

# Web-Programmierung (WPR)

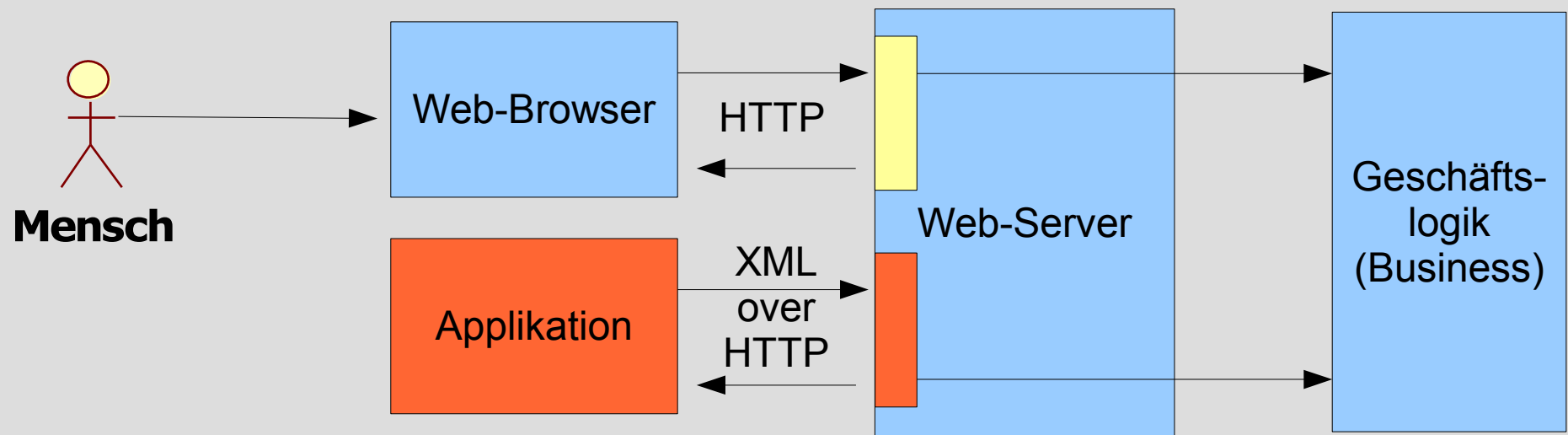
Vorlesung X.

Web Services  
Teil 2

<mailto:wpr@gruner.org>

# 21 Web Service

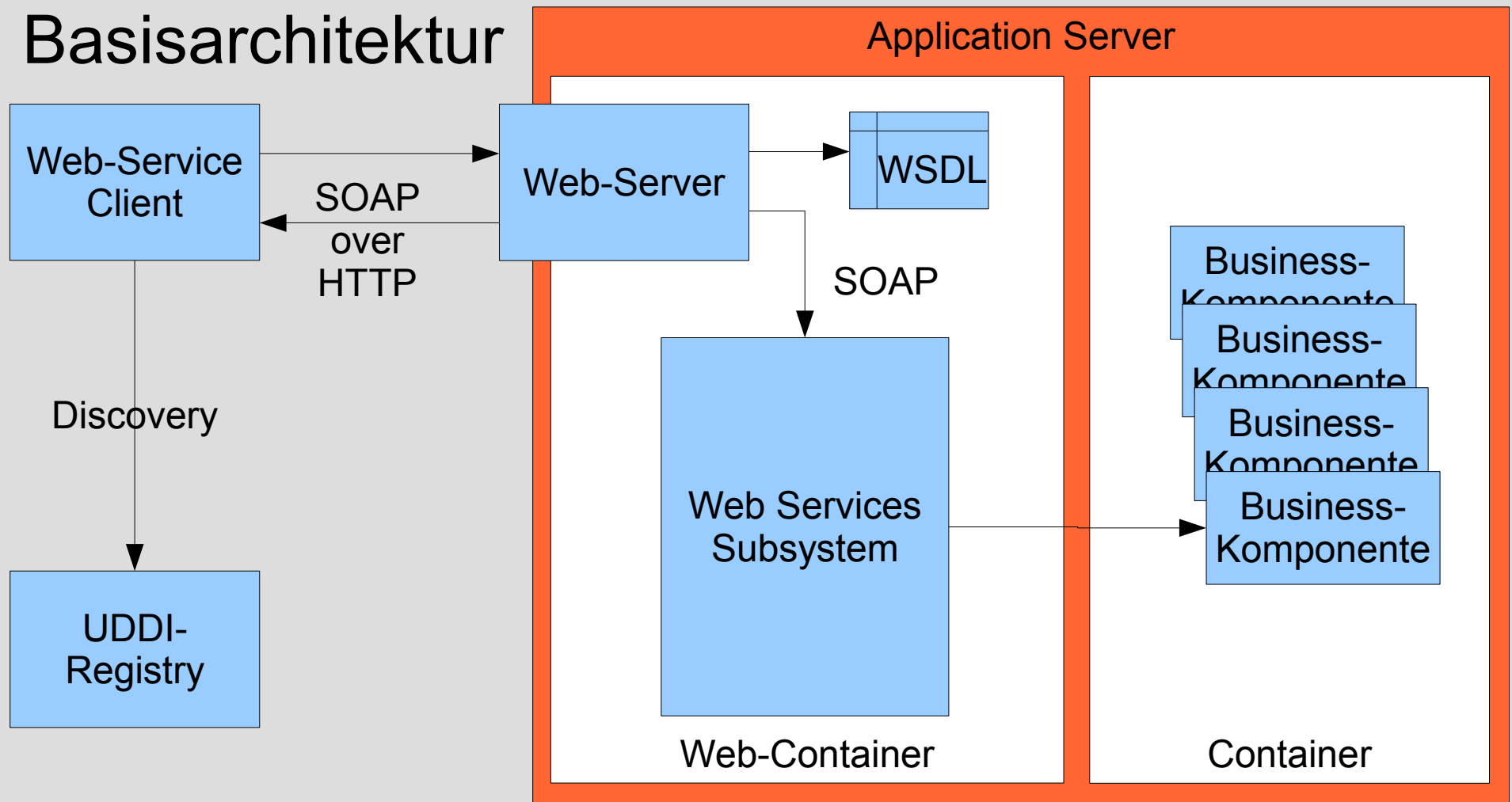
- World Wide Web seit Anfang 1990er Jahren



Applikationszentriertes Web

# 21.1 Grundlagen

## Basisarchitektur



## 21.1 XML-RPC

- Remote Method Invocation via XML

### XML-RPC-Definition

## 21.1 XML-RPC

- **Alternative XML-RPC** <http://www.xmlrpc.com>

```
<?xml version="1.0"?>
<methodCall>
  <methodName>warenkorb.addPosition
  </methodName>
  <params>
    <param>
      <value>Monitor</value>
    </param>
    <param>
      <value><int>10</int></value>
    </param>
  </params>
</methodCall>
```

## 21.1 XML-RPC

- XML-RPC Response

```
<?xml version="1.0"?>
<methodResponse>
  <fault><value><struct>
    <member>
      <name>faultCode</name>
      <value><int>4</int></value>
    </member>
    <member>
      <name>faultString</name>
      <value><string>Too many params. </string>
    </value>
    </member>
  </struct></value></fault>
</methodResponse>
```

## 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

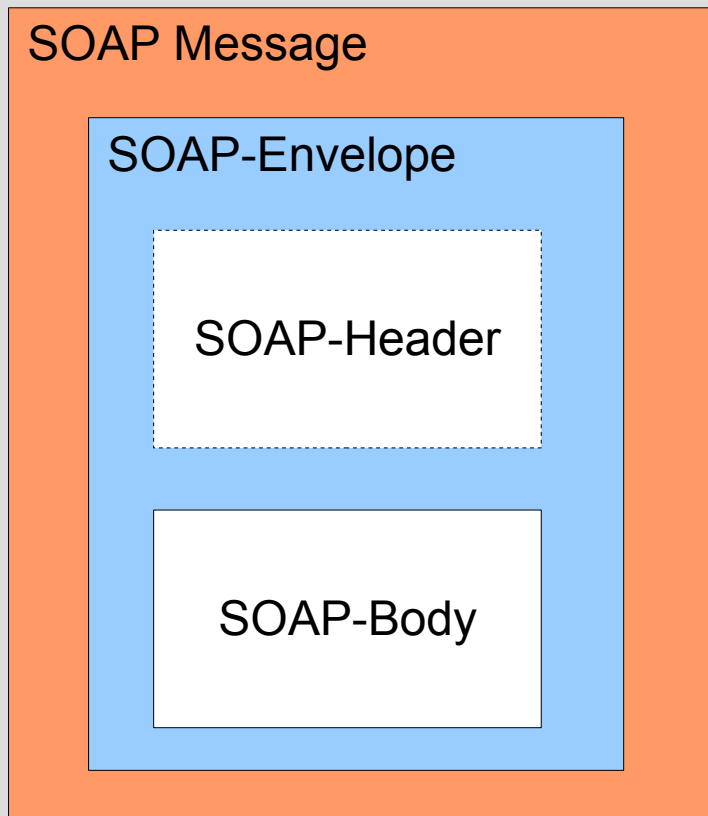
### ☰ 21.2 SOAP

```
...  
<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

- Nachrichtenformat



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

## 21.2 SOAP

- Beispiel:

```
double getQuote (String id) ;
```

### Request

```
<soapenv:Envelope ...>  
  <soapenv:Body>  
    <ns1:getQuote xmlns:ns1="http...">  
      <id xsi:type="xsd:string">401</id>  
    </ns1>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Missing Namespaces:

- `xmlns:soapenv`, `xmlns:xsd`, `xmlns:xsi`

## 21.2 SOAP

- Beispiel:

```
double getQuote (String id) ;
```

### Response

```
<soapenv:Envelope ...>  
  <soapenv:Body>  
    <ns1:getQuoteResponse xmlns:ns1="..">  
      <ns1:quoteReturn xsi:type="xsd:double">  
        19.5</ns1:quoteReturn>  
    </ns1:getQuoteResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Missing Namespaces:

- xmlns:soapenv, xmlns:xsd, xmlns:xsi

## 21.2 SOAP

- Beispiel:

```
double getQuote (String id) ;
```

### Response (bei Fehler)

```
<soapenv:Envelope ...>  
  <soapenv:Body>  
    <soapenv:Fault>  
      <faultcode>Server.error</faultcode>  
      <faultstring>no such op.</faultstring>  
    </soapenv:Fault>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Missing Namespaces:

- xmlns:soapenv, xmlns:xsd, xmlns:xsi

## 21.2 SOAP

- Transport
  - ♦ HTTP, SMTP, FTP , ...
  - ♦ Auswahl => **Binding**
- HTTP-Übertragung
  - ♦ HTTP-Request => SOAP-Request
  - ♦ HTTP-Response => SOAP-Response

## 21.2 SOAP

- HTTP-Header SOAP Request

```
POST /axis/services/vinoExchange HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, multipart-
    related, text/*
Host:127.0.0.1
SOAPAction: ""
Content-Length: 449
<?xml..>
<soapEnvelope>
    . . . .
</soapEnvelope>
```

## 21.2 SOAP

- HTTP-Header SOAP Response

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
```

```
Date: . . .
```

```
Server: Apache Coyote/1.0
```

```
Connection: close
```

```
<?xml..>
```

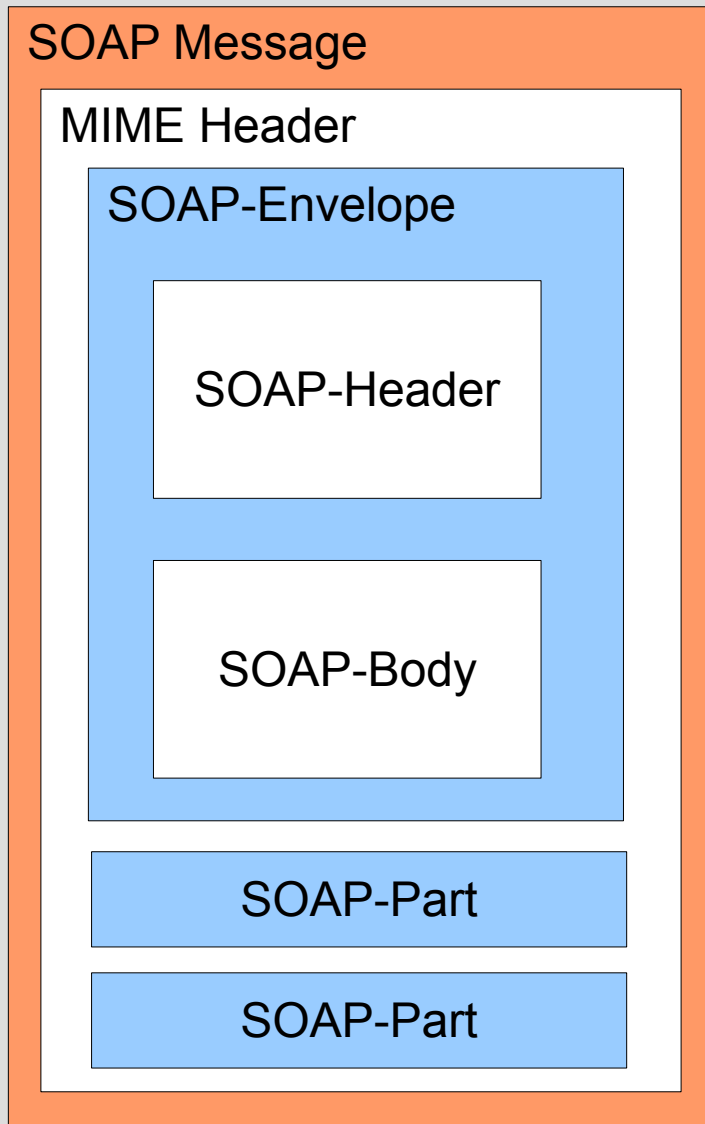
```
<soapEnvelope>
```

```
. . . .
```

```
</soapEnvelope>
```

## 21.2 SOAP

- SOAP-Message mit Attachments





## 21.2 SOAP

- HTTP-Header SOAP Request

```
POST /soap/servlet/messagerouter HTTP/1.0
```

```
Content-Type: multipart/related;
```

```
    boundary: "----=_Part_0_747212.10269376";
```

```
SOAPAction: "urn:vinoexchange:interest"
```

```
----=_Part_0_747212.10269376
```

```
Content-Type: text/xml; charset=utf-8
```

```
<soapEnvelope> ...
```

```
</soapEnvelope>
```

```
----=_Part_0_747212.10269376
```

```
Content-Type: image/gif
```

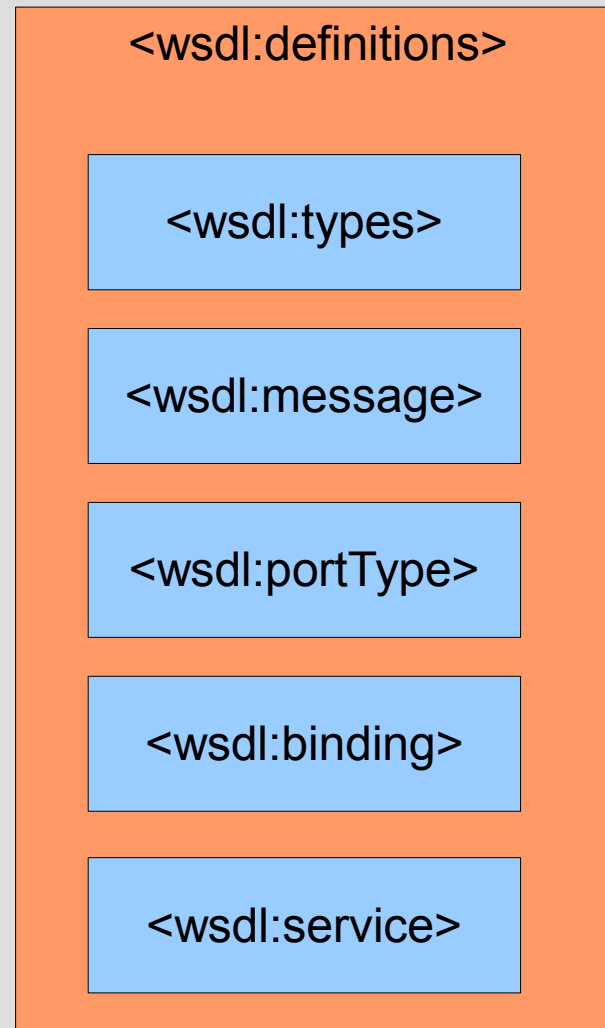
```
.....
```

```
----=_Part_0_747212.10269376
```

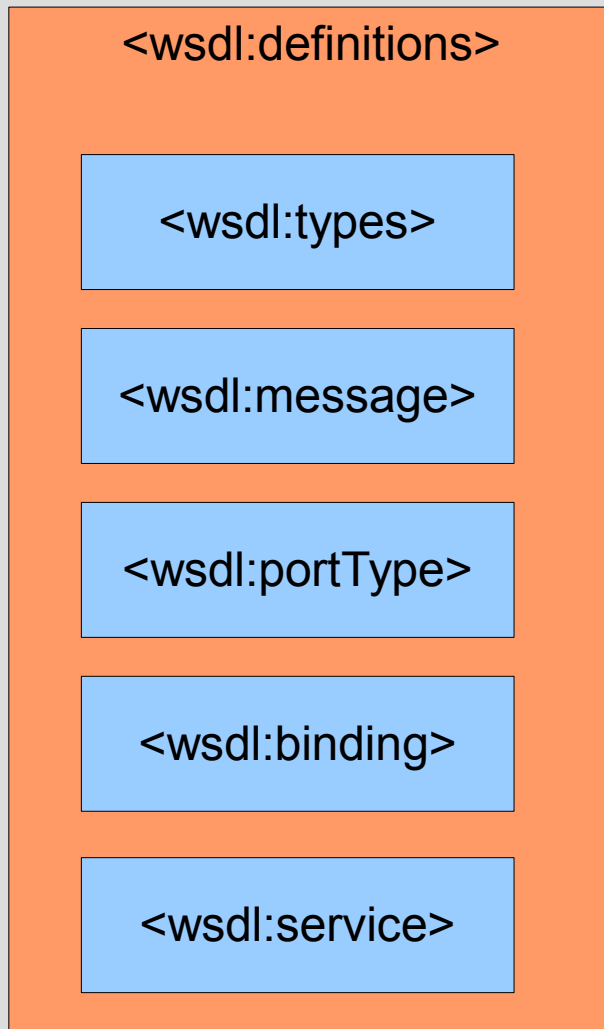
## 21.3 WSDL

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

## 21.3 WSDL



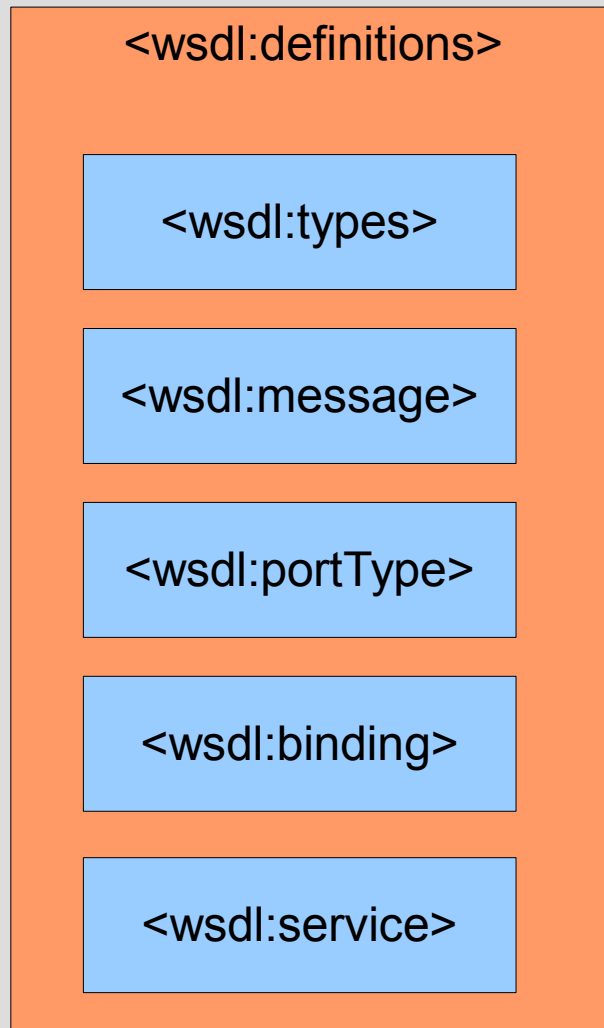
## 21.3 WSDL



- **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:portType`  
abstrakte Webservice  
Schnittstelle

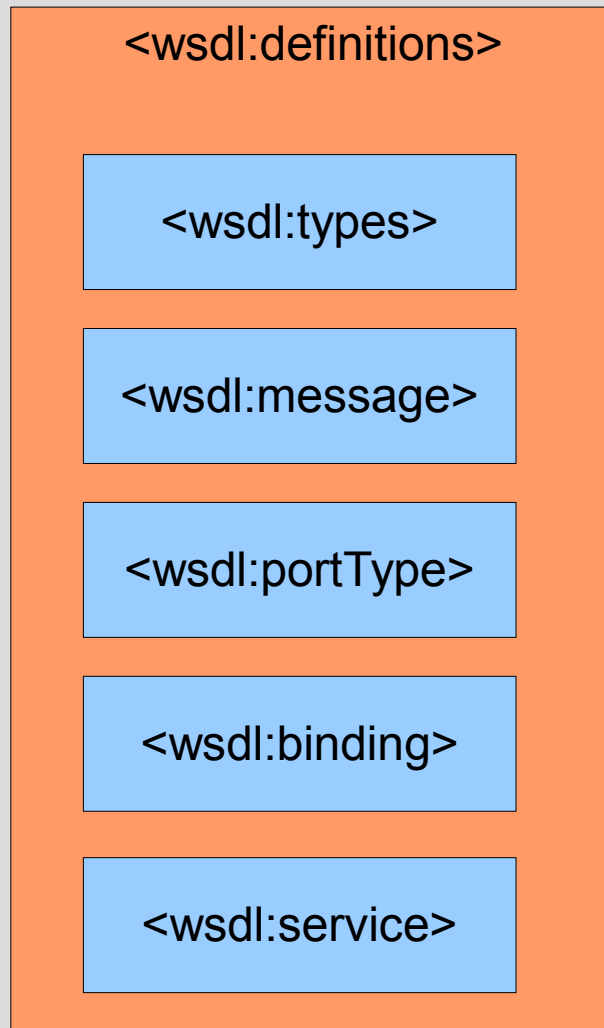
```
<wsdl:portType  
  name="CalculatePort">  
  <wsdl:operation  
    name="addValue">  
    <wsdl:input/>  
    <wsdl:output/>  
  </wsdl:operation>  
</wsdl:portType>
```

## 21.3 WSDL

- **wSDL:portType** –
  - ♦ Abstrakte Schnittstelle
  - ♦ 1..n Operationen

<b>Operation</b>	<b>Beschreibung</b>
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

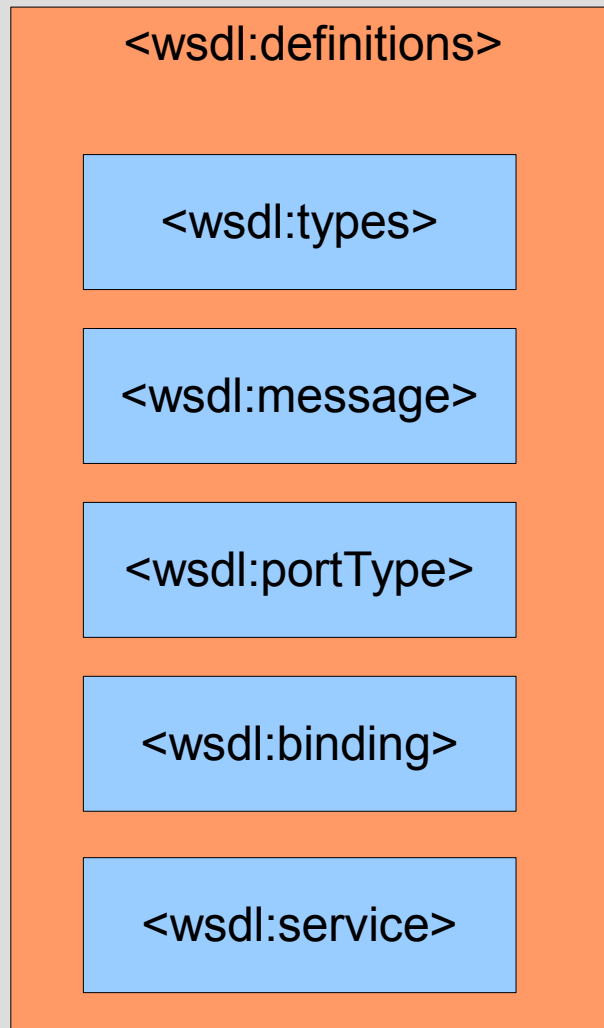
## 21.3 WSDL



- `wsdl:message` (Nachrichten)

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

## 21.3 WSDL



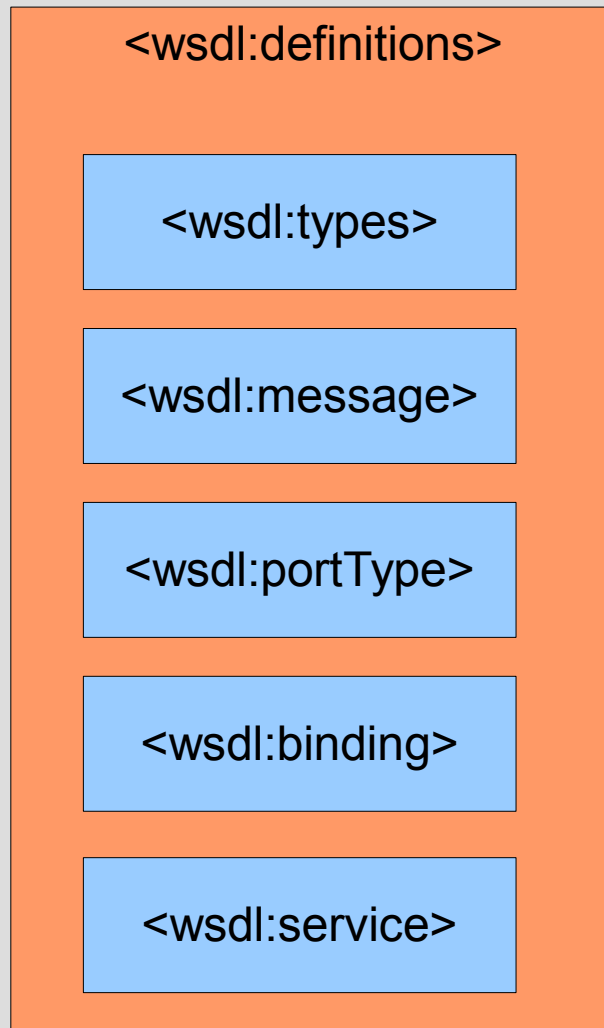
- `wsdl:binding`  
Bindung an Protokoll  
Nachrichtenprotokoll

```
<wsdl:binding
  name="CalculatePort">
  <wsdl:soapbinding
    style="rpc"
    transport="http:..">
    <wsdl:operation>
    </wsdl:operation>
  </wsdl:soapbinding>
</wsdl:binding>
```

„<http://schemas.xmlsoap.org/soap/http>“



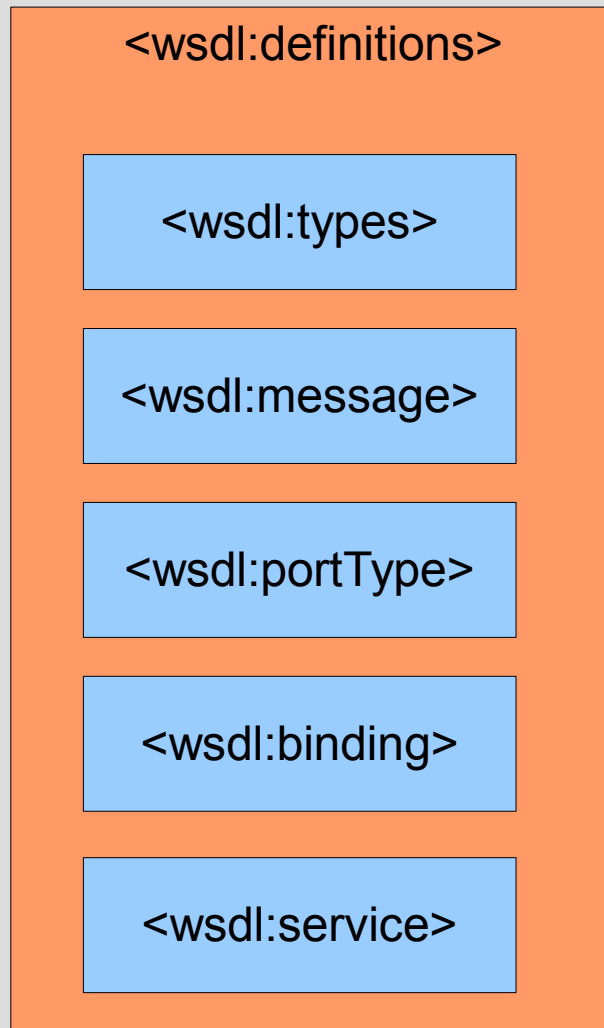
## 21.3 WSDL



- `wsdl:service`  
1..n -> Port

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

## 21.3 WSDL



- **wsdl:definitions**  
(Namensräume)

```
<wsdl:definitions  
  xmlns:wsdl="..."  
  targetNamespace="..."  
  xmlns:wsdlsoap="..."  
  xmlns:soapenc="..."  
>  
</wsdl:definitions>
```

## 21.3 WSDL

Developing a Web Service ?

How ?

Where to start ?