

WSDL

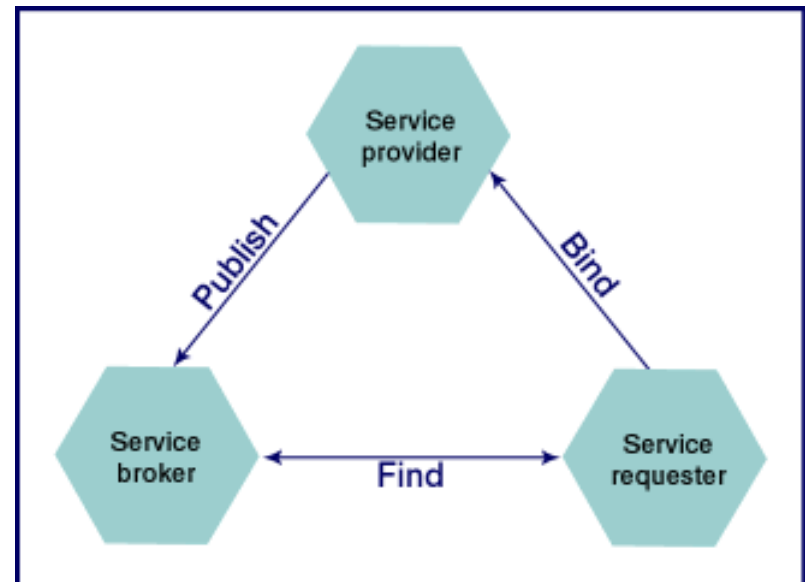
Web Services Description Language

Zum Ablauf

- Einleitung
 - Webservices und WSDL
 - Grundlagen (XML - Schema und Namespaces)
- WSDL
 - Syntax
 - Beispiel
- Zusammenfassung

Web Services Architektur (nach IBM)

- „Web Services are self-contained, modular applications that can be described, published, located and invoked over a network, generally, the World Wide Web.“
- 3 Rollen
- 3 Basisoperationen



Web Services Architektur

- Verringerte Komplexität durch Kapselung
- Zusammenarbeit der Webservices braucht erst zur Laufzeit festgelegt werden (jit integration)
- Web Services können in verschiedenen Sprachen geschrieben werden

Entstehung von WSDL

- Entwickelt von Ariba, IBM, Microsoft
- Dem W3C 2001 als Vorschlag zur Beschreibung von Web Services vorgelegt
- Verwendung gängiger Standards (XML, HTTP)
- Weiterentwicklung in Arbeitsgruppe

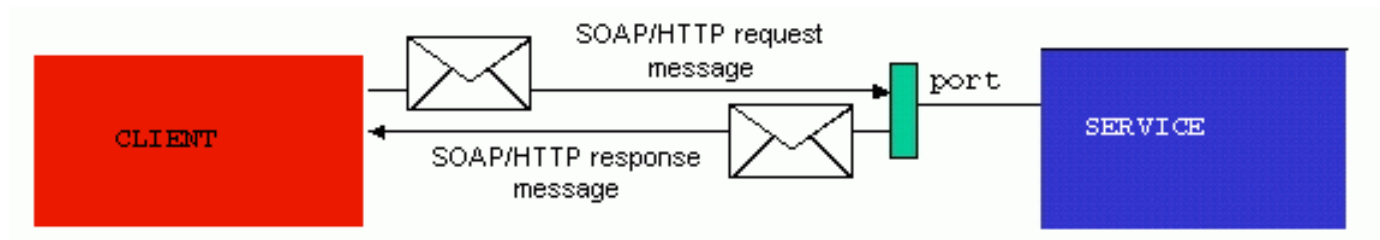
Was ist WSDL

- XML – basierte Beschreibungssprache
- namespace: <http://schemas.xmlsoap.org/wsdl>
- WSDL beschreibt Web Services als Sammlung von Kommunikationsendpunkten, die Nachrichten austauschen können.

Wie funktioniert WSDL

- Beschreibt abstrakt
 - Die verwendeten Datentypen (types)
 - Alle verwendeten Nachrichten (message)
 - Die aufrufbaren Methoden und Ihre Parameter (portType)
- Beschreibt konkret
 - Welches Protokoll verwendet wird (binding)
 - Den Kommunikationsendpunkt (service)
- Beschreibt nicht die Reihenfolge der Interaktionen

WSDL und SOAP



- Transportprotokoll http
- SOAP erweitert den HTTP-Header
- Nachrichten Inhalt: XML
- WSDL beschreibt Struktur der Nachricht

UDDI

- Zentrales Repository zum Anbieten und Auffinden von Web Services
- Betrieben von IBM, HP und Microsoft
- White , Yellow und Green Pages
 - Adresse und Kontaktinformationen
 - Zuordnung zu bestimmten Industriebereichen
 - Technische Beschreibung des Web Services

XSD - XML Schema

- xmlns:xsd=„http://www.w3.org/2001/XMLSchema“
- Definition eigener Datentypen
 - complexType:
durch Kombination bereits beschriebener oder built-in Datentypen
 - simpleType:
durch Restriktion der built-in Datentypen

Namespaces

- Definition: An XML namespace is a collection of names, identified by a URI reference [RFC2396], which are used in XML documents as element types and attribute names.

(<http://www.w3.org/TR/REC-xml-names/>)

Beispiel: Google

- Kostenloser Web Service im Beta-Stadium
 - Auffinden von Web Dokumenten
 - Ausliefern von Web Dokumenten
 - Rechtschreibprüfung
- Pro Lizenz Limitiert auf 1000 Anfragen am Tag
- <http://api.google.com/GoogleSearch.wsdl>

WSDL – Dokumentstruktur

WSDL – Dokument

definitions

types

schema

message

part

portType

operation

binding

operation

service

port

<definitions>

- Ist das Wurzelement des WSDL-Dokuments
- Attribute
 - Name
 - Namespace-Deklaration(en)
- Child-Elemente
 - Documentation
 - Import
 - Service
 - ...

<definitions> - GoogleSearch

```
<?xml version="1.0"?>
```

```
<definitions name="GoogleSearch"
```

```
  targetNamespace="urn:GoogleSearch"
```

```
  xmlns:typens="urn:GoogleSearch"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
```

```
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
```

```
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

...

```
</definitions>
```

<types>

- Definition der Datentypen, die in Nachrichten verwendet werden können
- Üblicherweise Benutzung des xsd – Typsystems
- Jedem Element wird ein Typ aus dem Typsystem zugeordnet

<types> - GoogleSearch

<types>

```
<xsd:schema xmlns=http://www.w3.org/2001/XMLSchema targetNamespace="urn:GoogleSearch">
```

```
...
```

```
<xsd:complexType name="DirectoryCategoryArray">
```

```
<xsd:complexContent>
```

```
<xsd:restriction base="soapenc:Array">
```

```
<xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:DirectoryCategory[]"/>
```

```
</xsd:restriction>
```

```
</xsd:complexContent>
```

```
</xsd:complexType>
```

```
<xsd:complexType name="DirectoryCategory">
```

```
<xsd:all>
```

```
<xsd:element name="fullViewableName" type="xsd:string"/>
```

```
<xsd:element name="specialEncoding" type="xsd:string"/>
```

```
</xsd:all>
```

```
</xsd:complexType>
```

```
</xsd:schema>
```

```
</types>
```

<message>

- Definiert Nachrichten die bei der Kommunikation verwendet werden können
- Message besteht aus mehreren Parts
- Part enthält Element oder Type

<message> - GoogleSearch

```
<message name="doGetCachedPage">  
  <part name="key"          type="xsd:string"/>  
  <part name="url"         type="xsd:string"/>  
</message>
```

...

```
<message name="doGoogleSearchResponse">  
  <part name="return"  
    type="typens:GoogleSearchResult"/>  
</message>
```

Operationen

- Stellen abstrakt eine Methode des Web Services dar
- Werden durch Messages beschrieben
 - Input
 - Output
 - Fault
- Müssen als einziges WSDL - Element keinen eindeutigen Namen haben (Überladung)

<portType>

- Gruppierung zusammengehörender Operationen
- Vier Kommunikationsformen:
 - Request-Response
 - One-Way
 - Solicit-Response
 - Notification

<portType> - GoogleSearch

```
<portType name="GoogleSearchPort">  
  <operation name="doGetCachedPage">  
    <input message="typens:doGetCachedPage"/>  
    <output message="typens:doGetCachedPageResponse"/>  
  </operation>  
  <operation name="doSpellingSuggestion">  
    <input message="typens:doSpellingSuggestion"/>  
    <output message="typens:doSpellingSuggestionResponse"/>  
  </operation>  
  <operation name="doGoogleSearch">  
    <input message="typens:doGoogleSearch"/>  
    <output message="typens:doGoogleSearchResponse"/>  
  </operation>  
</portType>
```

Einschub: Erweiterungselemente

- Erweitern WSDL-Elemente um technologiespezifische Details
- Müssen einem von WSDL verschiedenen namespace zugeordnet sein
- Werden üblicherweise in den <binding> und <port> Element eingesetzt
- Jedes Protokoll hat seinen eigenen Satz an Erweiterungselementen

<binding>

- Beschreibt die Bindung einer abstrakten Einheit (message, operation, portType) an ein Protokoll
- Erhält als Attribut ein portType - Element
- Bindet jede Operation an Protokoll
- Legt Formatierung für Nachrichten fest

<binding> - GoogleSearch

```
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  ...
  <operation name="doGoogleSearch">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded"
        namespace="urn:GoogleSearch"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded"
        namespace="urn:GoogleSearch"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation></binding>
```

<service>

- Gruppe zusammengehörender Ports
- <port> definiert Kommunikationsendpunkt durch Adresse und Protokollbindung
- Verschiedene Ports können Alternativen zueinander sein
- Ports kommunizieren nicht untereinander

<service> - GoogleSearch

```
<service name="GoogleSearchService">  
  <port name="GoogleSearchPort"  
    binding="typens:GoogleSearchBinding">  
    <soap:address  
      location="http://api.google.com/search/beta2"/>  
    </port>  
  </service>
```

Software

- Omniopera – Graphischer WSDL, XML und XSD Editor
- IBM`s Web Services Toolkit
- Microsoft SOAP Toolkit 2.0

Literatur

- <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- <http://www.w3.org/TR/wsdl12-bindings>
- Scott Short: Webdienste mit dem .NET Framework entwickeln. Microsoft Press 2002 (Kapitel 5)
- <http://www.ibm.com/developerworks/web/library/w-ovr/>

Zusammenfassung

- WSDL ist XML–basierte IDL
- Unterstützt Plattform unabhängige Entwicklung und Interaktion von Web Services
- Basierung auf weitverbreiteten Techniken (xml, http) ermöglicht einen großen Einsatzbereich