



## WebServices with PHP5

International PHP Conference 2005, 11/09/2005

Björn Schotte



## Björn Schotte - „PHP Pioneer“

- 1998/1999: PHP-Center.de
- 2000: world's first PHP Conference
- 2001: editor-in-chief PHP Magazin
- 2001: International PHP Conference
- 2001: co-founder of ThinkPHP
- 2004: refounding Mayflower GmbH



## ThinkPHP – The LAMP Experts



- high-level software development since 2001
- PHP trainings for more than 300 people from Top50 companies
- PHP Support – <http://www.thinkphp.de/support/>
  - Enterprise PHP Support
  - Guaranteed reaction times up to 2 hours
  - Basic, Professional, Gold – 24/7 available on request
- Fostering the PHP community and PHP itself in Germany
- „ThinkPHP by Mayflower“ – a division of Mayflower GmbH

- ~ 35 employees
- „ThinkPHP – by Mayflower“
- Customers include:
  - Siemens COM
  - Vaillant Group
  - Telefónica Germany
  - Landeshauptstadt München
  - HypoVereinsbank
  - Sixt AG
  - Stadtwerke München (SWM)
  - Bundeskriminalamt
  - ...
- 4 managing partners (founders of ThinkPHP)
- 2 offices (Munich and Würzburg)
- well-known in the PHP community
- Know how is more than PHP-only (High Performance web architecture, .NET, ...)
- Big projects in the range of man-years with PHP (HVB, Vaillant, Sixt, Telefónica)
- OpenSource Software vendor (i.e. PHProjekt, Outlook Sync, ...)

# WebServices – short introduction

- RPC: Remote Procedure Calls
- distributed Applications over the Web
- mostly using HTTP as transport protocol
- sometimes SMTP
- mostly XML based – thus language independant
- Used for application integration
- Alternative to CORBA or DCOM
- Cross-platform



## WebServices – a hype?

---

- industry dream of totally distributed applications
- hyped SOAP, one of the RPC protocols
- the hype began around 1998/1999
- reality: sometimes too complex, not easy to setup, sometimes not easy to understand
- SOAP as a mostly bloated protocol on HTTP



# SOAP

- Simple Object Access Protocol
- works over HTTP (but also with SMTP)
- state-less as in HTTP
- W3C standard



# Why using SOAP

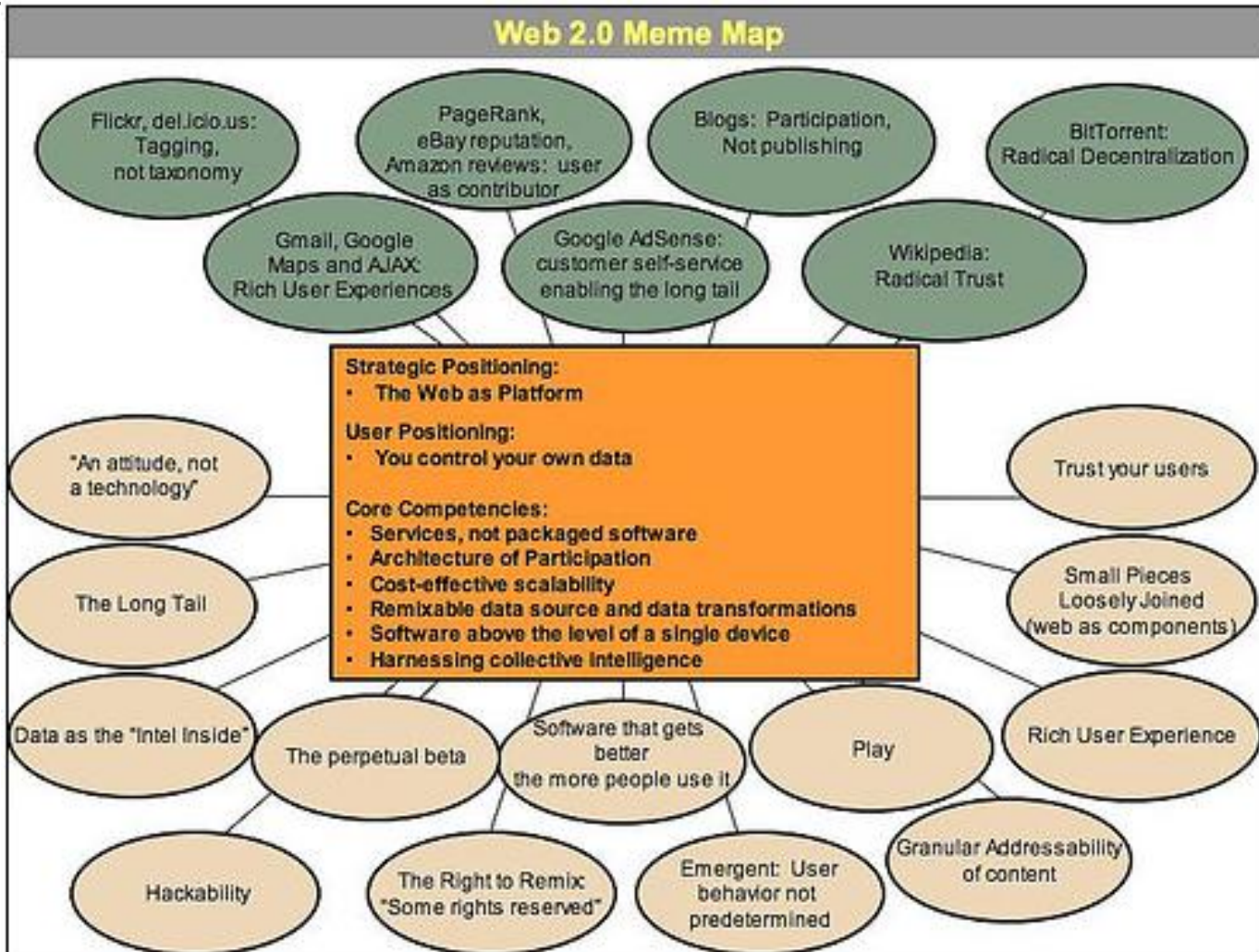
- SOA – service oriented applications
- based on HTTP
- value-added service to web applications
  - imagine: integration of your application's functionality in the customer's CRM tool



# SOAP looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:amazon="http
://soap.amazon.com">
  <SOAP-ENV:Body>
    <namespace37:ManufacturerSearchRequestResponse xmlns:namespace37="http://soap.amazon.com">
      <return xsi:type="amazon:ProductInfo">
        <TotalResults xsi:type="xsd:string">165</TotalResults>
        <Details SOAP-ENC:arrayType="amazon:Details[10]" xsi:type="SOAP-ENC:Array">
          <Details xsi:type="amazon:Details">
            <Url xsi:type="xsd:string">http://www.amazon.com/...</Url>
            <Asin xsi:type="xsd:string">B00004OCCR</Asin>
            <ProductName xsi:type="xsd:string">OXO Good Grips Salad Spinner</ProductName>
            <Catalog xsi:type="xsd:string">Kitchen</Catalog>
            ...
```

# Hype leads to „Web 2.0“



## Web 2.0 – everything is open

---

- open APIs
- users control their own data
- new applications can appear by just combining different Web2.0 tools (Flickr etc.)
- new social platform hype: [www.ning.com](http://www.ning.com)



# RPC/WebServices in PHP

- ext/xmlrpc
- ext/xmlrpci
- NuSOAP
- PEAR::SOAP
- ext/soap (already existed in PHP4!)
- Manual: <http://www.php.net/soap>
- ReST



# What is ReST?

- ReST is simple:

<http://url/someservice.php?mode=foo&param1=bar>

- gives a specific result that you can interpret
- disadvantage: you have to define the „protocol“ itself
- It points out that WebServices from YaHoo! etc. will mostly be used over the ReST interface because it's simple

## WebServices – what to use?

- ext/soap with PHP5
- backed by cooperation of Zend & IBM with upcoming support for enhanced SOAP standard
- The most reliable SOAP implementation with PHP, especially when .NET and Java SOAP clients are talking to your PHP server

## WebServices – a small PHP client

- create instance of SoapClient
- configurable constructor (stack tracing, exception handling yes/no etc.)
- define details of the Webservice to be accessed
  - Recommend to use WSDL – everything else is a mess
- ext/soap transparently maps the remote methods as if it was a PHP class method!

# SOAP Client - example

```
<?php
$client = new SoapClient(NULL,
    array("location" => "http://services.xmethods.net:80/soap",
        "uri" => "urn:xmethods-CurrencyExchange",
        "style" => SOAP_RPC,
        "use" => SOAP_ENCODED));

$method_name = "getRate";
$method_params = array(new SoapParam("United States", "country1"),
    new SoapParam("Canada", "country2"));

$soap_options = array("uri" => "urn:xmethods-CurrencyExchange");
$retval = $client->__call($method_name,
    $method_params,
    $soap_options);

print "Every $1 USD will give you \$$retval Canadian Dollars";

?>
```

## Please use WSDL

- Web Service Description Language
- a very bloated protocol
- but makes it easier for you to use the SOAP webservice!
- can be hand written, but there are tools that do the work for you
- should be hand written in order to create a very small WSDL file

## WebService Client using WSDL

```
<?php
    $wsdl = "http://www.xmethods.net/sd/20
01/CurrencyExchangeService.wsdl";
    $client = new SoapClient($wsdl);

    $retval = $client-
>GetRate("United States", "Euro");

    print "Every $1 USD will give you \$$r
etval Euros";
?>
```

## Configuration in php.ini

- ext/soap can cache the WSDL file
- soap.wsdl\_cache\_enabled = 1
- soap.wsdl\_cache\_dir = „/my/tmp“
- soap.wsdl\_cache\_ttl = 3600



# PHP as a SOAP WebServer

- Create the WSDL file as needed for your web service
- Consult the W3C standard how to do that
- ... but, some example



## Complex types

- imagine you want to return not only a value (i.e. boolean), but a list of values (i.e. an array)
- `return array('foo', 'bar');` won't work in the SOAP server method
- wrong SOAP response XML will be generated
- instead, deal with `SOAPVar()` class that is provided by `ext/soap` to create the correct XML response
- .NET especially has problem with lists that contain lists (i.e. arrays of arrays)
- you have to define the inner list outside of the outer list in the WSDL
- then everything goes fine 😊

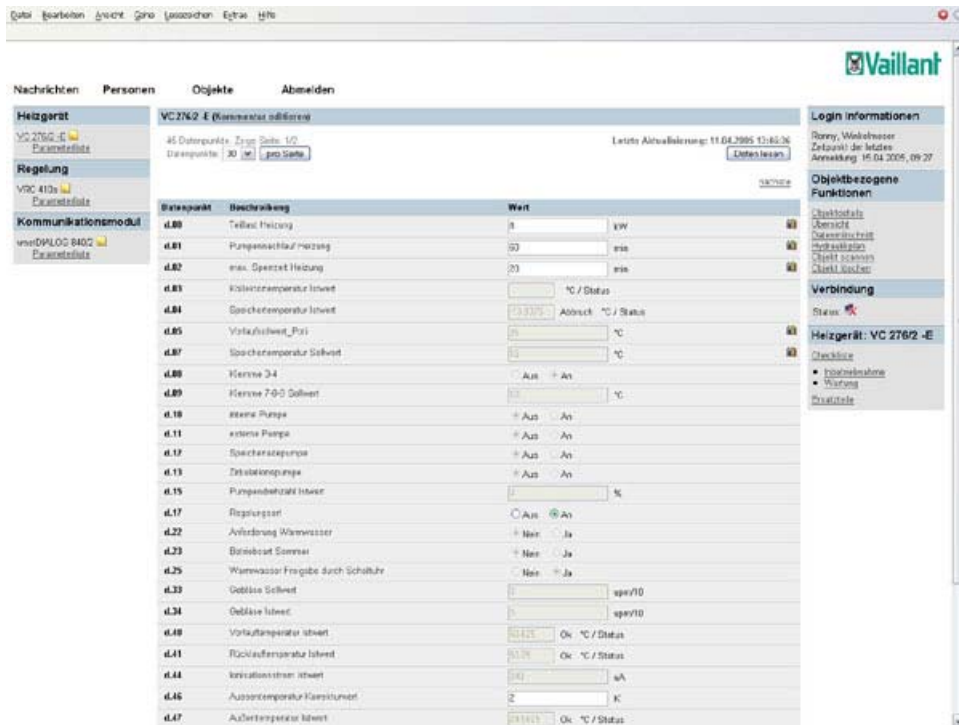
# Complex types in SOAP

```
| <xsd:complexType name="DataValueItem">
|     <xsd:all>
|         <xsd:element name="dpvalue" type="xsd:string"/>
|         <xsd:element name="writeable" type="xsd:boolean"/>
|     </xsd:all>
| </xsd:complexType>
|
| <xsd:complexType name="DataValueList">
|     <xsd:complexContent>
|         <xsd:restriction base="soapenc:Array">
|             <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="typens:DataValueItem[]" />
|         </xsd:restriction>
|     </xsd:complexContent>
| </xsd:complexType>
```

```
class DataValueItem {
    var $dpvalue;
    ...
    function DataValueItem($dpvalue, ...) {
        $this->dpvalue = (string)utf8_encode($dpvalue);
    }
}

class MyWebService {
    function mySOAPFunction() {
        $arr = somewhatGetDPArray();
        foreach ($arr as $data) {
            $ret[] = new SOAPVar(new
DataValueItem($d[„value“],...), SOAP_ENC_OBJECT,
„DataValueItem“, „urn:NamespaceofWebservice“);
        }
        return $ret;
    }
}
```

# Use Case at a large PHP project



Let's have a look at the code

---

MAYFLOWER





**Thank you!**

Björn Schotte  
Mayflower GmbH  
Pleichertorstr. 2  
97070 Würzburg  
+49 (931) 35 9 65 - 0  
[schotte@mayflower.de](mailto:schotte@mayflower.de)

