Seminarvortrag
„Serviceorientierte Softwarearchitekturen“

Architekturen für bereits vorhandene Altsysteme

Michael Kuls

Gliederung

- Einführung
- Grundlegende Modelle
- Grundlegende Komponenten
- Architekturen

Einführung

- Altanwendung und Altsysteme?
- Probleme durch heterogene Systeme
  - unterschiedliche Programmiersprachen, Strukturen und Techniken
- Middleware
  - plattformabhängig und aufwendig in der Entwicklung
- Chancen durch einheitlichen Standard
  - komplett unabhängig

Einführung (2)

- einfache Integration
- sanfte Migration
- lose Komponenten

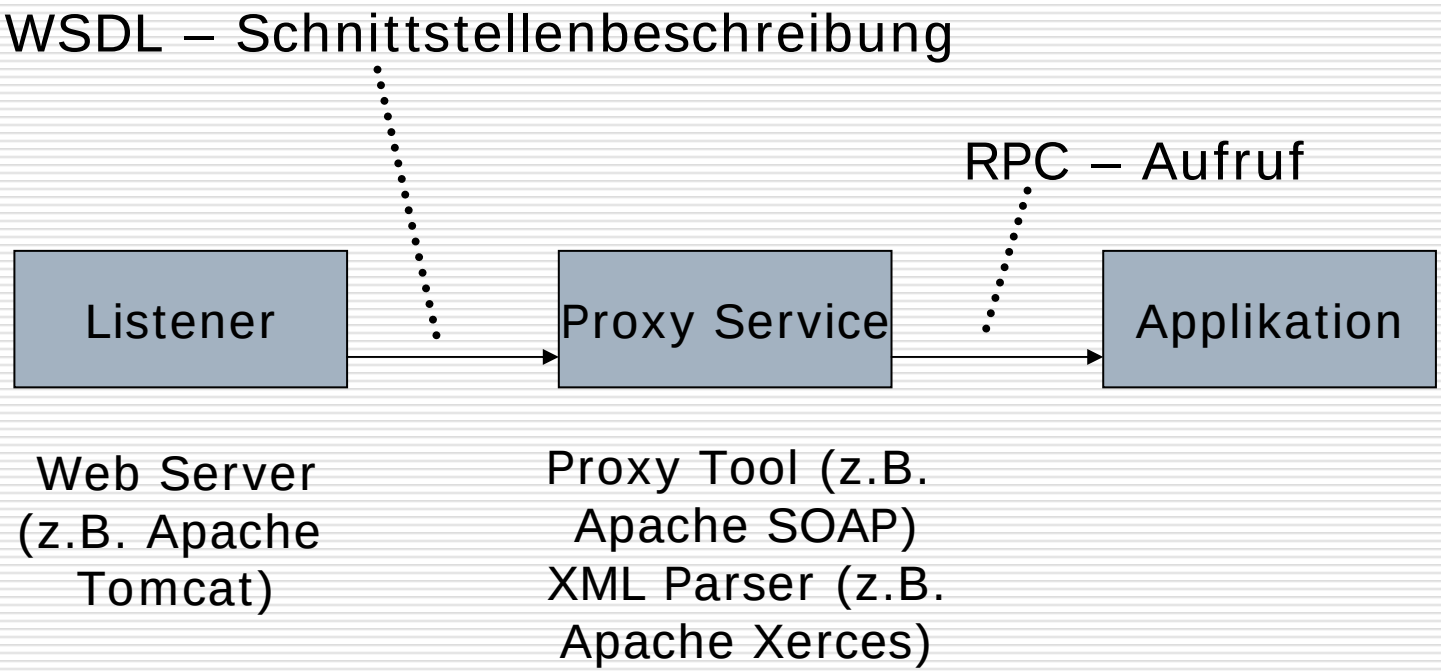
Gliederung

- Einführung
- Grundlegende Modelle
  - Proxy Service
  - Wrapper Service
    - Vergleich der bisherigen Services
  - Coordination Service
- Grundlegende Komponenten
- Architekturen

Proxy Service

- Abbild der Programmschnittstelle
- keine eigene Logik
- WSDL wird automatisch erzeugt
- arbeitet ausschließlich über RPC

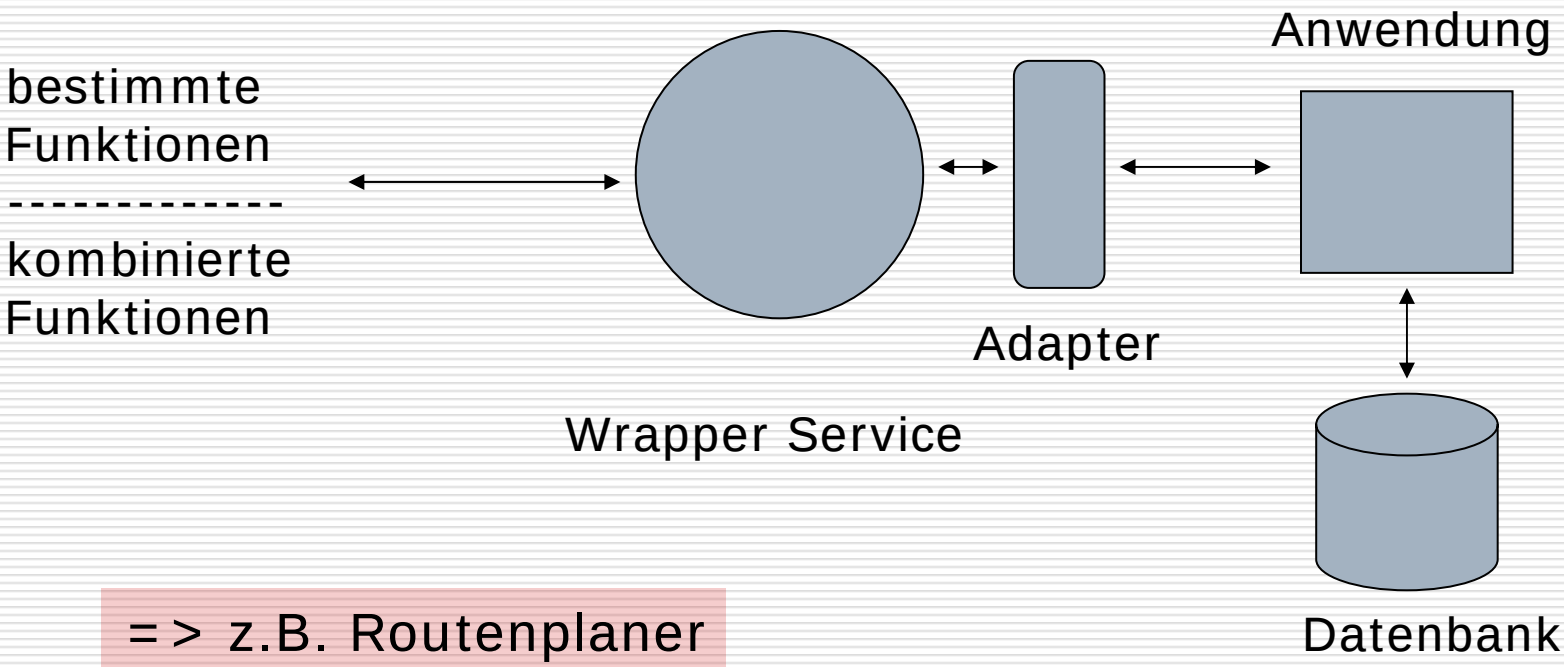
Proxy Service (2)



Wrapper Service

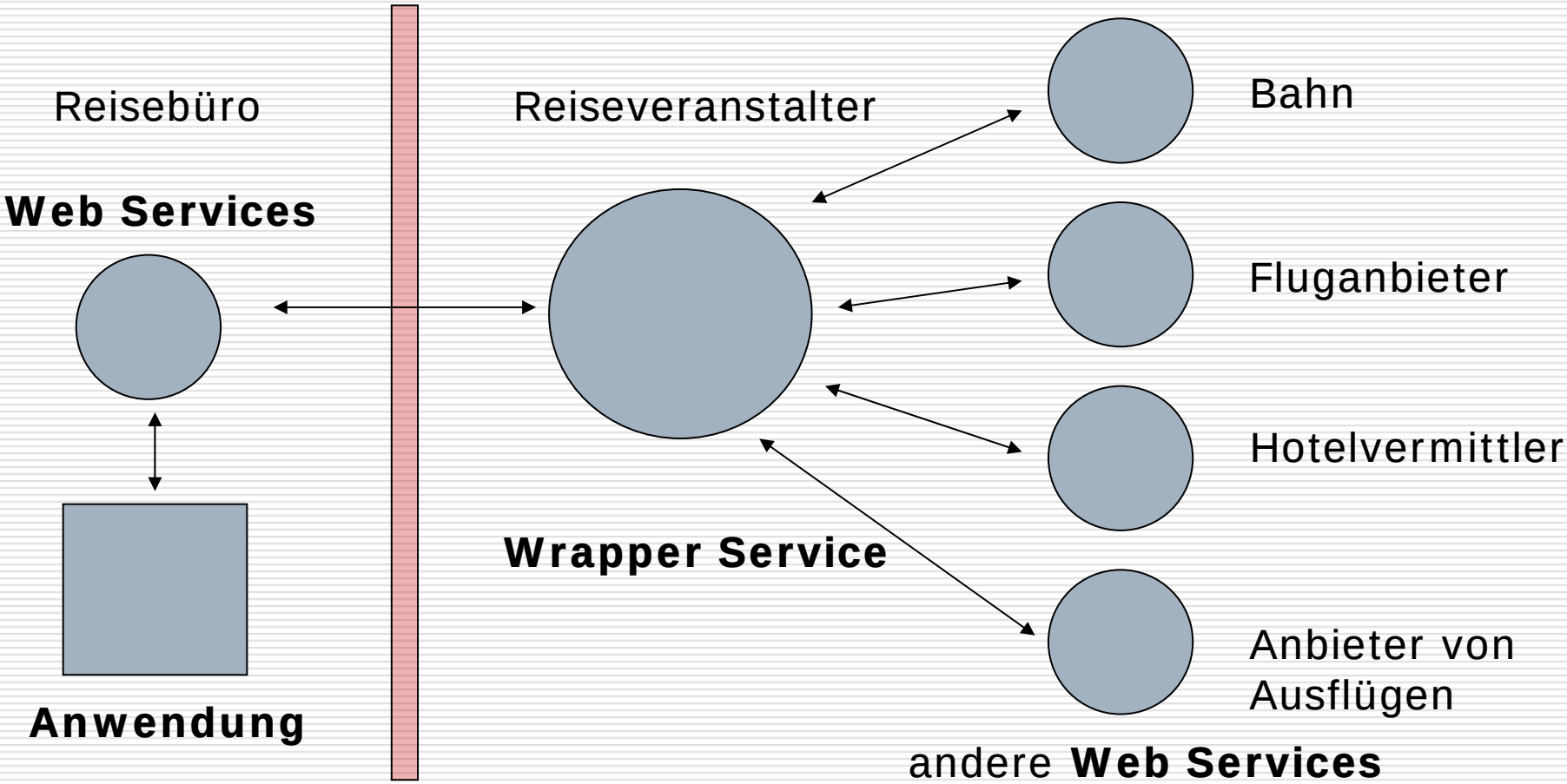
- umschließt das Altsystem und deren Komponenten
- bildet Funktionen ab
- normalerweise keine eigene Logik
  → Business-Service
- arbeitet über RPC und Nachrichten
- wird meistens selbst entwickelt

Wrapper Service (2)



=> z.B. Routenplaner

Wrapper Service (3)



Vergleich der bisherigen Services

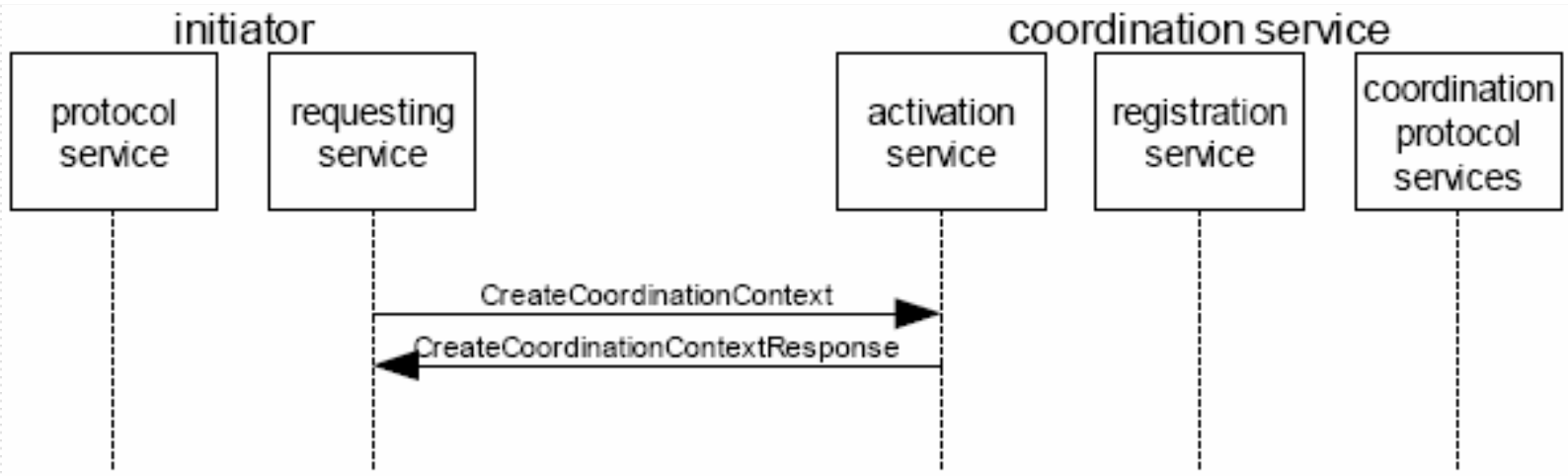
Proxy Service	Wrapper Service	Business Service
wird in der Regel automatisch erstellt	wird manuell entwickelt	wird manuell entwickelt
spiegelt nur die komplette Schnittstelle der Altanwendung	kapselt meist mit Hilfe eines Adapters eine Funktionalität bzw. eine Kombination der Funktionen der Altanwendung	kapselt die komplette Logik der Altanwendung zu einem Service und fügt neue Logiken hinzu
Nachrichtentyp: RPC	Nachrichtentyp: RPC oder Dokument	Nachrichtentyp: Dokument

Coordination Service

- koordiniert drei Services
  - context creation
  - registration for coordination context
  - protocol selection
- Kontext
- Aktivität bzw. Activity
- Selbstständigkeit
- Atomic bzw. Business Transaction

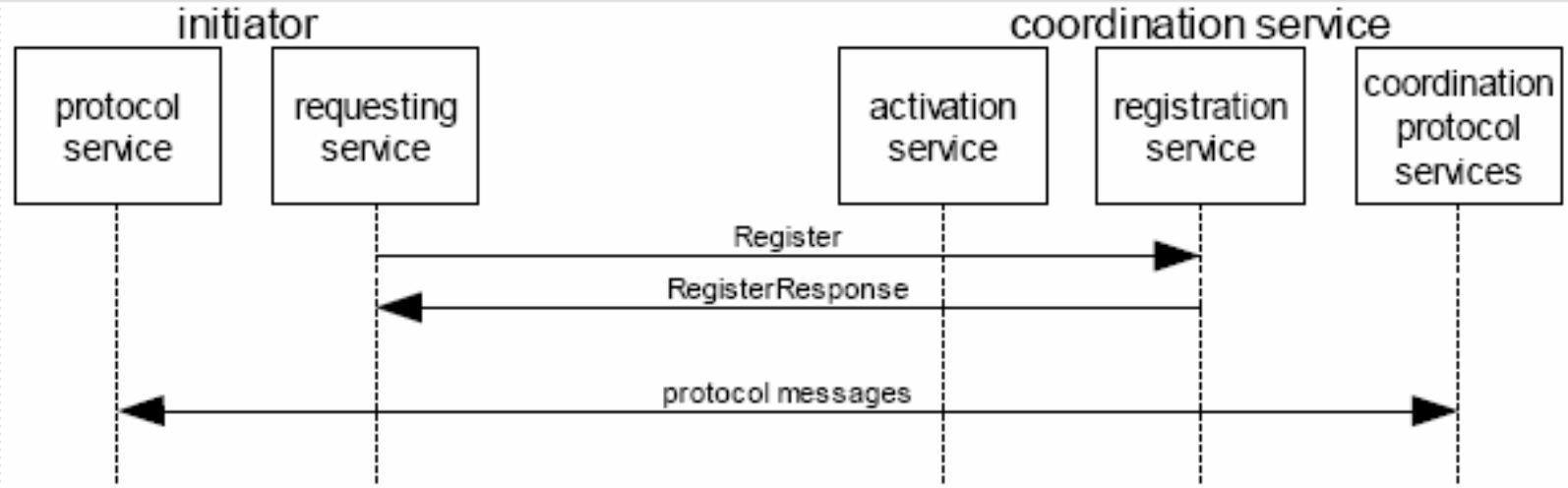
Coordination Service (2)

- context creation
  - Kontext wird erstellt
  - ausgehend vom Initiator
  - Aktivität entsteht durch eine positive Antwort des Services



Coordination Service (3)

- registration for coordination context
  - „Frage und Antwort“ – Prinzip
  - bei erfolgreicher Verbindung: Protokoll wird gleich mitgestartet



Gliederung

- Einführung
- Grundlegende Modelle
- Grundlegende Komponenten
  - Adapter
  - Intermediaries
  - Interceptors
- Architekturen

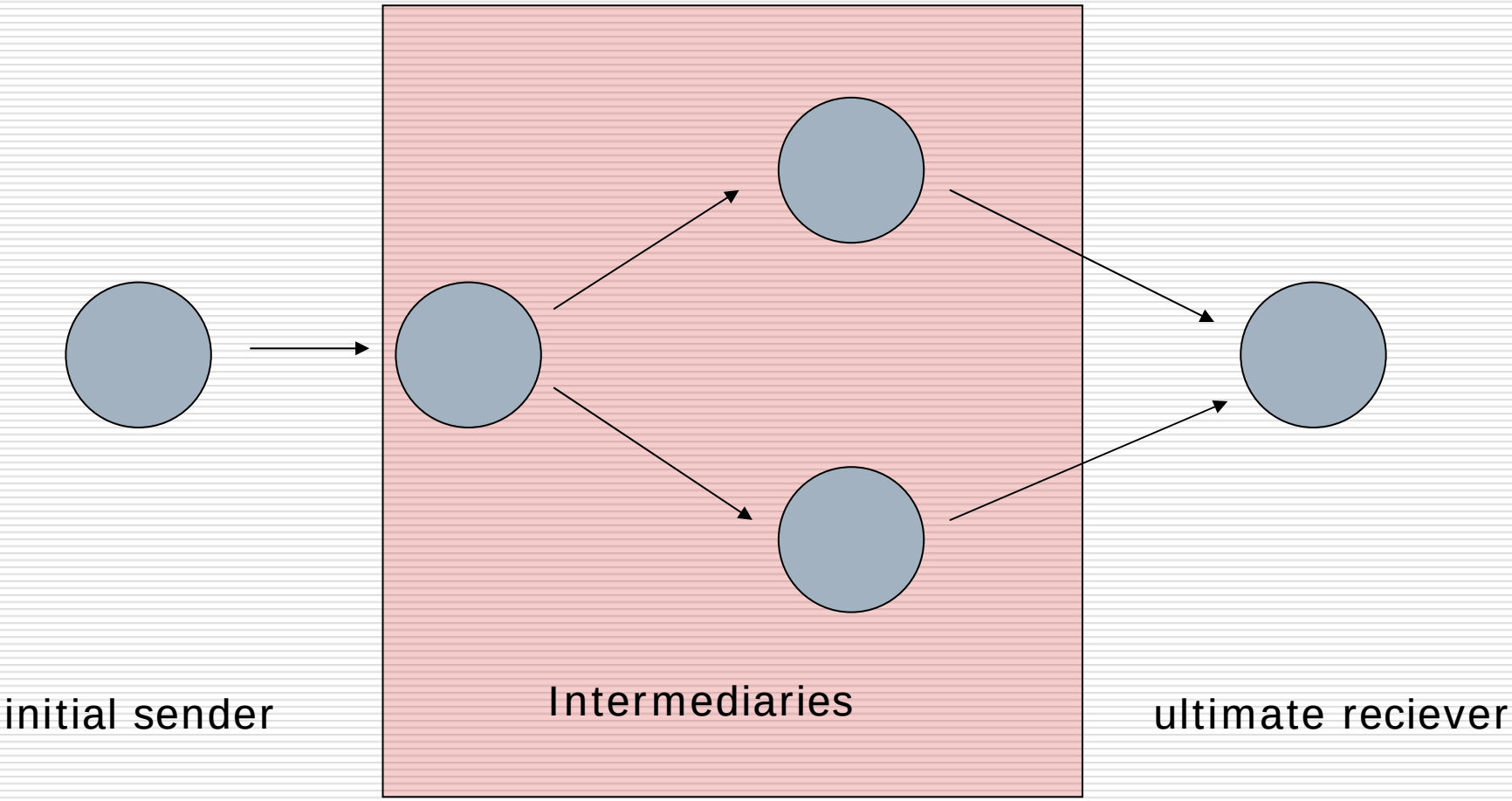
Adapter

- eigenständige Softwarekomponenten
- Brücke zwischen Programmen bzw. Plattformen
- großer Stellenwert
- Einsatzgebiete:
  - zwischen zwei Altsystemen
  - zwischen einem Web Service und einem Altsystem
  - zwischen Web Service und einer Datenbank

Intermediaries

- zwischen Stationen möglich
- zwei Kategorien:
  - forwarding Intermediaries
    - nur Weiterleitung
  - active Intermediaries
    - Änderungen am Header
- keine eigene Logik gestattet
- Routing

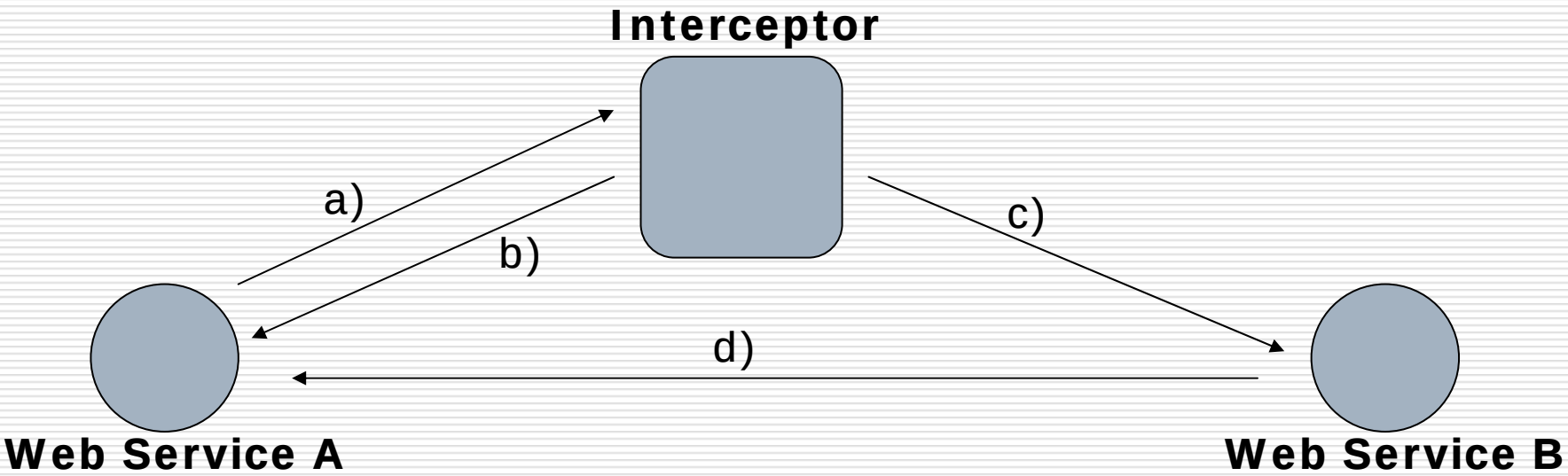
Intermediaries (2)



Interceptors

- unabhängige Softwarekomponenten
- verändern die Nachricht
- können Entscheidungen treffen
- Einsatzmöglichkeiten:
  - Sicherheit
  - Überprüfung auf Vollständigkeit

Interceptors (2)



a) Nachricht von Web Service A an Web Service B
b) Interceptor lehnt ab und sendet eine Antwort an Web Service A

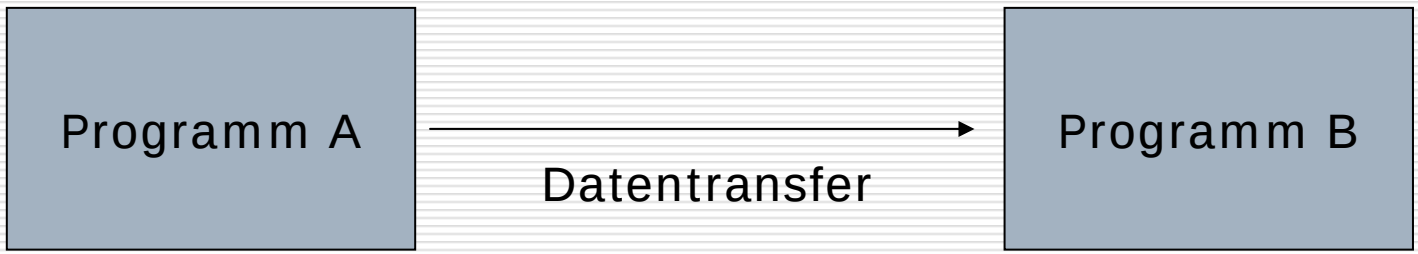
ODER

c) Interceptor akzeptiert und leitet weiter an Web Service B
d) Web Service B sendet seine Antwort auf die Nachricht von Web Service A

Gliederung

- Einführung
- Grundlegende Modelle
- Grundlegende Komponenten
- Architekturen
  - Datentransfer nur in eine Richtung
  - Punkt zu Punkt Datentransfer
  - Datenverkehr mit Datenbanken

Datentransfer in eine Richtung



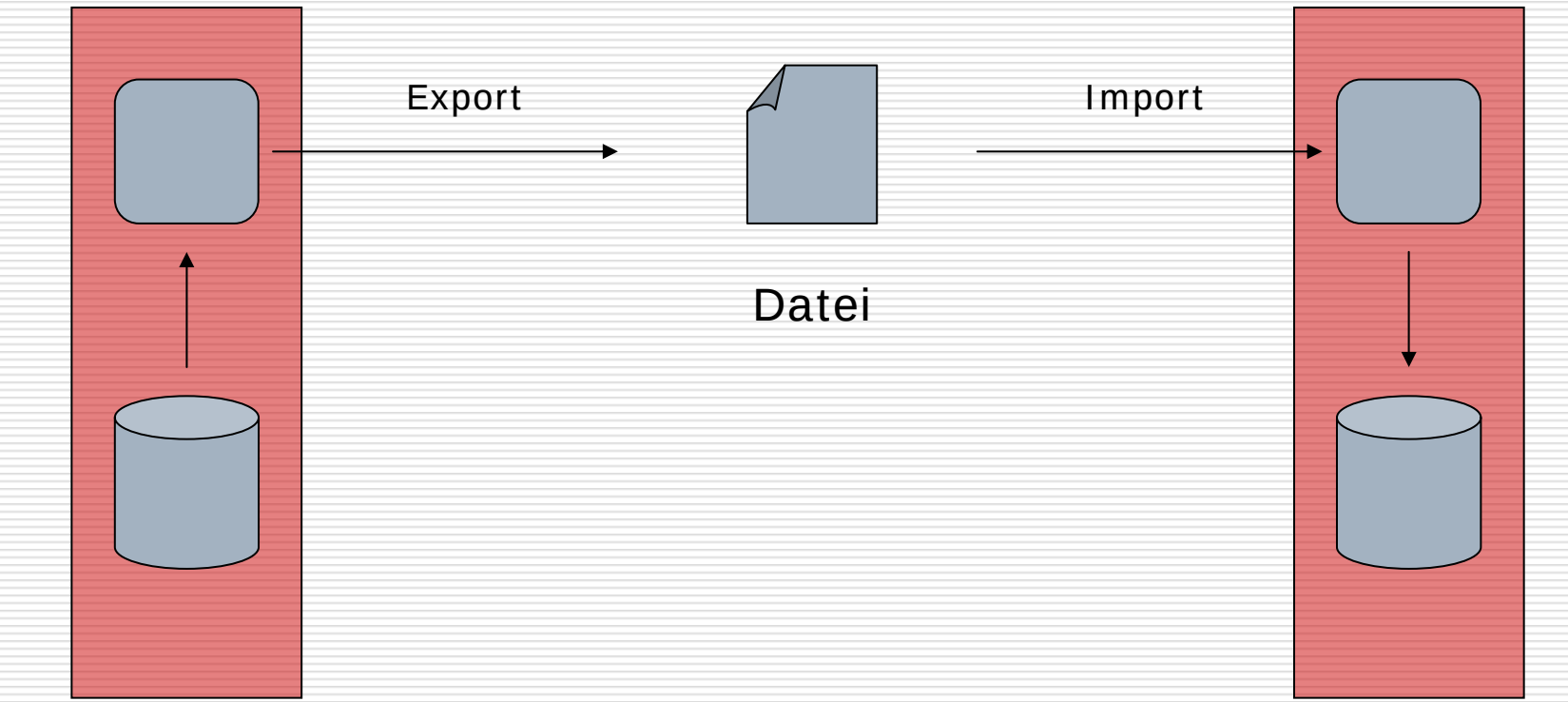
- zwei Variationen:
  - batchgesteuerter Datenzugriff
  - direkter Datenzugriff
- einfach und günstig

Datentransfer in eine Richtung (2)
Batchgesteuerter Datentransfer

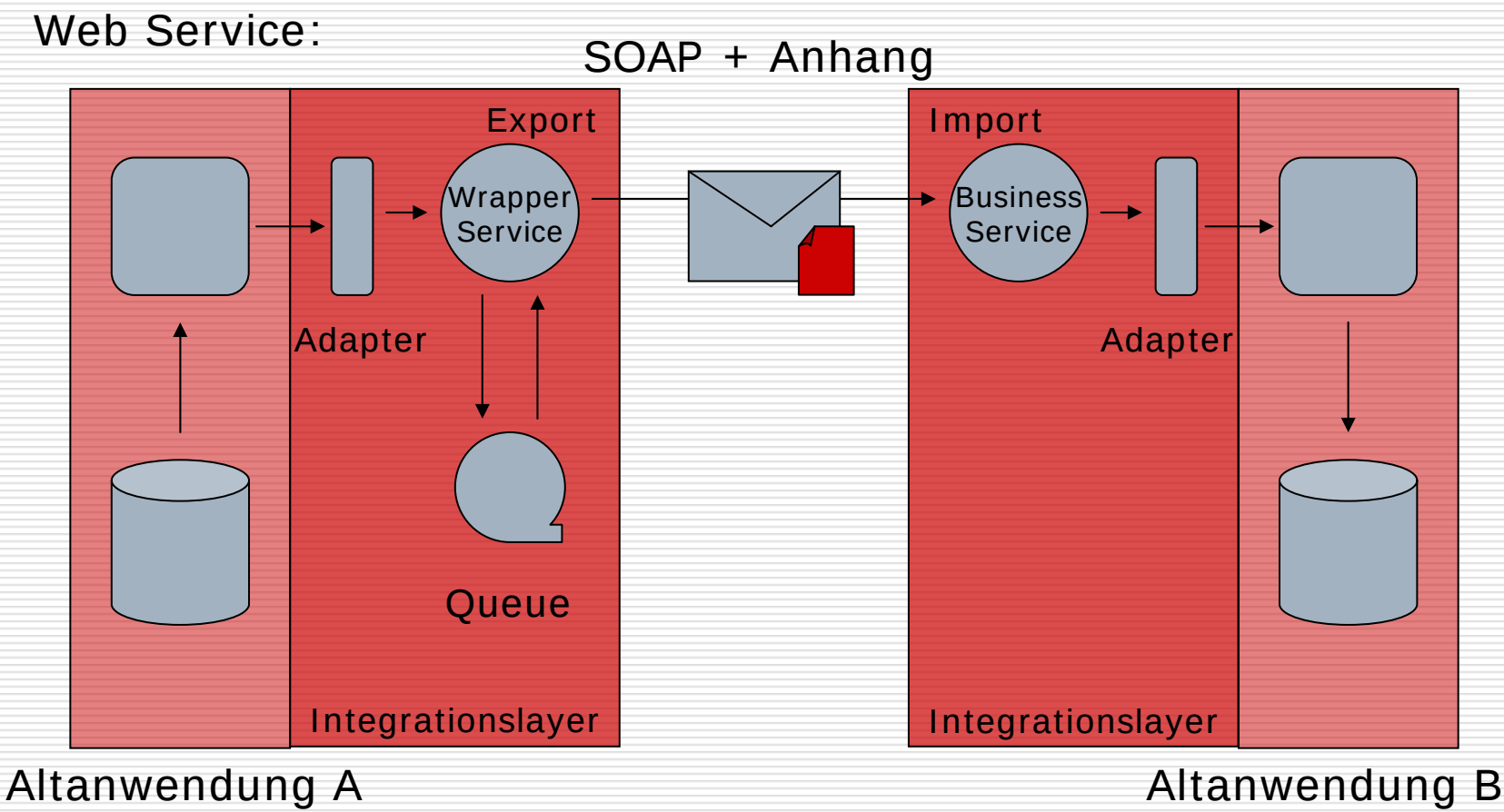
- Traditionell:
  - Verzeichnis in der Mitte
  - manueller bzw. automatischer Auslöser
  - Daten werden in Dateien ausgetauscht
  - für „real-time“ zu großer Aufwand
- Web Services:
  - Integrationsschicht mit Web Service
  - z.B. Wrapper Service (Export) und Business Service (Import)
  - SOAP Nachrichten (optional mit Anhang)
  - Queue (optional)

Datentransfer in eine Richtung (3)
Batchgesteuerter Datentransfer

Traditionell:



Datentransfer in eine Richtung (4)
Batchgesteuerter Datentransfer



Datentransfer in eine Richtung (5)
Batchgesteuerter Datentransfer

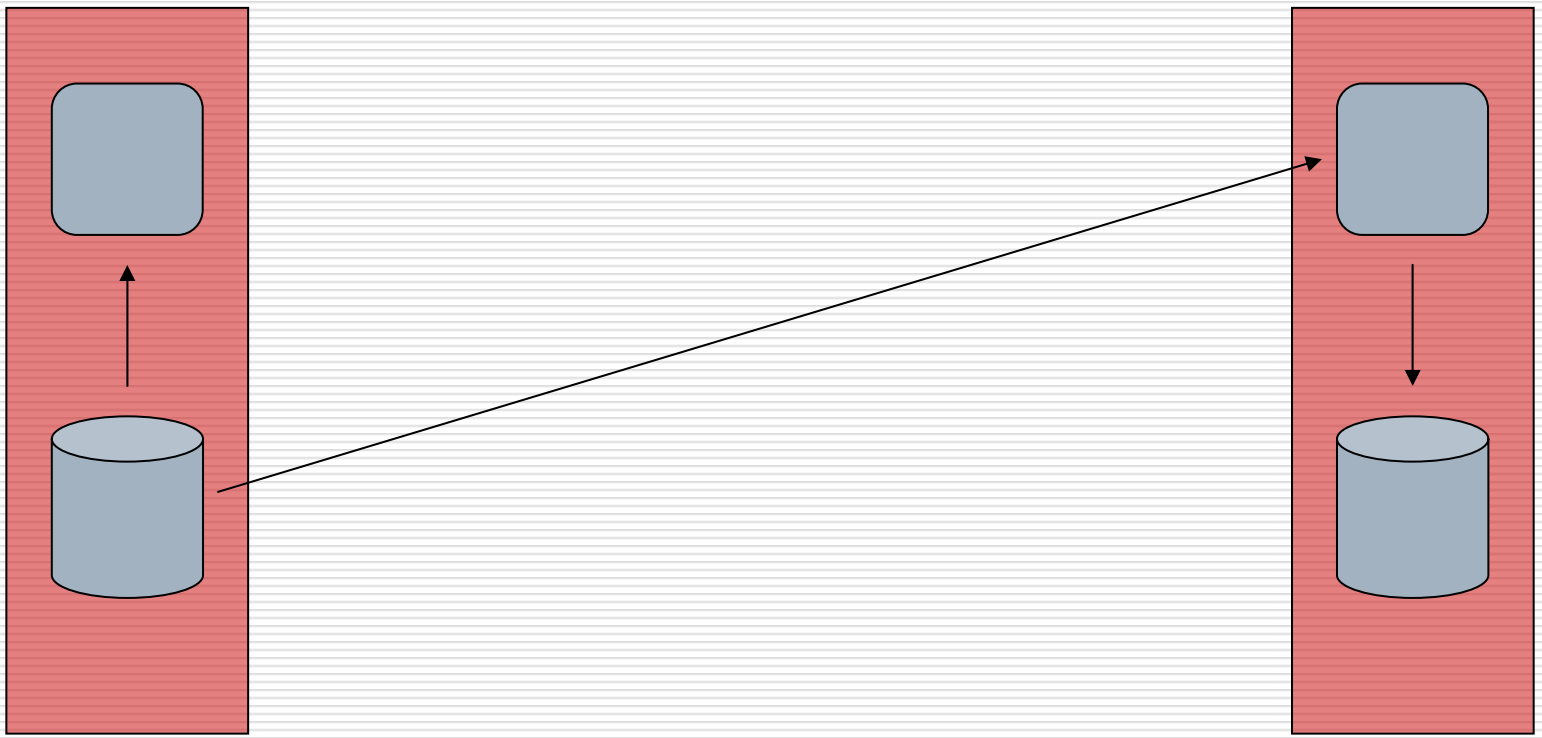
Merkmal	Traditionell	SOA
Datentransfer	manuell oder automatisch	manuell oder automatisch
Transformation	manuell oder durch Anwendung	Integrationsschicht (durch Logik bzw. XSLT)
Datenformat	Datei	SOAP Anhang oder als XML
Validierung	Anwendungen validieren	Integrationslayer validiert auch XSD Schema möglich
Performance	Intervall oder manuell	„Angebot bzw. Nachfrage“
Sicherheit	Verzeichniszugriffsrechte	keine Zwischenstation; Digitale Signaturen
Transaktion	Nicht möglich	Atomic Trans. Coordinator
Kompatibilität	ein Datenformat	versch. Datenformate (XML)
Erweiterbarkeit	Programmlogik	zusätzlich die Web Services

Datentransfer in eine Richtung (6)
Direkter Datenzugriff

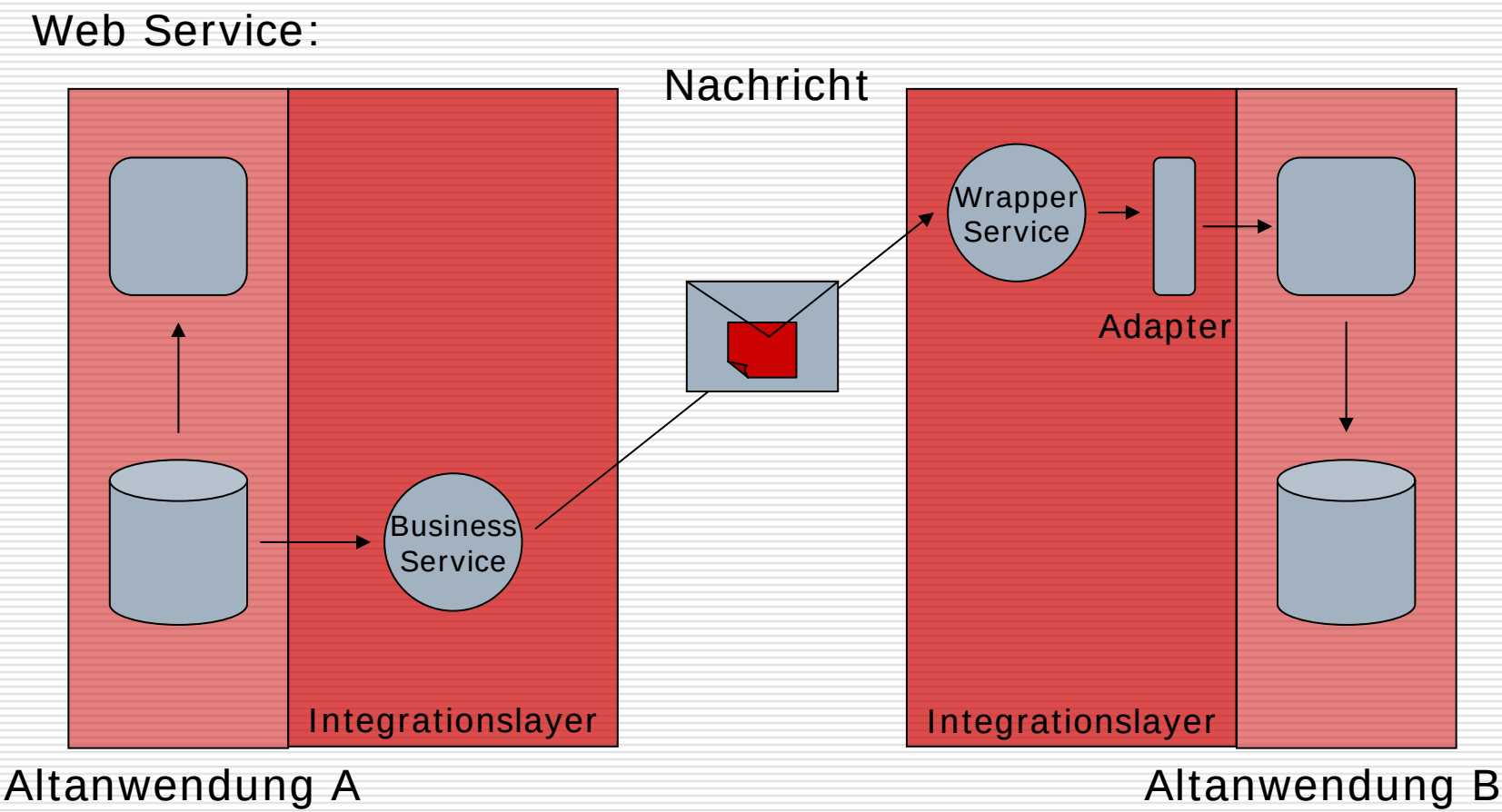
- Traditionell:
  - Zugriff auf Daten andere Applikation
  - Zugriffsteuerung durch Datenbankzugang mit eigenen Views
  - „real-time“
- Web Services:
  - Integration Layer mit Web Service
  - z.B. Wrapper Service bzw. Business Service steuern Zugriff bzw. erzeugen Abfragen
  - SOAP Nachrichten (optional mit Anhang)

Datentransfer in eine Richtung (7)
Direkter Datenzugriff

Traditionell:



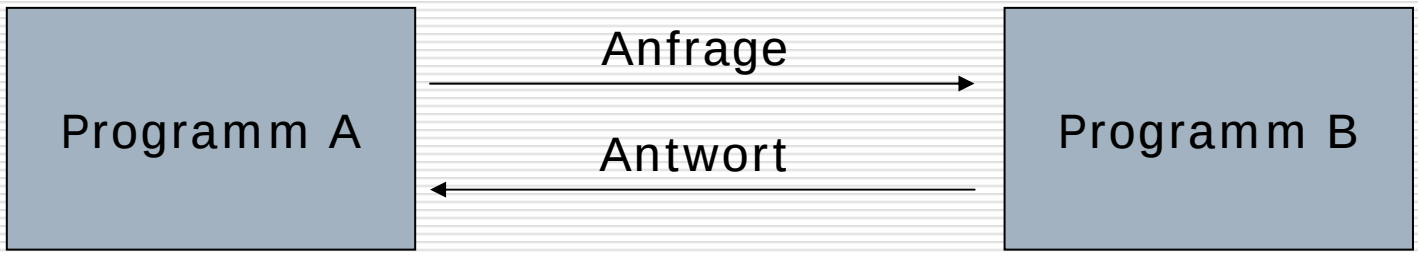
Datentransfer in eine Richtung (8)
Direkter Datenzugriff



Datentransfer in eine Richtung (9) Direkter Datenzugriff

Merkmal	Traditionell	SOA
Datentransfer	Datenzugriffsprotokoll	Services
Transformation	keine benötigt	XML
Datenformat	ODBC	SOAP mit XML
Validierung	Datenbank validiert	Integrationslayer validiert auch XSD Schema möglich
Sicherheit	Datenbank	WS-Security Framework; Digitale Signaturen
Kompatibilität	weitere Anwendungen; DBMS	versch. Datenformate (XML)
Erweiterbarkeit	Datenbank (Views, Indizes)	Businesslogik

Punkt-zu-Punkt Datentransfer



- verschiedene Umgebungen:
  - homogene Systeme (fest verbunden)
  - heterogene Systeme (fest verbunden)
  - mehrere Altsysteme
  - homogene Systeme (komponentenbasiert)
  - heterogene Systeme (komponentenbasiert)

Punkt-zu-Punkt Datentransfer (2)
Homogenes bzw. heterogenes System

- Traditionell:
  - feste Kommunikations-parameter
  - Abhängigkeiten
- Web Services:
  - SOAP & XML
  - Integrationsschicht
- Traditionell:
  - zusätzlich Adapter-komponenten zu Transformation unter den Anwendungen
- Web Services:
  - gleicher Aufwand wie beim homogenen System

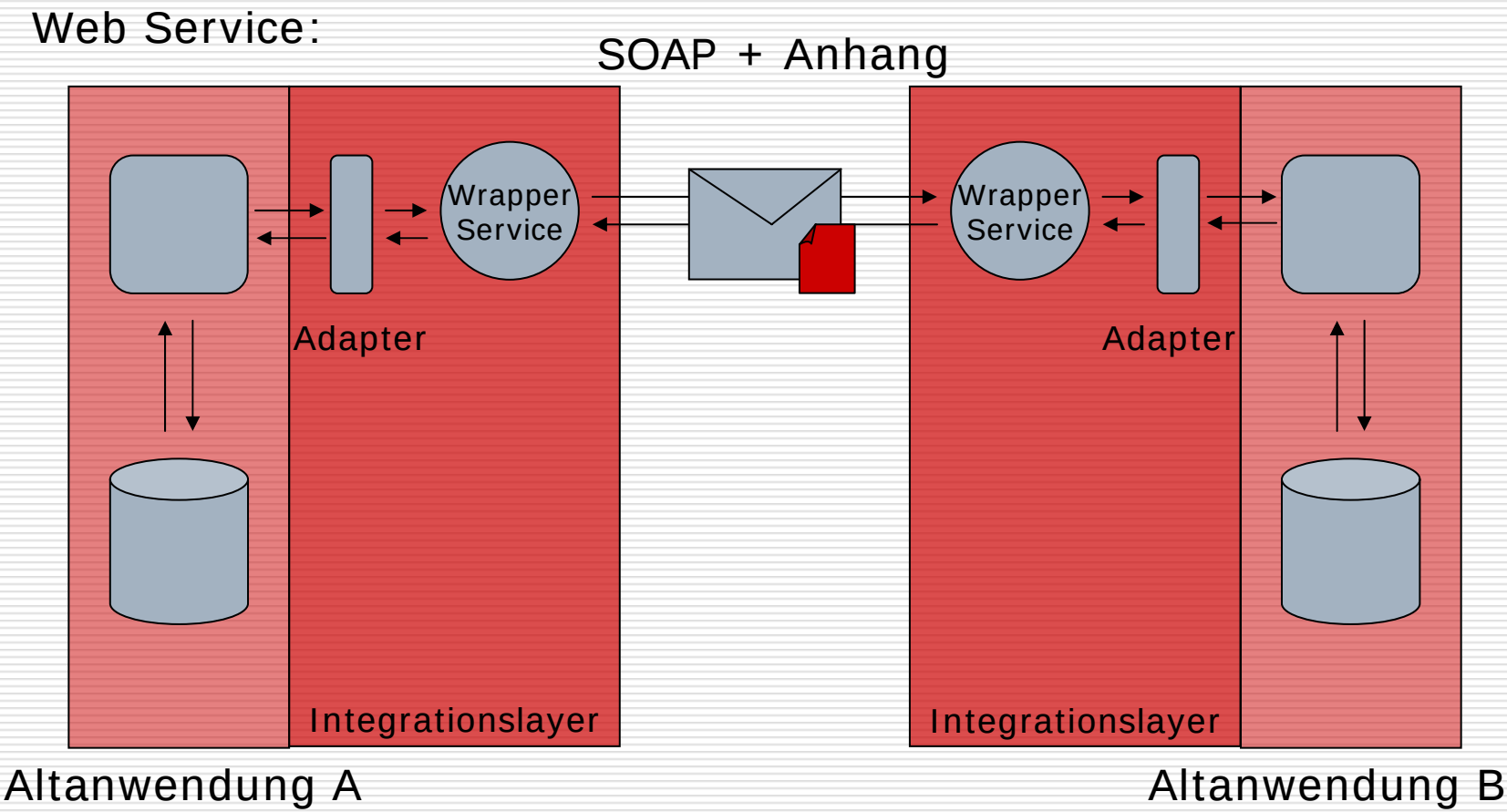
Punkt-zu-Punkt Datentransfer (3)
Homogenes System

Traditionell:

Punkt-zu-Punkt Datentransfer (4)
Heterogenes System

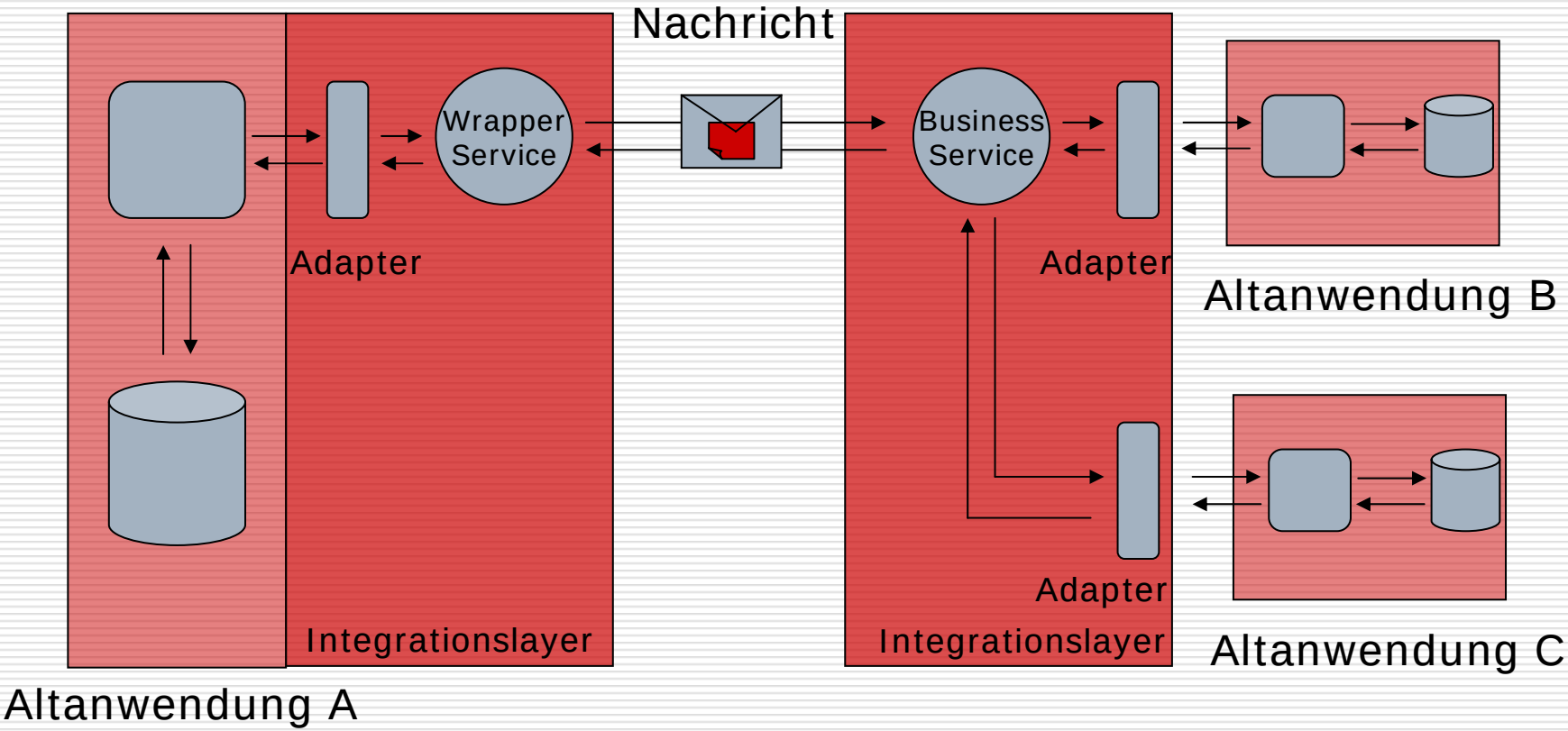
Traditionell:

Punkt-zu-Punkt Datentransfer (5)
Homogenes bzw. heterogenes System



Punkt-zu-Punkt Datentransfer (6)
Mehrere Altsysteme

Web Service:

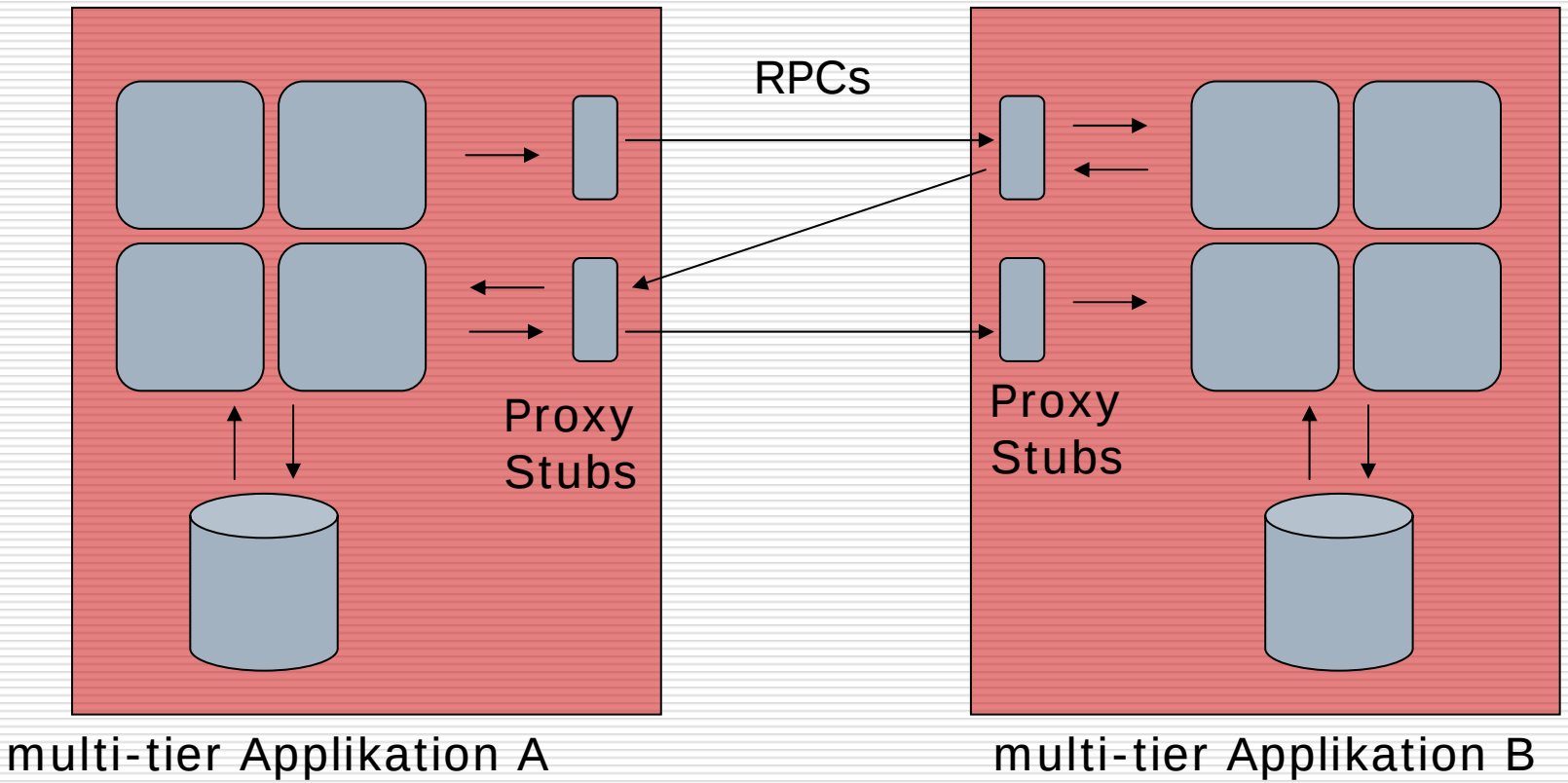


Punkt-zu-Punkt Datentransfer (7)
Homo- bzw. heterogenes komp. System

- Traditionell:
  - entfernte Kommunikation durch Proxy Stubs
  - RPC
- Web Services:
  - SOAP & XML
  - Integrationsschicht
- Traditionell:
  - Middleware
  - Adapter für die plattform-übergreifende Kommunikation
- Web Services:
  - gleicher Aufwand wie beim homogenen System
  - keine Adapter notwendig

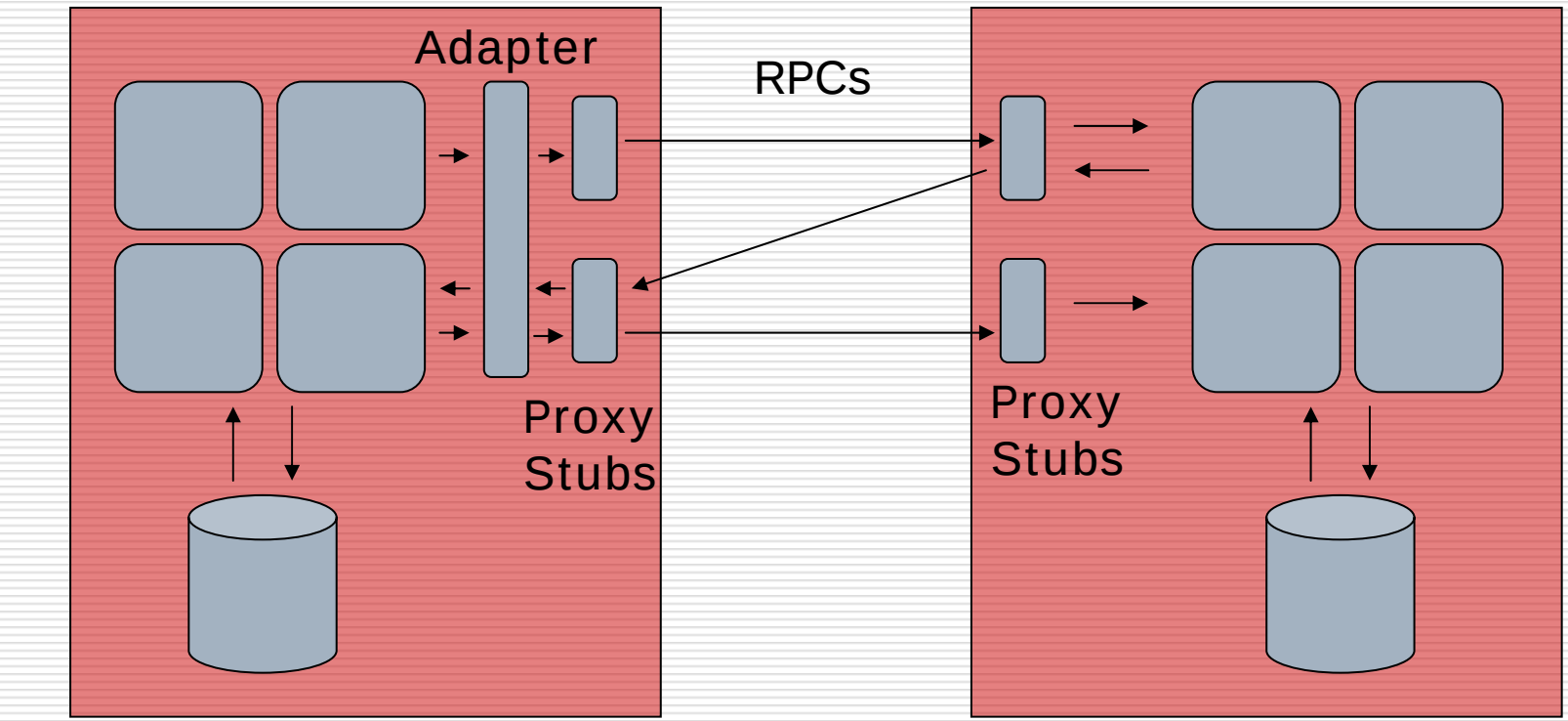
Punkt-zu-Punkt Datentransfer (8)
Homogenes System (komponentenbasiert)

Traditionell:



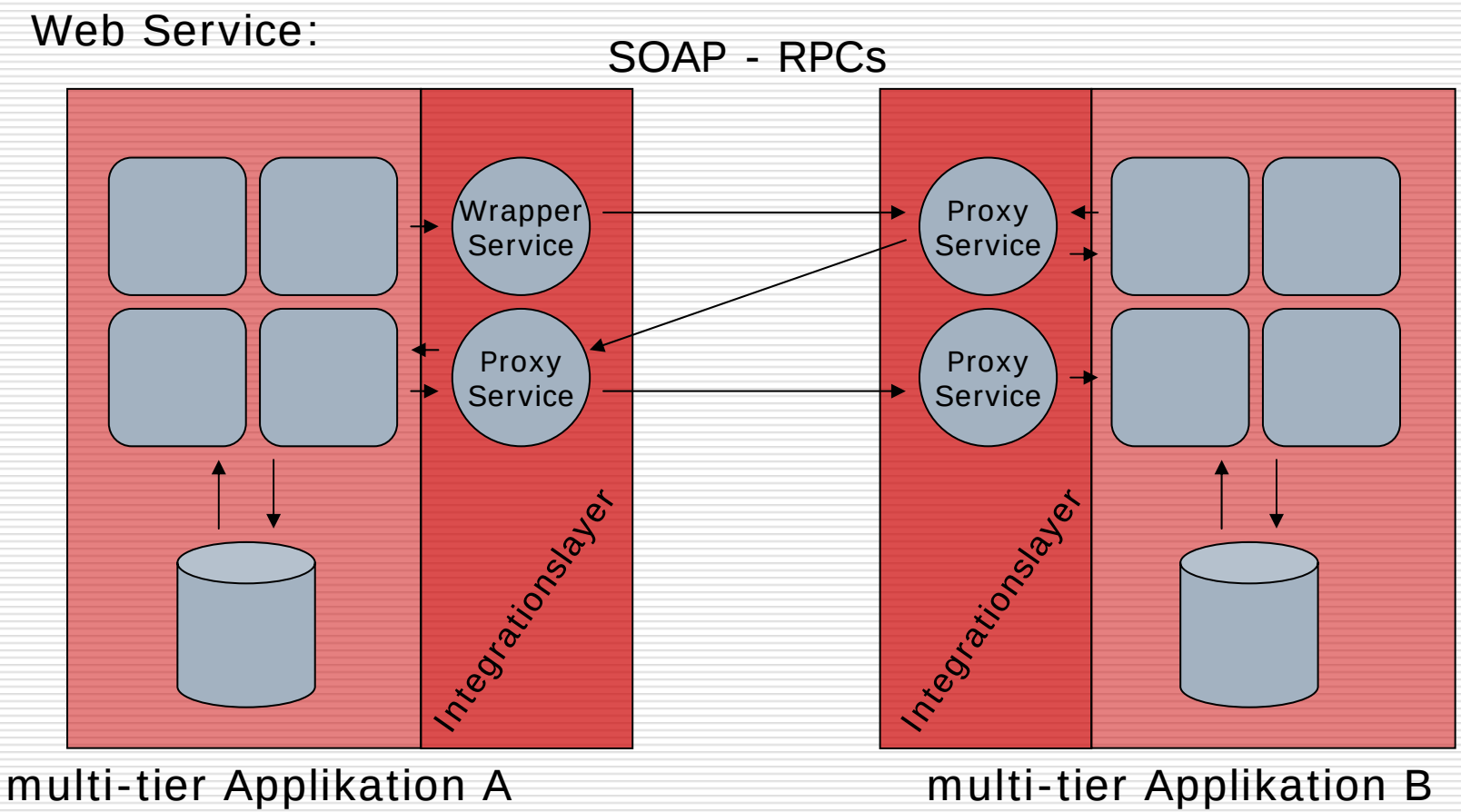
Punkt-zu-Punkt Datentransfer (9)
Heterogenes komponentenbasiertes System

Traditionell:

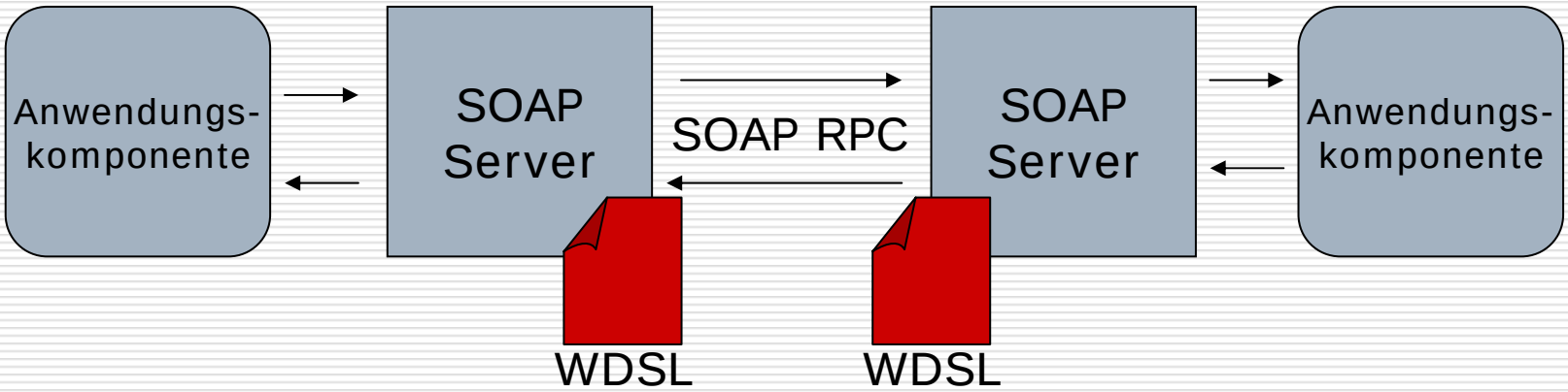


Punkt-zu-Punkt Datentransfer (10)

Homo- bzw. heterogenes komp. System

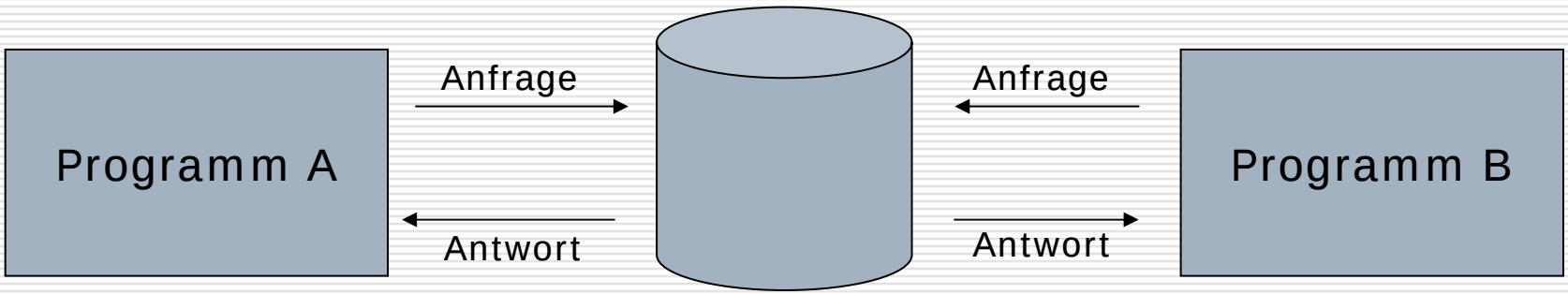


Punkt-zu-Punkt Datentransfer (11)
SOAP-Server und Komponenten



- Vorteil
  - keine Adapter für Protokolltransformation
    - weniger Aufwand
    - einfacher

Architektur für zentrale Datenbanken

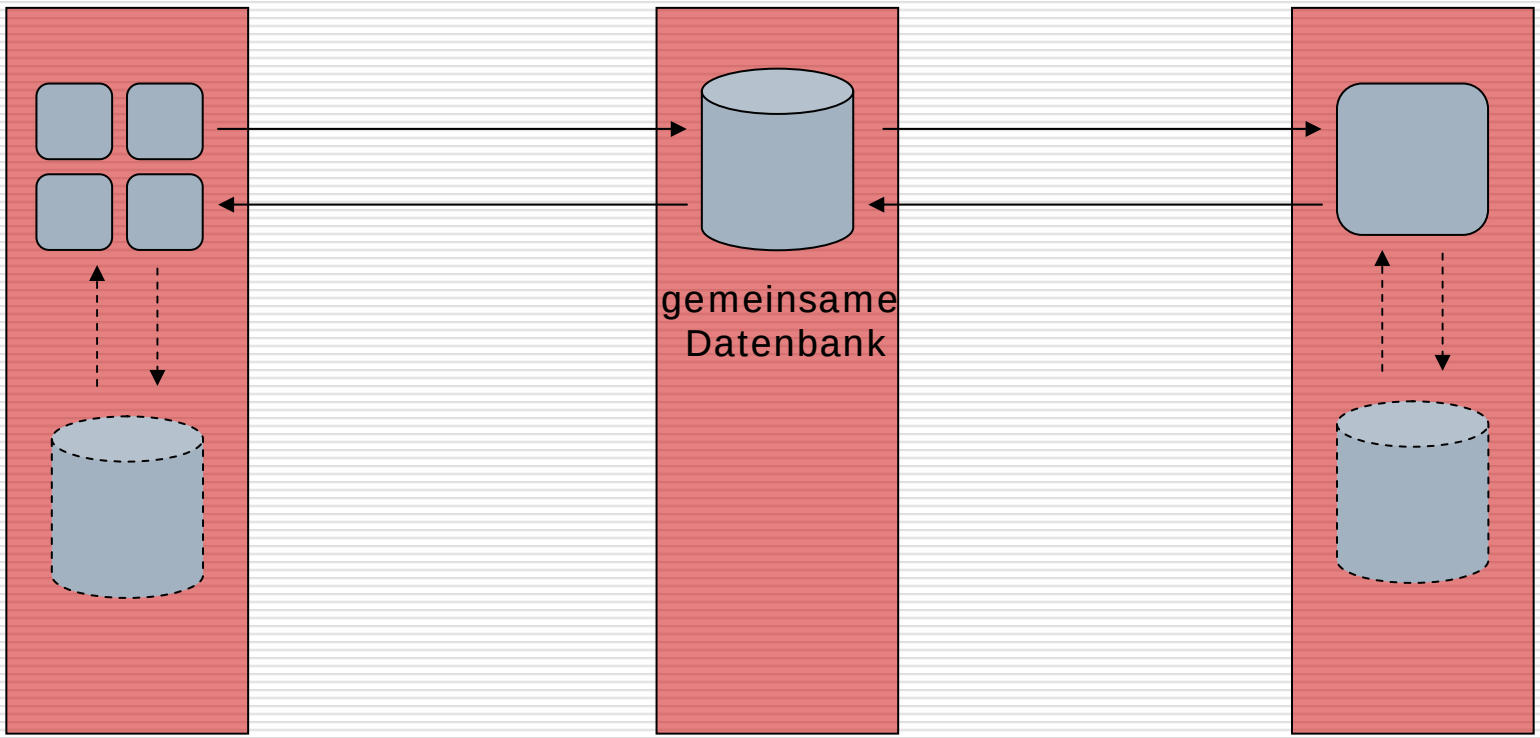


gemeinsame Datenbank

- Traditionell:
  - Zugriffsteuerung über die Datenbank
- Web Service:
  - Wrapper Service als Anlaufpunkt
  - somit generischer Datenzugriff

Architektur für zentrale Datenbanken (2)

Traditionell:

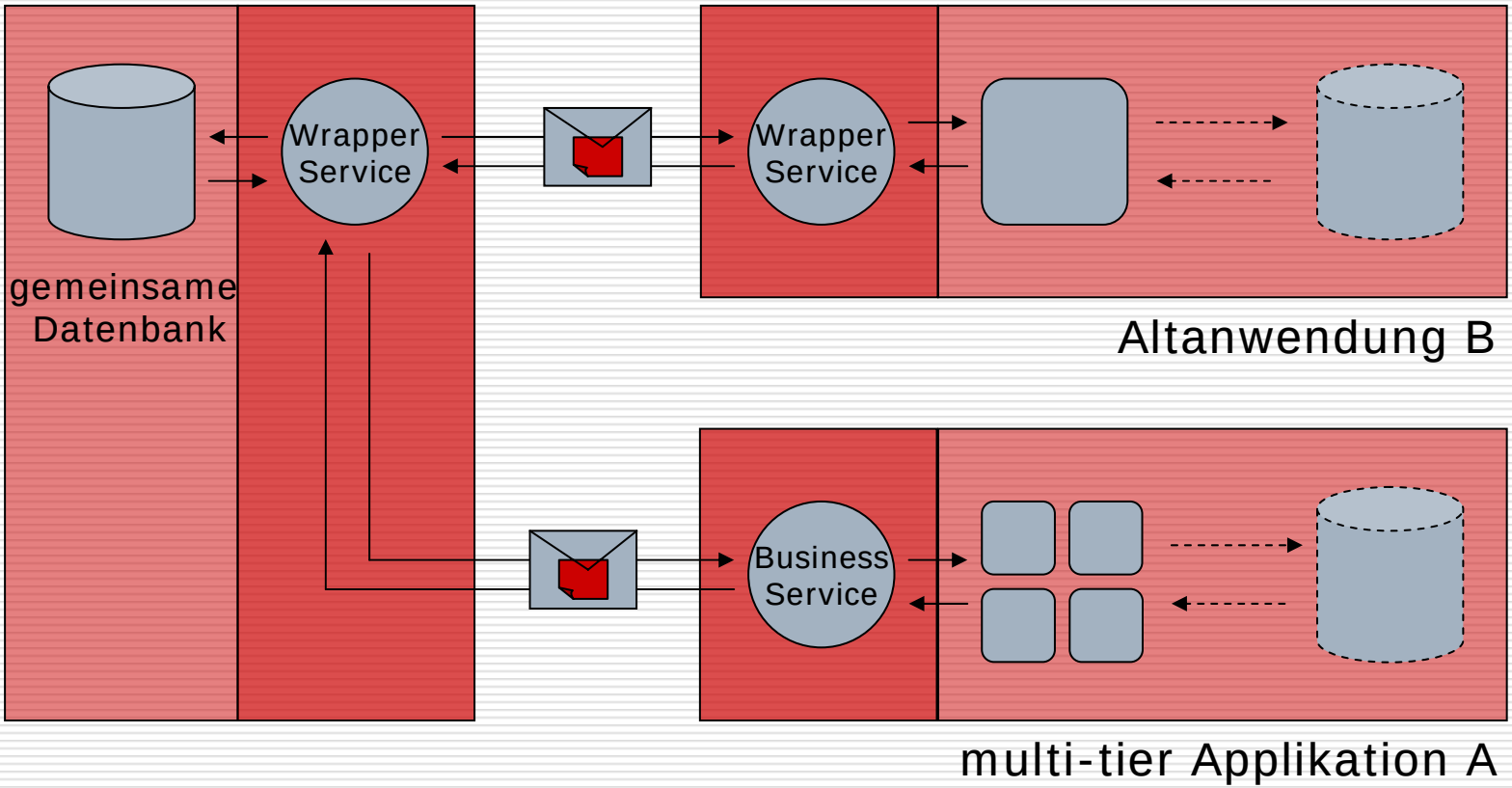


multi-tier Applikation A

Altanwendung B

Architektur für zentrale Datenbanken (3)

Web Service:



Vielen Dank für die Aufmerksamkeit

Wer hat die erste Frage?

Die Quellenangaben sind in der Ausarbeitung aufgelistet.