

Verteilte Systeme

Vorlesung 11
Sebastian Iwanowski
FH Wedel

Verteilte Systeme

1. Innovative Beispiele aus der Praxis
2. Allgemeine Anforderungen und Techniken verteilter Systeme
3. Die Client-Server-Beziehung und daraus entstehende Fragestellungen
4. Nebenläufigkeitstechniken in Java
5. Entfernte Aufrufe
6. Objektmigration
7. Agententechnologie
8. Dienstevermittlung
9. Synchronisation von Daten
10. Konzepte zur Erzielung von Fehlertoleranz
- 11. Web Services**

**Die meiste des folgenden Materials stammt von
Ulrike Hammerschall
(Autorin des Buchs *Verteilte Systeme und Anwendungen*)**

Historische Einordnung

Webanwendungen der 1. Generation: HTML-Seiten



Am Webserver liegen statische HTML Seiten.

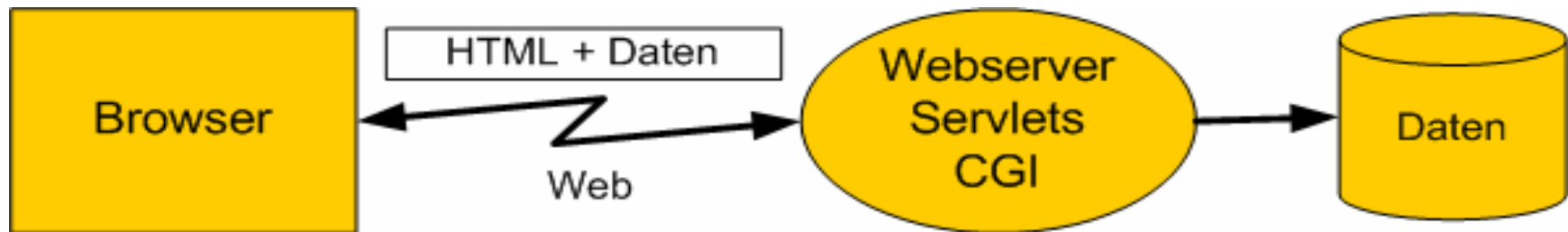
Diese werden über HTTP in den Browser geladen.

Technologie ab Ende der 1980er Jahre verfügbar.

Mensch-Maschine-Kommunikation

Historische Einordnung

Webanwendungen der 2. Generation: Dynamische HTML Seiten: CGI, Servlets



Webseiten werden dynamisch (abhängig von Anfrageergebnissen) aufgebaut.
Zugriff auf hinter der Seite liegende Daten ist möglich.

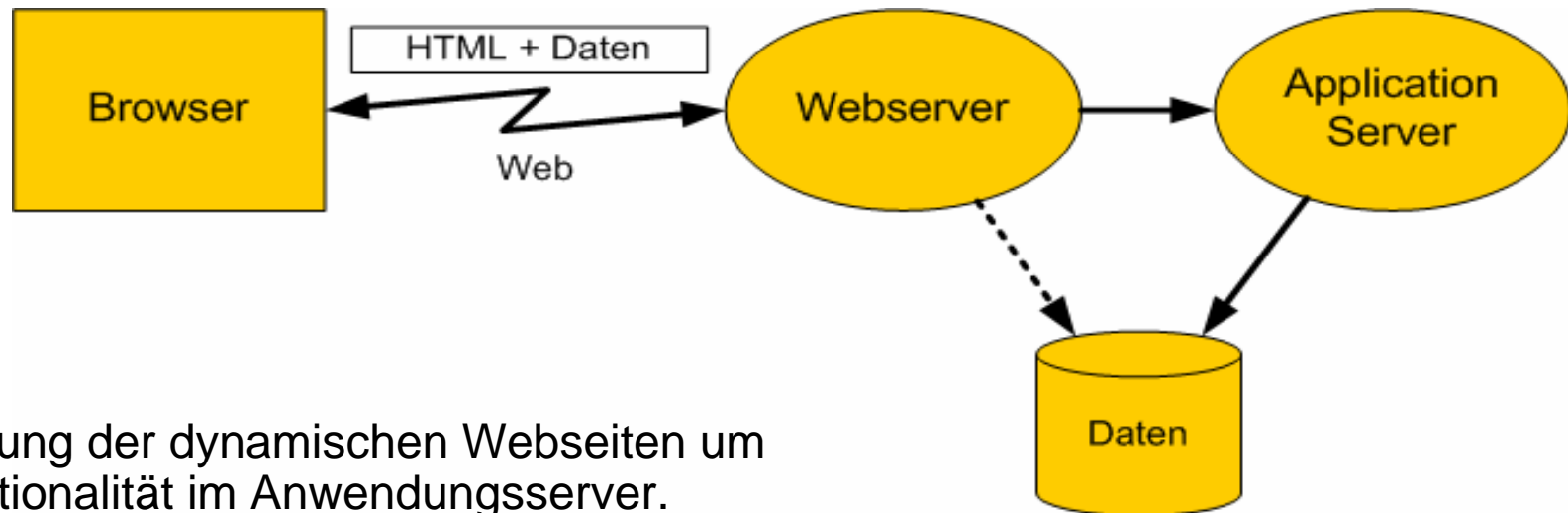
Beispiele für Technologien:

- CGI (Common Gateway Interface): Prozessbasierte Bearbeitung von HTTP-Requests über Skripten (Perl, C)
- Servlets / JSP: Threadbasierte Bearbeitung von HTTP-Requests (nur für Java)

Mensch-Maschine Kommunikation

Historische Einordnung

Webanwendungen der 3. Generation: Einführung von Mehrschichten-Architekturen



Erweiterung der dynamischen Webseiten um Funktionalität im Anwendungsserver.

Der Webserver leitet die Aufrufe an den Anwendungsserver weiter:

- bearbeitet die Anfrage.
- schickt Ergebnisse an den Webserver zurück.
- Ergebnisse werden von Webserver über HTTP an den Browser zurückgeschickt.

Vorteile: Skalierbarkeit!

Mensch-Maschine Kommunikation

Historische Einordnung

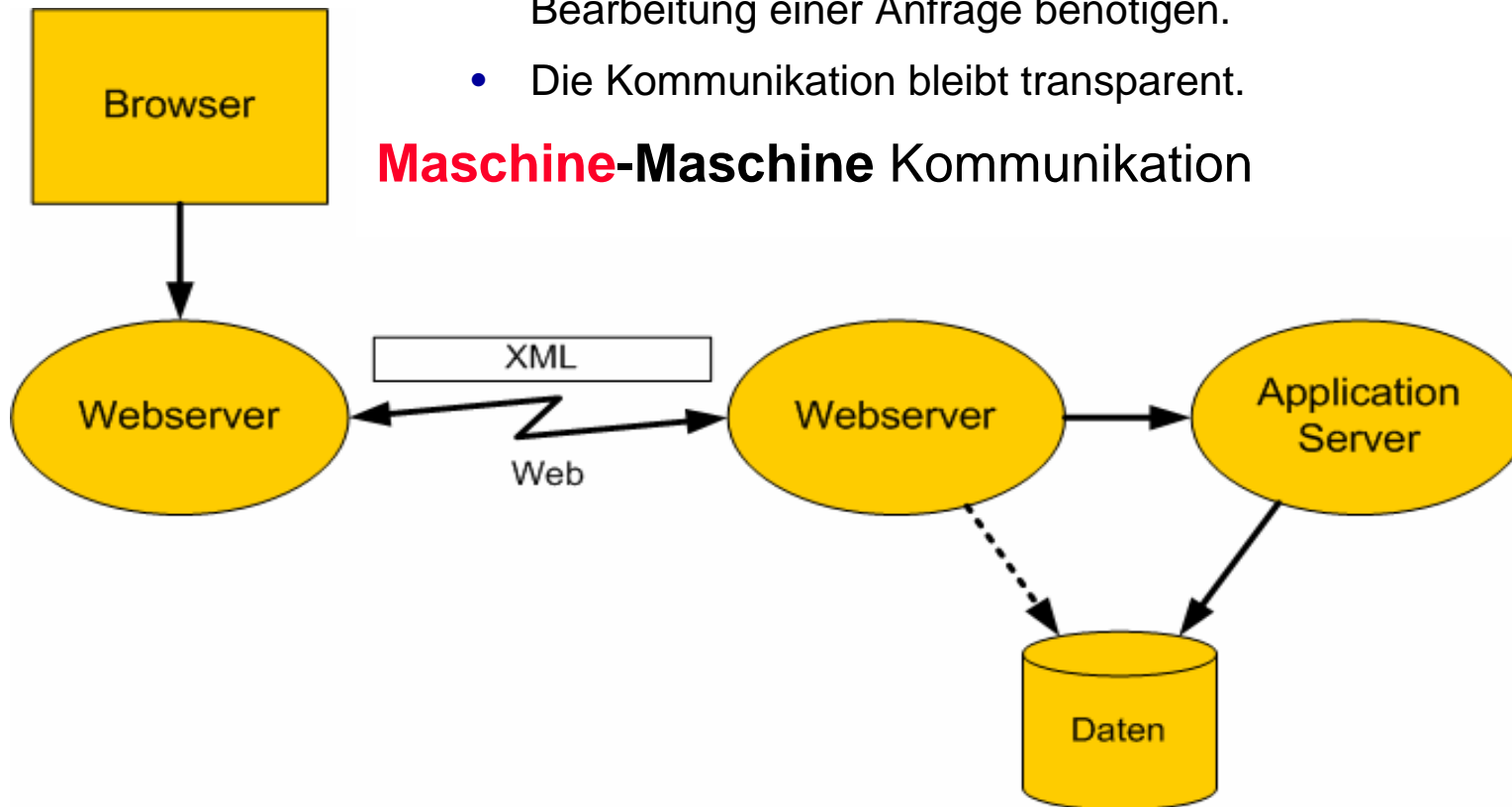
Webanwendungen der 4. Generation: Webservices

Anwendungen (auf Webservern) kommunizieren unabhängig vom Anwender.

Vision Automatisierung:

- Anwendungen suchen sich automatisch Dienste im Web, die sie zur Bearbeitung einer Anfrage benötigen.
- Die Kommunikation bleibt transparent.

Maschine-Maschine Kommunikation



Was sind Webservices?

A web service is any service that is available over the internet, uses a standardized XML messaging system, and is not tied to any one operating system or programming language.

Ethan Cerami, Autor des Buchs *Web Services Essentials*, 2002

A web service is self-contained, modular, uses open interfaces and is based on internet standards.

UDDI-Konsortium (OASIS)

A web service is written in XML, is affiliated to a URI, and supports direct interaktion between software agents.

W3C

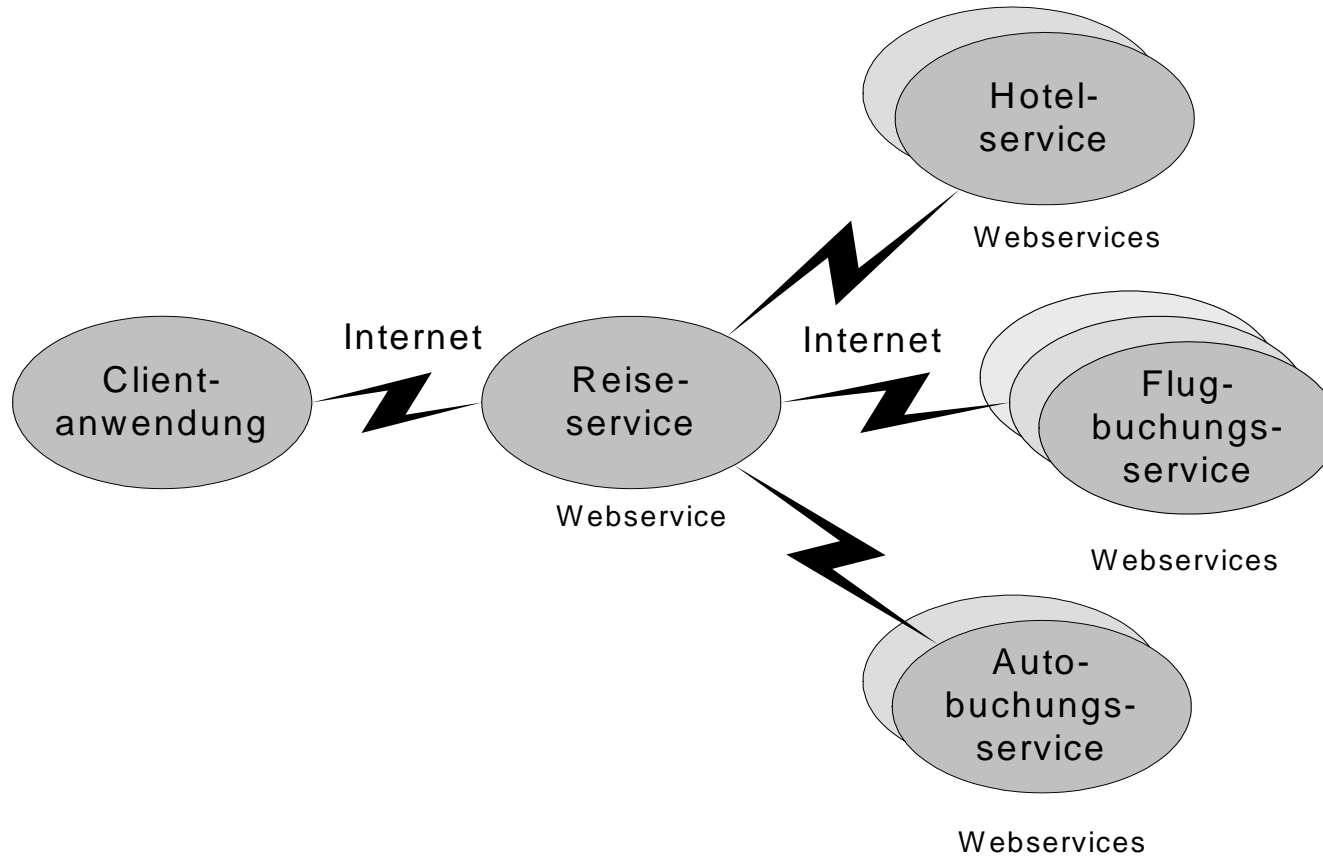
Webservices

- ermöglichen webbasierte Remote Procedure Calls.
- sind einer der Hypes der letzten Jahre.
- sind eine sehr junge Technologie (kamen ca. 2000 auf im Zusammenhang mit Microsoft .Net).
- sind nichtsdestotrotz eine vielversprechende Technologie der Zukunft.
- basieren auf Standards, die (hauptsächlich) vom W3C (World Wide Web Consortium) vorangetrieben werden.

Ulrike Hammerschall, 2004

Webservices

„Ein visionäres Beispiel“ (nach Hammerschall)



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)

SOAP

- ist der Standard für das Middleware-Protokoll der Webservices.
- realisiert z.B. den entfernten Prozeduraufruf
- kann auf verschiedenen Trägerprotokollen aufsetzen:
 - HTTP (häufigstes Trägerprotokoll)
 - SMTP
 - Messaging Service
 - CD, Diskette
 - ...
- wird in XML beschrieben.
- wird wie XML vom W3C verwaltet.

SOAP

Komponenten einer SOAP-Beschreibung:

Envelope

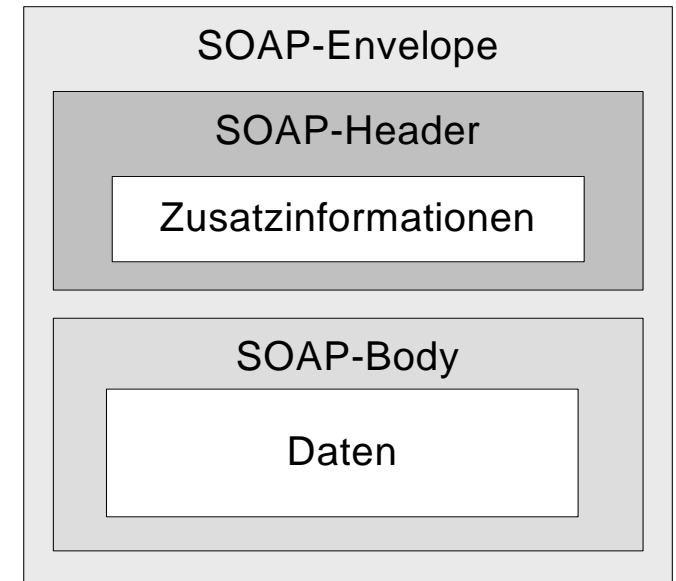
- dient als Container für die gesamte SOAP Nachricht
- muss Konventionen genügen, damit die Nachricht als SOAP Nachricht identifiziert werden kann.

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.



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)

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.

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>`

....

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.

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>

.....

WSDL

Aufbau einer WSDL-Schnittstelle

.....

<binding>

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

</binding>

<service>

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

</service>

</definitions>

UDDI

Umfang von UDDI

Verzeichnisdienst für Webservices.

Veröffentlicht werden:

- WSDL Definitionen von Webservices
- Zusatzinformationen zu Inhalten, verantwortliche Organisation, grobe thematische Einordnung,

Definiert zwei APIs:

- Publishing API: Anbieter können über dieses API ihren Webservice veröffentlichen.
- Inquiry API: Anwender können über dieses API nach einem Webservices suchen.

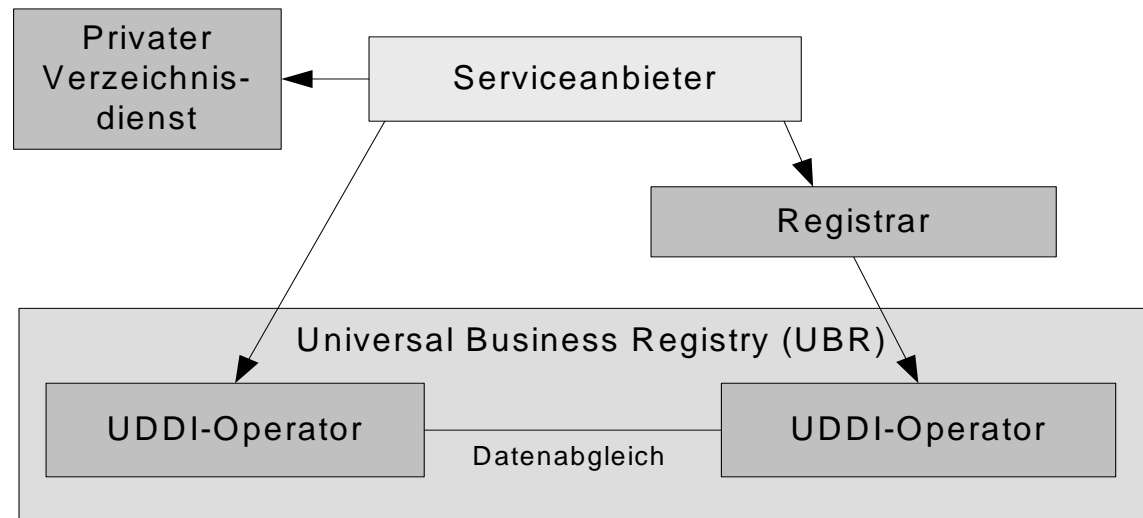
Historie von UDDI

- ursprünglich im Rahmen eines unabhängigen Projekts entwickelt: www.uddi.org (Beginn 2000).
- wurde 2002 an OASIS übergeben.

UDDI

UDDI-Organisationsstrukturen

- Der UDDI Standard gibt feste Organisationsstrukturen vor.
- Ein UDDI-Operator ist ein Anbieter einer UDDI Registry, welche die Anforderungen der *UDDI Operators Specification* bezüglich Verfügbarkeit, Sicherheit und Performance erfüllt. (Anm.: Nur wenige Unternehmen sind dazu in der Lage, z.B. IBM, Microsoft, SAP)
- Die UDDI-Operatoren bilden in ihrer Gesamtheit die UBR.
- Daneben sind private Verzeichnisdienste möglich.
- Registrare unterstützen Serviceanbieter bei der Veröffentlichung ihres Webservices.

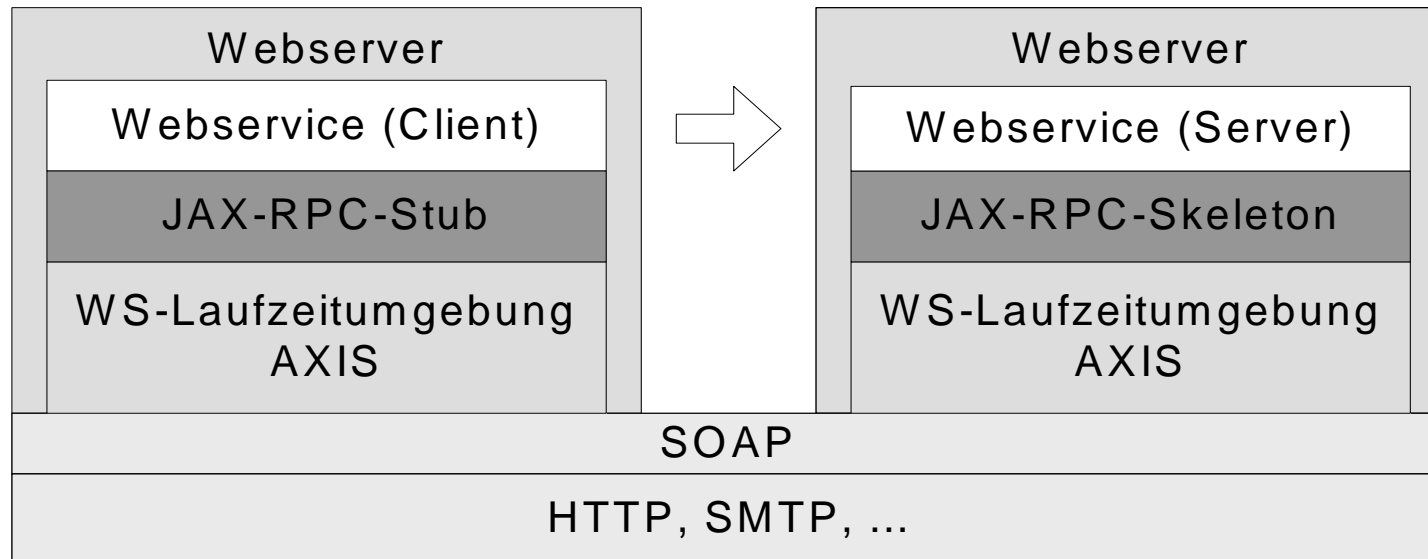


Ablauf einer Webservice-Anwendung

1. Der Anbieter eines Webservices publiziert den Service mit Hilfe einer WSDL-Beschreibung in einem UDDI Service.
2. Der Anwender (eine Anwendung) sucht anhand von Merkmalen einen geeigneten Service.
3. Der Anwender erhält von UDDI eine Referenz (URL) auf den Service
4. Der Anwender holt sich über die URL die WSLD Beschreibung vom Serviceanbieter.
5. Der Anwender generiert sich aus der WSDL Beschreibung den Stub und kann den Dienst nutzen (dynamic binding)
... oder der Anwender erhält den Stub bereits zur Compilezeit und kann jeder Zeit ohne Umweg über Verzeichnisdienst den Webservice nutzen. (static binding)

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 WSLD-Schnittstelle generiert.



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/>

OASIS: WS-Security 2004, BTP, SAML, UDDIv2 Specifications: <http://www.oasis-open.org/specs/index.php>

OASIS: UDDIv3 Specification: http://uddi.org/pubs/uddi_v3.htm

OASIS: BPEL: <http://www.oasis-open.org/committees/wsbpel>

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

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

Beim nächsten Mal:

***Aktuelle Plattformen für die Entwicklung
verteilter Systeme
Zusammenfassung der Vorlesung***