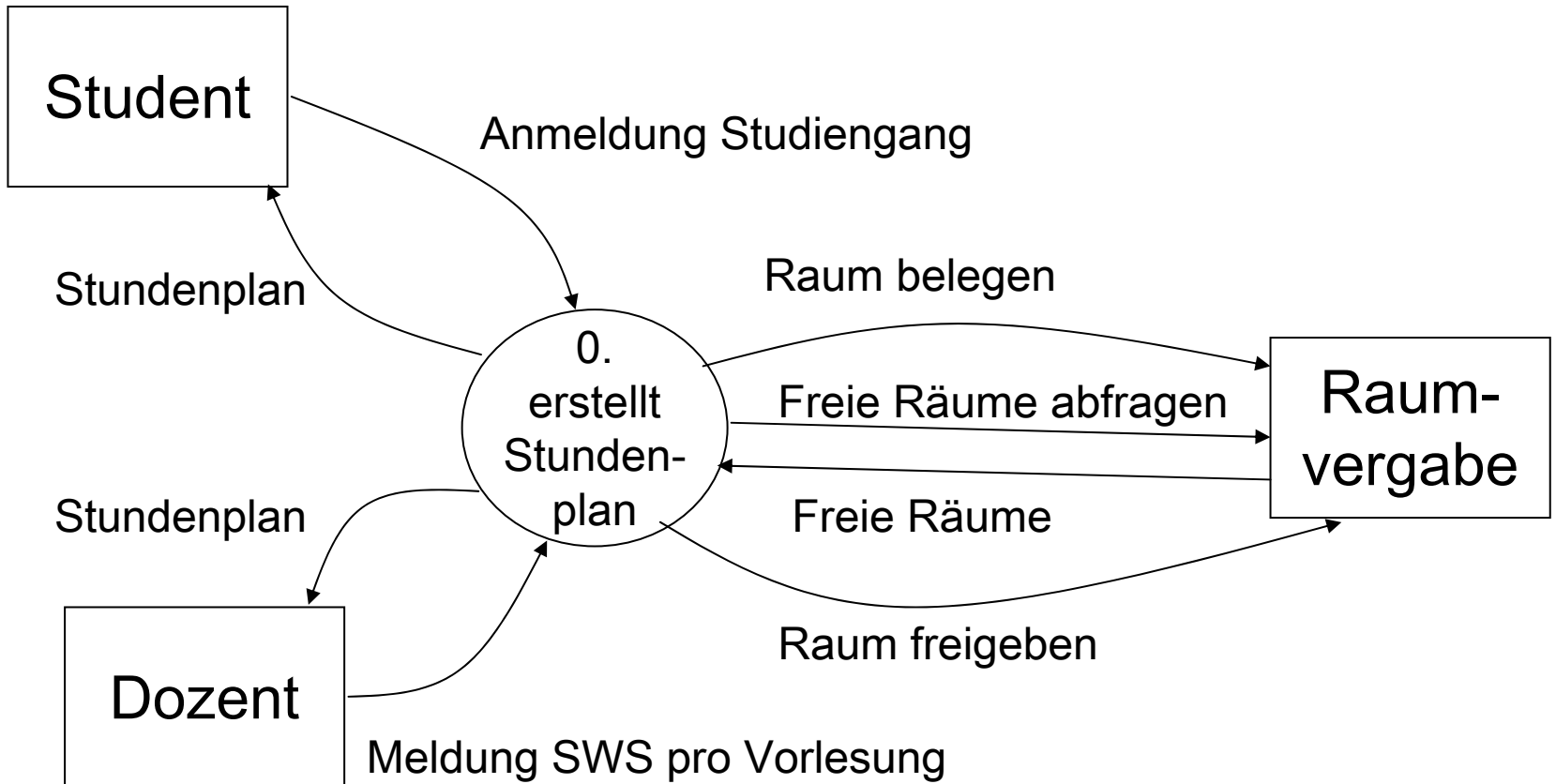
- Semantische Regeln:
  - Beschreibt Anwendungsbereich des Systems
  - Zeigt Datenflüsse über die Systemgrenzen
  - Ist die Zusammenfassung von Diagramm 0
  - Kann es von derselben Schnittstelle mehrere Instanzen geben, wird sie einmal dargestellt
  - Schnittstelle muss ursprüngliche Quelle oder Senke einer Information angeben
  - Wahl einer Schnittstelle abstrahiert von der konkreten Eingabe oder Ausgabe

# Kontextdiagramm / Beispiel

---

- Prozess 0
  - Dekan erstellt Stundenplan für Studenten eines Studiengangs und Semesters sowie für Dozenten
- Schnittstellen
  - Student
    - Schreibt sich für einen Studiengang ein
  - Dozent
    - Meldet SWS pro Vorlesung dem Dekan
  - Raumvergabe (System)
    - Abfrage von Räumen
    - Raum belegen und wieder freigeben

# Kontextdiagramm / Beispiel



# Kontextdiagramm

---

- Angemessene Abstraktion der Datenflüsse
  - Zu detailliert: unübersichtlich, überladen
  - Zu abstrakt: nichts sagend
- Leitfaden
  - Außenstehenden müssen wesentlichen Informationen über die Umwelt erkennen
  - Problembezogene Namensgebung
  - Alle Datenflüsse auf selben Abstraktionsniveau

# Kontextdiagramm

---

- In der Geschäftsprozessintegration
  - Kontextdiagramm für zu automatisierenden Prozess aufstellen
  - Datenflüsse mit umgebenden Schnittstellen analysieren
- Startpunkt für Entwurf Schnittstellen
  - Anwender zu neuem System
  - Bestehende Systeme zu neuem System