



Web Services mit JAVA

Anelia Mircheva
Moritz Steiner

Teleseminar Web Services (SS 04)
Universität Karlsruhe (TH) /
Universität Mannheim

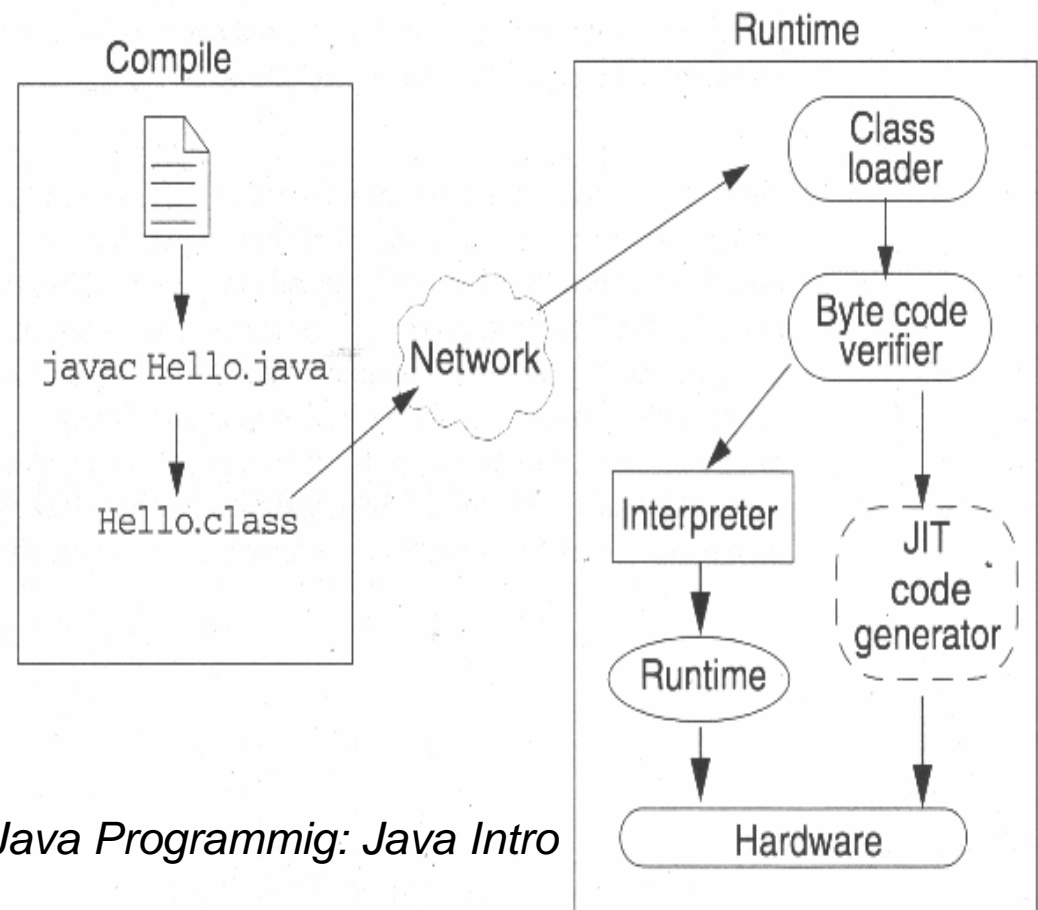


Agenda

- JAVA allgemein
- JWSDP und die APIs
- Apache AXIS
- UDDI mit Java
- Unterschiede Java - .NET

JAVA allgemein

- Objekt-orientiert & verteilt
 - Objekt
 - Umfangreiche Klassenbibliotheken
- Plattformunabhängig und portierbar
 - Bytecode und die virtuelle Maschine
- Robust & sicher



Quelle: Java Programmig: Java Intro



Java und Web services

- Anforderungen an Web services
 - Kommunikation zwischen verschiedenen Informationssystemen
 - XML
 - Kommunikation zwischen verschiedenen Computer Plattformen
 - leicht zu erstellen
 - skalierbar, sicher, effizient
 - Java APIs für XML und J2EE Plattform
- Java Web Services Developer Pack (JWS DP)



Agenda

- JAVA allgemein
- JWSDP und die APIs
- Apache AXIS
- UDDI mit Java
- (Vergleich Java .NET)



Was ist Java WSDP

- “A free integrated toolkit used to build and test XML applications, Web services, and Web applications with the latest Web service technologies and standards implementations”;
- Größe: 26M; Plattform: Windows, Unix;

Quelle: JWSDP Overview by Sun Microsystems

JWSDP 1.3:

<http://java.sun.com/webservices/downloads/webservicespack.html>

JWS Tutorial:

<http://java.sun.com/webservices/downloads/webservicestutorial.html>

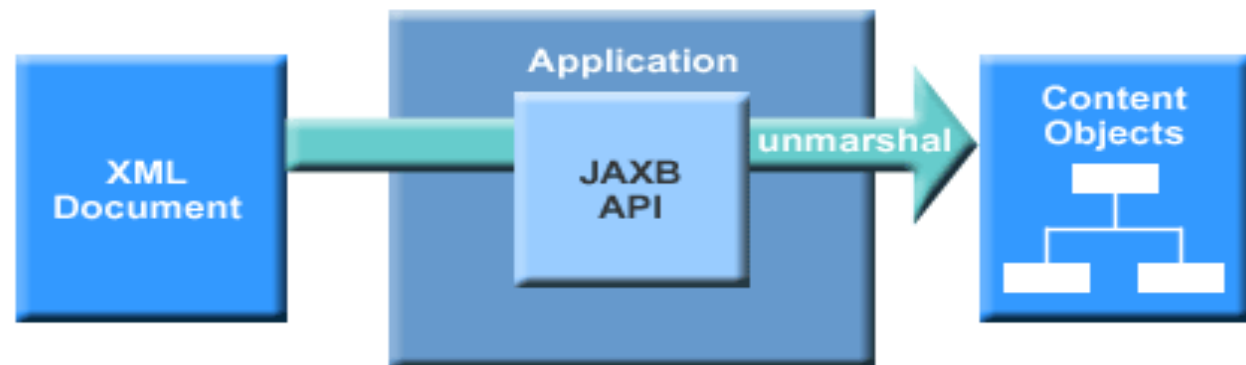


Was ist im JWSDP

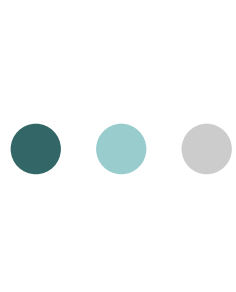
- Java Architecture for XML Binding (JAXB) v1.0.2;
- Java API for XML Processing (JAXP) v1.2.4;
- Java API for XML-based RPC (JAX-RPC) v1.1;
- SOAP with Attachments API for Java (SAAJ) v1.2;
- Java API for XML Registries (JAXR) v1.0.5;
- JavaServer Faces (JSF) v1.0 EA4;
- JavaServer Pages Standard Tag Library (JSTL) v1.1 EA;
- XML and Web Services Security v1.0 EA2;
- Java WSDP Registry Server v1.0_06;
- Ant Build Tool 1.5.4;
- Apache Tomcat v5 development container;
- Ws-I Supply Chain Management Sample Application 1.0;

Java Architecture for XML Binding (JAXB)

- Java Architecture for XML Binding (JAXB)
 - Mapping zwischen XML Dokumente und Java Objekte;
 - Compilieren eines XML Schemas in eine oder mehr Java Klassen;
 - Die folgende Operationen können ausgeführt werden:
 - **unmarshal** von XML Inhalt in eine Java Repräsentation;



Quelle: **JAXB** by Ed Ort and Bhakti Mehta



Java Architecture for XML Binding (JAXB): unmarshaling

//Erstellen eines JAXBContext Objektes.

```
import javax.xml.bind.JAXBContext;  
JAXBContext jc = JAXBContext.newInstance("test.jaxb");
```

//Erstellen eines Unmarshaller Objektes.

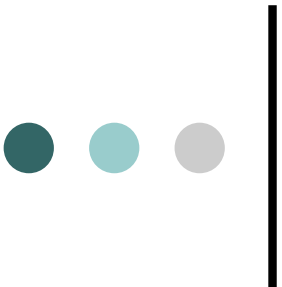
```
import javax.xml.bind.Unmarshaller;  
Unmarshaller unmarshaller = jc.createUnmarshaller();
```

//Aufruf der unmarshal Methode.

```
Collection collection= (Collection) unmarshaller.unmarshal(new  
File( "books.xml"));
```

//Verwende die get Methoden zum Zugriff zur XML Datei.

```
CollectionType.BooksType booksType = collection.getBooks();  
List bookList = booksType.getBook();
```



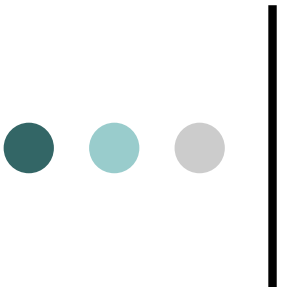
Java Architecture for XML Binding (JAXB): other operations

- **access** ein XML Dokument (in zwei Schritte)
- Binden das Schema für das XML Dokument.
 - Alle JAXB Implementationen liefern ein tool, binding compiler genannt, für binding das Schema
- Unmarshal das Dokument in inhaltliche Java Objekte
 - **update** an XML document
- Aktualisieren eines unmarshalled inhaltlichen Baumes und dann marshals ihn zurück zu einem XML Dokument
- Direkter Zugriff zu dem Objekt, das aktualisiert werden muss
- get Methode für Zugriff und set Methode zur Aktualisierung der Datei
 - **marshal** die Java Repräsentation in XML Inhalt.
- der Gegensatz von unmarshalling.



Java API for XML Processing (JAXP)

- unterstützt Bearbeitung von XML Dokumenten mittels DOM, SAX, und XSLT
 - Document Object Model (DOM):
 - Hierarchie von Objekten repräsentiert
 - Baumstruktur, die die Struktur von dem XML Dokument imitiert
 - Simple API for XML (SAX):
 - Serie von Ereignissen(events)
 - Jedes Ereignis repräsentiert eine Transition im XML Dokument
 - XSL Transformations (XSLT):
 - Sprache für Transformierung von XML Dokumenten in anderen XML(oder HTML) Dokumenten;



Java API for XML Processing (JAXP): DOM model

- Bereitstellen einer Instanz von `DocumentBuilderFactory`.
- Bereitstellen einer Instanz von `DocumentBuilder`

```
DocumentBuilder builder;
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
String location = "http://myserver/mycontent.xml";
try {
    builder = factory.newDocumentBuilder();
    Document document = builder.parse(location);
} catch (SAXException se) {
    // handle error
} catch (IOException ioe) {
    // handle error
} catch (ParserConfigurationException pce) {
    // handle error
}
```



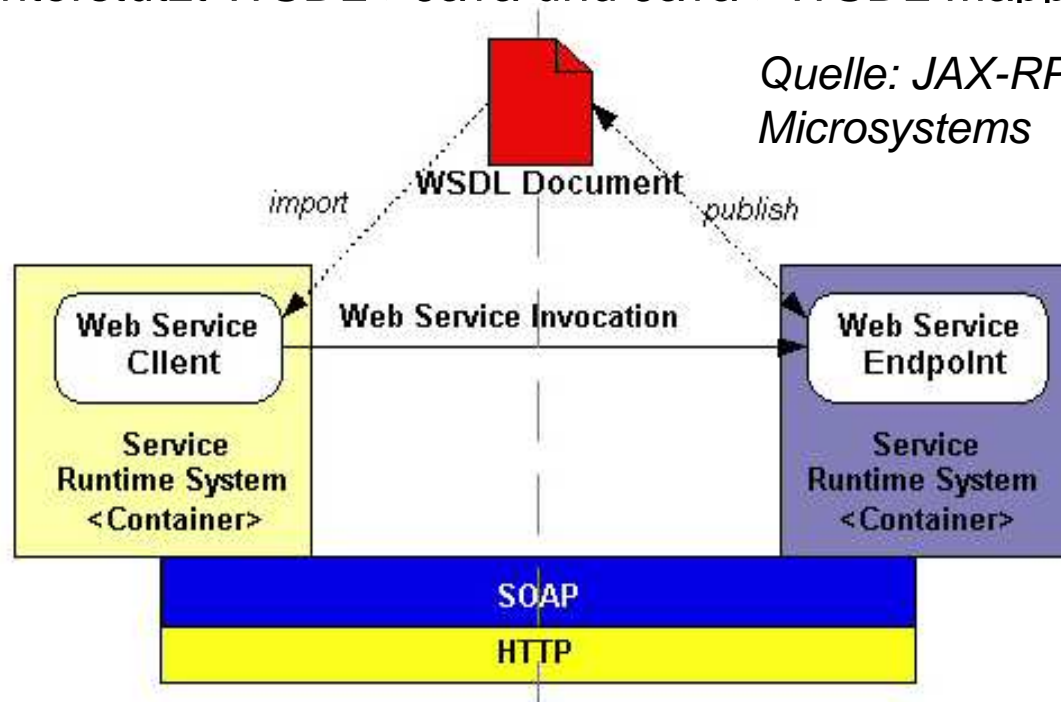
Java API for XML Processing (JAXP): SAX model

Parsing mit SAX benötigt drei Sachen: event handler, eine parser Instanze, und ein XML Dokument zum parsing.

```
SAXParser parser;  
DefaultHandler handler = new MyApplicationParseHandler();  
SAXParserFactory factory = SAXParserFactory.newInstance();  
try {  
    parser = factory.newSAXParser();  
    parser.parse("http://myserver/mycontent.xml", handler);  
} catch (SAXException se) {  
    // handle error  
} catch (IOException ioe) {  
    // handle error  
} catch (ParserConfigurationException pce) {  
    // handle error  
}
```

Java API for XML-Based RPC (JAX-RPC)

- RPC Mechanismus
 - Distributed client/server model
- JAX-RPC, SOAP und HTTP
- unterstützt WSDL->Java und Java->WSDL Mapping



Quelle: JAX-RPC Overview by Sun Microsystems



SOAP with Attachments API for Java (SAAJ)

- einheitliche Methode zum Senden von XML Dokumenten durch Internet von der Java Plattform
- Was ist eine SOAP Meldung (message):
 - I. SOAP message
 - A. SOAP part
 1. SOAP envelope
 - a. SOAP header (beliebig)
 - b. SOAP body

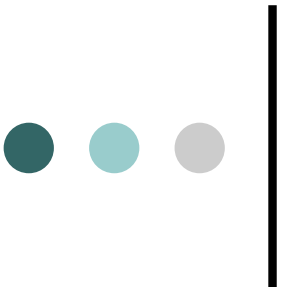
In der SAAJ API:

SOAPMessage Klasse -> SOAP message

SOAPPart Klasse -> SOAP part

SOAPEnvelope Schnittstelle -> SOAP envelope

u.s.w.



SOAP with Attachments API for Java (SAAJ)(2)

- *Getting a Connection*

```
SOAPConnectionFactory factory=SOAPConnectionFactory.newInstance();  
SOAPConnection connection = factory.createConnection();
```

- *Creating a Message*

```
MessageFactory messageFactory = MessageFactory.newInstance();  
SOAPMessage message = messageFactory.createMessage();
```

- *Populating a Message*

```
SOAPPart soapPart = message.getSOAPPart();  
SOAPEnvelope envelope = soapPart.getSOAPEnvelope();  
SOAPBody body = envelope.getSOAPBody();  
SOAPBodyElement bodyElement =  
    body.addBodyElement( envelope.createName("text", "hotitems",  
        "http://hotitems.com/products/gizmo");  
bodyElement.addTextNode("some-xml-text");
```

- *Sending a Message*

```
SOAPMessage response = soapConnection.call(message, endpoint);
```




Java API for XML Registries (JAXR)

- Registers
 - beinhalten Information über Web services und deren Anbieter.
 - Information ist kategorisiert anhand von dem Industriebereich und der geographischen Lage.
 - UDDI und ebXML
- Verbindung zu einem Register herstellen
 - Die *ConnectionFactory* Klasse in JAXR API
- Abfragen das Register
 - anhand von Organisation-name-pattern, oder anhand von Klassifizierung (nach Industriebereich oder Lage)
 - Das *javax.xml.registry* packet liefert viele Methoden, genau wie *findOrganizations()*, *findServices()*, *findConcepts()*, u.s.w., zum Abfragen ein Register.



Java API for XML Registries (JAXR)(2)

- Speichern von Information im Register
 - Die *BusinessLifeCycleManager* Schnittstelle in dem *javax.xml.registry* Packet
 - Benutzername und Kennwort
 - Aktualisierung der vorhandene Datensätze
 - *deleteServices()*;
 - *saveServices()*;
 - Eine in dem Register vorhandene Organisation kann durch die *deleteObjects()* Methode entfernt werden, oder durch *deprecateObjects()* Methode abgelehnt werden.



Zusammenfassung

- nothing specific to Java in Web services?
- Web services can be written in languages as varied as Java, C++, C#, and Perl
- features that make Java attractive for general server programming - the same ones that make Java attractive when writing new Web-service-based systems as well.
 - rich libraries
 - straightforward execution model
 - and portability
- variety of libraries, each presenting a slightly different API and functionality.
- Web service standards such as SOAP, UDDI, and ebXML appeared (being key for next-generation Web-based applications), but had no direct support in Java.



Agenda

- JAVA allgemein
- APIs
- Apache AXIS
- UDDI mit Java
- Unterschiede Java - .NET



Was ist AXIS?

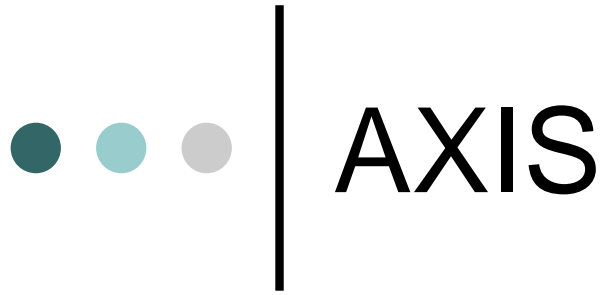
Apache eXtensible Interaction System

SOAP-Engine: Java-Framework für
SOAP verarbeitende Komponenten



AXIS abstrakt

- Entstehung
- Architektur
- Deployment
- Interoperabilität
- WSDL Unterstützung
- Anforderungen
- Sicherheit
- Tools



Serverseite



Calculator.java

```
12 public class Calculator {
13
14     Complex a = new Complex(5, 4);
15
16     public Complex getA() {
17         return a;
18     }
19     public void setA(Complex a) {
20         this.a = a;
21     }
22
23     public void add(int a, int b) {
24         this.a.add(new Complex(a, b));
25     }
26
27 }
```




Complex.java

```
14 public class Complex {
15     double real;
16     double imag;
17
18     public Complex(double a, double b){
19         real=a;
20         imag=b;
21     }
22
23     public void add(Complex b){
24         this.real+=b.real;
25         this.imag+=b.imag;
26     }
27
28
29     public double getImag() {
30         return imag;
31     }
32     public double getReal() {
33         return real;
34     }
35     public void setImag(double imag) {
36         this.imag = imag;
37     }
38     public void setReal(double real) {
39         this.real = real;
40     }
41 }
```



Instant Deployment

```
copy Calculator.java <your-webapp-root>  
/axis/Calculator.jws
```

Das war's!

Web Service nun ansprechbar unter:

```
http://localhost:8080/axis/Calculator.jws
```



Custom Deployment

- Instant Deployment begrenzt Konfigurationsmöglichkeiten stark
- volle Flexibilität nur mit *Web Service Deployment Descriptor (WSDD)*



Java 2 WSDL

Java2WSDL Kommandozeilenaufruf

```
org.apache.axis.wsdl.Java2WSDL  
-o Calculator.wsdl  
-l "http://localhost:8080/...  
/Calculator"  
-n "CalculatorService"  
-p "calculatorwebservice"  
calculatorwebservice.Calculator
```

DEMO



WSDL 2 Java

```
java org.apache.axis.wsdl.WSDL2Java  
  --server-side <wsdl-file-URL>
```

- für jedes `<service>`: die Dateien `deploy.wsdd` und `undeploy.wsdd`

```
java org.apache.axis.client.AdminClient  
  deploy.wsdd
```

- Weitere Dateien die nur für den Client interessant sind

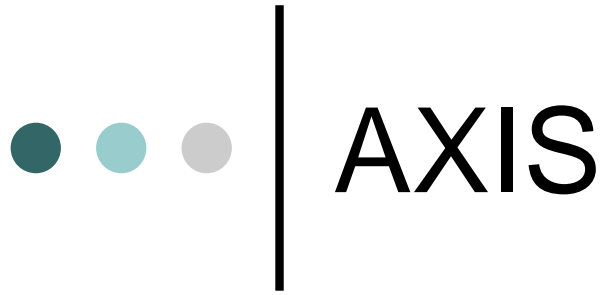
DEMO



WSDD

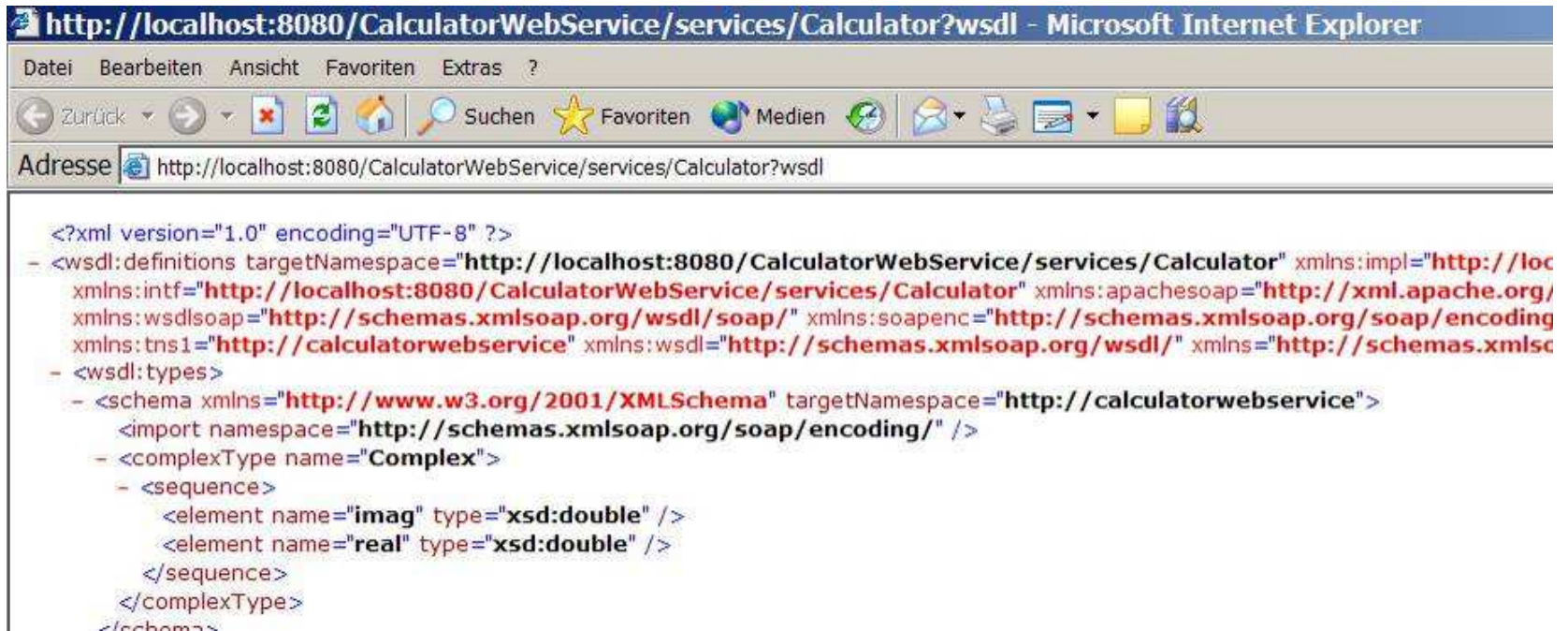
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <deployment
3     xmlns="http://xml.apache.org/axis/wsdd/"
4     xmlns:ns="http://calculatorwebservice"
5     xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
6     <service name="Calculator" provider="java:RPC">
7         <parameter name="className" value="calculatorwebservice.Calculator"/>
8         <parameter name="allowedMethods" value="*" />
9         <parameter name="scope" value="Application" />
10
11     <typeMapping
12         xmlns:ns="http://calculatorwebservice"
13         qname="ns:Complex"
14         type="java:calculatorwebservice.Complex"
15         serializer="org.apache.axis.encoding.ser.BeanSerializerFactory"
16         deserializer="org.apache.axis.encoding.ser.BeanDeserializerFactory"
17         encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
18     />
19 </deployment>
```

```
java org.apache.axis.client.AdminClient deploy.wsdd
```



Clientseite

Java 2 WSDL



```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://localhost:8080/CalculatorWebService/services/Calculator" xmlns:impl="http://loc
  xmlns:intf="http://localhost:8080/CalculatorWebService/services/Calculator" xmlns:apachesoap="http://xml.apache.org/
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding
  xmlns:tns1="http://calculatorwebservice" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsc
- <wsdl:types>
  - <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://calculatorwebservice">
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    - <complexType name="Complex">
      - <sequence>
        <element name="imag" type="xsd:double" />
        <element name="real" type="xsd:double" />
      </sequence>
    </complexType>
  </schema>
```

DEMO



WSDL 2 Java

generiert Stubs, Skeletons und Datentypen

```
java org.apache.axis.wsdl.WSDL2Java  
  <wsdl-file-URL>
```



DEMO

CalculatorClient.java

```
17 public class CalculatorClient {
18     public static void main(String[] args) {
19
20         try {
21             Calculator service = new CalculatorServiceLocator().getCalculator();
22             Complex temp = null;
23             System.out.print("service.getA()=");
24             temp=service.getA();
25             System.out.println(temp.getReal()+" , "+temp.getImag());
26
27             System.out.println("service.add(new Complex(99,23))");
28             service.add(99, 23);
29
30             System.out.print("service.getA()=");
31             System.out.println(service.getA().getReal()+" , "+service.getA().getImag());
32         }
33         catch (RemoteException ex) {
34             System.out.println("ex=" + ex);
35         }
36         catch (ServiceException ex) {
37             System.out.println("ex=" + ex);
38         }
39     }
40 }
41 }
```

DEMO



Vorteile von AXIS

- Geschwindigkeit
- Flexible Konfiguration und Erweiterbarkeit
- Transport - Framework
- Stabilität
- WSDL Unterstützung über Code Generatoren
- JAX-RPC und SAAJ Konformität
- Erweiterungen für Sicherheit
- Provider für den Zugriff auf Enterprise JavaBeans
- ...



Agenda

- JAVA allgemein
- APIs
- Apache AXIS
- UDDI mit Java
- Unterschiede Java - .NET

UDDI Browser

The screenshot shows a Microsoft Internet Explorer browser window displaying the UDDI Business Registry search results. The address bar shows the URL <https://uddi.ibm.com/ubr/find>. The page title is "Find Service Results - Microsoft Internet Explorer". The browser's menu bar includes "Datei", "Bearbeiten", "Ansicht", "Favoriten", and "Extras". The toolbar contains navigation buttons like "Zurück", "Suchen", "Favoriten", and "Medien".

The main content area features the IBM logo and a search bar. Below the logo, there are navigation links: "Home", "Products & services", "Support & downloads", and "My account". The breadcrumb trail reads "IBM Corporation > Services/UDDI > Find".

The page title is "UDDI Business Registry Version 2" with the subtitle "Universal Description, Discovery, and Integration". The main heading is "Find Service Results". Below this, a message states: "Your query returned a total of 6 matching service(s). Press the **New Search** button to search again."

A table titled "Services" lists the following results:

Service Name	Description	Owning Business
Number Play	Routines manipulating numbers	SoapSample
NumberAdderService	This web service is just a test	Mark Walsh
NumberCalculator	None	Calculation
NumberGuess	Provides a simple number guessing game	Gama System
NumberToWords	This Number to words conversion utility can be used in accounting packages	HeodeS
NumberToWords	Converts a Number to its Word Counterpart	Vikramaditya Singh

DEMO



NumberTest.java

```
--  
15 public class NumberTest {  
16     public static void main(String[] args) {  
17         try {  
18             NumberToWordsSoap service = new NumberToWordsLocator().  
19                 getNumberToWordsSoap();  
20             System.out.println("service.inWordsIN(33.44)=" + service.inWordsIN(33.44));  
21         }  
22         catch (Exception ex) {  
23             System.out.println("ex=" + ex);  
24         }  
25     }  
26 }
```

DEMO



UDDI4j

```
56     UDDIProxy proxy = new UDDIProxy();
57     // Select the desired UDDI server node
58     proxy.setInquiryURL(config.getProperty("inquiryURL"));
59     proxy.setPublishURL(config.getProperty("publishURL"));
60     //creating vector of Name Object
61     Vector names = new Vector();
62     names.add(new Name("S"));
63     // Setting FindQualifiers to 'caseSensitiveMatch'
64     FindQualifiers findQualifiers = new FindQualifiers();
65     Vector qualifier = new Vector();
66     qualifier.add(new FindQualifier("caseSensitiveMatch"));
67     findQualifiers.setFindQualifierVector(qualifier);
68     // Find businesses by name
69     // And setting the maximum rows to be returned as 5.
70     BusinessList businessList = proxy.find_business(names, null, null, null, null, findQualifiers, 5);
71     Vector businessInfoVector = businessList.getBusinessInfos().getBusinessInfoVector();
72     for (int i = 0; i < businessInfoVector.size(); i++) {
73         BusinessInfo businessInfo = (BusinessInfo)businessInfoVector.elementAt(i);
74         // Print name for each business
75         System.out.println(businessInfo.getNameString());
76     }
```



Agenda

- JAVA allgemein
- APIs
- Apache AXIS
- UDDI mit Java
- Unterschiede Java - .NET



Unterschiede Java - .NET

- .NET Web Services können in allen .NET-Sprachen geschrieben werden
 - Aber: Microsoft-WS sind nicht ebXML-konform!
- Java Web Services sind nicht an einen Web Server oder einen Servlet- bzw. EJB-Container gebunden
 - Viele Open Source-Implementierungen verfügbar
 - Aber: Manche Toolkits arbeiten nur mit (teurem) EJB-Container
- Toolunterstützung in NET durchgängiger, dadurch Entwicklung einfacher
 - Toolkits für Java stark unterschiedliche Konzepte
- Web Services-Unterstützung für mobile Geräte in Java besser



AXIS und UDDI

Quellen

- Apache AXIS, **Documentation, Installation, User's Guide, Developer's Guide, Integration Guide, Architecture Guide, Reference Guide, Reading Guide, Requirements**, <http://ws.apache.org/axis/>
- Thilo Frotscher, **Advanced Web Services mit Apache Axis**, http://www.w-jax.de/konferenzen/w-jax03/powerworkshops/pw02_frotscher_1.pdf, 2003
- Torsten Langner, **Web services mit Java: Neuentwicklung und Refactoring in der Praxis**, Markt- und-Technik-Verlag, 2003
- www.uddi4j.org
- www.ibm.com/developerWorks UDDI



JWSDP und APIs für WS Quellen

- Java Web Services Tutorial

<http://java.sun.com/webservices/docs/1.3/tutorial/doc/index.html>

- Sample Applications:

<http://java.sun.com/webservices/docs/1.3/wsi-sampleapp/index.html>

- Developing Web Services with Java APIs for XML Using WSDP

(Joy Foster, Mick porter, Dreamtech Software, Inc., Natalie Wear)

- The Java Web Services Tutorial

(Eric Asmstrong, Stephanie Bodoff, Debbie Carson,...)

- Java Technologie and Web Services (by Sun Microsystems)

<http://java.sun.com/webservices/index.jsp>



Ende der Präsentation

Vielen Dank für Eure
Aufmerksamkeit!