

Web-Programmierung (WPR)

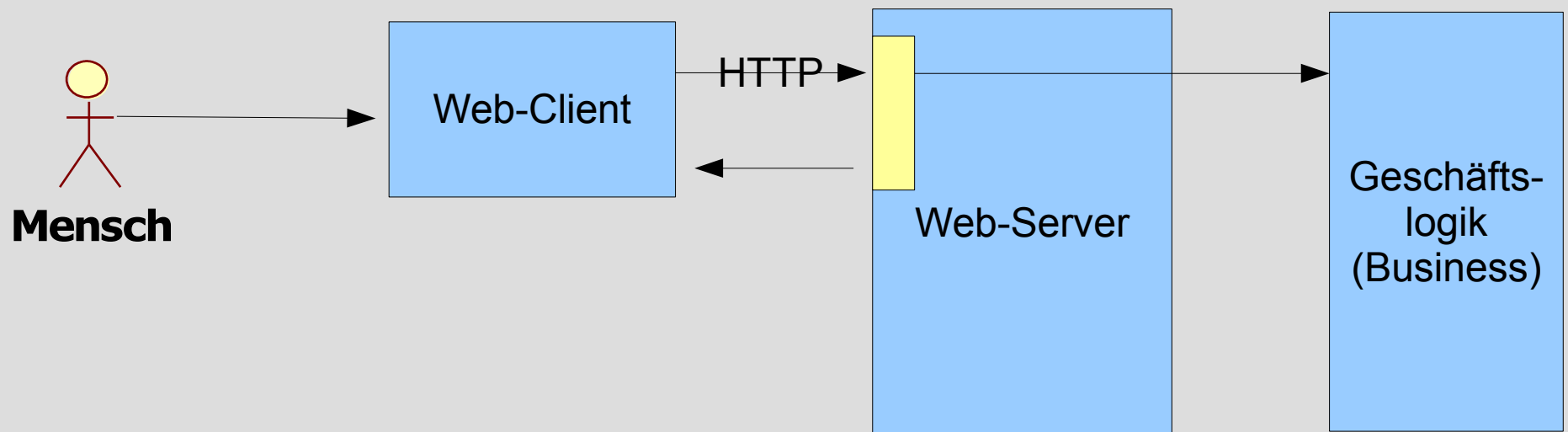
Vorlesung VIII.

Web Services

<mailto:wpr@gruner.org>

21 Web Service

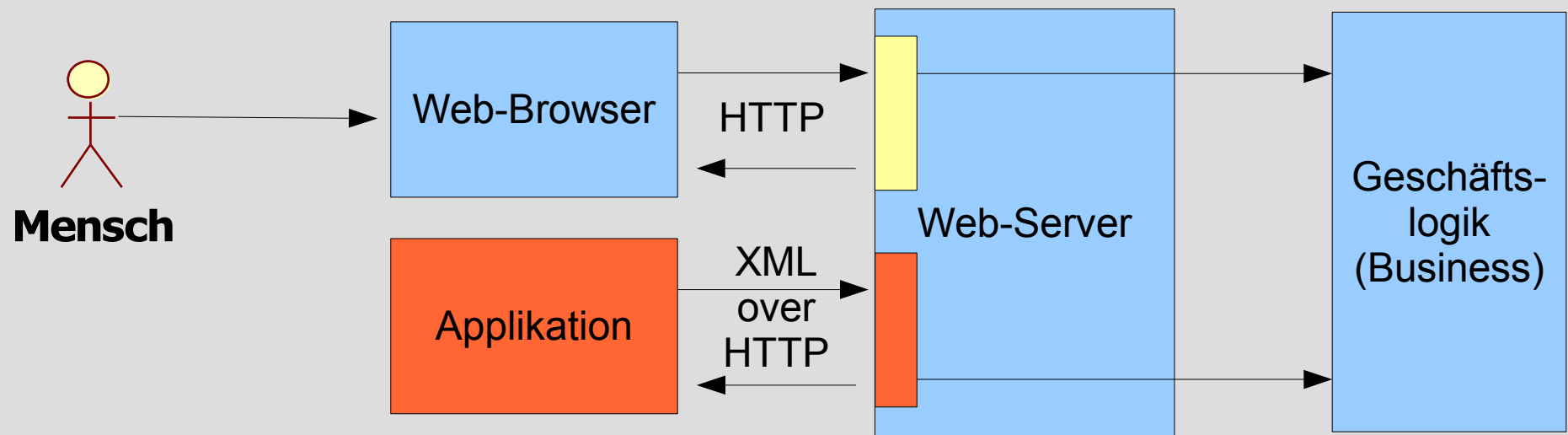
- World Wide Web seit Anfang 1990er Jahren



Personenorientiertes Web

21 Web Service

- World Wide Web seit Anfang 1990er Jahren



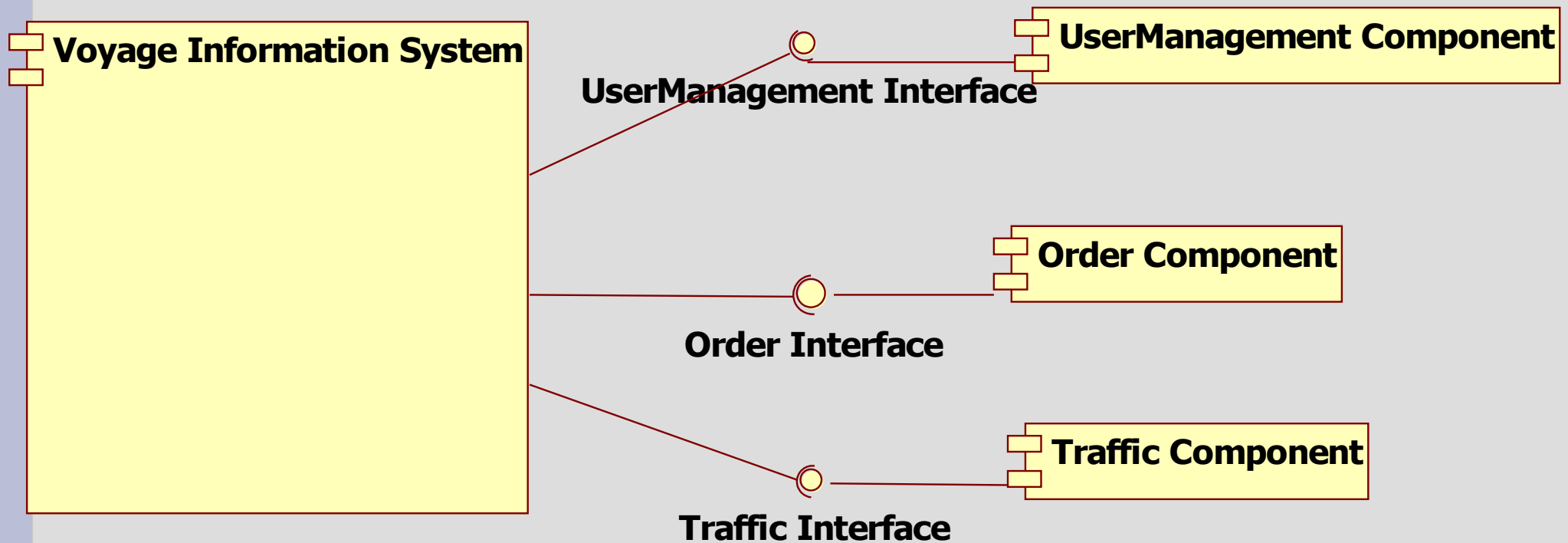
Applikationszentriertes Web

21.1 Grundlagen

- Softwaresysteme
=> verteilte Komponenten
- Monolithe => Komponenten mit Schnittstellen
- Komponentennutzung über Rechengrenzen
- Neue Geschäftsfunktionen
u.a. basierend auf exist. Komponenten
- Komponenten werden geteilt.

21.1 Grundlagen

- Komponentenverteilung



21.1 Grundlagen

- **Komponentenmodell**
 - Libraries
 - shared.lib
 - DLL-Dateien
 - Jar-Dateien
- **Verteilte Komponentenmodelle**
 - Objektmodelle
 - DCOM (Distributed Common Object Model)
 - CORBA (unabhängig)

21.1 Grundlagen

- Verteilte Komponentenmodelle
 - Anforderungen
 - Unabhängig vom Objektmodell
 - Kommunikation über Internet/HTTP
 - Informationsaustausch über Nachrichten
 - Universelle Sprache für Nachrichten notwendig
 - Lösung
 - Austausch von Nachrichten in XML-Format
 - Schnittstellendefinition enthält Methodennamen und Eingangsparameter, Rückgabeparameter (Typen)

21.1 Grundlagen

- Web-Service
 - ♦ Abgeschlossene Einheit, die definierten Funktionsumfang über eine Nachrichtenbasierte Schnittstelle zur Verfügung stellt.
 - ♦ Black-Box System
 - ♦ Veröffentlicht Schnittstelle
 - ♦ Nachrichtentransport über Internet Protokolle (z.B.: http, https)
 - ♦ Keine Unterscheidung zwischen internen u. Externen Services

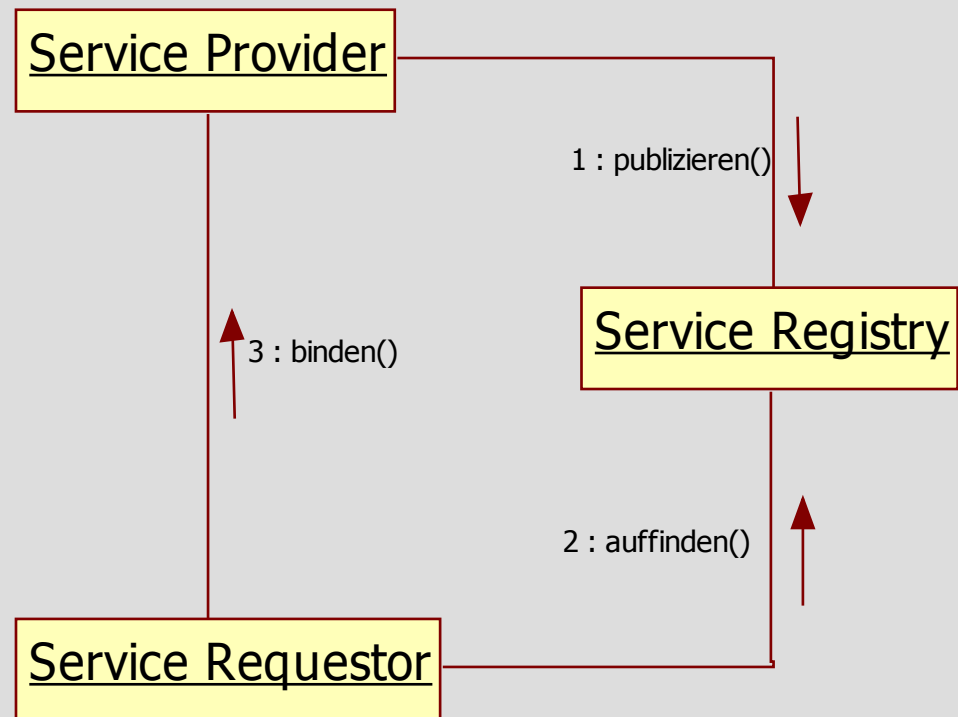
21.1 Grundlagen

- Service-orientierte Architektur (SOA)
 - ♦ Neues Verteiltes Komponentenmodell
 - ♦ Zentraler Baustein: Service
 - ♦ Serviceaufruf erfolgt in einem Schritt
 - ♦ Services werden „stateless“ implementiert (Skalierbarkeit)
 - ♦ Services werden zu Prozessen verknüpft
 - ♦ SOA = geschäftsorientierte Sicht auf IT

21.1 Grundlagen

- Bausteine und Standards
SOA definiert 3 Rollen

- ◆ Service Provider
- ◆ Service Requestor
- ◆ Service Registry



21.1 Grundlagen

- Standards
 - SOAP für Nachrichtenformat
 - WSDL für Beschreibung eines Web Services
 - Ähnlich wie CORBA-IDL
 - Auf Basis von XML
 - => Generierung von Service-Client
 - UDDI für das Publizieren und Auffinden
 - Universal Description, Discovery, and Integration
 - Enthält:
 - Informationen zum Unternehmen, der Service bereitstellt
 - Informationen zu angebotenen Services
 - Technische Spezifikation (Internet-Adresse einer WSDL-Datei)

21.1 Grundlagen

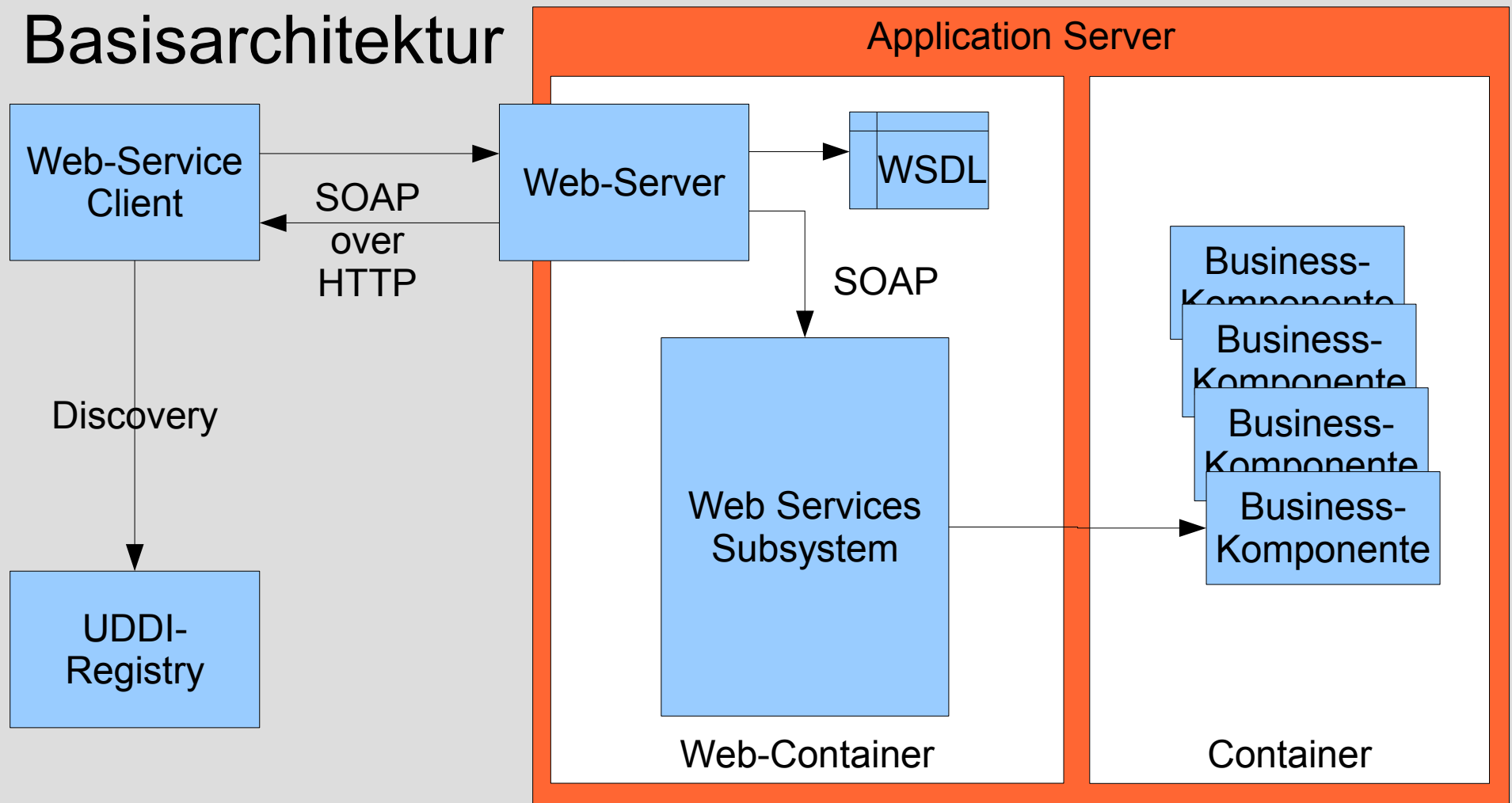
- Kommunikationsmodelle
 - ◆ RPC-orientierte Kommunikation („call and wait“)
 - ◆ Message-orientierte Kommunikation („fire and forget“)

21.1 Grundlagen

- Anwendungsmodelle für Web-Services
 - ◆ RPC
(synchron)
 - ◆ Dokumentenaustausch
(synchron od. asynchron)

21.1 Grundlagen

Basisarchitektur

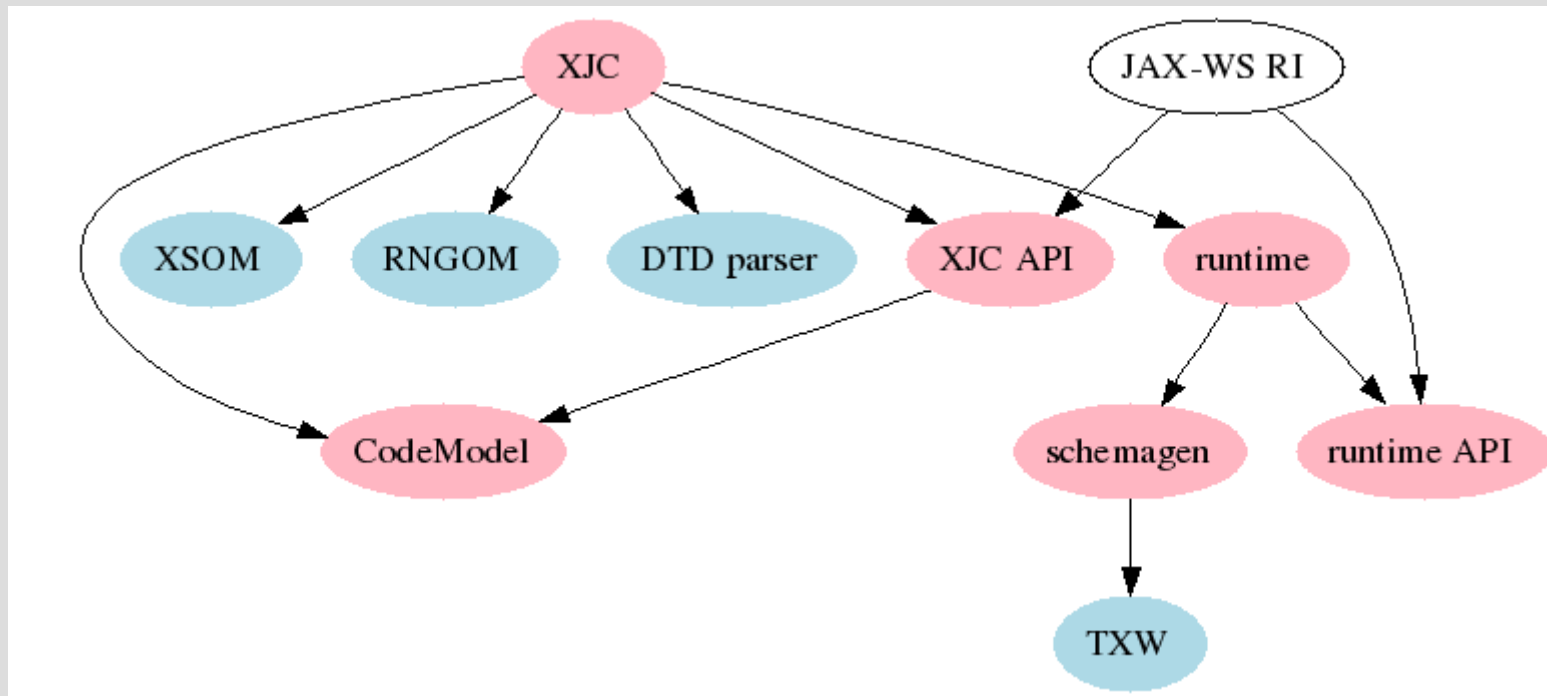


21.1 Grundlagen

- Web Service Subsystem
 - XML-Technologien
 - Parsen u. Validieren des Request Dokumentes
 - Transformieren des Dokumentes in gewünschtes Zielformat
 - Abbildung komplexer Datenstrukturen auf Objekte der Zielsprache
 - Abbildung von Objekten der Zielsprache auf XML-Strukturen
 - Generierung des Response Dokumentes

21.1 Grundlagen

- Java Architecture for XML Binding (JAXB)

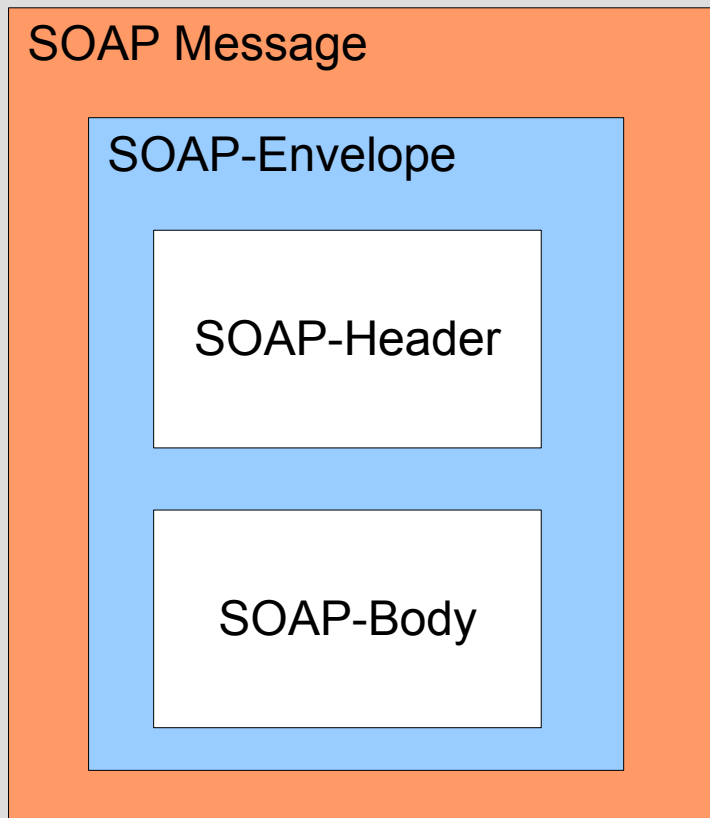


21.2 SOAP

- Simple Object Access Protocol (Alternative zu XML-RPC)
- Findet breite Unterstützung in Industrie
- IBM -> Apache -> Apache Axis
- Plattformunabhängig + XML-basiert
- W3C Standard seit 2000 (Version 1.1)
- SOAP-Spezifikation
 - Message-Syntax
 - Regeln für Codierung von Datentypen
- Aktuelle Version 1.2
- ca. 60 Implementierungen (Stand 2004)

21.2 SOAP

- Nachrichtenformat



- ♦ SOAP-Body (RPC)
 - Service
 - Methode
 - Parameter
- ♦ DOCUMENT
 - Beliebiges XML-Dokument

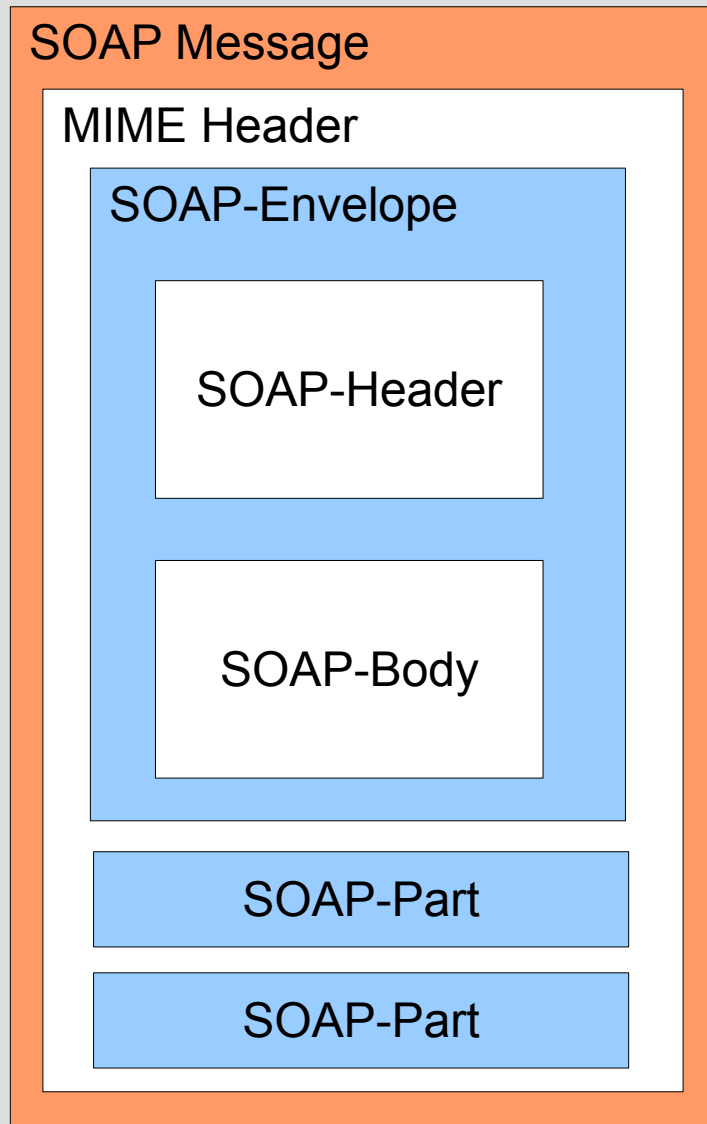
21.2 SOAP

☰ 21.2 SOAP

```
...  
<xsd:element name="Envelope" type="tns:Envelope"/>  
<xsd:complexType name="Envelope">  
<xsd:sequence>  
  <xsd:element ref="tns:Header" minOccurs="0"/>  
  <xsd:element ref="tns:Body" minOccurs="1"/>  
</xsd:sequence>  
</xsd:complexType>  
.....
```

21.2 SOAP

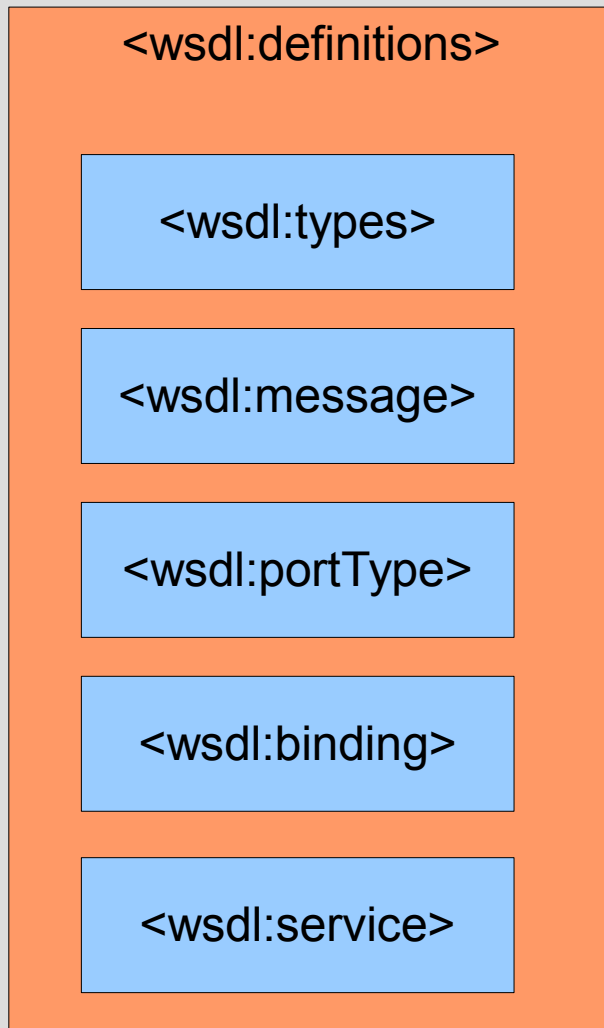
- SOAP-Message mit Attachments



21.3 WSDL

- Web Service Description Language
 - ◆ Öffentliche Methoden mit Signaturen
 - ◆ Unterstützte Transport Protokolle
 - ◆ Adresse des Web Services
 - ◆ Der Typ-Schemata für den Datenaustausch
- => Schnittstellenbeschreibung (IDL)

21.3 WSDL



- `wSDL:definitions` (Namensräume)

```
<wSDL:definitions
  xmlns:wSDL="..."
  targetNamespace="..."
  xmlns:wSDLsoap="..."
  xmlns:soapenc="..."
  >
</wSDL:definitions>
```

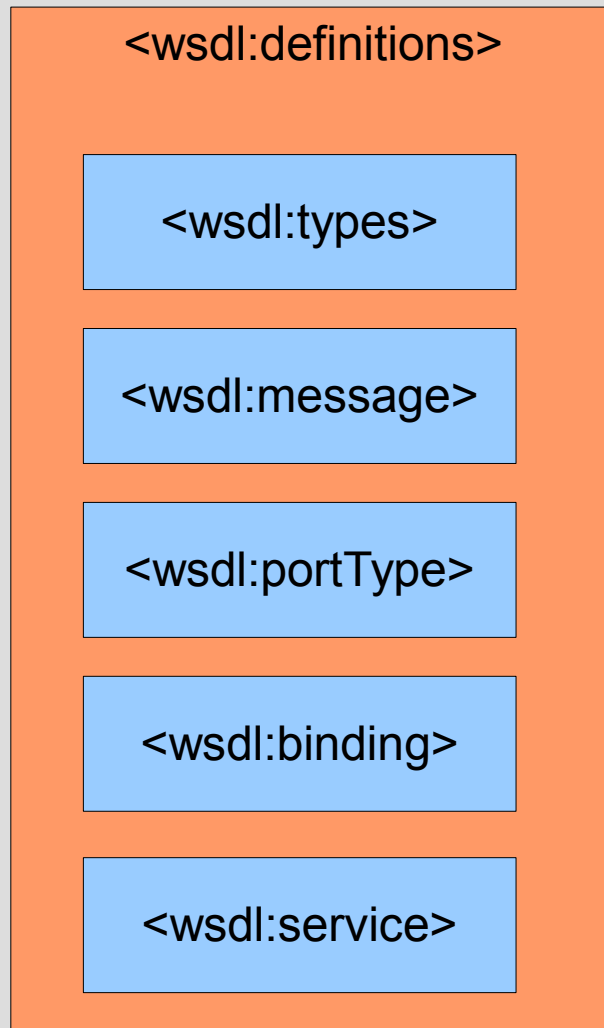
21.3 WSDL



- **wSDL:types**

```
<wSDL:types>  
  <xs:schema>  
    <xs:element name="orderItem">  
      <xs:complexType>  
        <xs:sequence>  
          <xs:element ref="description"/>  
        </xs:sequence>  
        <xs:attribute name="amount"  
          use="required" type="xs:integer"/>  
        <xs:attribute name="orderId"  
          use="required" type="xs:integer"/>  
      </xs:complexType>  
    </xs:element>  
  </xs:schema>  
</wSDL:types>
```

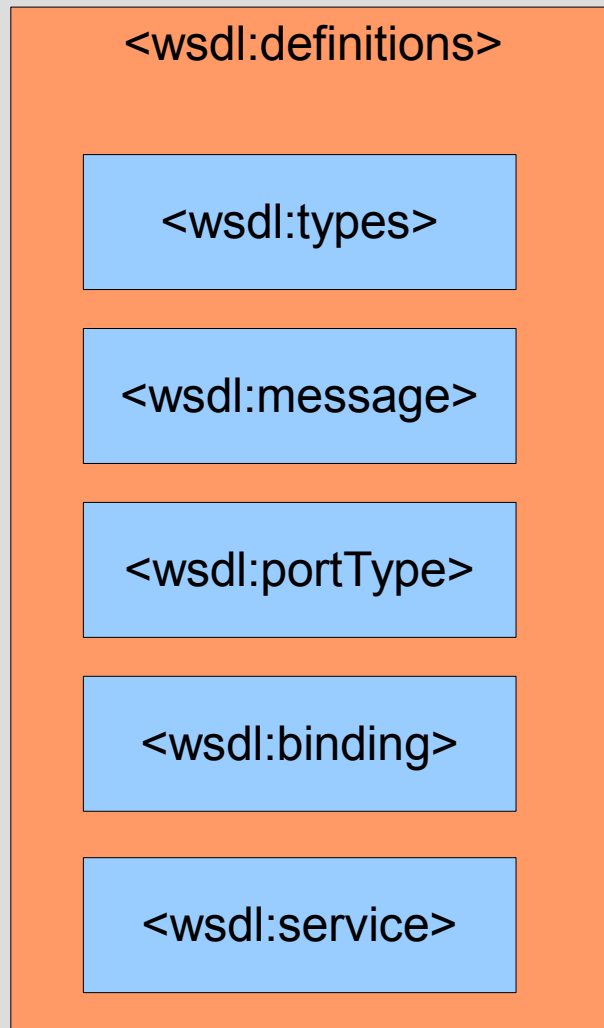
21.3 WSDL



- `wsdl:message` (Nachrichten)

```
<wsdl:message
  name="getQuoteReq">
  <wsdl:part name="id"
    type="xsd:string"/>
</wsdl:message>
```


21.3 WSDL



- **wsdl:portTypes**
abstrakte Webservice
Schnittstelle

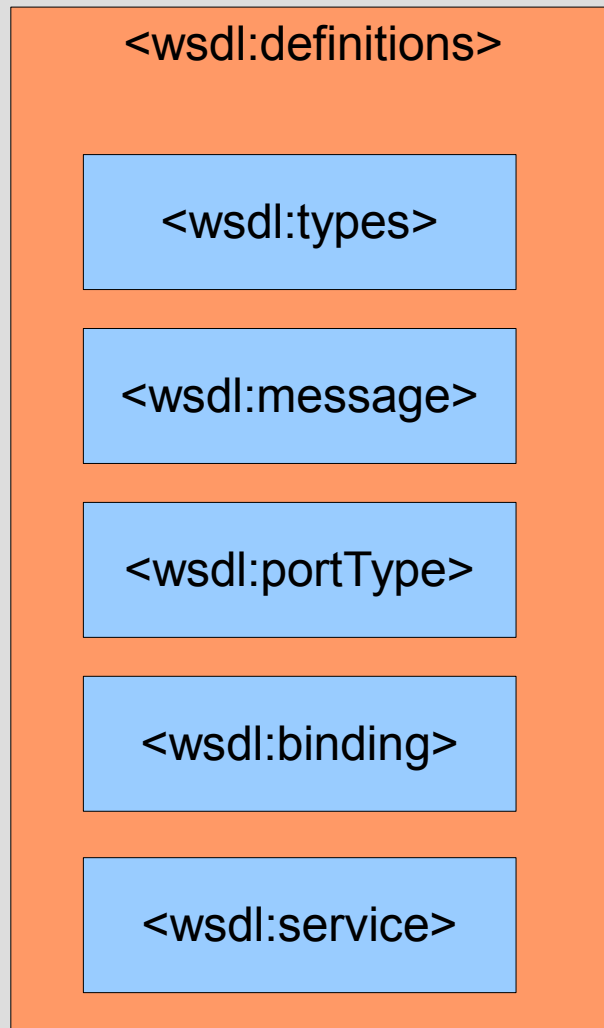
```
<wsdl:portType  
  name="CalculatePort">  
  <wsdl:operation  
    name="addValue">  
    <wsdl:input/>  
    <wsdl:output/>  
  </wsdl:operation>  
</wsdl:portType>
```

21.3 WSDL

- **wSDL:portTypes** –
 - Abstrakte Schnittstelle
 - 1..n Operationen

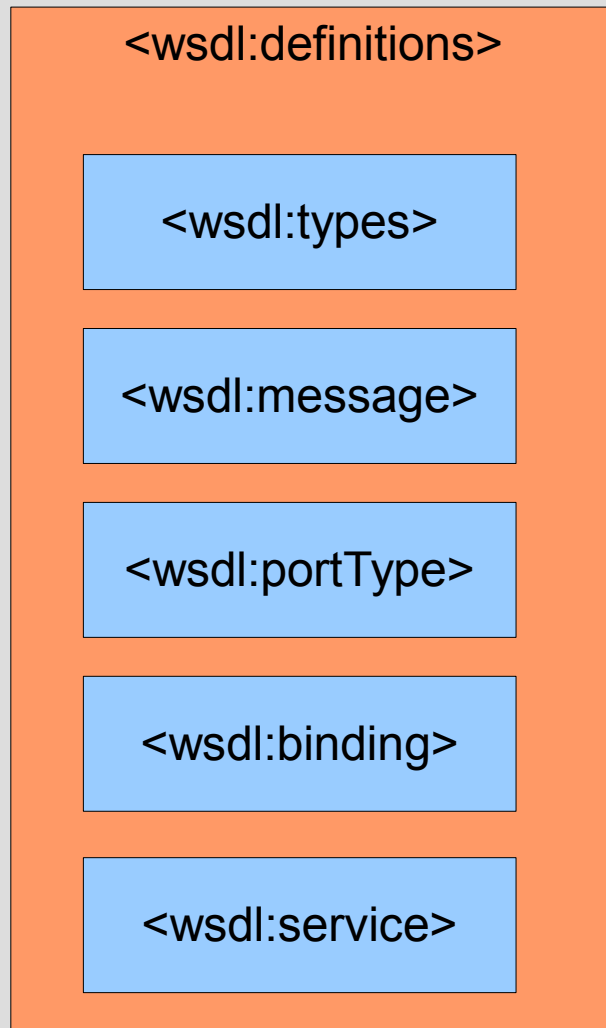
Operation	Beschreibung
One-Way	Client sendet Nachricht ohne Response zu erwarten
Request-Response	Client sendet Nachricht an Service und dieser reagiert mit Response
Solicit-Response	Service sendet Nachricht an Client und bekommt von diesem eine Response
Notification	Der Webservice sendet Nachricht an den Client

21.3 WSDL



- **wSDL:binding**
Bindung an Protokoll
Nachrichtenprotokoll
- ```
<wSDL:binding
 name="CalculatePort">
 <wSDL:soapbinding
 style="rpc"
 transport="http:..">
 <wSDL:operation>
 </wSDL:operation>
 </wSDL:soapbinding>
</wSDL:binding>
```

## 21.3 WSDL



- `wsdl:service`

```
<wsdl:service name=
 "CalculateService">
 <wsdl:port name="">
 <wsdl:address
 location="http://.">
 </wsdl:address>
 </wsdl:port>
</wsdl:service>
```