

Vorlesung "Software-Engineering"

Prof. Ralf Möller, TUHH, Arbeitsbereich STS

■ Vorige Vorlesung

- Techniken für die Entwicklung von Standardsoftware
- Anpassung durch Adaption von Referenzmodellen (Bsp. SAP)
- Modellierung von Geschäftsprozessen (Bsp.: ARIS)

■ Heute

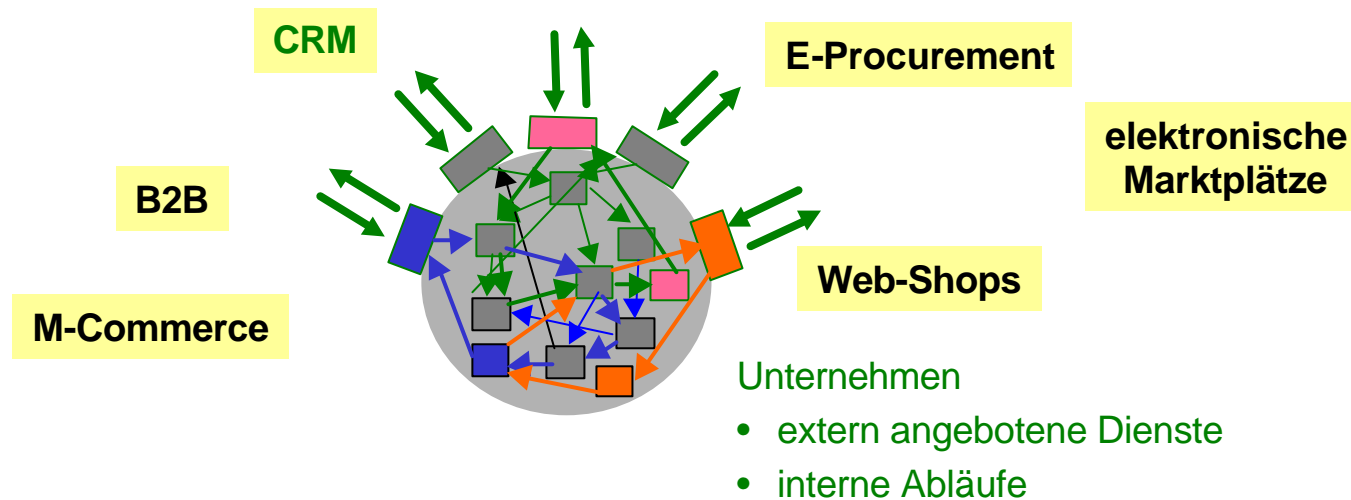
- Enterprise Application Integration

Danksagung

- Diese Vorlesung enthält Material aus zwei Präsentationen:
- Enterprise Application Integration (EAI), Michael Kaib, Booz·Allen & Hamilton, Philipps Universität Marburg
- Enterprise Application Integration: Von der daten- zur prozeßorientierten Kopplung von Anwendungssystemen, Manfred Reichert, Universität Ulm, Abt. Datenbanken und Informationssysteme

Motivation

Ständig neue „Trends“ – ständig neue Herausforderungen



⇒ Folge:

- ständig neue Produkt- und Service-Angebote
- die in die existierenden betrieblichen Abläufe integriert werden müssen

⇒ Fragen:

- Wie rasch geht das und zu welchen Kosten?
- Mit welchem Fehlerrisiko?
- Wie entgeht man der „Wartungsfall“



Motivation

Umdenken erforderlich!

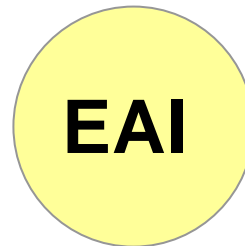
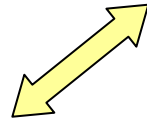
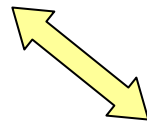
- Abkehr von
 - lediglich implizit realisierten Abläufen / Prozessen
 - „hart verdrahteter“ Implementierung der Prozesse
- Hin zu
 - explizit modellierten Prozessen
 - Trennung von Ablauflogik und Applikationscode
 - expliziter Prozess-Steuerung und -Überwachung
 - nahtloser Integration existierender Applikationen
 - und ...
- *Process Automation* und *Enterprise Application Integration* ➔ wichtige technologische Basis für Umsetzung einer **Enterprise Architecture**



Enterprise Application Integration (EAI)

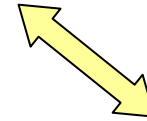
Integrationsart:

- Präsentations-, Daten- oder Anwendungsebene
- Anwendungsintegration:
 - direkte Kopplung
 - Broker
 - prozessorientiert



Middleware:

- DB-Access (z.B. JDBC)
- Funktionsorientiert (z.B. RPC)
- Transaktionsorientiert (z.B. TP-Monitor)
- Message-orientiert (MOM)
- Komponentenorientiert (z.B. EJB / Application Server)



Kommunikationsmodell:

- synchrone Kopplung (z.B. Request/Reply)
- asynchrone Kopplung (z.B. Publish / Subscribe)
- Messaging vs. definierte SS

Services:

- herstellerabhängige Integrationservices (z.B. für Verwendung von Kreditkarten)
- Fehler-/Ausnahmebehandlung

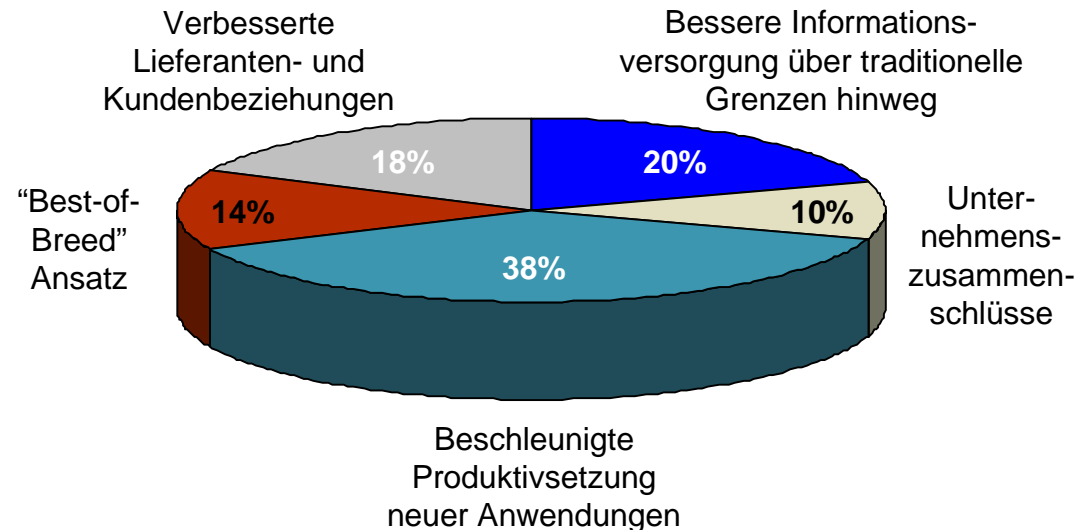


Der Bedarf der Integration betrieblicher Anwendungssysteme in Unternehmen ist heute so groß wie niemals zuvor

- ▶ Fortune 1000 Unternehmen geben 100 Mrd. US\$ pro Jahr für die Integration von IT-Systemen aus
- ▶ 30% aller IT-Ausgaben werden für die Integration von Systemen aufgewendet (Forrester Research)
- ▶ 35% aller IT-Budgets werden für die Integration ausgegeben (Gartner Research)



Gründe für die Anwendungsintegration



Die Ursachen für diesen Integrationsbedarf sind vielschichtig

Treiber des Integrationsbedarfes in Unternehmen

Unzureichende Planung der Unternehmens-IT-Architektur

- ▶ Systemauswahl auf funktionaler Ebene, getrieben durch Technologietrends ohne zureichende Berücksichtigung von Integrationsbedarfen
- ▶ Hoher Wartungsaufwand, hohe Projektdauer (“time-to-market”) für neue Systeme

Standardsoftwarepakete (z.B. ERP)

- ▶ Standardsoftwarepakete (z.B. ERP) haben sich in der Unternehmenspraxis durchgesetzt und müssen mit anderen IT-Systemen integriert werden
- ▶ Softwarepakete erlauben typischerweise nicht die Modifikation des Quellcodes und Schnittstellen sind noch unzureichend und schwer zu nutzen

Altanwendungen

- ▶ ERP und andere Softwarepakete haben Altanwendungen nicht zu 100% abgelöst - Performanz und Zuverlässigkeit von Mainframe-Systemen sowie hohe Ablösekosten sind der Grund

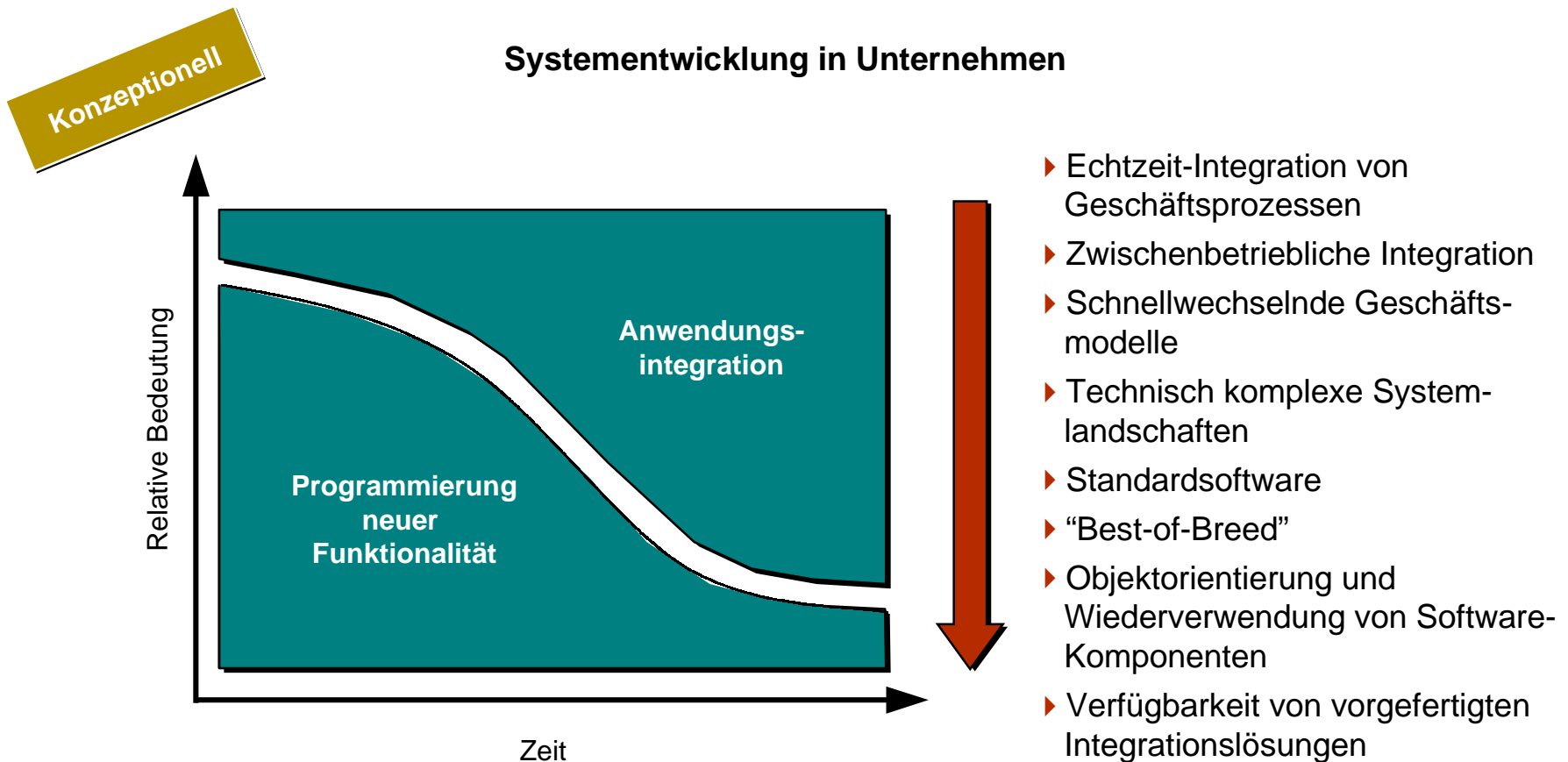
Unternehmenszusammenschlüsse

- ▶ Unternehmenszusammenschlüsse bedingen die Notwendigkeit schneller Integration möglicherweise sehr heterogener IT-Landschaften

E-Commerce

- ▶ Neue Kundenanforderungen machen die Integration der Kundenschnittstelle mit den Back-end-Systemen notwendig
- ▶ Business-to-Business E-Commerce fordert den Informationsaustausch mit Anwendungen außerhalb des Unternehmens, die von Handelspartnern besessen und betrieben werden

Die IT-Abteilungen vieler Unternehmen verfolgen bei der Systementwicklung heute einen integrationszentrierten Ansatz



Wesentlich ist die Unterscheidung zwischen der Ex-ante- und der Ex-post-Integration

Anwendungsintegration

Ex-ante Integration

“Ganzheitliche” Konzeption eines neu einzuführenden integrierten Informationssystems (“Grüne Wiese”)

- ▶ Beschreibungsmodelle integrierter Informationsverarbeitung (z.B. KIM, Mertens, Scheer)
- ▶ “Vorintegrierte” Standardsoftware mit mehr oder weniger komfortablen Schnittstellen nach außen (z.B. ERP Pakete wie SAP)

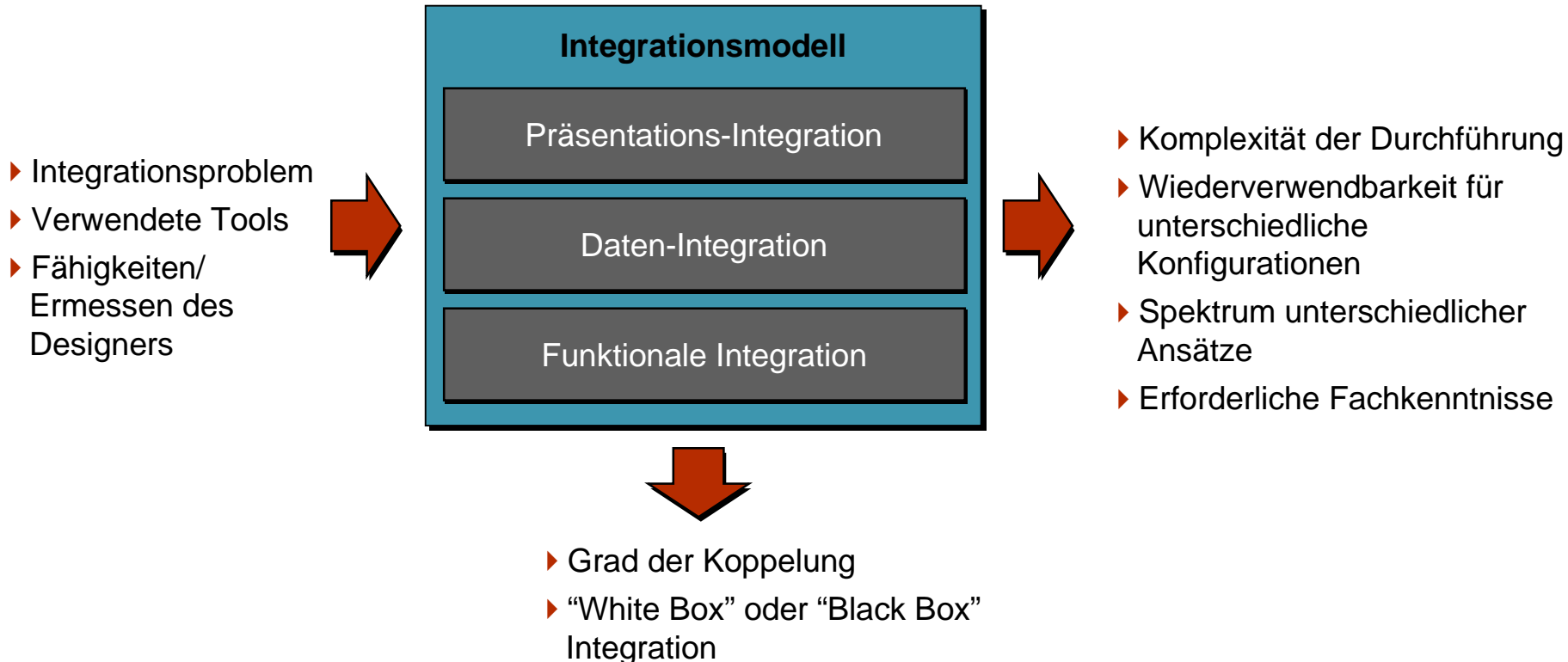
Ex-post Integration

Schaffung betrieblicher Anwendungssysteme durch die Kombination existierender Anwendungen oder Systemkomponenten und ggf. Ergänzung durch neue Teile

- ▶ “Globales Integrationskonzept”
- ▶ Heterogenität in bezug auf Betriebssysteme, Netzwerke, Hardwareplattformen, Entwicklungsumgebungen oder -konzepten
- ▶ Umsetzung einer integrierten Anwendungssystem-Architektur im Unternehmen und ggf. über Unternehmensgrenzen hinweg

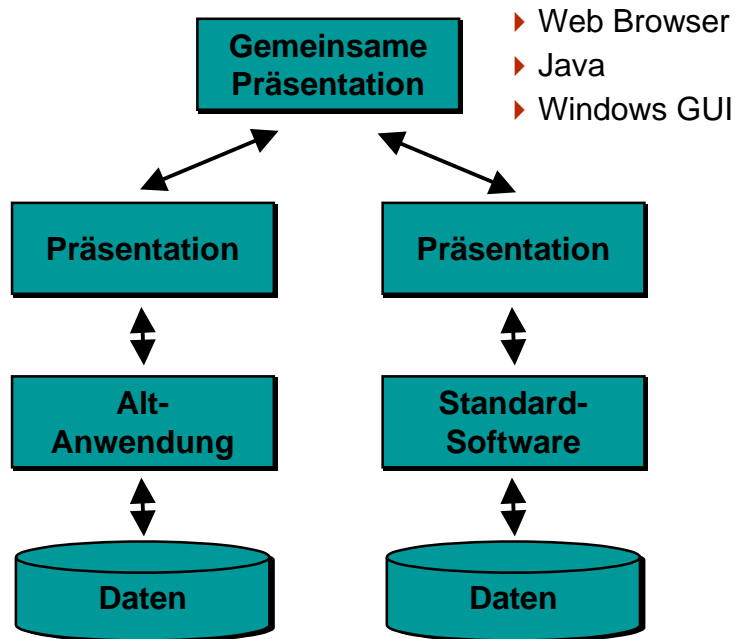
Die Integration von Anwendungssystemen kann an drei unterschiedlichen Punkten der Applikation ansetzen: der Präsentations-, der Daten- oder der Funktionsebene

Wahl des Integrationsmodells



Das Präsentations-Integrationsmodell integriert über die Benutzerschnittstellen der Anwendungen

Präsentations-Integrationsmodell

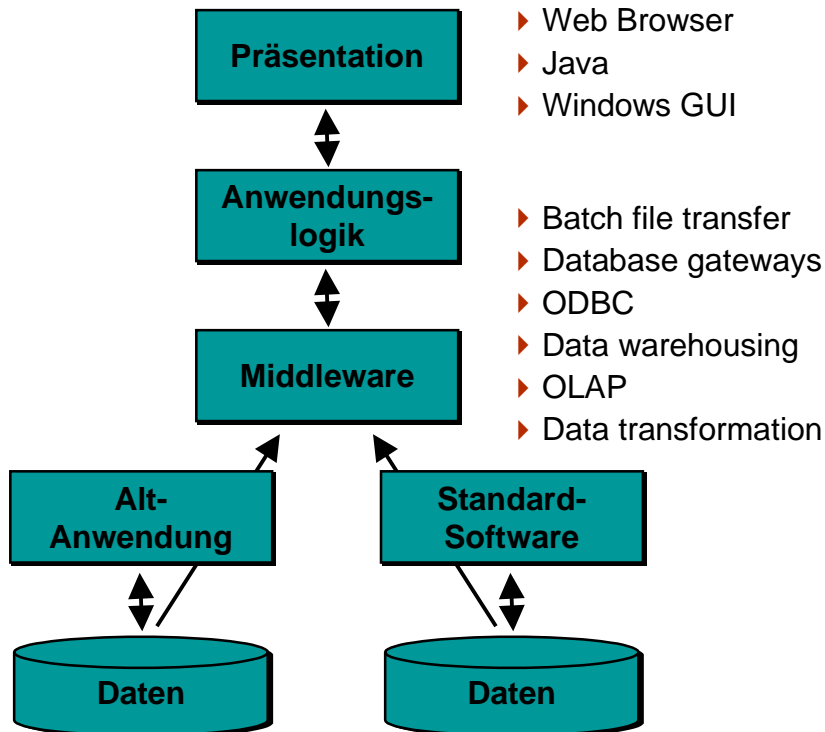


- + Einfach und schnell durchzuführen
- + Gut dokumentierte oder selbsterklärende Präsentationslogik
- + Verfügbarkeit leistungsfähiger Tools („Screen Scraping“)
- Funktionalität beschränkt auf die Präsentationen der integrierten Anwendungen
- Mögliche Performance Probleme durch zusätzliche Softwareschicht

- Beispiele:**
- ▶ Schaffen einer MS Windows Schnittstelle zu einer Mainframe-Anwendung
 - ▶ Gemeinsames HTML Interface zu einer SAP R/3 Anwendung und einer Mainframe Anwendung
 - ▶ Gemeinsames Java Interface für mehrere Mainframe Anwendungen

Die Daten-Integration umgeht die Präsentations- und Logik-ebenen und greift zum Zweck der Integration direkt auf die Datenbanken einer Anwendung zu

Daten-Integrationsmodell



- ▶ Web Browser
- ▶ Java
- ▶ Windows GUI

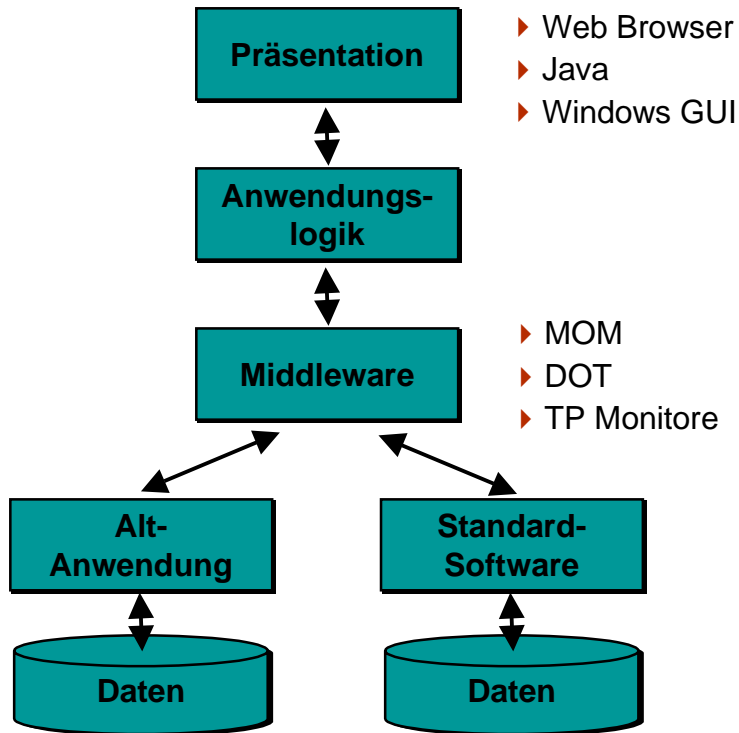
- ▶ Batch file transfer
- ▶ Database gateways
- ▶ ODBC
- ▶ Data warehousing
- ▶ OLAP
- ▶ Data transformation

- + Erfordert keine Modifikation der Datenbanken oder Applikationslogik
- + Zugriff auf ein breiteres Datenspektrum als über die Präsentationsebene verfügbar, auch auf Untermengen von Daten
- + Bei leicht zugänglichen Datenbanken schnelle Integration möglich
- + Erprobte Tools und Techniken
- + hohe Wiederverwendbarkeit
- Notwendigkeit der erneuten Programmierung der Logik
- Abhängigkeit vom verwendeten Datenmodell (Wartungsaufwand)
- Mögliche Integritätsprobleme durch das Umgehen der Applikationslogik

- Beispiele:**
- ▶ Integration von Kundendaten aus Sybase, IBM DB2 und SAP R/3 Datenbanken in eine Call-Center-Anwendung
 - ▶ Nächtlicher Batch-File-Transfer von Oracle und Mainframe Daten in ein EIS System
 - ▶ Zugriff auf Peoplesoft und spezifische Oracle Datenbanken über ODBC-Schnittstellen

Das Funktionale Integrationsmodell setzt direkt an der Logikebene der Anwendung an

Funktionales Integrationsmodell



- + Weites Spektrum lösbarer Integrationsprobleme (einschließlich Präsentations- und Datenintegration)
 - „Zero Latency Enterprise“
 - „Straight Through Processing“
- + Hohe Wiederverwendbarkeit der Softwarekomponenten
- Höhere Komplexität, Lernkurve
- Modifikation der Anwendungen nötig
- Zugriff auf Logik kann schwierig sein, wenn der Zugriff auf Quellcode oder APIs nicht gegeben sind

- Beispiele:**
- ▶ Aktualisierung einer Kundenadresse aus dem Call Center in einer Java Anwendung, einer Mainframeanwendung und einer Oracle Datenbank
 - ▶ Bearbeitung einer Online Bestellung über Auslieferung und Bezahlung hinweg
 - ▶ Schaffung einer einheitlichen Kundensicht über alle betrieblichen Anwendungen und Datenbanken durch Schnittstellen der Middleware zu allen Anwendungen

Enterprise Application Integration (EAI) stellt einen neuen Ansatz zur Integration von Anwendungssystemen dar

- ▶ EAI ist **kein Produkt**, sondern ein **Ansatz zur Entwicklung integrierter Anwendungssysteme**
- ▶ Dieser Ansatz ist eine Kombination aus **Architektur, Technologie und Methodik**
- ▶ Er ist geprägt durch die Verfügbarkeit von Softwareprodukten, die **vorgefertigte Integrationslösungen** bereitstellen
- ▶ EAI fokussiert sich auf die **Integration von Geschäftsprozessen und Daten**, während traditionelle Methoden oft nur datenorientiert sind
- ▶ EAI beinhaltet die Ziele der **Wiederverwendung** sowie der **Verteilung von Prozessen und Daten**
- ▶ EAI ermöglicht es Nutzern, Anwendungen auch **nahezu ohne Kenntnis der technischen Details** zu integrieren

Das grundlegende Ziel von EAI ist es, Organisationen die schnelle und einfache Integration von Anwendungen zu ermöglichen

Definition Enterprise Application Integration (EAI)

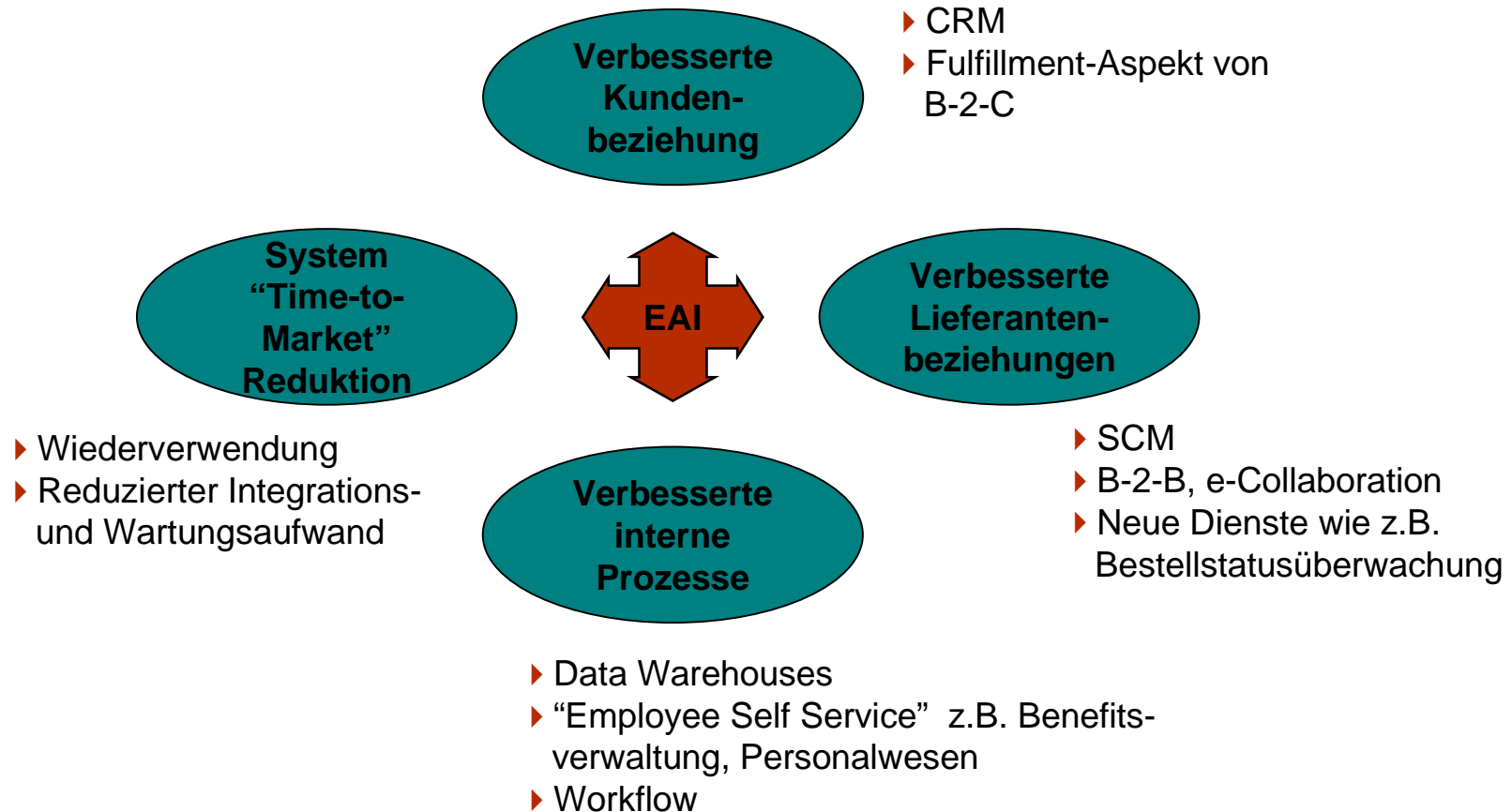
Unter Enterprise Application Integration (EAI) versteht man die Schaffung von betrieblichen Anwendungssystemen durch die Kombination einzelner Anwendungen unter Verwendung einer gemeinsamen Middleware.

Middleware bezeichnet dabei anwendungsunabhängige Technologien, die Dienstleistungen zur Vermittlung zwischen Anwendungen anbieten. Dabei verbirgt Middleware die Komplexität der zugrunde liegenden Betriebssysteme und Netzwerke, um die einfache Integration verschiedener Anwendungen zu erleichtern.

EAI unterscheidet sich von traditionellen Ansätzen zur Anwendungsintegration

Traditionelle Anwendungsintegration	EAI
<ul style="list-style-type: none">▶ Datenexport/-import Programme und traditionelle Middleware (z.B. Datenbankgateways)	<ul style="list-style-type: none">▶ Middleware-basierte Integrationsplattformen
<ul style="list-style-type: none">▶ Hoher Eigenentwicklungsanteil	<ul style="list-style-type: none">▶ Vorgefertigte Integrationslösungen
<ul style="list-style-type: none">▶ „Punkt-zu-Punkt“-Verbindungen	<ul style="list-style-type: none">▶ „Any-to-any“ und „many-to-many“ Verbindungen
<ul style="list-style-type: none">▶ Datenfokus	<ul style="list-style-type: none">▶ Geschäftsprozeßfokus
<ul style="list-style-type: none">▶ Anwendungs- und technologieabhängig	<ul style="list-style-type: none">▶ Anwendungs- und technologieunabhängig
<ul style="list-style-type: none">▶ Statisch	<ul style="list-style-type: none">▶ Konfigurierbar

Aus betriebswirtschaftlicher Sicht liegt der Nutzen von EAI in der Unterstützung kritischer neuer Lösungen



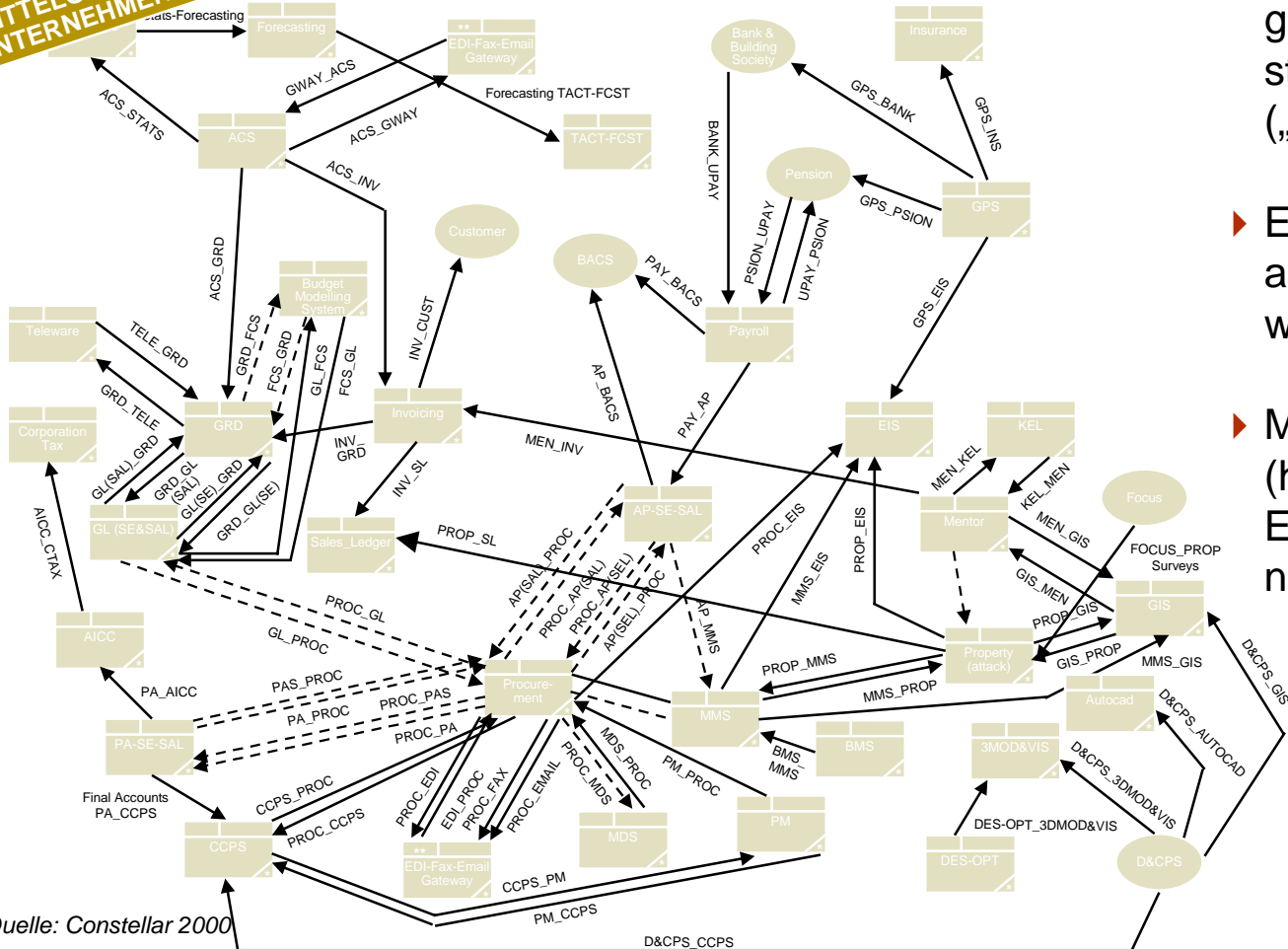
Letztendlich strebt EAI nach der Automatisierung von Geschäftsprozessen über Funktions-, Applikations- und ggf. Unternehmensgrenzen hinweg

- ▶ Die meisten Geschäftsprozesse im Unternehmen sind bereits automatisiert - jedoch existieren sie oft auf verschiedenen Anwendungssystemen, die nur lose gekoppelt sind und u.U. manuelle Kontrollen, Autorisierungen oder die erneute Eingabe von Daten beinhalten
- ▶ Das Ziel der Prozeßautomation durch EAI ist es, Informationen von Anwendung zu Anwendung weiterzureichen und dabei Regeln anzuwenden, um ein bestimmtes Geschäftsziel zu erreichen
- ▶ Damit stellt Prozeßautomation durch EAI eine zusätzliche Ebene definierbarer und zentral verwalteter Prozesse (oder Workflow) bereit, die auf in einer Anzahl von Anwendungen abgebildeter Prozesse und Datenstrukturen aufsetzen
- ▶ Das Design und die Verwaltung dieser Metaprozesse können durch spezifische Tools (Workflow Engine) oder durch Funktionalitäten der EAI-Middleware unterstützt werden

Punkt-zu-Punkt-Verbindungen führen langfristig zu einem Schnittstellen-Chaos

BEISPIEL EINES MITTELGROßEN UNTERNEHMENS

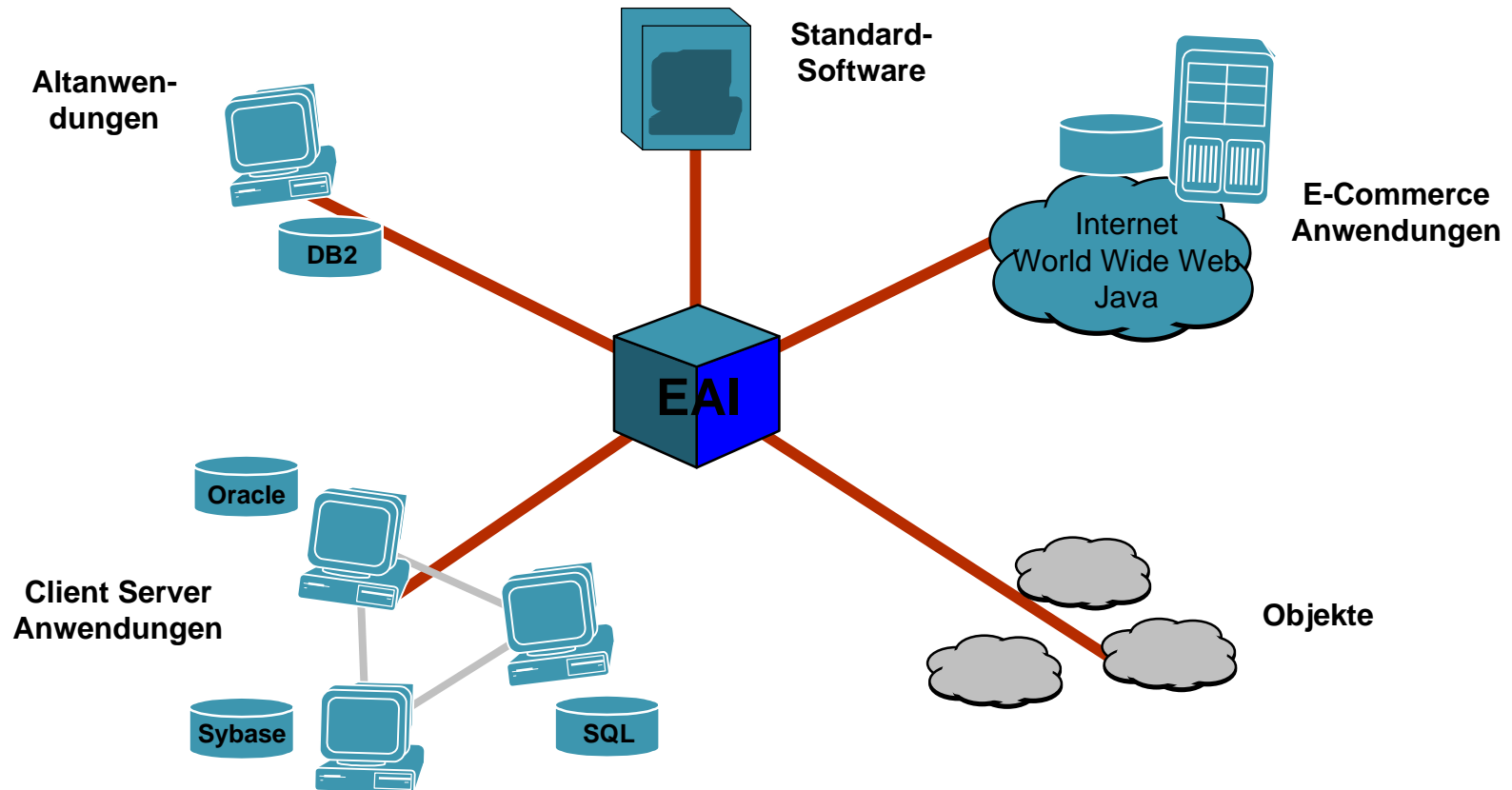
„Schnittstellen-Chaos“



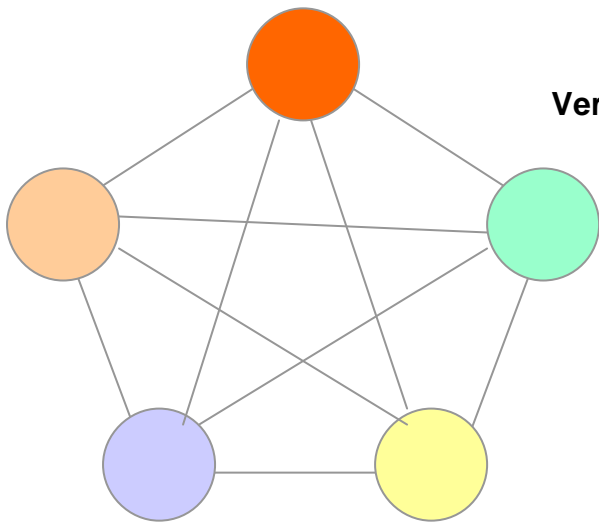
- ▶ Entwicklungsressourcen gebunden durch Schnittstellenprogrammierung („Verkabelung“)
- ▶ Enormer Wartungsaufwand z.B. Releasewechsel
- ▶ Mangelnde Flexibilität (hohe Einführungszeiten, Entwicklungskosten) bei neuen Anwendungen

EAI minimiert die Anzahl der benötigten Schnittstellen zu einer gemeinsamen Middleware

Typische EAI „Hub-and-Spoke“ Topologie

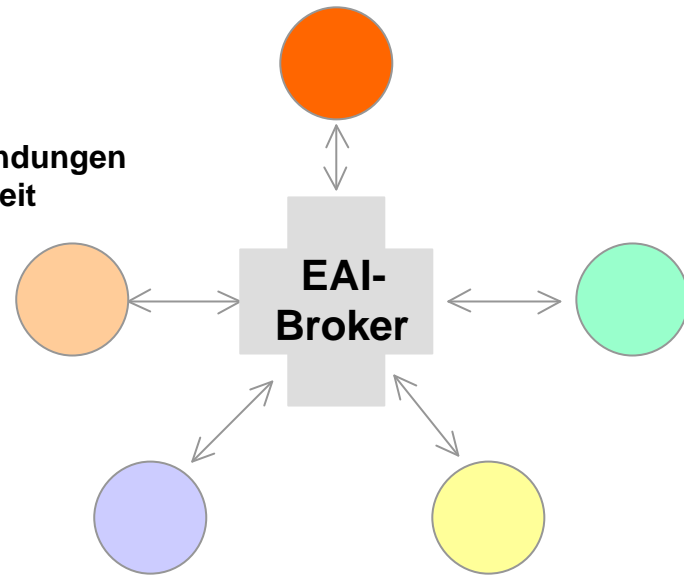


EAI: Direkte Kopplung vs. Message Broker



**Punkt-zu-Punkt-Integration
(direkte Kopplung)**

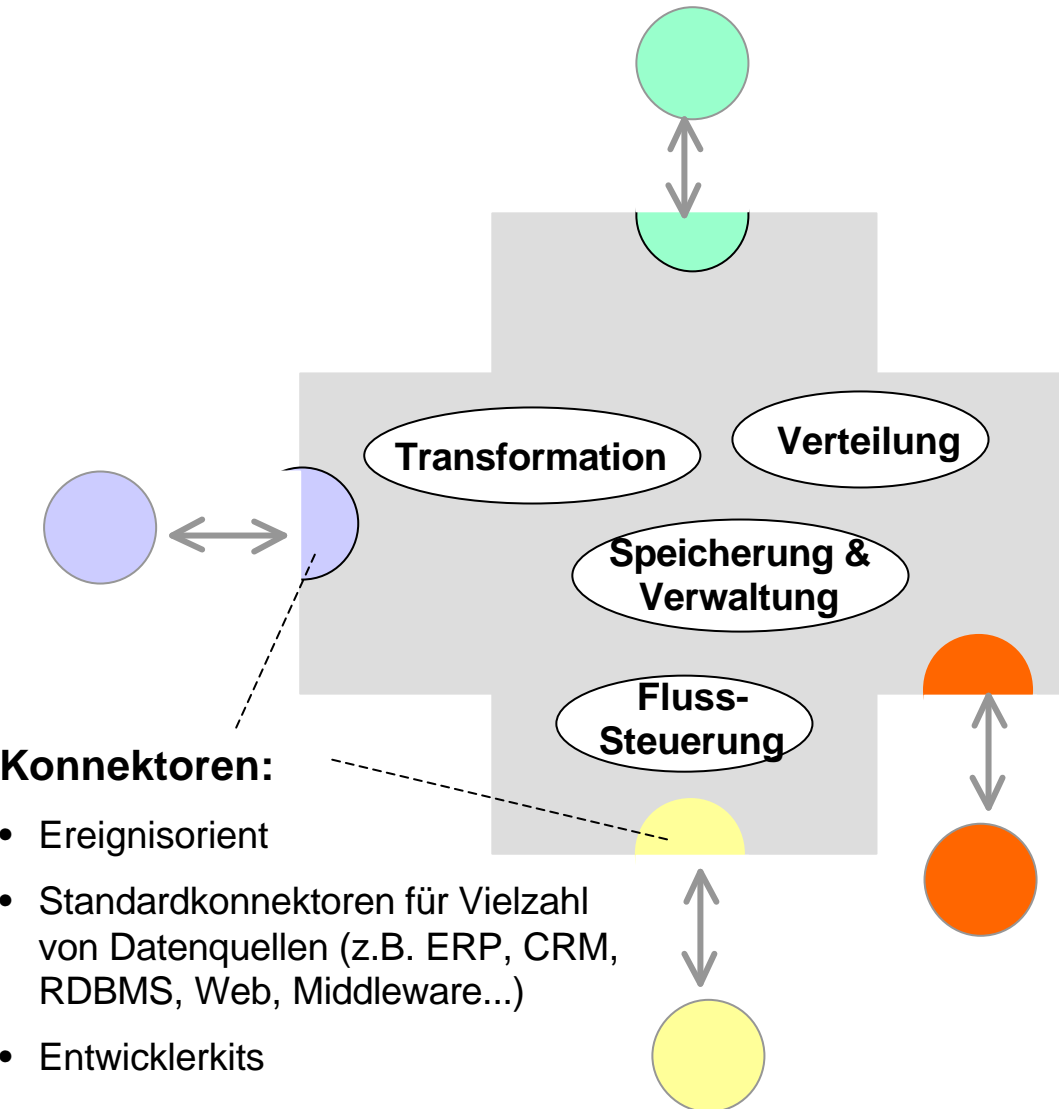
Verknüpfung von Anwendungen
über zentrale Einheit



Message Broker



Beispiel: Message Broker



- **Synchrone Kopplung:**

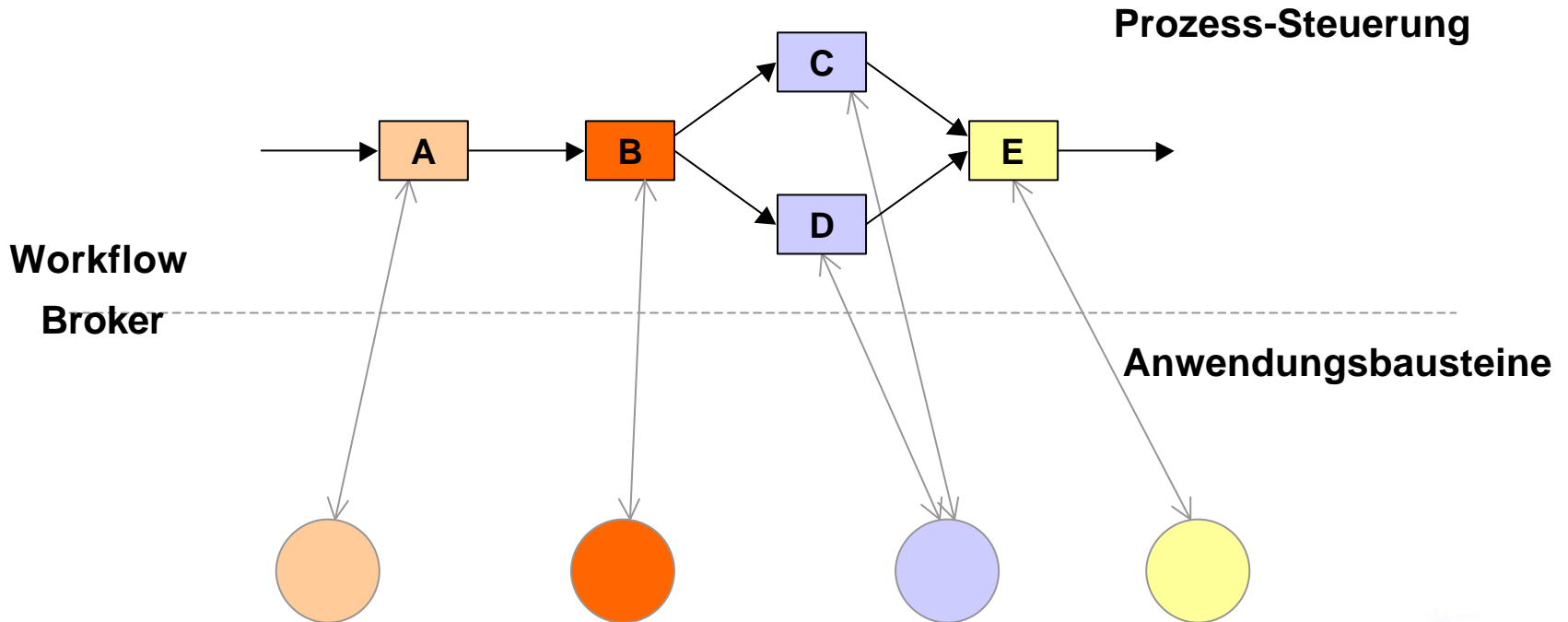
- Request / Reply
- Polling
- One Way

- **Asynchrone Kopplung**

- Message Passing
- Publish / Subscribe
- Broadcast



EAI: Prozessorientierte Kopplung (1)

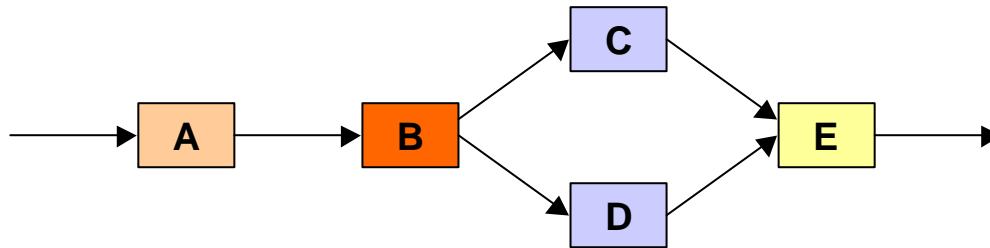


Anwendungsintegration über Workflow:

- Schnelle, flexible Prozessgestaltung
- Schnelle Anpassbarkeit der Prozesse
- Übergänge zwischen den Anwendungen

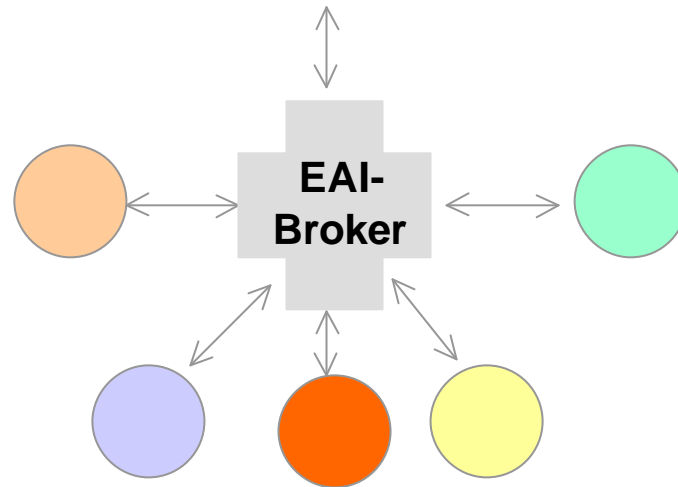


EAI: Prozessorientierte Kopplung (2)



Workflow

Broker



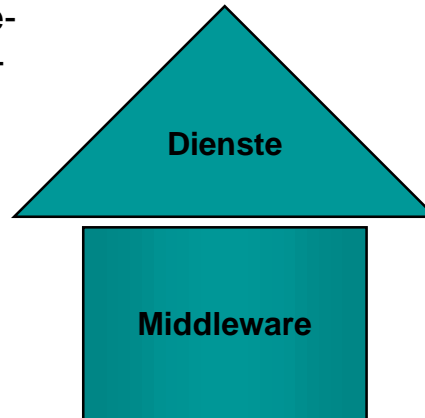
Das Ergebnis von EAI im Unternehmen ist ein globales Integrationskonzept im Sinne einer EAI Architektur

- ▶ Im Rahmen eines EAI Projektes muß im Unternehmen durch die Anwendung einer geeigneten Methode und die Auswahl geeigneter Technologien im Rahmen der Umsetzung die EAI Architektur physisch konstruiert werden
- ▶ Die EAI Architektur identifiziert die Systemkomponenten, beschreibt die Beziehungen zwischen diesen Komponenten und präsentiert Standards sowie Richtlinien oder Vorgaben für das Design einzelner Komponenten
- ▶ Die Anforderungen an die EAI Architektur leiten sich aus den Unternehmenszielen ab
- ▶ Bei der Umsetzung des Integrationskonzeptes gibt es drei sich ergänzende Ansätze
 - Erweiterung existierender Systemkomponenten
 - Austausch von Komponenten durch neue technologische Lösungen
 - Verbindung der Systemkomponenten untereinander

Eine EAI Architektur als strukturierte Kombination von Technologien besteht aus vier wesentlichen Bausteinen

EAI Architektur Bausteine

- ▶ Dienste unterstützen die Middleware oder die Kommunikationsfähigkeiten
- ▶ Z.B. Directory, Versionsmanagement, Sicherheit, Notifikationen, Workflow

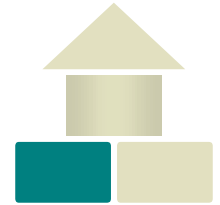


- ▶ Zwei Typen: synchron und asynchron
- ▶ Eine gute EAI Architektur unterstützt beide Modelle

- ▶ Middleware verbirgt die Komplexität von Quell- und Zielsystem

- ▶ Zwei primäre Methoden: Messaging und Schnittstellen
- ▶ Konnektoren schaffen den Zugang zu den Anwendungen

Synchrone und asynchrone Kommunikation kann auf unterschiedliche Art und Weise umgesetzt werden

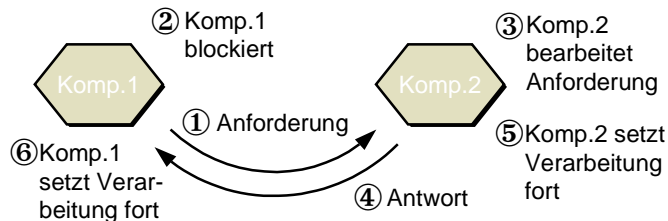


Kommunikationsmodell

Synchrone Kommunikation

Der Sender einer Anforderung wartet auf die Antwort, bevor er intern mit der Verarbeitung fortfährt

► „Request / Reply“



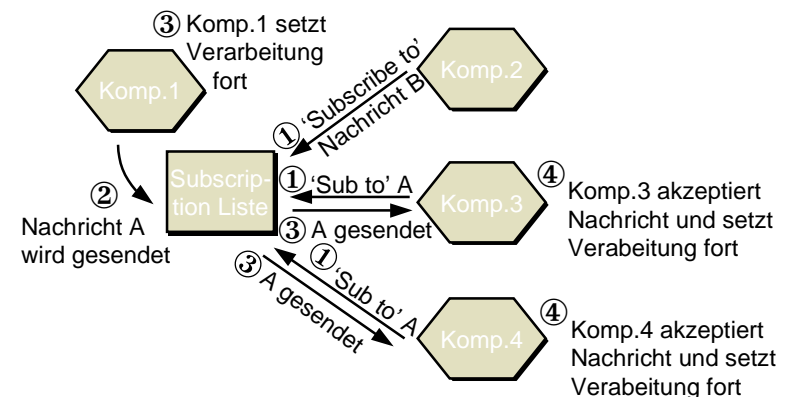
Weitere Arten der synchronen Kommunikation :

- „One way“
- „Synchronous Polling“

Asynchrone Kommunikation

Keine Abhängigkeit von interner Verarbeitung und Kommunikation bei Sender und Empfänger

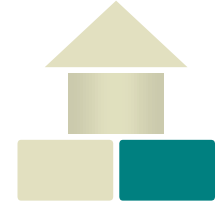
► „Publish/Subscribe“



Weitere Arten der asynchronen Kommunikation :

- „Message Passing“
- „Broadcast“

Unter der Integrationsmethode versteht man den gewählten Ansatz, um eine Anforderung oder Nachricht von einem Sender zu einem Empfänger zu bekommen



Messaging

- ▶ Der Sender erstellt eine Nachricht, die die Kontrollinformationen bzgl. der auszuführenden Aktion sowie die zur Ausführung benötigten Daten enthält
- ▶ Nachrichten sind anwendungsunabhängig
- ▶ Geringer Koppelungsgrad zwischen Applikationen - alle benötigten Informationen sind in der Nachricht enthalten
- ▶ Hohe Flexibilität, da die Kontrollinformationen leicht verändert oder ergänzt werden können
- ▶ Erfordert gute Dokumentation oder spezielle Dienste, da die Verbindungen von Nachrichten zu Applikationen nicht leicht sichtbar sind

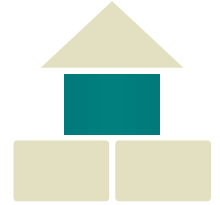
Schnittstellen Definitionen

- ▶ Der Sender kommuniziert über eine Schnittstelle, die die aufrufbaren Aktionen definiert, alle benötigten Daten werden über diese Schnittstelle gesendet
- ▶ Schnittstellen sind anwendungsspezifisch
- ▶ Der Integrationsprozeß gleicht traditionellen Programmstrukturen (Procedure Call, Funktionen) mit dem Unterschied, daß Schnittstellen nach außen sichtbar sind und von anderen Anwendungen genutzt werden können
- ▶ Schwieriger zu verändern oder erweitern
- ▶ Selbsterklärend bezüglich verfügbarer Funktionen

Konnektoren / Adapter

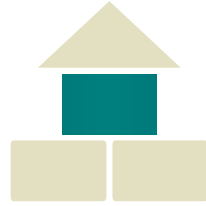
- ▶ Unabhängig von der Integrationsmethode werden Konnektoren / Adapter als Zugangspunkt zur Anwendung benötigt
- ▶ Konnektoren / Adapter sind Logikbausteine mit dem einzigen Zweck, strukturierten Zugang zur Präsentation, Daten oder Logik einer Applikation zu ermöglichen. Dabei verbirgt er die Komplexität der Übersetzung einer Nachricht oder Anforderung zum Gebrauch durch eine Anwendung
- ▶ Konnektoren / Adapter bieten dabei Zusatzdienste an

EAI Architekturen basieren auf Middleware als zentralem Bestandteil verteilter Anwendungssysteme



- ▶ Software zur Kommunikation zwischen zwei oder mehreren Anwendungssystemen
- ▶ verbirgt dabei die Komplexitäten von Quell- und Zielsystem
- ▶ Vielzahl von technischen Lösungen
- ▶ zunehmende Verschmelzung der Funktionalitäten
- ▶ entwickeln sich zu “Commodities”

Basierend auf den ihnen zugrundeliegenden Technologien lassen sich verschiedene Typen von Middleware unterscheiden



Grundlegende Typen von Middleware

Remote Procedure Calls (RPC)	<ul style="list-style-type: none">▶ Integration auf Prozedurebene▶ Ermöglicht die Ausführung von Prozeduren über ein Netzwerk
Database Access Middleware	<ul style="list-style-type: none">▶ Integration auf Datenebene durch Zugriff auf verteilte Daten in Dateien oder Datenbanken▶ Ermöglicht Abfragen oder Datenbewegungen über ein Netzwerk
Message oriented Middleware (MOM)	<ul style="list-style-type: none">▶ Basiert auf Nachrichten als Integrations-methode▶ Zur Erzeugung, Veränderung, Speicherung und Kommunikation von Nachrichten
Distributed Object Technology (DOT)	<ul style="list-style-type: none">▶ Ausweitung der Konzepte objektorientierter Technologie auf verteilte Anwendungen▶ Schnittstellen lassen Applikationen wie Objekte erscheinen
Transaction Processing Monitors (TPMs)	<ul style="list-style-type: none">▶ Stellt die Integrität von Transaktionen sicher▶ Unterstützt Funktionalitäten wie Rollback, Fail-over, Auto Restart, Error Logging und Replikation

Anforderungen an EAI-Architekturen

● Allgemeine Anforderungen

- Plattform- und Programmiersprachenunabhängigkeit
- Verwendung von Standards
- Skalierbarkeit

● Anforderungen an die Build-Time

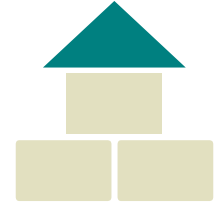
- Introspektion
- Auswahl / Wiederverwendung von Bausteinen
- Adäquate Beschreibung
- ...

● Anforderungen an die Run-Time

- Verteilungstransparenz
- Synchroner / asynchroner Aufrufe
- Late Binding
- Datenflusssteuerung
- Ausnahmebehandlung
- Sicherheit
- Transaktionen
- Nebenläufigkeit
- Visuelle Integration
- ...



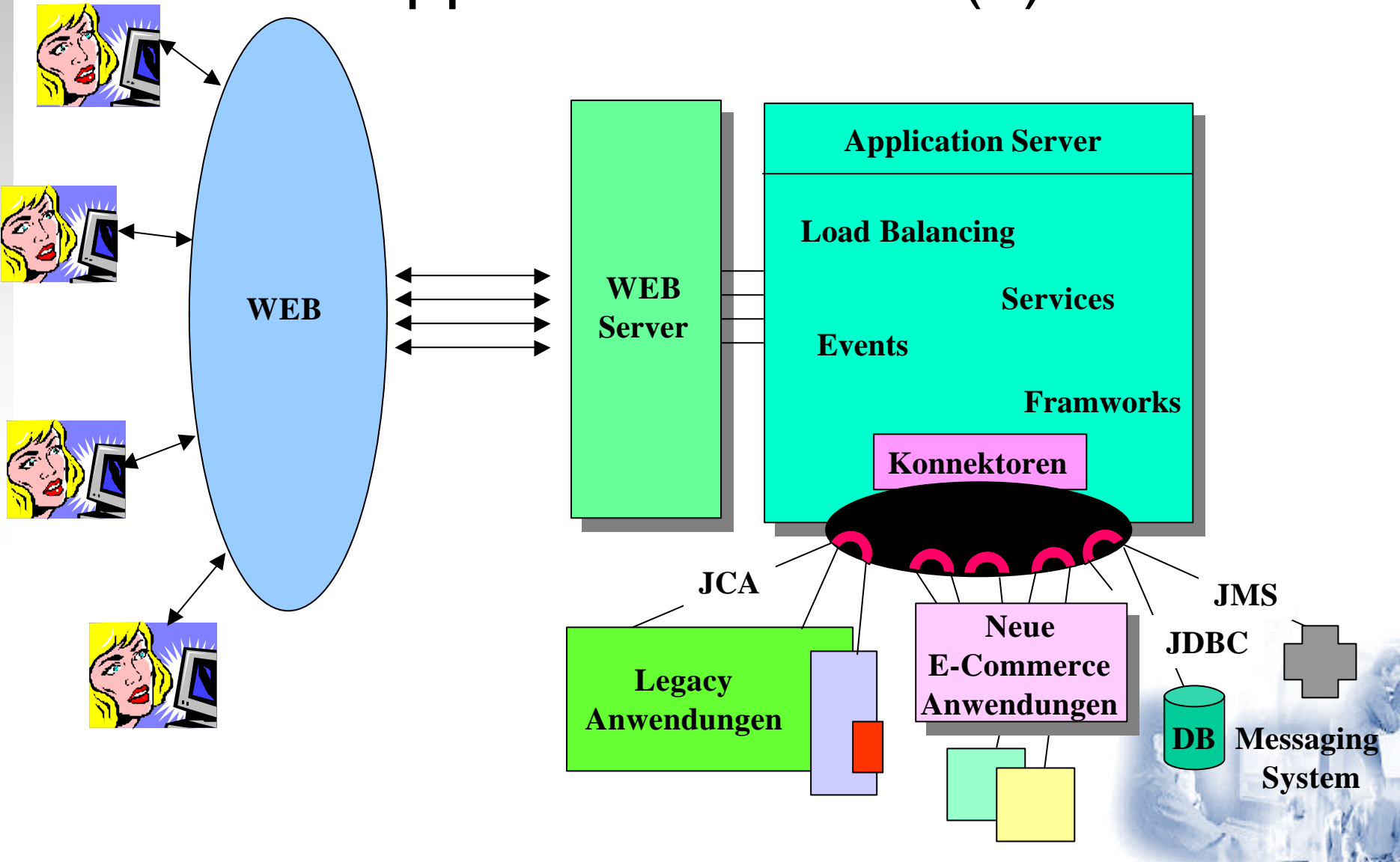
Dienste sind die funktionale Erweiterung der Grundbestandteile einer EAI-Architektur



Beispielhafte Zusatzdienste

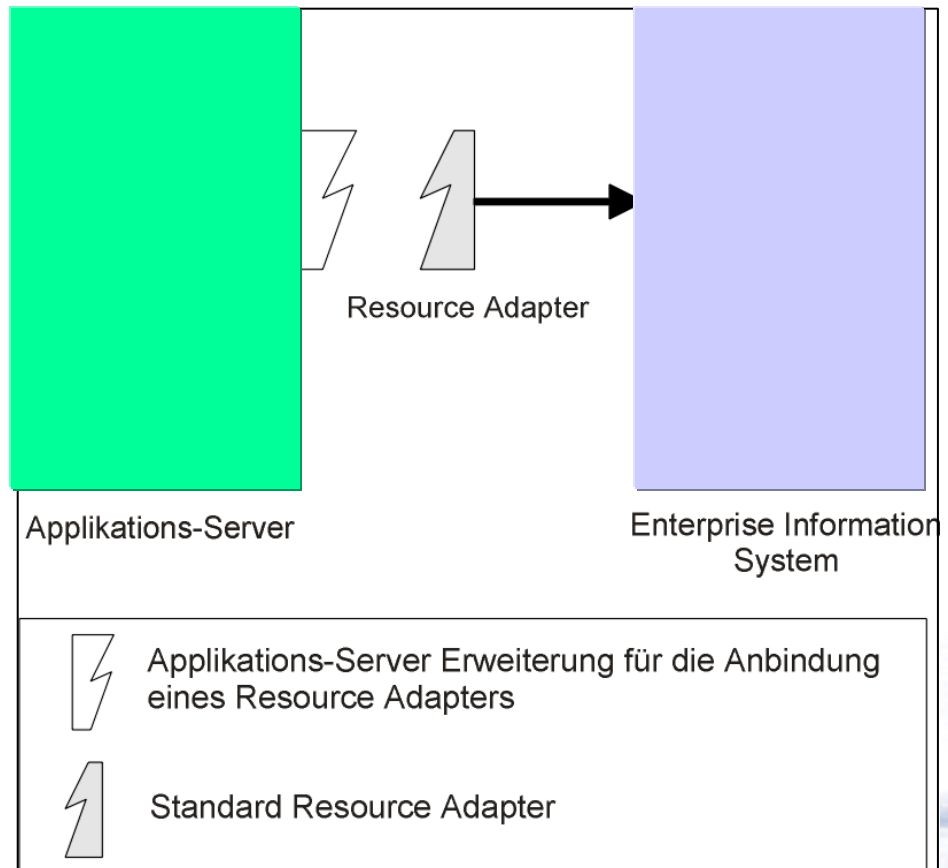
- ▶ Verzeichnis (Directory) der Komponenten, Schnittstellen, Nachrichten und Metadaten
- ▶ Versionsmanagement (Life Cycle) von Objekten und Nachrichten
- ▶ Sicherheit, insb. Authentisierung, Autorisation und sichere Kommunikation
- ▶ Konvertierung und Transformation von Daten
- ▶ Konsistenz / Persistenz von Zustandsdaten
- ▶ Identifikation und Verfolgung von Ereignissen
- ▶ Notifikationen
- ▶ Workflow

Application Server (1)

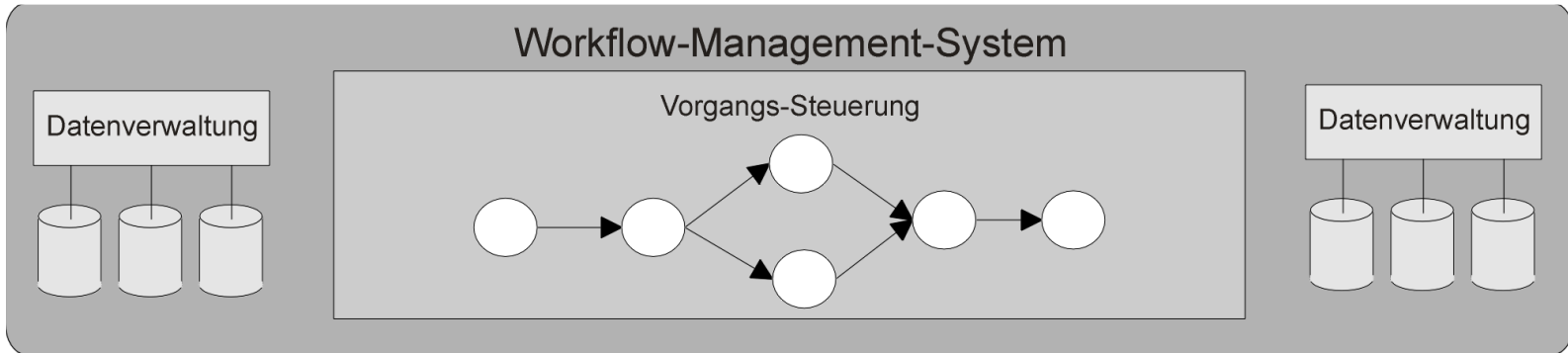


Application Server (2)

J2EE Connector Architecture



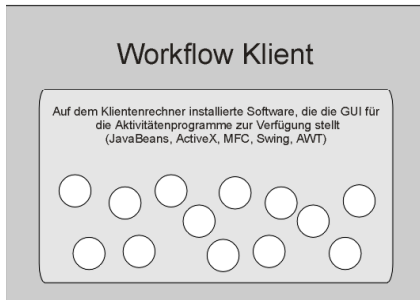
EAI: Workflow Broker



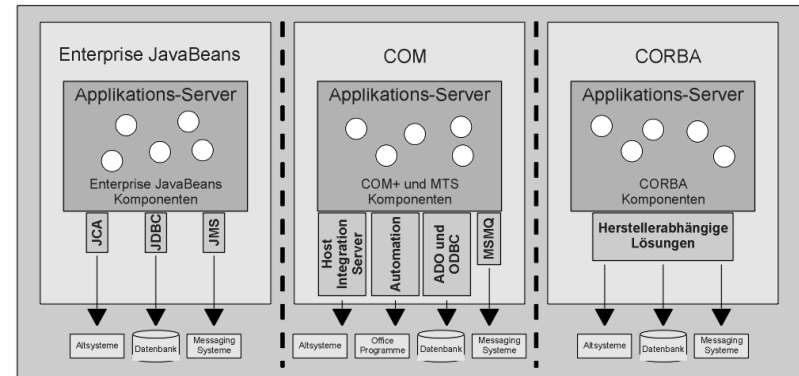
Manuelle Aktivitäten



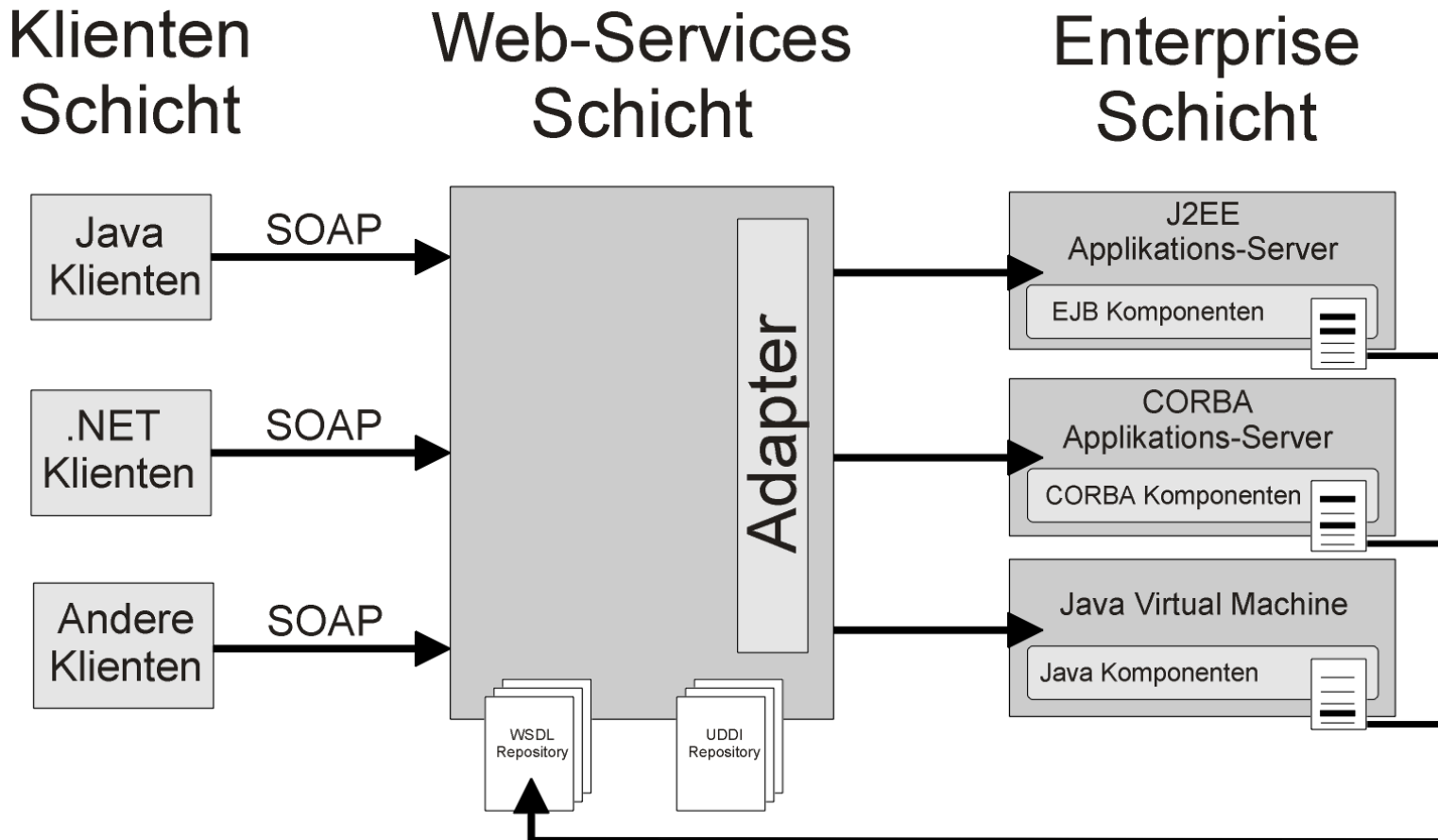
Automatische Aktivitäten



Aufruf der Anwendungslogik

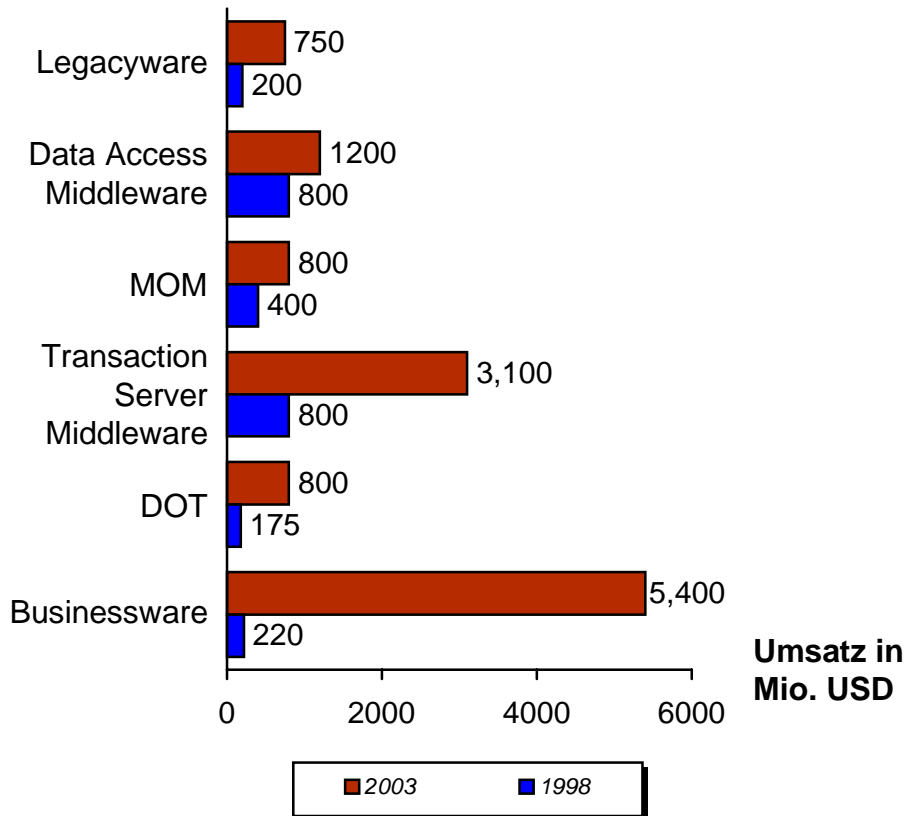


Anwendungskopplung über Web Services (1)

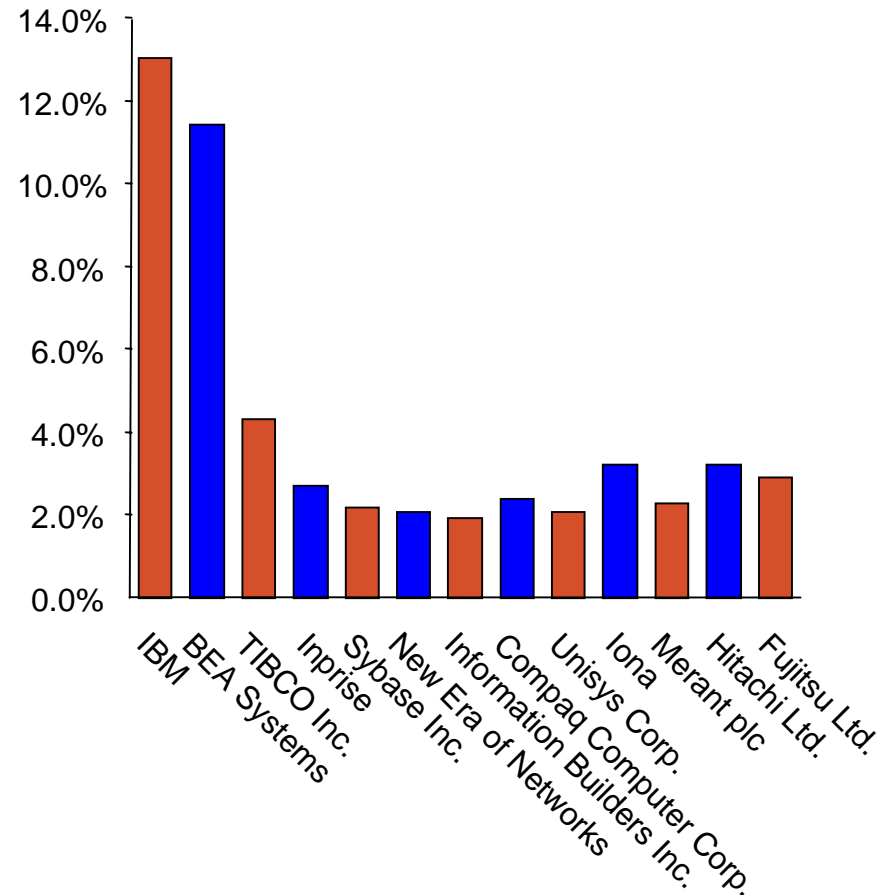


Der Middleware Markt erfährt durch EAI ein enormes Wachstum mit zahlreich vertretenen Anbietern

Middleware Markt Wachstumserwartungen

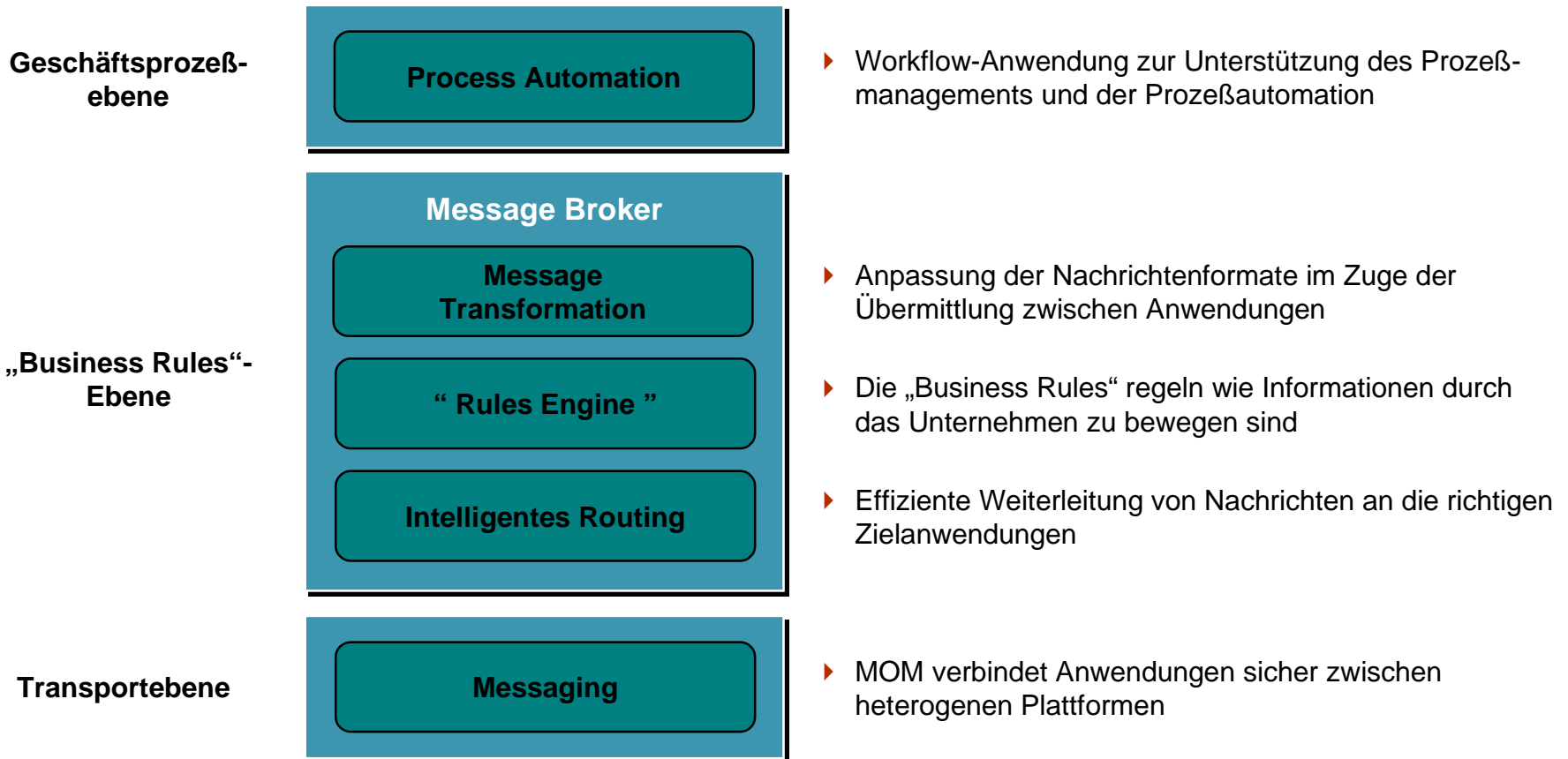


Führende Anbieter

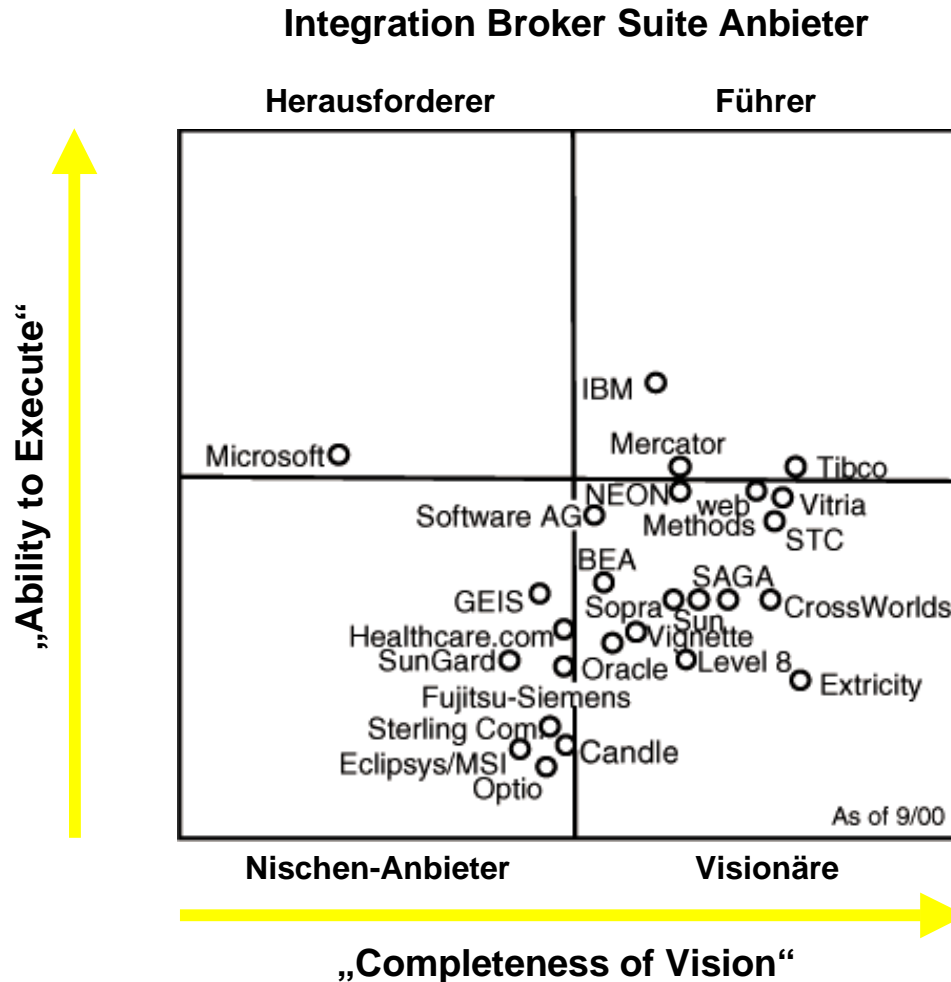


Die größten Erwartungen werden zur Zeit in sog. „Business-ware“ Produkte oder „Integration Broker Suites“ gesetzt

Integration Broker Suites



Auch für dieses „EAI Produkt“ gibt es zahlreiche Anbieter, jedoch noch ohne klaren Marktführer



Auch wenn EAI zur Zeit stark produktgetrieben ist, so handelt es sich vor allem um eine komplexe Architekturmaßnahme

Globale Reichweite

- ▶ Die Festlegung des Integrationskonzeptes macht unternehmensweite strategische Entscheidungen notwendig, die von den Geschäftszielen abgeleitet werden müssen

Standards und Richtlinien

- ▶ Eine EAI Architektur erfordert die Einhaltung technischer Standards und festgelegter Verhaltensweisen

Einbindung der IT Organisation

- ▶ Design und Wartung der EAI Architektur erfordern die entsprechenden organisatorischen Verantwortlichkeiten (insb. IT Architekten und IT Sicherheitsgremien)

Methodenkenntnisse

- ▶ Die effektive Umsetzung erfordert eine robuste Methodik (Projektmanagement, Technologieauswahl, Risikomanagement, Change Management)