

- **S**ervice **O**riented
- **A**rchitecture

# Gliederung

- SOA-Grundlagen
  - Definition
  - Bestandteile
  - 10 Schritte zur SOA
- Realisierung mit Hilfe von Web Services
  - Definition
  - Standards
    - UDDI
    - WSDL
    - SOAP
- Fazit

# Definition

SOA ist ein **Managementkonzept** und setzt erst in zweiter Linie ein Systemarchitekturkonzept voraus

- Managementkonzept:  
eine an Geschäftsprozessen  
ausgerichtete Infrastruktur
  - Systemarchitektur:  
Bereitstellung fachlicher  
Dienste und Funktionalitäten  
in Form von Services
- Software-Design Ansatz
  - Technologie unabhängig
  - Kein grundlegend neuer Ansatz im Software-Design (Corba)

# Definition

- Elementarer Grundgedanken der SOA sind die Trennung der Zuständigkeiten nach fachlichen Gesichtspunkten, sowie die Kapselung technischer Details. Damit überträgt die SOA altbewährte Prinzipien der Softwarearchitektur auf die Domäne der Anwendungslandschaft.
- 
- *Informatik Spektrum 17, Okt. 2005*
-

# Charakteristische Merkmale

- Architekturmuster, das den Aufbau einer Anwendungslandschaft aus einzelnen fachlichen Bausteinen, die jeweils eine klar umrissene fachliche Aufgabe haben, beschreibt
- Bausteine sind lose gekoppelt, indem sie ihre Funktionalitäten in Form von Services anbieten => keine starken logischen oder physikalische Abhängigkeiten

# Charakteristische Merkmale

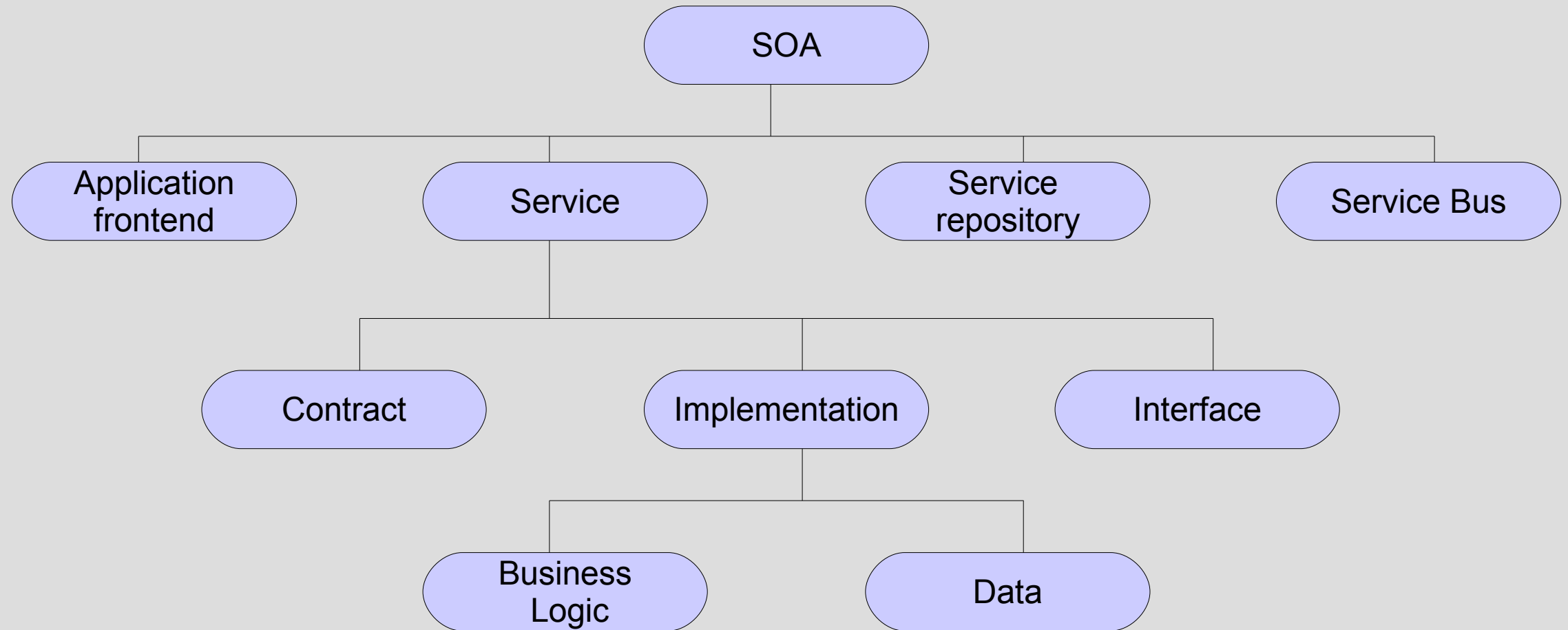
## Services/Dienste

- SOA beschreibt Funktionalitäten als modulare und wiederverwendbare Services/Dienste
- *Service*
  - fest definierte Leistung
  - Grobkörnig (Kompletten Schritt innerhalb eines Geschäftsprozesses)
  - Element eines oder mehrerer größerer Verarbeitungsabläufe
  - Verbirgt alle Implementationsdetails
  - plattformunabhängig
  - Wiederverwendbar
  - Standardisierte Schnittstellen

# Beispiel

- Online-Brokerage kann in einer SOA auf folgenden Diensten aufsetzen
  - Wertpapierdienst, der handelbare Wertpapiere nennt
  - Depotdienst, der Zugriff auf das Wertpapierdepot des Anwenders erlaubt
  - Marktdatendienst, der zu einem Wertpapier die aktuellen Börsenkurse liefert
  - Ordnerdienst, der Kauf-/Verkaufsaufträge übermittelt
  - Archivdienst, der alle Daten der Transaktion sicher archiviert
- Dienste werden aus Webanwendung aufgerufen

# Elemente einer SOA





# Hauptbestandteile einer SOA

- Service-Provider
- Service-Broker
- Service-Consumer

# Service-Provider

- Erstellt, kauft oder mietet einen Service
- Stellt den Service zur Verfügung
- Stellt Spezifikationen des Services zur Verfügung
  - Allgemeine Informationen zum Service
  - Beschreibung des Service Interface

# Service-Broker

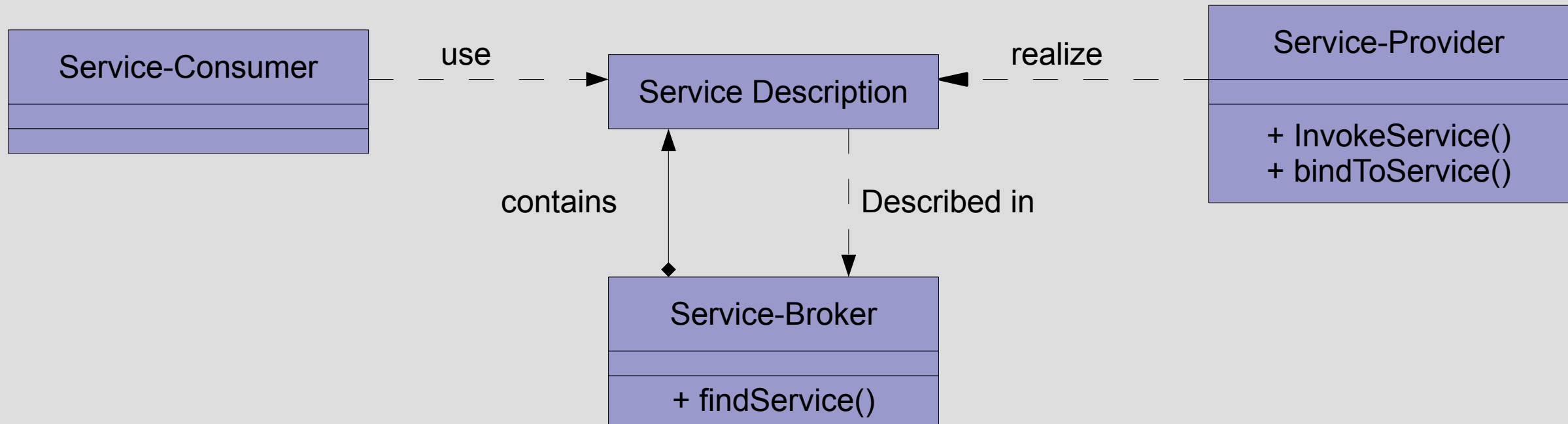
- Optional
- Verwaltet Services eines Service-Providers über die Service-Registry
- Darstellung angebotener Services meist durch ein Web-Frontend
- Hauptfunktionen:
  - Registrieren neuer Services
  - Ändern, löschen registrierter Services
  - Suche nach verfügbaren Services

# Service-Consumer

- Sucht einen Service
- Benutzt den Service

# Conceptual model of a SOA

Consumer Obtains Service description  
from Broker or directly from Provider  
(Broker is optional)



# 10 Schritte zur SOA

## Schritte 1-3

- Think big, start small
  - Mit wenigen Services anfangen
- Fachabteilungen einbinden
  - kennen sich am besten mit den Prozessen aus
- Bestandsaufnahme
  - vorhandene IT-Ressourcen in der Anfangsphase so weit wie möglich nutzen (SOA-Grundsatz), nötige Bestandsaufnahme in 2 Schritten:
    - 1.Schritt: Datenquellen und Anwendungen dokumentieren, die von der ersten SOA-Implementierung betroffen sind
    - 2.Schritt: Inventarisierung sämtlicher Hard- und Softwaresysteme

# 10 Schritte zur SOA

## Schritt 4

- Erste Services einbinden
  - redundante Logik in den beteiligten Applikationen identifizieren und diese als Services definieren
  - Beispiel: Anlegen einer Kundendatei
    - Mehrere Altanwendungen erledigen diese Funktion häufig auf unterschiedlichen Wegen
    - separater Service, den alle Applikationen gemeinsam nutzen, => Redundanzen auflösen, Wartungsaufwand vermindern

# 10 Schritte zur SOA

## Schritte 5-6

- Registry installieren
  - Anfangs reicht einfache Website, die Services auflistet
  - Steigt Anzahl an Services muss eine "echte" Registry erstellt werden
  - Grundlage für Registry bildet der Verzeichnisdienst "Universal Description, Discovery and Integration" (UDDI)
- Governance regeln
  - Governance in diesem Kontext als Kombination von Workflow-Regeln: für den Service verantwortlich?  
Qualitätsprobleme?
  - Registries dienen als Governance-Instrument der SOA-Infrastruktur (Besitzer, Versionsverwaltung, Serviceschnittstellen...)
  - Definition von Serviceschnittstellen und deren Verwaltung



# 10 Schritte zur SOA

## Schritt 7

- Sicherheit planen
  - Anbindung externer Partner als natürliche Ergänzung  
=>Vorteile aber auch Sicherheitsrisiken
  - Absicherung von XML-Messages über relativ simples Framework,  
am häufigsten benutzt:
    - WS-Security
    - SOAP
    - WSDL

# 10 Schritte zur SOA

## Schritt 8

- Messaging Infrastruktur aufbauen
  - Kleine SOA-Implementierung
    - Direkte synchrone XML-Verbindung auf Basis der SOAP-Spezifikation
  - Steigt Komplexität und Umfang, braucht man asynchrone verlässliche Messaging-Funktionen
    - Klassische EAI-Plattform
    - Enterprise Service Bus (ESB)
    - Java Messaging Service (JMS)
    - Web-Service Spezifikation WS-Reliable-Messaging

# 10 Schritte zur SOA

## Schritt 9

- Service-Management einrichten
  - Verwalten der Services
  - Überwachung von Zustand, Leistung
  - Prüfen ob Service-Levels eingehalten werden
  - Failover-Mechanismen aufsetzen
  - Mehrere Standards, z.B.
    - Web Services Distributed Management (WSDM)
    - WS-Management

# 10 Schritte zur SOA

## Schritt 10

- Services orchestrieren (aus Services größere Anwendungen zusammenfügen)
  - Unabdingbar, um der SOA-Vision von prozessgestützten "Composite Applications" näher zu kommen
  - In der Praxis nur selten verwendet, da sehr komplex
  - In diesem Kontext genannter Standard
    - Business Process Execution Language (BPEL)

# SOA Realisierung durch Web Services

Es gibt auch noch andere Technologien zur Umsetzung von SOA, aber:

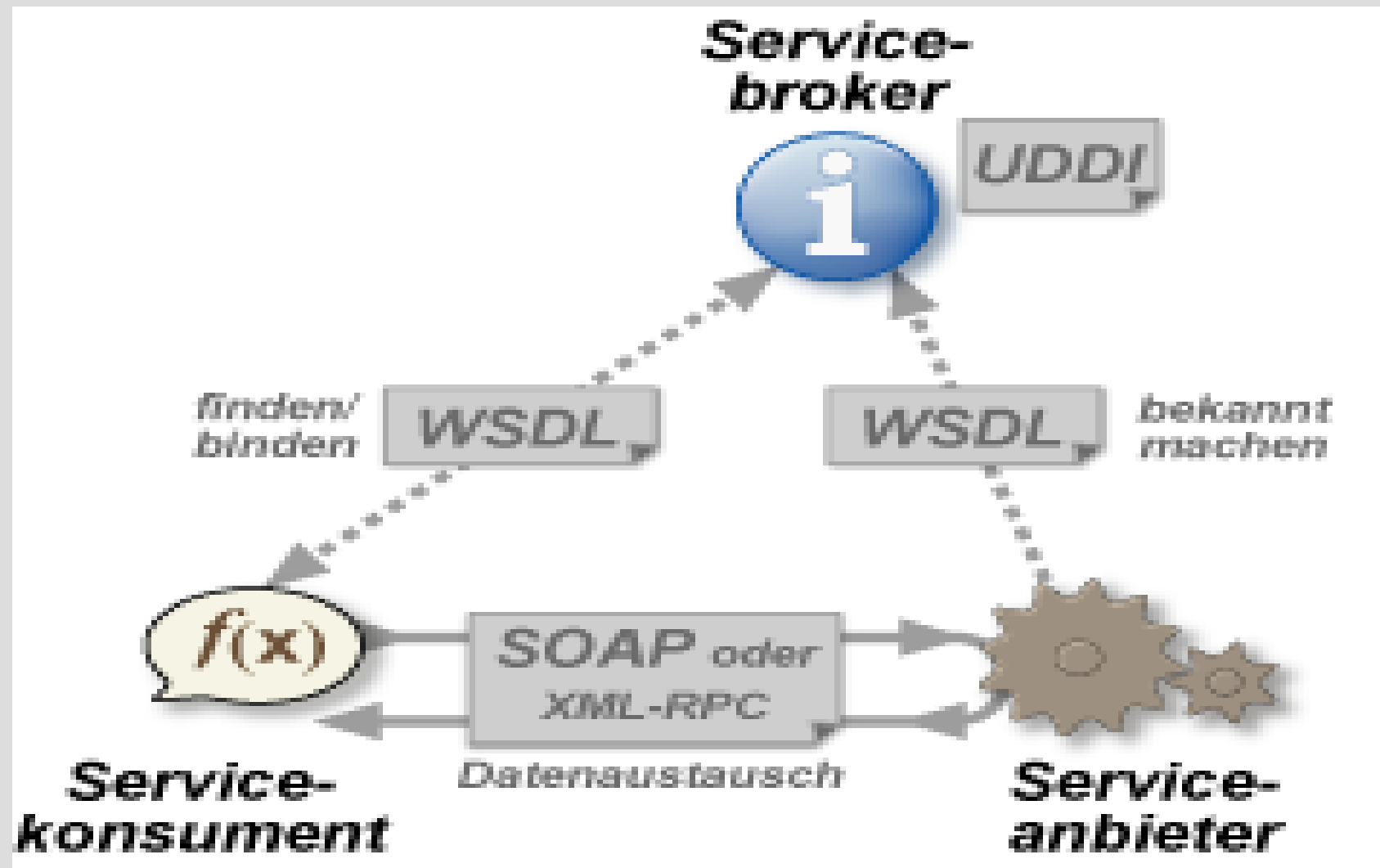
- WS werden seit Jahren als tragende Technologie zur Entwicklung einer SOA diskutiert
- WS werden von vielen als der ideale Ansatz zur Umsetzung der SOA angesehen

# SOA Realisierung durch Web Services

Was ist ein Web Service?

- Software-Anwendung
- Über URI eindeutig identifizierbar
- unterstützt direkte Interaktion mit anderen Services
- Kommunizieren untereinander mit XML-basierten Nachrichten, die über Internetprotokolle transportiert werden
- Nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und Funktionen auf entfernten Rechnern aufrufen
- 4 Standards auf XML-Basis als Grundlage
  - UDDI: als Verzeichnisdienst
  - WSDL: zur Beschreibung der Methoden und Parameter
  - SOAP (oder XML-RPC): zur Kommunikation

# Architektur von WS



# Funktionsweise von WS

- Anbieter veröffentlicht seine Dienste beim Broker
- Konsument sucht/wählt Dienst beim Broker
- Dynamische Anbindung des Konsumenten an den Anbieter  
=> Konsument greift nun auf Methoden zu
- WS bilden die 3 wichtigsten Teile der Zusammenarbeit zwischen Server und Client ab: Zusammenfinden, Binden und den Datenaustausch



# Beispiel für WS

Interaktion zwischen Fluggesellschaft und Reisebüro

- FG (Anbieter) stellen Möglichkeiten zum suchen bzw. buchen von Flügen über WS zur Verfügung
- RB (Konsument) bieten auf ihrer Webseite Flüge verschiedener FG an, von denen sie zur Laufzeit über UDDI erfahren
- Kunde kann auf Webseite des RBs Preise und Termine vergleichen und direkt buchen

# UDDI

- Universal Description, Discovery and Integration
- Verzeichnisdienst
- Dient zum veröffentlichen und auffinden von WS
- SOAP-Schnittstelle
- 3 Arten von Informationen
  - "White Pages"
  - "Yellow Pages"
  - "Green Pages"

# UDDI

## Informationsarten

- Informationen in den "White Pages"
  - Namensregister
  - Auflistung der Anbieter mit allen Detailangaben
  - Kontaktinformationen (Tel., Fax, Email...)
- Informationen in den "Yellow Pages"
  - Branchenverzeichnis
  - Spezifische Suche nach Services
  - Klassifiziert Services anhand internationaler Standards, z.B. UNSPSC
- Informationen in den "Green Pages"
  - Informationen über Geschäftsmodell des Unternehmens
  - Technische Details zu den Services
  - Auskunft über Geschäftsprozesse

# WSDL

- Web Service Description Language
- Definiert plattform-, programmiersprachen- und protokollunabhängige XML-Spezifikationen zur Beschreibung von WS
- WSDL-Dokument beinhaltet funktionelle Angaben zu:
  - Schnittstelle
  - Zugangsprotokoll und Details zum Deployment
  - Alle Informationen zum Zugriff auf den Service, in maschinenlesbarer Form

# WSDL

## Hauptelemente

- <definitions>: root-Element
- <types>
  - Definition der Datentypen (XML-Schema Syntax zur Definition)
- <messages>
  - Def. der übertragenen Daten
  - Ein oder mehrere logische Teile
- <binding>
  - Definiert Message-Format und Protokolldetails
- <ports>
  - Spezifiziert eine Adresse für eine Bindung, URI
- <service>
  - Fassen eine Menge von eventuellen Ports zusammen

# WSDL

## Hauptelemente

- <portType>:
  - Wichtigste WSDL-Element
  - Definiert einen Web Service, seine Operationen und die Messages
  - Operation-Types
    - One-Way: Input-Message vom Client
    - Request-Response: Input-Message vom Client => Service sendet Antwort (Output-Message)
    - Solicit-Response: Service sendet Message und erwartet Antwort vom Client
    - Notification: Server sendet Output-Message

# WSDL

## Struktur

```
<definitions>  
  <types>  
    definition of types.....  
  </types>  
  <message>  
    definition of a message.....  
  </message>  
  <portType>  
    definition of a portType.....  
  </portType>  
  <binding>  
    definition of a binding.....  
  </binding>  
  <service>  
    definition of a service.....  
  </service>  
</definitions>
```

# WSDL

## Beispiel

Service der nach Erhalt eines Aktiennamens den aktuellen Wert zurückgibt

```
<definitions name="StockQuote"  
  targetNamespace="http://example.com/stockquote.wsdl"  
  xmlns:tns="http://example.com/stockquote.wsdl"  
  xmlns:xsd1="http://example.com/stockquote.xsd"  
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
  xmlns="http://schemas.xmlsoap.org/wsdl/">  
  .  
  .  
  .  
  .
```



# WSDL

## Beispiel

```
.....  
<types>  
  <schema targetNamespace="http://example.com/stockquote.xsd"  
    xmlns="http://www.w3.org/2000/10/XMLSchema">  
    <element name="TradePriceRequest">  
      <complexType>  
        <all>  
          <element name="tickerSymbol" type="string"/>  
        </all>  
      </complexType>  
    </element>  
    <element name="TradePrice">  
      <complexType>  
        <all>  
          <element name="price" type="float"/>  
        </all>  
      </complexType>  
    </element>  
  </schema>  
</types>  
.....
```

# WSDL

## Beispiel

.....

```
<message name="GetLastTradePriceInput">  
  <part name="body" element="xsd1:TradePriceRequest"/>  
</message>
```

```
<message name="GetLastTradePriceOutput">  
  <part name="body" element="xsd1:TradePrice"/>  
</message>
```

```
<portType name="StockQuotePortType">  
  <operation name="GetLastTradePrice">  
    <input message="tns:GetLastTradePriceInput"/>  
    <output message="tns:GetLastTradePriceOutput"/>  
  </operation>  
</portType>
```

.....

# WSDL

## Beispiel

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

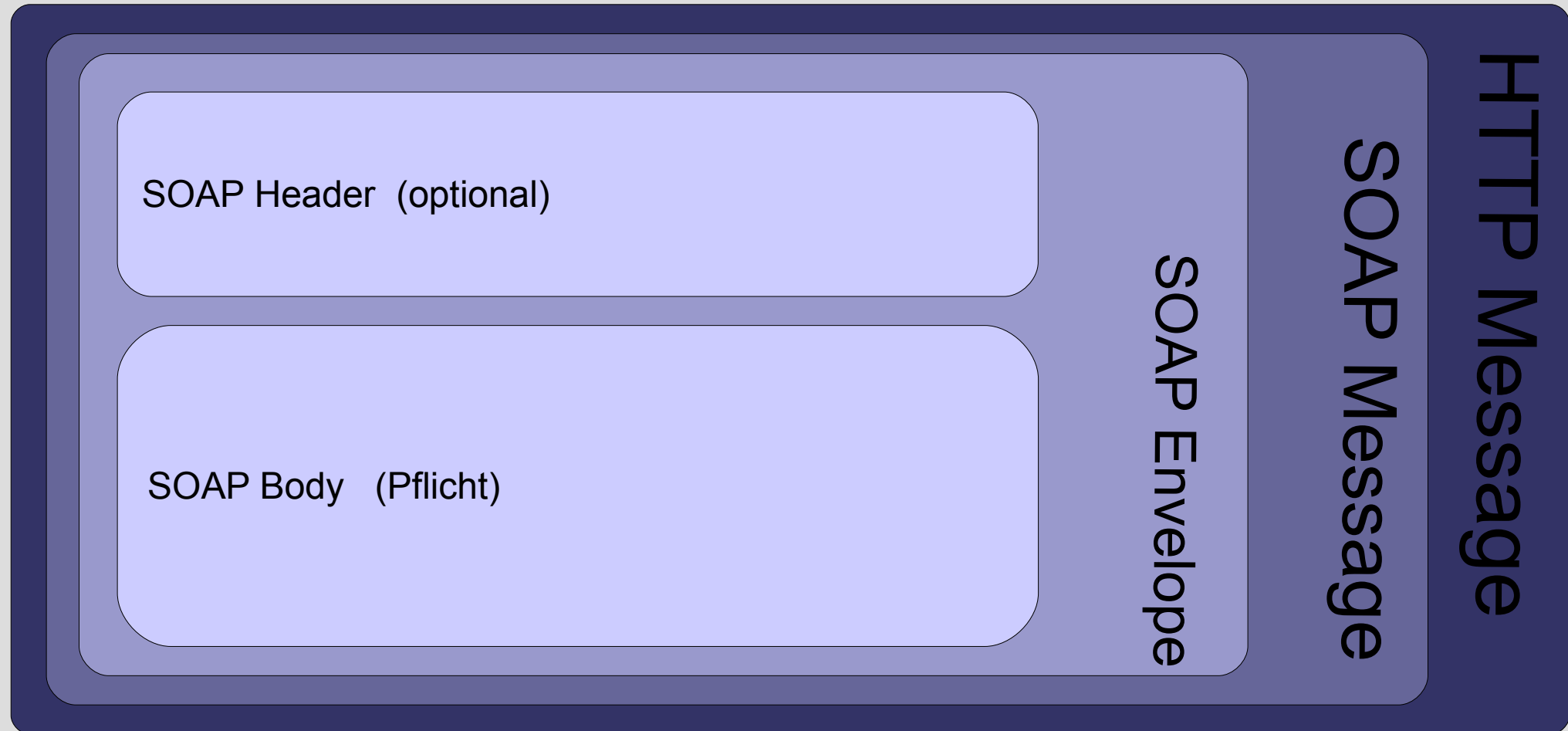
</definitions>
```

# SOAP

- Simple Object Access Protocol
- Internet-/Kommunikationsprotokoll
- XML-basierte, strukturierte und typisierte Informationen
- Plattform- und programmiersprachenunabhängig
- Zur direkten Kommunikation zwischen Anwendungen
- Meist HTTP/TCP zum Transport (Kompatibilität zu gängigen Netzwerk-Architekturen z.B. Firewalls; HTTPS)
- Nachricht wird in Container verschickt, der SOAP-Envelope
- W3C Empfehlung

# SOAP

## Aufbau einer Nachricht



Header: Metainformationen (zum Routing, zur Verschlüsselung....)

Body: Nutzdaten

# SOAP

## Beispiel

GetStockPriceRequest Anfrage wird an den Server gesendet, Anfrage hat StockName Parameter, ein Price Parameter wird zurückgegeben

Skelett einer SOAP Message:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    ...
    <soap:Fault>
      ...
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

# SOAP

## Beispiel: SOAP Request

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
```

} HTTP Header

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

# SOAP

## Beispiel: SOAP Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
```

} HTTP Header

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```



- **Fazit**

# Fazit

## Vorteile SOA

- IT-Kosten senken
- Wiederverwendbarkeit
- Flexibilität, durch Neukombination bestehender Services
- Reduktion der Komplexität, durch Kapselung der Implementationsdetails hinter Schnittstellen

# Fazit

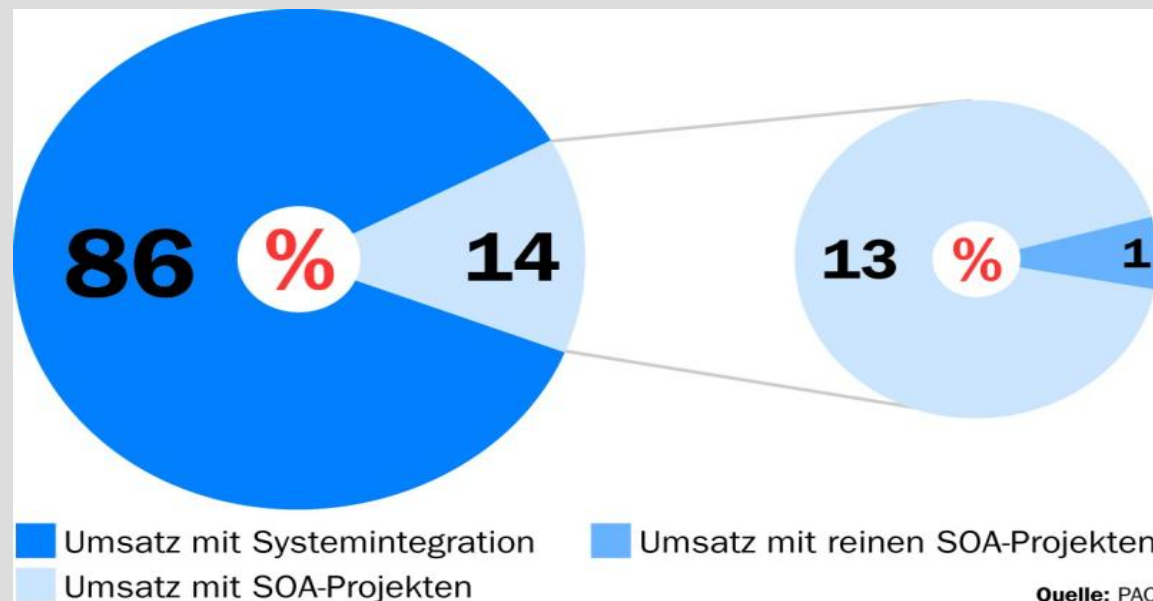
## Schwierigkeiten SOA

- Vorteile nur, wenn Geschäftsmodell und -prozesse service-orientiert sind
- Mangel an IT-Spezialisten mit fundiertem betriebswirtschaftlichen Know-How
- Vorteile entwickeln sich meist erst langfristig

# Fazit

## aktuelle Entwicklung

- Weltweit 8,6 Milliarden Dollar in SOA-Projekte im letzten Jahr
- Bis 2010 bis zu 33,8 Milliarden Dollar
- Allerdings:  
"SOA verbreitet sich im Schneckentempo"  
(Studie [www.computerwoche.de](http://www.computerwoche.de))



# Literatur und Quellen

- <http://de.wikipedia.org>
- <http://www.computerwoche.de>
- <http://www.w3schools.com>
- Informatik Spektrum, Oct. 7 2005
- Erl, Thomas: Service oriented architecture – Concepts, technologie and designs, Pearson Education, USA 2005
- Wolfgang Herrmann: In zehn Schritten zur SOA. Computerwoche 07.12.2005
- IBM White Paper, Service-oriented modeling and architecture, [www.DeveloperWorks.com](http://www.DeveloperWorks.com)