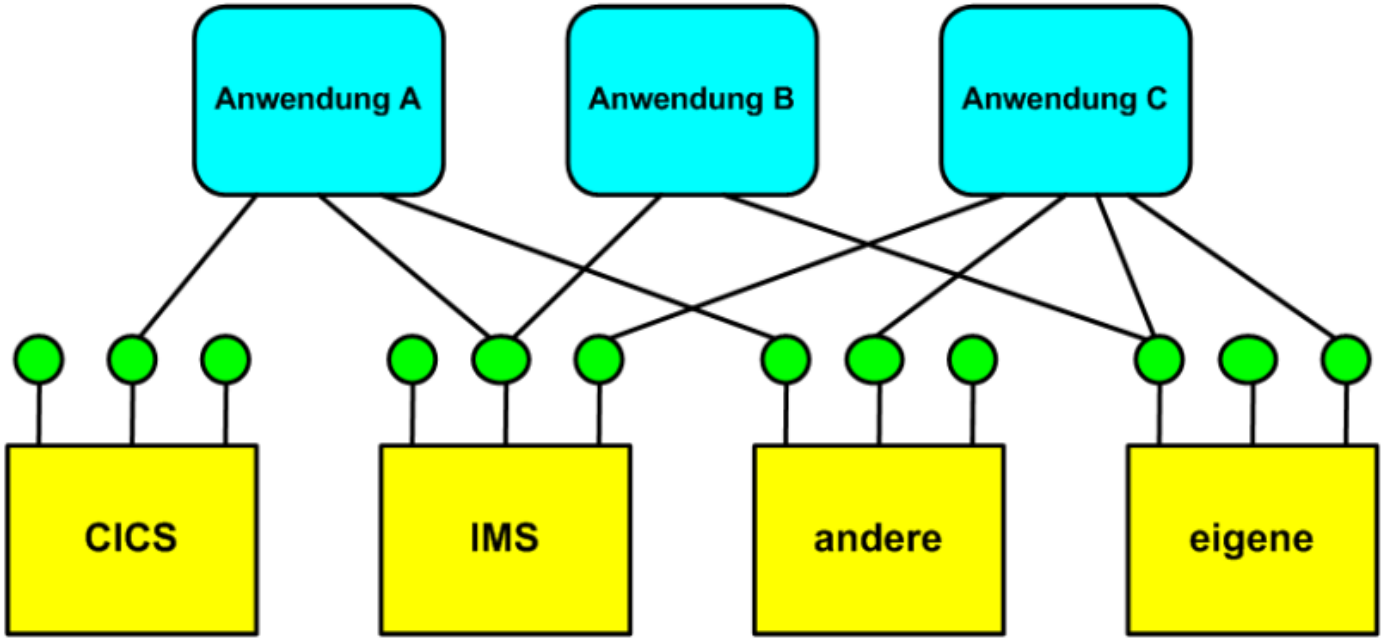
Mainframe Internet Integration

Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

SS2013

Service Oriented Architecture Teil 2

Web Services



Web Service als Lösung für das Application Integration Problem

Frage: Wie können in einem Unternehmen die unterschiedlichen Klientenanwendungen (unterschiedliche Plattformen, unterschiedliche Betriebssysteme, unterschiedliche Sprachen) flexibel auf einen größeren Satz Serveranwendungen zugreifen ?

Die moderne Antwort ist, indem die Kommunikation zwischen Klient und Server als „Web Service“ implementiert wird.

Im Prinzip ist dies ein Service, den man über ein Standard Internet Protokoll (wie z.B. SMTP, FTP und andere) aufrufen kann, und der XML für die Beschreibung von Daten, Nachrichten, Schnittstellen usw. benutzt. Gemeint ist aber fast immer ein Service der über http mit dem Simple Object Access Protocol (SOAP) aufgerufen wird. Die Schnittstelle des Services wird hierbei mittels der Web Service Definition Language (WSDL) beschrieben.

Was ist ein Web Service?

Von der W3C Web Services Architecture Working Group stammt die folgende Definition für Web Services:

“A Web Service is a software application identified by a Uniform Resource Identifier *(*URI*)* , whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web Service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols.

A uniform Resource Identifier (URI) is a compact string of characters used to identify or name a resource on the Internet.”

Beispiel Yahoo Portal

Die meisten Yahoo Dienste kommen in Wirklichkeit von anderen Web Sites: Reisen, Wetter, Landkarten, oder Web Suche mit Hilfe von Google. Diese Dienste sind in das Yahoo Portal als eigene Dienste integriert.

Uniform Resource Identifier

Ein Uniform Resource Identifier (Abk. URI; engl. für „einheitlicher Bezeichner für Ressourcen“) ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer abstrakten oder physischen Ressource dient. URIs werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, Aufruf von Web Services, aber auch z. B. E-Mail-Empfängern) im Internet und dort vor allem im WWW eingesetzt.

Beispiele für Uniform Resource Identifier:

- **http://de.wikipedia.org/wiki/Uniform_Resource_Identifier**
- **ftp://ftp.is.co.za/rfc/rfc1808.txt**
- **file:///etc/fstab**
- **geo:48.33,14.122;u=22.5**
- **ldap://[2001:db8::7]/c=GB?objectClass?one**
- **mailto:John.Doe@example.com**
- **tel:+1-816-555-1212**
- **telnet://192.0.2.16:80/**

Der modernste Remote Procedure Call

Nach dem Sun RPC, dem DCE RPC, CICS DPL, Corba und RMI ist ein Web Service die modernste Ausprägung eines Remote Procedure Calls.

Hierbei definiert SOAP die Implementierung des eigentlichen RPC. WSDL ist die Beschreibung der Schnittstelle. Das Besondere eines Web Service besteht darin, dass

- **Nachrichten zwischen Klient und Server im XML Format übertragen werden,**
- **die Beschreibung der Schnittstelle eines Services ebenfalls im XML Format (in WSDL) erfolgt.**

Ähnlich wie der CORBA Naming Service und der Java JNDI (oder Java Registry) verfügt auch der Web Service über einen eigenen Naming und Directory Service, als „Universal Description, Discovery and Integration“ (UDDI) bezeichnet.

UDDI ist jedoch umstritten und wird nur begrenzt eingesetzt. In vielen Fällen verwenden Web Services vorhandene Directory Dienste wie LDAP oder Microsoft Active Directory, die praktisch den gleichen Funktionsumfang bieten.

Web Services erlauben vielfach eine „quick and dirty“ Implementierung eines Prototyps. Geht man über das Prototyp Stadium hinaus, wächst die Komplexität einer Implementierung oft dramatisch an. Was ursprünglich fehlte:

- **Sicherheit - HTTPS kann benutzt werden, ist aber unabhängig vom Web Service Mechanismus,**
- **skaliert schlecht,**
- **Transactionssteuerung, Flußsteuerung,**

wurde zwischenzeitlich durch Erweiterungen des Web Service Standards verbessert, auf Kosten der Komplexität.

Das Simple Object Access Protocol (SOAP) ist weder simple noch Objekt-orientiert. Deshalb benutzt man heutzutage nur noch den Begriff SOAP, nicht aber mehr den Begriff "Simple Object Access Protocol“.

Web Services Technologien

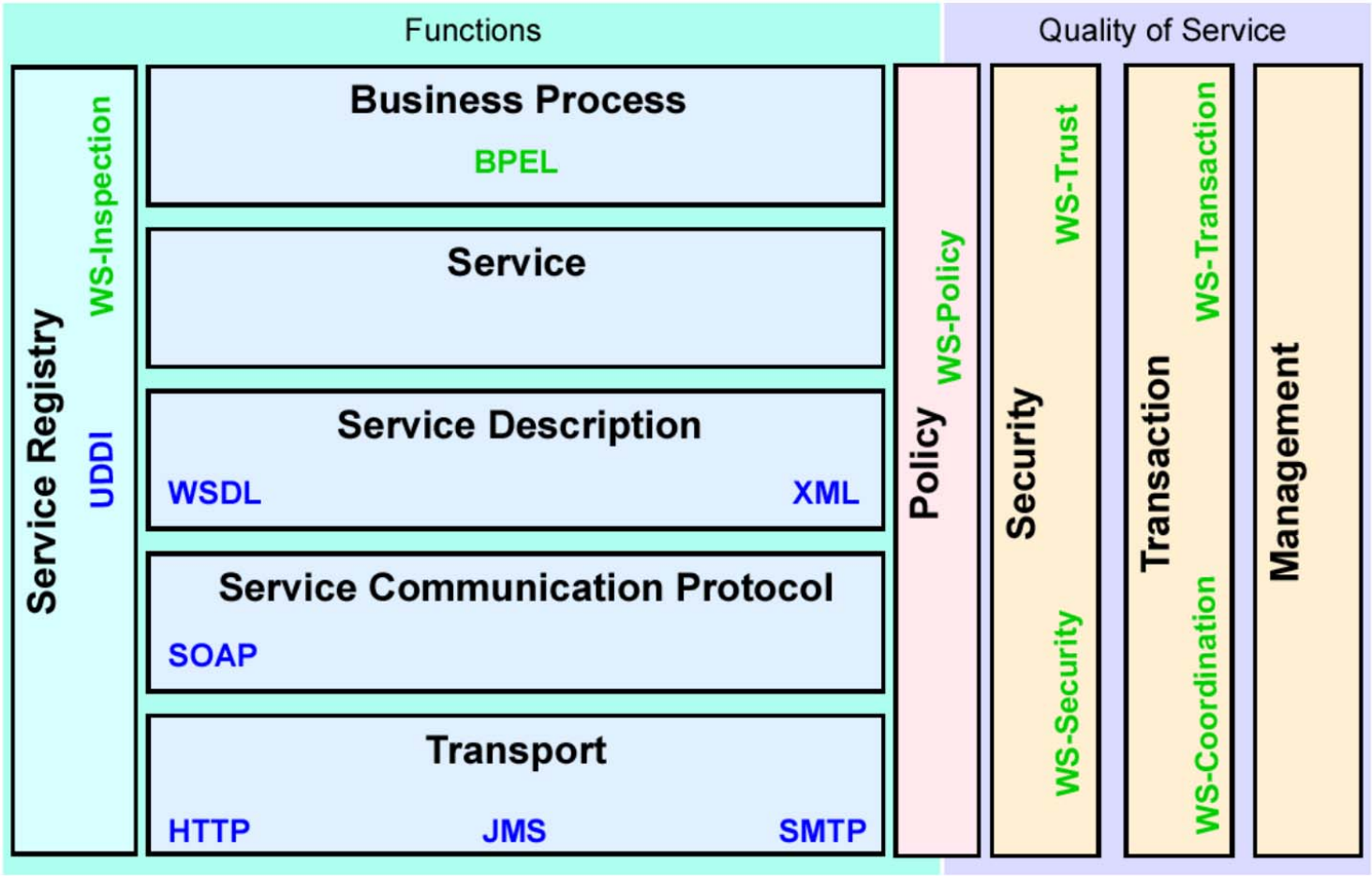
SOAP ist ein Remote Procedure Call (RPC) Protokoll. Es benutzt häufig HTTP für den Transport (Alternativen sind alle Protokolle, die Text übertragen können, z.B. FTP, MQSeries). XML beschreibt das Format der übertragenen Daten. Es ersetzt damit z.B. die IDL, die beim klassischen RPC oder bei CORBA benutzt wird, oder die Schnittstellen-Beschreibung bei Java RMI. SOAP ist unabhängig von der verwendeten Programmiersprache und dem jeweiligen Betriebssystem.

WSDL ist eine XML- Beschreibung der Schnittstellen-Definitionen eines Web Service, vergleichbar mit der Schnittstellen-Beschreibung bei Java RMI. WSDL beschreibt Formate der Anforderungs- und Antwort-Nachrichtenströme, mit denen Funktionsaufrufe an andere Programm-Module abgesetzt werden.

UDDI ist ein XML Dienstekatalog (vergleichbar mit einer XML Version von LDAP). UDDI stellt ein Verzeichnis von Adress- und Produktdaten sowie Anwendungs-Schnittstellen der verschiedenen Web Service Anbieter dar.

Die Web Services Entwicklung begann als eine IBM - Microsoft Kooperation, und erhielt breite Unterstützung in der Industrie. Spätere Erweiterungen lösten das Firewall - HTTP Problem, verbesserten Sicherheit und Reifegrad, und fügten Transaktionsdienste hinzu.

Web Services bilden einen Baustein für eine Service-Orientierte Architektur (SOA), siehe Teil 4.



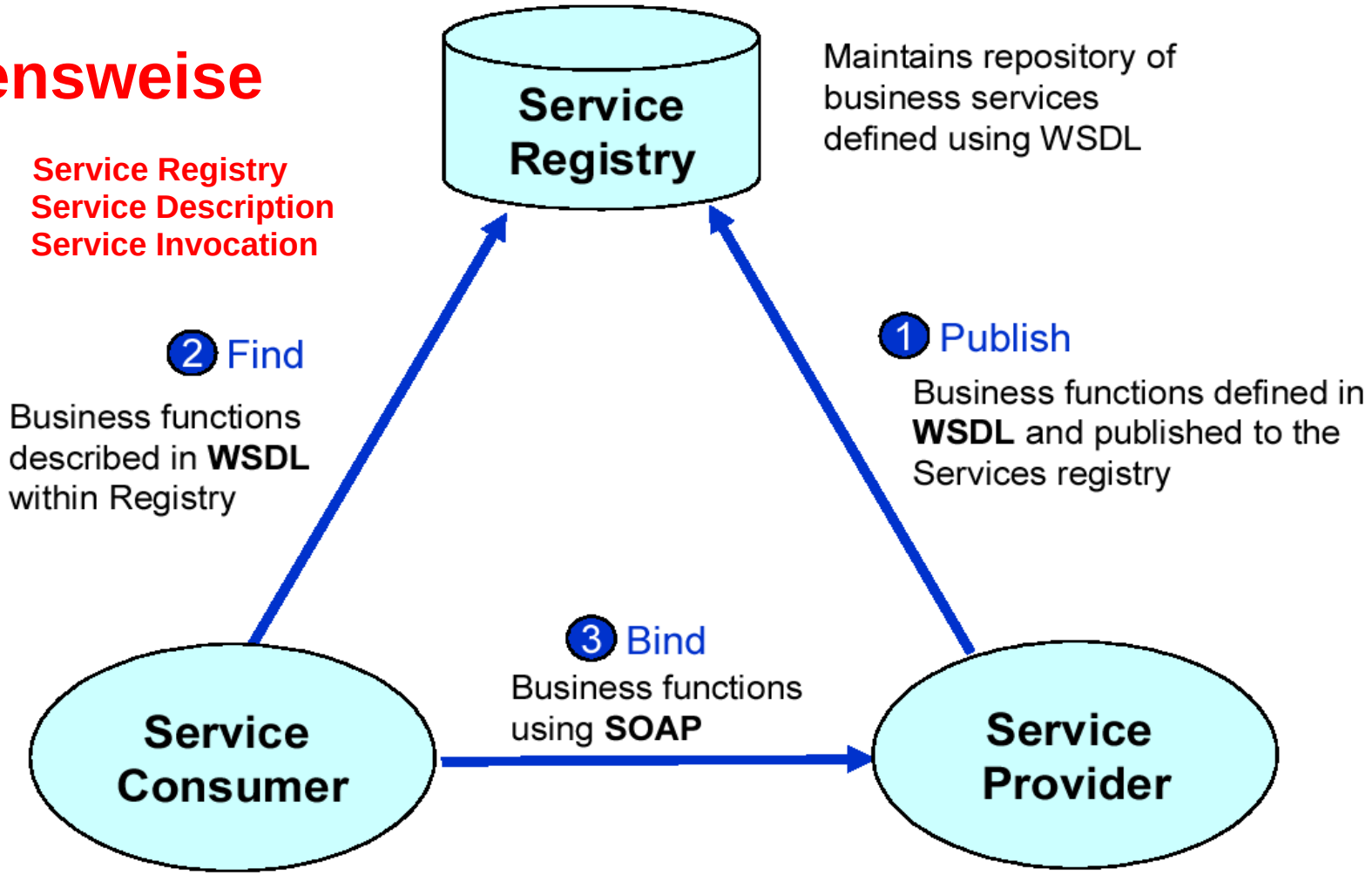
Bestandteile des Web Services Architecture Stack

Über die Jahre entwickelte sich Web Services zu einer komplexen, vielschichtigen Architektur. Die ursprüngliche Konfiguration bestand aus SOAP, WSDL und UDDI und spezifizierte http als Übertragungsprotokoll.

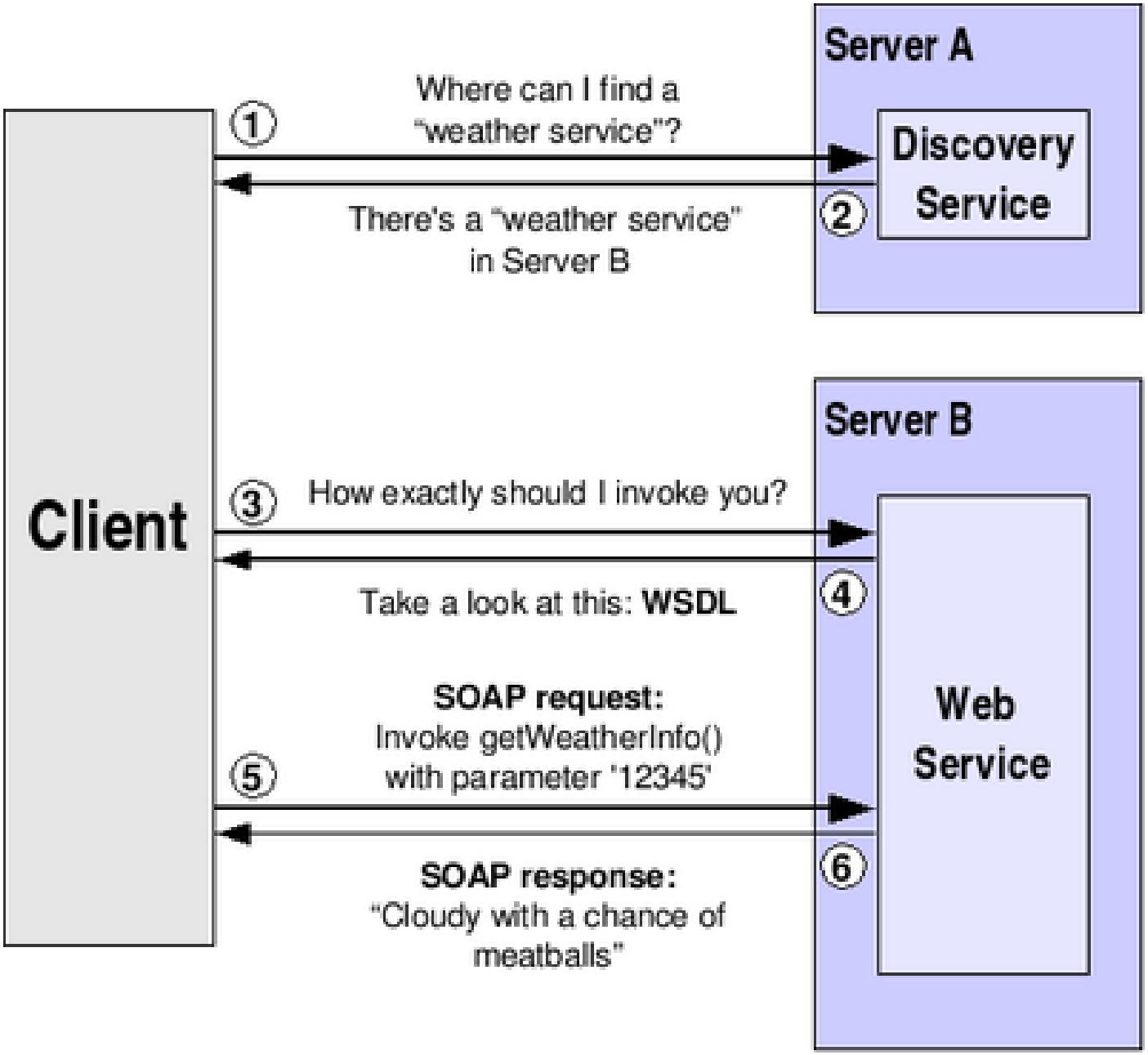
BPEL (Business Process Evaluation Language) wird in Teil 4 erläutert.

Vorgehensweise

UDDI (or Alternative); Service Registry
WSDL: Service Description
SOAP: Service Invocation



1. **Ein Service Provider (Server) erstellt einen neuartigen Web Service und publiziert diese Tatsache durch einen Eintrag (Reklame) in einem Verzeichnis, dem Registry. UDDI ist der Registry Standard für Web Services; es können aber (und werden häufig) auch andere Registry Implementierungen benutzt werden.**
2. **Ein Interessent (Klient, Service Consumer) findet einen ihn interessierenden Service im Registry und liest die Beschreibung und Zugriffsdaten.**
3. **Der Klient greift auf den Service zu.**



Beispiel

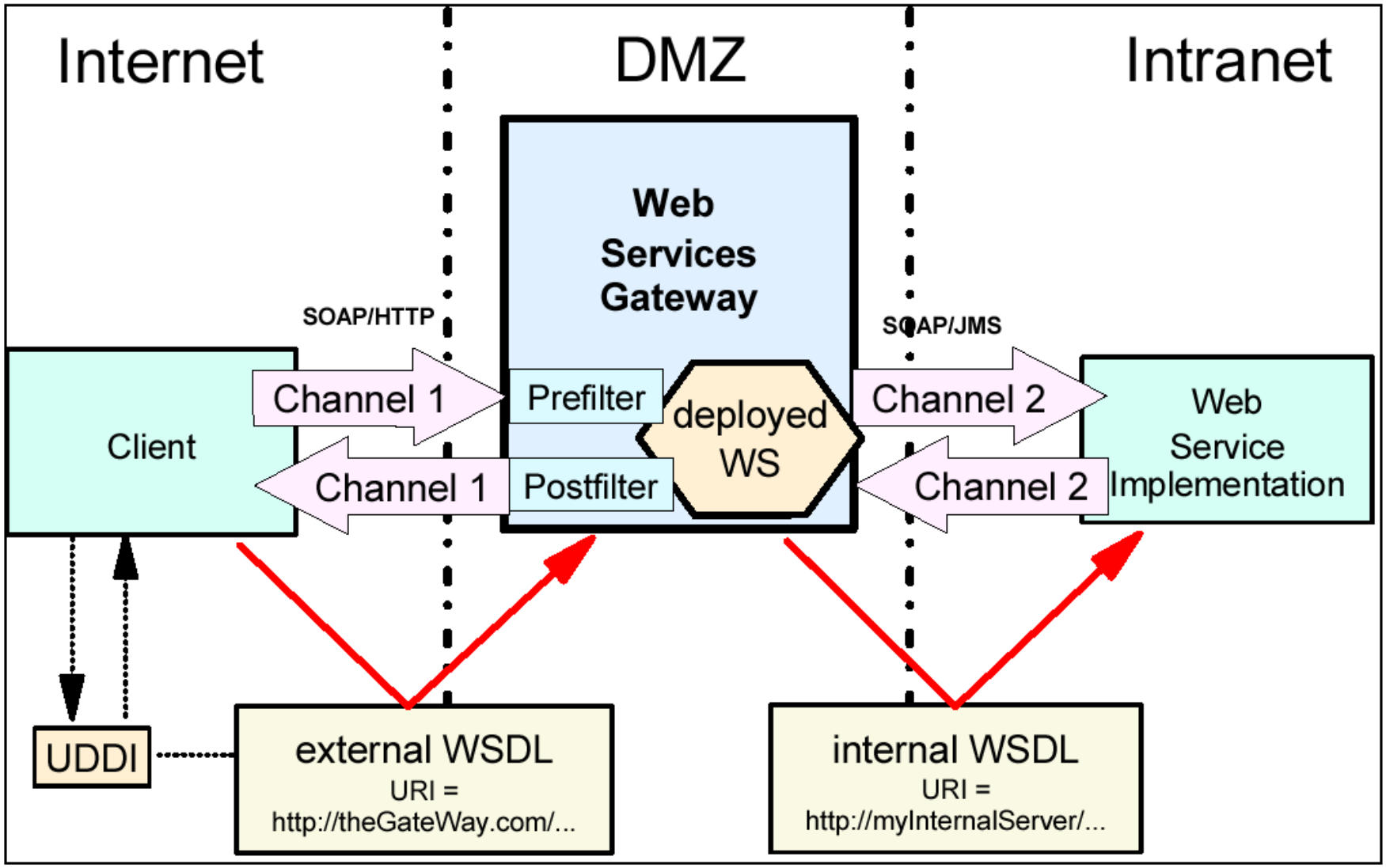
1. Ein Klient, der an einem Wetterbericht interessiert ist, greift auf einen UDDI Server zu (Server A).

2. Der UDDI Server (Registry) schlägt dem Klienten den Server B vor. Server B stellt Wetter-berichte zur Verfügung.

3. Der Klient greift auf den WSDL Service des Servers B zu.

4. Die WSDL Antwort (im XML Format) enthält Anweisungen, wie man auf den Wetterdienst zugreift.

5. Der Klient benutzt das SOAP Protokoll um die Wetterinformation abzufragen.



Die Benutzung von HTTP und Port 80 ermöglicht Internet Zugriff und umgeht den Enterprise Firewall. Für eine quick und dirty Feasibility Demonstration ist das attraktiv. Nach der Erweiterung um Industry Strength Eigenschaften, z.B. durch ein Web Services Gateway, werden Web Services so komplex wie z.B. CORBA.

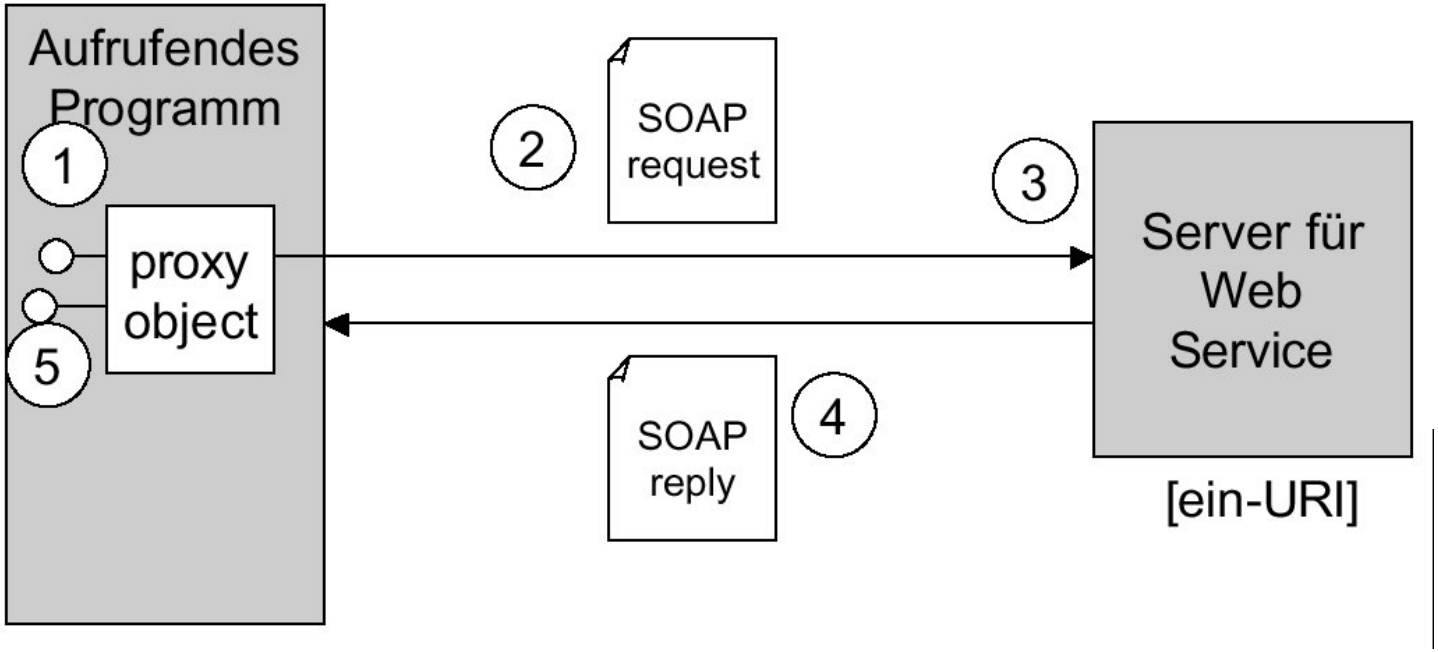
SOAP

SOAP ist ein zustandsloses Protokoll (wie http), welches zum Austausch von One-way Nachrichten zwischen SOAP Klienten (Sender) und SOAP Server (Empfänger) konzipiert ist. Mit SOAP lassen sich darüber hinaus komplexere Interaktionsformen (beispielsweise Request/Response oder Request/Multiple Response) anhand der Kombination von einfachen Nachrichten und zugrundeliegenden Transportprotokollen realisieren.

SOAP macht prinzipiell keine Aussagen über

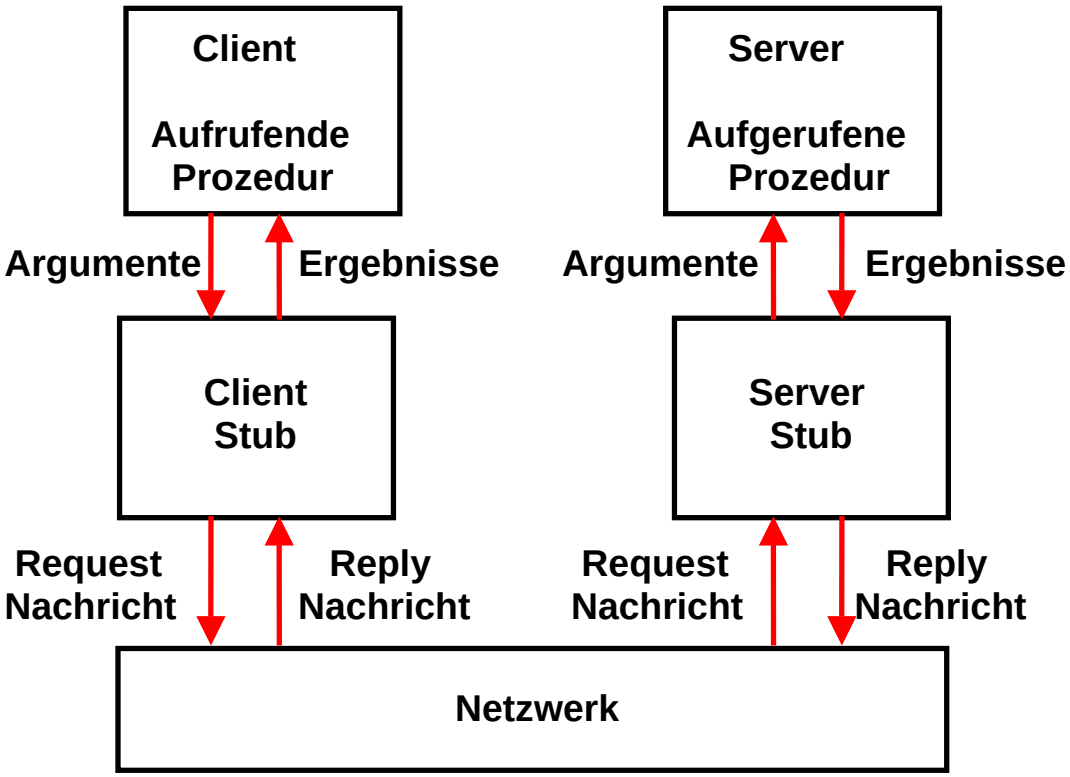
- **die Semantik der zu transportierenden Daten,**
- **das Routing von Nachrichten,**
- **Verlässlichkeit des Transfers oder**
- **die Überwindung von Firewalls.**

Es stellt lediglich ein Framework zum Nachrichtenaustausch dar. Hierzu baut es auf vorhandenen Transport-protokollen wie HTTP oder SMTP auf.



Beim SOAP Remote Procedure Call erfolgt der Aufruf des remote Web Service über eine HTTP Message Request. Die Antwort des Web Service Servers erfolgt über eine HTTP Message Response.

An Stelle einer IDL (Interface Definition Language) Beschreibung wird eine XML Beschreibung eingesetzt.



Remote Procedure Call

Client und Server laufen als zwei getrennte Prozesse.

Die beiden Prozesse kommunizieren über Stubs. Stubs sind Routinen, welche Prozedur-Aufrufe auf Netzwerk RPC Funktionsaufrufe abbilden.

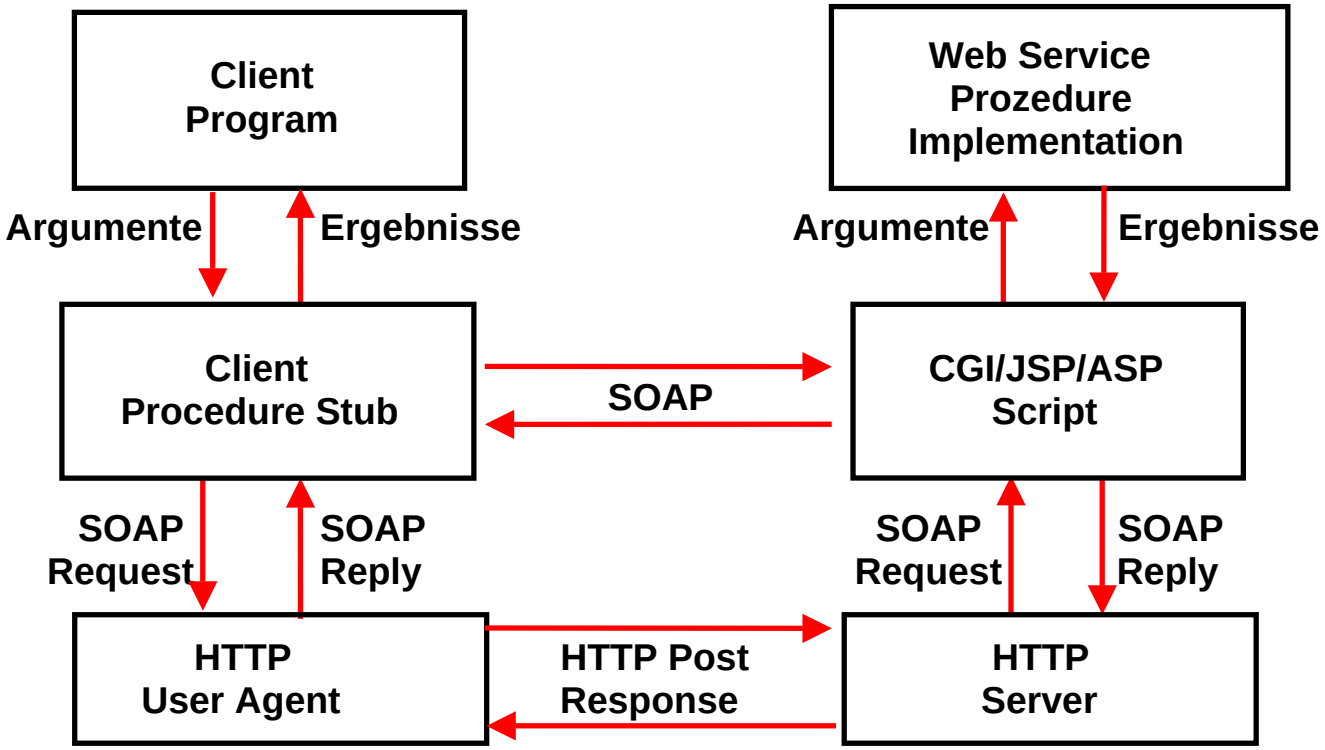
Ein Server-seitiges RPC Programm stellt den Klienten seine Dienste über eine Interface (Schnittstelle) zur Verfügung. es definiert seine Interface unter Benutzung einer Interface Definition Language (IDL).

Ein Stub Compiler liest die IDL Schnittstellen-Beschreibung und produziert zwei Stub Proceduren für jede Server Procedure (client side stub und server side stub).

Der klassische RPC benutzt die Bezeichnung Server Stub. Modernere RPCs verwenden statt dessen die Bezeichnung Skeleton. Die Bezeichnungen Server Stub und Skeleton sind austauschbar.

Stubs bzw. Skeletons implementieren eine RPC Runtime. Sie übersetzen Aufrufe der Client API in Sockets und dann in Nachrichten, die über das Netz übertragen werden.

Beim CICS Distributed Program Link (DPL) benötigt der Aufruf eines entfernten Systems lediglich dessen Name (TRID) und die Bezeichnung der COMMAREA. Eine IDL ist nicht erforderlich.



Web Services Remote Procedure Call

Der Client Procedure Stub erzeugt eine SOAP Message, die an den Server Stub weitergegeben wird.

Client und Server Stubs sind häufig in Java geschrieben und benutzen z.B. Java Server Pages. Alternativen sind CGI Programme oder (im Falle von Microsoft Implementierungen) Active Server Pages. Aber auch Cobol ist auf der Server Seite häufig zu finden.

Soap (1)

Für die Nutzung entfernter Ressourcen und zum Aufruf entfernter Methoden gibt es mehrere klassische Ansätze. Ein einfaches Verfahren ist die Anfrage mit Parametern an einen Web-Server, der den Inhalt zurückliefert.

Zur Verteilung von Programmen haben sich CORBA, RPC, aber auch RMI bewährt. Web Services haben den Vorteil, dass das Protokoll HTTP verbreitet ist.

CORBA und RMI haftet der Nachteil an, dass ihre Kommunikation einen offenen Port erfordert, den Sicherheitsbeauftragte in einem Unternehmen nicht unbedingt tolerieren. Zudem sind CORBA/RMI Lösungen nicht wirklich einfach zu implementieren. Idealerweise verheiratet eine Technik beide Vorteile: HTTP-Kommunikation und objektorientierte Konzepte. Web Serves benutzen HTTP und damit standardmäßig Port 80.

HTTP hat den Vorteil (und den Nachteil) dass es für Web-Seiten in der Regel keine Beschränkungen durch Firewalls gibt. Die beiden Kommunikationsparteien regeln mit SOAP einen entfernten Methodenaufruf, der die Parameter und Ergebnisse in XML kodiert und überträgt.

Durch die unabhängigen und anerkannten Web-Standards bietet SOAP gegenüber RMI oder CORBA und auch sonstigen RPC Protokollen den Vorteil, nicht an eine Programmiersprache gebunden zu sein.

SOAP selbst beschreibt die Art und Weise, wie Inhalte (Serialisierung) und die XML-Daten übertragen werden. Die Übertragung selbst hat mit SOAP aber eigentlich nichts zu tun, denn dafür sind andere Protokolle definiert.

Soap (2)

Da Microsoft federführend bei der Spezifikation ist und das Dotnet-Framework seine Kommunikation vollständig auf SOAP aufbaut, wird es in der Zukunft eine Reihe von Java-Clients geben, die auf SOAP-Dienste von anderen Anbietern aufbauen, z.B. Microsoft.

Den Vorteilen stehen allerdings auch einige Nachteile gegenüber.

Die XML-Repräsentation der Dokumente macht die Datenmengen groß; ein Parsen der Parameter und Ergebnisse auf beiden Seiten ist erforderlich. In Umgebungen, die performante Übertragung fordern – etwa bei der Kommunikation mobiles Endgerät und Server –, wird sich SOAP deshalb nicht so schnell durchsetzen. Für z/OS existiert ein spezielles Hardware/Software Produkt, die XML Data Appliance, welche auf der Server Seite den XML Parsing Aufwand verbessert. . Siehe http://jedi.informatik.uni-leipzig.de/de/Vorles/Integration/zbx/zBX03.pdf#page=22

Des Weiteren ist auch die Sicherheit von SOAP-Verbindungen problematisch. Ein Client könnte sich mit einem Server verbinden und einen Aufruf starten, obwohl die Berechtigung fehlt. Sicherheitseigenschaften müssten erst auf der Server-Seite implementiert werden.

Die in Klartext übertragenen Nachrichten bilden ein weiteres Problem, was jedoch durch SSL gelöst werden kann.

Verteilte Programmierung mit RMI und SOAP http://openbook.galileodesign.de/javainsel5/javainsel18_000.htm

Technische Realisierung von SOAP

Ein Client-Programm besorgt sich, wie bei entfernten Programmen üblich, eine Referenz auf das entfernte Objekt. Das ist eine URL auf einem Server. Der Server empfängt eine normale HTTP- POST-Anfrage. Diese enthält eine XML-kodierte Nachricht (Content-Typ ist einfach text/hmtl), in der die aufzurufende Methode und ihre Parameter kodiert sind. Der Server nimmt diese Nachricht entgegen, parsed das empfangene XML-Dokument und leitet die Anfrage an die Methode weiter. Diese produziert die Ausgabe, die wiederum als XML-Dokument über die Antwort vom Server zum Client geschickt wird. Der Client nimmt das Ergebnis entgegen, und die Kommunikation ist beendet.

SOAP bietet für entfernte Methodenaufrufe einige Standard-Datentypen an. Zu diesen gehören einfache Datentypen wie Ganzzahlen, Fließkommazahlen, Zeit- und Datumsangaben und Binärdaten.

Weiterhin unterstützt SOAP zusammengesetzte Datentypen wie Strukturen und Arrays. Wie diese Daten nun tatsächlich in eine XML-Nachricht umgesetzt werden, braucht man glücklicherweise nicht zu wissen. Als Endanwender kommen wir mit der Nachricht nicht in Kontakt.

Web Service Definition Language
WSDL

Die Web Services Description Language (WSDL) ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Web Services zum Austausch von Nachrichten auf Basis von XML. WSDL ist ein industrieller Standard des World Wide Web Consortium W3C.

WSDL beschreibt einen Web Service als eine Menge von Schnittstellen, die mehrere mögliche Interaktionen anhand von Operationen mit einer Anwendung spezifizieren. Auf die Anwendung kann über definierte Adressen unter Verwendung festgelegter Kommunikationsprotokolle zugegriffen werden. Es werden im Wesentlichen die Operationen definiert, die von außen zugänglich sind, sowie die Parameter und Rückgabewerte dieser Operationen. Im Einzelnen beinhaltet ein WSDL-Dokument funktionelle Angaben zu:

- **der Schnittstelle**
- **Zugangsprotokoll**
- **Alle notwendigen Informationen zum Zugriff auf den Service, in maschinenlesbarem Format**

WSDL Beschreibungselemente

Für die WSDL Beschreibung eines Web Services werden sechs XML Sprachelemente definiert:

types
Definition der Datentypen, die zum Austausch der Nachrichten benutzt werden

message (Nachricht)
Abstrakte Definitionen der übertragenen Daten, bestehend aus mehreren logischen Teilen, von denen jeder Teil mit einer Definition innerhalb eines Datentypsystems verknüpft ist.

portType (andere Bezeichnung Interface)
Eine Menge von abstrakten Arbeitsschritten (Operations), die von einem (oder mehreren) Ports unterstützt werden. Es existieren vier Typen (Operations) von ausgetauschten Nachrichten:

- **One-way: Der Service bekommt eine Input-Message vom Client.**
- **Request-response: Der Service bekommt einen Request (Input-Message) vom Client und sendet eine Antwort (Output-Message).**
- **Solicit-response: Der Service sendet eine Message und erwartet eine Antwort vom Client.**
- **Notification: Der Service sendet eine Output-Message.**

binding (Bindung)
Bestimmt das konkrete Protokoll und Datenformat für die Arbeitsschritte und Nachrichten, die durch einen bestimmten Port-Typ gegeben sind. Das Binding wird normalerweise mittels SOAP verwirklicht: andere Möglichkeiten sind IIOP, Dotnet, Java Message Service (JMS), oder WebSphere MQ

port (andere Bezeichnung endpoint)
Spezifiziert eine Adresse für eine Bindung, also eine Kommunikationsschnittstelle, üblicherweise ein URI. Eine Menge von Ports definiert normalerweise einen Service. In WSDL 2.0 wurde die Bezeichnung zu Endpoint geändert.

service (Service)
Fasst eine Menge von verwandten Ports zusammen.

Kompliziert, nicht war ?

Dies soll nur ein Überblick sein. Falls Sie tiefer einsteigen wollen: Die Zusammenhänge werden wesentlich klarer, wenn Sie die Web Service Tutorials auf unserem Rechner durchführen.

Siehe hierzu

http://www.informatik.uni-leipzig.de/cs/Tutor/RDz/WebTutorials/index.html