

JEE5

Webservices

Agenda

- Einführung Webservices
 - SOA
 - SOAP/WSDL/UDDI
- Webservices mit JEE5
 - Überblick über die verschiedenen Techniken
 - Vertiefung in eine gängige Technik
- Über den Tellerrand: ColdfusionÜbungen

JEE 5

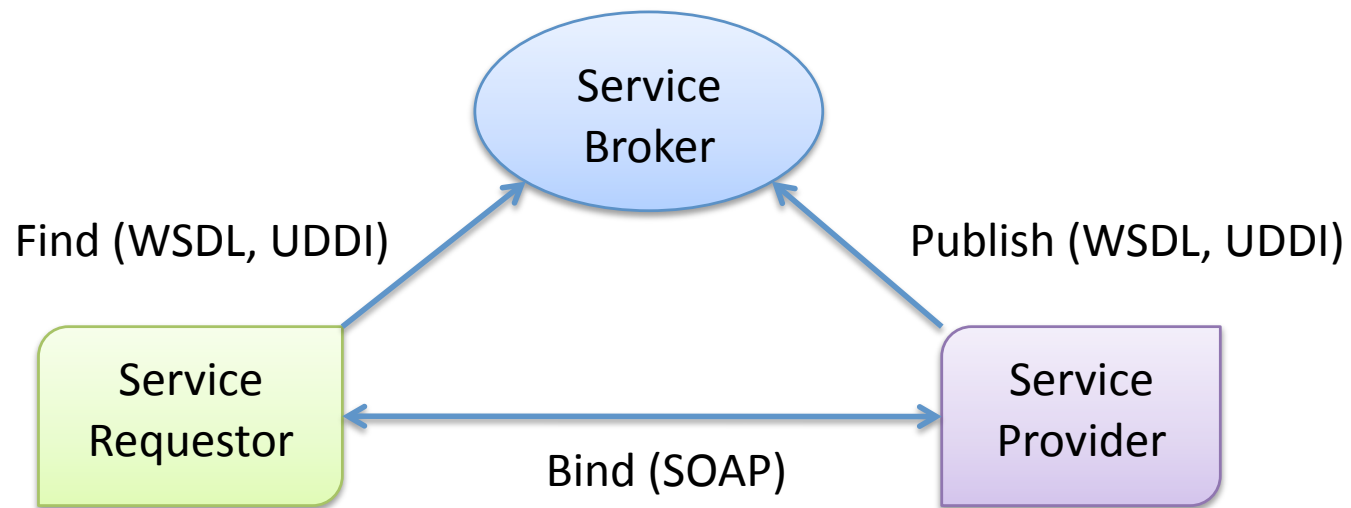
Webservices Einführung

Was sind Webservices?

- Dienste die auf XML-Standards basieren
 - Vereinfachen die Integration von Daten und Anwendungen
 - Ermöglichen eine Kooperation von Benutzern, Gruppen und Organisationen
 - Standardisierte plattformunabhängige Protokolle
- > XML Web Services ist ein Softwaredienst, der im Web über SOAP zur Verfügung gestellt, mit einer WSDL-Datei beschrieben und in der UDDI-Technologie registriert wird
[WS-VLS 07]

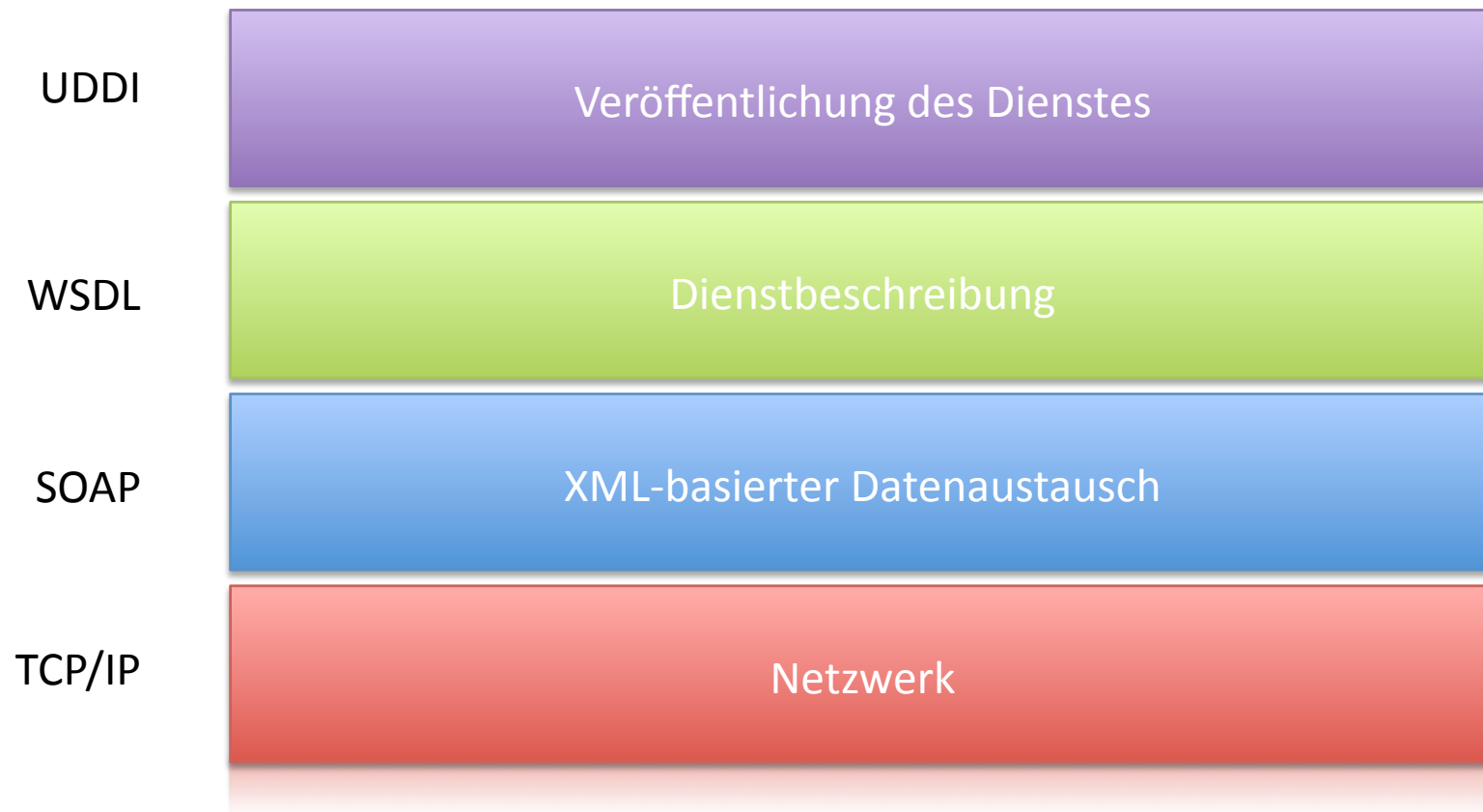
Was sind Webservices?

- Service-oriented architecture



Was sind Webservices?

- Webservice Protokollstack



Anwendungsgebiete von Webservices

- Integration von Anwendungen bzw. Anwendungskomponenten
 - Ziele der Integration:
 - B2B Anwendungen
 - EAI (Enterprise Application Integration)
 - Anwendungen, die keine lokale Installation benötigen, sondern über das Internet verfügbar gemacht werden
- > Technisch gesehen ist ein Webservice eine Softwarekomponente, die eine wohldefinierte Schnittstelle implementiert [WS-VLS 07]

Webservices im Unternehmen

- Rasche Entwicklung und Verwaltung von Web Services als ein Schlüssel für geschäftlichen Erfolg
- Einsatzmöglichkeiten:
 - Integration von Unternehmensanwendungen
 - Verbesserung von Business-to-Business-Diensten
 - Verbesserung von Business-to-Customer-Diensten
 - Unterstützung von Mitarbeitern im Außendienst
 - Entwickeln von effizienteren verteilten Anwendungen sowie Peer-to-Peer-Anwendungen

Webservices - Grundlagen

- XML-RPC
 - Nach Veröffentlichung der ersten XML Spezifikation: XML-Remote Procedure Calls (XML-RPC)
 - Idee: Methoden (Prozeduren) auf einem entfernten Rechner ausführen zu lassen und mit dem Ergebnis weiterarbeiten zu können
- SOAP
 - Weiterentwicklung von XML-RPC

XML RPC [1]

- XML RPC Bsp Request:

```
POST /xml-rpc.app HTTP/1.1
User-Agent: Java-Client
Accept-Encoding: gzip, deflate
Content-Type: text/xml
```

```
<?xml version="1.0" encoding="UTF-8">
<methodCall>
  <methodName>calcMaximum</methodName>
  <params>
    <param><value><int>2</int></value></param>
    <param><value><int>4</int></value></param>
  </params>
</methodCall>
```

XML RPC [2]

- XML RPC Bsp Response:

```
<?xml version="1.0" encoding="UTF-8">  
<methodResponse>  
  <params>  
    <param><value><int>4</int></value></param>  
  </params>  
</methodResponse>
```

XML RPC - Vorteile

- Sehr einfach zu realisieren
 - > führte zur schnellen Verbreitung
 - > und großen Popularität des Standards

XML RPC - Nachteile

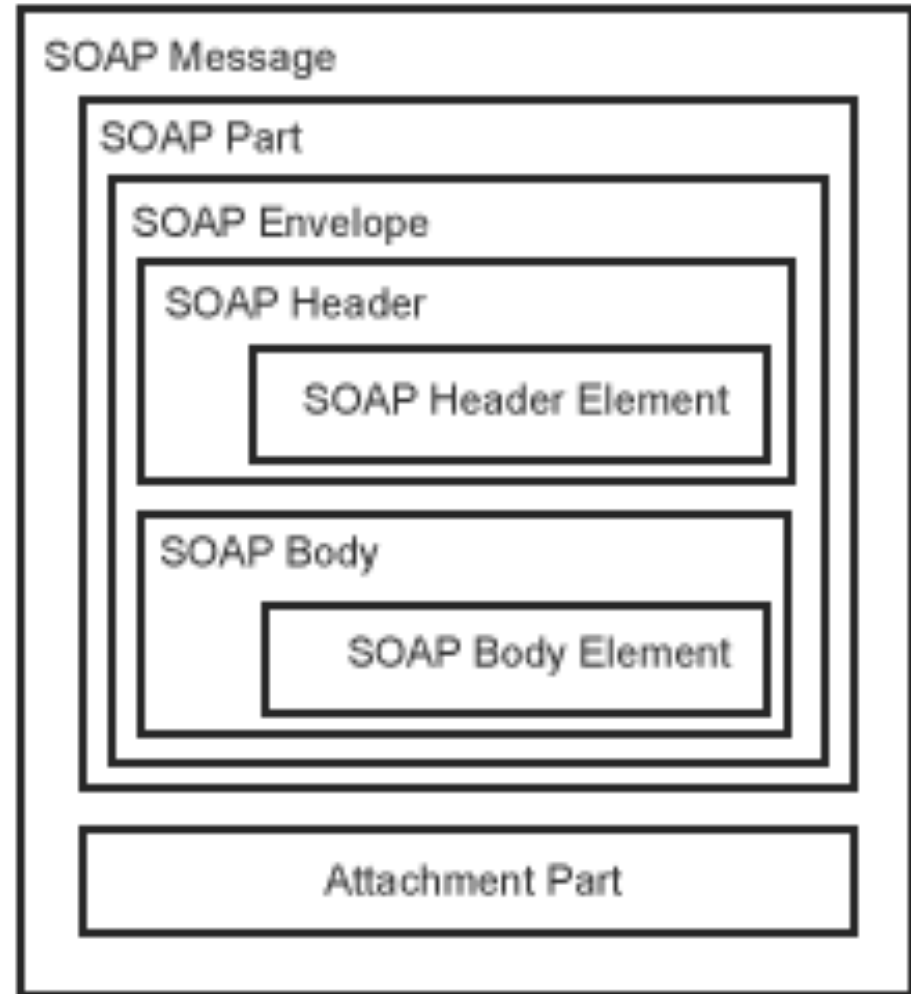
- Wichtige Sprach-Elemente fehlen
 - Ungenaue Kodierung der Datentypen
 - Zeichenketten nur in 7-bit ASCII
 - ISO-8601 für Datum und Uhrzeit ohne Zeitzonen
 - Aufwendiges Kodieren von binären Daten
 - Alle Daten werden mit base64 Codierung übertragen
 - Führt zu 30% Overhead
 - Aufwendiges en-/decoding
 - Fehlende Meta-Beschreibung
 - Programmierer müssen genau wissen wie die Methodensignatur aussieht
- Daher heute kaum mehr in „nackter Form“ im Einsatz

Simple Object Access Protocol (SOAP) [1]

- Kommunikationsprotokoll für XML Webservices um Daten zwischen Systemen auszutauschen und entfernte Methodenaufrufe durchzuführen
- Meist Kommunikation über HTTP (TCP)
 - Request & response Schema
- XML Datenformat

Simple Object Access Protocol (SOAP) [2]

- Header:
 - Metainformationen der Nachricht (routing, Verschlüsselung, etc)
- Body
 - Nutzdaten, die vom Empfänger interpretiert werden
- Anhänge
 - Binäre Dateien
 - ≥ 1 MIME Header pro Part



Simple Object Access Protocol (SOAP) [3]

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <contact status="new">
      <name>Thomas Stark</name>
      <address>
        <e-mail>thomas@masterclass.de</e-mail>
      </address>
    </contact>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


WSDL

- Web Services Description Language
- Eine Sprache, womit die Schnittstellen sowie Ein- und Ausgabeparametern beschrieben werden
- Verwendung in Kombination mit SOAP und XML-Schema, um Web Services im Internet anzubieten

UDDI

- Universal Description, Discovery and Integration
- Verzeichnisdienst
- UDDI besitzt eine SOAP-Schnittstelle
- Unterscheidung zwischen drei Arten von Informationen:
 - White Pages (Basisinformationen)
 - Yellow Pages (Servicekategorisierung)
 - Green Pages (Schnittstellenbeschreibungen)

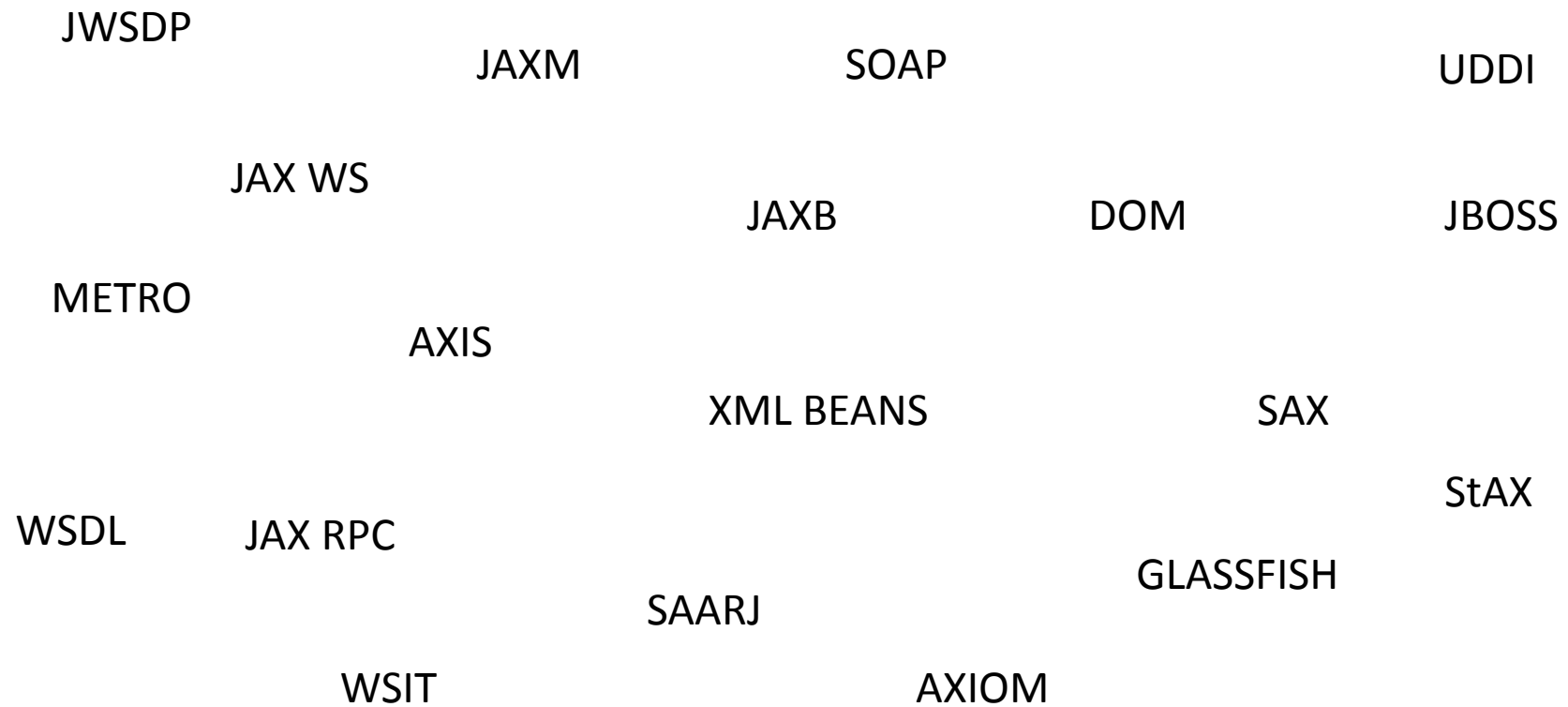
JEE 5

Java Webservices

Webservices in JAVA



Webservices in JAVA



Webservices in JAVA

- Verschiedene Möglichkeiten
 - JAXM (JAVA API for XML Messaging)
 - JWSDP (JAVA Web Service Development Pack)
 - JAX-WS

 - Apache AXIS
 - XINS (Interface for Network Services)
 - JBossWS

JAXM – Java Api for XML Messaging [1]

- JSR 67
- Zuletzt in dieser Form in java SE 1.5
- Letztes Release 2006
- RI von Sun mit Codenamen »The M Project« (mit Stand 2003)
- Standard für Nachrichten und deren Austausch zwischen verschiedenen Systemen auf Basis von XML
- das genaue Format der Nachricht und die möglichen Protokolle werden offen gelassen
- Formate: z.B.
 - SOAP
 - ebXML

JAXM – Java Api for XML Messaging [2]

- Folgende Ziele werden mit JAXM verfolgt:
 - Unterstützung für Standard-Nachrichtenformate und Adressen
 - Sichere Auslieferung von Nachrichten,
 - Verfolgung des Weges einer Nachricht
 - Unterstützung für Nachrichtentemplates
 - Definition der zu nutzenden Datentypen und Formate
 - Definition von Zugriffskontrolle, Sicherheitsmechanismen sowie Authentifikation und Autorisation

JAXM – Java Api for XML Messaging [3]

- Bauen einer Soap-Nachricht mit `javax.xml.soap.*`

-> Live Demo

„manuelles“ Erstellen einer SOAP Message in JAVA

JWSDP – Java Web Service Development Pack

- Freies SDK
- Alle Komponenten von JWSDP sind nun Teil von GlassFish und WSIT (Webservice Interoperability Technology)
- Einige Komponenten sind in Java SE 6
 - JAXP (Java Api for XML Processing)
 - JAX-RPC (-> später in JAX-WS integriert)
 - JAX-WS (Java Api for XML WebServices)
 - JAXB (Java Architecture for XML-Binding)
 - SAAJ (SOAP Attachments API for Java)
 - JAXR (Java Api für XML Registries)

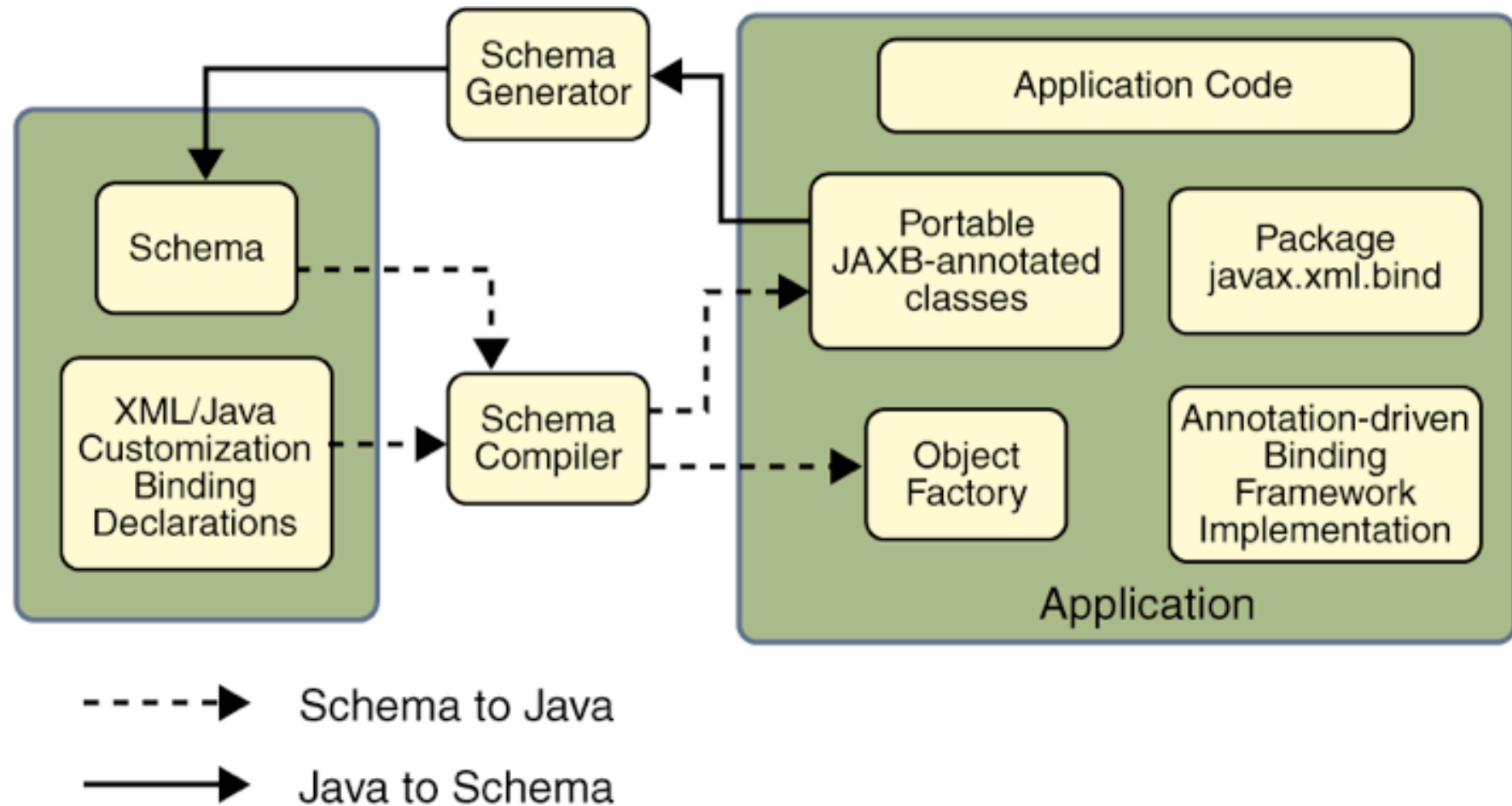
JAX-B Grundlagen

- Inhaltlich:
 - Java Architecture for XML-Binding
 - „The Java Architecture for XML Binding (JAXB) provides a fast and convenient way to bind between XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications.“ (Sun, Java Web Services Tutorial)
- Organisatorisch
 - wurde im Rahmen des Java Community Processes (JCP) entworfen
 - JAXB 1.0 als JSR (Java Specification Request) 31
 - JAXB 2.0 als JSR 222
 - Referenzimplementierung von SUN seit JSE 1.6 im JDK enthalten
 - Ursprung: Java Web Services (Referenzimplementierung in Glassfish / SUN Application Server)

JAX-B Features

- Generierung von Java-Datentypen aus einem XML Schema
- unterstützt vollständige XML Schema Spezifikation
- objektbasierte Sicht auf XML-Daten
- bequemer & performanter Zugriff auf XML Daten aus Java heraus
 - Diverse XML-parsing APIs verwendbar: DOM, StAX, SAX

JAX-B Aufbau



JAX-B Datentypen-Mapping (Auszug)

XML Schema Type	Java Data Type
xsd:string	java.lang.String
xsd:integer	java.math.BigInteger
xsd:int	int
xsd.long	long
xsd:short	short
xsd:decimal	java.math.BigDecimal
xsd:float	float
xsd:double	double
xsd:boolean	boolean
xsd:byte	byte
xsd:QName	javax.xml.namespace.QName

Java Api for XML Webservices (JAX-WS)

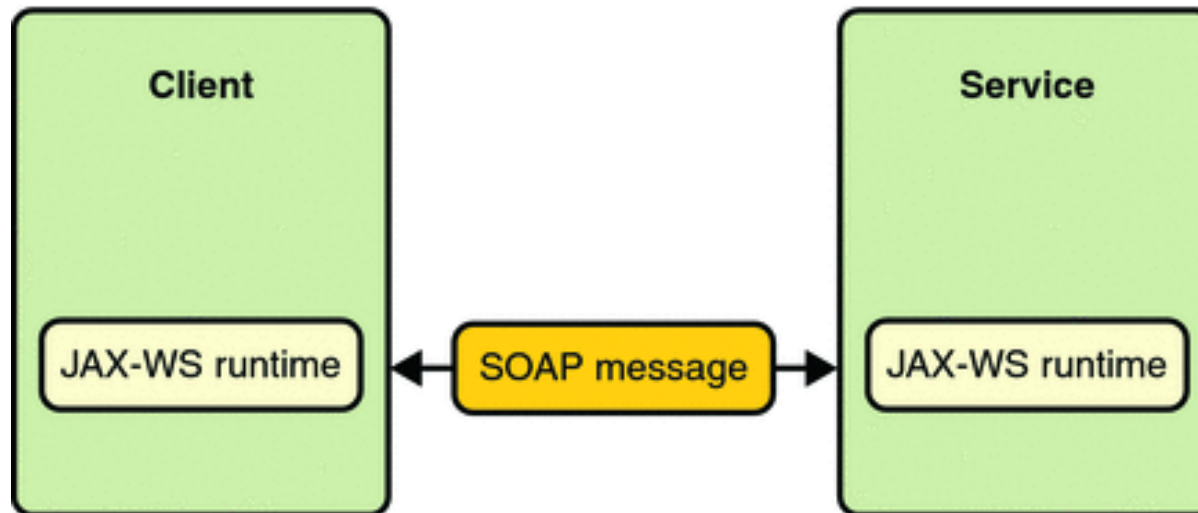
- JSR 224
- ehem. JAX-RPC
- Unterstützt
 - SOAP 1.1 + 1.2
 - WSDL 1.1
 - MTOM (WS mit Anhängen)
 - Seit JAX-WS 2.1:
 - API für asynchrone WebServices
 - WS-Adressing
- Gesteuert über Annotations

Java Api for XML Webservices (JAX-WS)

- `wsgen` & `wsimport`
- Referenz-Implementierung JAX-WS RI ist Teil von
 - Glassfish und des Metro Stacks
 - Sun Java Application Server 9.x
 - Axis 2
 - JBoss WS
- JAX-WS ist nachrüstbar in Application Servern
 - Voraussetzung: JAX-B

Java Api for XML Webservices (JAX-WS)

- Kommunikation zwischen einem JAX-WS Web Service und dem Client



Java Api for XML Webservices (JAX-WS)

- Code-Beispiel

```
package ws;

import javax.jws.*;

@WebService(name="BeispielService")
public class myWebservice{

    @WebMethod
    public String getWelcomeMessage(String name){
        return "Welcome " + name;
    }
}
```

Java Api for XML Webservices (JAX-WS)

- Annotations
 - @PostConstruct
 - Aufruf vom Container bevor der WS antwortet
 - @PreDestroy
 - Aufruf bevor der Endpoint wieder entfernt wird

 - @WebService
 - Definiert eine Klasse als Webservice
 - @WebMethod
 - Definiert eine Methode als Webservice
 - @OneWay
 - Asynchrone Webservice-Methode

Java Api for XML Webservices (JAX-WS)

- Zu beachten
 - Webservice Methoden dürfen nicht
 - Private
 - Static
 - Final
 - Abstract
 - Webservice Klassen müssen einen leeren Standard-Konstruktor haben

JAX-WS Erweiterungen

- WSIT (Web Service Interoperability Technology)
 - Gemeinsames Projekt mit Microsoft zur Interoperabilität zu Windows Communications Framework (WCF)
- Projekt „Metro“
 - Stammt aus Projekt Glassfish
 - Beinhaltet
 - Transport
 - Reliability
 - Transaction
 - Security

Erstellen eines Webservices

- wsgen
 - The wsgen tool generates JAX-WS portable artifacts used in JAX-WS web services. The tool reads a web service endpoint implementation class (SEI) and generates all the required artifacts for web service deployment, and invocation.
- wsgen - Demo

Abonnieren eines Webservices

- `wsimport`
 - The `wsimport` tool generates JAX-WS portable artifacts, such as:
 - Service Endpoint Interface (SEI)
 - JAXB generated value types (mapped java classes from schema types)
 - ... The generated Service class can be used to invoke the Web Service endpoint.

- `wsimport` - Demo

Abonnieren eines Webservices

- artefakte aus wsimport ins Projekt übernehmen

```
public class PersonenClient {  
  
    @WebServiceRef(wsdlLocation="http://pfad_zur_wsdl")  
    static PersonenServiceService service;  
  
    public static void main(String[] args) {  
        service = new PersonenServiceService();  
        PersonenService port = service.getPersonenServicePort();  
        ...  
    }  
    ...  
}
```


Apache Axis

- Axis 1
 - Entstanden aus einem IBM Projekt (SOAP4J)
 - Übergeben an Apache Foundation (-> Apache SOAP)
 - Schlechte Performance (DOM Parser)
 - -> Version 3
 - Redesign mit SOAP 1.1, WSDL 1.1, SAAJ 1.1 und JAX-RPC 1.0 sowie dem SAX Parser
 - Apache AXIS (Apache eXtensible Interaction System)
 - Sehr populär um 2003/2004
 - Weiterentwicklungen (z.B. StAX)
 - Allerdings u.A. Deployment bemängelt

Apache Axis

- Axis 2
 - Redesign von Axis 1
 - Axis 1 wird immernoch gepflegt, jedoch gibt es keine Weiterentwicklungen
 - Was ist neu?
 - Verwendung von StAX + AXIOM (DOM ähnlicher Zugriff)
 - Splitten des Request und Response Schemas auf 2 separate Engines -> Asynchrone WS, Contract first, Contract last
 - Deployment-Deskriptoren werden durch Annotations „ersetzt“
 - Data-Binding mittels XML-Beans und JiBX
 - Unterstützung von JAX-WS 1.4.1 (2.0 experimentell)

Apache Axis

- Integration von JAX-WS in Axis
 - Größte Erweiterung seit dem Entstehen
 - Erst ab Version 1.3 auf der Anforderungsliste als eine stabile Version von JAX-WS vorlag
 - Aktuell 1.4.1
 - 2.0 experimentell
 - Voraussetzung für JAX-WS: JAX-B
 - Axis 2 liefert **keine** Werkzeuge wie java2wsdl
 - wsgen
 - Wsimport aus Java 1.6 (JAX-WS RI)
 - Ad-Hoc WSDL generierung

Fazit: Webservices

- Vorteile
 - Hohe Kompatibilität
 - Loose Kopplung zwischen Client und Server
 - Plattformunabhängig
 - Synchroner und Asynchroner Kommunikation
- Nachteile
 - Performance und Overhead
 - Sicherheit / Transaktionsunterstützung / Ausfallsicherheit

Fazit: JAVA

- Durch JAX-WS wurde eine allgemeine Schnittstelle für den Entwickler geschaffen
 - schnelle Entwicklungszeit
 - Plattformunabhängig (Server)

Über den Tellerrand: Coldfusion

- 1994 von Allaire Corp. Entwickelt
- Weitere Produkte Jrun, HomeSite
- WDDX: Web Distributed Data eXchange (1998)
 - In frühen Version von Coldfusion integriert
 - Kann als Vorgänger von WebServices bezeichnet werden
 - XML+DTD
- Coldfusion als erster Application Server überhaupt
- 2001 von Macromedia übernommen
 - Redesign auf J2EE seit Coldfusion Version 6.0
 - 100% Java Compatible (Sun certified)

Über den Tellerrand: Coldfusion

- Integration in die Windows-Welt
 - Exchange Support
 - ASP.net Support
 - DLL integration
- Code ist auf 2 Arten zu Schreiben
 - TAG-Basierend >.

```
<cfif #var# GTE 3>...<cfelse..</cfif>
```
 - Skript-Basierend

```
<cfscript>  
    if(#var# >= 3){  
        ...  
    }else ...  
</cfscript>
```

Über den Tellerrand: Coldfusion

- Coldfusion Webservice:

```
<cfcomponent>
  <cffunction name="helloMessage" access="remote"
  returntype="string" >
    <cfargument name="name" type="string" required="yes">
    <cfset ret= "Hallo #name#">
    <cfreturn ret>
  </cffunction>
</cfcomponent>
```

Aufruf über <http://...:8080/componente.cfc?wsdl>

Quellen

- Vorlesungsskript: Andreas Judt (FZI) 2006
- Introducing JAX-WS 2.0 With the Java SE 6 Platform
http://java.sun.com/developer/technicalArticles/J2SE/jax_ws_2/
- JAX-WS Reference Implementation Project
<https://jax-ws.dev.java.net/>
- Java EE 5 Tutorial
<http://java.sun.com/javaee/5/docs/tutorial/doc/bnayk.html>
- Metro Specification
https://metro.dev.java.net/guide/Metro_Specifications.html
- Apache Axis2: Bleibt alles anders?
<http://it-republik.de/jaxenter/artikel/Bleibt-alles-anders-1967.html>
- JAX-WS Artikel
<http://it-republik.de/jaxenter/artikel/Happy-JAX-WS-ing%21-1983.html>
<http://entwickler.de/zonen/portale/psecom,id,101,online,1986.html>
- FH-Gießen-Friedberg – SOA, SS07 (Markus Keim)
<http://homepages.fh-giessen.de/~hg52/lv/soa/vortraege/jax-ws.pdf>
- Java EE 5 - Einstieg für Anspruchsvolle (Thomas Stark)
ISBN-10: 3-8273-2362-2

Fragen?

Vielen Dank für die Aufmerksamkeit