

Verteilte Systeme

2. Die Client-Server-Beziehung und daraus resultierende Techniken

2.5 Kommunikation mit dem Webservice-Standard

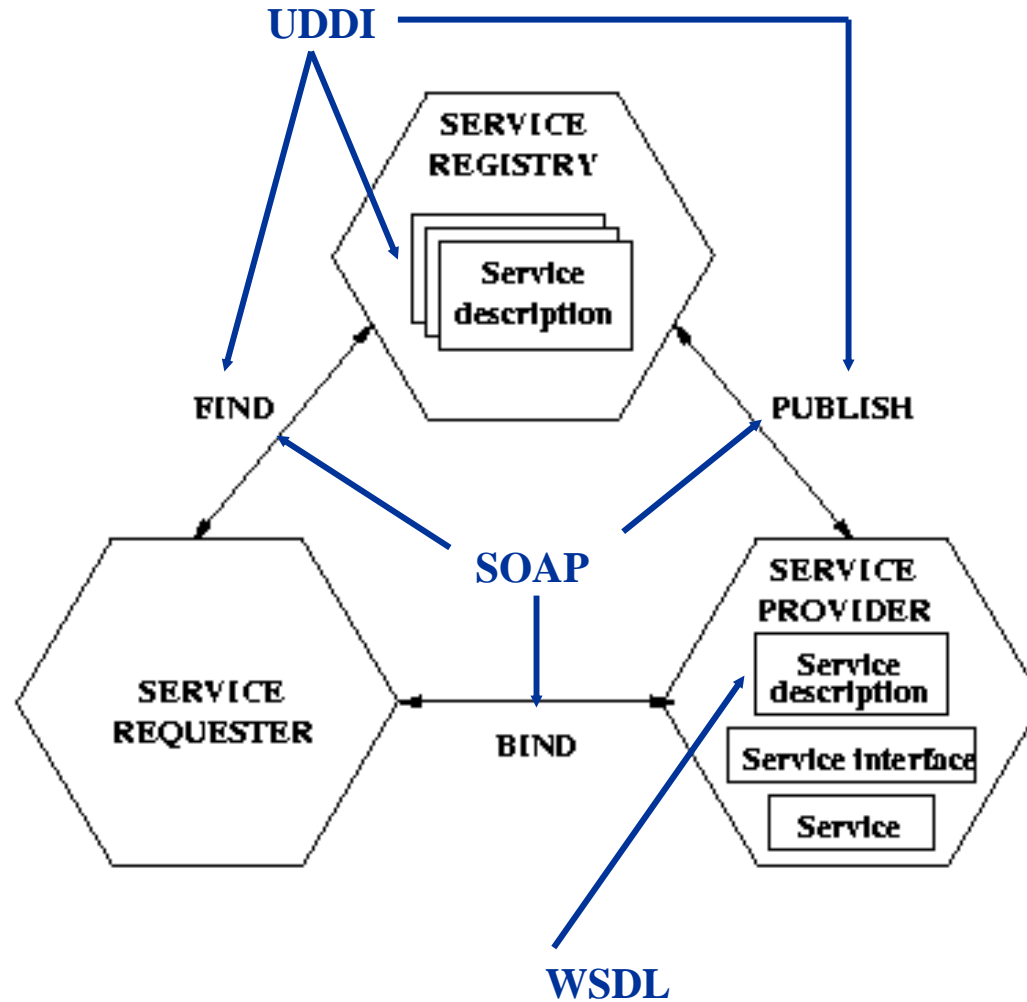
Sebastian Iwanowski
FH Wedel

Webservices: Bestandteile

Protokollebenen:

- **Suchdienst** **UDDI:** Universal Description and Discovery Interface
- **Nachrichtenbeschreibung** **WSDL:** Web Services Description Language
- **Nachrichtenkodierung** **SOAP:** Simple Object Access Protocol
Ältere Version: **XML-RPC**
- **Nachrichtentransport** **HTTP, HTTPS**
 - Die Standards bauen aufeinander auf und greifen ineinander über
 - Die Web Services Standards werden von allen Großen unterstützt (Microsoft, Sun, HP, IBM, ...)
 - Für alle Standards gibt es Java-APIs (J2EE) und Microsoft-APIs (.NET)

Webservices: Bestandteile



aus Alonso / Pautasso: graduate course in Lappeenranta,
<http://www.inf.ethz.ch/personal/alonso/teaching.html>

SOAP

Entstehungsgeschichte und Ziele

- 1999 initiiert vom W3C, immer noch verantwortlich
- wurde geschaffen zur Realisierung entfernter Prozeduraufrufe
- sollte einfachere Alternative zu CORBA IIOP/GIOP sein
 - Trägerprotokoll: HTTP
- Später Kern zur Entwicklung weiterer Paradigmen (SOA)
 - Weitere Trägerprotokolle möglich (SMTP, Messaging Service)

SOAP

Komponenten einer SOAP-Beschreibung:

Envelope

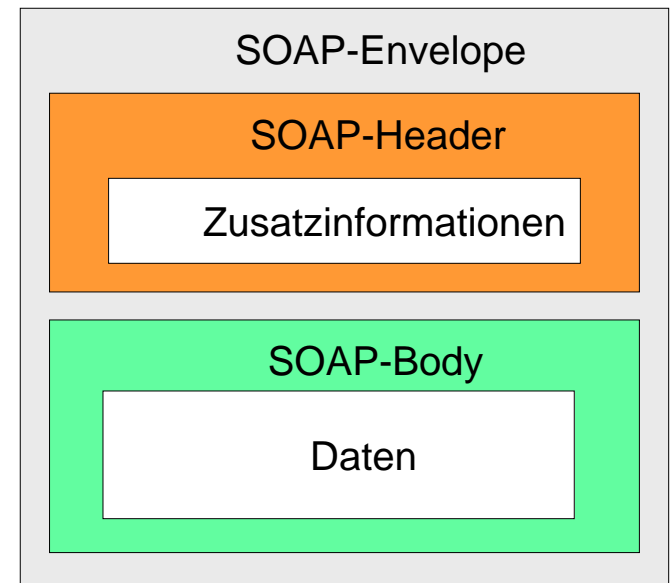
- dient als Container für die gesamte SOAP Nachricht
- muss Konventionen genügen, damit die Nachricht als SOAP Nachricht identifiziert werden kann.

Header (Optional)

- Wie der Header verwendet wird, ist weitgehend Sender und Empfänger überlassen
- Platz für weitere Standards (Sicherheit, Transaktionsmanagement)

Body

- Nutzdaten
- Funktionsaufrufe mit Daten in XML Format
- Es werden drei Arten von Nachrichten unterschieden: Request, Response, Fault.

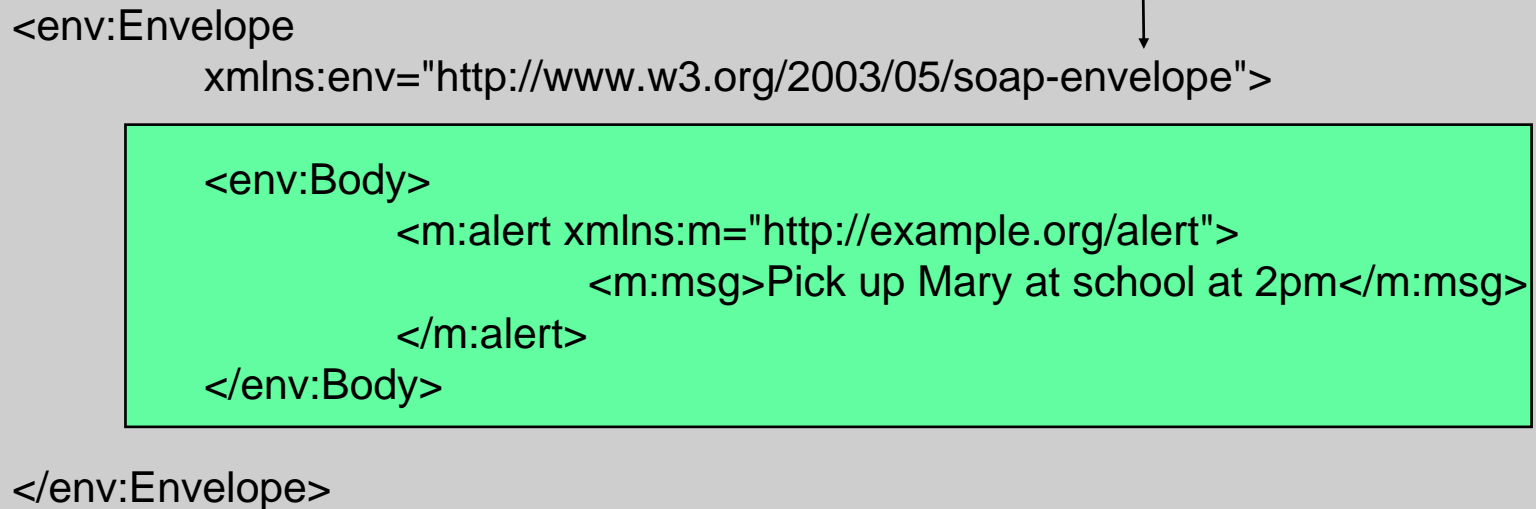


aus Hammerschall: *Verteilte Systeme und Anwendungen*

SOAP

Minimale SOAP-Beschreibung:

XML name space identifier for SOAP envelope



The diagram illustrates the structure of a SOAP envelope. It consists of an outer `<env:Envelope>` element with an attribute `xmlns:env="http://www.w3.org/2003/05/soap-envelope"`. Inside this envelope is a `<env:Body>` element, which is highlighted in green. The body contains an `<m:alert>` element with an attribute `xmlns:m="http://example.org/alert"` and a text child element `<m:msg>Pick up Mary at school at 2pm</m:msg>`. The entire structure is enclosed in `</env:Envelope>`.

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

SOAP

SOAP-Beschreibung mit Header und Body:

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol"
      env:mustUnderstand="true" >
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

SOAP

Versenden von SOAP-Nachrichten mit HTTP

HTTP implementiert ein Request/Response Modell der Kommunikation:

- SOAP-Frage wird mit HTTP-Request verschickt.
- SOAP-Antwort wird mit HTTP-Response verschickt.

Da HTTP zustandslos ist, ist auch eine SOAP Kommunikation zustandslos:

- Ein Service kann keinen Zustand zwischen zwei SOAP Aufrufen halten.
- Das kann durch explizite Implementierungen auf den Anwendungsschichten umgangen werden.

Sicherung der Verbindung:

- beispielsweise durch SSL (Secure Socket Layer)

aus Hammerschall: *Verteilte Systeme und Anwendungen*

SOAP

Versenden von SOAP-Nachrichten mit HTTP: Request

POST /StockQuote HTTP/1.1

Host: www.stockquoteserver.com

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

Content-Length: nnnn

SOAPAction: "Some-URI"

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

```
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

SOAP

Versenden von SOAP-Nachrichten mit HTTP: Response

HTTP/1.1 200 OK

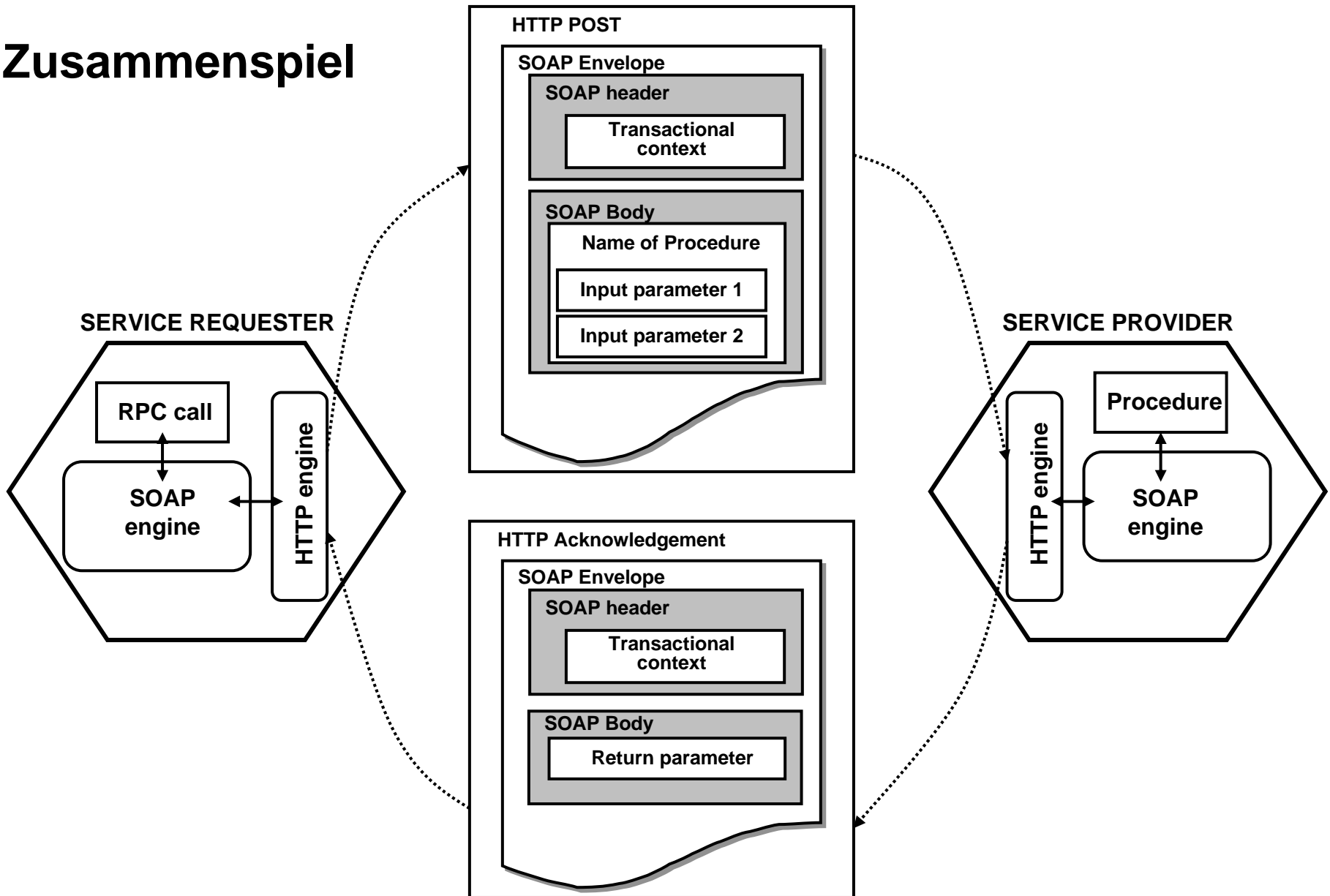
Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

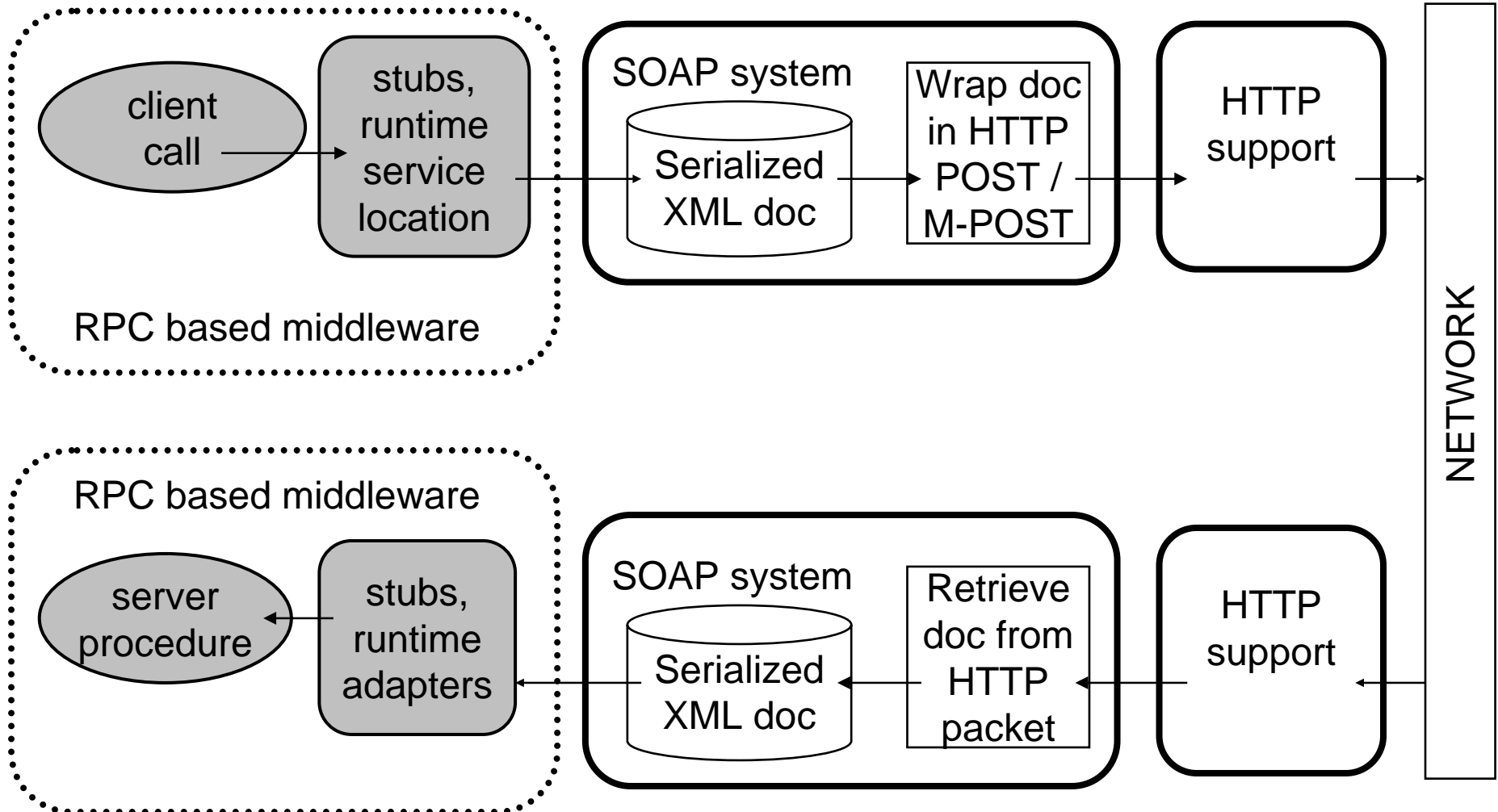
SOAP

Zusammenspiel



SOAP

RPC mit SOAP



aus Alonso / Bussler: EDBT-WebService-Tutorial,
<http://www.inf.ethz.ch/personal/alonso/teaching.html>

WSDL

Was ist WSDL ?

- Schnittstellensprache zur Veröffentlichung von Diensten als Webservices

Was sind die Ziele von WSDL ?

- Schaffung eines von Applikationen lesbaren Zugangs zu angebotenen Diensten
- weitgehend automatisierte Integration von Diensten zu neuen Anwendungen
- Integration zur Laufzeit

Wie sieht eine WSDL-Beschreibung aus ?

- Vollständige Formulierung in XML
- Der WSDL-Standard definiert ein spezifisches XML-Schema.

aus Hammerschall: *Verteilte Systeme und Anwendungen*

WSDL

Aufbau einer WSDL-Schnittstelle

<definitions>

Wurzelement eines WSDL Dokuments

definiert Namen und Namensraum des Services sowie den Namensraum der verwendeten Standards

<types>

enthält alle Datentypdefinitionen, die für den Aufruf des Services benötigt werden und nicht im Standard von XML-Schema des W3C definiert sind.

</types>

<message name="Message1">

definiert die Nachrichten, die bei einem SOAP Aufruf übertragen werden

Wenn mehrere Nachrichten vorhanden sind (z.B. für Eingabeparameter und Rückgabewert), so werden mehrere Nachrichten definiert.

Eine Nachricht kann aus logischen Teilelementen bestehen, so genannten Parts.
Ein Part definiert ein Name-Wert Paar zu den Parametern einer Nachricht.

</message>

....

aus Hammerschall: *Verteilte Systeme und Anwendungen*

WSDL

Aufbau einer WSDL-Schnittstelle

.....

<portType>

beschreibt die Methoden, die der Web Service anbietet

Zur Definition der Parameter werden die in Message definierten Nachrichten verwendet.

Jede Methode wird durch <operation> Input-Info OutputInfo FaultInfos </operation> deklariert.

Für den Aufruf unterstützt WSDL vier Kommunikationstypen:

- **One-way:** Der Client sendet eine Nachricht an den Web Service, eine Antwort wird nicht erwartet.
- **Request-Response:** Der Client sendet eine Nachricht und erhält vom Web Services eine Antwort.
- **Solicit Response:** Der Server sendet eine Nachricht und erhält vom Client eine Antwort.
- **Notification:** Der Server sendet eine Nachricht an den Client, eine Antwort wird nicht erwartet.

</portType>

.....

Anmerkung

- messages und portType gehören nur zu WSDL 1.1
- In WSDL 2.0 ersetzt durch interface (erste Beispiele in Dostal / Jeckle et al.)

nach Hammerschall: *Verteilte Systeme und Anwendungen*

WSDL

Aufbau einer WSDL-Schnittstelle

.....

<binding>

definiert Nachrichtenformate und Transportprotokoll zur Übertragung der Aufrufe
Meistens wird das SOAP Binding eingesetzt.

Für jede bei port-Type deklarierte Operation muss hier ein konkretes Binding stehen.

</binding>

<service>

definiert alle für den Zugriff auf den Dienst notwendigen Informationen
wie Netzwerkadresse und Portnummer.

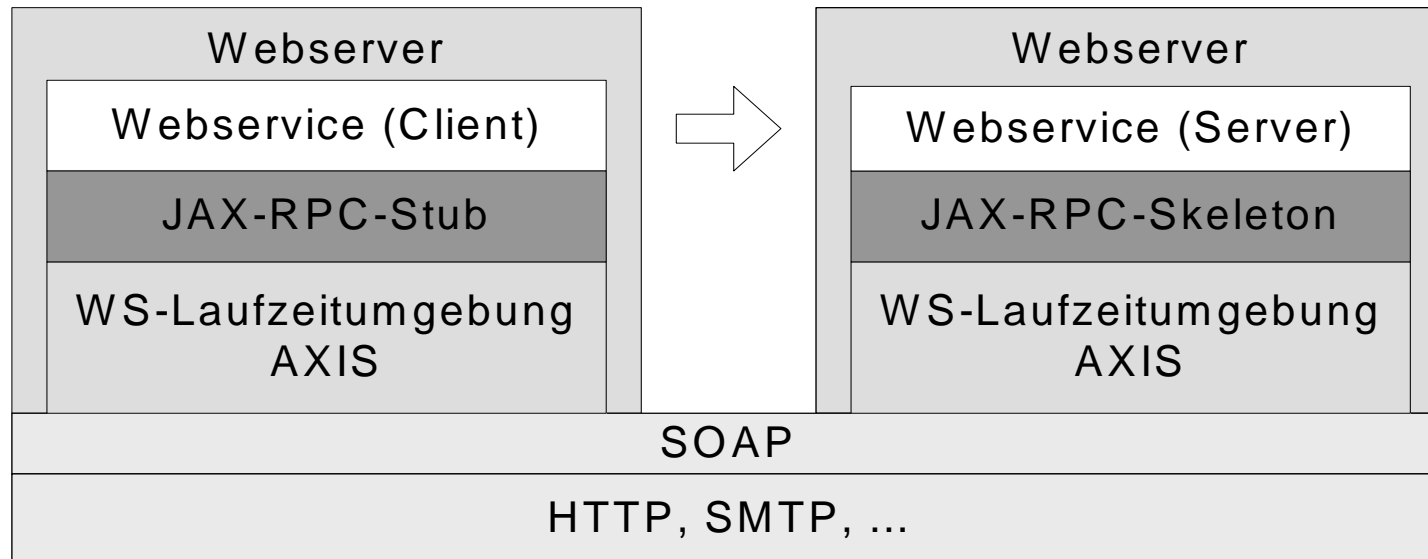
</service>

</definitions>

nach Hammerschall: *Verteilte Systeme und Anwendungen*

JAVA-Anbindung für Webservices

- Die Java-APIs for XML-basiertes RPC (JAX-RPC) sind der Standard einer Schnittstelle für Java-Anwendungen auf Webservices.
- SOAP wird um eine Java Schnittstelle erweitert.
- Zur Entwicklung von Webservices wird ein Service-Endpoint-Interface in Java definiert.
- Mit Hilfe eines Compilers wird daraus die WSDL-Schnittstelle generiert.
- Aus einer vorhandenen WSDL-Schnittstelle können auch die Java-Interfaces generiert werden.



aus Hammerschall: *Verteilte Systeme und Anwendungen*

Links zu Standards

W3C: XML Specification. <http://www.w3.org/TR/REC-xml/>

W3C: WSDL Specification. <http://www.w3.org/TR/wsdl>

W3C: SOAP Specification. <http://www.w3.org/TR/soap/>

JCP: JAX-RPC. <http://java.sun.com/xml/downloads/jaxrpc.html>

Apache Axis Projekt. <http://ws.apache.org/axis/>