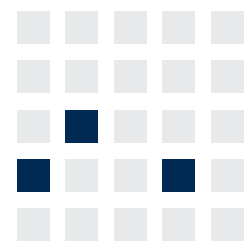




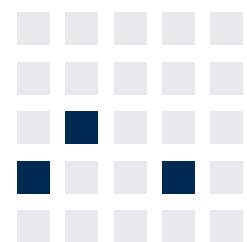
Architekturen betrieblicher Anwendungssysteme

Enterprise Application Integration



Lehrstuhl für Wirtschaftsinformatik
Prozesse und Systeme

Universität Potsdam



Chair of Business Informatics
Processes and Systems

University of Potsdam

Univ.-Prof. Dr.–Ing. habil. Norbert Gronau
Lehrstuhlinhaber | Chairholder

Mail August-Bebel-Str. 89 | 14482 Potsdam | Germany
Visitors Digitalvilla am Hedy-Lamarr-Platz, 14482 Potsdam
Tel +49 331 977 3322

E-Mail ngronau@lswi.de
Web lswi.de

Lernziele

- Was bedeutet Integration im Kontext betrieblicher Anwendungssysteme?
- Welche Ansätze zur Integration gibt es? Welche Ansätze eignen sich in Abhängigkeit der Situation?
- Welche Bedeutung kommt der prozessbezogenen Integration zu?
- Wie unterscheiden sich Integration auf Daten, Objekt und Prozessebene
- Wozu dient eine Middleware im Kontext von Integration?
- Wie können Integrationen flexibel konfiguriert werden?
- Wie können Prozesse über mehrere Anwendungen hinweg kombiniert werden?
- Wie wird eine Middleware definiert?
- Was sind Kernfunktionen von Middleware?
- Welche Middlewareparadigmen unterscheidet man?
- Was ist ein Enterprise Service Bus im Kontext von Enterprise Application Integration?
- Was ist der Unterschied zwischen Enterprise Application Integration und Middleware?



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

Integrationstechnologien

Middleware

Enterprise Application Integration

Beispiele von Integrationen

Was bedeutet Integration?

Allgemeine Begriffsfindung (Wörterbuch)

- Wiederherstellung eines Ganzen
- Einbeziehung, Eingliederung in ein großes Ganzes
- Zustand in dem sich etwas befindet, nach dem es integriert worden ist

Integration bezogen auf Informationssysteme

- Unternehmens**interne** Integration von Anwendungssystemen = Enterprise Application Integration (EAI)
- Unternehmens**übergreifende** Integration von Anwendungssystemen = Business-to-Business-Integration (B2B-Integration)

Der Integrationsbegriff ist ein mehrdimensionales Thema mit verschiedenen Sichten und Aspekten und besitzt sowohl Zustands- als auch Prozessaspekte.

Zustandsaspekte der Integration

Integrationsobjekt

- Was soll integriert werden?
- Z.B. Geschäftsstrategien, Währungssysteme, Anwendungssysteme, etc.

Integrationsstärke

- Wie soll integriert werden?
- Ausmaß an Autonomie eines Integrationsobjektes
- Feste, mittlere und lose Verbindung, gemessen am Aufwand der Desintegration

Integrationsbereich

- Kennzeichnung durch Integrationsobjekt
- Z.B: Integrationsobjekt = CRM-System -> Integrationsbereich = Anwendungssysteme in einer Anwendungssystemlandschaft
- Unterschiedliche Methoden und Herangehensweisen für Integrationsbereiche

Der Zustandsaspekt der Integration beschreibt das **Was** und **Wie** der Integration.

Prozessaspekte der Integration

Prozess der Integrierung

- Innerhalb eines Integrationsprojektes
- Überführung von einem Integrationszustand in einen anderen
- Auswahl eines Integrationsansatzes mit Methoden und Mittel

Integrationsansatz

- Methode, die zur Integration angewandt wird
- Integrationsvorgehen
- Integrationswerkzeuge und -technologien

Die Prozessaspekte der Integration beschreiben den Vorgang der Integrierung in Projektform.

Integrationsansätze zwischen Anwendungssystemen

Middleware

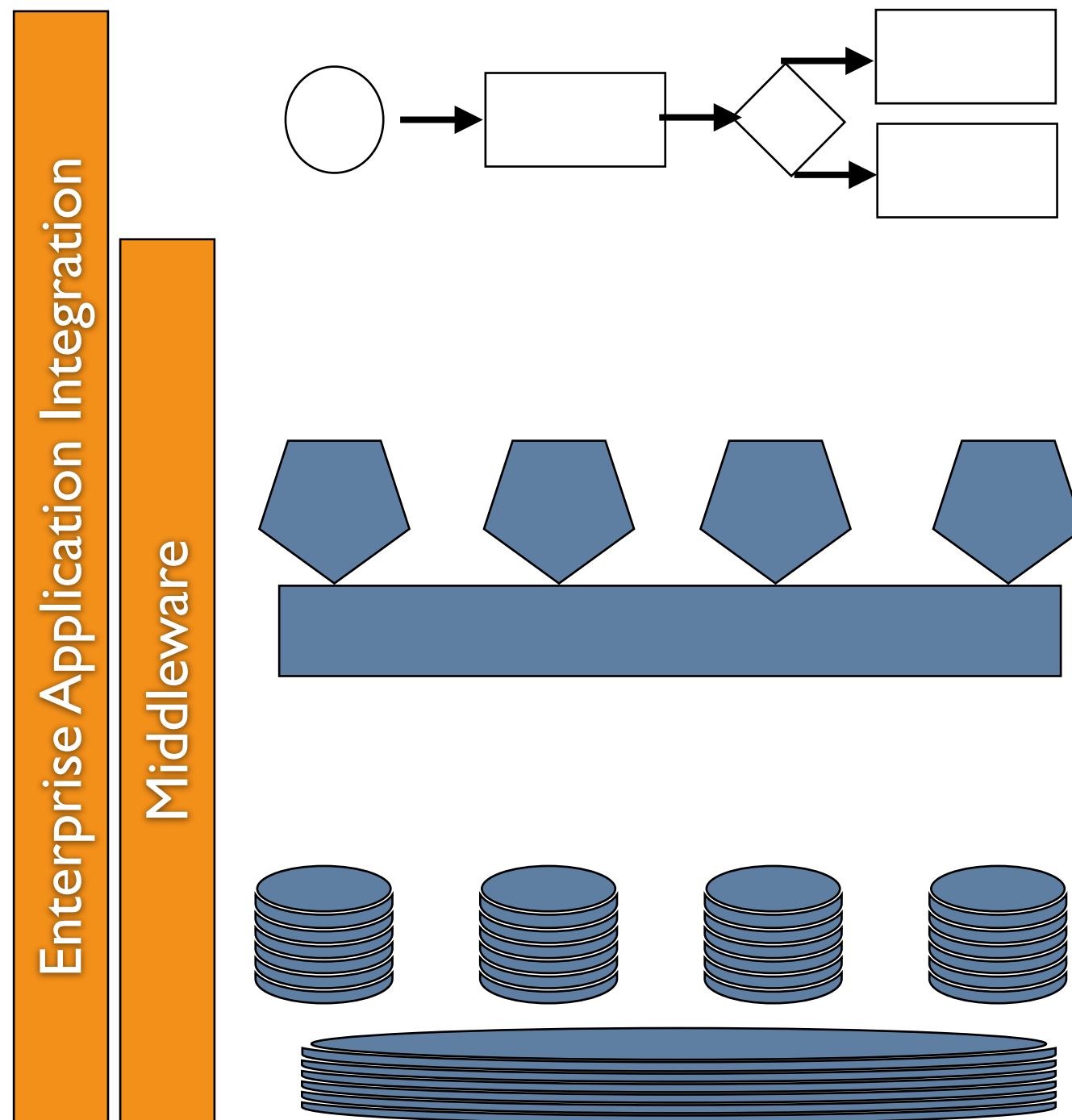
- Abstraktionskonzept einer Softwareschicht
- Vermittlung zwischen heterogenen, auf unterschiedlichen Plattformen laufenden und über Netzwerke verbundenen Anwendungssystemen und Datenquellen
- Integration zu unternehmens-/ organisationsübergreifenden Anwendungen
- Integration über Schnittstellenadaptation

Enterprise Application Integration (EAI)

- Integration entlang des definierten Geschäftsprozesses
- Beinhaltet die technischen Konzepte der Middleware
- Erweiterung um Geschäftslogik in der Vermittlung zwischen Applikationen
- Daten werden in der richtigen Abfolge an einzelne Funktionen übergeben
- Lose gekoppelte Elemente

Middleware und EAI schließen Lücken zwischen Anwendungssystemen, damit Anwendern und Kunden ein nahtloser Service geboten werden kann.

Integrations Ebenen



Prozessintegration

- Integration über Businesslogik in Prozessen
- Daten und Funktionen werden nach Wertschöpfungsfunktion verbunden

Anwendungsintegration

- Message Broker
- Nabe-Speiche-Architektur
- Business-to-Business-Integration mit externen Partnern

Datenintegration

- Data-Warehouse
- Föderierte Datenbanken
- Enterprise Service Bus



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

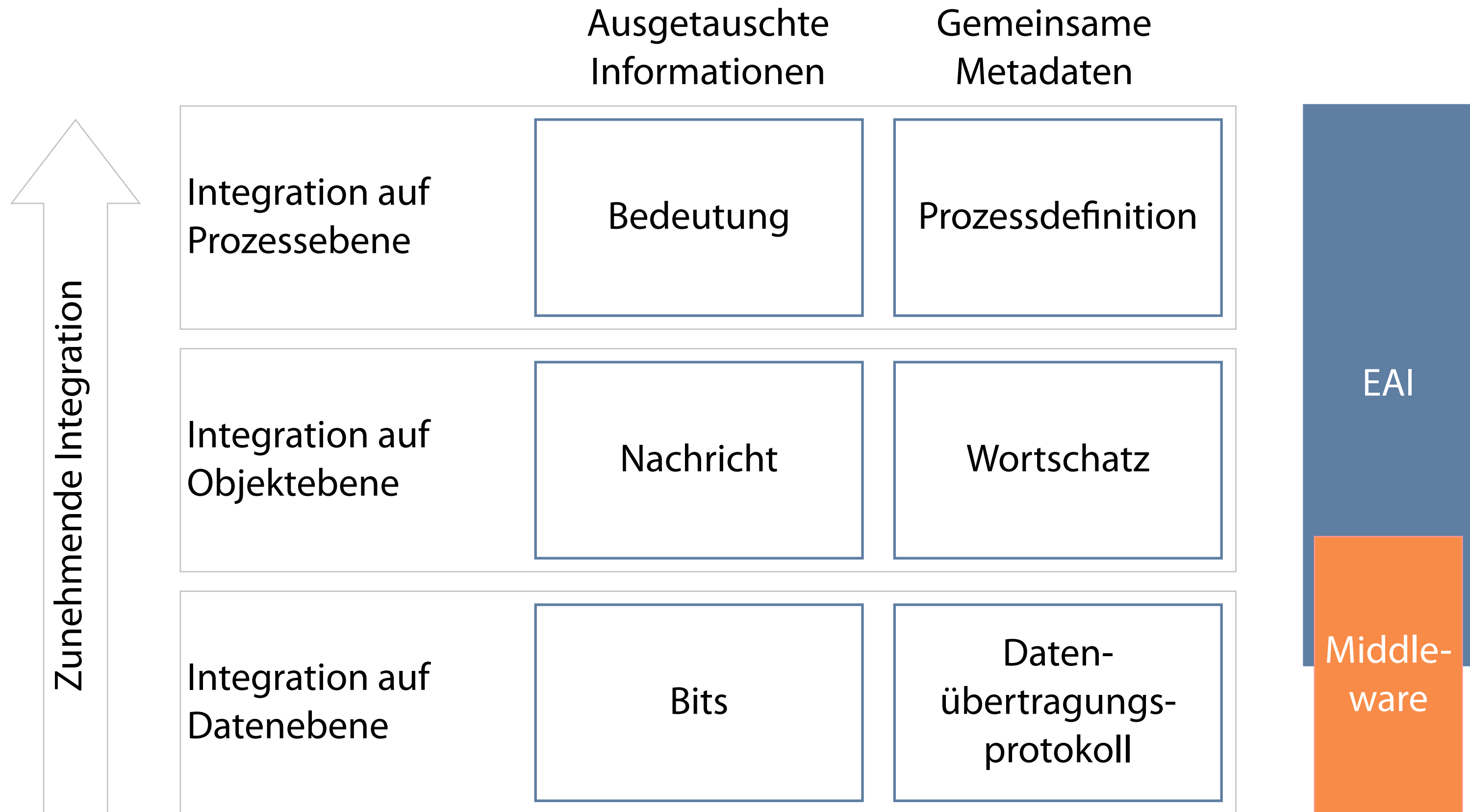
Integrationstechnologien

Middleware

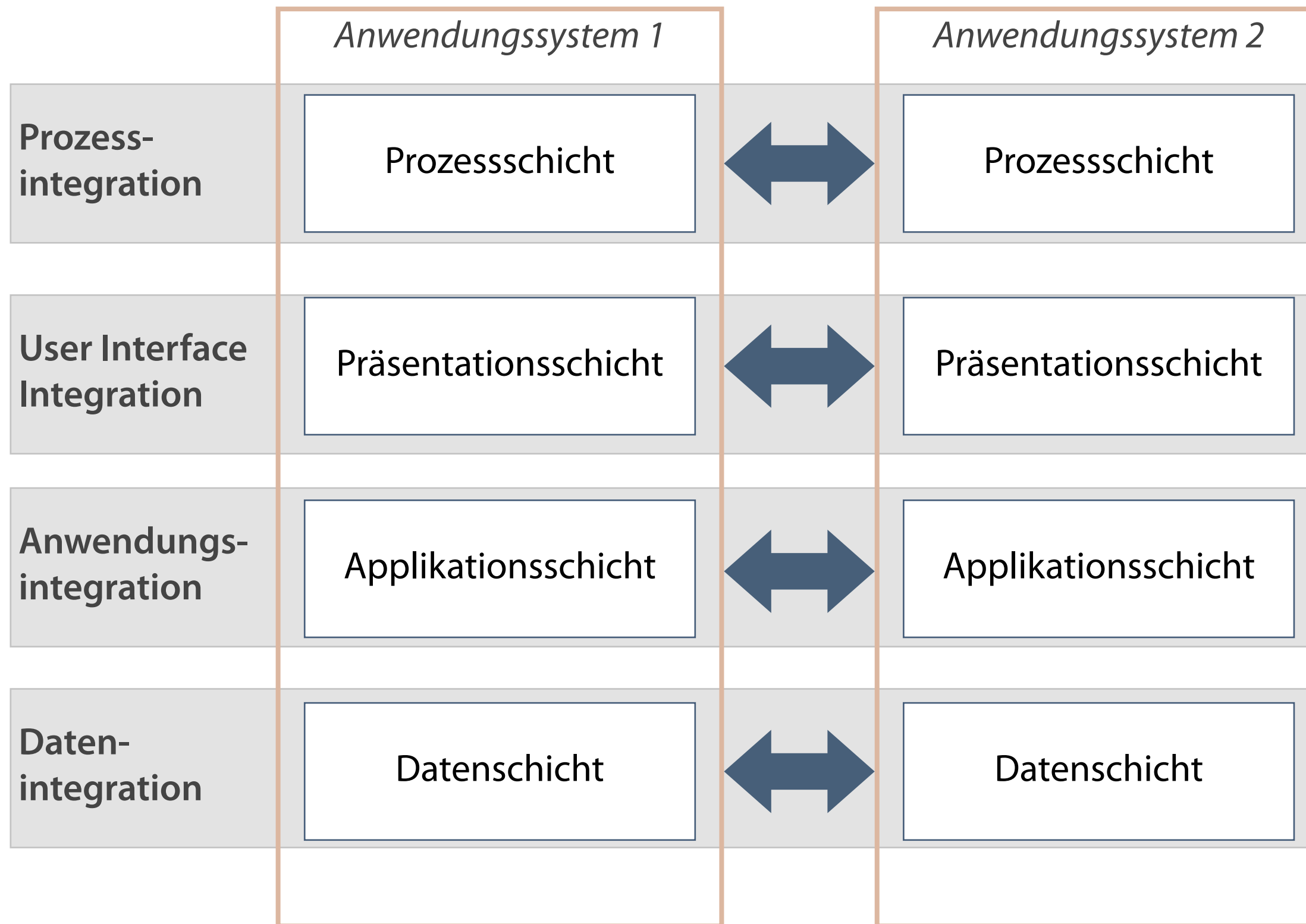
Enterprise Application Integration

Beispiele von Integrationen

Integrationsansätze zwischen Anwendungssystemen



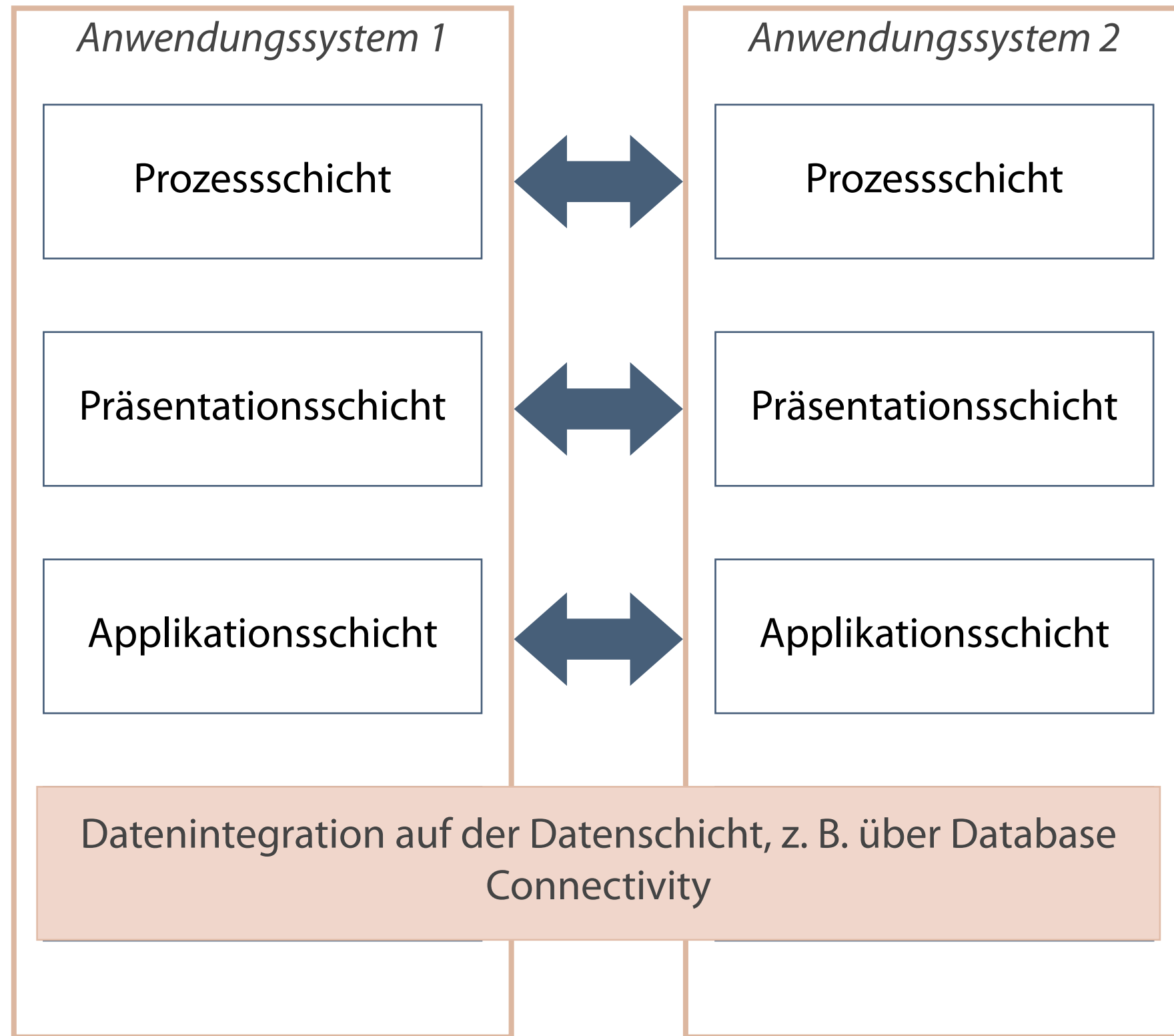
Integrationsansätze



Überblick

- Aufbauend auf Schichtenmodell der betrieblichen Anwendungssysteme
- Integrationsansätze für alle Schichten
- Auswahl erfolgt im Projekt

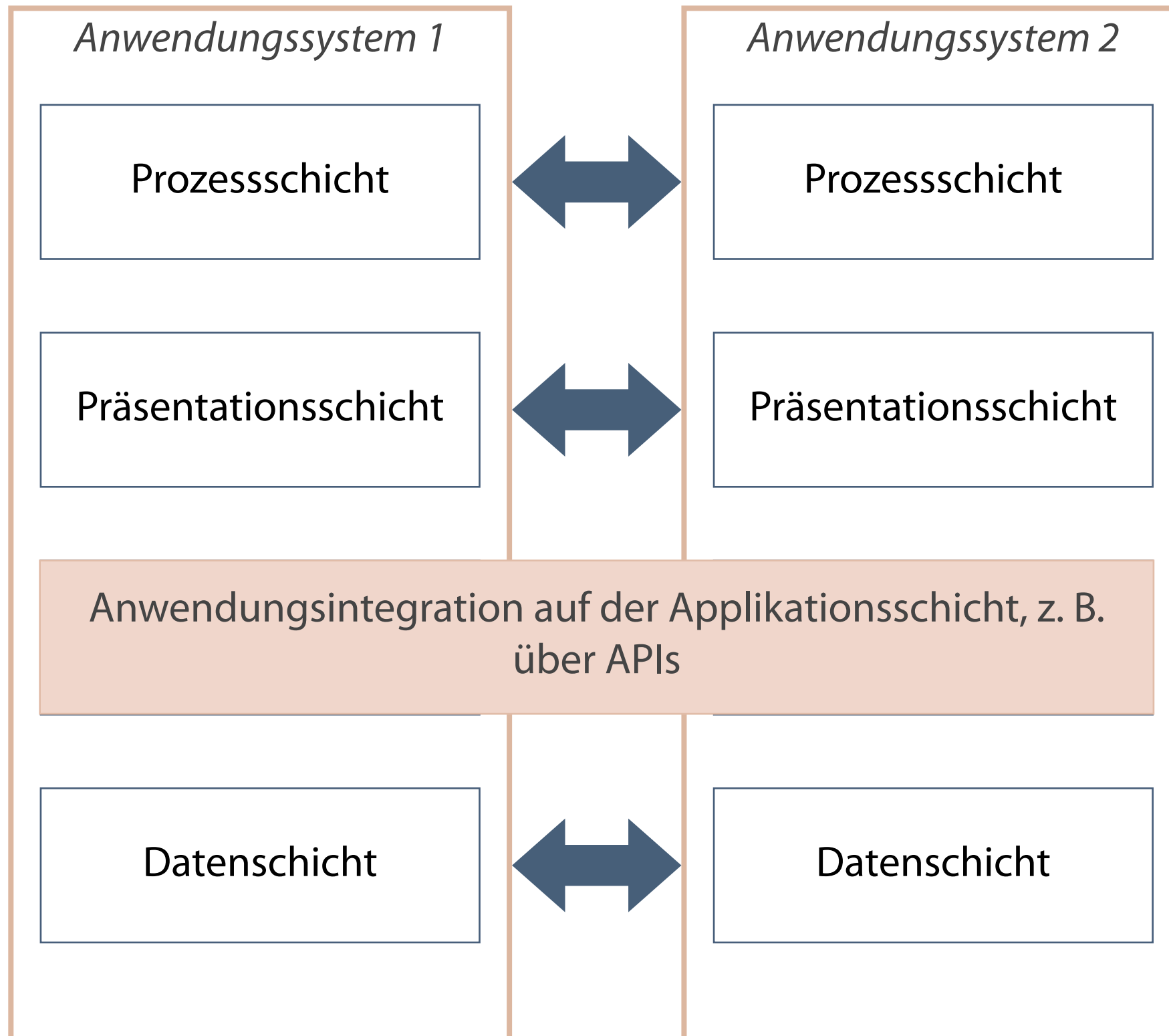
Datenintegration



Datenintegration

- Entnehmen von Daten aus einem Datenspeicher
- Überführen in einen anderen Datenspeicher
- Verständnis für Datenbanktechnologien und Fluss der Daten durch das Unternehmen
- Gefahr von Inkonsistenzen bei direktem Zugriff

Anwendungsintegration



Anwendungsintegration

- Findung von Anwendungen durch standardisierte oder proprietäre Schnittstellen
- Vorteil: Konsistenz der Daten durch Geschäftslogik
- Nachteil: API-Restriktionen

Application Programming Interfaces

- Schnittstellen zur Nutzung von Softwarefunktionalität
- Standardmethoden, wie Remote Message Invocation, CORBA, RPC
- Verbindung zu Application Server und Datenbanken

Integrationsmethoden

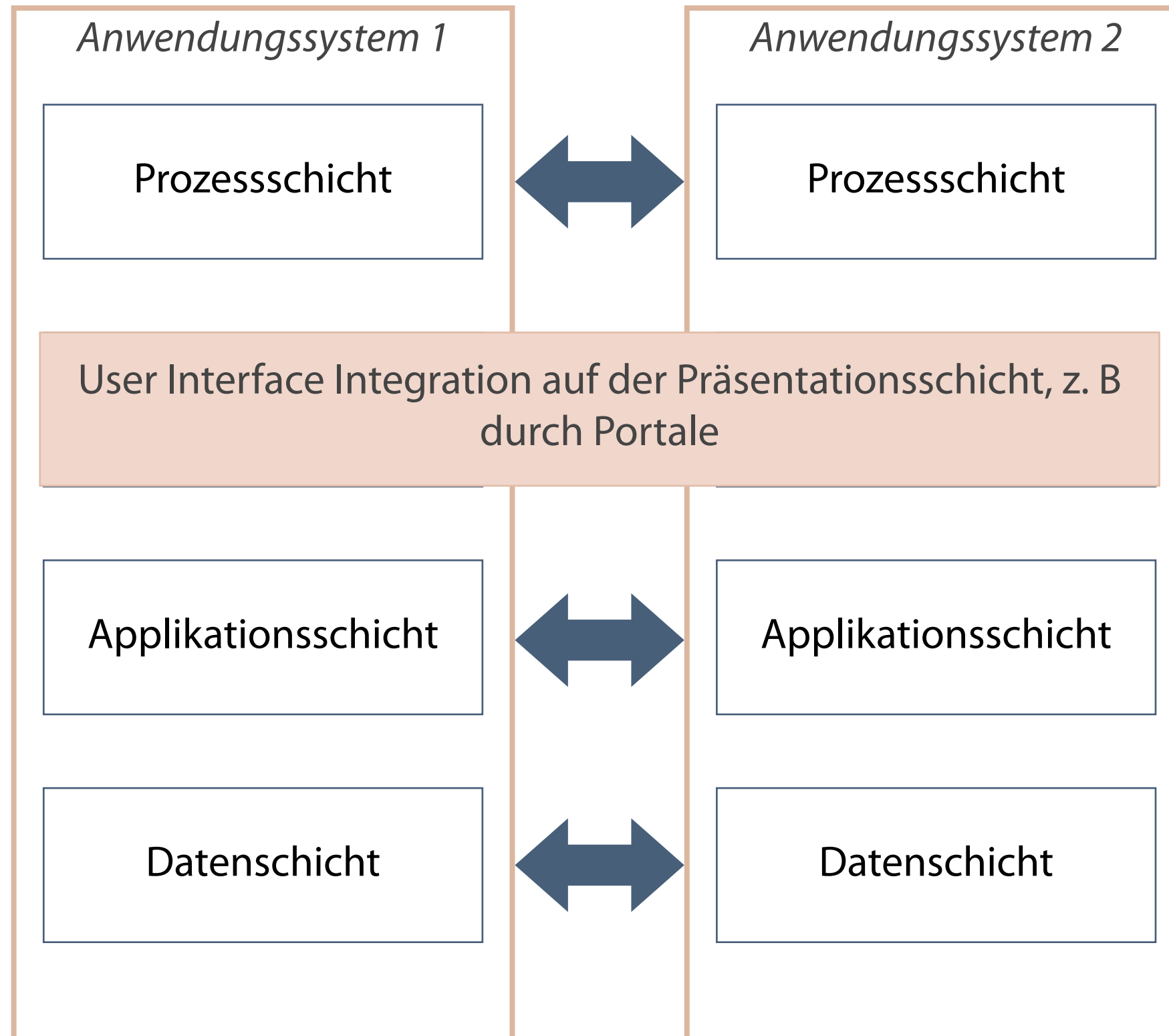
Method Integration

- Auch als Objektintegration bezeichnet
- Spezialfall der Anwendungsintegration
- Systemübergreifende Verteilung und Nutzung der Geschäftslogik
- Gegenseitiger Aufruf von Methoden unterschiedlicher Systeme
- Client/Server-Architektur: Geschäftslogik befindet sich auf gemeinsamen Server (Application Server); Aufruf von Anwendungen durch Server
- Komponentenorientierung: Anbieten von Methoden über Schnittstellen

Wrapper

- Häufige Lösung für Anwendungsintegration für Systeme ohne APIs
- Zugänglich machen von Geschäftsprozessen als Methoden in Objekten durch Umschließen des originalen Quellcodes
- Anwendung = verteiltes Objekt
- Primäre Aufgabe = Kapselung der gespeicherten Daten in einer Datenquelle und ihrer Schnittstelle nach außen
- Pro Datenquelle ein Wrapper

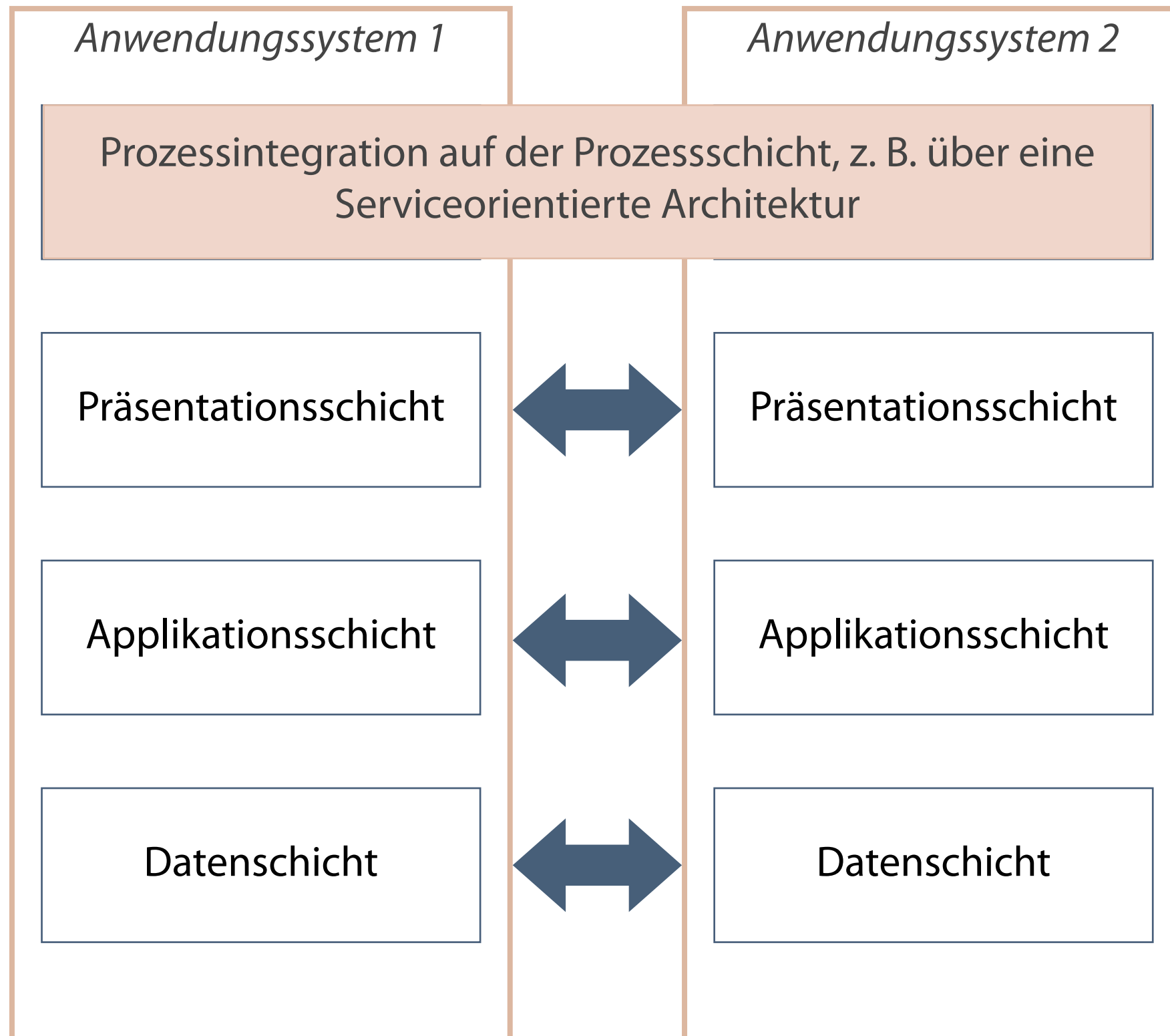
Integrationsansätze



User Interface Integration

- Zugriff auf Daten und Prozesse in Applikation über Benutzungsoberfläche
- Einsatz bei Legacy-Systemen wie Mainframe-anwendungen ohne API oder Datenbankschnittstelle
- Kein direkter Eingriff in den Quellcode
- Kenntnisse des internen Datenspeichermechanismus, der Präsentation, der Daten an der GUI und der Anwendungslogik erforderlich
- Probleme in Leistungsfähigkeit, Stabilität, Skalierbarkeit
- Programmatischer, mechanischer Zugriff einer Middleware auf die Benutzerschnittstelle durch Emulatoren

Integrationsansätze



Prozessintegration

- Höhere Abstraktionsebene als informationsorientierte Integrationsansätze
- Geschäftsprozessautomatisierung
- Gemeinsame Abbildung systemübergreifender Geschäftsprozesse
- Ganzheitliche Betrachtung von Prozessen
- Business Process Management
- Berücksichtigung aller Phasen: Entstehung, Steuerung und Kontrolle, Überprüfung und Optimierung

Überblick

- Definition der Prozesse auf einer genügend abstrakten Ebene
- Modellierungssprache mit grafischen Modellierungstools
- Trennung von Geschäftsprozessebene und darunter liegenden Integrationsschichten

Prozesssteuerung koordiniert

- Arbeitsabläufe
- Informationsflüsse
- Interaktionen mit den Anwendungen gemäß dem Prozessmodell

Prozessmodellierung definiert

- Prozesse
- Zugriffe auf Anwendungen und Ressourcen
- Abläufe
- Informationsflüsse

Prozesskontrolle

- Prozessmonitoring
- Prozessoptimierung



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

Integrationstechnologien

Middleware

Enterprise Application Integration

Beispiele von Integrationen



Middleware

- Middleware als grundlegende Technologie
- Middleware = Software auf einer Zwischenschicht; ermöglicht die Kommunikation mehrerer Systeme
- Middleware-Technologie auf jeder Ebene des Schichtenmodells
- Hauptvertreter sind Application Server und Message Broker

Zentraler Bestandteil einer Integrationsinfrastruktur ist eine Middlewarekomponente.

Beispiel

ERP-CRM-Kopplung



„Nullvariante“



Variante 1



Variante 2

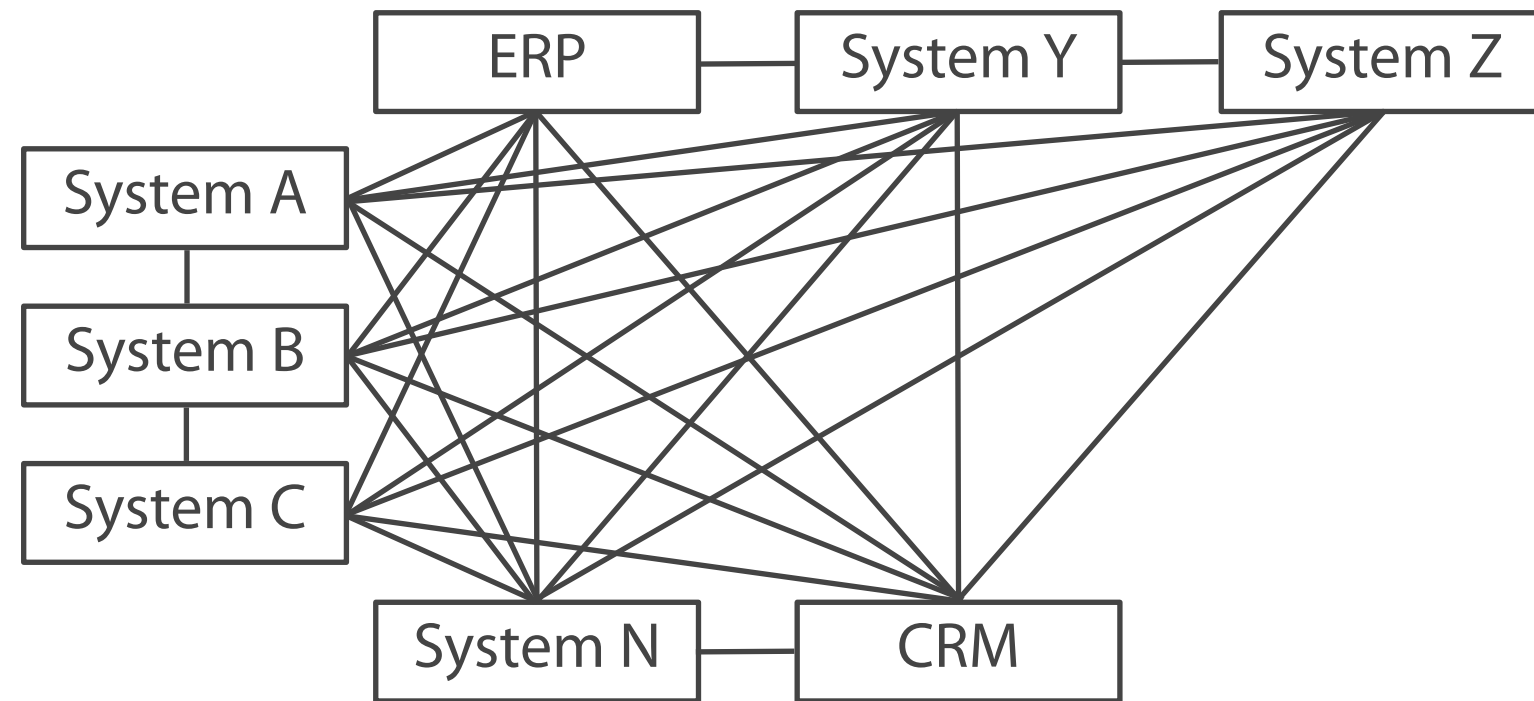


Variante 3



Die Umsetzung dieser Funktionen benötigt die Integration verschiedener Anwendungs- und Backendsysteme.

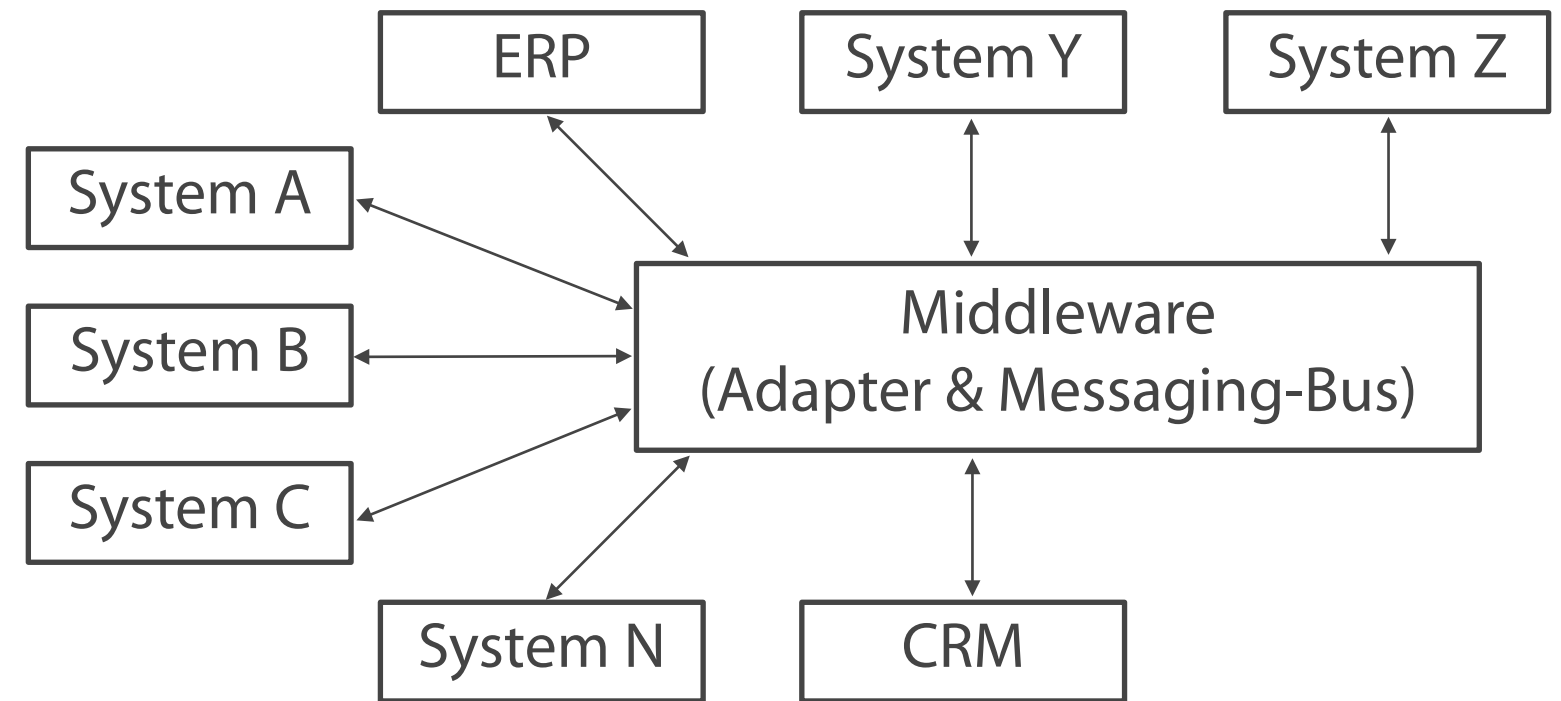
Integrationsformen



Point to Point Integration

- Jedes System ist direkt mit jedem notwendigen System verbunden
- Individuelle Anpassung der Schnittstellen
- Dezentraler Aufbau der Systemlandschaft
- Feste Kopplung

Eine weitere Integration kann durch eine durchgängige Service-orientierte Architektur erfolgen.



Zentrale Integrationsplattformen

- Bereitstellung einer zentralen Plattform
- Zum Teil mit vorgefertigten Adaptern

Beispiel

ERP-CRM-Kopplung

ERP-Systeme mit CRM	Autonome CRM-Systeme	Front Office CRM
<ul style="list-style-type: none">• optimale ERP-Integration• Prozesse im Mittelpunkt• hohe Kosten	<ul style="list-style-type: none">• größter CRM-Umfang• hoher Aufwand für Integration• hohe Kosten	<ul style="list-style-type: none">• hohe Nutzerakzeptanz• hoher Aufwand für Integration• geringe Kosten
SAP, Oracle	Siebel (Oracle), Applix, Point, Clarify (Amdocs), Microsoft Dynamics, Cobra CRM, Salesforce, Update CRM, KANA, eGain	Team 4, Act!, Sugar CRM

Die Integration der CRM-Aufgaben kann in unterschiedlichem Umfang im ERP-System erfolgen.

Message Broker

- Nachrichtenvermittler zwischen zwei oder mehreren Systemen
- Verwaltung und Überwachung der Schnittstellen
- Eine Schnittstelle pro System

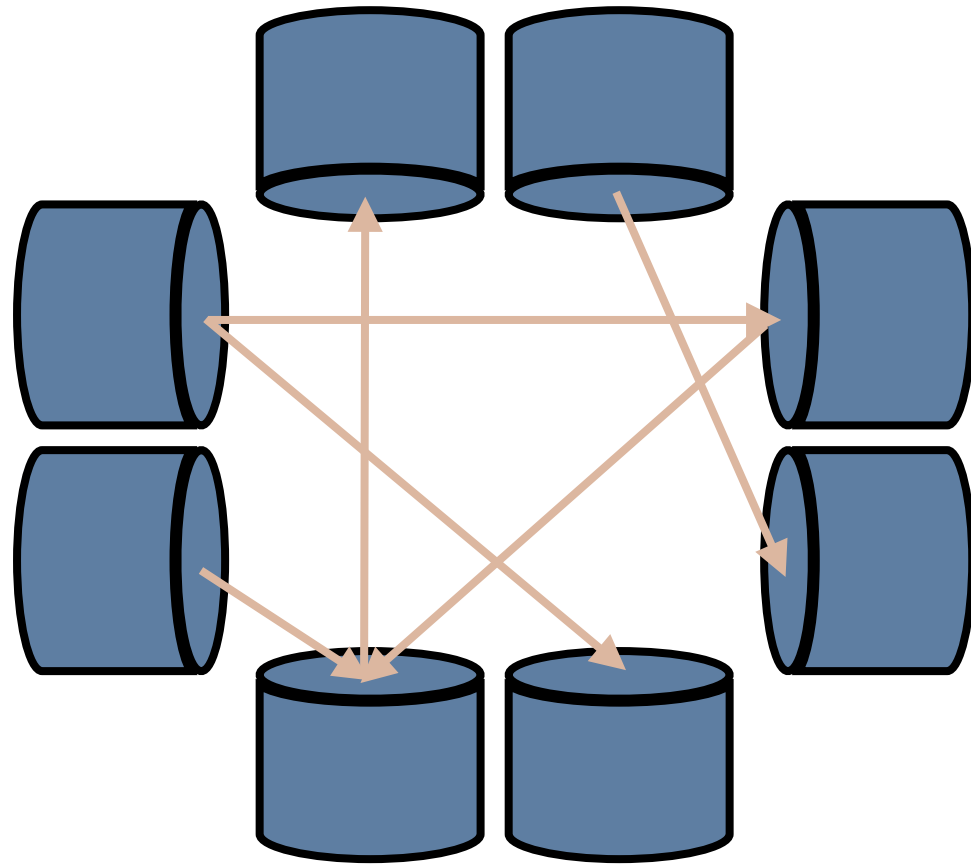
Aufgabe des Message Brokers

- Weiterleitung der Datenpakete an die jeweilige Anwendung
- Bei Bedarf Umwandeln der Daten in geforderte Datenformate

Adapter bzw. Konnektor

- Ausgleichen und Verbergen der Unterschiede verschiedener Schnittstellen
- Verwaltung der Beziehung zwischen Anwendungen

Prinzipien von Integrationsarchitekturen



SOA

- Dezentraler Aufbau der Systemlandschaft
- Standardisierte und wiederverwendbare Schnittstellen
- Lose Kopplung von Systemen



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

Integrationstechnologien

Middleware

Enterprise Application Integration

Beispiele von Integrationen

Entwicklung von Middleware (1/2)

Middleware der 1. Generation

- Ziel = Reduzierung des Integrationsaufwands, durch Minimierung der zu implementierenden Schnittstellen
- Rein technische Anbindung, ohne Berücksichtigung semantischer Aspekte
- I.d.R. auf unternehmensinternen Einsatz beschränkt

Typologie der 1. Generation

- Database Middleware (Database Connectivity)
- Remote Procedure Calls (RPC)
- Object Request Broker
- Transaktionsmonitor
- Nachrichtenorientierte Middleware

Die Middleware der 1. Generation zielen allein auf die Minimierung der Schnittstellen.

Entwicklung von Middleware (2/2)

Middleware der 2. Generation

- Blick auf semantische Integration
- Verfahren zur (automatischen) Transformation der Informationen
- I.d.R. Laufzeitumgebung

Application Server

- Zentrale Plattform
- Zugriff auf grundlegende Dienste (z.B. Middleware-Funktionalität der 1. Generation)
- Zugänglich machen von Daten und Diensten für weitere Systeme
- Skalierbar, meist durch Cluster-Lösungen

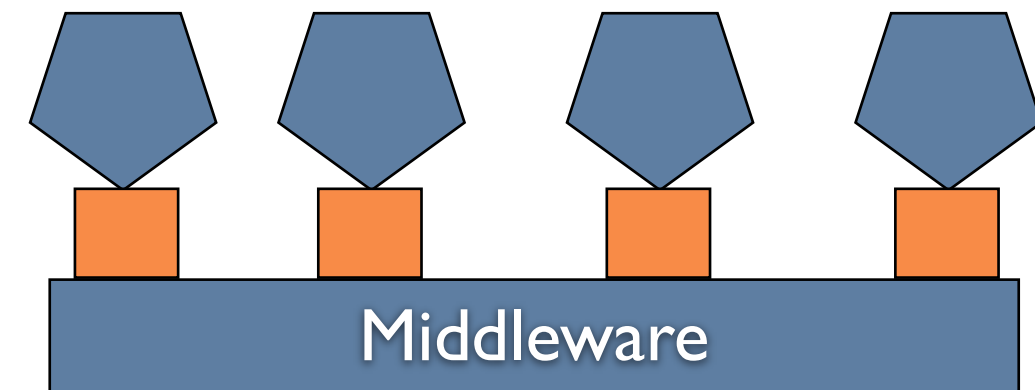
Application Server = Integrationsplattform, wenn

- Datenkonvertierungsdienste angeboten werden
- Verbindung von Komponenten unterschiedlicher Herkunft möglich
- Integration von Daten unterschiedlicher Quellen möglich

Die Middlewaretechnologien der 2. Generation ermöglichen eine Integration entlang der Wertschöpfungskette.

Kernfunktionen von Middleware

- Anbieten einer Reihe von einheitlicher Dienste und Schnittstellen
- Sicherstellen konsistenter Datenflüsse ohne Netzwerkstruktur abzubilden
- Schafft homogene Struktur für heterogene Anwendungen
- Funktion der verteilten Anwendung muss unabhängig von Betriebssystem, Netzwerkprotokoll und Ort verfügbar sein
- Sicherung der Interoperabilität zwischen unterschiedlichen Applikationen
- Schnittstellen zu Anwendungen ermöglichen Gestaltung der Gesamtlösung aus wiederverwendbaren, anpassbaren und portierbaren Bausteinen (Einzelanwendungen)



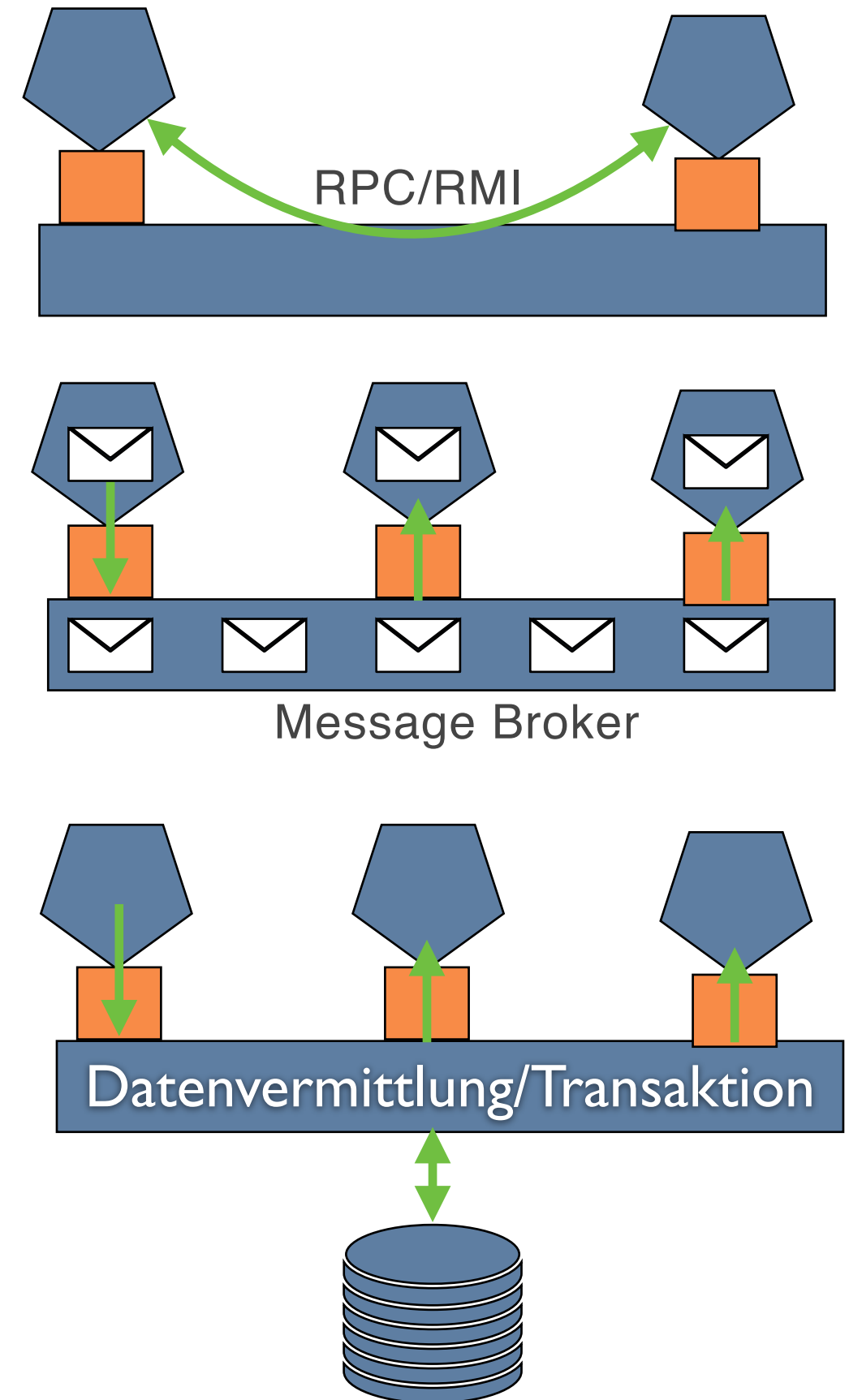
Middlewareparadigmen

Kommunikation

Daten

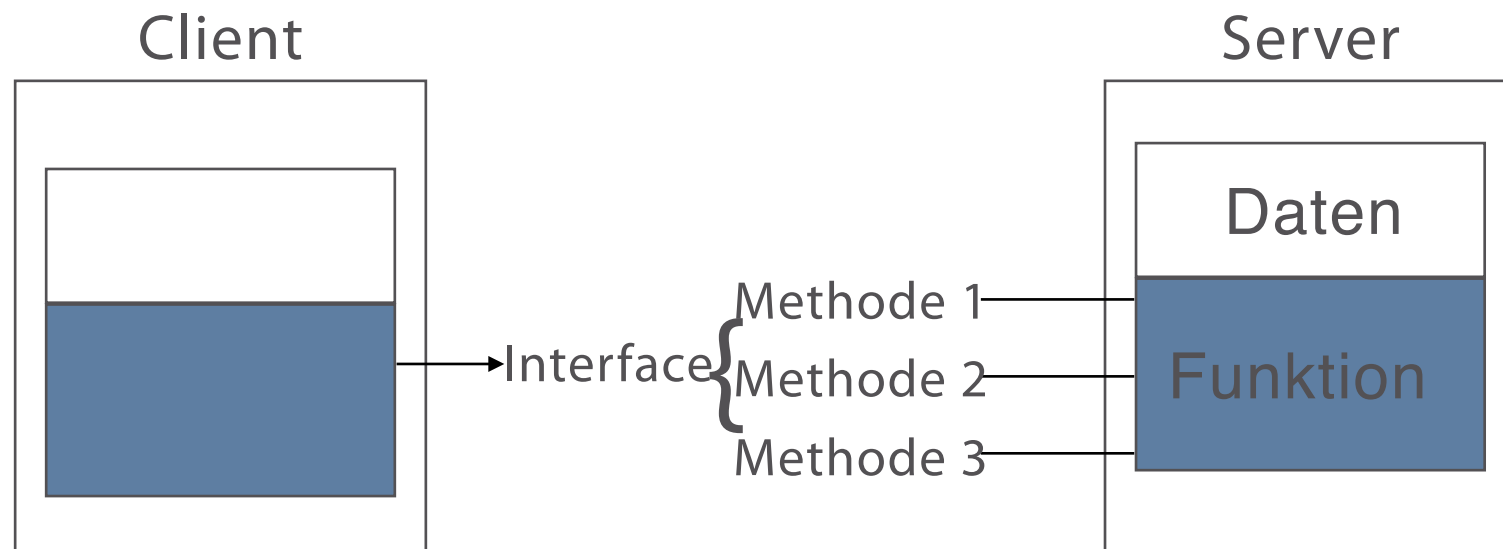
Anwendungen

- Funktionsaufruforientiert z.B. über RPC oder RMI: Client ruft Funktion auf Serverkomponente auf
- Nachrichtenorientiert (Message Broker): Lose Kopplung von Applikationen über den Austausch von Nachrichten (Publish/Subscribe)
- Datenhaltungsorientiert: Anfragevermittlung an verteilte Datenbanken
- Transaktionsorientiert: Koordiniert systemweit den Zugriff auf Ressourcen (primär Datenbanken und Dateien) über das Prinzip der Transaktion
- Objektorientiert (Object Request Broker): verteilte Objekte, die netzwerk- und teilweise programmier-sprachenübergreifend Anwendungen verbinden
- Komponenten-orientiert: Verbindung von am Markt verfügbaren Anwendungen (Business Components)



Funktionsaufruforientierte Middleware

Remote Procedure Call (RPC)



Merkmale

- Prozeduren anderer Rechnern ist in der gleichen Art und Weise wie eine lokale Prozedur aufzurufen

Funktionsweise

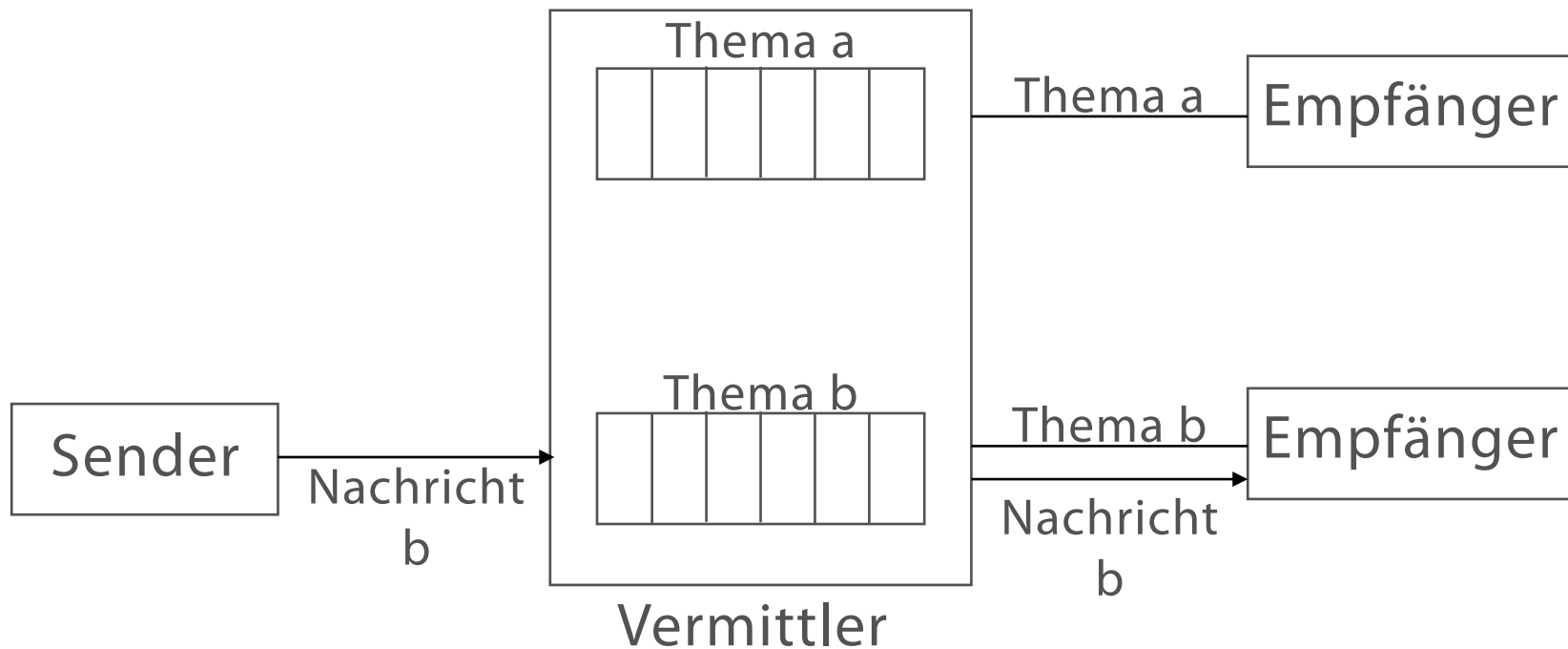
- RPCs ermöglichen einen entfernten Aufruf eines Clients auf einem Server für eine Prozedur
- Jeder Server im Netz stellt eine Anzahl von Diensten bereit, die mit einer RPC-Prozedur angefordert werden können

Vorteile

- Ermöglichung der Aufgabenverteilung

Nachrichtenorientierte Middleware

Publish-Subscribe



Merkmale

- Lose Kopplung der Komponenten
- Komponenten sind verteilt und haben Status

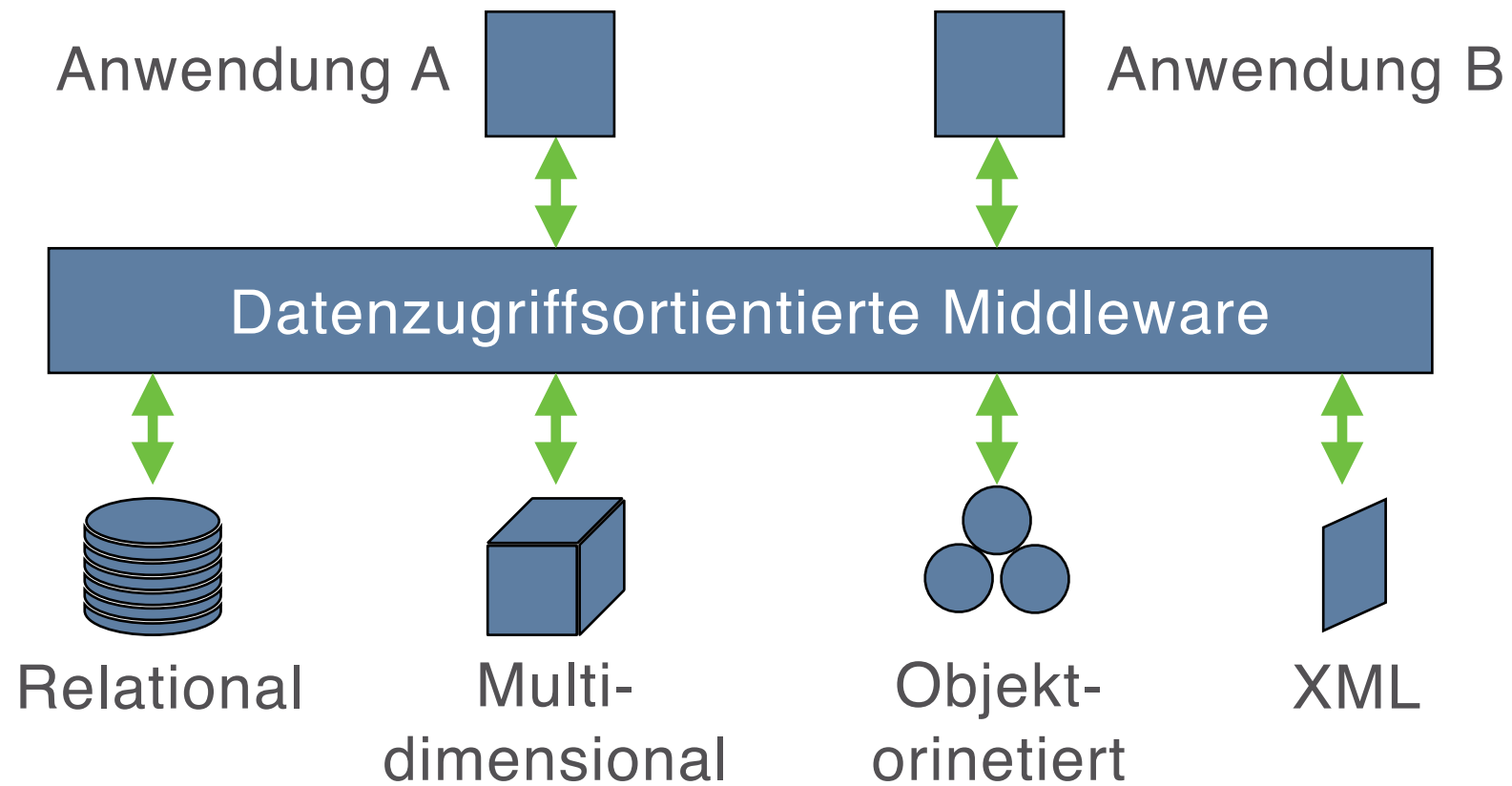
Funktionsweise

- Asynchrone Kommunikation zwischen Client und Server mittels Nachrichten in Form von Warteschlangen
- Sender stellt Nachricht in die Warteschlange, die durch Vermittler verteilt wird
- Empfänger holt die Nachricht zu einem beliebigen Zeitpunkt ab

Vorteile

- Steigerung der Performance durch Bündelung der Nachrichten
- Realisierung unabhängig der Plattform sowie Programmiersprache

Datenhaltungsorientierte Middleware



Merkmale

- Bereitstellung von Schnittstellen für Anwendungen
- Abbildung der Datensätze in das benötigte Format

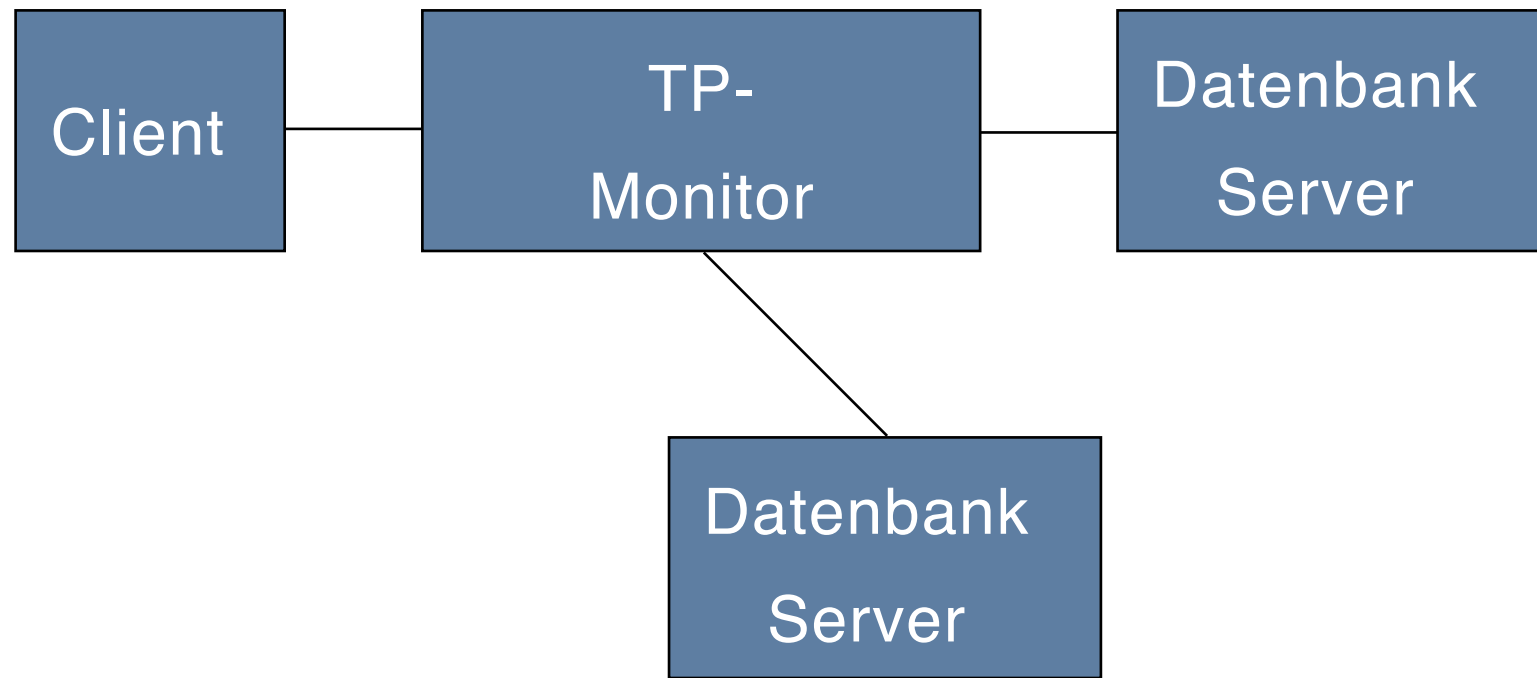
Funktionsweise

- Eine vermittelnde Schicht abstrahiert Daten verschiedener Datenbanken und stellt diese Anwendungen zur Verfügung

Vorteile

- Gemeinsames Datenmodell wird nicht vorausgesetzt

Transaktionsorientierte Middleware



Merkmale

- Konsistenz der Daten nach ACID
 - Atomarität
 - Konsistenz
 - Isolierheit
 - Dauerhaftigkeit

Funktionsweise

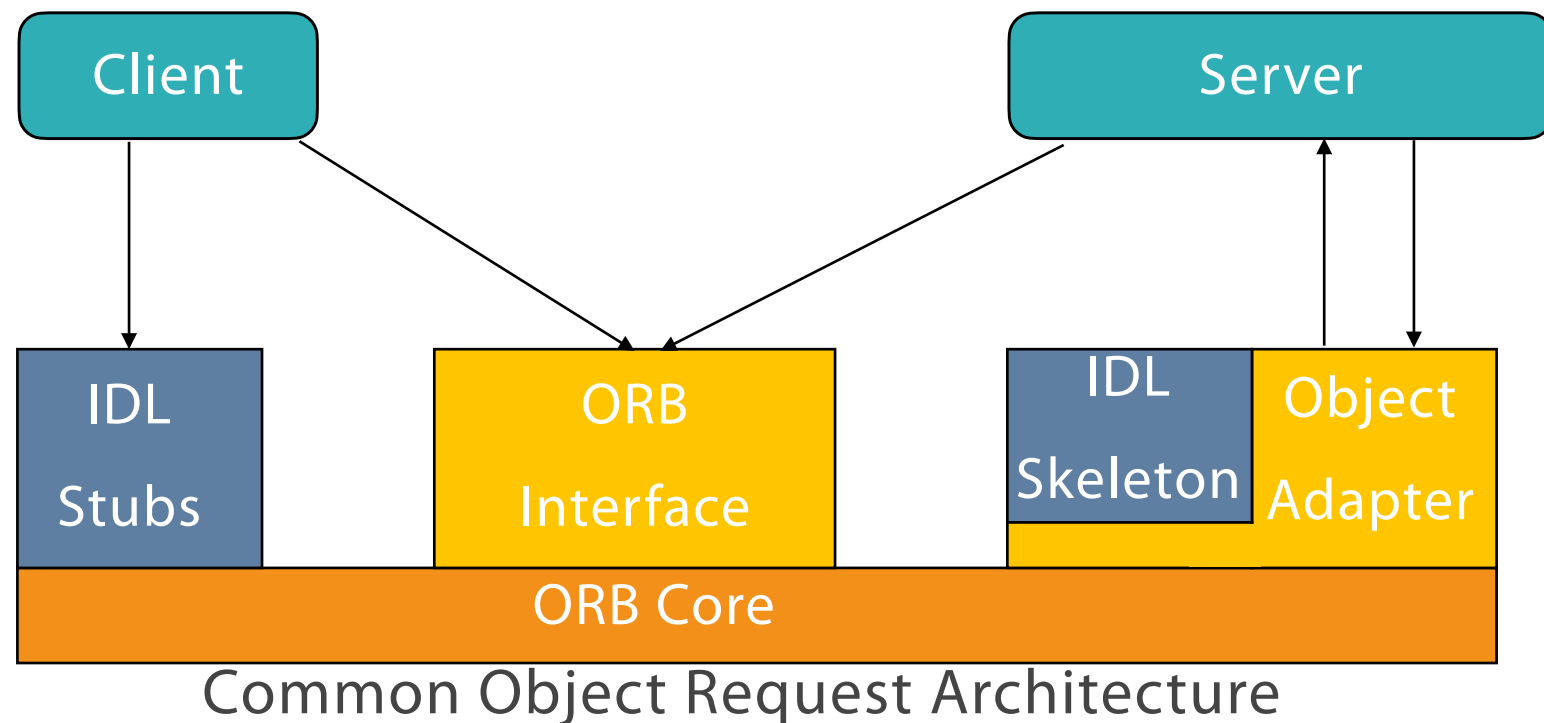
- Abwicklung und Ausführung von Transaktionen durch Stapelverarbeitung z.B. eines Client-Server Systems für Flugbuchungen

Vorteile

- Portabilität der Anwendungen

Objektorientierte Middleware

Object-Request Broker (ORB)



Merkmale

- Methoden-Anfragen der Klienten werden an die dienstbringenden Objekte (Server) und den Klienten zur Verfügung gestellt

Funktionsweise

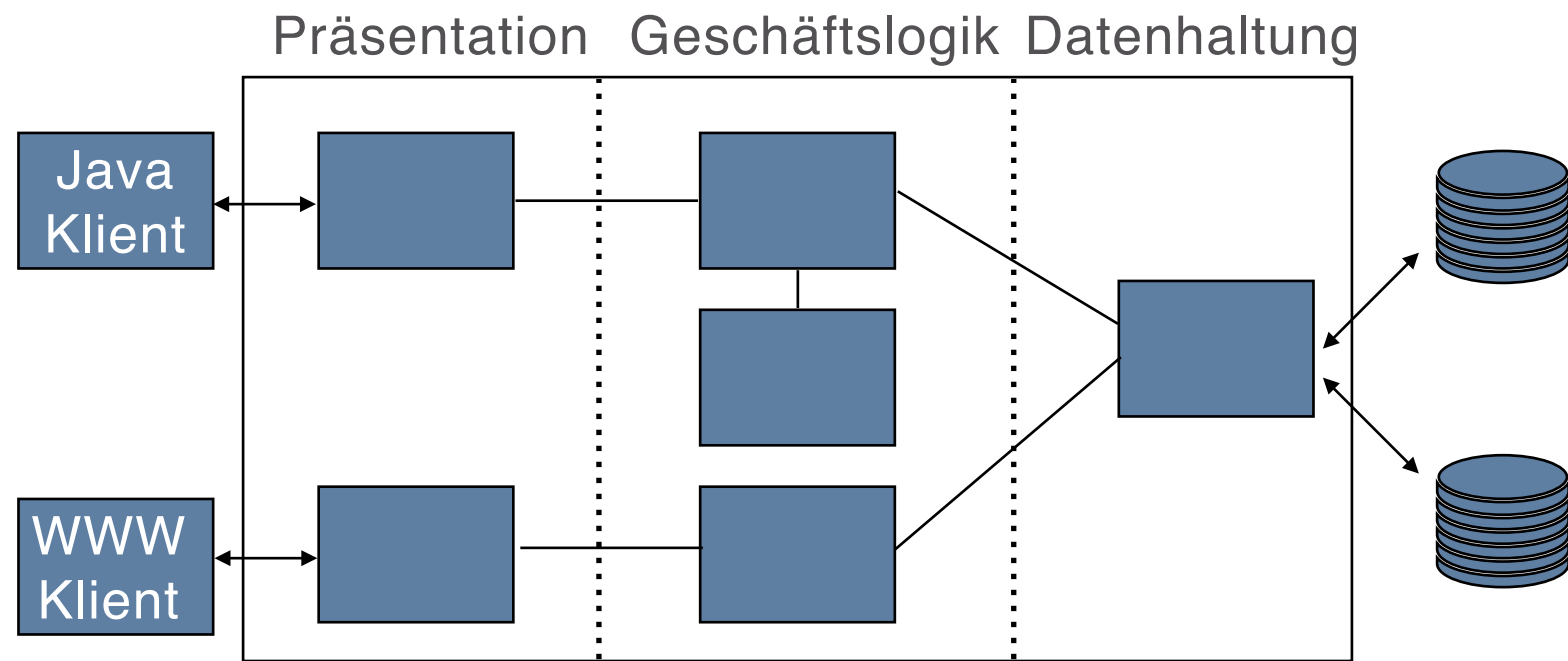
- Verteilte Objekte auf unterschiedlichen Adressräumen werden mit Hilfe des ORBs als Vermittlerrolle zusammengeführt

Vorteile

- Effizientes Zusammenarbeiten verteilter Systeme durch Interoperabilität und Portabilität

Objektorientierte Middleware

Object-Request Broker (ORB)



Merkmale

- Lose Kopplung
- Verteilte Komponenten
- Kombinieren und Konfigurieren statt Programmieren

Funktionsweise

- Anhand eines Komponentenmodells als Architekturschema können aus einzelnen Software-Bausteinen komplexe Anwendungen zusammengebaut werden

Vorteile

- Wiederverwendung von Komponenten
- Kombination von neuen Komponenten aus bereits existierenden

Arten von Middleware

Bereitstellung häufiger Funktionen

- Abstraktion der Netzwerkprogrammierung
- Aufgaben: Kommunikation, Datentransformation (einheitliches Format), Verbergen von Heterogenität
- Vorteil: alle beteiligten Komponenten können Daten gleich interpretieren und standardisiert kommunizieren
- Beispiele: CORBA RMI, Java RMI, JMS
- Synchron: bei hoher Interaktion zwischen Anwendungen, z.B. über Remote Procedure Calls oder Remote Method Invocation
- Asynchron: bei loser Kopplung der Systeme, z.B. über nachrichtenorientierte Middleware (Message Queuing)

Integration und Automatisierung

- Verbindung von externen und internen Systemen durch einheitliche Bereitstellung von Datentransformationen, einheitlicher Konnektivität, Kombinierbarkeit, Authentifizierung und Messaging
- Datenstreaming: Replikation von Datensätzen in gemeinsamen Zwischenspeicher und asynchrone Übertragung an die angeschlossenen Applikationen (z.B. über Apache Kafka)
- Teilweise Geschäftsprozessautomatisierung: Implementierung von Business Rules und Vorverarbeitung eingehender Daten (Konsistenzprüfung, Umformung, etc.)



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

Integrationstechnologien

Middleware

Enterprise Application Integration

Beispiele von Integrationen

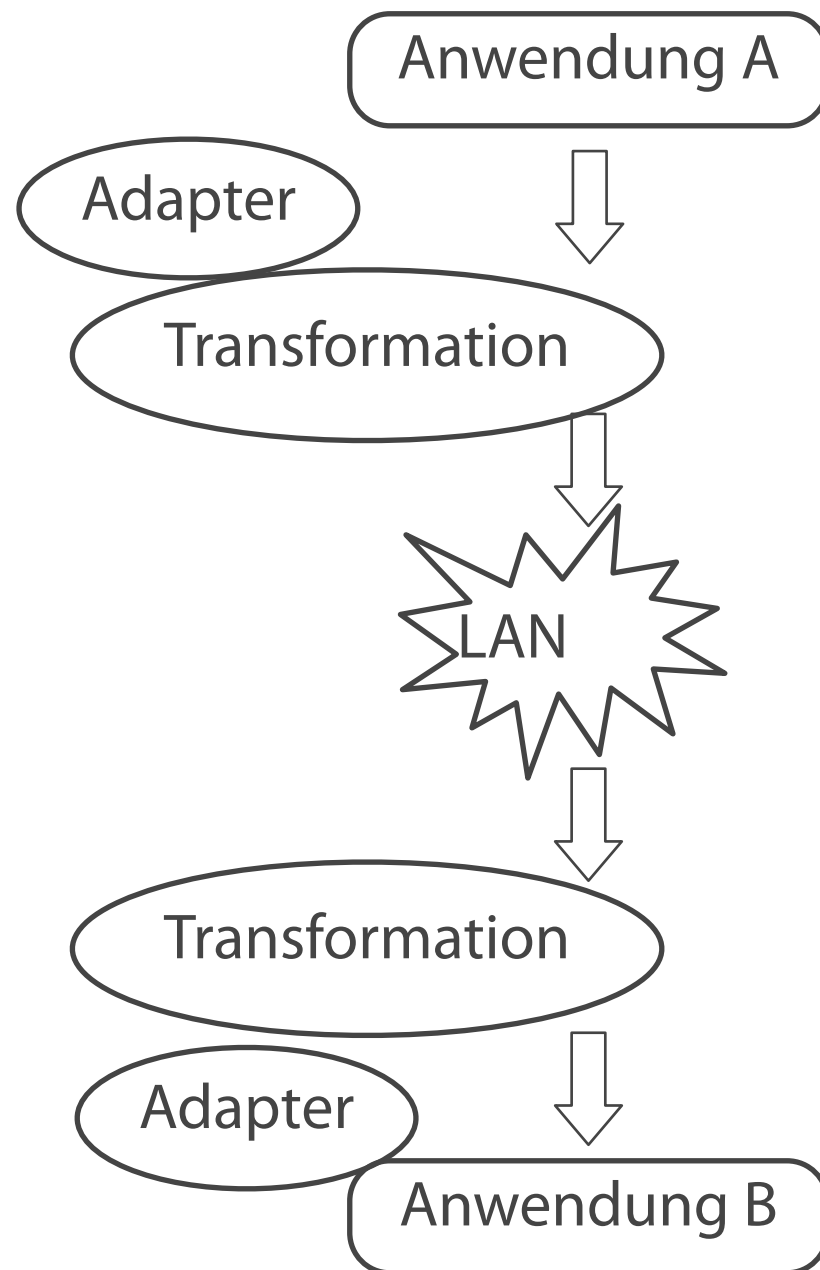
Historische Entwicklung

Überblick

- Entstehung komplexer, sich überschneidender Applikationen durch erst zentrale, spätere dezentrale Systementwicklung
- Erkennung der Relevanz des Themas der Applikationsintegration Mitte der 1980er Jahre durch große Unternehmen
- Notwendigkeit der Integration durch Aufkommen der ERP-Systeme in den 1990er Jahren
- Nicht nur Datenintegration, sondern systemübergreifend entlang des Geschäftsprozesses

Die Anwendungen kommunizieren über das firmeneigene Netzwerk miteinander.

Enterprise Application Integration



Definition

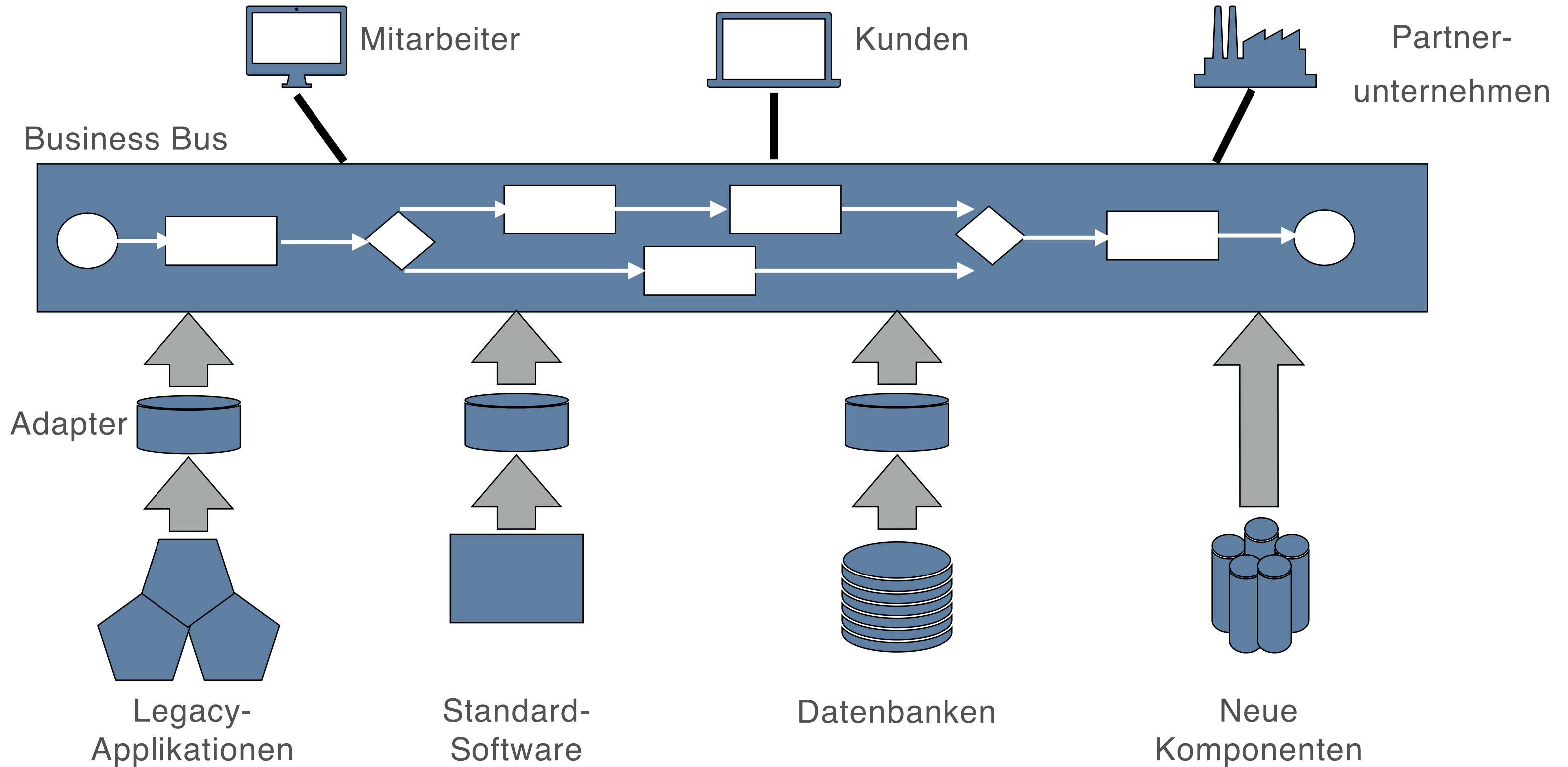
- Management bestehender Strukturen, Prozesse und IT-Systeme
- Zusammenführung heterogener Daten, Anwendungen und Prozesse innerhalb von Unternehmensgrenzen
- Zusammenarbeit von ursprünglich unabhängig voneinander entwickelten und funktionierenden Systemen
- Gewährleistung des Informationsaustauschs zwischen den voneinander unabhängig entwickelten Systemen

Umfang

- Das Bereitstellen von Adaptern bzw. Konnektoren
- Regelbasiertes Routing
- Transformation und Übersetzung der Daten

Die Anwendungen kommunizieren über das firmeneigene Netzwerk miteinander.

Schematische Darstellung EAI



Enterprise Service Bus

- Dienst = fachlich oder technisch zusammengehörige Teilmenge von Geschäftsfunktionen
- Logische und physische Kopplung von Diensten zur Gesamtarchitektur der Anwendungslandschaft
- Bus überträgt Daten in einem standardisierten Format
- Ersetzt direkte physische Kopplung (Peer-to-Peer-Architektur) mit eigenen Layer, der Dienste von Dienst Anbietern für die Dienstanutzer zur Verfügung stellt
- Nachrichten werden über Dienste zu Endpunkten (Applikationen) gesendet

Adapter

- erlauben wiederverwendbare Überbrückung zwischen Endpunkten und Middleware

Integrationsdienst

- Transformation: Überbrücken von unterschiedlichen Datenformaten und -modellen
- Routing: Weiterleitung von Nachrichten nach bestimmten Regeln
- Orchestrierung: Steuerung des Nachrichtenflusses nach Prozessmodell

Kollaborationsprotokolle (Beispiele)

ebXML (OASIS)

- XML für elektronische Geschäftsprozesse
- Standardfamilie (ISO 15000) beinhaltet Architekturspezifikation, Geschäftsprozessspezifikation, Registrierdienst, Informationsmodell und Nachrichtendienst
- Herstellerunabhängig
- Branchenübergreifender Einsatz
- Implementiert in einer Reihe von kommerziellen Produkten

BizTalk (Microsoft)

- Adaptoren für Standardsoftware
- Engine zur Modellierung von Geschäftsprozessen
- Aktivitätsmonitoring
- EDI-Funktionalität (X12, EDIFACT)
- Unterstützung von Webservices
- Integration in MS-Produkte

Operative Nutzenpotenziale durch EAI

Nutzen	Nutzenpotential in Geschäftsprozessen	Nutzenpotential in IT-Abteilungen
Kosten	<ul style="list-style-type: none">■ Daten werden einmal eingegeben■ Papierverbrauch wird reduziert■ Geringe Übertragungskosten	<ul style="list-style-type: none">■ Wiederverwendung von Komponenten■ Point to Point Schnittstellen verringern■ Keine Veränderungen von Altsystemen
Zeit	<ul style="list-style-type: none">■ Beschleunigung von Abläufen■ Verringerung von Medienbrüchen■ Onlineverfügbarkeit von Daten	<ul style="list-style-type: none">■ Modifikation durch ex-post Integration von Komponenten■ Administrative Vorgänge werden beschleunigt
Qualität	<ul style="list-style-type: none">■ Verringerung von Redundanz durch Mehrfacheingabe■ Erhöhung der Datenqualität	<ul style="list-style-type: none">■ Validierte Komponenten von Altsystemen können genutzt werden■ Durch Prozessredesign werden Fehler aufgedeckt

Weitere Nutzenpotentiale durch EAI

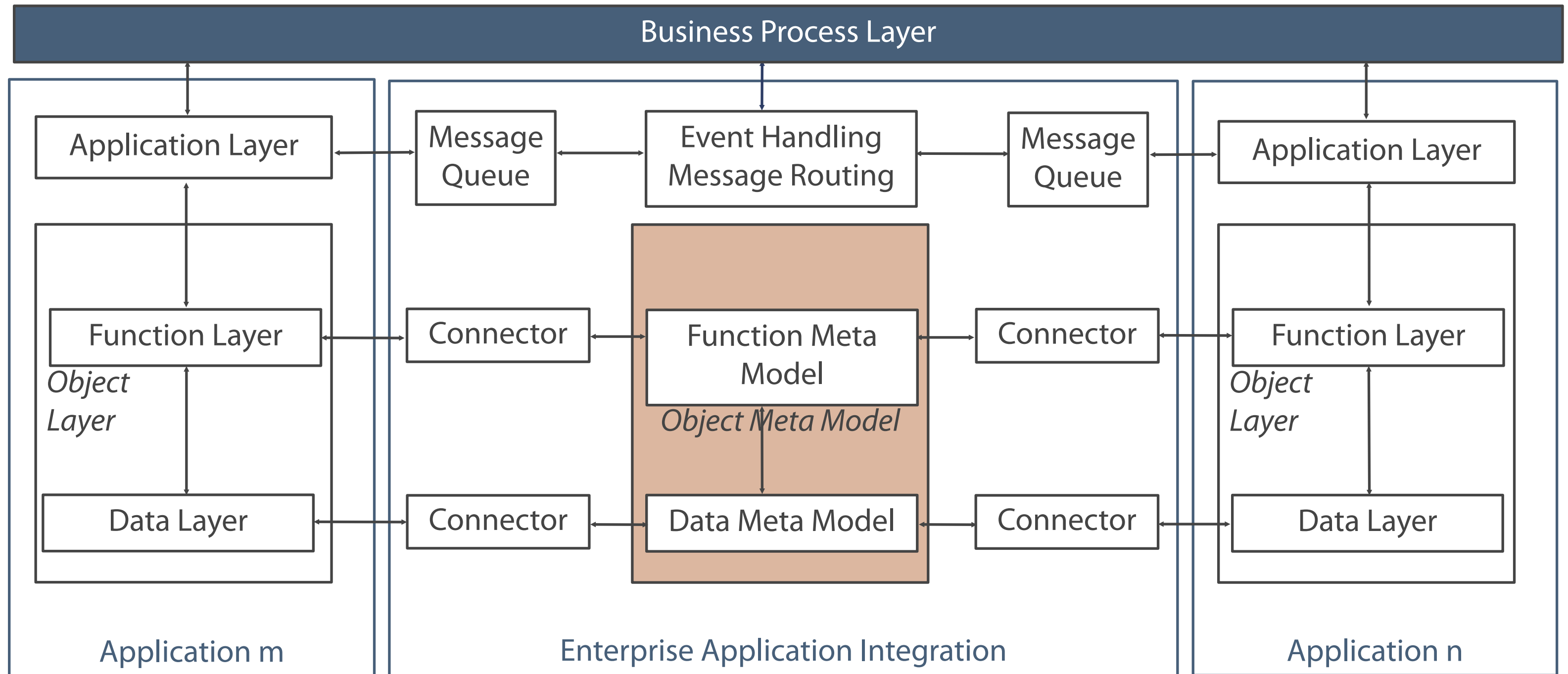
Strategische Nutzenpotentiale der EAI

- Steigerung der Integrationsfähigkeit für künftige Anwendungen und Prozesse
- Kosteneinsparungen in Merger&Acquisition-Projekten
- Schnellere Reaktion auf Marktveränderungen
- Standardisierung im IT- und Prozessbereich
- Investitionsschutz (IT-Investitionen)

Anforderungen an EAI-Lösungen

- Betrachtung von Individual- und Standardsoftware
- Prozessorientierter Austausch betriebswirtschaftlich relevanter Daten
- Keine Veränderung der Syntax oder Semantik der Daten in operativen Systemen
- Betrachtung der Geschäftsprozesse bildet Voraussetzung für EAI

Vier-Ebenen EAI-Architekturmodell



Datentransformation

Überblick

- Übermittlung und Transformation von Daten
- Inhaltliche Anwendungsdaten oder Daten zur Anwendungs- oder Prozesssteuerung
- Transformation = Überführung eines Datensatzes von einem Format in ein anderes
- Transformationsregeln zur Durchführung der Transformation

	Local Match	S1	S2 elements(s)	Matching expression
1.	1:1, Element-Level	Price	Amount	Amount = Price
2.	n:1, Element-Level	Price, Tax	Cost	Cost = Price*(1+Tax/100)
3.	1:n, Element-Level	Name	FirstName, LastName	FirstName, LastName = Extract(Name,...)
4.	n:1, Structur-Level (n:m, Element-Level)	B.Title B.PuNo P.PuNo P.Name	A.Book, A.Publisher	A.Book, A.Publisher = Select B.Titel, P.Name From B,P Where B.PuNo=P.PuNo



Integration - Eine Begriffsbestimmung

Integrationsansätze zwischen Anwendungssystemen

Integrationstechnologien

Middleware

Enterprise Application Integration

Beispiele von Integrationen

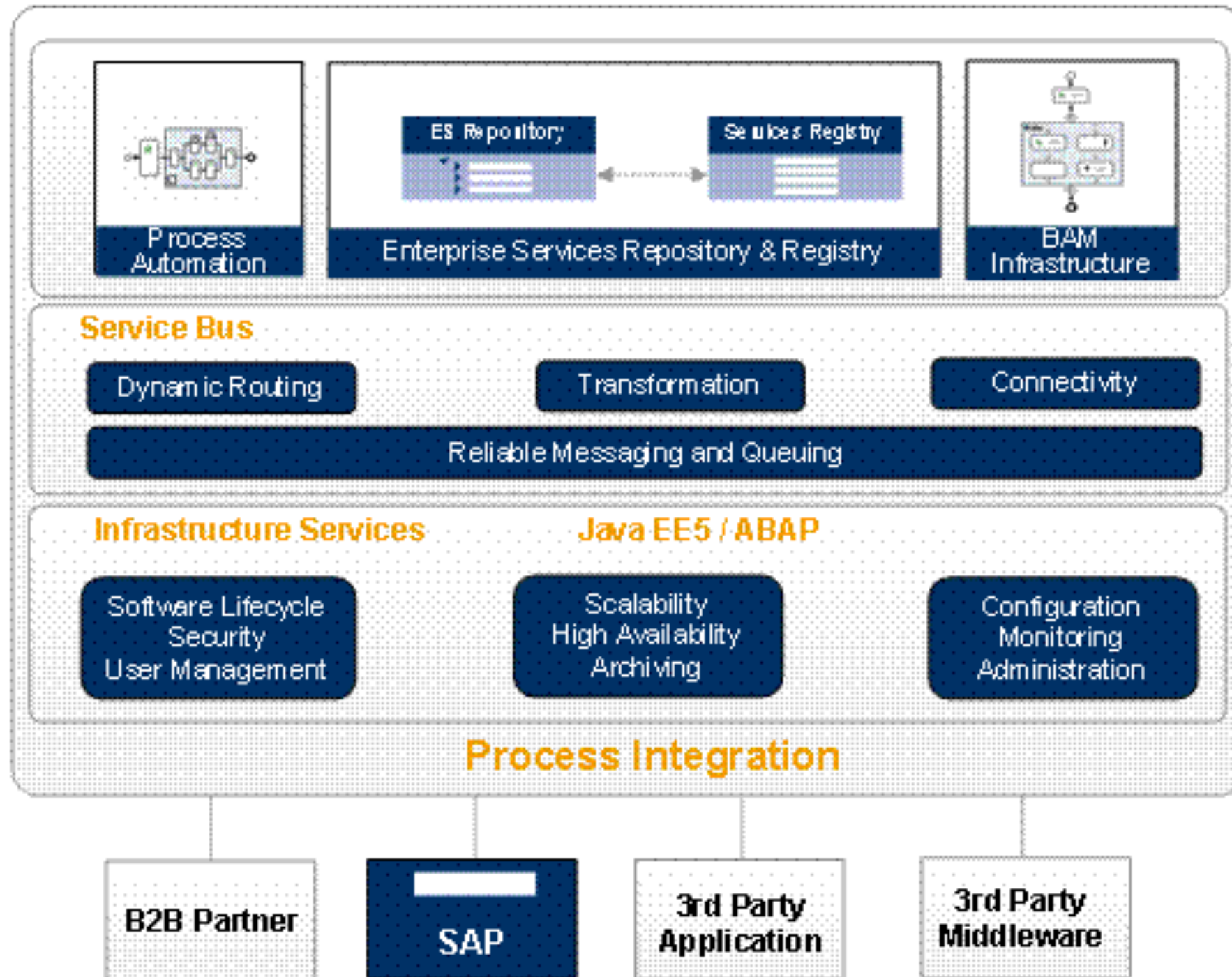
Integrationskomponenten von ERP-Anbietern

- SAP PI / SAP PO
- Infor Intelligent Open Network

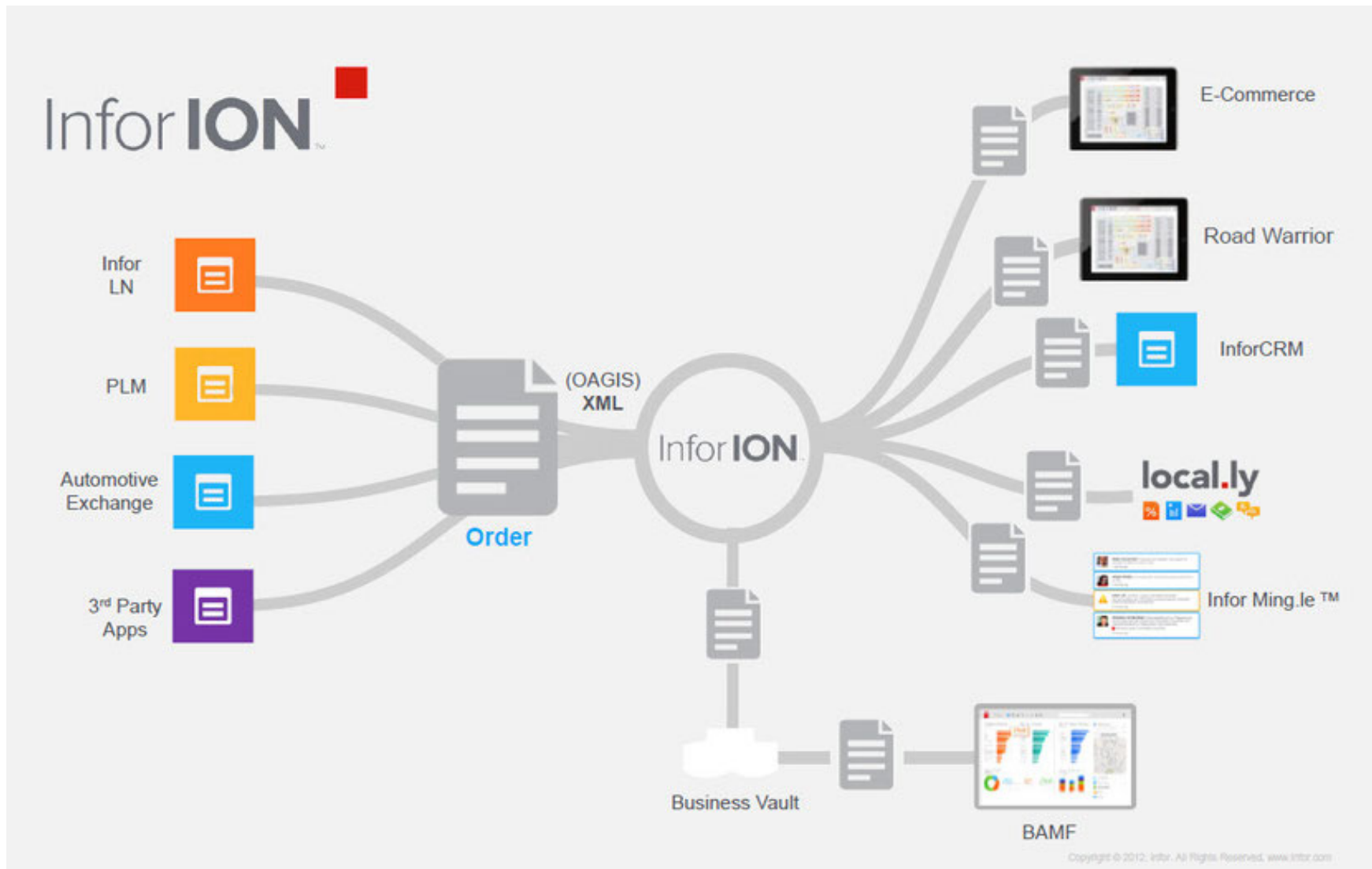
First tier

- PSI Penta PSibus ESP
- Comarch
- Applus Business Integation Framework
- proAlpha Integration Workbench

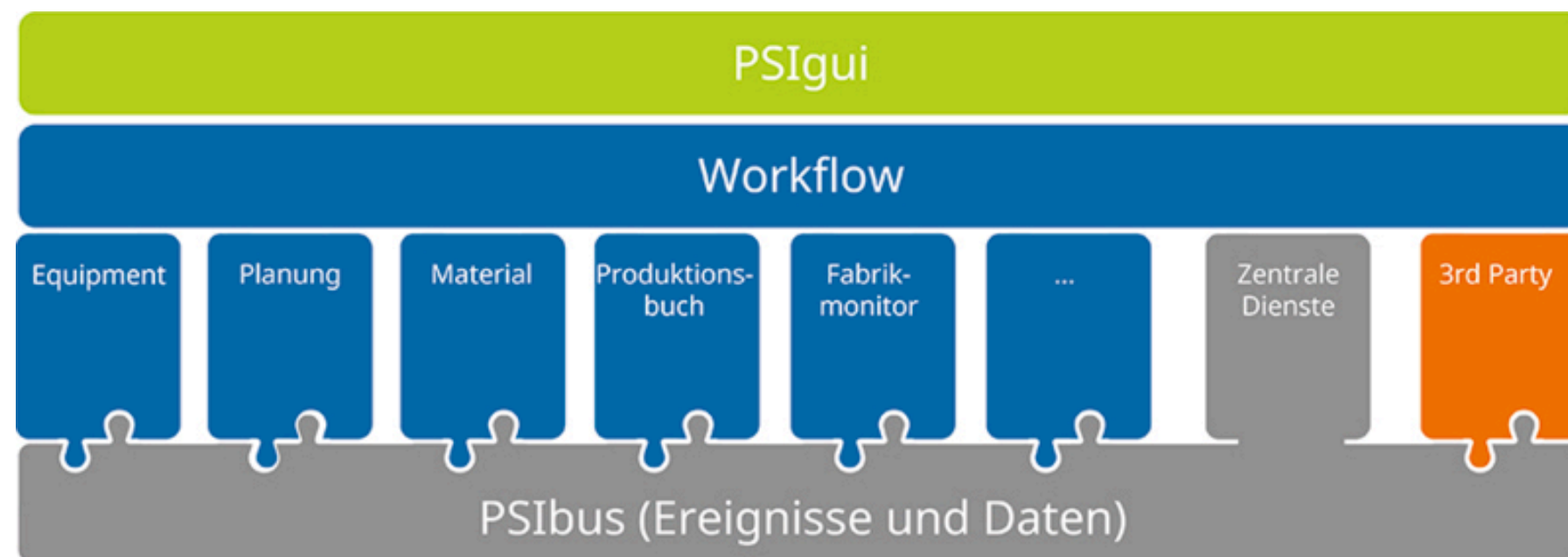
Second tier



- Bestandteil der SAP NetWeaver Middleware-Komponente (EAI-Plattform)
- Serviceorientierten Architektur für Geschäftsanwendungen (Enterprise SOA)
- Orientierung an allgemein anerkannten Standards
- XML-basierte Kommunikation über HTTP
- Übertragung anwendungsspezifischer Inhalte über Messages in einem frei definierbaren XML-Schema
- Über Integrationsserver oder direkt zwischen den beteiligten Systemen
- SAP Process Orchestration (PO) vereint Business Process Management, Business Rule Management und PI

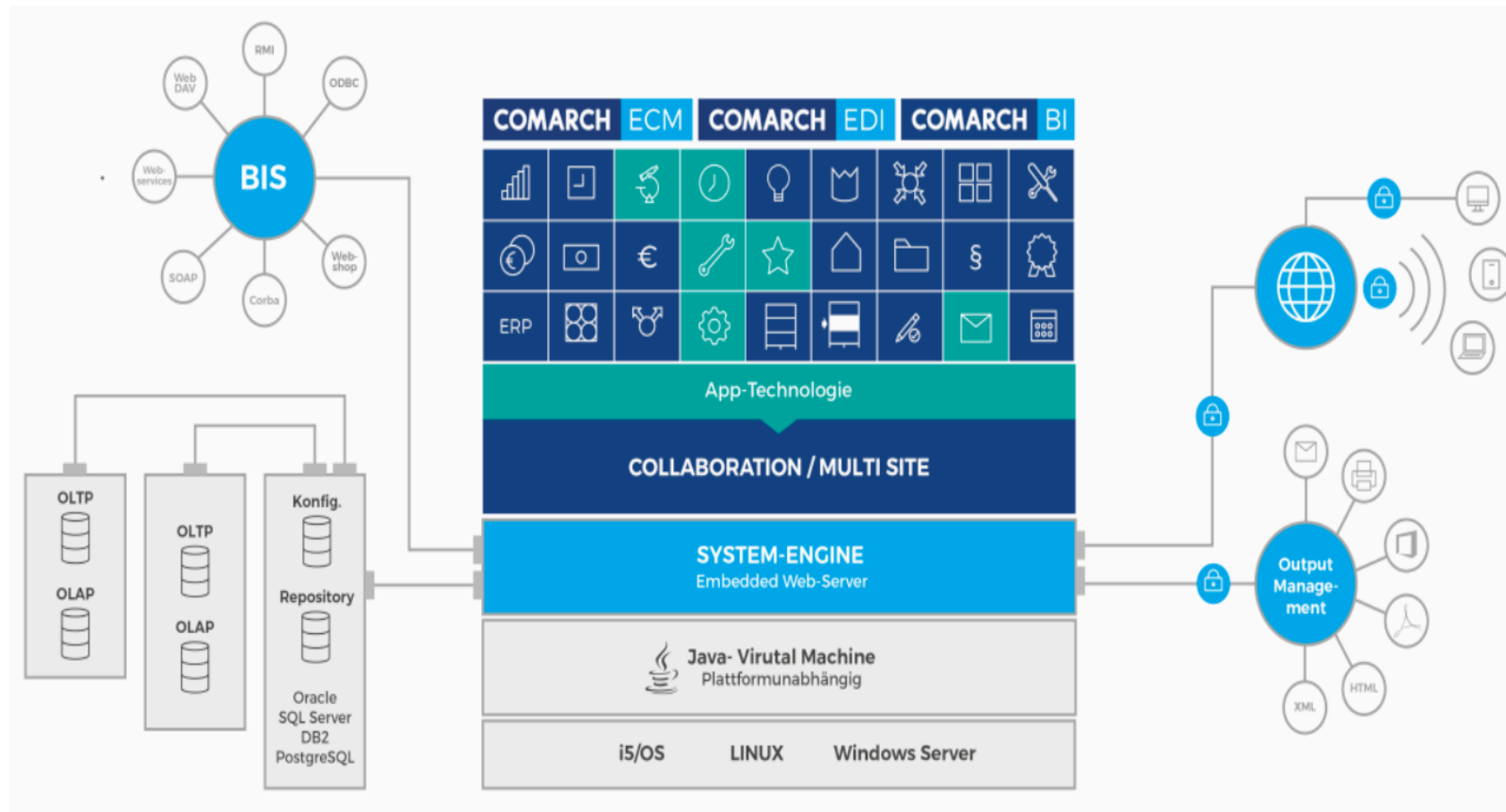


- Infor ION ist Middleware-Integrationsplattform
- basiert auf XML
- Verteilung zwischen Anwendungen über Publish und Susbcribe-Modell
- Kopien werden in Big-Data-Repository gehalten (Business Vault)
- vereint Funktionen zur Überwachung und Optimierung von Geschäftsabläufen auf einer Plattform

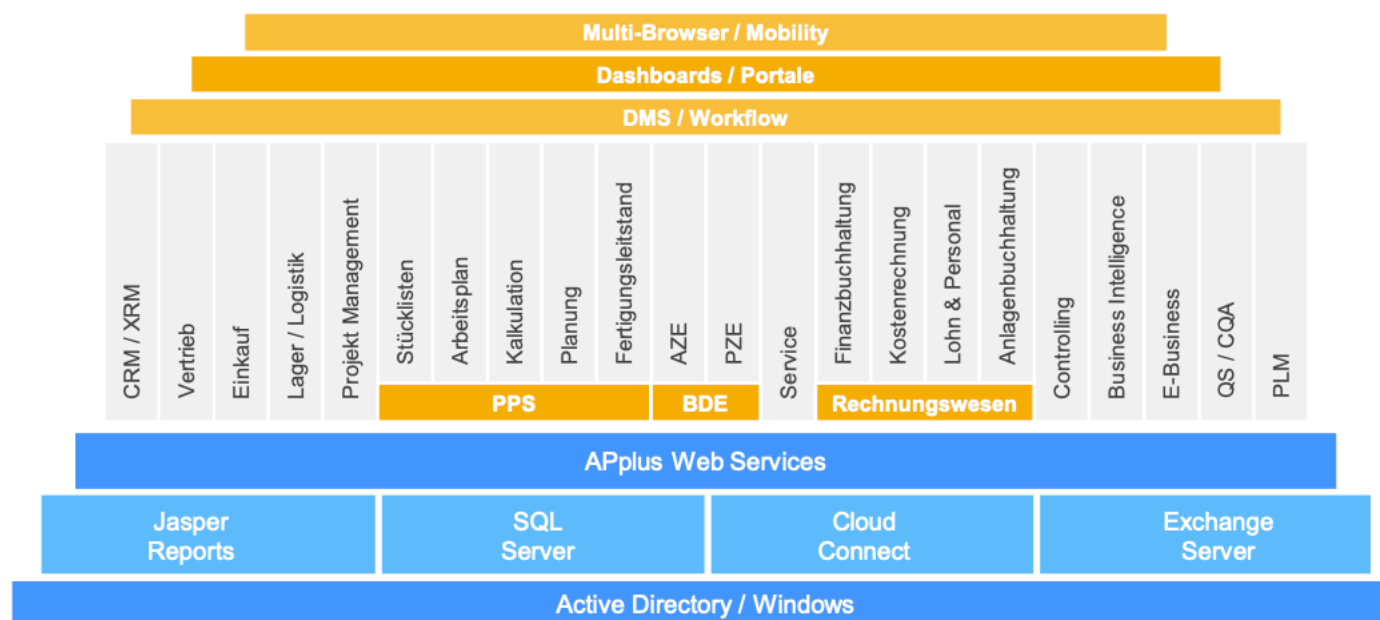
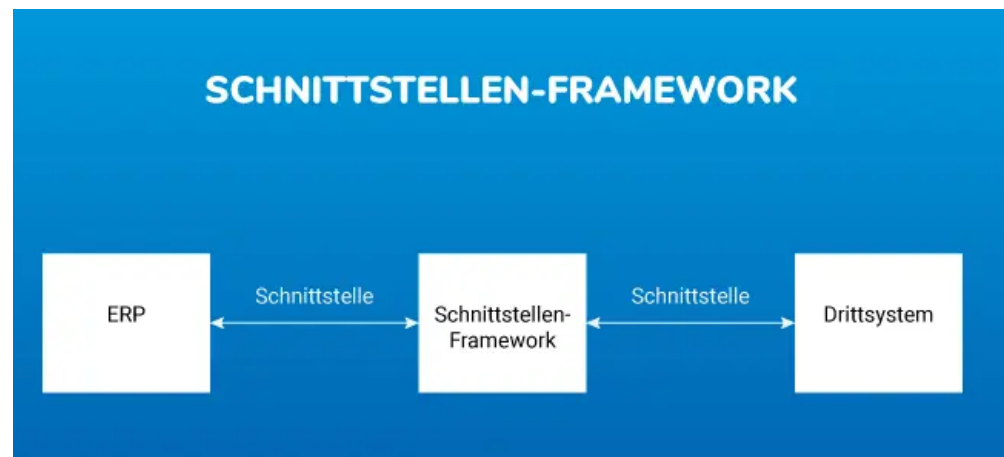


- Integration über PSibus (Enterprise Service Bus)
- enthält Workflowengine für die Steuerung von Prozessen (über XML in Workflowengine)
- Nachrichtenaustausch kann asynchron erfolgen
- Netzwerkverbindung wird über „Keep alive“-Methoden überwacht
- Standardprotokolle werden unterstützt über Plugins (RESTful API und SOAP Webservice verfügbar)

COMARCH



- SOA-Architektur mit kanal- und kanalübergreifenden Logiken
- Integration mit Business Integration Service (BIS)
- stellt flexible Schnittstelle dar
- synchroner und asynchroner Austausch von Daten mit Drittsystemen möglich
- kann manuell über Dialog-Anwendungen, automatisiert über Workflow-Management, über Verarbeitungsaufträge oder als entfernter Aufruf über Standards
- Web-Services, CORBA und WebDav-Protokoll werden unterstützt



- Business Integration Framework (BIF) ist Schnittstellenadapter, der Drittsysteme ohne Programmieraufwand anbinden kann
- Bei Anbindung einer neuen Anwendung Import von XSD-Datei (technische Schnittstellenbeschreibung mit Kommunikationsparametern) und Konfiguration des Daten-Mappings erforderlich
- Externe Anwendung schickt Datenpaket zunächst an APplus Transport-Mechanismus ECom, der Parser des BIF aufruft
- BIF führt syntaktische Prüfung von Paketen durch und liest Daten aus und wandelt sie in Datenbankobjekte um, die APplus interpretieren kann
- Mithilfe von ECom wird mit Drittsystem über dessen Schnittstelle kommuniziert



- Integration Workbench (INBW) als zentrale Plattform für Datenaustausch
- erfolgt über INWB-Schnittstelle mit dem Datenaustauschsystem "Sonic Enterprise Service Bus" (ESB)
- Daten werden mit Hilfe von INWB-Nachrichten (XML-Dateien) ausgetauscht
- für Datenaustausch der unterschiedlichen Datensatzarten gibt es je eine INWB-Nachrichtenart
- INWB-Nachrichtenarten werden anhand von Nachrichtentypen (XSD-Dateien) unterschieden, die in der Systemverwaltung hinterlegt sind
- Nachrichtenversand und Nachrichtenübernahme in der Regel automatisiert über permanente Jobs
- Steuerung der INWB über INWB-Konsole
- WorkflowAutomation für Überwachung des Datenaustauschs

Literatur

Aier, S.; Schönherr, M. (Hrsg.) (2004): Enterprise Application Integration - Flexibilisierung komplexer Unternehmensarchitekturen. Gito-Verlag (Berlin), 2004.

Asseco (2022). ERP-Integration: Drittsysteme ohne Programmieraufwand anbinden und Kosten sparen. [online] <https://www.applus-erp.de/ressourcen/blog/erp-integration-drittsysteme-ohne-programmieraufwand-anbinden-und-kosten-sparen/> (abgerufen am 31.10.2023)

Britton, C., & Bye, P. (2004). IT architectures and middleware: strategies for building large, integrated systems. Amsterdam: Addison-Wesley, 2004 Pearson Education.

Comarch (2022). Comarch ERP Enterprise.[online] <https://www.comarch.com/erp/comarch-erp-enterprise/> (abgerufen am 31.10.2023)

Gronau, N. (2014): Enterprise Resource Planning: Architektur, Funktionen und Management von ERP-Systemen, 2. Auflage. München 2014.

Infor 2022. Intelligent Open Network (ION). [online] <https://www.infor.com/de-de/products/ion> (abgerufen am 31.10.2023)

Kaib, M. (2002). Enterprise Application Integration. Wiesbaden: Deutscher Universitäts-Verlag.

Karcher A. (2016). Middleware. In: Enzyklopädie der Wirtschaftsinformatik [online] <https://wi-lex.de/index.php/lexikon/entwicklung-und-management-von-informationssystemen/systementwicklung/softwarearchitektur/middleware/>(abgerufen am 31.10.2023)

Lebender, M., et al. (2003): Business Integration Software. Werkzeuge, Anbieter, Lösungen. Fraunhofer IRB Verlag (Stuttgart), 2003.

Microsoft (2020). Microsoft BizTalk Server-Dokumentation. [online] <https://docs.microsoft.com/de-de/biztalk/>

OASIS (2020). OASIS ebXML Messaging Services TC. [online] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-msg (abgerufen am 31.10.2023)

OMG 2020: Common Object Request Broker Architecture. [online] <https://www.omg.org/spec/CORBA/About-CORBA/> (abgerufen am 31.10.2023)

ProAlpha (2022). Integration Workbench (INWB). [online] https://support.proalpha.com/72/german/topic/s__p_inwb_konzept.html (abgerufen am 31.10.2023)

PSI penta (2022). PSI penta/MES. [online] https://www.psi-automotive-industry.de/fileadmin/files/downloads/PSI_PAI/Broschueren/psi-mes-de-web.pdf (abgerufen am 31.10.2023)

Puppet (2020): Middleware Magic. [online] <https://docs.puppetlabs.com/mcollective/images/middleware-magic.gif> (abgerufen am 03.08.2020)

SAP (2022). Process Integration. [online] https://help.sap.com/doc/saphelp_nwpi71/7.1/de-DE/0f/80243b4a66ae0ce10000000a11402f/frameset.htm (abgerufen am 31.10.2023)

Schönherr, M. (2005): Enterprise Application Integration (EAI und Middleware). Grundlagenarchitekturen und Auswahlkriterien. ERP Management 1/2005, Heft 1, S. 25-29.

Thränert, M. (2005): Integration - Eine Begriffsbestimmung, In: Fähnrich, K.; Thränert, M.; Wetzel, P. (Hrsg.): Umsetzung von kooperativen Geschäftsprozessen auf eine internetbasierte IT-Struktur. Leipziger Beiträge zur Informatik, Band III, (Leipzig) 2005, S. 11-22.