

Web Services

XML, WSDL, SOAP und UDDI
Einblicke und Ausblicke

Beschreibung Zugriff auf Web Services - SOAP

- Inhalt
 - Einleitendes Beispiel
 - Die SOAP Spezifikation
 - SOAP Envelope
 - SOAP Header
 - SOAP Body
 - SOAP Fehler
 - RPC Konventionen
 - Datentyp Mapping
 - HTTP Bindung
 - Versionskontrolle
 - SOAP Message Verarbeitung

Beschreibung Zugriff auf Web Services - SOAP

- Inhalt
 - ..
 - SOAP Message Verarbeitung
 - SOAP und Namensräume
 - SOAP 1.2
 - SOAP Multipart MIME Attachments
 - SOAP in bestehenden IT Landschaften
 - SOAP Ausblick

Zugriff auf Web Services - SOAP

Einleitung

- *SOAP stellt den Web Services den Transport zur Verfügung.*
 - SOAP ist vermutlich die wichtigste Web Service Technologie.
 - Der Web ist heute das wichtigste Kommunikations-Netzwerk
 - XML ist das wichtigste Daten-Repräsentation-Format
 - SOAP verbindet beide.
 - SOAP kann man als eine Erweiterung des HTTP Protokolls ansehen.
 - HTTP senden HTML Seiten
(**GET** für Einweg-Kommunikation)
 - SOAP sendet XML Messages mit Hilfe von HTTP.

Zugriff auf Web Services - SOAP

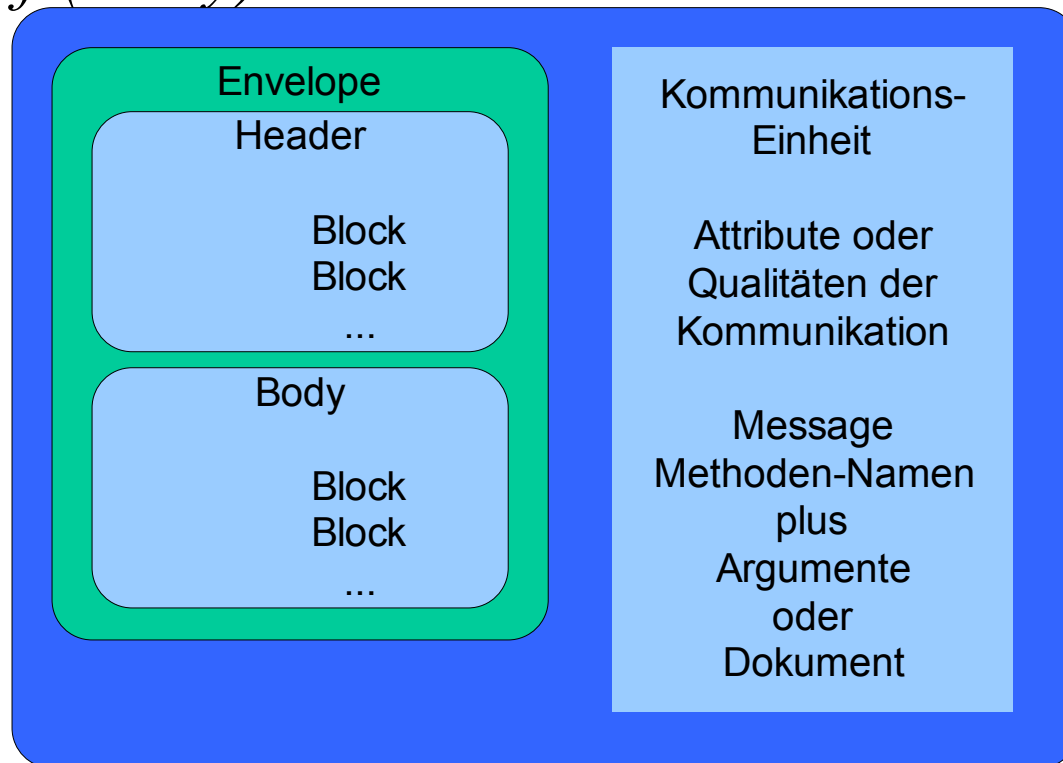
Einleitung

- *SOAP wird von einem SOAP Prozessor zur Verfügung gestellt.*
 - Ein HTTP Listener (Apache, IIS) muss einen *SOAP Prozessor* zur Verfügung stellen (Apache Axis, ...).
 - Der SOAP Prozessor muss in der Lage sein, die eintreffenden XML (SOAP) Messages zu interpretieren.
 - Der SOAP Prozessor läuft auf einem *SOAP Node*.
 - SOAP selber verfügt über kein eigenes Objektmodell.
 - SOAP definiert nur einfache Einweg-Kommunikation.
 - Darauf aufbauend können Request / Response und komplexere Szenarios aufgebaut werden.

Zugriff auf Web Services - SOAP

Einleitung

- *SOAP besteht aus drei Hauptbestandteilen*
- Umschlag (Envelope), Kopf (Header),
Rumpf (Body).



Zugriff auf Web Services - SOAP

Einleitung

- *SOAP überbrückt Web Service Implementationen.*
 - SOAP transportiert XML Dokumente über das Web (und eventuell weiterer Netzwerke), um Web Services anzusteuern.
 - SOAP Messages sind spezielle XML Dokumente.
 - Die SOAP Spezifikation ist sehr locker und leider zu offen:
 - In speziellen Gremien wird versucht, die Implementierungen unterschiedlicher Hersteller kompatibel werden zu lassen.
 - SOAP kann ganze Dokumente austauschen oder mittels RPC kommunizieren.

Zugriff auf Web Services - SOAP

Ein erstes Beispiel

- *SOAP kann auch zum Broadcasting eingesetzt werden.*
 - Das folgende Beispiel beschreibt eine Einweg-Broadcast SOAP Message, im Rumpf steht die eigentliche Aufgabe (Terminkoordination).

Beispiel: soap114.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  - <env:Header>
    - <n:broadcastService xmlns:n="http://www.xmlbus.com/broadcastServices">
      <n:list>PDA, Cell, Email, VoiceMail, IM</n:list>
    </n:broadcastService>
  </env:Header>
  - <env:Body>
    - <m:Function xmlns:m="http://www.xmlbus.com/broadcastServices/send">
      <m:message>Hallo, Du bist wie immer wieder mal zu spaet!</m:message>
    </m:Function>
  </env:Body>
</env:Envelope>
```


Zugriff auf Web Services - SOAP

Ein erstes Beispiel

- *Binden der SOAP Message an HTTP POST.*
 - Request header
POST /broadcastService HTTP/1.1
Host: www.xmlbus.com
Content-Type: text/xml; charset=„utf-8“
Content-Length:

 - <?xml version=,1.0 ` ?>
...SOAP Dokument (Request)

 - Response header
HTTP/1.1 200 OK
Content-Type: text/xml; charset=„utf-8“
Content-Length: ...
<?xml version=„1.0“ ?>
... SOAP Antwort

Zugriff auf Web Services - SOAP

Ein erstes Beispiel

- *Typischerweise werden eintreffende SOAP Messages mit HTTP Antworten verknüpft.*
 - Im obigen Beispiel wird der Service (**/broadcastService**) aufgerufen.
 - Dieser Service interpretiert den Body Block und führt die **send** Methode aus (gemäss der **Function** Angabe).
 - Diese wird, falls überhaupt, eine Antwort in Form einer HTTP Response generieren (**HTTP/1.1 200 OK**).

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

- *SOAP (Userland, Microsoft) wurde als Erweiterung von XML-RPC (Firma Userland)*
 - Heute existieren unterschiedliche Implementationen
 - IONA
 - IBM
 - MS
 - .NET
 - VC++, VBasic
 - Perl
 - Apache Axis
 - Ursprüngliche Zielsetzung
 - Keep it simple!
 - Der SOAP Standard definiert nur den kleinsten gemeinsamen Nenner, das Basis-Protokoll.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

- *SOAP definiert die Elemente und Regeln des XML Messagings.*
 - Start und Ende eines *Envelope*, der das XML Dokument einschliesst.
 - Optionale *Header* für zusätzliche Informationen (Security, Transaktionskoordination, ...).
 - *Serialisierung* der Datentypen, speziell im RPC Style.
 - *Bindung* an HTTP (und Adressierung eines entsprechenden SOAP Prozessors):
 - Der SOAP Prozessor leitet Anfragen für die Bearbeitung an EJB's, Servlets, .NET Objekte, CORBA Objekte, ... weiter.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

- *SOAP Syntax-Blöcke werden Handlern zugeordnet.*
 - Wichtiger Bestandteil der SOAP Syntax sind unabhängige Blöcke.
 - Blöcke können von passenden Handlern bearbeitet werden.
 - Envelope Elemente werden mithilfe unterschiedlicher Namensräume definiert, um Namenskonflikte zu vermeiden.
 - Das SOAP Protokoll wird in Form von Layern definiert.
 - Dies erlaubt die Kombination unterschiedlicher Protokolle
 - » OMG IIOP
 - » JMS
 - » IETF Blocks Environment Extension Protocol (BEEP)
 - Nur die HTTP Bindung wird genauer beschrieben.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Envelope

Envelope ist das *top-level XML Element* in einer SOAP Message.

– Der Umschlag ist das äusserste Element einer SOAP Message.

- Es entspricht dem Root Element des XML Dokuments.

- Namensraum

 - SOAP V1.1 :

 - » `<env:Envelope`

 - `xmlns:env="http://www.xmlsoap.org/soap/envelope">`

 - SOAP V1.2

 - » `<SOAP-ENV:Envelope`

 - `xmlns:env="http://www.w3.org/2001/12/soap-envelope">`

- Der Umschlag legt den Start und das Ende einer SOAP Message fest.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Envelope

- *Namespaces identifizieren die Envelope Version und den Encoding Style einer SOAP Message.*
 - Ein Encoding Schema ist optional und definieren komplexe Datenstrukturen, **arrays** und **structs**.
 - SOAP V1.1 :
 - » **<env:Envelope**
xmlns:env="http://www.xmlsoap.org/soap/encoding">
 - SOAP V1.2
 - » **<SOAP-ENV:Envelope**
xmlns:env="http://www.w3.org/2001/12/soap-encoding">

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Envelope

- *Die HTTP Bindung spezifiziert den Ort des Dienstes.*

- Beispiel

- `POST /OrderEntry HTTP/1.1`

`Host: www.xmlbus.com`

`Content-Type: application/soap; charset..`

```
<SOAP-ENV:Envelope
```

```
xmlns:env="http://www.w3.org/2001/12/
```

```
soap-encoding">
```

```
...
```


Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

SOAP Header

- *Headers sind optional; es können auch mehrere vorhanden sein.*
 - Die Bedeutung, die Semantik des Headers wird typischerweise in einem XML Schema festgelegt.
 - Headers ergänzen eine SOAP Message durch zusätzliche Funktionalitäten.
 - Security
 - Transaktionen
 - Quality of Service
 - Headers sind direkte Child Nodes des Envelope Elements.
 - *Die Definition der Header sollte besser standardisiert werden!*

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

SOAP Header

- *Das mustUnderstand Attribut.*
 - Ein Sender kann von Empfänger verlangen, dass er den Header verstehen muss:
 - Falls der Empfänger dies nicht tut, muss er die SOAP Message zurückweisen und eine Fehlermeldung generieren.

- Beispiel

```
– <SOAP-ENV:Header>
  <t:Transaction
    xmlns:t=„www.xmlbus.com“
    SOAP-ENV:mustUnderstand=„1“>
    <!-- 1 = true //-->
    5
  </t:Transaction>
</SOAP-ENV:Header>
```

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

SOAP Body

- *Der Rumpf (Body) enthält die Informationen, welche an den Empfänger gesandt werden sollen.*
 - Der Rumpf ist im Umschlag enthalten und auf die Header(s) folgend.
 - Der Rumpf ist ein Child Node des Umschlags, seine Semantik wird in einem XML Schema beschrieben.
 - Beispiel
 - Request
 - `<SOAP-ENV:Envelope . . . >`
`<SOAP-ENV:Header> . . .`
`<SOAP-ENV:Body>`
`<m:GetOrderStatus`
`xmlns:m=„www.xmlbus.com/OrderEntry“>`
`<orderNr>1732</orderNr>`
`</m:GetOrderStatus>`
`</SOAP-ENV:Body>`
`. . .`
`</SOAP-ENV:Envelope>`

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

SOAP Faults

- *Falls eine SOAP Message nicht verarbeitet werden kann, wird ein Fehler gemeldet.*
 - Bei der HTTP Bindung
 - HTTP Fehler liegen im Bereich 200 ... 299
 - SOAP Fehler im Bereich 500 ... 599
 - Pro Antwort auf eine SOAP Anfrage kann lediglich ein Fehlercode mitgegeben werden.
 - Im SOAP Fault Body können folgende Elemente enthalten sein
 - **<faultcode>** : ein XML-qualifizierter Name (**Sender** , **Receiver**)
 - **<faultstring>** : erklärender Text zum Fehlercode
 - **<faultactor>** : URI des SOAP Prozessors, der den Fehler generierte
 - **<detail>** : Applikations-spezifische Infos; falls diese fehlen, wurde die Body Information nicht verarbeitet.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

SOAP Faults

- *Beispiel*

- HTTP/1.1 500 Internal Server Error
Content-Type: text/xml; charset=„utf-8“
Content-Length: ...

```
<env:Envelope xmlns:env=„...“>
<env:Header>
  <V:Upgrade xmlns:V=„../soap-upgrade“>
    <envelope qname=„nls:Envelope“
      xmlns:nls=„...“>
    </envelope>
  </V:Upgrade>
</env:Header>
<env:Body>
  <env:Fault>
    <faultcode>env:VersionMismatch
    </faultcode>
    <faultstring>Version mismatch
    </faultstring>
  </env:Fault> ...
```

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Konventionen

- *Das Mapping von SOAP Messages auf RPC Mechanismen ist optionaler Teil der Spezifikation.*
 - Optional können auch Sicherheitsinformationen, Transaktionskontexte oder andere Attribute des RPC mitgegeben werden.
 - RPC Aufrufe werden mittels **structs** modelliert.
 - In und In/Out Parameter werden als **accessor** bezeichnet.
 - Der Name der Request **struct** ist identisch mit dem Methodennamen.
 - Das Ergebnis eines RPC's kann eine Fehlermeldung oder eine Response Message sein, aber nicht beides.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

Beispiel: soap130.xml

Kapitel4\soap130.xml - Microsoft Internet Explorer

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd1="http://www.xmlbus.com/OurchaseOrderService-xsd"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <SOAP-ENV:Body>
  + <ns1:postPurchaseOrder
    xmlns:ns1="http://www.xmlbus.com/PurchaseOrderService">
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

POST

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

Beispiel: soap130.xml

Kapitel4\soap130.xml - Microsoft Internet Explorer

```
- <SOAP-ENV:Body>
- <ns1:postPurchaseOrder
  xmlns:ns1="http://www.xmlbus.com/PurchaseOrderService">
- <order xsi:type="xsd:PurchaseOrder">
  <CompanyName xsi:type="xsd:string">Whoopy</CompanyName>
- <Items xsi:type="SOAP-ENC:Array" SOAP-
  ENC:arrayType="xsd:item[1]">
+ <item xsi:type="xsd:item">
  </Items>
+ <Address xsi:type="xsd:Address">
  </order>
</ns1:postPurchaseOrder>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Kunden-
spezifisches Array

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

Beispiel: soap130.xml

Kapitel4\soap130.xml - Microsoft Internet Explorer

```
- <Items xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="xsd:item[1]">
- <item xsi:type="xsd:item">
  <Price xsi:type="xsd:float"> 130.75</Price>
  <PartID xsi:type="xsd:string"> 5613</PartID>
  <Description xsi:type="xsd:string"> SOAP in 21
  Stunden</Description>
  <Quantity xsi:type="xsd:int"> 5</Quantity>
</item>
</Items>
+ <Address xsi:type="xsd:Address">
</order>
</nsl:postPurchaseOrder>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Artikel-
Beschreibung

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

Beispiel: soap130.xml

Kapitel4\soap130.xml - Microsoft Internet Explorer

```
</item>
</Items>
- <Address xsi:type="xsd:Address">
  <State xsi:type="xsd:string">SG</State>
  <PostalCode xsi:type="xsd:string">8640</PostalCode>
  <City xsi:type="xsd:string">Rapperswil</City>
  <Line2 xsi:type="xsd:string" />
  <Country xsi:type="xsd:string">Schweiz</Country>
  <Line1 xsi:type="xsd:string">Oberseestrasse 10</Line1>
</Address>
</order>
</nsl:postPurchaseOrder>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Adress-
Beschreibung

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

- *Möglicher Java Code des Dienstes (Bean)*

- `public class PurchaseOrderService {`

```
    public float postPurchaseOrder(PurchaseOrder po)
    {
        Item items[] = order.getItems();
        float total = 0.0f;
        for (int i=0; i<items.length; i++) {
            total= total+
                items[i].getPrice()*
                items[i].getQuantity();
        }
        return total;
    }
// ...
}
```

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

RPC Beispiel

- *Mögliche Antwort*

```
<?xml version="1.0" encoding="utf-8" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd1="http://www.xmlbus.com/PurchaseOrderService-xsd"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <SOAP-ENV:Body>
- <ns1:postPurchaseOrderResponse
  xmlns:ns1="http://www.xmlbus.com/PurchaseOrderService">
- <return xsi:type="xsd:float">5678.90</return>
  </ns1:postPurchaseOrderResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Data Type Mapping

- *Einfache SOAP Datentypen sind die gleichen wie die Schema Datentypen*
 - SOAP Datentypen können einfach sein
 - Integer, Float, Text, Date
 - SOAP Datentypen können komplex sein
 - Structures, Arrays... wie im SOAP Encoding Schema definiert.
 - Arrays können auch verschachtelt sein
 - Sender und Empfänger müssen das selbe Encoding Schema verwenden, SOAP macht kaum Vorschriften.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Data Type Mapping

- *XML Instanzen werden durch das Mapping von Eingabedaten auf Schemas generiert.*
 - Schemas generieren / definieren die Syntax
 - Ablauf
 - Ein Web Service generiert eine Ausgabe
 - Dazu werden die Daten als Quelle für eine XML Instanz benutzt.
 - Das XML Dokument wird übermittelt (mittels SOAP).
 - Beim Empfänger
 - Das Schema wird benutzt, um das empfangene XML Dokument zu analysieren, die Daten zu extrahieren oder das Dokument in irgend einer Form weiter zu bearbeiten.

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

HTTP Binding

- *Die HTTP Bindung garantiert, dass eine SOAP Message korrekt mittels HTTP übertragen werden kann.*
 - Die HTTP Bindung ist die einzige, welche in der Spezifikation enthalten ist.
 - Die Spezifikation garantiert damit, dass SOAP Messages über HTTP übertragen werden können und vom HTTP Server interpretiert werden können, also SOAP, nicht als XML!
 - Beispiel
 - Siehe weiter vorne

Zugriff auf Web Services - SOAP

Die SOAP Spezifikation

Versionskontrolle

- *Die Version von SOAP wird über die Schema-Version kontrolliert.*
 - SOAP definiert pro Version eine Schema-Version:
 - V1.1 SOAP Umschlag
,<http://schemas.xmlsoap.org/soap/envelope>
 - V1.2 SOAP (neu bei W3C)
,<http://www.w3.org/2001/12/soap-envelope>
 - Der SOAP Prozessor muss die Korrektheit der Version jeweils überprüfen (Sender und Empfänger)
 - Falls Versionskonflikte auftreten, muss eine **VersionMismatch** SOAP Fehlermeldung generiert werden.
 - Ab Version 1.2 können mithilfe des
,<http://www.w3.org/2001/12/soap-upgrade>
Namespace Identifiers die Envelope Versionen aufgelistet werden.

Zugriff auf Web Services - SOAP

SOAP Message Processing

- *Das Schema definiert, wie Messages interpretiert werden müssen.*
 - Eine SOAP Message kann maschinell nur verstanden werden, wenn die Schemata dazu bekannt sind.
 - Das populärste XML Schema ist in diesem Sinne WSDL.
 - Verschiedene Schemas zu SOAP sind auf dem Web unter **schemas.xmlsoap.org** erhältlich:
 - Schema Encoding <http://www.w3.org/2001/06/soap-encoding>
 - Schema Envelope <http://schemas.xmlsoap.org/soap/envelope/>
 - Schema HTTP <http://schemas.xmlsoap.org/wsdl/http/>
 - Schema MIME <http://schemas.xmlsoap.org/wsdl/mime/>
 - Schema SOAP <http://schemas.xmlsoap.org/wsdl/soap/>
 - Schema WSDL <http://schemas.xmlsoap.org/wsdl/>

Zugriff auf Web Services - SOAP

SOAP Message Processing

- *Beispiel: Auszug aus dem Envelope Schema.*

```
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.xmlsoap.org/soap/envelope/"
  targetNamespace="http://schemas.xmlsoap.org/soap/envelope/"
  <!-- Envelope, header and body -->
  <xs:element name="Envelope" type="tns:Envelope" />
- <xs:complexType name="Envelope">
  - <xs:sequence>
    <xs:element ref="tns:Header" minOccurs="0" />
    <xs:element ref="tns:Body" minOccurs="1" />
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
      processContents="lax" />
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax" />
  </xs:complexType>
  <xs:element name="Header" type="tns:Header" />
+ <xs:complexType name="Header">
  <xs:element name="Body" type="tns:Body" />
+ <xs:complexType name="Body">
```

Header ist optional

Zugriff auf Web Services - SOAP

SOAP Message Processing

- *Beispiel: Envelope Schema globale Attribute*

```
+ <xs:complexType name="Body">
  <!-- Global Attributes. The following attributes are intended to be
  usable via qualified attribute names on any complex type referencing them.
- <xs:attribute name="mustUnderstand" default="0">
- <xs:simpleType>
  - <xs:restriction base="xs:boolean">
    <xs:pattern value="0|1" />
  </xs:restriction>
  </xs:simpleType>
</xs:attribute>
  <xs:attribute name="actor" type="xs:anyURI" />
+ <xs:simpleType name="encodingStyle">
  <xs:attribute name="encodingStyle" type="tns:encodingStyle" />
+ <xs:attributeGroup name="encodingStyle">
  <xs:element name="Fault" type="tns:Fault" />
+ <xs:complexType name="Fault" final="extension">
+ <xs:complexType name="detail">
</xs:schema>
```

Zugriff auf Web Services - SOAP

SOAP Message Processing

- *Der Encoding Style wird in einem Schema festgelegt.*
 - Das Schema "<http://www.w3.org/2001/XMLSchema>" enthält die Definitionen der gängigen Datentypen, einfache und komplexe.
 - Applikations-spezifische Schemas definieren die Semantik der Applikationsdaten.
 - Optional können im Header Schemata definiert werden
 - Diese sind auf xmlsoap.org und W3C nicht festgelegt worden.
 - Header und Body Teile der SOAP Message gehören zusammen
 - Im Header wird die Sicherheit, Qualität, ... festgelegt.

Zugriff auf Web Services - SOAP

SOAP Message Processing

- *Messages können durch Zwischenknoten geleitet werden.*
 - Zwischenknoten können die Headerinformationen modifizieren, aber nicht die Rumpfinformationen.
 - Das Routing-Protokoll selber wird in der Spezifikation nicht festgelegt:
 - IBM und MS haben WS-Routing als Proposal publiziert.

Zugriff auf Web Services - SOAP

SOAP Einsatz von Namespaces

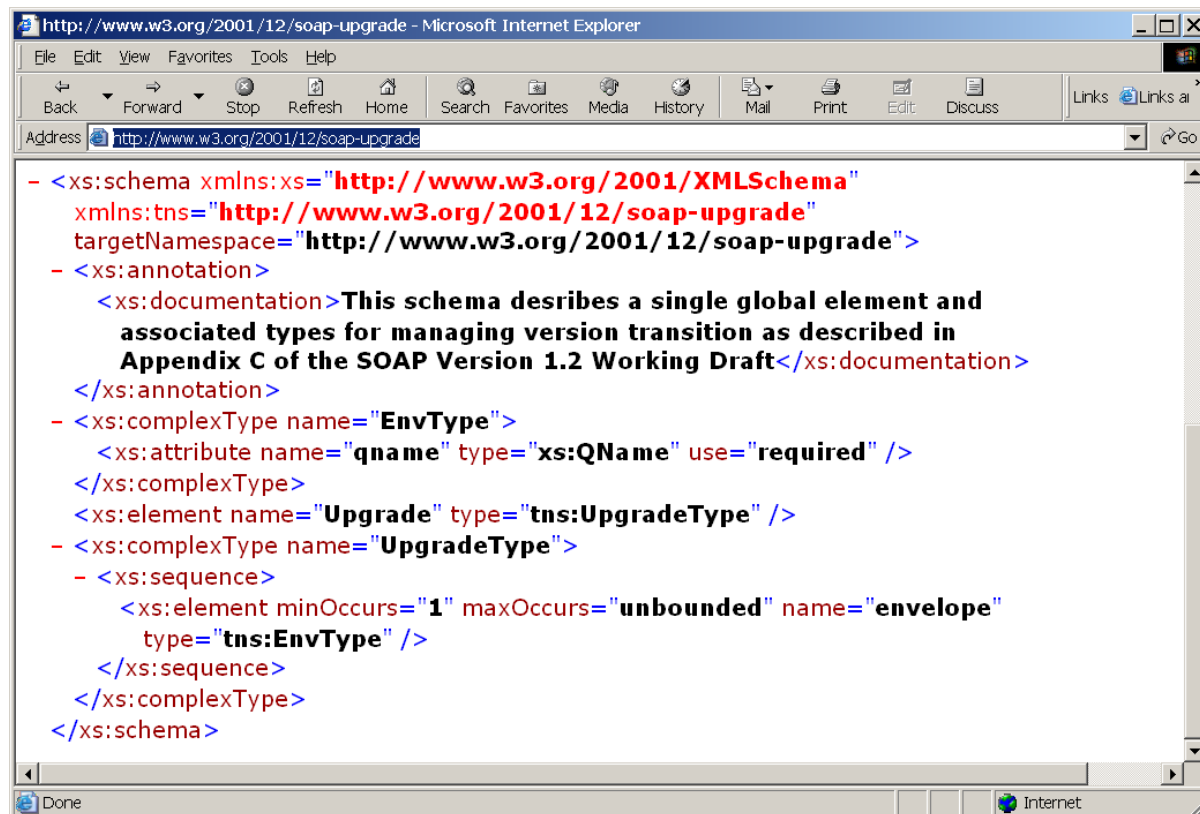
- *Namespaces qualifizieren Blocknamen und spezifizieren Attribute.*
 - SOAP Messages verwenden Namensräume, um die Elementnamen zu qualifizieren.
 - Die Namensräume der aktuellen Version (v1.2 bei W3C) sind:
 - **`http://www.w3.org/2001/12/soap-envelope`**
Elemente und Attribute für die Hauptteile oder Pflichtteile
 - **`http://www.w3.org/2001/12/soap-encoding`**
Elemente und Attribute der Datentypen & Serialisierung
 - **`http://www.w3.org/2001/12/soap-faults`**
Fehlercodes
 - **`http://www.w3.org/2001/12/soap-upgrade`**
Liste der unterstützten SOAP Versionen

Zugriff auf Web Services - SOAP

SOAP Einsatz von Namespaces

- *Beispiel*

- <http://www.w3.org/2001/12/soap-upgrade>
Liste der unterstützten SOAP Versionen



The screenshot shows a Microsoft Internet Explorer browser window with the address bar containing <http://www.w3.org/2001/12/soap-upgrade>. The main content area displays the XML Schema for SOAP 1.2 Upgrade, which defines a single global element and associated types for managing version transition. The schema includes an annotation describing the schema's purpose, a complex type 'EnvType' with a required 'qname' attribute, and a complex type 'UpgradeType' containing a sequence of an 'envelope' element.

```
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.w3.org/2001/12/soap-upgrade"
  targetNamespace="http://www.w3.org/2001/12/soap-upgrade">
- <xs:annotation>
  <xs:documentation>This schema describes a single global element and
  associated types for managing version transition as described in
  Appendix C of the SOAP Version 1.2 Working Draft</xs:documentation>
</xs:annotation>
- <xs:complexType name="EnvType">
  <xs:attribute name="qname" type="xs:QName" use="required" />
</xs:complexType>
  <xs:element name="Upgrade" type="tns:UpgradeType" />
- <xs:complexType name="UpgradeType">
  - <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded" name="envelope"
      type="tns:EnvType" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Zugriff auf Web Services - SOAP

SOAP 1.2

- *Einige Änderungen*
 - Die Spezifikation wurde in mehrere Teile aufgeteilt
 - Part 0 : Primer; Part 1 : Messaging Format; Part 2 : Adjuncts
 - Neue und zusätzliche Fehlercodes
 - Im HTTP Binding wurde **text/xml** durch **application/soap+xml** ersetzt
 - Für RPC wurde ein **<response>** Element definiert.

Zugriff auf Web Services - SOAP

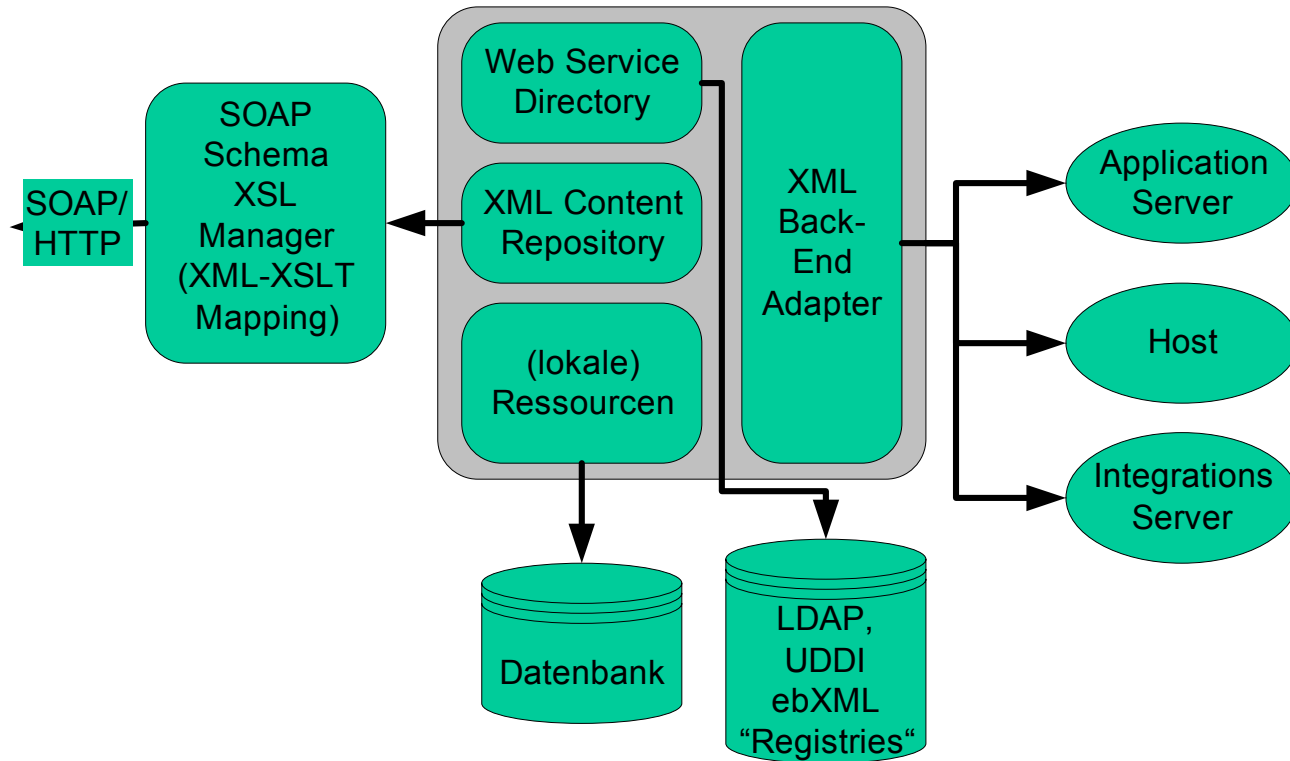
SOAP Multipart MIME Attachments

- *SOAP mit Attachments können binäre Daten und umfangreiche XML Dokumente versenden.*
 - Attachments gestatten eine wesentlich effizientere Übertragung komplexer Daten mittels SOAP.
 - Dazu wird eine ContentID im MIME Umschlag definiert
 - Beispiel
 - `<SOAP-ENV:Body>`
 - `<photo>`
 - `<photo1 href=„newhamshire.jpg@nh.com“/>`
 - `</SOAP-ENV:Body>`
 - `Content-Type: image/jpg`
 - `Content-Transfer-Encoding: binary`
 - `Content-ID: „ newhamshire.jpg@nh.com“`
 - `... Bild ...`

Zugriff auf Web Services - SOAP

SOAP Applikations-Integration

- *SOAP Prozessoren benötigen lokale Metadaten-Manager.*



Zugriff auf Web Services - SOAP

SOAP - Die Zukunft

- *SOAP wird laufend weiterentwickelt.*
 - Aktuelle Themen:
 - Security
 - Transaktionen
 - Quality-of-Service Policies
 - Objekt Referenzen
 - Garbage Collection
 - Fehlertoleranz
 - Work Flow / Business Prozesse

Zugriff auf Web Services - SOAP

Referenzen

- Spezifikation und Schema
 - [,http://schemas.xmlsoap.org/wsdl/](http://schemas.xmlsoap.org/wsdl/)
 - [,http://www.w3.org/TR/wsdl/](http://www.w3.org/TR/wsdl/)
 - [,http://schemas.xmlsoap.org/wsdl/](http://schemas.xmlsoap.org/wsdl/)
WSDL Framework Namensraum.
 - [,http://schemas.xmlsoap.org/wsdl/soap](http://schemas.xmlsoap.org/wsdl/soap/)
SOAP Umschlagsdefinition im WSDL Binding für SOAP
 - [,http://schemas.xmlsoap.org/wsdl/http](http://schemas.xmlsoap.org/wsdl/http/)
HTTP GET und POST Bindung für SOAP
 - [,http://schemas.xmlsoap.org/wsdl/mime](http://schemas.xmlsoap.org/wsdl/mime/)
MIME Bindung für SOAP
 - [,http://schemas.xmlsoap.org/soap/encoding](http://schemas.xmlsoap.org/soap/encoding/)
SOAP 1.1 Encoding Schema
 - [,http://schemas.xmlsoap.org/soap/envelope](http://schemas.xmlsoap.org/soap/envelope/)
SOAP 1.1 Umschlag
 - [,http://www.w3.org/2001/XMLSchema](http://www.w3.org/2001/XMLSchema)
XML Schema Namensraum

Zugriff auf Web Services - SOAP

Referenzen

- Die Namensräume der aktuellen Version (v1.2 bei W3C) sind:
 - <http://www.w3.org/2001/12/soap-envelope>
Elemente und Attribute für die Hauptteile oder Pflichtteile
 - <http://www.w3.org/2001/12/soap-encoding>
Elemente und Attribute der Datentypen & Serialisierung
 - <http://www.w3.org/2001/12/soap-faults>
Fehlercodes
 - <http://www.w3.org/2001/12/soap-upgrade>
Liste der unterstützten SOAP Versionen

Zugriff auf Web Services - SOAP

Referenzen

- Verschiedene Schemas zu SOAP sind auf dem Web unter schemas.xmlsoap.org erhältlich:
 - Schema Encoding <http://www.w3.org/2001/06/soap-encoding>
 - Schema Envelope <http://schemas.xmlsoap.org/soap/envelope/>
 - Schema HTTP <http://schemas.xmlsoap.org/wsdl/http/>
 - Schema MIME <http://schemas.xmlsoap.org/wsdl/mime/>
 - Schema SOAP <http://schemas.xmlsoap.org/wsdl/soap/>
 - Schema WSDL <http://schemas.xmlsoap.org/wsdl/>