

Hauptseminar Internet-Dienste

Sommersemester 2004 Boto Bako

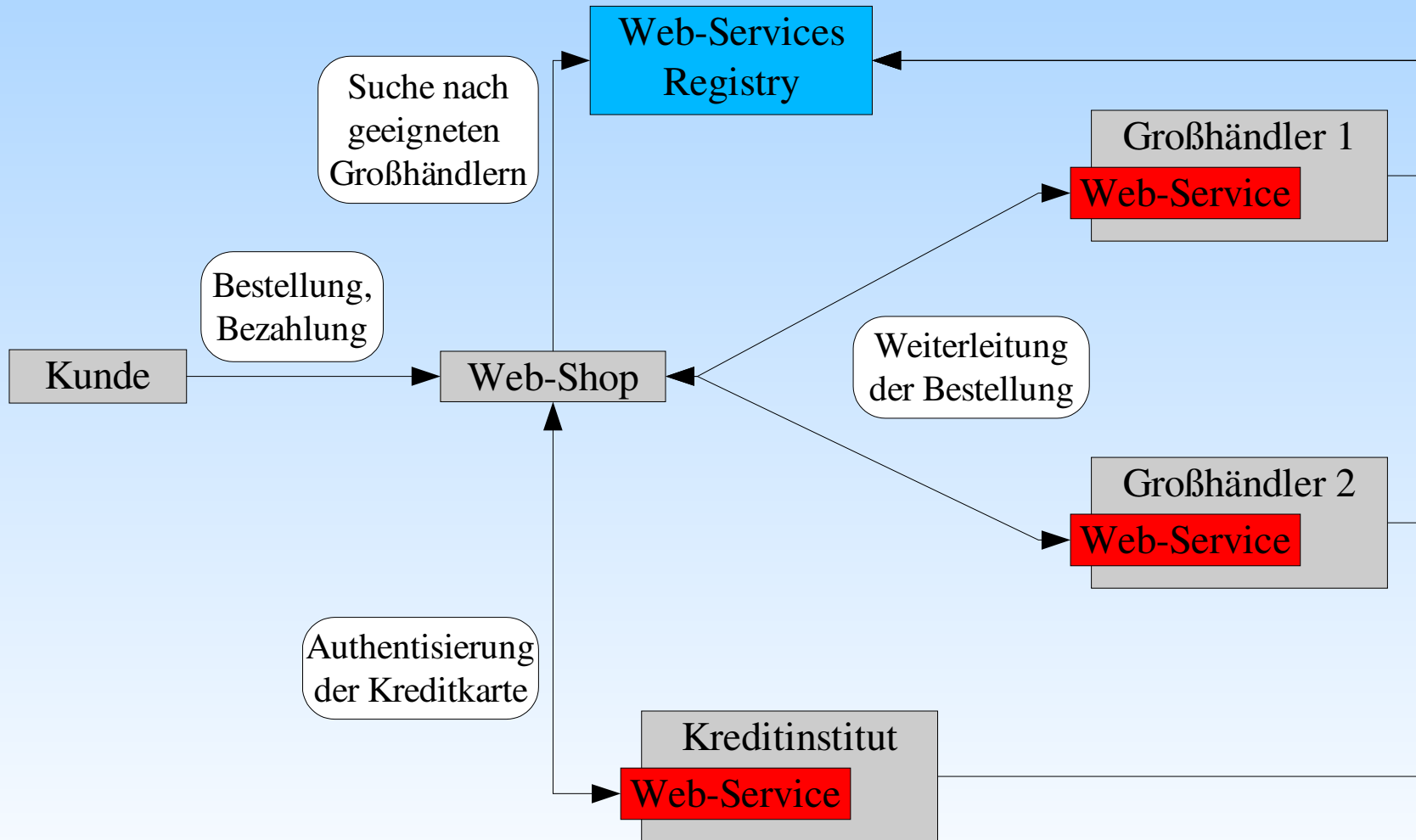
Webservices

- 1 Einführung
- 2 Verwendete Standards
- 3 Web-Services mit Java
- 4 Zusammenfassung

Was sind Web-Services?

- Web-Services sind angebotene Dienste im Internet
- Angesprochen durch XML-basierte Standardschnittstellen
- Lose Kopplung heterogener Systeme
- Ziel: automatisierte Kommunikation von Applikationen über das Internet

Beispielszenario: Onlineshop



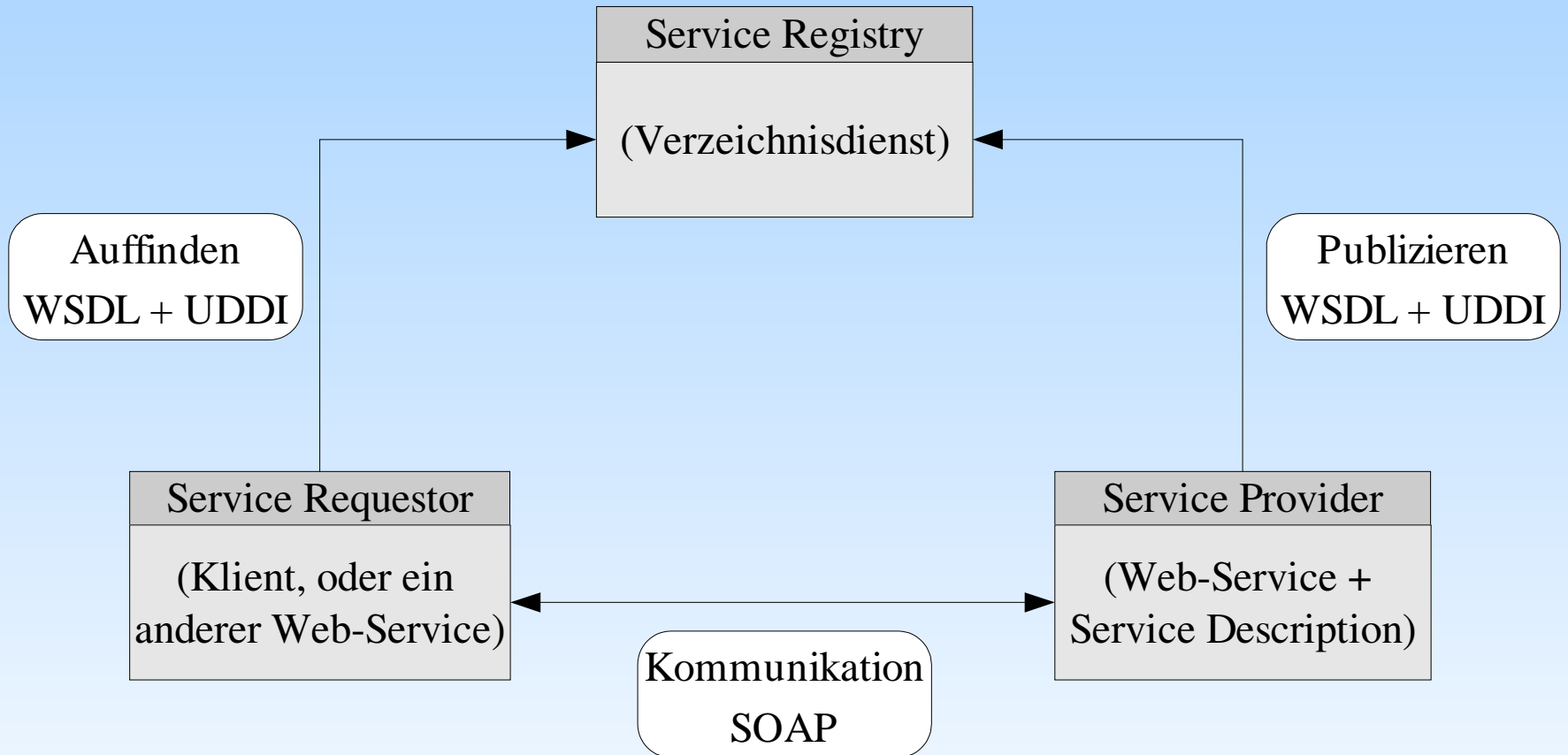
Warum Web-Services?

- Unabhängig von Plattform und Sprache:
 - Implementierung mit CORBA, EJB, ...
- Vision vom Markt der Komponenten:
 - Softwarefirmen verkaufen Web-Services-Software
 - Dienstanbieter bieten Web-Services an
- Nachteile bisheriger Komponentenmodelle:
 - Unterschiedliche Protokolle
 - Unterschiedliche Schnittstellenbeschreibungen
- Web-Services sind kein Ersatz, sondern eine Vereinheitlichung bisheriger Technologien

Standards

- Die Vereinheitlichung wird durch Einhaltung folgender Standards erreicht:
 - Kommunikation über SOAP
(*Simple Object Access Protocol*)
 - Beschreibung von Diensten über WSDL
(*Web Services Description Language*)
 - Auffinden und Registrierung über UDDI
(*Universal Service Description, Discovery, and Integration*)

Rollen in Web-Services



SOAP

- Protokoll zum Austausch von Nachrichten
- Nachrichten sind XML-Dokumente
- Mit Nachrichtenverbund komplexere Kommunikationsparadigmen realisierbar:
 - Request-Reply-Protokolle
 - Multicast Protokolle
- Beliebiges Trägerprotokoll, meist HTTP

SOAP: Aufbau einer Nachricht (1)

■ SOAP Envelope

- „Umschlag“ der Nachricht, eine XML-Datei
- XML-Namensraum für SOAP-Tags
- Enthält Header und Body

■ Header (optional)

- Enthält anwendungsspezifische Daten, bspw. für Authentisierung, Transaktionen und Routing
- Zwischenknoten können Header verändern

■ Body

- Enthält die eigentliche Nachricht

SOAP: Aufbau einer Nachricht (2)

```
<SOAP:Envelope xmlns:SOAP= http://www.w3.org/2003/05/soap-envelope >
```

```
<SOAP:Header>
```

```
<signature> ... </signature>
```

```
</SOAP:Header>
```

```
<SOAP:Body>
```

```
<a:withdraw xmlns:a="http://eineBank.com/">
```

```
<a:account>1234</a:account>
```

```
<a:amount>200.25</a:amount>
```

```
</a:withdraw>
```

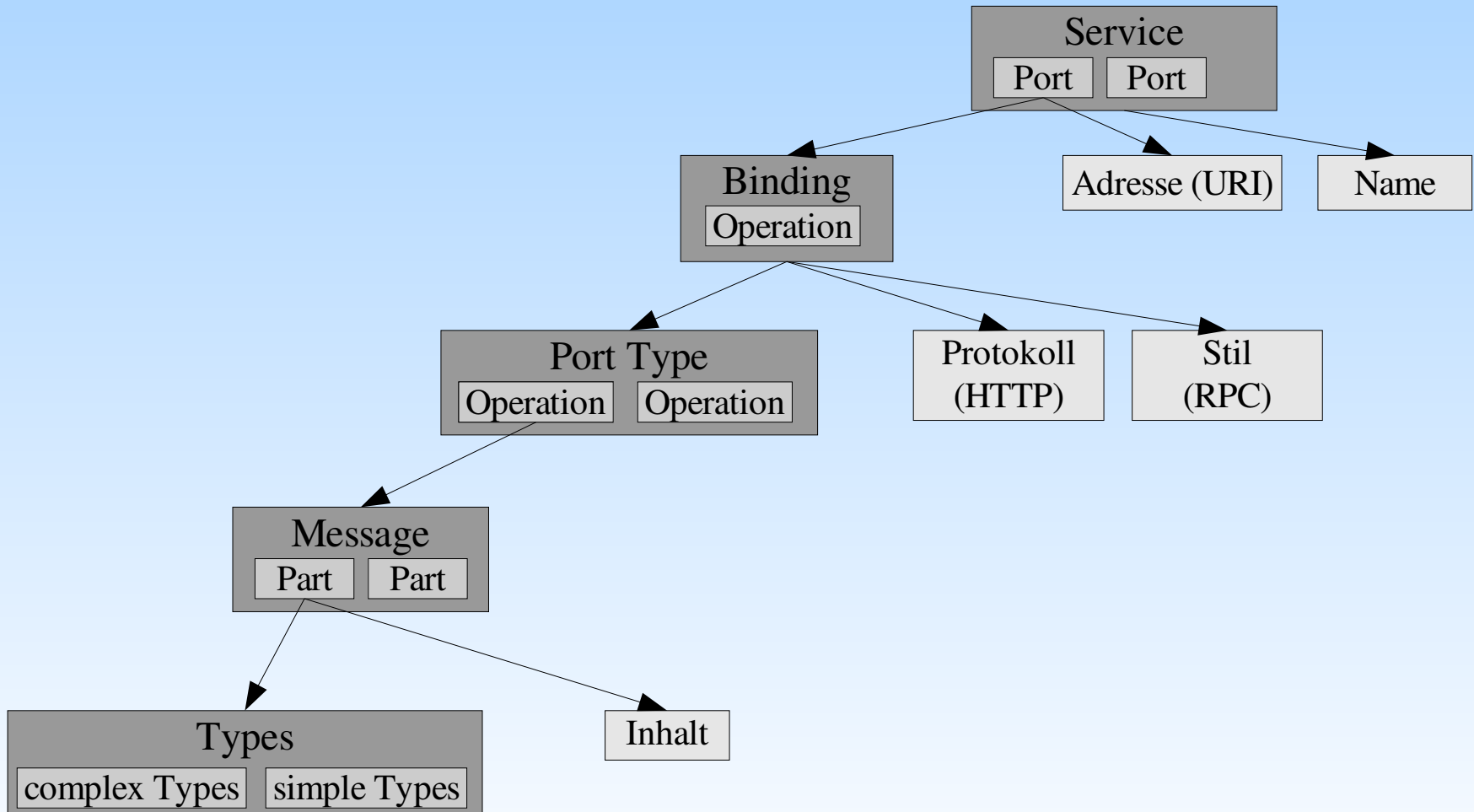
```
</SOAP:Body>
```

```
</SOAP:Envelope>
```

WSDL

- Schnittstellenbeschreibung für Web-Services
- XML-Dokument mit folgenden Elementen:
 - *Types*: Definition der Datentypen
 - *Messages*: Definition von Nachrichtentypen
 - *Port Types*: Definition der Operationen mit den dazugehörigen Nachrichten
 - *Binding*: Bindung an das Übertragungsprotokoll
 - *Service*: Beschreibung einer konkreten Dienstinstanz mit Namen, verwendeten Ports, Bindung und Adresse

WSDL: hierarchischer Aufbau



UDDI

- Die UDDI Spezifikation definiert einen Registry-Dienst für Web-Services
- UDDI Registry-Dienst ist ein Web-Service zum Beschreiben, Suchen und Publizieren von Web-Services und Geschäftseinheiten
- Dienstleistung einer UDDI-Registry:
 - *White Pages*: Namensregister
 - *Yellow Pages*: Kategorienregister
 - *Green Pages*: liefern technische Details zum Dienstzugang

Java Web Services Developer Pack

- Toolkit zum Entwickeln von Web-Services
- Enthält folgende Pakete:
 - SOAP with Attachments API for Java (SAAJ)
Erstellung von SOAP Nachrichten mit Anhängen
 - Java API for XML-based RPC (JAX-RPC)
Java<->WSDL Mapping und Integration mit EJB, Servlets und JSP
 - XML and Web Services Security
 - Und vieles mehr

Apache Axis (1)

- SOAP-Engine von Apache:
 - Server für die Ausführung von Web-Services
 - Java2WSDL für die Generierung des WSDL-Dokuments anhand eines Java-Interface
 - WSDL2Java für die Generierung des Klienten-Stubs und des Service-Wrappers
 - Admin-Client-Tool für das Einspielen von Web-Services
- Entwicklung entfernter Objekte in Anlehnung an RMI:
 - Statt rmic WSDL2Java

Apache Axis (2)

- Apache Axis implementiert die JAX-RPC API
- Kann in Application-Server (z.B. Tomcat) integriert werden
- Overhead von SOAP und WSDL wird vom Java-Programmierer versteckt
- Einfache Umstellung vorhandener Dienste auf Web-Services

Zusammenfassung (1)

- Standardisierter Zugriff auf Applikationen
 - Eine Art „maschinenlesbares“ World Wide Web
- Kommunikation zwischen Applikationen unabhängig von Sprachen, Plattformen und Protokollen
- Eine Vereinheitlichung vorhandener Technologien
- Neue Web-Services können aus vorhandenen zusammengebaut werden

Zusammenfassung (2)

■ Vorteile von Web-Services:

- Akzeptierte Technologie zur Interaktion in heterogenen Systemen
- Breite Unterstützung durch die Industrie
- Geringe Integrationskosten vorhandener Systeme

■ Nachteile von Web-Services:

- Kommunikationsoverhead
- Nicht ausgereifte Standards
- Komplexität und Anzahl der beteiligten XML-Dokumente
- Eng gekoppelte Systeme schwer realisierbar