

Webservices

REST vs. SOAP



Amine El Ayadi
INF-M2 Anwendungen 1 (SS 2008)
Department Informatik
HAW Hamburg
17. Juni 2008

Agenda

- Einführung & Motivation
- Webservices
- SOAP Webservices
- REST Webservices
- REST vs. SOAP Webservices
- Ausblick
- Quellen

Einführung & Motivation

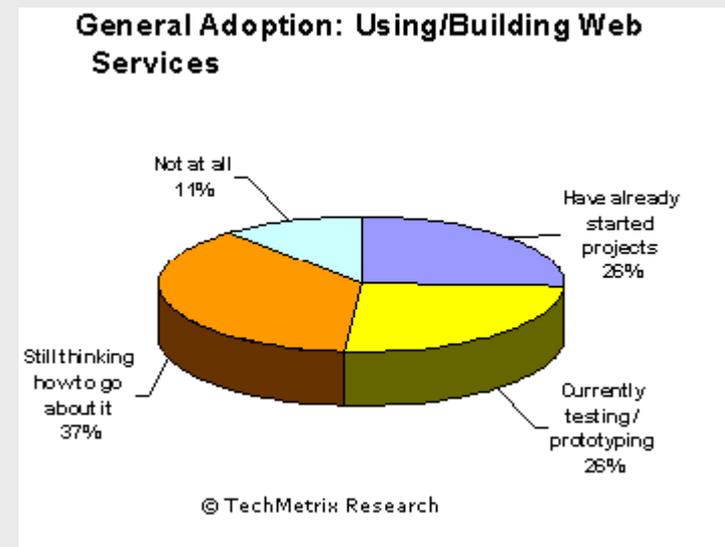
Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- ***SOA (Service Oriented Architecture)***
 - Architektur für die Integration unterschiedlicher Softwaresysteme in Unternehmen
 - Grundlegende serviceorientierte Sichtweise auf IT-Architekturen
 - Anforderung
 - Programmiersprachen unabhängig sein
 - Geringe annahmen
 - *Lose Kopplung* notwendig
- ❖ ***Webservices*** kann man als Umsetzung dieser Perspektive mit Internet-Technologien betrachten

Einführung & Motivation

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- Verwendung/Aufbau-Webservices
- 26 %: Webservices werden bereits eingesetzt
- 26 %: Webservices befinden sich in der Projektphase
- 37 %: Webservices werden in Erwägung gezogen
- 11 %: haben sich noch nicht mit diesem Thema auseinandergesetzt



Verwendung/Aufbau-Webservices[HiWS]

Webservices

Agenda | Einführung & Motivation | **Webservices** | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Webservices - Idee

- Wohldefinierte Schnittstellen sollen den Zugriff auf Daten erleichtern
- Idee solcher Schnittstellen ist aber nicht neu: Sun-RPC, COM+, CORBA
- Neu dabei ist: standardisierter Austausch mit Hilfe von in XML kodierten Daten über bekannte Datenübertragungsprotokolle (http, https, smtp, ...)

Webservices

Agenda | Einführung & Motivation | **Webservices** | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Webservices - worum geht es?

- Ein Webservice ist eigentlich nicht viel mehr als eine dynamische Webseite
- Der Unterschied besteht darin, dass normale Webseiten die Sprache HTML verwenden und sie normalerweise ein Browser dem Anwender präsentiert. Im Unterschied dazu liefert ein Webservice XML-Daten, die nicht für die unmittelbare Anzeige gedacht sind, sondern von einem speziellen Client-Programm weiterverarbeitet werden ...
[14]

Webservices

Agenda | Einführung & Motivation | **Webservices** | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Webservices - Entwicklung

- 1998: Dave Winer veröffentlicht XML-RPC
- 1999: SOAP 1.0 veröffentlicht
- 2000: W3C-Konsortium (Microsoft, IBM, Sun ...) verabschieden SOAP-Spezifikation (Simple Object Access Protocol)
- 2000: REST (Representational State Transfer) wird von Roy Fielding in seiner Dissertation beschrieben
- 2001: WSDL (Web Services Description Language)
- 2002: Google und Amazon veröffentlichen Webservice-Schnittstellen
- 2003: SOAP Version 1.2
- 2005: Yahoo! veröffentlicht Webservice-Schnittstelle

SOAP Webservices

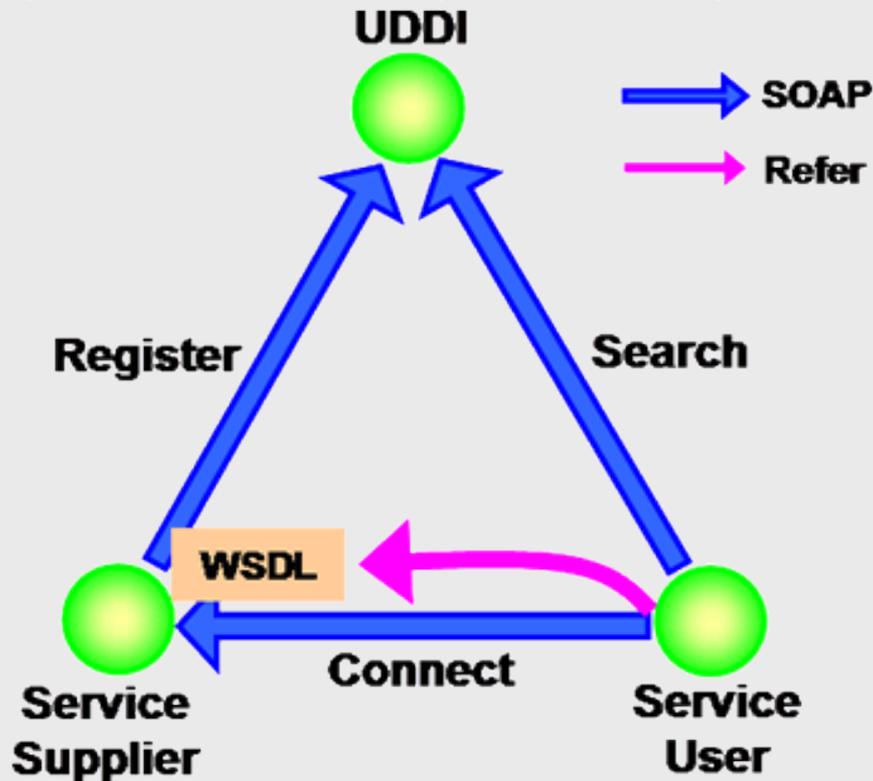
Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ SOAP Webservices

- werden beschrieben
 - durch **WSDL** (Web Service Description Language)
- werden bekannt gemacht
 - mittels **UDDI** (Universal Description, Discovery and Integration)
- werden aufgerufen
 - über **SOAP** (Simple Object Access Protocol)

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

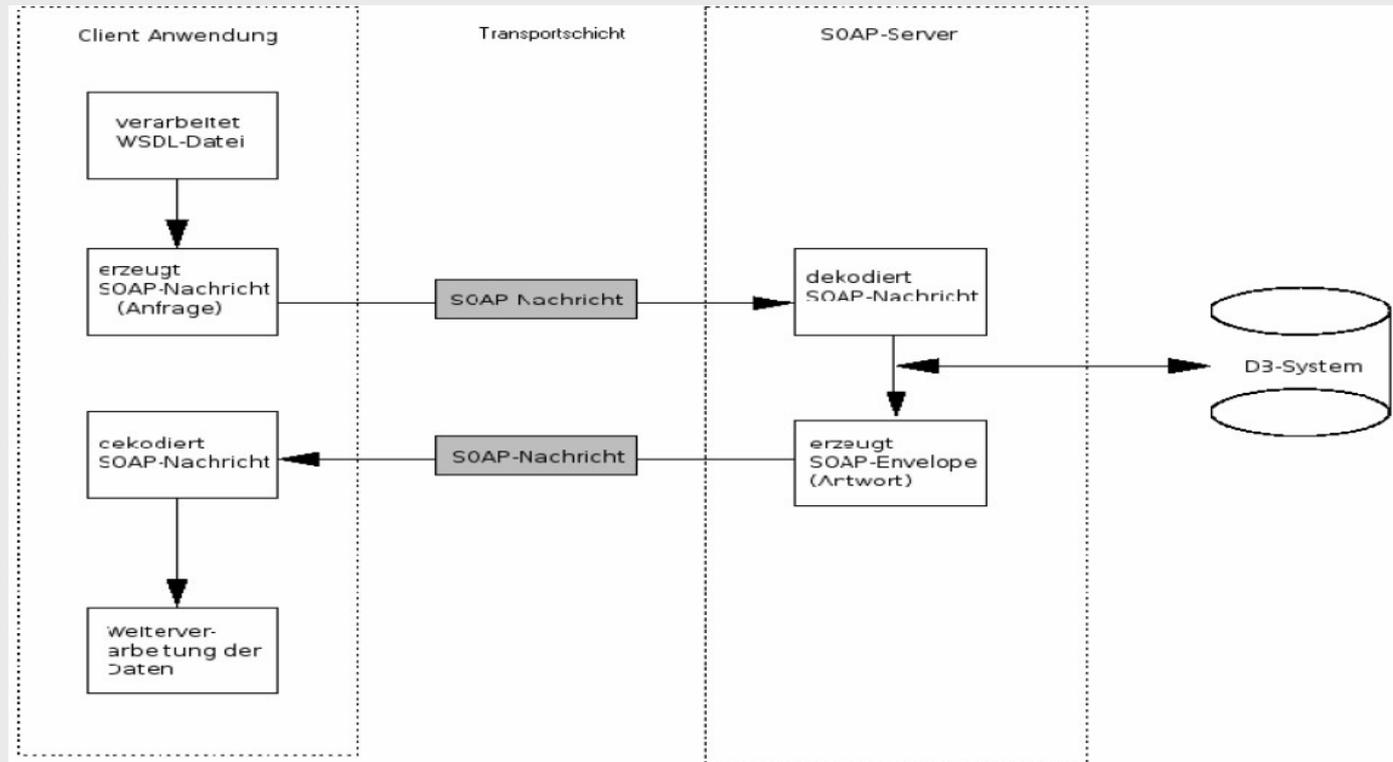


Quelle: Web Services [15]

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

SOAP Webservices – Arbeitsweise SOAP/WSDL [14]



SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **SOAP (Simple Object Access Protocol)**
 - Netzwerkprotokoll zum Austausch XML-basierte Nachrichten über ein Computernetzwerk
 - Kodierung der Nachricht in XML
 - Transportiert XML-Nachrichten(Dokumente) über standardisierte Internet-Technologien: HTTP, SMTP

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **SOAP (Simple Object Access Protocol)**
 - Beschreibung des Service mittels Web Service Description Language (WSDL)
 - Stellt eine Standard-Verpackungsstruktur bereit
 - Nachfolger von XML-RPC
 - Einfache Lesbarkeit, aber viel Overhead
 - Größere Software-Firmen unterstützen SOAP

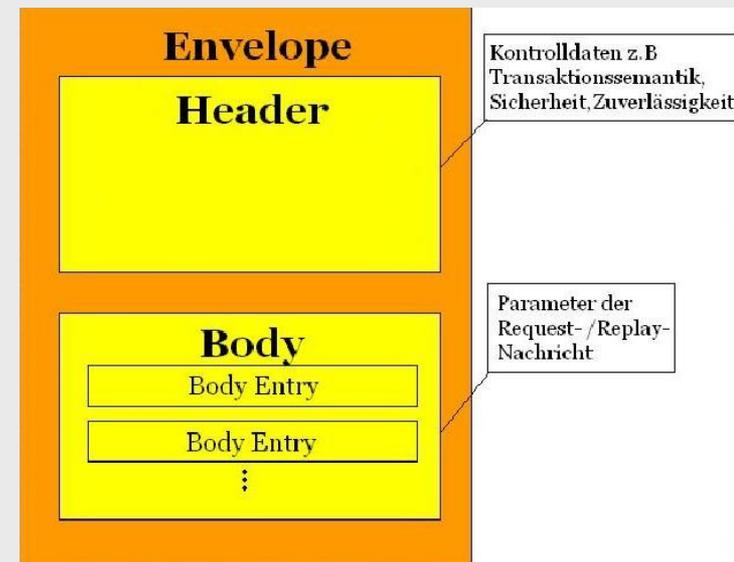
SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Aufbau einer SOAP-Nachricht

□ SOAP Messages enthalten einen **Envelope**, einen **Header** und einen **Body**.

- **Envelope (Umschlag)**
- **Header (Kopf)**
- **Body (Rumpf)**



SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ SOAP Request - Beispiel

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <getProductDetails xmlns="http://warehouse.example.com/ws">  
      <productID>827635</productID>  
    </getProductDetails>  
  </soap:Body>  
</soap:Envelope>
```

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ SOAP Response - Beispiel

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productID>827635</productID>
        <description>3-Piece luggage set. Black Polyester.</description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ WSDL (Web Service Description Language)

- XML-Derivat zur Beschreibung der Schnittstellen von Webservices
- Welche Methoden der Webservice anbietet
- Mit welchen Parametern die Methoden aufzurufen sind und was sie zurückliefern
- Wie nimmt jemand Kontakt zu meinem Webservices auf
- Wie müssen die Anfragen formatiert werden

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ WSDL Anwendung [5]



SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ UDDI (Universal Description, Discovery and Integration)

- Standardisierter Verzeichnisdienst (Veröffentlichung und Entdeckung von Services)
- Weltweite Registrierungsstelle von Webservices für die Ankündigung- Beschreibung von Diensten
- Entdeckung - Auffinden von Diensten
- Integration - Verwalten von Diensten
- Registrierungsstelle (über viele Knoten verteilt, replizieren)
- Enthält Daten und Services von Unternehmen
- Besitzt eine SOAP-Schnittstelle

SOAP Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ SOAP Werkzeuge

- Axis
- XSUL (auch bekannt als XSOAP4)
- WebSphere Integration Developer (WID)
- Microsoft Visual C# 2005 Express Edition
- WebSphere Studio Application Developer Integration Edition
- Rational Application Developer for WebSphere Software

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **REST (Representational State Transfer)**
 - REST: Alternativer Ansatz zu SOAP/WSDL (auch RESTful Webservices)
 - Eigentliche Bedeutung
 - Kein Standard, eher ein **Architekturstil**
 - Beschreibt, wie man Standards anwendet
 - Idee des Webs
 - Kommunikation per HTTP
 - Kommandos: GET, POST, PUT, DELETE
 - Erreichbarkeit mittels URI
 - Mögliche Operationen/Dienste als Links
 - Oft gebrauchte, lose Bedeutung
 - Webservices mit HTTP/XML, aber ohne SOAP/WSDL

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Merkmale einer REST Anwendung

- Client Server: pull-based Interaction: Konsumenten „ziehen“ vom Server Informationen
- Stateless: jede Anfrage muss alle Informationen beinhalten
- Cache: Netzwerkkommunikation gering zu halten
- Uniform Interface: einheitlicher Zugriff auf Resources per: HTTP POST, PUT, GET...
- Namentliche Ressourcen: System aus Resource-Konzepten, die per URI adressiert sind [5]
- Verlinkung von Resources: ermöglicht dem Client das Springen von einem «State» zum nächsten
- In Schichten aufgebaute Komponenten: Cache oder Proxy für Performance und Sicherheit [5]

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Rest- Dokumentwechsel mittels XLink**
 - Mit jedem Dokument kann der Client tiefer in die Anwendung einsteigen
 - Als Einstiegspunkt reicht eine einzige URL aus
 - Die Idee der aus dem Web bekannten Hypertext-Dokumente wird so auf Webservices übertragen [5]

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

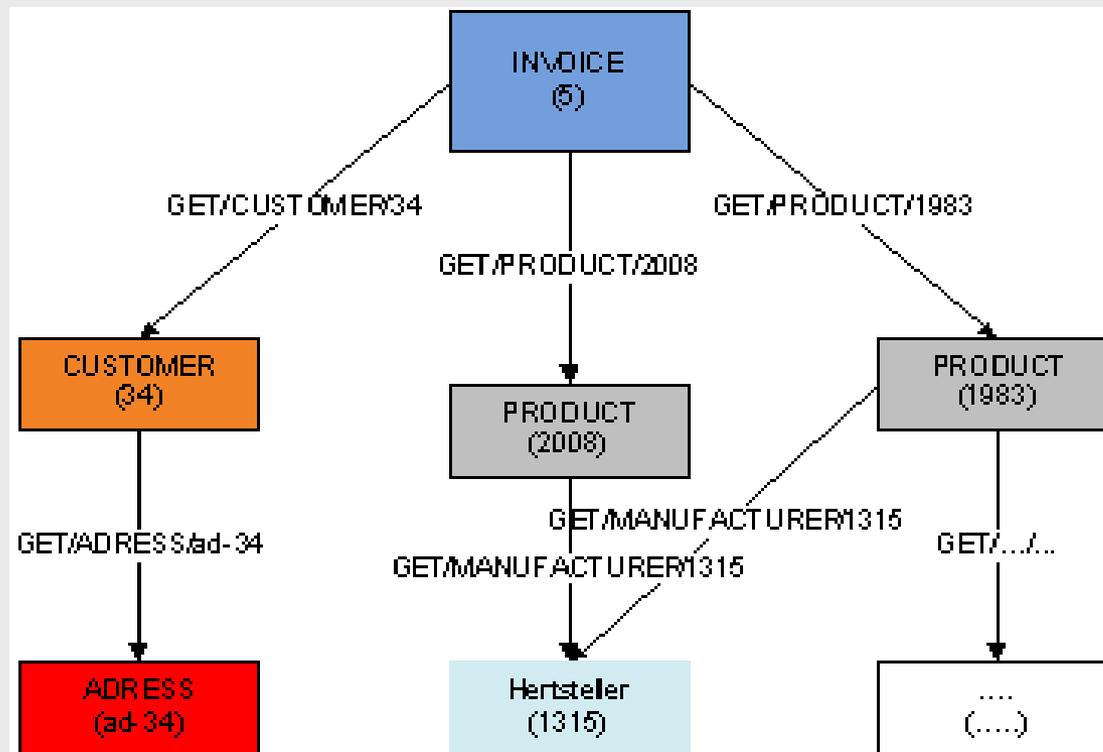
■ Rest- Kommunikationsprotokoll

- HTTP unterstützt einen festen Satz von vier Operationen:
 - **GET**: Mit GET fordert der Klient Daten vom Server ab
 - **POST**: Mit POST schickt der Klient neu anzulegende Daten an den Server
 - **PUT**: Mit PUT aktualisiert der Klient bereits existierende Daten auf dem Server
 - **DELETE**: Mit DELETE löscht der Klient Daten auf dem Server

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Rest- Navigation über Links u. HTTP GET [5]



REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ REST- Beispielanwendung (1)

□ Request:

GET /rest/INVOICE (http://localhost:8080/rest/INVOICE/)

□ Responce:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<INVOICEList xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/1/">1</INVOICE>
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/2/">2</INVOICE>
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/3/">3</INVOICE>
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/4/">4</INVOICE>
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/5/">5</INVOICE>
```

```
  <INVOICE xlink:href="http://localhost:8080/rest/INVOICE/6/">6</INVOICE>
```

```
</INVOICEList>
```

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ REST- Beispielanwendung (2)

□ Request

GET /rest/INVOICE/5 (http://localhost:8080/rest/INVOICE/5)

□ Responce

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<INVOICE xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <ID>5</ID>
```

```
  <CUSTOMERID xlink:href="http://localhost:8080/rest/CUSTOMER/34/">34</CUSTOMERID>
```

```
  <POSITION nr="1" quantity="5">
```

```
    <PRODUCT xlink:href="http://localhost:8080/rest/PRODUCT/1983" nr="4501">
```

```
      <DISCRIPTION>Beschreibung Artikel 1</DISCRIPTION>
```

```
    </PRODUCT>
```

```
  </POSITION>
```

```
  <POSITION nr="2" quantity="2"> ..... </POSITION>
```

```
  <TOTAL>20.80</TOTAL>
```

```
</INVOICE>
```

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Vorteile von REST [7]**
 - Skalierbarkeit
 - Zuverlässigkeit
 - Anbindung von Fremdsystemen
 - Unabhängig installierbare Komponenten
 - Komposition von Diensten

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Vorteile von REST - Skalierbarkeit

- Da alle Interaktionen in REST statuslos sind, müssen die Dienste keinen Status speichern und können somit Speicherplatz sparen
- Der enorme Wachstum des World Wide Web hat gezeigt, in welchem Ausmaß Webtechnologien skalieren

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ Vorteile von REST - Zuverlässigkeit

- Das Fehlen eines Zustandes erhöht ebenso die Zuverlässigkeit eines Dienstes, da dadurch der Wiederherstellungsprozess erleichtert wird.
- Ein fehlgeschlagener Dienst kann schneller gestartet werden, weil keine Rücksicht auf einen Zustand genommen werden muss.

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Vorteile von REST - Anbindung von Fremdsystemen**
 - Über verschiedene Gateways kann auf eine Fülle von Systemen zugegriffen werden
 - Die Details von Fremdsystemen werden hinter Schnittstellen versteckt

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Vorteile von REST - Unabhängig installierbare Komponenten**
 - Unabhängiges “Deployment” von Komponenten
 - Für sehr große Systeme ist ein unabhängiges Deployment eine Grundvoraussetzung
 - Stabilität der Kommunikation zwischen Komponenten
 - Gegenüber Änderungen tolerante Schnittstellen

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Vorteile von REST - Komposition von Diensten**
 - Einzelne REST-Services können leicht zusammen genutzt werden
 - Über den globalen und universellen Adressraum der URLs können die Grenzen einer Anwendung leicht überschritten werden
 - Ein Dokument verweist einfach auf eine Ressource, die sich in einer anderen Organisation befindet

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- **Wie realisiere ich einen REST WS?**
 - Ressourcen identifizieren, URIs designen
 - Formate auswählen (D.h. welche Repräsentation die Ressource hat (HTML, HTTP))
 - Methodensemantik beschreiben
 - Response-Codes abbilden (404, 403, 500...), von denen es in HTML schon für eigentlich jeden denkbaren Anwendungsfall einen gibt. Bspw. “Methode ist nicht mehr unter dieser URI erreichbar”

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ REST Best Practise [5]

- Schema für jede Art der Repräsentation anbieten
- XML als Datenformat verwenden
- XLink für Verweise
- Bestehende XML Formate verwenden
 - Wenn möglich Format für Semantisches Web (RDF, RSS, FOAF, DAML, OIL)

REST Webservices

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

■ REST-Werkzeuge

- Java
- HTTP-Client
- Restlet
- Xins
- Crispy

REST vs. SOAP

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

	SOAP	REST
Schnittstellen beschreibung	WSDL	keine, generisch
Adressmodell	URI	URI
Schnittstelle	Anwendung spezifisch	generisch (GET, POST, ...)
Discovery	UDDI	Generische Schnittstellen & Verweise
Status	(Server) / Client	Im Client
Akzeptanz in der Industrie	Hoch	unbewusst
Werkzeuge	Unüberschaubar	wenige spezielle , viele Standardtools
Transport	HTTP, SMTP, JMS, ...	HTTP
Standards	W3C u.a.	Ist ein Stil
Performance	< 30 ms	< 3 ms – beliebig

Quelle: REST vs. SOAP [5]

Ausblick

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | **Ausblick** | Quellen

- REST
 - Einfach zu erlernen und umzusetzen und funktioniert auch in einem maximal heterogenen Umfeld
 - Verwendet ausschließlich Standards wie HTTP oder URIs
 - Macht sich die Ideen des Webs wie Hypertext und einen unbegrenzten globalen Adressraum zunutze
 - Damit sind REST basierte Web Services wirkliche Web Services im Gegensatz zu RPC basierter Middleware
- SOAP:
 - Umfangreiche Tool-Unterstützung mind. für den RPC-Stil
 - Macht SOAP interessant
 - Verteilte Transaktionen zwischen den Systemen werden unterstützt

Ausblick

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | **Ausblick** | Quellen

Die Entscheidung für REST oder SOAP ist sicher nicht einfach

■ SOAP

- Wenn es um Prozess-Steuerung und –Monitoring geht
- Wenn im Unternehmen verteilte Transaktionen zwischen den Systemen benötigt werden

■ REST

- Für maximale Performance und Skalierbarkeit für sehr viele parallele Benutzer

Quellen

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- [1] Web Services Platform Architecture, Weerawarana u.a., Prentice Hall PTR (2005)
- [2] http://www.developer.com/java/web/article.php/10935_2207371_1
(letzter Zugriff: 15.06.08)
- [3] http://www.oio.de/public/xml/rest_webservices.htm (letzter Zugriff: 15.06.08)
- [4] <http://www.extreme.indiana.edu/xgws/xsul/> (letzter Zugriff: 15.06.08)
- [5] <http://www.oio.de/m/soap-xmlrpc-rest/index.htm> (letzter Zugriff: 15.06.08)
- [6] <http://msdn.microsoft.com/vstudio/express/default.aspx> (letzter Zugriff: 15.06.08)
- [7] http://elib.uni-stuttgart.de/opus/volltexte/2006/2618/pdf/FACH_0056.pdf
(SOA vs. REST basierend auf Google, eBay, Amazon, Yahoo!)
(letzter Zugriff: 15.06.08)
- [8] <http://ws.apache.org/axis/java/reference.html> (letzter Zugriff: 15.06.08)

Quellen

Agenda | Einführung & Motivation | Webservices | SOAP Webservices |
REST Webservices | REST vs. SOAP | Ausblick | Quellen

- [9] <http://jakarta.apache.org/commons/httpclient/> (letzter Zugriff: 15.06.08)
- [10] <http://www.restlet.org/> (letzter Zugriff: 15.06.08)
- [11] <http://xins.sourceforge.net/> (letzter Zugriff: 15.06.08)
- [12] <http://crispy.sourceforge.net/> (letzter Zugriff: 15.06.08)
- [13] <http://crispy.sourceforge.net/index.html> (letzter Zugriff: 15.06.08)
- [14] Web Services – Einsatzmöglichkeiten für das Information Retrieval im WWW Fabio Tosques & Philipp Mayr
- [15] <http://www.nec.co.jp/middle/WebOTX-e/function/webservice.html> (letzter Zugriff: 15.06.08)
- [HiWS] Hiccup In Web Services Adoption? 04/14/2005 By Jean-Christophe Cimetiere, TechMetrix Research

Ende

Vielen Dank für die Aufmerksamkeit!