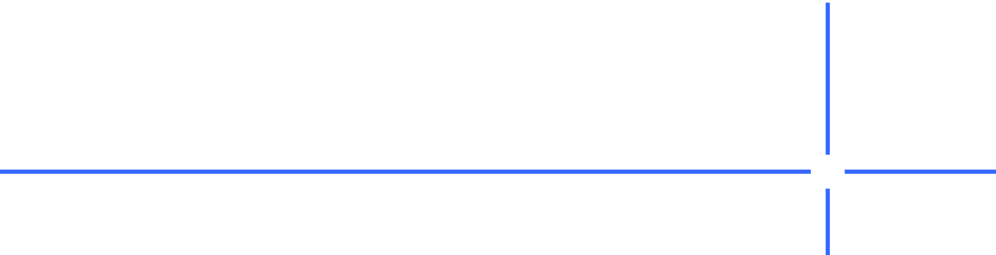


# Tutorial: Web Services

Was ist das?

Wozu sind Web Services nützlich?

Wie implementiere ich eigene Web Services



*„Bei Web Services handelt es sich um Softwarebausteine, die auf verschiedenen Netzwerkrechnern laufen und über das Internet zu einer Anwendung verbunden werden.“*

*Profi4Project.com, 19.07.2002*



# Was sind Web Services?

- Ist jetzt jeder Web Browser und jeder Web Server ein Web Service?
- Nein, denn die Idee ist es, dass die Programme eines Web Services über ein XML-gekapseltes Protokoll kommunizieren.



# Protokolle

- Time Services: ntp
- User Verwaltung: yp-services
- File Transfer: ftp
- ...
- jeder Dienst hat ein eigenes Protokoll...

# Protokolle

<b>Layer</b>	<b>Protokoll/Standard</b>
Messaging	HTML, MVOP, ...
Transport	HTTP, SMTP, FTP, NTP, YP
Network	TCP/IP, UDP

# XML-gekapselte Protokolle

<b>Layer</b>	<b>Protokoll/Standard</b>
Content	Content Information
Messaging	SOAP, XML-RPC
Transport	HTTP, SMTP, FTP
Network	TCP/IP, UDP

Die Protokolle und Standards aller Layer nur in Auswahl.

# Web Service Technology Stack

<b>Layer</b>	<b>Protokoll/Standard</b>
Discovery	UDDI, DISCO, WSIL
Description	WSDL, RDF
Messaging	SOAP, XML-RPC
Transport	HTTP, SMTP, FTP
Network	TCP/IP, UDP

Die Protokolle und Standards aller Layer nur in Auswahl.



# Hintergrund und Motivation

- Die Integration wird mit Web Services nicht neu erfunden.
  - Client-Server Architektur ist alt bekannt.
  - Durch standardisierte Schnittstellen und Protokolle wird die Technik entschieden einfacher.
  - Damit ist eine Konzentration auf die Modellierung und das Management möglich.
- ...also, lasst uns aufbrechen, von monolithischen Client-Server-Architekturen zu komponentenbasierten Web Services ...*





# Vorläufer

- CORBA (Component Object Request Broker Architecture)
- DCOM (Distributed Component Object Model)
- Enterprise Java Beans (in J2EE)



## Blick in die „reale Welt“

- .Net My Services
- .Net Passport
- Google
- Amazon
- VASCODA
- ...
- Warum gibt es nur so wenige wirkliche Implementationen?

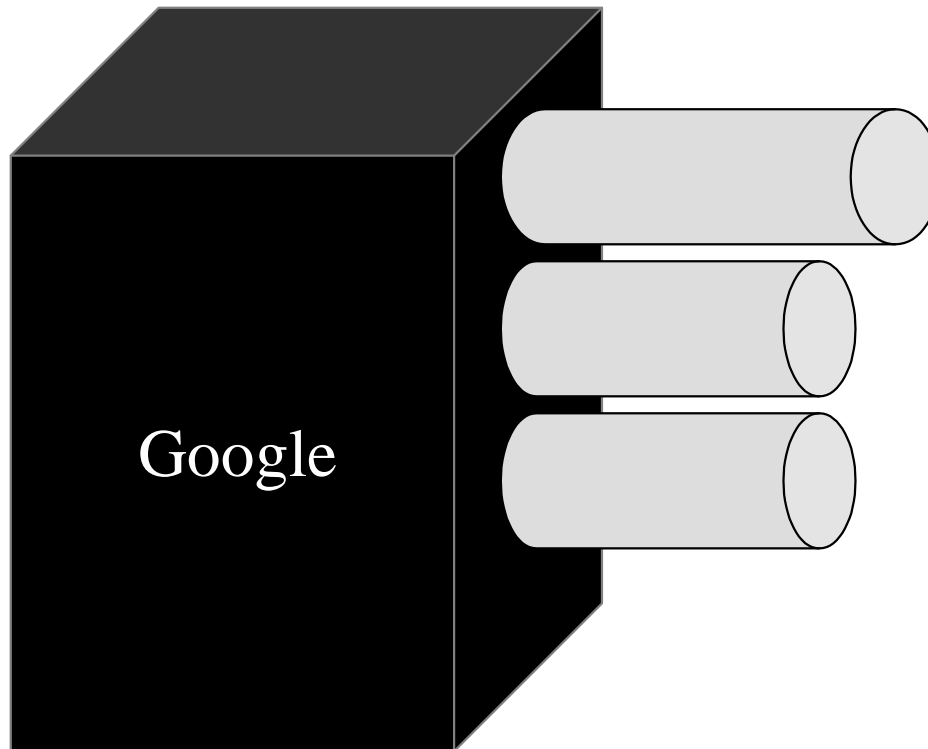


## Blick in die „reale Welt“

- .Net My Services
- .Net Passport
- Google
- Amazon
- VASCODA
- [www.xmethods.org](http://www.xmethods.org)
- Warum gibt es nur so wenige wirkliche Implementationen?

Viele Beispiele, die meisten haben aber eher einen beschränkten Funktionsumfang und sind im wesentlichen nicht mehr ein als Proof-of-Concept.

# Beispiel: Google

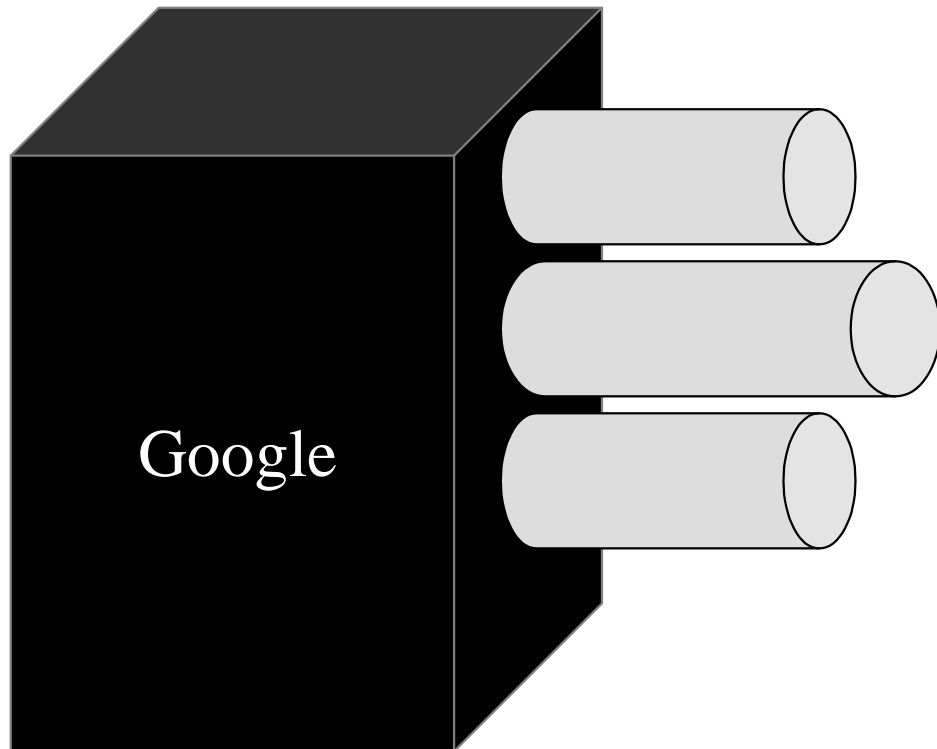


[www.google.com/apis/](http://www.google.com/apis/)



---

# Beispiel: Google



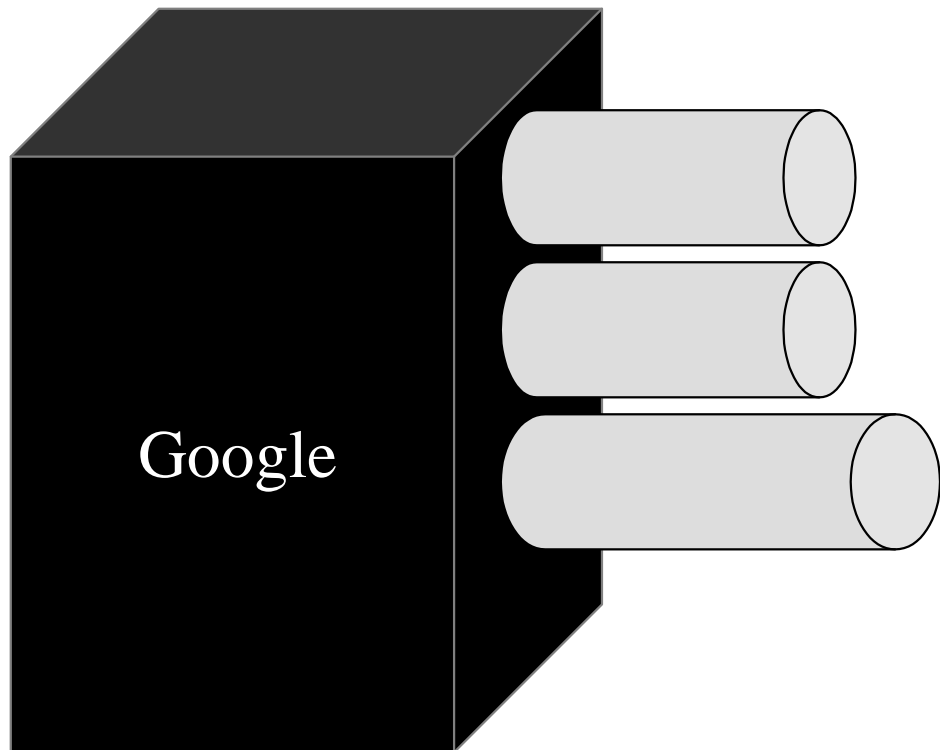
- Search Web Service
  - Monitoring
  - Marktforschung
  - Alerting
  - eigene Suchmaske
  - ...

[www.google.com/apis/](http://www.google.com/apis/)



---

# Beispiel: Google



- Rechtschreibprüfung

[www.google.com/apis/](http://www.google.com/apis/)

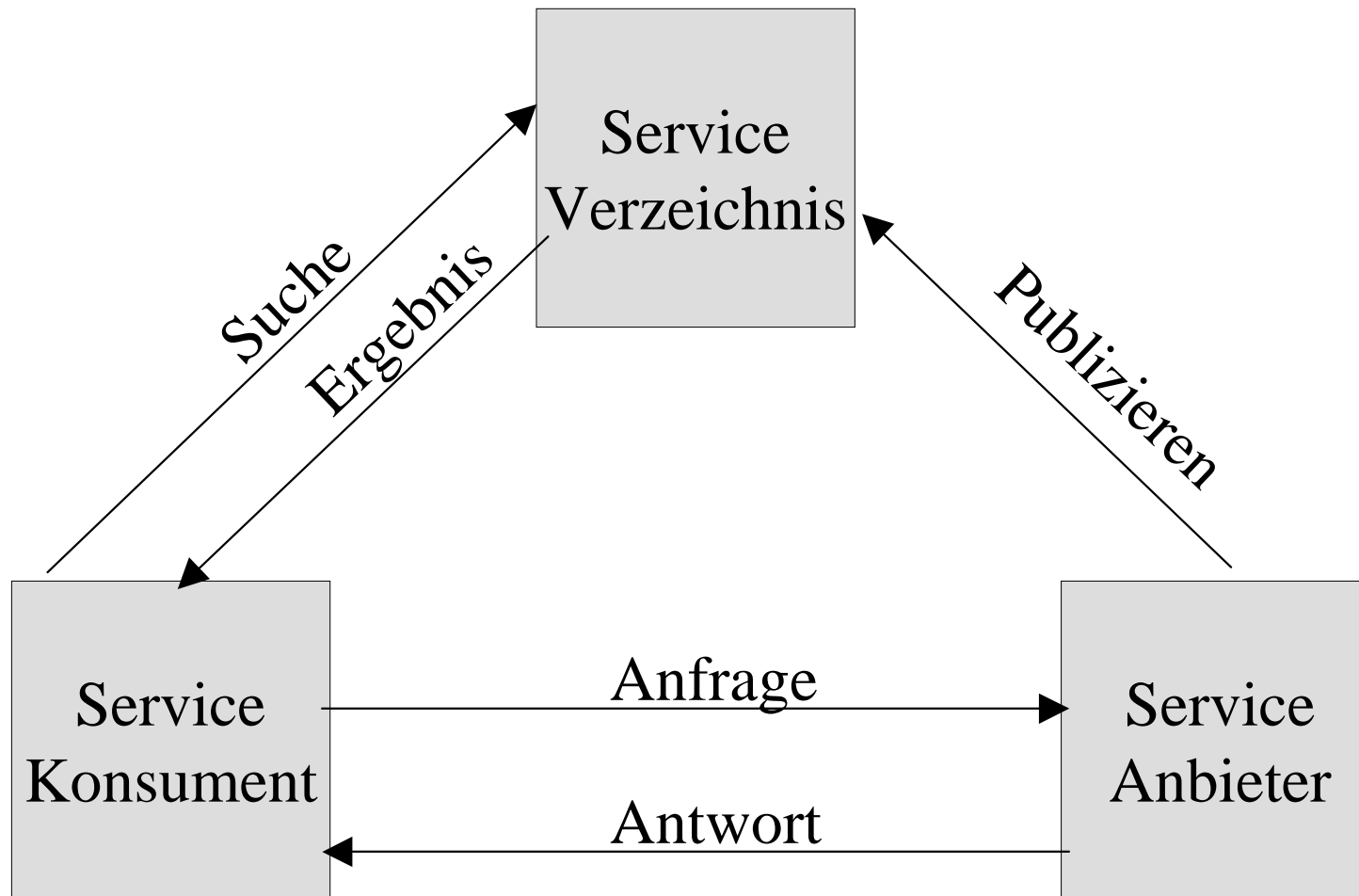




# Wie aktuell sind Web Services?

- Antwort: Es gibt seit Ende 2002 quasi einen Hype.
- .Net
- Ungelöste Probleme:
  - Kaum Standards zur Sicherheit der Dienste
  - Es fehlt an Ideen für ein Business-Modell
    - Subscription based services
    - Transaction based business model
  - Es fehlt an Ideen für wirklich neue Dienste

# Grundlegende Architektur: SOA







# Standards

## ➤ SOAP (v 1.2 W3C Standard)

Protokoll für die Nachrichten, die zwischen Web Service-Konsument und Web Service-Anbieter ausgetauscht werden. XML-basiert. Beruht ursprünglich auf Entwicklungen von Microsoft und IBM. Transport-Protokoll ist meist, aber nicht zwingend, HTTP.



# Standards

- SOAP (v 1.2 W3C Standard)
- WSDL (W3C)

Standard für die Beschreibung dessen, was zwischen Konsument und Anbieter geschickt wird.

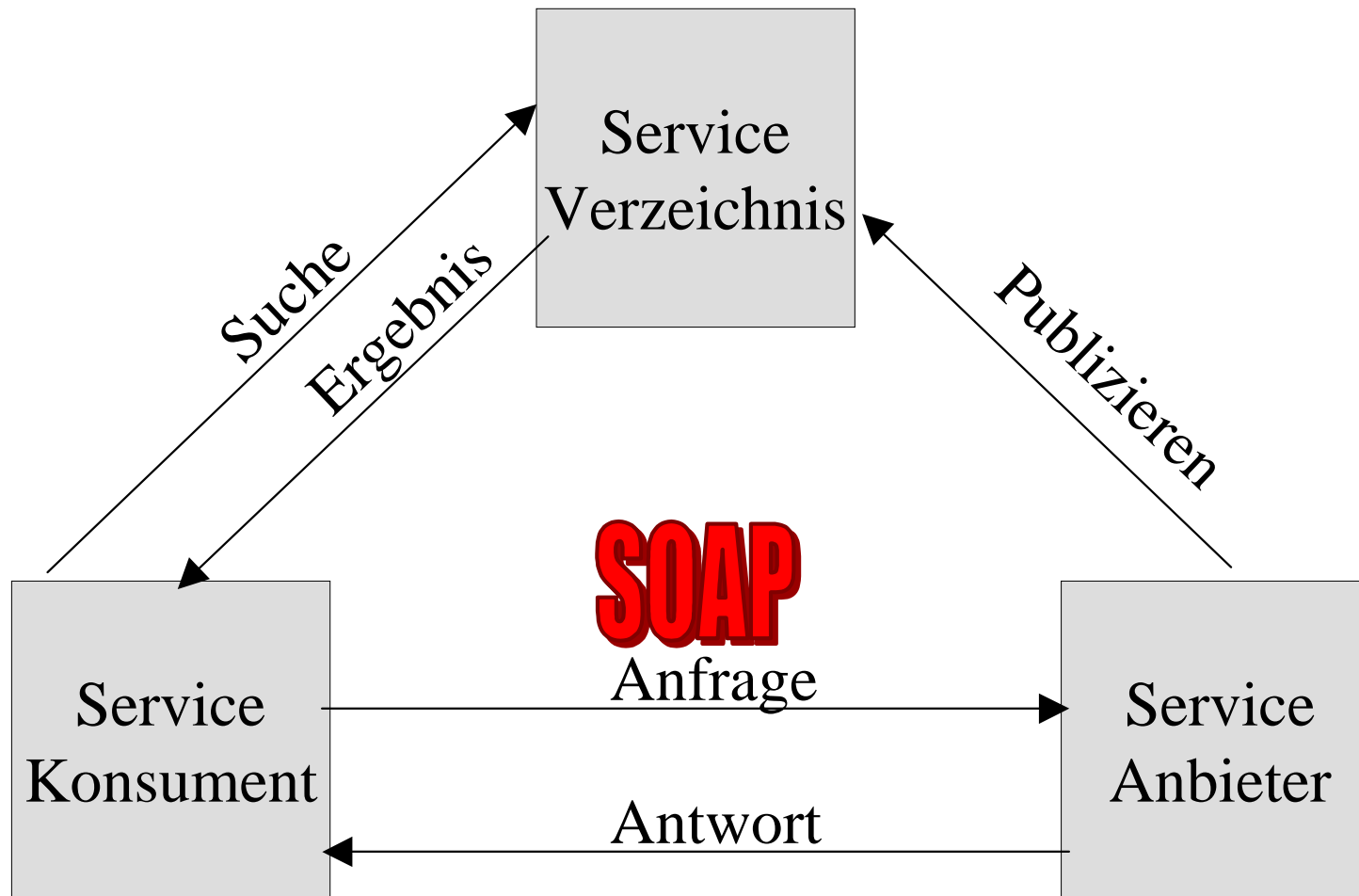


# Standards

- SOAP (v 1.2 W3C Standard)
- WSDL (W3C)
- UDDI (OASIS)

Bildet die Basis für einen Verzeichnisdienst für Web Services.

# Grundlegende Architektur: SOA





# Was ist SOAP?

- Simple Object Access Protocol
- XML-basiertes Protokoll zur Kommunikation zwischen Anwendungen
- Entwickelt 1998 von Microsoft (XML-RPC)
- Weiterentwickelt von Microsoft und IBM (SOAP 1.0)
- Seit 2000 Weiterentwicklung durch W3C (Version 1.2)



# Was ist SOAP?

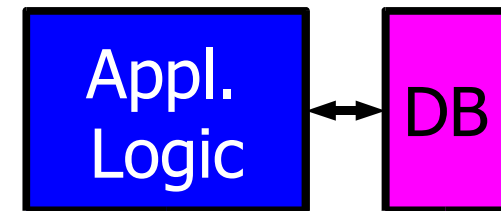
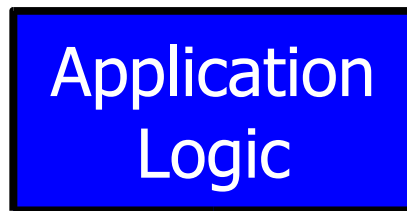
- Uns interessiert hier insbesondere SOAP als Protokoll zur Kommunikation zwischen Web Service Consumer und Web Service Provider.
- Es gibt zwei Interaktion-Muster:
  - Remote Procedure Call (RPC)
  - Document Exchange (document/literal)



## RPC vs. D/L

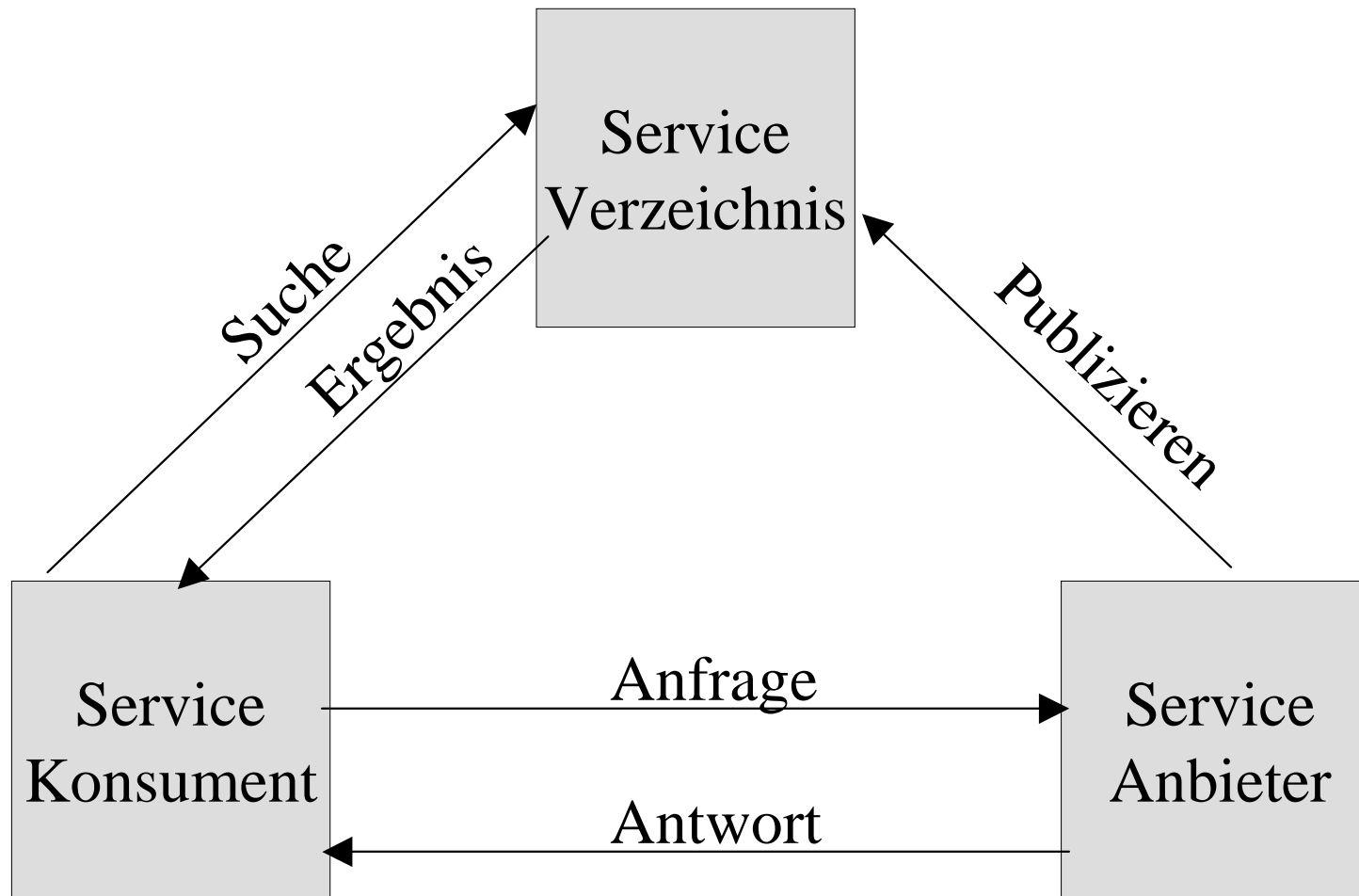
- **RPC-kodiertes SOAP**
  - Enthält einen Methodenaufruf
  - Definiert Datentypen
- **Document/literal – kodiertes SOAP**
  - Enthält nur die eigentliche Nachricht
  - Keine Typisierung
  - .NET verwendet D/L standardmäßig

# Web Service Architektur

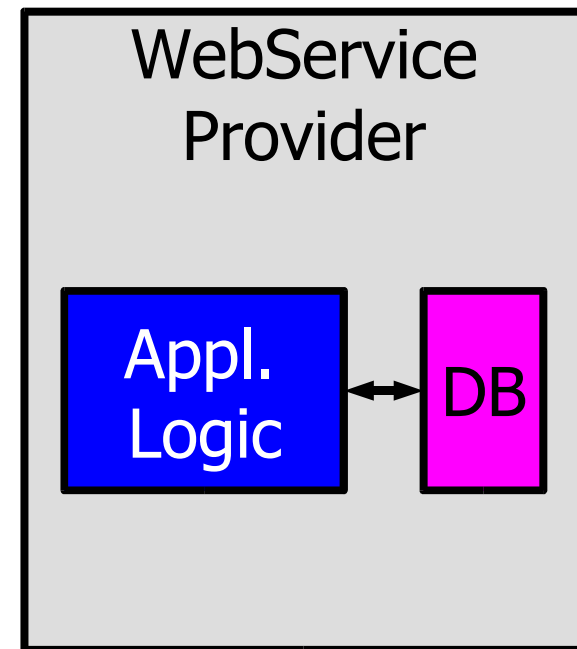
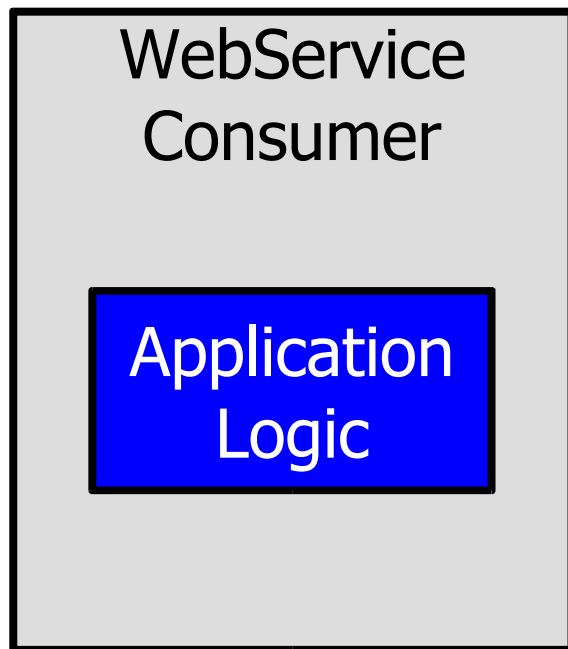




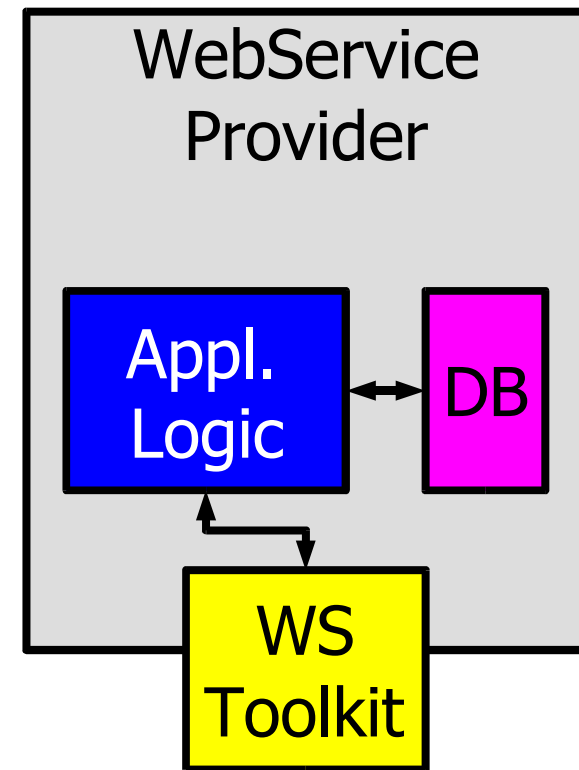
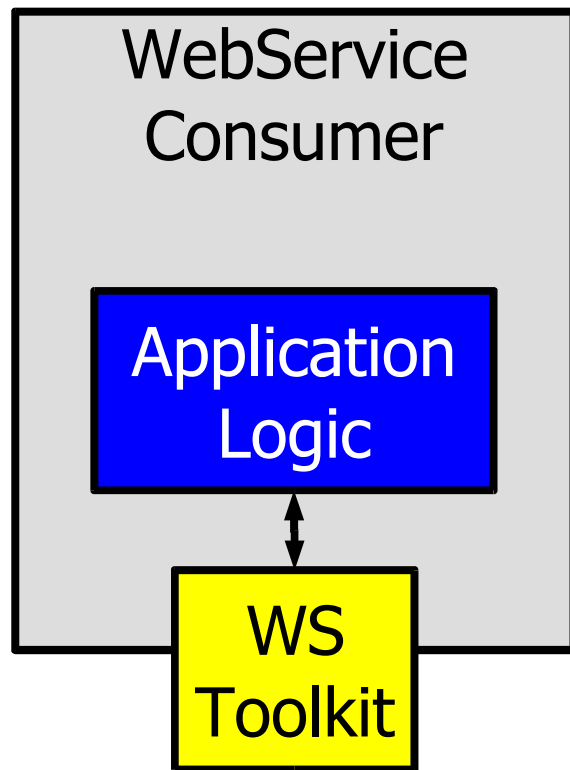
# Grundlegende Architektur: SOA



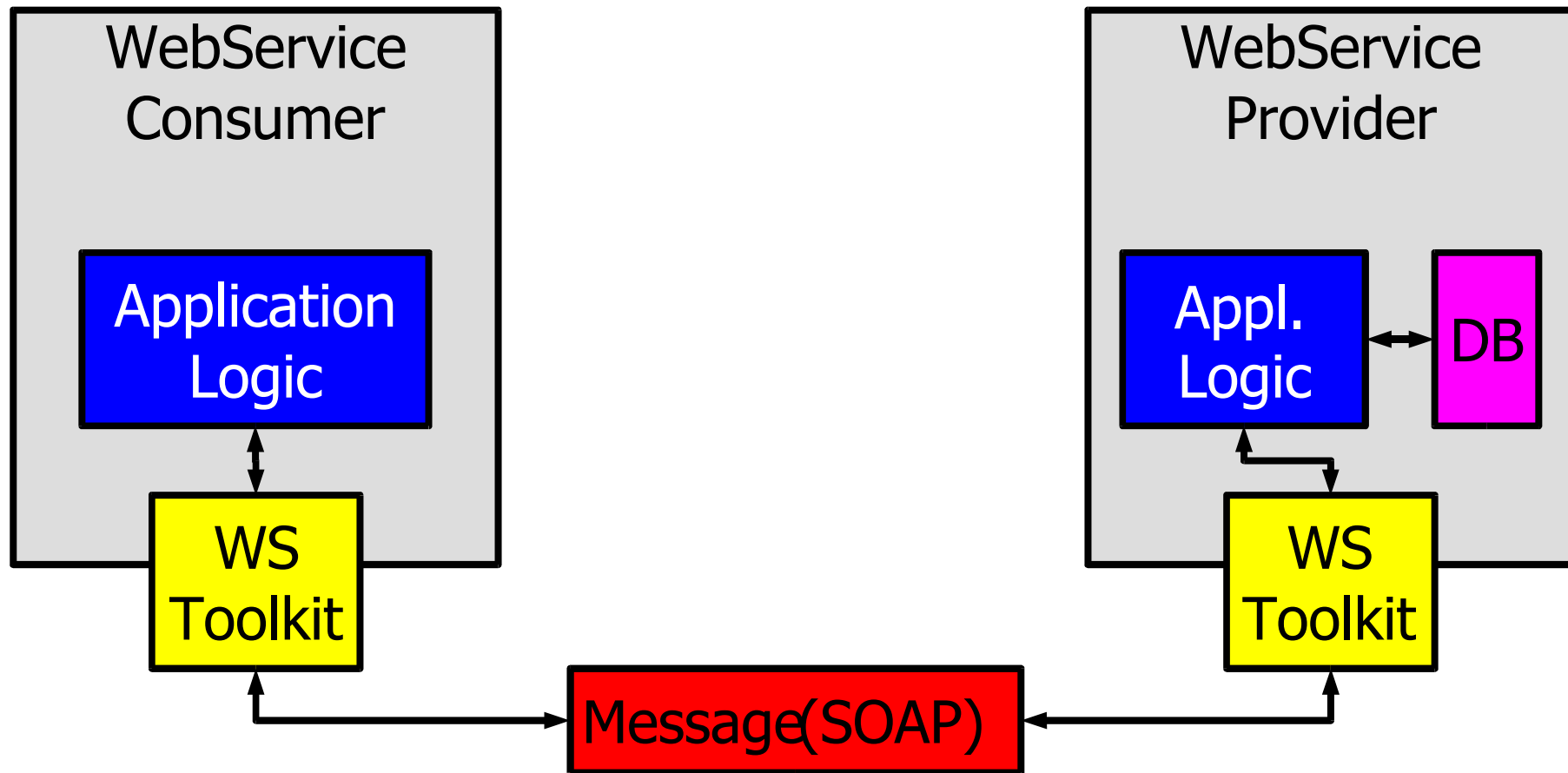
# Web Service Architektur



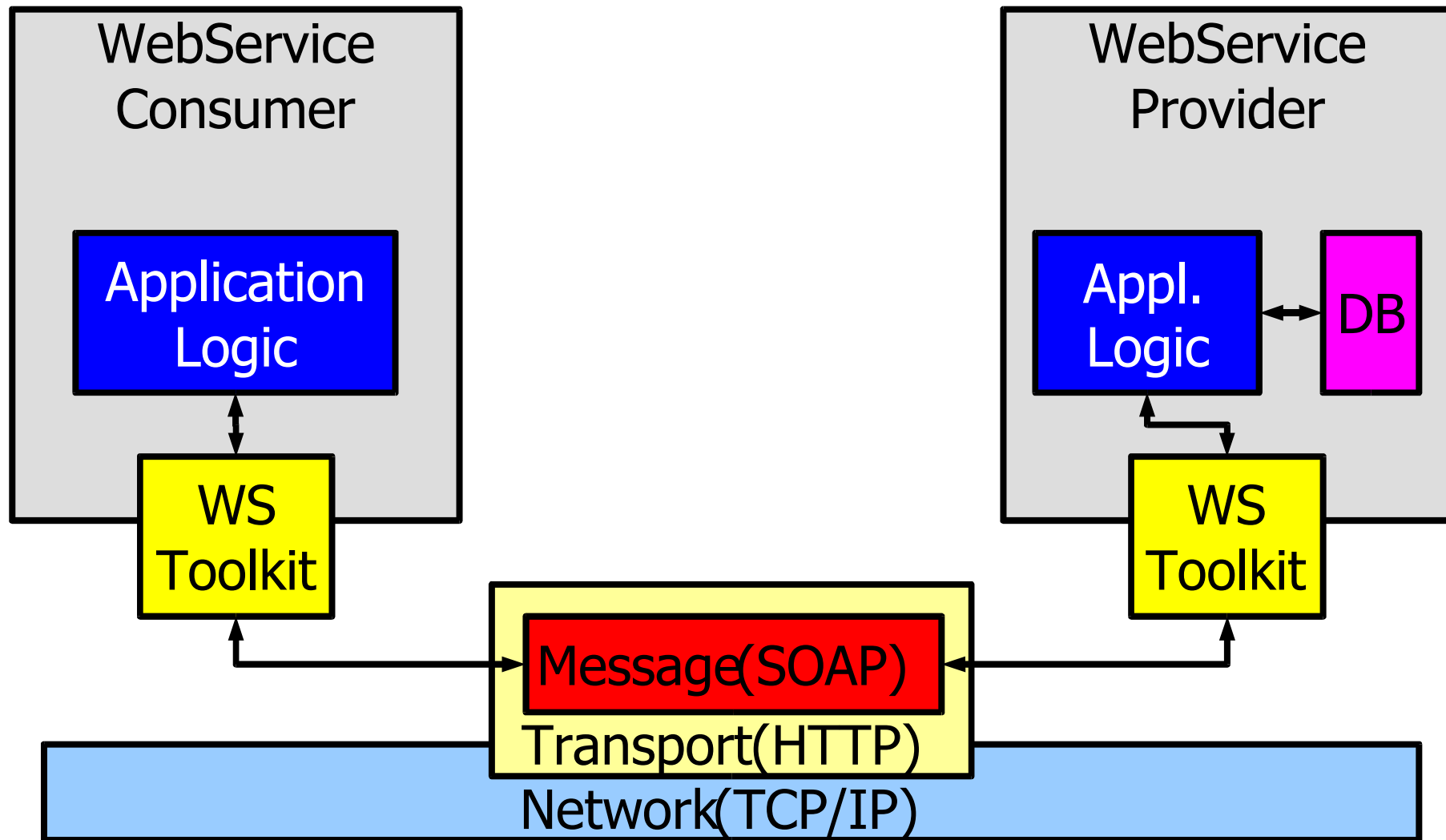
# Web Service Architektur



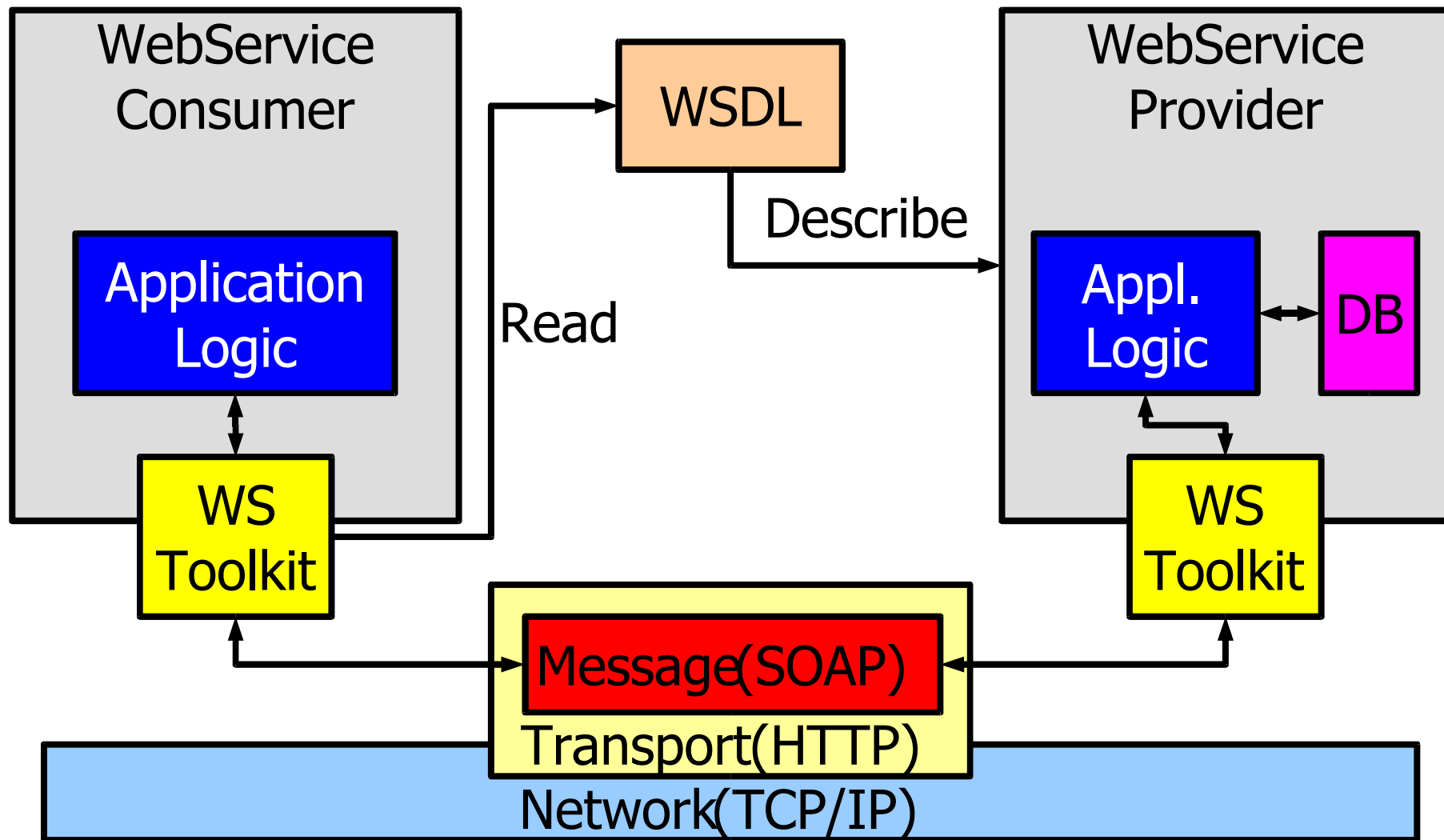
# Web Service Architektur



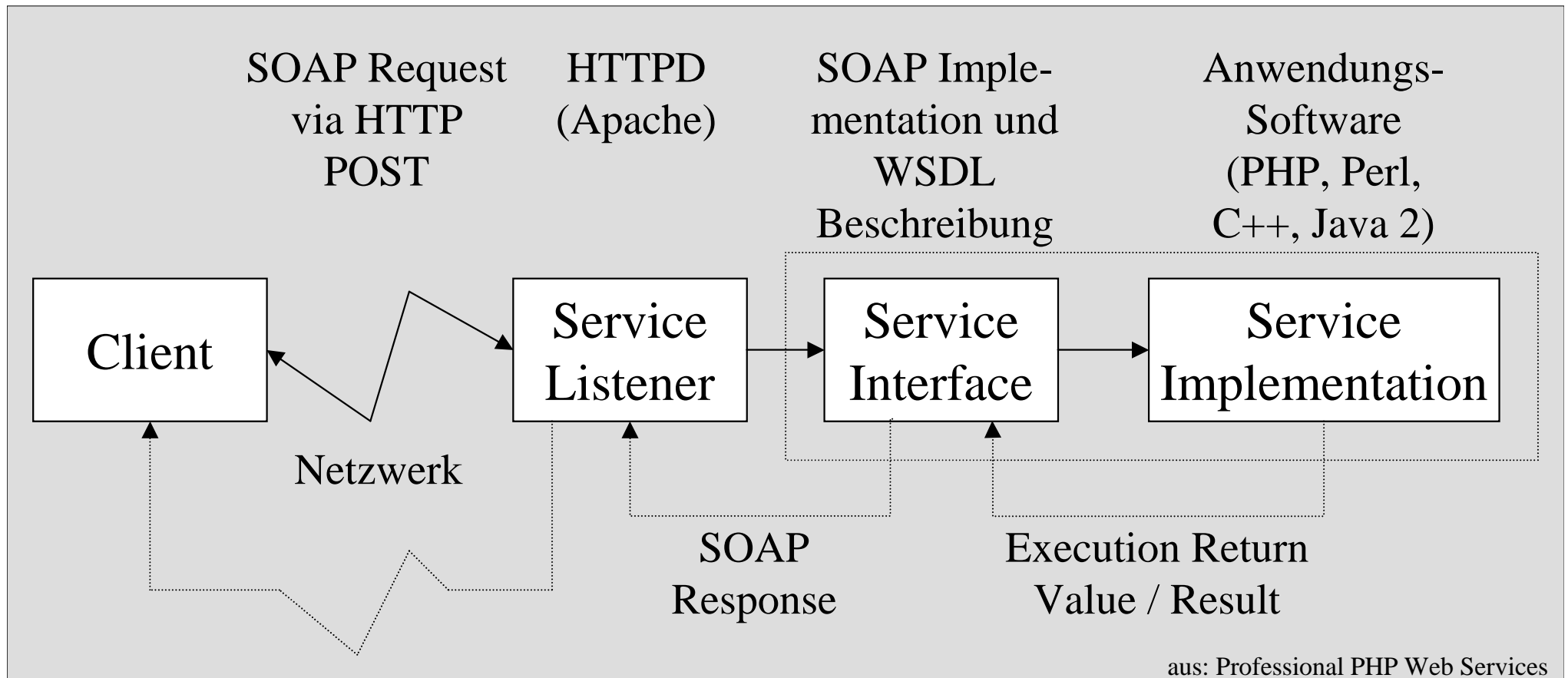
# Web Service Architektur



# Web Service Architektur



# Core Software Components



# SOAP-RPC

Provider exportiert Funktion " sqrt":

```
function sqrt($what) {  
    return sqrt($_);  
}
```

Consumer ruft die exportierte Funktion auf:

```
$server->sqrt(9);
```



# SOAP-RPC

Provider exportiert Funktion " sqrt":

```
function sqrt($what) {  
    return sqrt($_);  
}
```

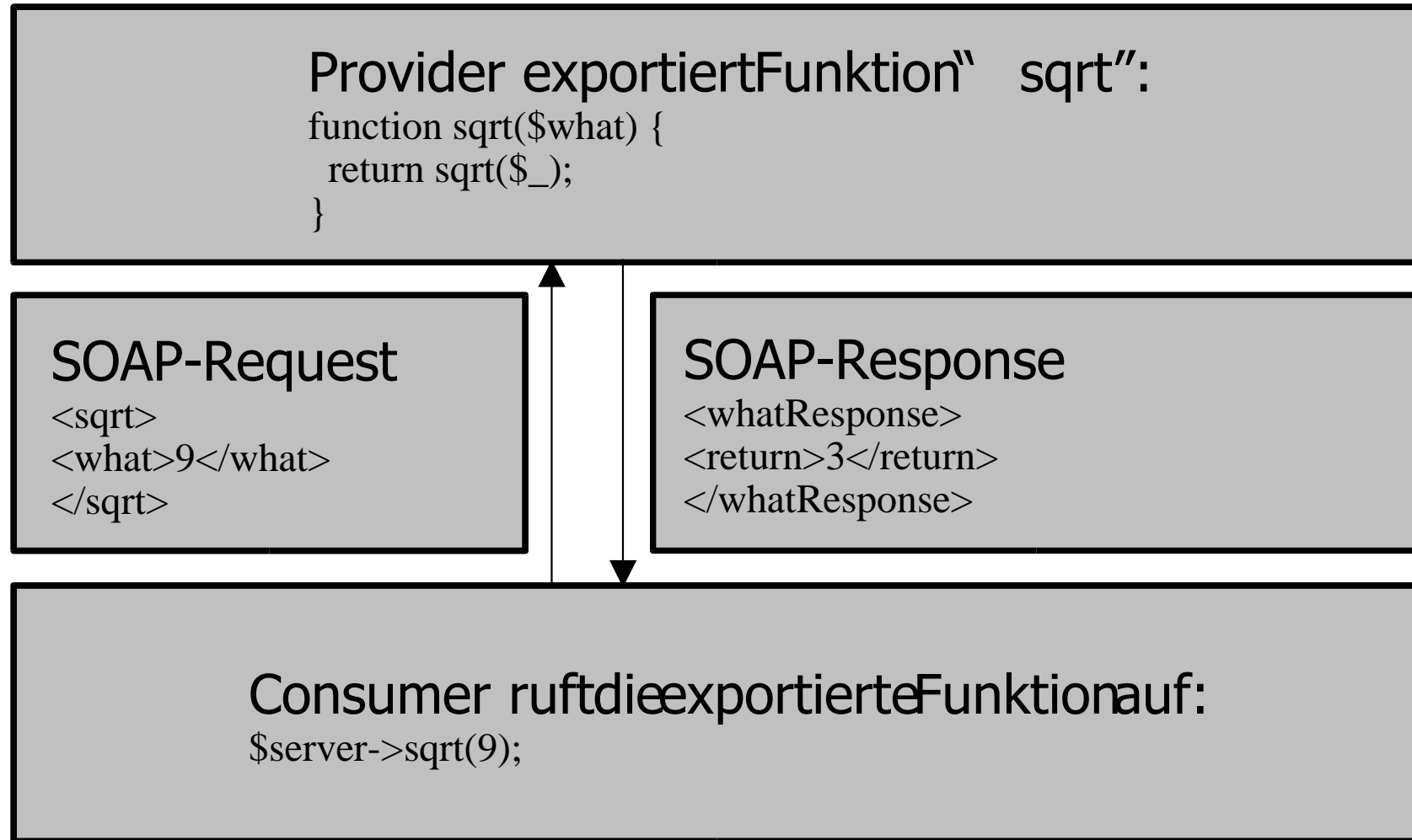
## SOAP-Request

```
<sqrt>  
<what>9</what>  
</sqrt>
```

Consumer ruft die exportierte Funktion auf:

```
$server->sqrt(9);
```

# SOAP-RPC





# APIs

- Perl, Java, C++, .NET, PHP, ...
- PHP
  - PHP SOAP
  - PEAR::SOAP
  - nuSOAP
  - eZ soap

# Wie nutze ich die APIs?

- nuSOAP besteht aus einer ZIP-Datei, die das Skript nusoap.php enthält. Diese sind in die eigenen Skripte zu integrieren, fertig.
- PEAR::SOAP wird mit dem PEAR-Framework installiert, dieses funktioniert in den aktuellen Perl-Versionen unter Linux und Windows reibungslos. Allerdings sind vorher weitere Pakete in PHP zu integrieren (PEAR erklärt das aber verständlich):
  - Mail\_Mime
  - HTTP\_Request
  - Net\_URL
  - Net\_Dime

# Weil es so schön einfach ist: Ein „Hello nuSOAP“ Provider

```
<?php
require_once ( "nusoap.php" );
$server = new soap_server();

$server -> register ( "Abfrage" );

// Beliebige Abfrage
function Abfrage() {
    return "Hallo Welt";
}

$server -> service( $HTTP_RAW_POST_DATA );
exit();
?>
```

... noch einfacher ist der Consumer zu implementieren...

```
<?php
require_once ( "nusoap.php" );
$client = new soapclient( "http://localhost/
                           nusoap/anbieter.php" );

$ergebnis = $client->call( "Abfrage" );

print $ergebnis;
?>
```

# Anfrage des Consumers an den Provider

```
<?xml version="1.0" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:si="http://soapinterop.org/xsd"
xmlns:ns6="http://testuri.org"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ns6:Abfrage>
<target xsi:type="xsd:string">all</target>
</ns6:Abfrage>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# Die Antwort des Providers

```
<?xml version="1.0" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
...
xmlns:ns6="http://testuri.org"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ns6:AbfrageResponse>
Hallo Welt
</ns6:AbfrageResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```






## Beispiel: VASCODA

- [www.vascoda.de](http://www.vascoda.de)
- Es gibt ein Portal (Service-Consumer)
- Es gibt viele Service-Provider
- VASCODA nutzt SOAP-RPC
- Jeder Server exportiert die Funktion `query`
- SOAP-Request enthält Suchkriterien
- SOAP-Result enthält Ergebnisdatensätze (im VASCODA-eigenen Format)

- Home
- Neue Suche
- Merkliste
- News
- Über Uns
- Kontakt
- Hilfe
- Impressum
- Disclaimer
- AGB's

## Suche

[English Version](#)

 Suche starten

Suchen in:



Ingenieur- und Naturwissenschaften



Medizin- und Biowissenschaften



Wirtschafts- und Sozialwissenschaften



Geistes- und Kulturwissenschaften / Regionen

## Fachzugänge

Ingenieur- und Naturwissenschaften



Medizin- und Biowissenschaften



Wirtschafts- und Sozialwissenschaften



Geistes- und Kulturwissenschaften / Regionen



## Beachten Sie unser Buchmesse-Gewinnspiel

Gefördert durch:



Bundesministerium  
für Bildung  
und Forschung

**DFG**

14384 Treffer für [ atom\* ]

- Home
- Neue Suche
- Merkliste
- News
- Über Uns
- Kontakt
- Hilfe
- Impressum
- Disclaimer
- AGB's

- 25 [Physik](#)  
(ViFaPhys)
- 4 [Geschichte - Anglo-Amerika](#)  
(VLib-AAC)
- 1 [Mathematik](#)  
(MathGuide)
- 1 [Vorderer Orient und Nordafrika](#)  
(MENALIB)
- 7 [Politikwissenschaft und Friedensforschung](#)  
(politics and peace guide)
- 3 [Wirtschaftswissenschaften](#)  
(EconBIZ)
- 1354 [Sozialwissenschaften](#)  
(infoconnex)
- 1 [Geschichte - Übersicht](#)  
(CLIO-Online)
- 5820 [Medizin](#)  
(Informationsverbund Medizin)
- 1298 [Pädagogik](#)  
(infoconnex)

**Physik (ViFaPhys)**

[1-3] von 25 [1-10][11-20][21-25] ▶

- 1 [NIST Atomic Spectra Database](#), Atomic Energy Levels Data Center, Physics Laboratory, National Institute of Standards and Technology ...   
ViFaPhys
- 2 [Aladdin](#), Atomic and Molecular Data Unit, Nuclear Data Section, International Atomic Energy Agency   
ViFaPhys
- 3 [GENIE : A General Internet Search Engine for Atomic Data](#), GAPHYOR Data Center; International Atomic Energy Agency   
ViFaPhys

[\[zurück nach oben\]](#)

**Geschichte - Anglo-Amerika (VLib-AAC)**

[1-3] von 4 [1-4] ▶

- 1 [A-Bomb WWW Museum](#), [Editorial Team] A-Bomb WWW Project 1999, History Guide
- 2 [Historical Nuclear Weapons Test Films](#), 2001, History Guide
- 3 [Historical Reports \[about the United States nuclear testing program\]](#), 2001, History Guide

[\[zurück nach oben\]](#)

# VASCODA WSDL (Ausschnitt)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="urn:VASCODAService" targetNamespace="urn:VASCODAService" xmlns:typens="urn:VASCODAService"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://myLocation/vascoda.wsdl"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <types>
- <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:VASCODAService">
  - <xsd:complexType name="QueryResult">
    - <xsd:all>
      <xsd:element name="provider" type="xsd:string" />
      <xsd:element name="no_of_answers" type="xsd:int" />
      <xsd:element name="first_answer" type="xsd:int" />
      <xsd:element name="last_answer" type="xsd:int" />
      <xsd:element name="ANSWERS" type="typens:ArrayOfAnswerElement" />
    </xsd:all>
  </xsd:complexType>
- <xsd:complexType name="ArrayOfAnswerElement">
  - <xsd:complexContent>
    - <xsd:restriction base="soapenc:Array">
      <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:AnswerElement[]" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
- <xsd:complexType name="AnswerElement">
  - <xsd:all>
    <xsd:element name="creator" type="xsd:string" />
    <xsd:element name="title" type="xsd:string" />
    <xsd:element name="journal_title" type="xsd:string" />
    <xsd:element name="series_title" type="xsd:string" />
    <xsd:element name="date" type="xsd:int" />
    <xsd:element name="issn" type="xsd:string" />
    <xsd:element name="identifier" type="xsd:string" />
    <xsd:element name="source" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
</xsd:schema>
</types>
- <message name="queryRequest">
  <part name="type" type="xsd:string" />
  <part name="first_answer" type="xsd:int" />
  <part name="last_answer" type="xsd:int" />
  <part name="subject" type="xsd:string" />
  <part name="creator" type="xsd:string" />
  <part name="title" type="xsd:string" />
  <part name="date" type="xsd:int" />
</message>
- <message name="queryResponse">
  <part name="result" type="typens:QueryResult" />
</message>
```

Eine Kopie der VASCODA-wsdl liegt in:  
[vifaphys.tib.uni-hannover.de/WebServices/VASCODA/](http://vifaphys.tib.uni-hannover.de/WebServices/VASCODA/)

# Anfrage des Consumers (=VASCODA-Kern) an den Provider (=ViFa)

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:VASCODAService"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>

<ns4:query>
<type xsi:type="xsd:string">basic</type>
<first_answer xsi:type="xsd:int">1</first_answer>
<last_answer xsi:type="xsd:int">100</last_answer>
<subject xsi:type="xsd:string">atom*</subject>
<creator xsi:type="xsd:string"></creator>
<title xsi:type="xsd:string"></title>
<date xsi:type="xsd:int"/>
</ns4:query>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# Antwort des Providers (=ViFa) an den Consumer (=VASCODA-Kern)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  ...
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
<SOAP-ENV:Body>

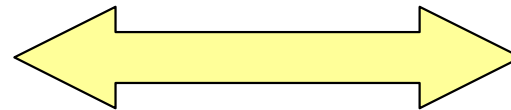
<ns4:queryResponse>
<result>
<provider xsi:type="xsd:string">ViFaPhys</provider>
<no_of_answers xsi:type="xsd:int">25</no_of_answers>
<first_answer xsi:type="xsd:int">1</first_answer>
<last_answer xsi:type="xsd:int">100</last_answer>

<ANSWERS xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="ns4:AnswerElement[25]" SOAP-ENC:
<answer xsi:type="ns4:AnswerElement">
<creator xsi:type="xsd:string">Atomic Energy Levels Data Center, Physics Laboratory, I
<title xsi:type="xsd:string">NIST Atomic Spectra Database</title>
<identifier xsi:type="xsd:string">http://vifaphys.tib.uni-hannover.de/fi.php?action=ol
<source xsi:type="xsd:string">ViFaPhys</source>
</answer>
<answer xsi:type="ns4:AnswerElement"> <creator xsi:type="xsd:string">Atomic and Molec
<title xsi:type="xsd:string">Aladdin</title>
...
```

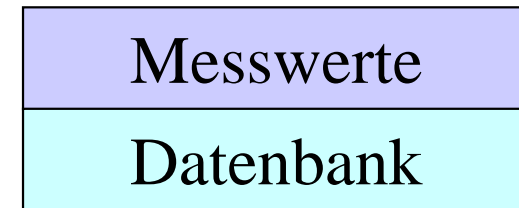
# Ein weiteres Beispiel:

Arbeitsgruppe A  
Theorie

Programm-Code



Arbeitsgruppe B  
Praktische Messungen



Gegenseitige Integration  
mittels Web Services

...die möglichen Anwendungen sind  
beinahe unbegrenzt

Arbeitsgruppe A  
Theorie

Programm-Code

Arbeitsgruppe B  
Praktische Messungen

Messwerte

Datenbank

Software zur graphischen Aufbereitung

Dienstleister



## Literatur, Weiterführendes

- **Professional PHP Web Services**  
Fuller, Fuecks, Egervari, Waters et.al., wrox, ISBN 1-861008-07-4
- **Professional Open Source Web Services**  
Sarang, Brown et.al., wrox, ISBN 1-861007-46-9
- **Using the Google Web APIs**  
Haris Kamath, [www.devshed.com/Server\\_Side/PHP/GoogleAPI/](http://www.devshed.com/Server_Side/PHP/GoogleAPI/)
- **Web Services mit PHP**  
Tobias Hauser, PHP Magazin 6, 2003, pp.34-38
- **Create you own Search Engine with PHP...**  
Ahm Asaduzzaman, [www.devarticles.com/art/1/422/](http://www.devarticles.com/art/1/422/)
- **Ausführliches Tutorial (Sammlung von Artikeln)**  
[www.phppatterns.com/index.php/link/category/15/](http://www.phppatterns.com/index.php/link/category/15/)



Herzlichen Dank für Ihre Aufmerksamkeit

Thomas Severiens

Institute for Science Networking

severiens@isn-oldenburg.de

Hinweis: Einige der Folien dieses  
Vortrages stammen von Christian Thiemann  
aus einem gemeinsamen Vortrag über VASCODA