

# »Web Services in der Praxis«

## **Teleseminar Web Services SS 2004**

Universität Mannheim / Universität Karlsruhe

Georg Hackenberg

[georg@eeco.de](mailto:georg@eeco.de)

Fabrizio Branca

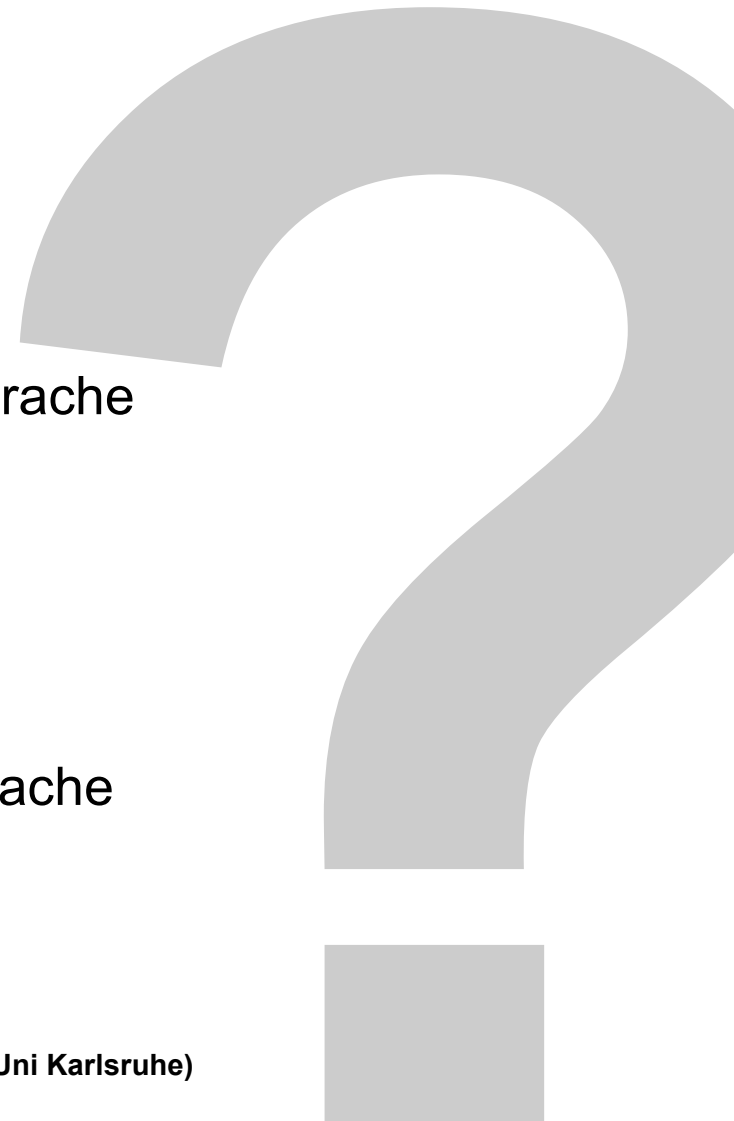
[mail@fabrizio-branca.de](mailto:mail@fabrizio-branca.de)

20.07.2004

# Umfrage

Wer hat schon einmal einen Web Service...

- Server in Java
  - Server in .Net
  - Server in PHP
  - Server in einer anderen Programmier- / Skriptsprache
  - Client in Java
  - Client in .Net
  - Client in PHP
  - Client in einer anderen Programmier- / Skriptsprache
- ...implementiert?



# Einführung

- Server in Java
- Server in .Net
- Server in PHP
- Server in einer anderen Programmier- / Skriptsprache
- Client in Java
- Client in .Net
- Client in PHP
- Client in einer anderen Programmier- / Skriptsprache

# Übersicht

- **PHP**
  - APIs (NuSOAP)
  - Konsumieren von Web Services (am Beispiel Google)
  - Erstellen eines Web Services (am Beispiel eines Adressbuchs)
- Entwicklungsumgebung „Eclipse“
- **Java**
  - Server
    - Vorstellung der Umgebung
    - Erstellen von Web Services
    - Automatisieren
  - Client

Web Services mit...

PHP



# PHP und Web Services

- Warum PHP?
  - Einfachheit und Mächtigkeit der Sprache (Speicherverwaltung / Klassen-Konzepte)
  - Popularität im Bereich Internet / WWW (Programmierer / Hoster)
- Unterstützung für Web Services
  - Ausgelegt auf Kommunikation via Netzwerk (HTTP, SMTP)
  - Externe Bibliotheken implementieren SOAP und die Konzepte für Client / Server

# SOAP-APIs in PHP

- SOAPx4
  - PEAR::SOAP
  - NuSOAP
  - SWSAPI (auch für Perl und Python)
- PHP-SOAP (C-Erweiterung für PHP)
- Krysalis SOAP (für das Krysalis Framework)
- eZ SOAP (für das eZ Publish CMS)

# NuSOAP im Überblick

- Entwickler: Dietrich Alaya
- Offizielle Webseite:  
<http://dietrich.ganx4.com/nusoap/>
- Schlechte Dokumentation aber viele Tutorials
- Unterstützung von SOAP 1.1 und WSDL 1.1
- Objektorientierte Implementierung
  - Klassen für Server und Client



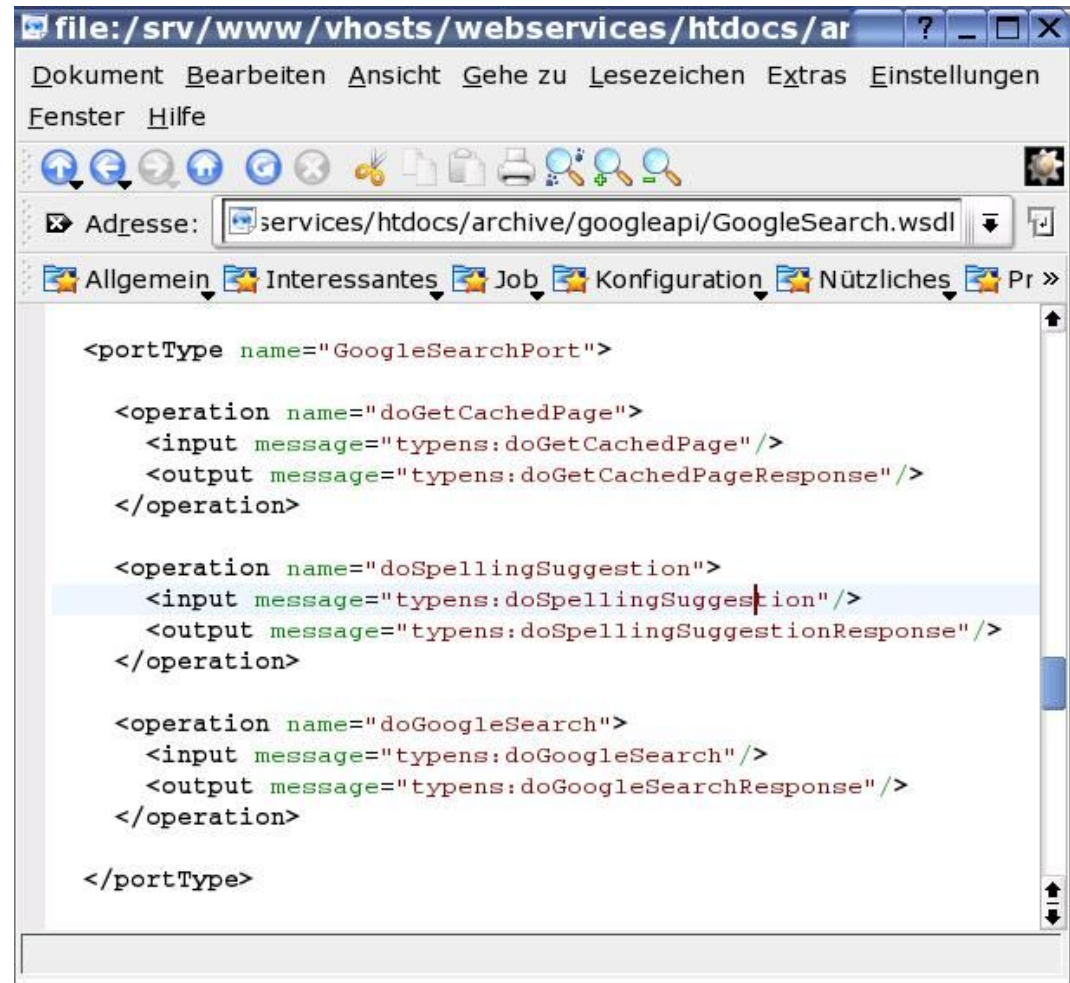
# Konsumieren eines Webservices

- Allgemein
  - Welche Operationen stellt der Service bereit?
  - Wie verwendet man diese Operation?
- NuSOAP
  - Wie sende ich eine Anfrage?
  - Wie werte ich das Ergebnis aus?

# ... am Beispiel Google

Welche Operationen stellt der Service bereit? [1]

- Quelle: WSDL-Datei
- Operationen:
  - doGetCachedPage
  - doSpellingSuggestion
  - doGoogleSearch
- Für uns interessant:
  - doGoogleSearch



The screenshot shows a web browser window with the address bar containing the file path: `file:/srv/www/vhosts/webservices/htdocs/ar...services/htdocs/archive/googleapi/GoogleSearch.wsdl`. The browser's menu bar includes options like 'Dokument', 'Bearbeiten', 'Ansicht', 'Gehe zu', 'Lesezeichen', 'Extras', and 'Einstellungen'. The address bar also shows a search bar and a dropdown menu with items like 'Allgemein', 'Interessantes', 'Job', 'Konfiguration', 'Nützliches', and 'Pr >>'. The main content area displays the XML content of the WSDL file, which defines the 'GoogleSearchPort' port type and its operations: 'doGetCachedPage', 'doSpellingSuggestion', and 'doGoogleSearch'. The 'doSpellingSuggestion' operation is highlighted in blue. The XML code is as follows:

```
<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>
```

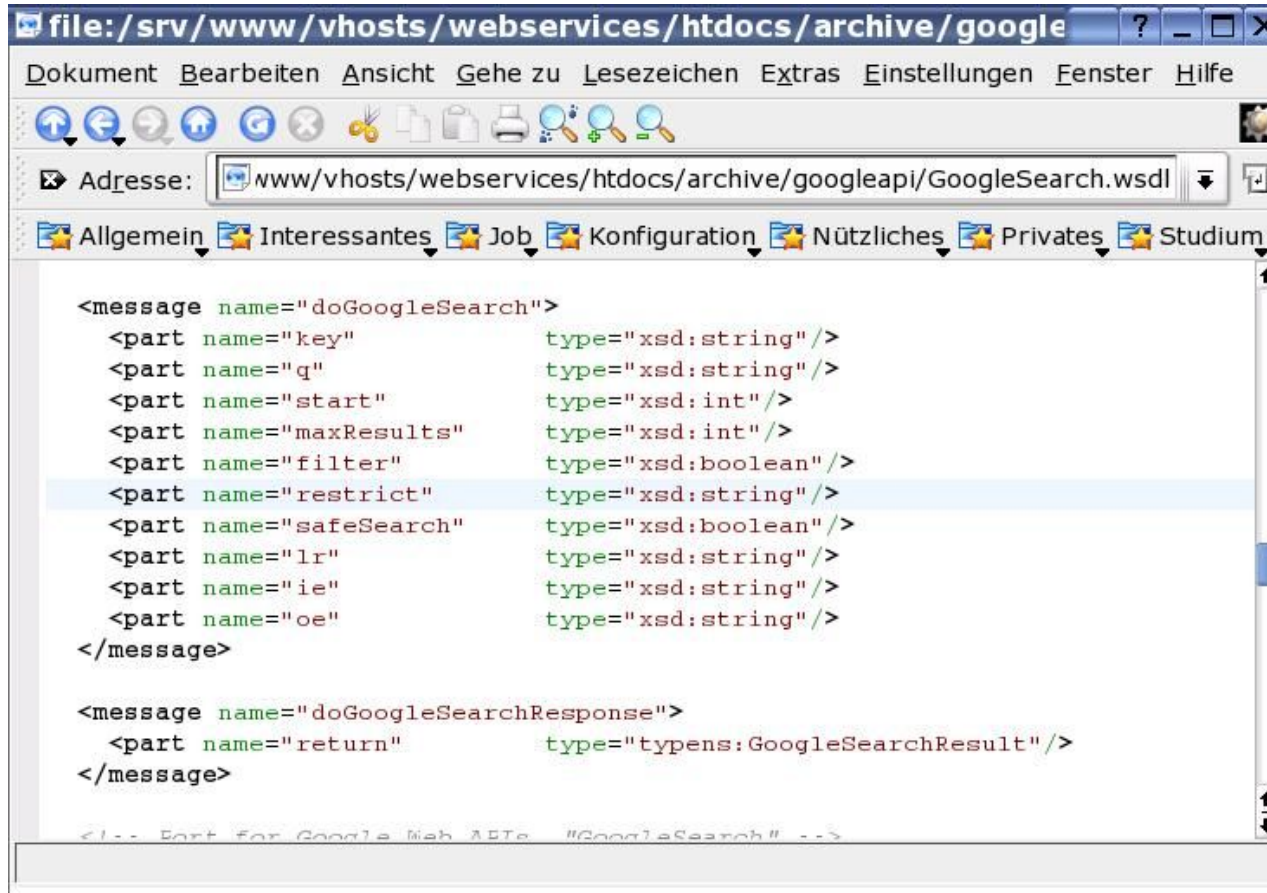
# ... am Beispiel Google (doGoogleSearch)

Wie verwendet man diese Operationen? [1]

- Quelle: WSDL-Datei
- Eingabe-Parameter (input)
  - Struktur aus `key`, `q`, `start`, ...
- Rückgabe-Wert (output)
  - Struktur aus `estimatedTotalResultsCount`, `resultElements`, `startIndex`, ...

# ... am Beispiel Google (doGoogleSearch)

Wie verwendet man diese Operationen? [2]



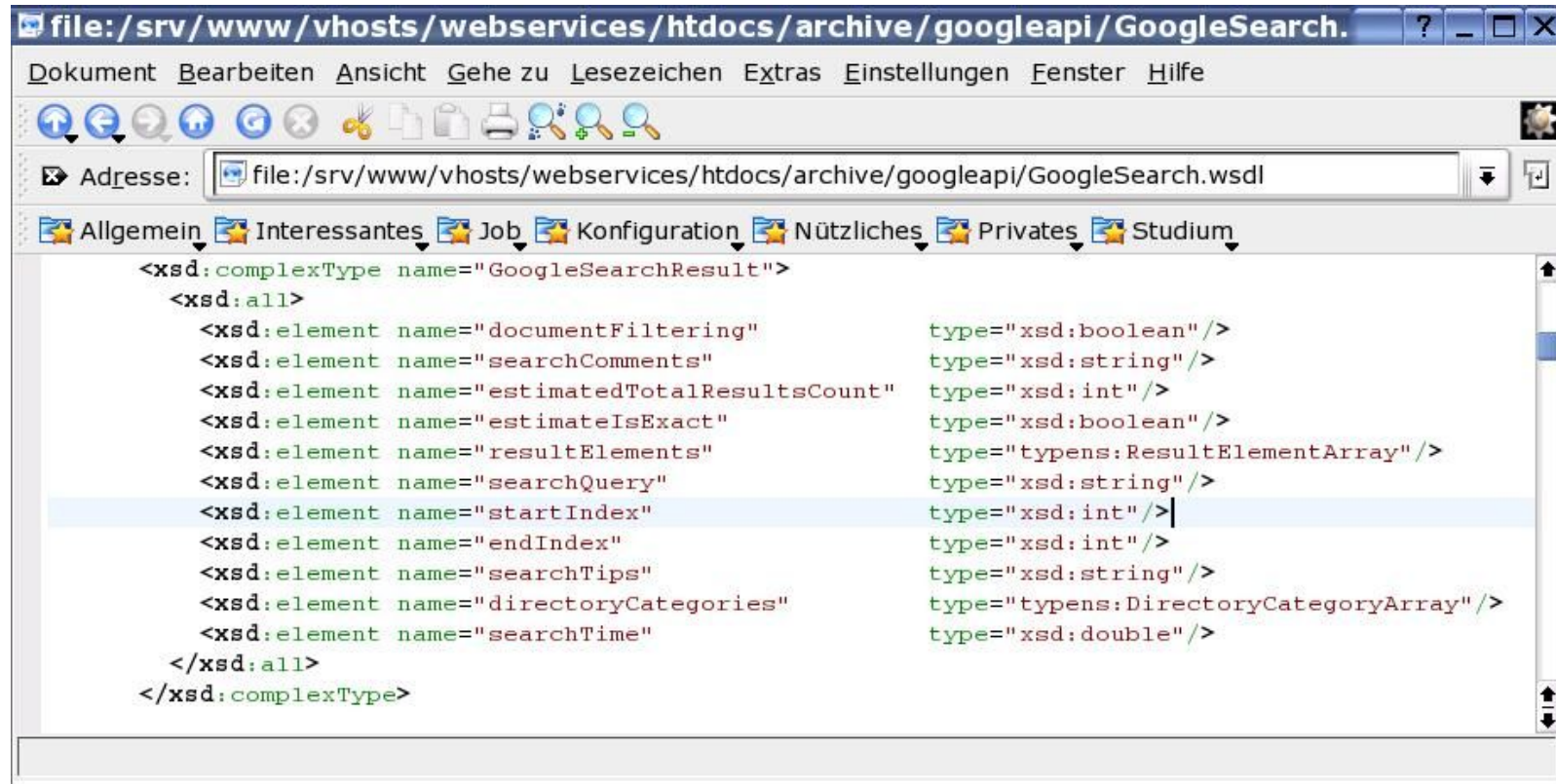
The screenshot shows a web browser window with the following details:

- Title Bar:** file:/srv/www/vhosts/webservices/htdocs/archive/google
- Menu Bar:** Dokument Bearbeiten Ansicht Gehe zu Lesezeichen Extras Einstellungen Fenster Hilfe
- Address Bar:** Adresse: www/vhosts/webservices/htdocs/archive/googleapi/GoogleSearch.wsdl
- Bookmark Bar:** Allgemein Interessantes Job Konfiguration Nützliches Privates Studium
- Main Content:** XML Schema Definition (WSDL) for the doGoogleSearch service.

```
<message name="doGoogleSearch">  
  <part name="key" type="xsd:string"/>  
  <part name="q" type="xsd:string"/>  
  <part name="start" type="xsd:int"/>  
  <part name="maxResults" type="xsd:int"/>  
  <part name="filter" type="xsd:boolean"/>  
  <part name="restrict" type="xsd:string"/>  
  <part name="safeSearch" type="xsd:boolean"/>  
  <part name="lr" type="xsd:string"/>  
  <part name="ie" type="xsd:string"/>  
  <part name="oe" type="xsd:string"/>  
</message>  
  
<message name="doGoogleSearchResponse">  
  <part name="return" type="typens:GoogleSearchResult"/>  
</message>  
  
<!-- Part for Google Web APIs "GoogleSearch" -->
```

# ... am Beispiel Google (doGoogleSearch)

Wie verwendet man diese Operationen? [3]



```
<xsd:complexType name="GoogleSearchResult">
  <xsd:all>
    <xsd:element name="documentFiltering" type="xsd:boolean"/>
    <xsd:element name="searchComments" type="xsd:string"/>
    <xsd:element name="estimatedTotalResultsCount" type="xsd:int"/>
    <xsd:element name="estimateIsExact" type="xsd:boolean"/>
    <xsd:element name="resultElements" type="typens:ResultElementArray"/>
    <xsd:element name="searchQuery" type="xsd:string"/>
    <xsd:element name="startIndex" type="xsd:int"/>
    <xsd:element name="endIndex" type="xsd:int"/>
    <xsd:element name="searchTips" type="xsd:string"/>
    <xsd:element name="directoryCategories" type="typens:DirectoryCategoryArray"/>
    <xsd:element name="searchTime" type="xsd:double"/>
  </xsd:all>
</xsd:complexType>
```

# ... am Beispiel Google (doGoogleSearch)

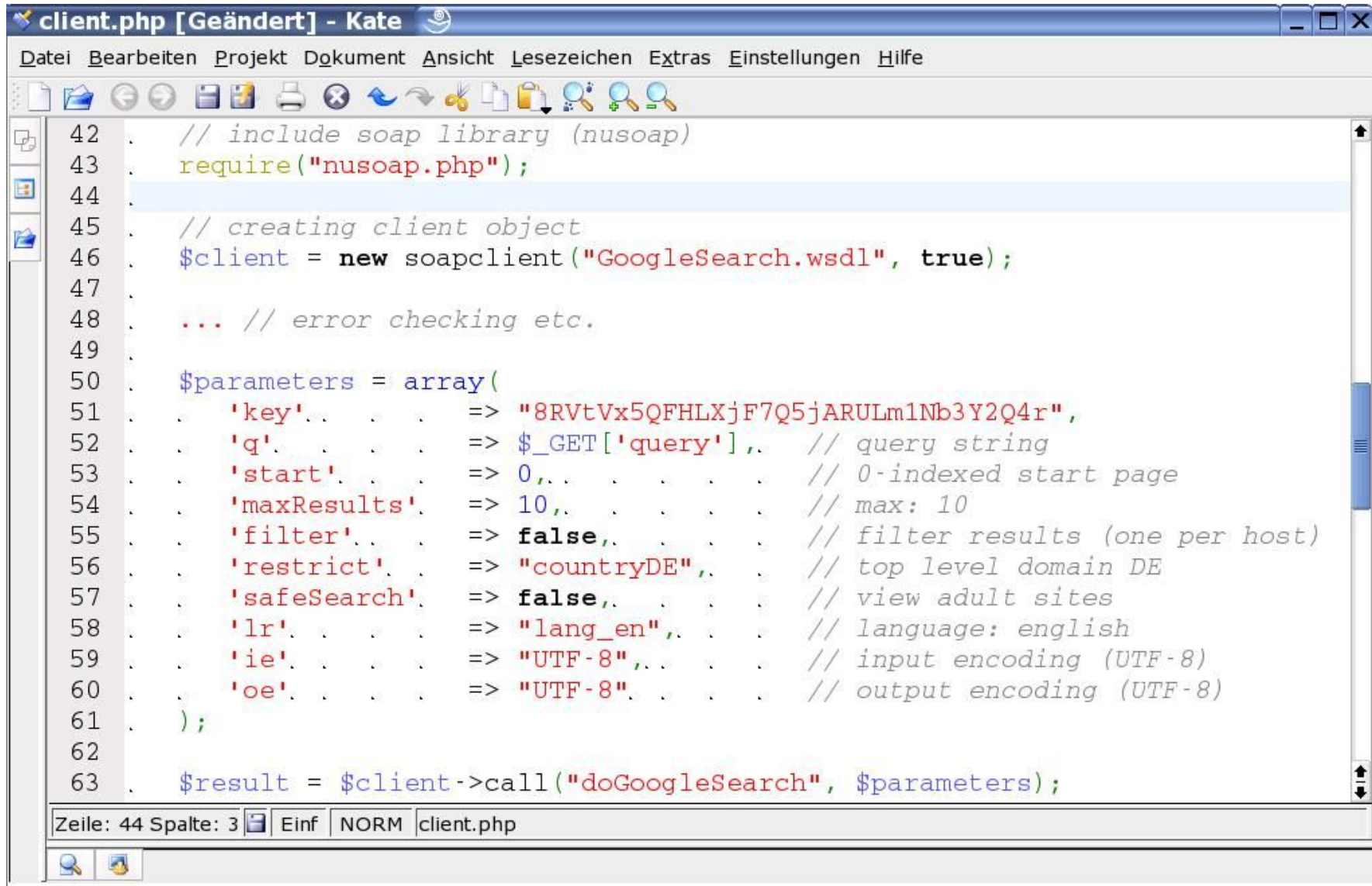
Wie verwendet man diese Operationen? [4]

```
<xsd:complexType name="ResultElement">
  <xsd:all>
    <xsd:element name="summary" type="xsd:string"/>
    <xsd:element name="URL" type="xsd:string"/>
    <xsd:element name="snippet" type="xsd:string"/>
    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="cachedSize" type="xsd:string"/>
    <xsd:element name="relatedInformationPresent" type="xsd:boolean"/>
    <xsd:element name="hostName" type="xsd:string"/>
    <xsd:element name="directoryCategory" type="typens:DirectoryCategory"/>
    <xsd:element name="directoryTitle" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>

<xsd:complexType name="ResultElementArray">
  <xsd:complexContent>
    <xsd:restriction base="soapenc:Array">
      <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:ResultElement []"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

# ... am Beispiel Google (doGoogleSearch)

Wie sende ich eine Anfrage?



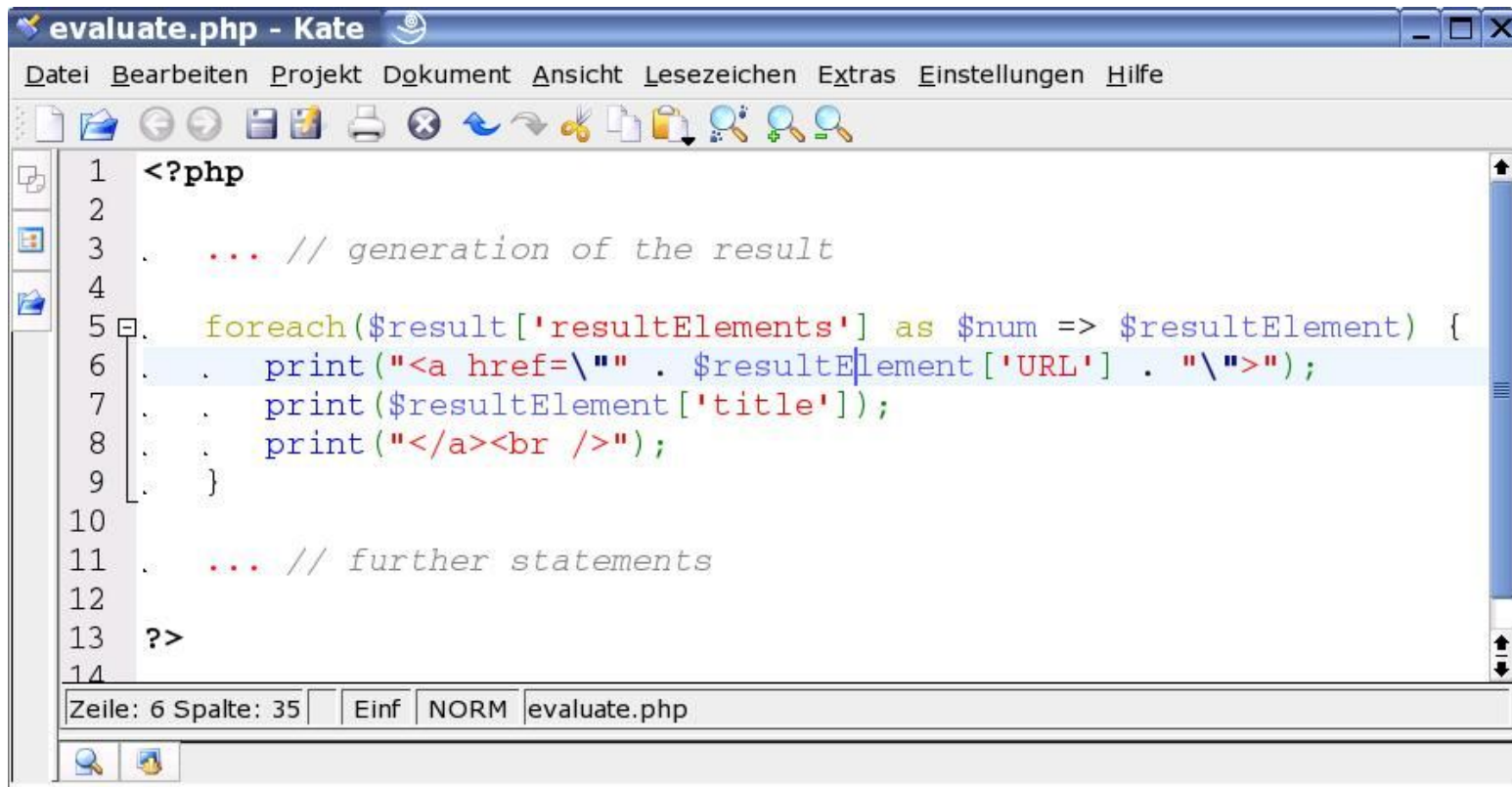
```
client.php [Geändert] - Kate
Datei Bearbeiten Projekt Dokument Ansicht Lesezeichen Extras Einstellungen Hilfe

42 // include soap library (nusoap)
43 require("nusoap.php");
44
45 // creating client object
46 $client = new soapclient("GoogleSearch.wsdl", true);
47
48 ... // error checking etc.
49
50 $parameters = array(
51     'key' => "8RVtVx5QFHLXjF7Q5jARULm1Nb3Y2Q4r",
52     'q' => $_GET['query'], // query string
53     'start' => 0, // 0-indexed start page
54     'maxResults' => 10, // max: 10
55     'filter' => false, // filter results (one per host)
56     'restrict' => "countryDE", // top level domain DE
57     'safeSearch' => false, // view adult sites
58     'lr' => "lang_en", // language: english
59     'ie' => "UTF-8", // input encoding (UTF-8)
60     'oe' => "UTF-8", // output encoding (UTF-8)
61 );
62
63 $result = $client->call("doGoogleSearch", $parameters);

Zeile: 44 Spalte: 3 Einf NORM client.php
```

# ... am Beispiel Google (doGoogleSearch)

Wie werte ich das Ergebnis aus?



The screenshot shows a Kate text editor window titled "evaluate.php - Kate". The menu bar includes "Datei", "Bearbeiten", "Projekt", "Dokument", "Ansicht", "Lesezeichen", "Extras", "Einstellungen", and "Hilfe". The toolbar contains various icons for file operations and editing. The code editor displays the following PHP code:

```
1 <?php
2
3 .   ... // generation of the result
4
5 .   foreach($result['resultElements'] as $num => $resultElement) {
6 .     print("<a href=\"\" . $resultElement['URL'] . \">");
7 .     print($resultElement['title']);
8 .     print("</a><br />");
9 .   }
10
11 .   ... // further statements
12
13 ?>
14
```

The status bar at the bottom indicates "Zeile: 6 Spalte: 35", "Einf", "NORM", and "evaluate.php".



# ... am Beispiel Google (doGoogleSearch)

Der Demo-Client ...

<http://www.fabrizio-branca.de/webservices/google/client.php>

# Erstellen von Web Services

- Allgemein
  - Welche Operationen sollen bereitgestellt werden?
  - Welche Parameter benötigen die Operationen?
- NuSOAP
  - Wie verarbeite ich Anfragen?
  - Wie veröffentliche ich den Webservice?

# ... am Beispiel eines Adressbuchs

Welche Operationen sollen bereitgestellt werden?

- Abfragen aller Adressen
- Hinzufügen einer Adresse
- Löschen einer Adresse
- Suchen von Adressen
- Sonstige Operationen

# ... am Beispiel eines Adressbuchs (search)


Welche Parameter benötigen die Funktionen?

- Eingabe-Parameter (input)
  - Zeichenkette `query`, nach der gesucht werden soll
- Rückgabe-Wert (output)
  - Liste aller Adress-Datensätze (Struktur aus `name`, `phone`, `email`), die auf die Suchanfrage passen

# ... am Beispiel eines Adressbuchs (search)

Wie verarbeite ich Anfragen?

- Hinzufügen der Datentypen
- Programmieren der Funktion
- Registrieren der Operation



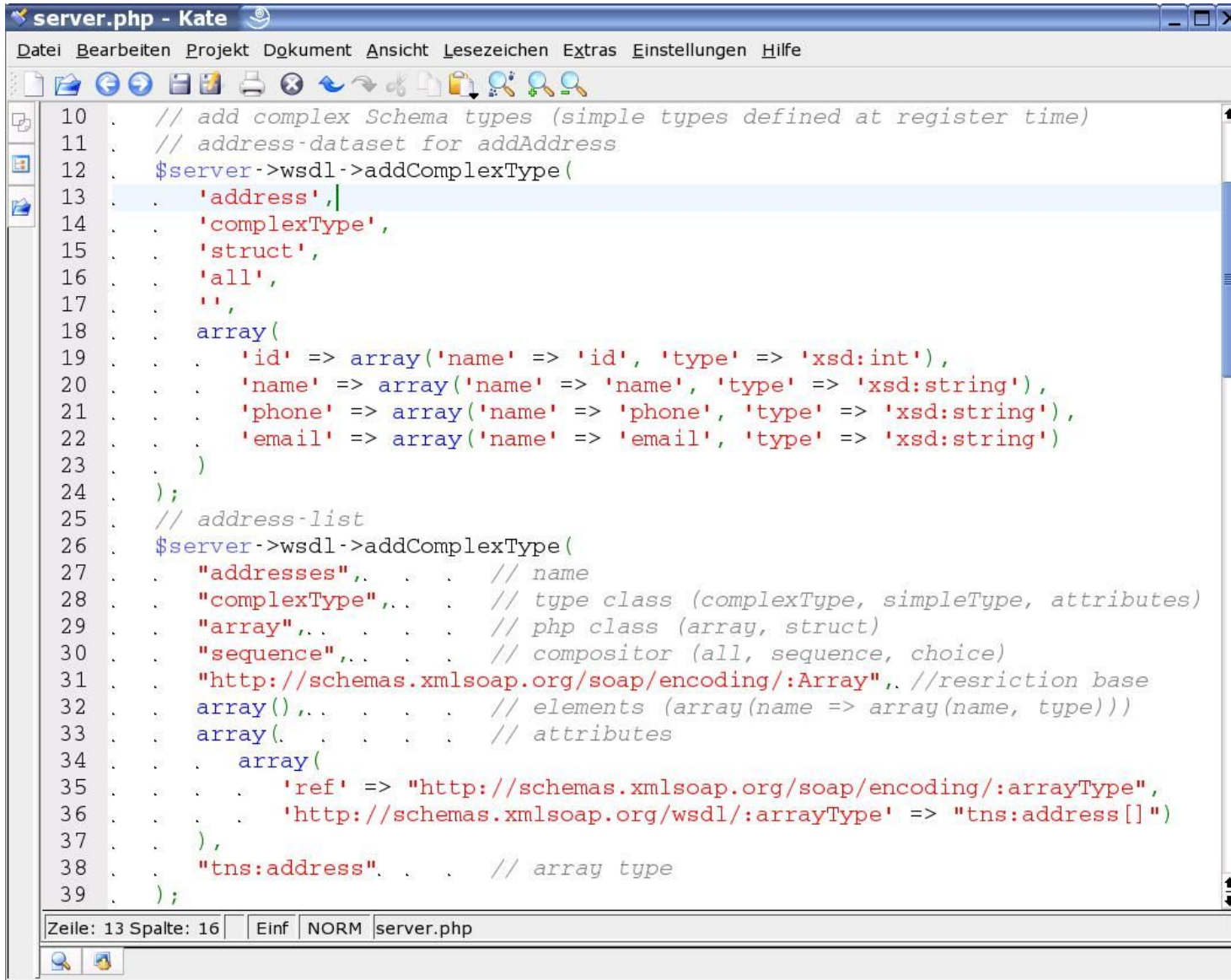
The screenshot shows a text editor window titled "server.php [Geändert] - Kate". The menu bar includes "Datei", "Bearbeiten", "Projekt", "Dokument", "Ansicht", "Lesezeichen", "Extras", "Einstellungen", and "Hilfe". The toolbar contains various icons for file operations and editing. The main text area displays the following PHP code:

```
2  
3 .   ... // preceding statements  
4  
5 .   // server setup  
6 .   // creating the server  
7 .   $server = new soap_server();  
8 .   // enable use of WSDL  
9 .   $server->configureWSDL('directory', 'http://webservices.thadude/directory/XMLSchema');  
10 .  $server->wsdl->schemaTargetNamespace = 'http://webservices.thadude/directory/XMLSchema';  
11 .  
12 .  ... // following statements  
13 .
```

The status bar at the bottom indicates "Zeile: 4 Spalte: 0", "Einf", "NORM", and "server.php".

# ... am Beispiel eines Adressbuchs (search)

Hinzufügen der Datentypen



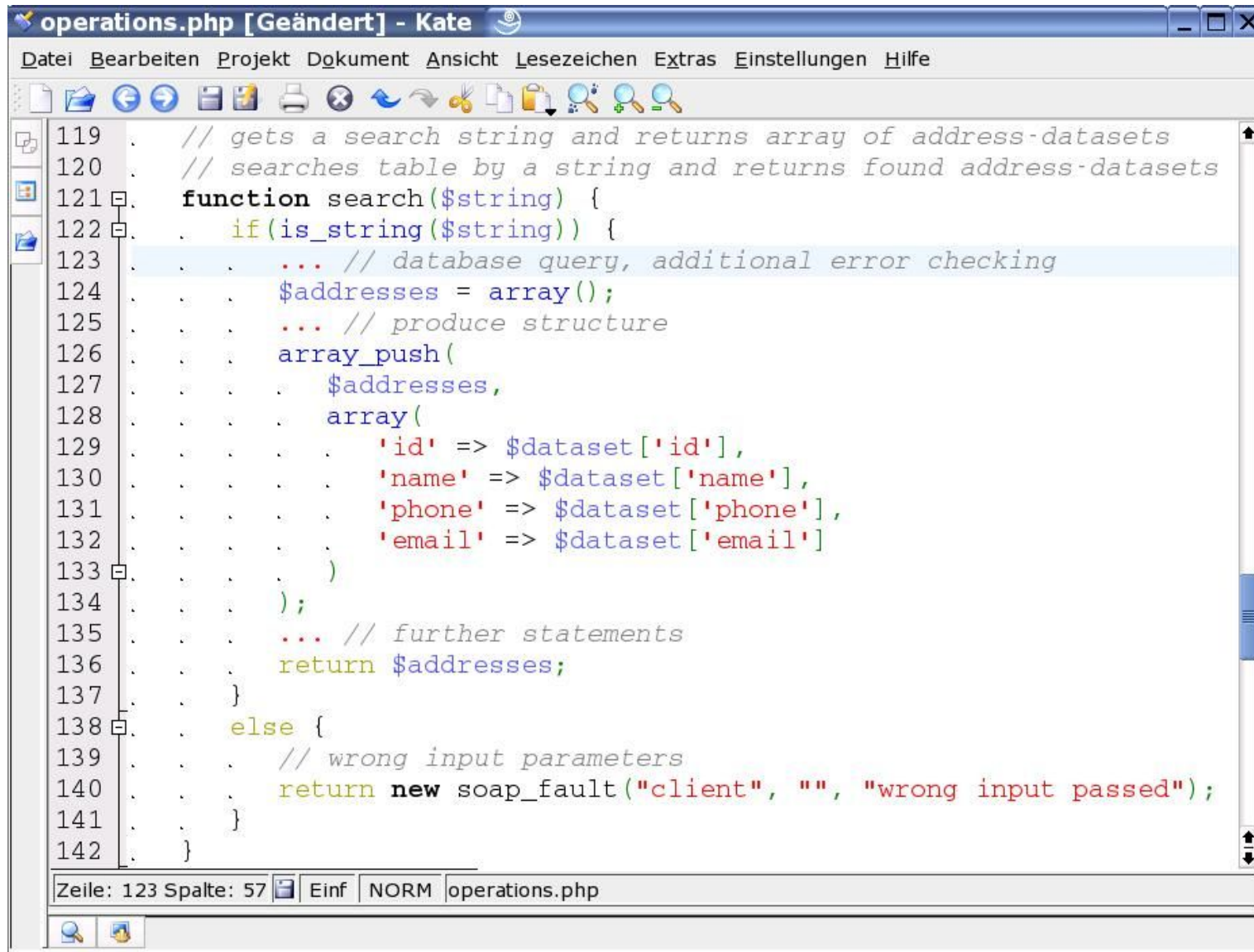
```
server.php - Kate
Datei Bearbeiten Projekt Dokument Ansicht Lesezeichen Extras Einstellungen Hilfe

10 . // add complex Schema types (simple types defined at register time)
11 . // address-dataset for addAddress
12 . $server->wsdl->addComplexType(
13 .     'address',|
14 .     'complexType',
15 .     'struct',
16 .     'all',
17 .     '',
18 .     array(
19 .         'id' => array('name' => 'id', 'type' => 'xsd:int'),
20 .         'name' => array('name' => 'name', 'type' => 'xsd:string'),
21 .         'phone' => array('name' => 'phone', 'type' => 'xsd:string'),
22 .         'email' => array('name' => 'email', 'type' => 'xsd:string')
23 .     )
24 . );
25 . // address-list
26 . $server->wsdl->addComplexType(
27 .     "addresses".. . . // name
28 .     "complexType".. . // type class (complexType, simpleType, attributes)
29 .     "array".. . . . // php class (array, struct)
30 .     "sequence".. . . // compositor (all, sequence, choice)
31 .     "http://schemas.xmlsoap.org/soap/encoding:Array".. //restriction base
32 .     array().. . . . // elements (array(name => array(name, type)))
33 .     array(.. . . . // attributes
34 .         array(
35 .             'ref' => "http://schemas.xmlsoap.org/soap/encoding:arrayType",
36 .             'http://schemas.xmlsoap.org/wsdl:ArrayType' => "tns:address[]"
37 .         ),
38 .         "tns:address".. . // array type
39 .     );

Zeile: 13 Spalte: 16 Einf NORM server.php
```

# ... am Beispiel eines Adressbuchs (search)

Programmieren der Funktion



The screenshot shows a code editor window titled "operations.php [Geändert] - Kate". The window contains PHP code for a search function. The code is as follows:

```
119 . // gets a search string and returns array of address-datasets
120 . // searches table by a string and returns found address-datasets
121 . function search($string) {
122 .     if(is_string($string)) {
123 .         ... // database query, additional error checking
124 .         $addresses = array();
125 .         ... // produce structure
126 .         array_push(
127 .             $addresses,
128 .             array(
129 .                 'id' => $dataset['id'],
130 .                 'name' => $dataset['name'],
131 .                 'phone' => $dataset['phone'],
132 .                 'email' => $dataset['email']
133 .             )
134 .         );
135 .         ... // further statements
136 .         return $addresses;
137 .     }
138 .     else {
139 .         // wrong input parameters
140 .         return new soap_fault("client", "", "wrong input passed");
141 .     }
142 . }
```

The status bar at the bottom of the editor shows "Zeile: 123 Spalte: 57", "Einf", "NORM", and "operations.php".





# ... am Beispiel eines Adressbuchs

Wie veröffentliche ich den Web Service?



```
5 .  
6 .   ... // preceding statements  
7 .  
8 .   // passing the soap request to the server  
9 .   $HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ? $HTTP_RAW_POST_DATA : "";  
10 .  $server->service($HTTP_RAW_POST_DATA);  
11 .  
12 .   ... // following statements  
13 .
```

Zeile: 10 Spalte: 41 Einf NORM service.php

# ... am Beispiel eines Adressbuchs

Der Demo-Service ...

<http://www.fabrizio-branca.de/webservices/directory/service.php>

Client: <http://www.fabrizio-branca.de/webservices/directory/client.php>

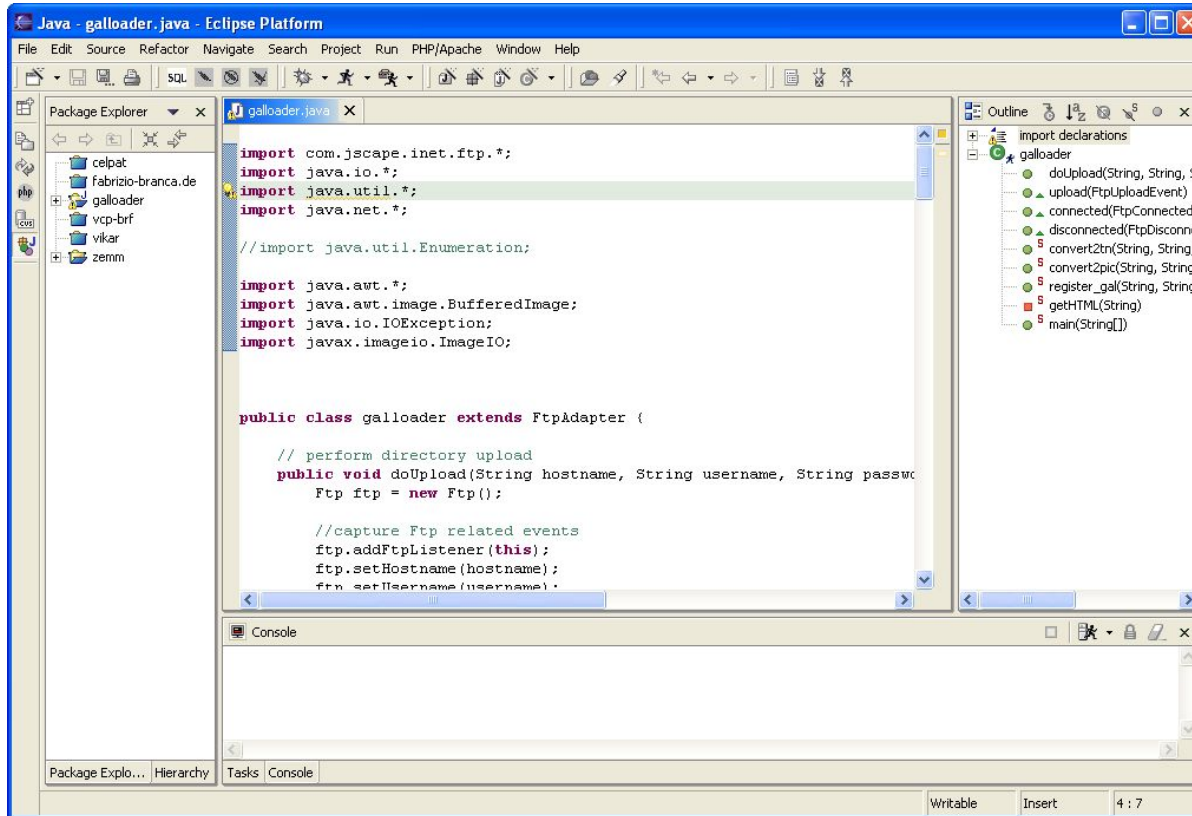
Web Services mit...

# Java



# Eclipse

»Eclipse is a kind of universal tool platform - an open extensible IDE for anything and nothing in particular.«



- Download [Eclipse]
- Aktuelle Version 3.0
- Java-based IDE
- Zahlreiche sehr gute Plugins:  
**PHP-Eclipse** [PHPEclipse]  
**SFTP** [KlompSFTP]  
**SQL** [EclipseSQL]  
uvm.,...

## Funktionen:

- syntax highlighting editor
- code completion
- source-level debugger
- class navigator
- file/project manager
- task-orientierte Entwicklung durch Perspektiven
- CVS-Unterstützung

# Apache Tomcat

## Servlet /JSP Container



- Aktuelle Version: 5.1
- Download [Tomcat]
- Einfache Installation
- Tomcat starten
  - Manuell (Skripte werden mitinstalliert)
  - Unter Win NT/2000/XP: als Service
  - Aus Eclipse-Plugin heraus [TomcatPlugin]

# Screenshot: Tomcat-Startseite



The screenshot shows a Mozilla Firefox browser window titled "Jakarta Project - Tomcat". The address bar displays "http://localhost:8080/index.html". The page content includes the Tomcat logo (a yellow cat) and the text "Tomcat Version 4.0.4". To the right is the "The Jakarta Project" logo (a colorful feather) and the URL "http://jakarta.apache.org". The main content area features a congratulatory message: "If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!". Below this, it explains that the page is the default Tomcat home page and provides the local file path: `$CATALINA_HOME/webapps/ROOT/index.html`. It also offers links to "Web Applications" (JSP Examples, Servlet Examples, WebDAV capabilities), "Documentation" (Tomcat Documentation), and "Miscellaneous" (Sun's Java Server Pages Site, Sun's Servlet Site). A paragraph explains that the page is the default Tomcat home page and provides the local file path: `$CATALINA_HOME/webapps/ROOT/index.html`. It also offers links to "Web Applications" (JSP Examples, Servlet Examples, WebDAV capabilities), "Documentation" (Tomcat Documentation), and "Miscellaneous" (Sun's Java Server Pages Site, Sun's Servlet Site). A paragraph explains that the page is the default Tomcat home page and provides the local file path: `$CATALINA_HOME/webapps/ROOT/index.html`. It also offers links to "Web Applications" (JSP Examples, Servlet Examples, WebDAV capabilities), "Documentation" (Tomcat Documentation), and "Miscellaneous" (Sun's Java Server Pages Site, Sun's Servlet Site). The browser's status bar shows "Done" and the Windows taskbar at the bottom includes the Start button, "Start Tomcat", and "Jakarta Project - Tom..." with a system clock showing 20:07.

Tomcat  
Version 4.0.4

The Jakarta Project  
<http://jakarta.apache.org>

**Web Applications**  
[JSP Examples](#)  
[Servlet Examples](#)  
[WebDAV capabilities](#)

**Documentation**  
[Tomcat Documentation](#)

**Miscellaneous**  
[Sun's Java Server Pages Site](#)  
[Sun's Servlet Site](#)

**If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!**

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

`$CATALINA_HOME/webapps/ROOT/index.html`

where "\$CATALINA\_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet

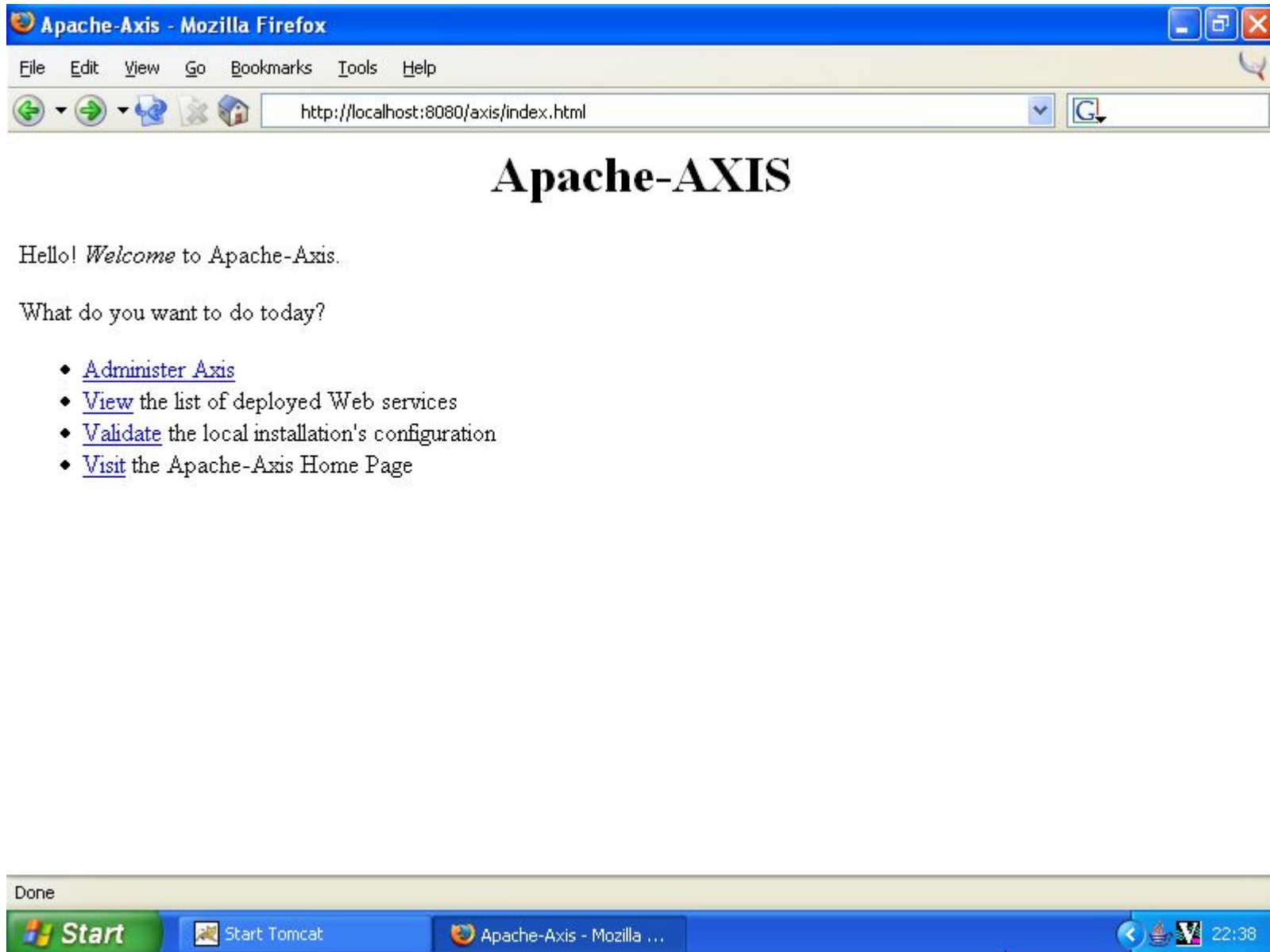
# Apache Axis

## Web Services Container



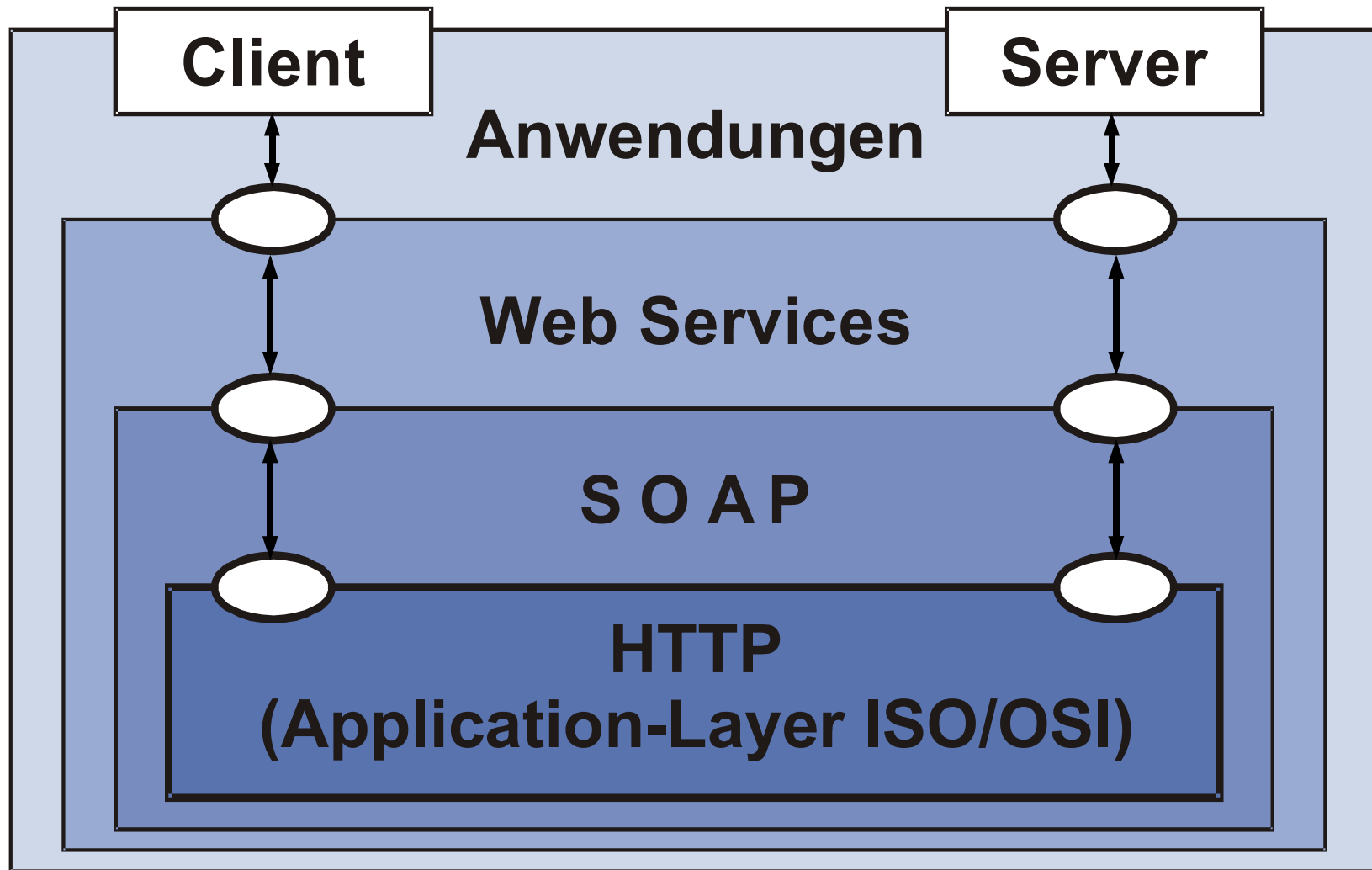
- Aktuelle Version: 1.1 (vorher: „Apache SOAP“)
- Download [ApacheAxis]
- Installation: Wird in Tomcat eingebunden (als Servlet)
- Aufgaben eines Web Service - Containers:
  - Verwaltet das Routen und Empfangen von SOAP-Nachrichten und
  - mapped empfangene SOAP-Nachrichten an die jeweiligen Web Services (Service Listener / Service Router)
  - bietet Tools um Web Services zu erstellen und zu deployen
    - Java2WSDL
    - WSDL2Java
    - AdminClient (deploy, undeploy)

# Screenshot: Axis-Startseite

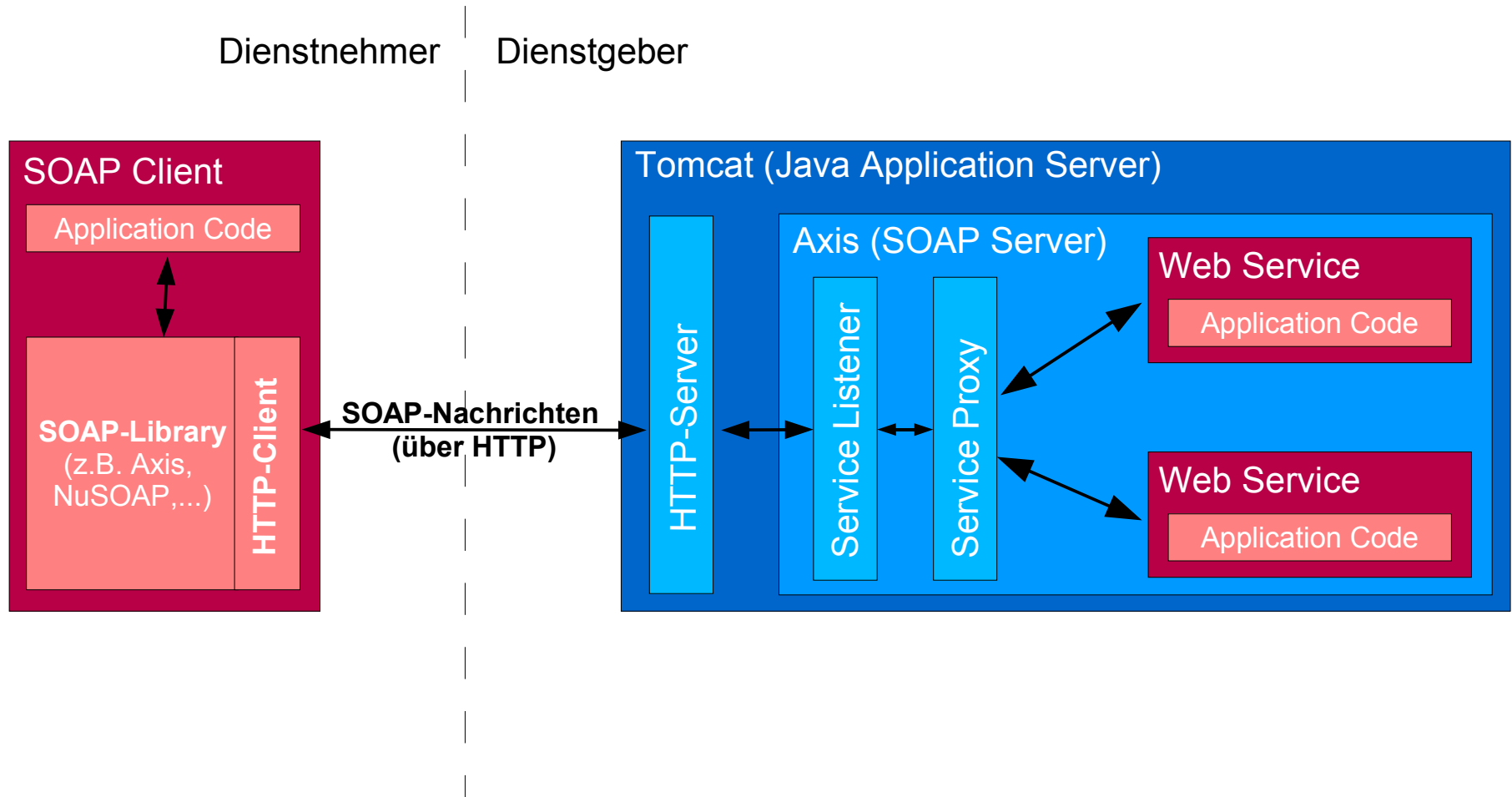




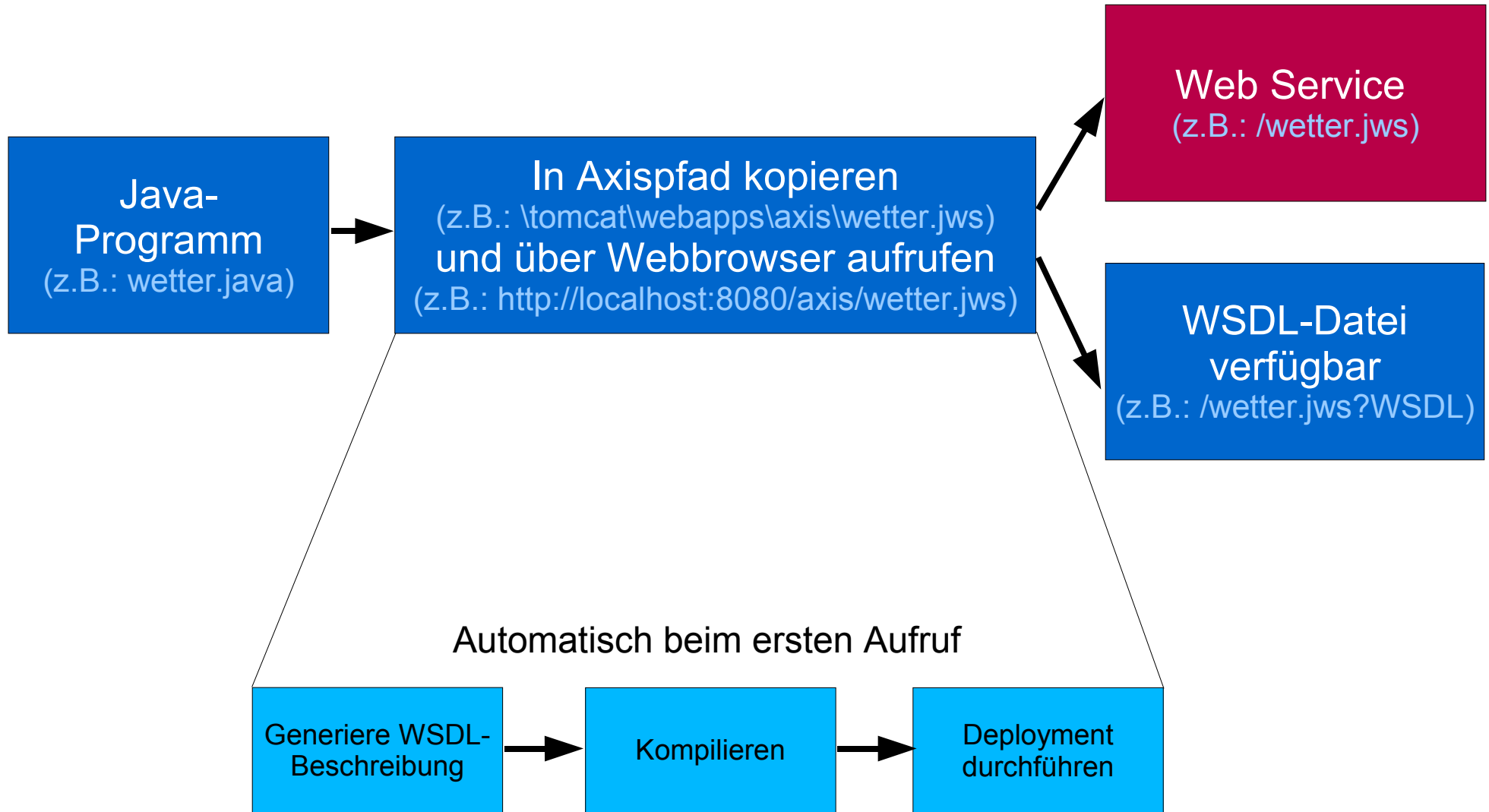
# Schichtenmodell



# Zusammenhang der Komponenten



# Server erstellen: Instant Deployment



# Server erstellen: Ausführlich

## 1. Erzeugung der Web-Service-Schnittstelle

Wetter.java

WetterImpl.java

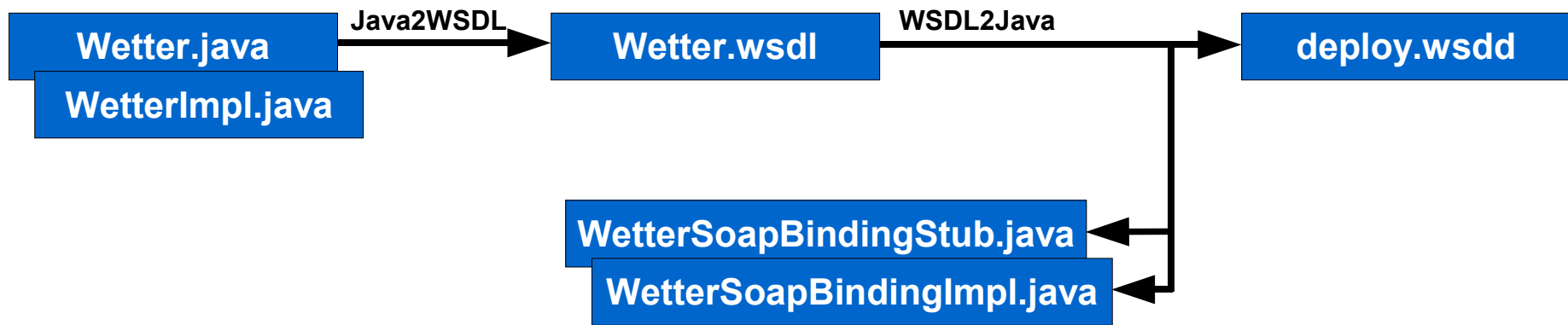
# Server erstellen: Ausführlich

## 2. Erzeugung eines WSDL-Dokuments



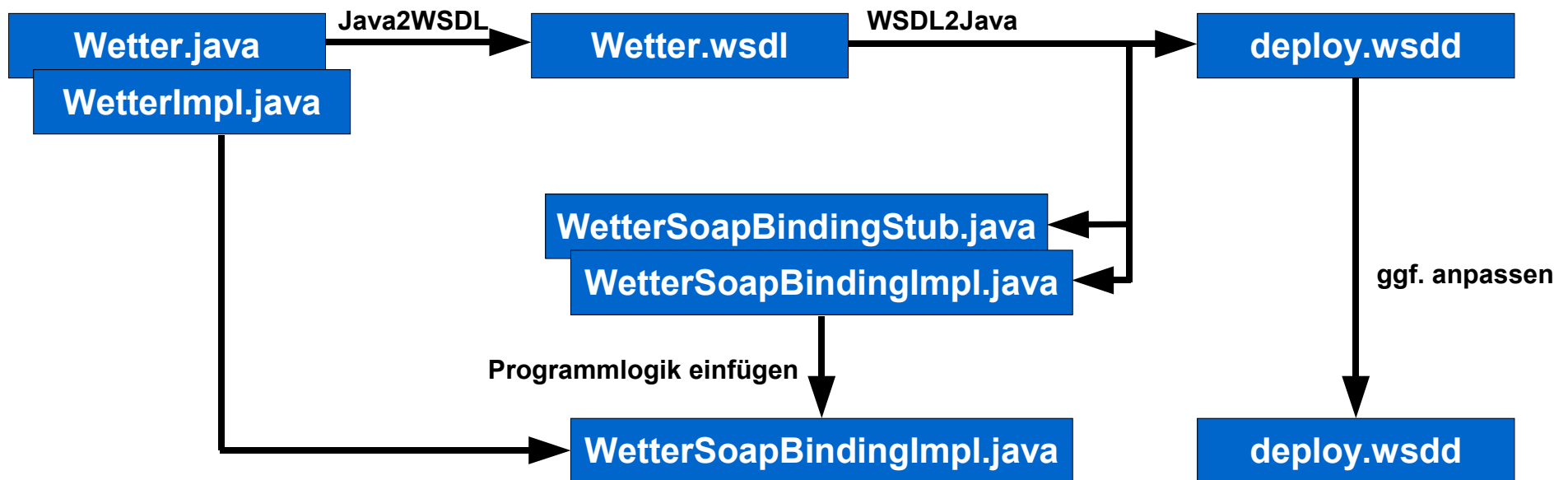
# Server erstellen: Ausführlich

## 3. Java-Dateien vorbereiten



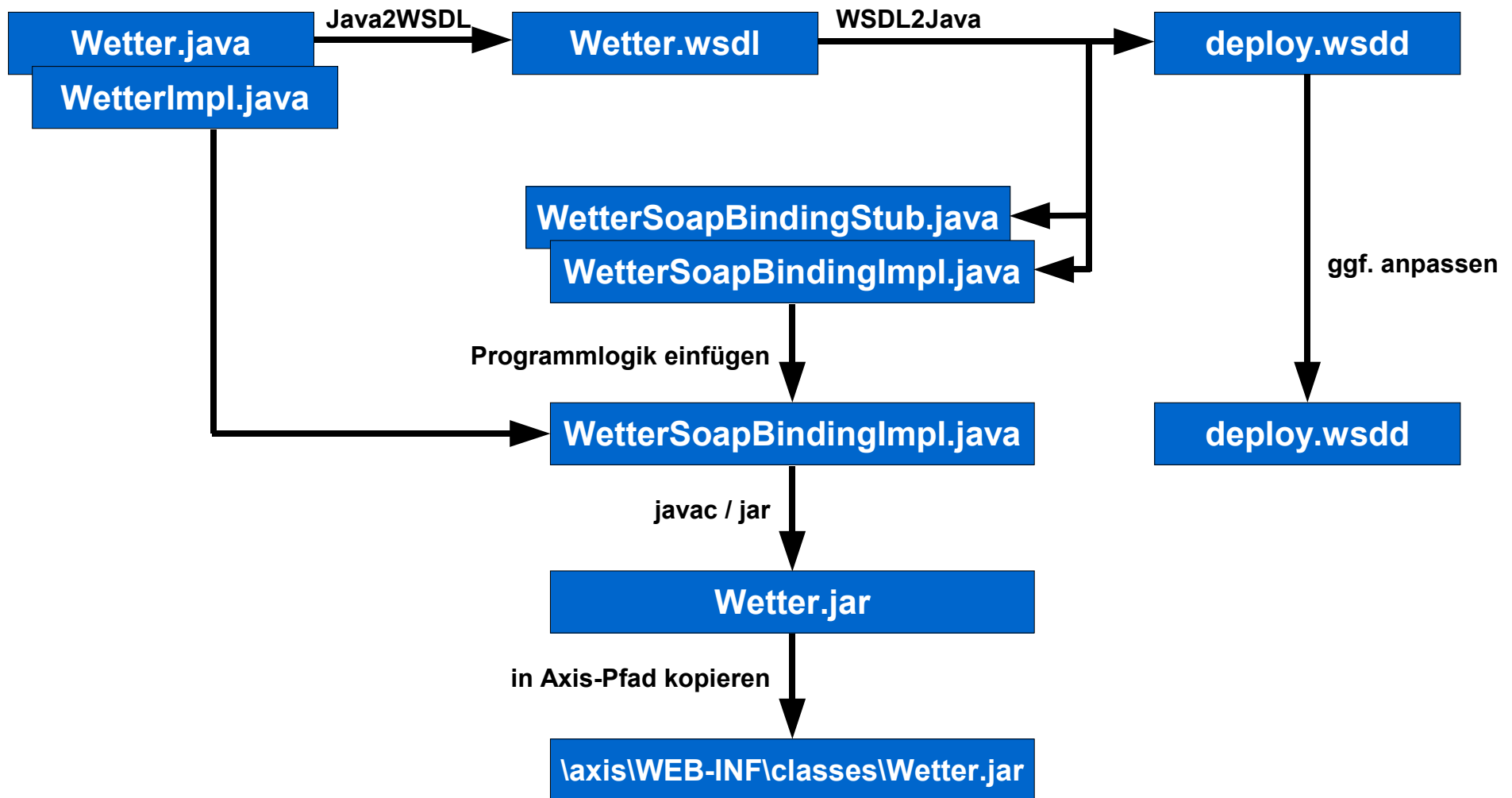
# Server erstellen: Ausführlich

## 4. Programmlogik einfügen, WSDO anpassen



# Server erstellen: Ausführlich

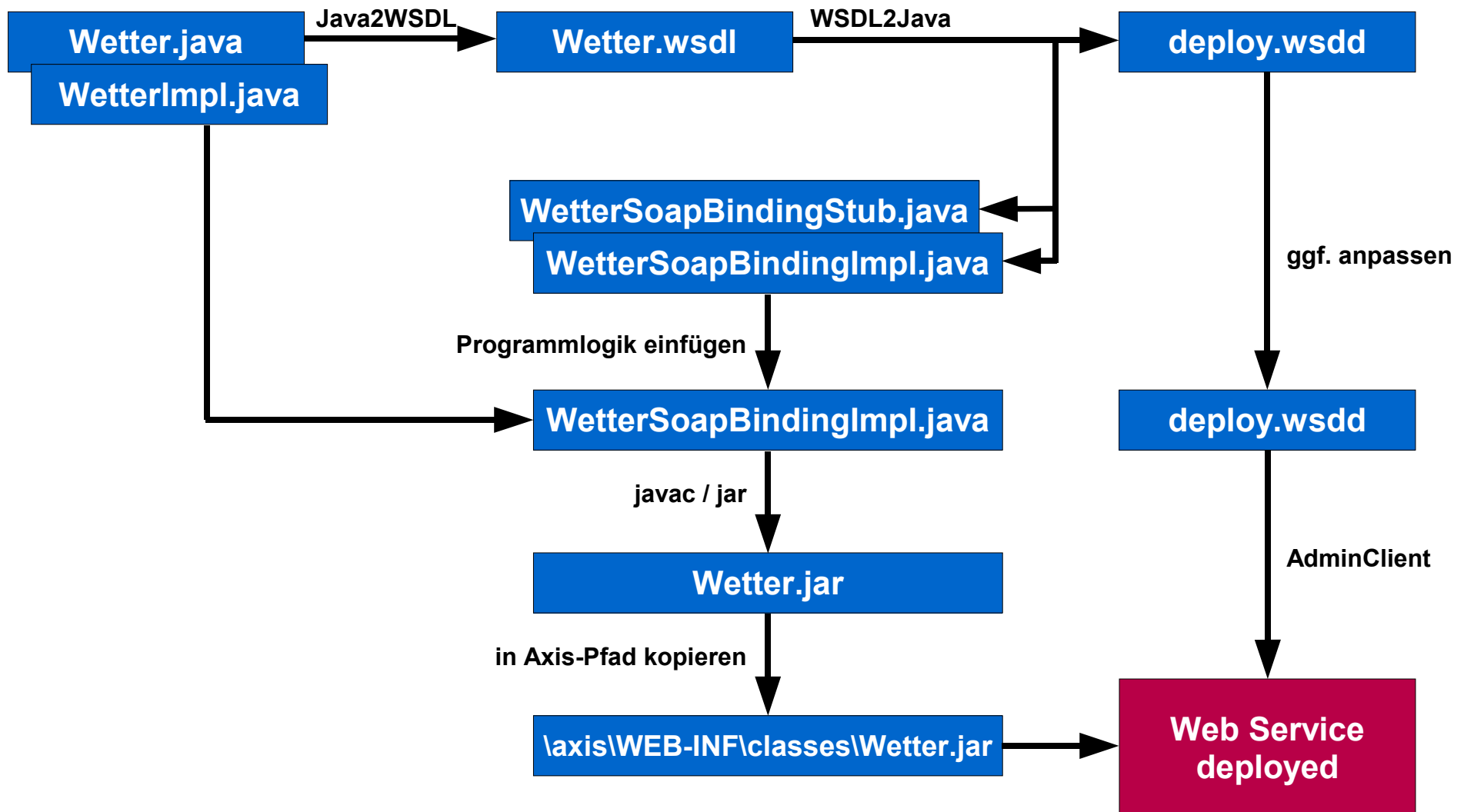
## 5. Kompilieren, Package, Kopieren





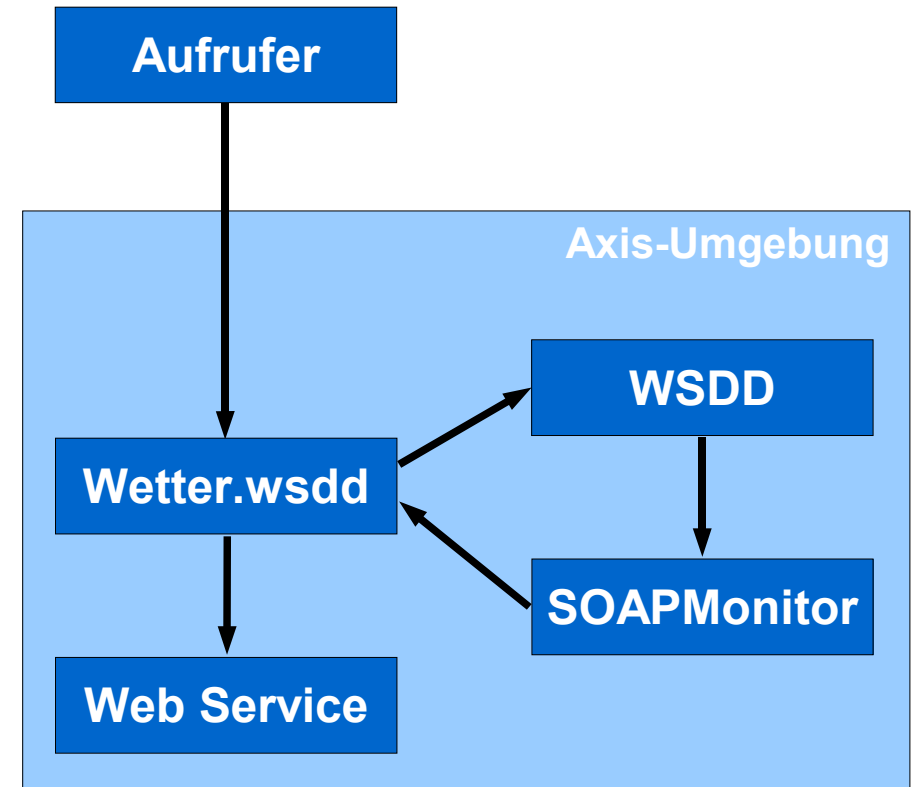
# Server erstellen: Ausführlich

## 6. Deployen



# Anpassen der WSDD-Datei / SOAPMonitor

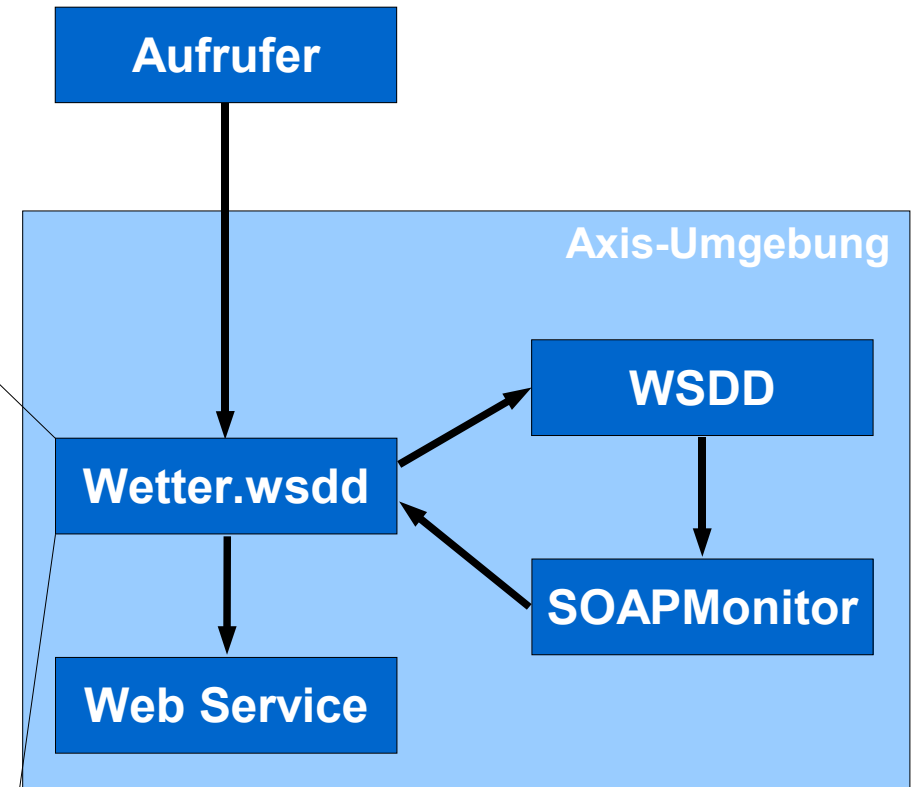
- SOAPMonitor:
  - Im Axis-Paket enthalten
  - zum Beobachten von SOAP Request und Response Messages
  - hilfreich beim Debuggen von Web Services



# Anpassen der WSDD-Datei / SOAPMonitor

- SOAPMonitor:
  - Im Axis-Paket enthalten
  - zum Beobachten von SOAP Request und Response Messages
  - hilfreich beim Debuggen von Web Services

```
Deployment Descriptor  
[...]  
<deployment  
  xmlns="http://xml.apache.org/axis/wsdd/"  
  xmlns:java="http://xml.apache.org/axis/wsdd/  
providers/java">  
  
  <service name="wetter" provider="java:RPC">  
  
    <requestFlow>  
      <handler type="soapmonitor"/>  
    </requestFlow>  
    <responseFlow>  
      <handler type="soapmonitor"/>  
    </responseFlow>  
  
  </service>  
</deployment>  
[...]
```



# Screenshot: SOAPMonitor-Applet

The screenshot displays the SOAPMonitor-Applet interface. At the top, a tab labeled "localhost" is active. Below it is a table with three columns: "Time", "Target Service", and "Status". The table contains one entry: "11:21:45 PM", "PersonLookup", and "Complete". Below the table are three buttons: "Remove", "Remove All", and "Filter ...".

Below the buttons, a header bar shows "Time: 11:21:45 PM", "Target Servic...PersonLookup", and "Status: Complete". The main area is split into two panes: "SOAP Request" and "SOAP Response".

**SOAP Request:**

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:getPerson soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <in0 xsi:type="xsd:int">4</in0>
    </ns1:getPerson>
  </soapenv:Body>
</soapenv:Envelope>
```

**SOAP Response:**

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:getPersonResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <getPersonReturn href="#id0"/>
    </ns1:getPersonResponse>
    <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <FName xsi:type="xsd:string">Evie</FName>
      <index xsi:type="xsd:int">4</index>
      <LName xsi:type="xsd:string">Evans</LName>
      <born xsi:type="xsd:dateTime">1959-02-27T09:00:00</born>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

At the bottom of the applet, there is a checkbox labeled "Reflow XML text" and a button labeled "Switch Layout". Below these are two buttons: "Start" and "Stop", and a status message: "The SOAP Monitor is started."

# Automatisieren: Batch-Dateien

```
call config.bat %1 %2 %3

:: Verzeichnis fuer kompilierte Urklassen
des Webservices anlegen
mkdir %classesDir%

:: Kompilieren der Urklassen des
Webservices in das Unterverzeichnis classes
javac -classpath %axisCP%;%apacheCP%
-sourcepath %input% -d %classesDir% %input%
\%wsDir%\*.java

:: Verzeichnis fuer WSDL Dateien anlegen
mkdir %wsdlDir%

:: Erstellen der WSDL-Datei aus der
Ursprungsdatei des Webservices (*.java)
java -cp %axisCP%;%apacheCP%;%classesDir%
org.apache.axis.wsdl.Java2WSDL --output %
wsdlDir%\%wsdlFile% --location %location%
-n urn:%namespace% %wsDir%.%wsClass%

:: Verzeichnis fuer die von WSDL2java
produzierten Daten erstellen (server)
mkdir %serverDir%

:: Server-Klassen aus der WSDL-Datei
generieren
java -classpath %axisCP%;%apacheCP%
org.apache.axis.wsdl.WSDL2Java --verbose -o
%serverDir% -d Session -s -p %wsDir%.ws %
wsdlDir%\%wsdlFile%
```

Dateien  
anpassen →

```
call config.bat %1 %2 %3

:: Verzeichnis des Webservices unter AXIS
in WEB-INF\classes\%wsDir%\ws anlegen
mkdir %axisRoot%\WEB-INF\classes\%wsDir%\ws

:: modifizierten WSDL2Java Dateien-Satz in
dieses Verzeichnis kopieren
xcopy %serverDir%\%wsDir%\ws %axisRoot%
\WEB-INF\classes\%wsDir%\ws

:: Ursprungs-Implementation des Webservices
an entsprechenden Ort unter AXIS kopieren
xcopy %input%\%wsDir%\%wsClass%Impl.java %
axisRoot%\WEB-INF\classes\%wsDir%

:: Webservice (*.java-Dateien) in AXIS
kompilieren
javac -classpath %axisCP%;%apacheCP%
-sourcepath %axisRoot%\WEB-INF\classes -d %
axisRoot%\WEB-INF\classes %axisRoot%\WEB-
INF\classes\%wsDir%\ws\*.java

:: Deploy der wsdd-Datei %deployFile%
java -classpath %axisCP%;%apacheCP%
org.apache.axis.client.AdminClient -l%
axisURL%/services/AdminService %deployFile%
```

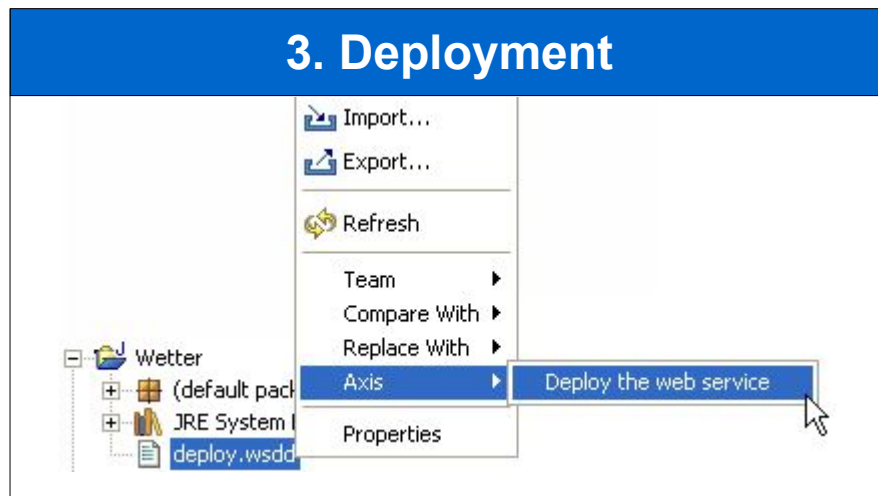
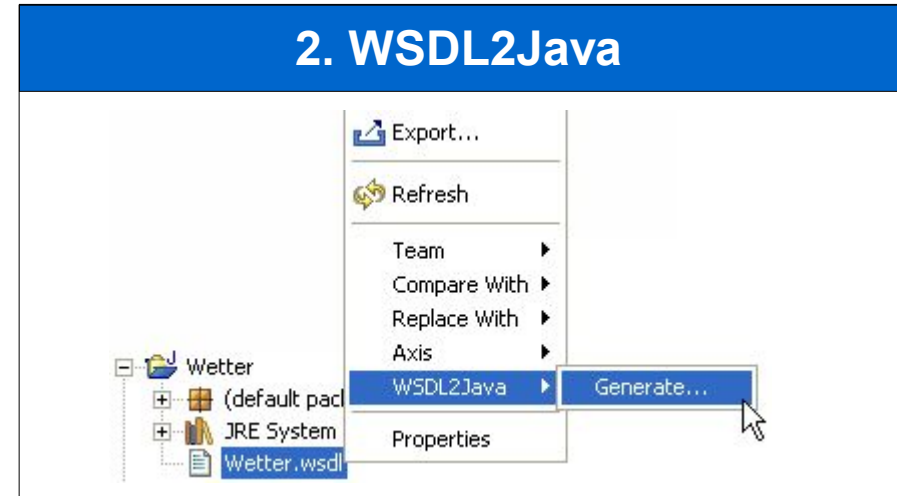
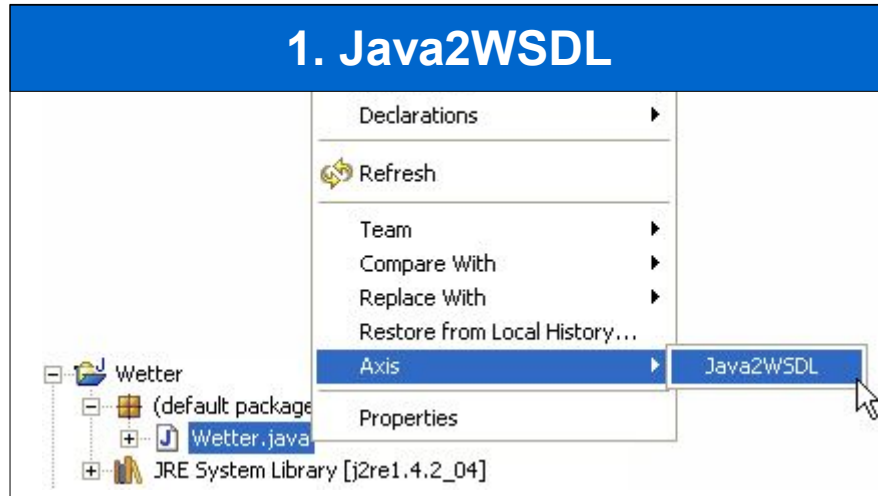
# Automatisieren: Apache Ant [Ant]

```
- <target name="prepare" description="Creates the build directory">
  <echo message="Creating the required directories...."/>
  <mkdir dir="${build}"/>
</target>
- <target name="compile" depends="prepare" description="Compiles the source code created by wsdl2java">
  <echo message="Compiling the source code created by wsdl2java...."/>
  - <javac srcdir="${src}" destdir="${build}" includes="*.java">
    <classpath refid="compile.classpath"/>
  </javac>
</target>
- <target name="wsdl2java" description="Building stubs, skeletons, and data types from WSDL">
  - <java classname="${axis.wsdl2java.class}" fork="yes">
    <arg line="--server-side ${service.wsdl.file}"/>
    <classpath refid="compile.classpath"/>
  </java>
</target>
- <target name="run-client" description="Calls UniVerwaltungService">
  - <java classname="${client.class}" fork="yes">
    <arg line="${Call.ProfName}"/>
    <classpath refid="run.classpath"/>
  </java>
</target>
- <target name="clean" description="Removes the build directory">
  <delete dir="${build}"/>
</target>
</project>
```



- Alternative zu Make
- XML-Dokument
- Kann von Eclipse aus ausgeführt werden
- build.xml in Projektverzeichnis
- Aufruf:  
ant wsdl2java

# Automatisieren: Eclipse Plugins



**(1) und (3)**  
**Improve Axis Plugin for Eclipse**  
Download: [Improve]

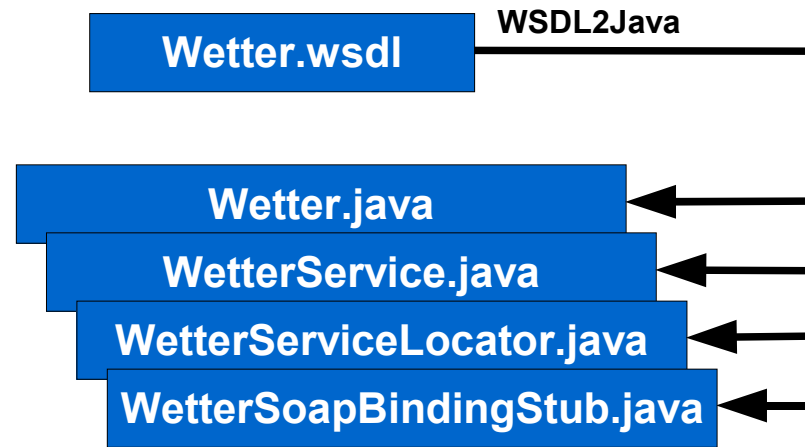
**(2)**  
**WSDL2Java Eclipse Plug-in**  
Download: [MySpotter]

# Client

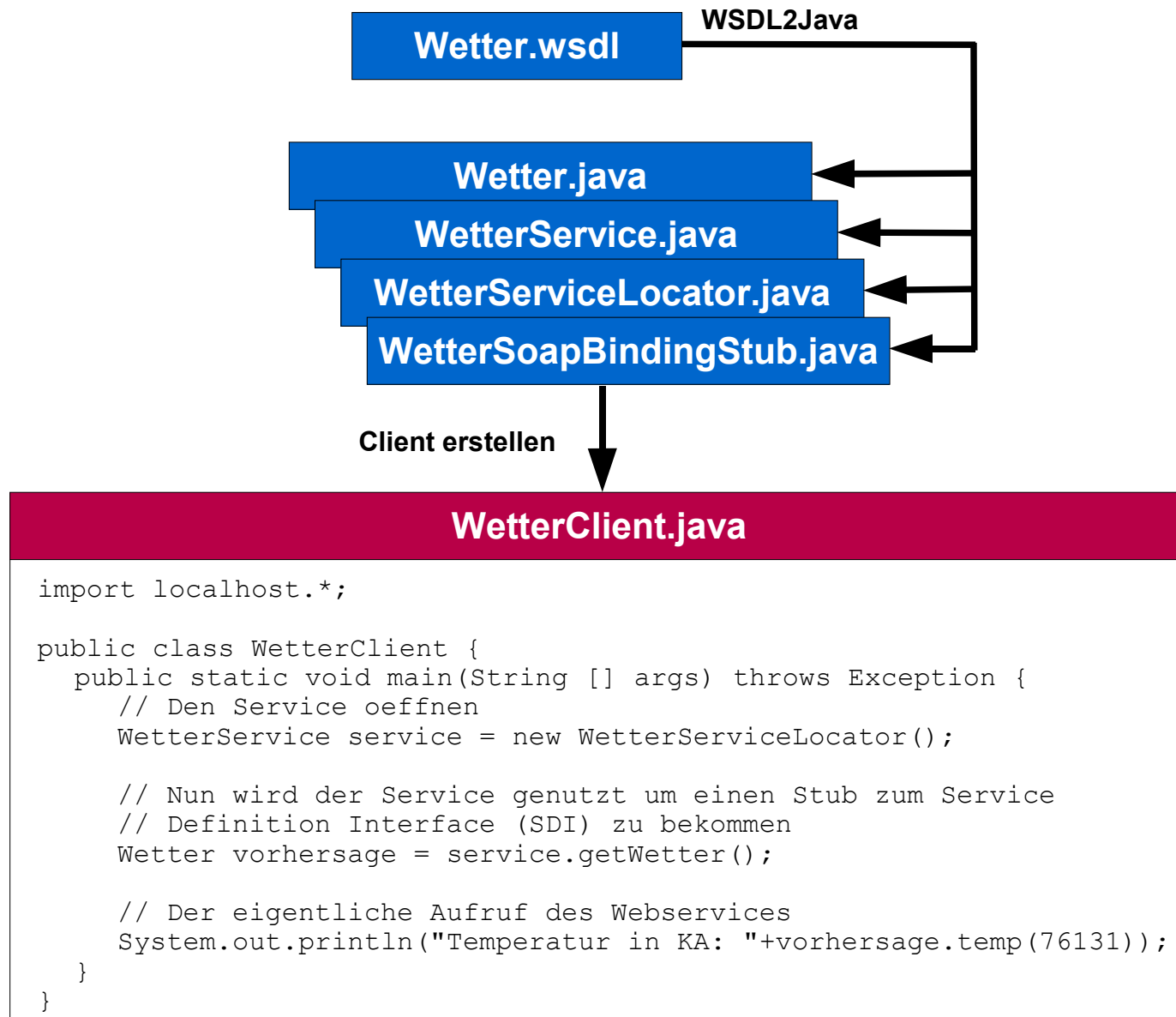
Wetter.wsdl



# Client



# Client



# Links

[Tomcat] **Tomcat Application Server**

<http://jakarta.apache.org/tomcat/>

[TomcatPlugin] **Sysdeo Eclipse Tomcat Launcher plugin**

<http://www.sysdeo.com/eclipse/tomcatPlugin.html>

[ApacheAxis] **Apache Axis**

<http://ws.apache.org/axis/>

[Ant] **Apache Ant**

<http://ant.apache.org/>

[Improve] **Improve Axis Plugin for Eclipse**

<http://www.improve-technologies.com/alpha/axis/>

[MySpotter] **WSDL2Java Eclipse Plug-in**

<http://www.myspotter.com/wsdl2java.shtml>

[Eclipse] **Eclipse**

<http://www.eclipse.org/>

[KlompSFTP] **Sftp File Synchronization Plugin**

<http://klomp.org/eclipse/org.klomp.eclipse.team.sftp/>

[PHPEclipse] **PHPEclipse**

<http://www.phpeclipse.de/>

[PHP] **PHP**

<http://www.php.net>

[NuSOAP] **NuSOAP**

<http://dietch.ganx4.com/nusoap/>