

Web Service Facade for PHP5

Andreas Meyer, Sebastian Böttner, Stefan Marr



Agenda

➤ Objectives and Status

- Architecture
- Framework Features

➤ WSD Generator

- PHP5 Reflection API

➤ Security Aspects

- used approach
- planned techniques
- Web Services Security
 - Username Token Profile 1.0
 - Further used WSS features

➤ Coding Guidelines

- PHPDoc Tags
- Examples based on current TT-Implementation

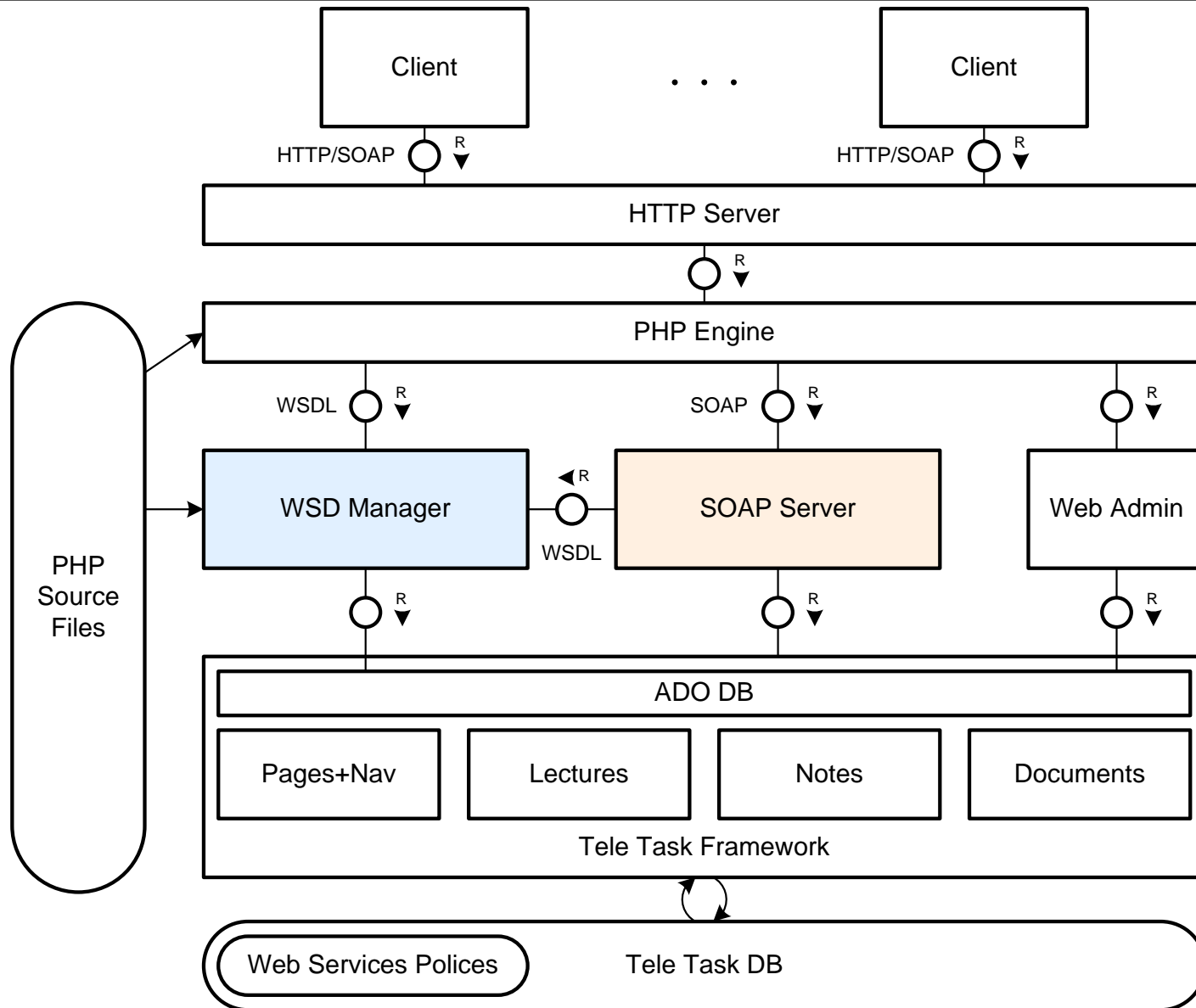
Objectives

- **Tool for generating WSDL-files from PHP5 code**
- **Inspect code and generate XSD-files for used parameter types**

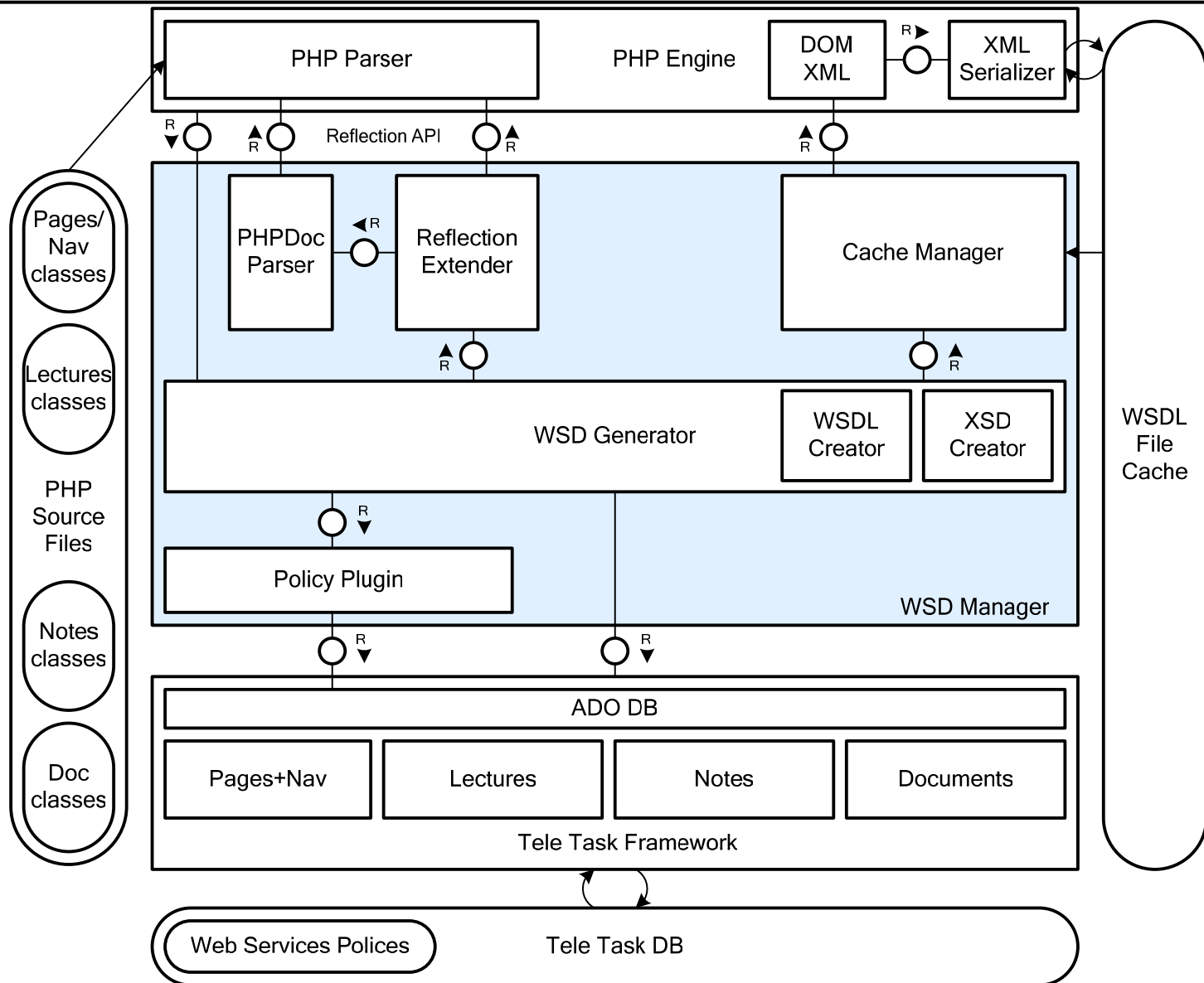
- **Building a framework**
 - **Combine tools**
 - **Provide SOAP-Server for TT**
 - **Consider security aspects**
 - **Personalized services**
 - **Authentication**

 - **Web-Based SOAP-Server Configuration**
 - **Example implementation based on old TT database**
 - **Part of framework documentation**
 - **Including guidelines and hints for usage**

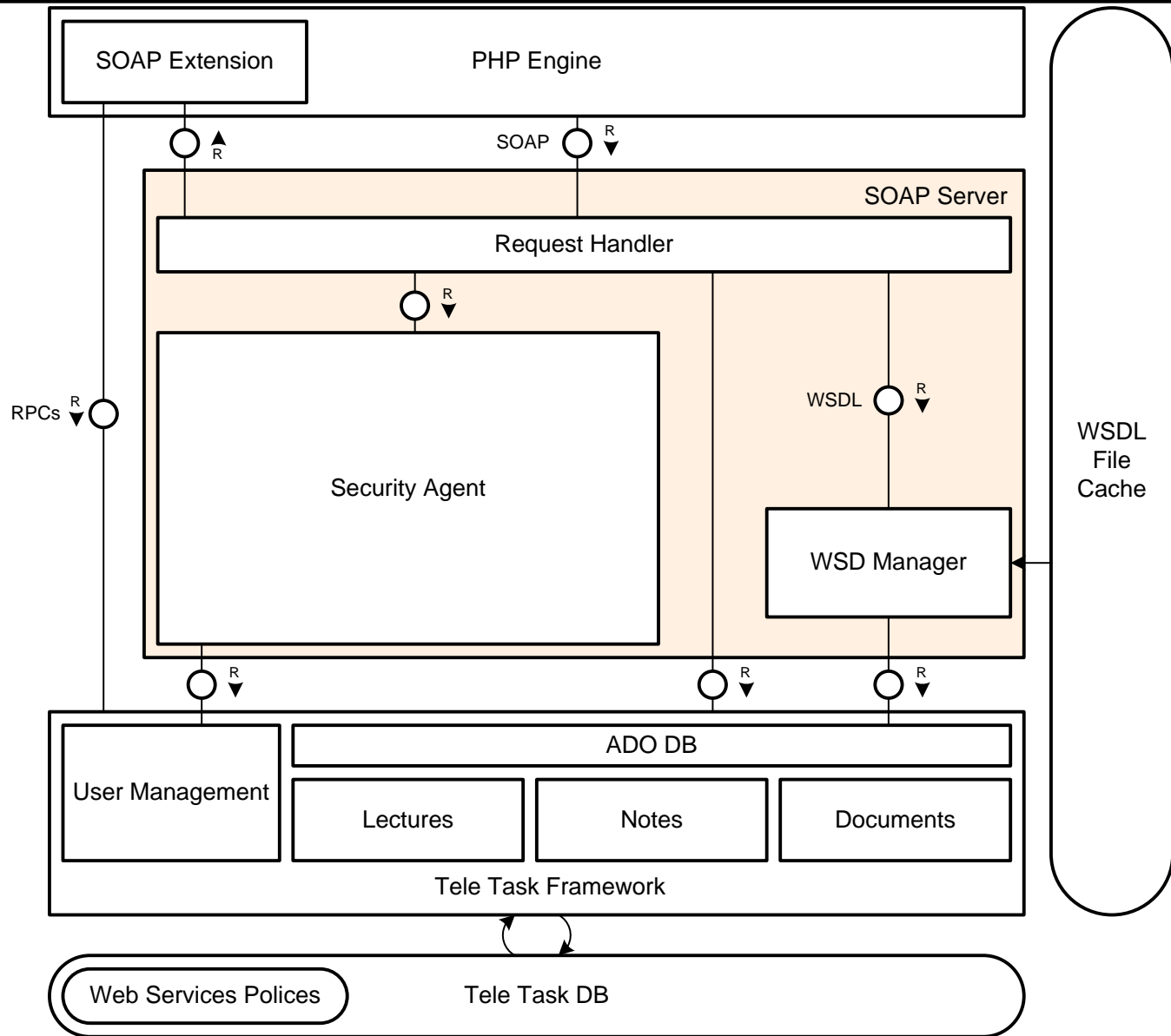
Architecture



WSD Manager

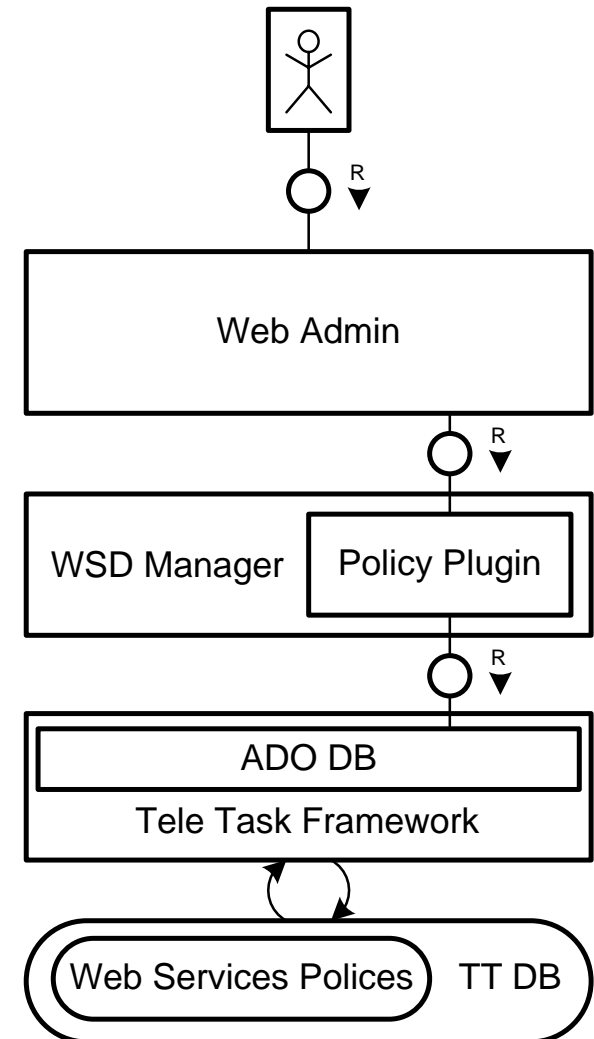


SOAP Server



Web Admin Features

- **Set polices for provided web services**
 - **Activate classes to provide Web Services**
 - **Choose published methods**
 - **Only public methods**
- **Adjust documentation published in WSDL**



Constrains for this Approach

➔ General expectations on classes intended to be used as Web Services

Web Services Classes

Lectures
+addLecture(item:Lecture):void +getLectures():Lecture[] +getLecture(id:integer):Lecture

SystemDictionary
+getSystemNamespaceTranslation(ns:string):array<string,string> +getSystemTranslation(ns:string,token:string,lang:string):string +getSystemCompleteLangs():string[] +getSystemAllLangs():string[] +registerSystemNamespace(ns:string,trans:array<string,array<string,string>>):void

➔ Problem: inputs via SOAP are only plain objects with members, no methods

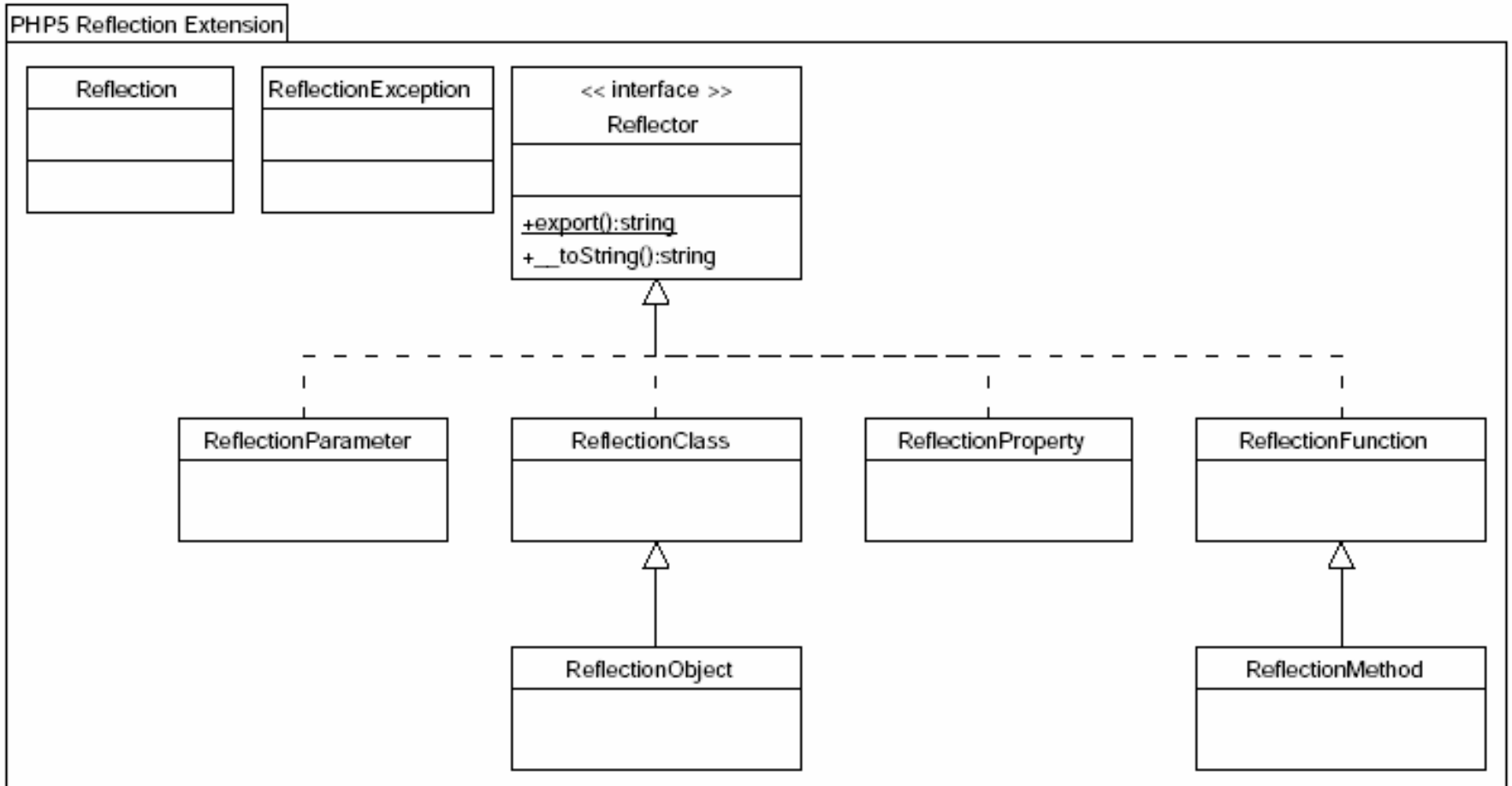
Status

- **Conceptual Design**
- **Security Standards**
 - **WSS Approach**
 - **HTTP based**
- **Generation of WSDL- and XSD-Files**
 - **Extended Reflection API**
- **Example Implementation**
- **Documentation**
 - **Style Guide**

WSD Generator - PHP5 Reflection API

- **PHP5 provides complete reflection API**
 - **reverse-engineer**
 - **Classes**
 - **Interfaces**
 - **Functions**
 - **Methods**
 - **Extensions**
 - **retrieve doc comments**
- **object-oriented extension to Zend Engine**
- **used to gather information for generate WSDL- and XSD-files**

PHP5 Reflection API



Security Aspects

Web Service Facade for PHP5

Andreas Meyer, Sebastian Böttner, Stefan Marr

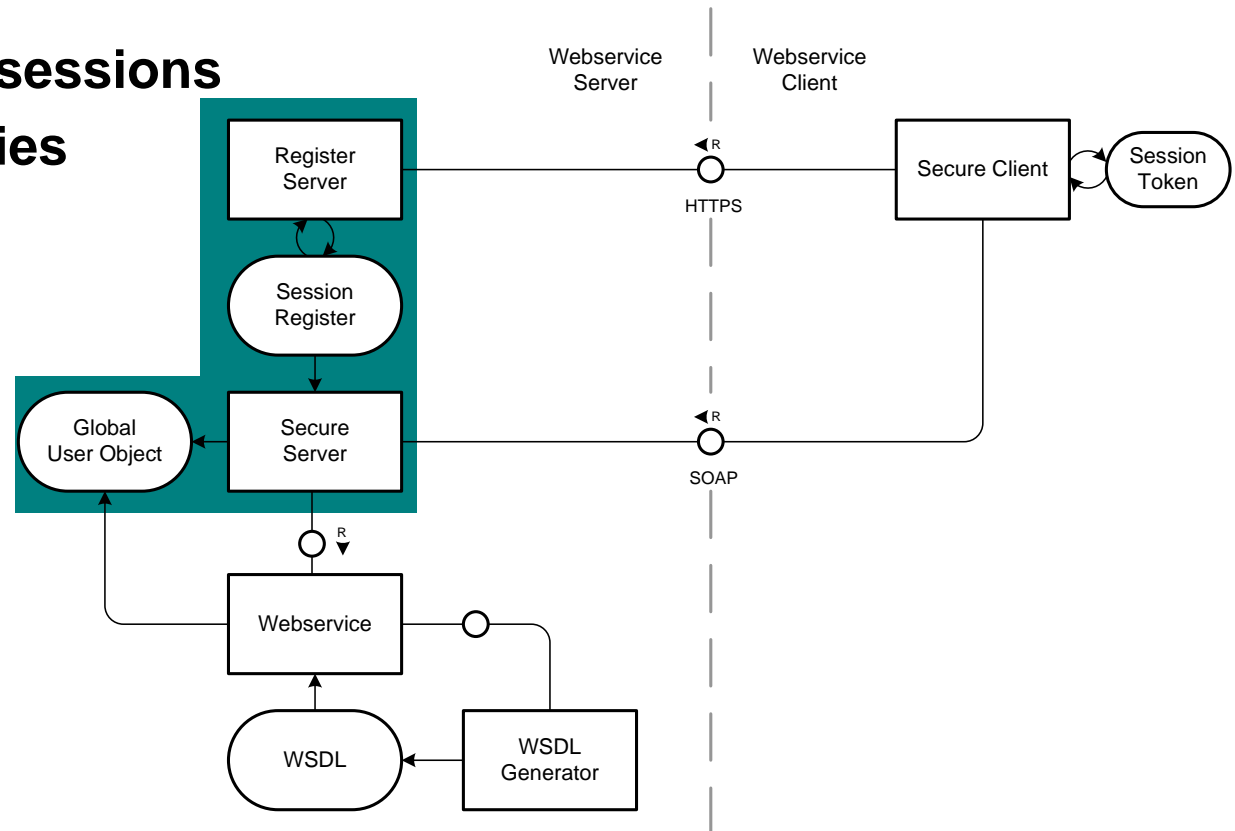


Security: Aims

- ➔ **usage of security aspects independently from WSDL-files**
- ➔ **prevent stateful webservices**
- ➔ **general procedure**
 - **a proxy catches the messages**
 - **controls the security aspects**
 - **forward the messages to a worker**
- ➔ **implemented classes should be unattached by security aspects**
- ➔ **implementation of two different possibilities**
 - **Token Framework**
 - **Username Token Profile 1.0**

➔ General Information

- client connects to the register server and gets a token depending on username and password
- by the use of this token the access to the user's functions is controlled
- usage of PHP sessions
- usage of cookies



➤ Advantages

- usage of existing standards
- stateful Web Services possible

➤ Disadvantages

- plaintext
 - counteractive measures
 - SSL
 - HTTPS
- stateful Web Service

➔ General Information

- implementation of parts of the OASIS Web Services Security (WSS)

- xml syntax:

```
<wsse:Security>
```

```
  <wsse:UsernameToken wsu:Id="our-Example">
```

```
    <wsse:Username> Andreas </wsse:Username>
```

```
    <wsse:Password Type="...#PasswordDigest">
```

```
      weYI3nXd8LjMNVksCKFV8t3rgHh3Rw==
```

```
    </wsse:Password>
```

```
    <wsse:Nonce>
```

```
      WScqanjCEAC4mQoBE07sAQ==
```

```
    </wsse:Nonce>
```

```
    <wsu:Created> 2003-07-16T01:24:32Z </wsu:Created>
```

```
  </wsse:UsernameToken>
```

```
</wsse:Security>
```

- Password_Digest

= Base64 (SHA-1 (nonce + created + password))

➔ security considerations

- the secret is put at the end of the input and not the front
- replay attacks: using message timestamps, nonces and caching
- recommends against replay attacks:
 - reject any UsernameToken using not both nonces and timestamps
 - using timestamp freshness limitation and rejecting all UsernameToken with “stale” timestamps
 - caching nonces for a period of time and rejecting all UsernameToken with already used nonces

➔ Advantages

- open standard (supported by IBM, SUN (java), ...)
- independent of PHP, e.g. other clients with different programming languages can use it
- there are only 'self-writing-alternatives'

➔ Disadvantages

- Password_Digest valid for a specified time-frame
 - counteractive measures: one-time nonce
- Possibly plaintext passwords

Coding Guidelines

Web Service Facade for PHP5

Andreas Meyer, Sebastian Böttner, Stefan Marr



Coding and Style Guidelines

- ➔ **WSDL-files are necessary to define communication between Web Service Client and Server**
 - **Interface specification of Web Service needed**
- ➔ **Documentation is added to compensate the lack of datatype info**
- ➔ **phpDocumentor-Tags:**
 - **existing parsers can be used**
 - **common standard**
- ➔ **Enhanced readability and easier maintenance as a plus**

WSDL Example

```
<wsdl:types>
  <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.tele-task.de/webservices/LectureWebService.wsdl">
    <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    <complexType name="LectureListType">
      <complexContent>
        <restriction base="SOAP-ENC:Array">
          <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="string[]" />
        </restriction>
      </complexContent>
    </complexType>
  </schema>
</wsdl:types>
<message name="getLectures" />
<message name="getLecturesResponse">
  <part name="return" type="tns:LectureListType" />
</message>
<portType name="getLecturesPortType">
  <operation name="getLectures">
    <input message="tns:getLectures" />
    <output message="tns:getLecturesResponse" />
  </operation>
</portType>
```

General Guidelines

- ➔ **One header block comment per file**
- ➔ **One comment per class, method or function**
- ➔ **Short documentation for every variable**
- ➔ **DocComments start with `/**` and end with `*/`, beginning with a description followed by the DocTags**
- ➔ **Maximum of 77 chars per line**
- ➔ **CamelCase, avoid underscores**

Datatype Declaration

➔ Tags needed for WSDL Parser:

- **@return datatype description**
 - States the datatype of the return value and additional information
- **@var datatype description**
 - States the datatype and additional information for variables
- **@param datatype \$paramname description**
 - States the datatype and information for function arguments
- **datatype may be**
 - Integer
 - String
 - Double
 - Boolean
 - AnyClass
 - array of a datatype (`string[]`, `integer[]`, `MyClass[]`, ...)
 - Associative arrays as: `array<datatype,datatype>`
- **No mixed**

Header Block Comments

```
<?php
//*****
//*****
/**
/** class.dbNews.php - used for instancing of dbNewsObjects
/**
/** Projekt: Teletask, Konzepte und Methoden der Web-Programmierung
/**
/** @package dbTables
/** @author Stefan Marr <stefan.marr@hpi.uni-potsdam.de>
/** @author Sebastian Böttner <sebastian.boettner@hpi.uni-potsdam.de>
/** @copyright 2005 by Andreas Meyer, Sebastian Böttner, Stefan Marr
/** @license http://example.org/examplelicense.php Example License
/** @lastchange: 2005-11-30 altered News, phpDocumentor-Tags added
/**
//*****
//*****
```

➔ **Short description**

➔ **Optional long description**

➔ **Project name**

➔ **At least:**

● **@package @author @copyright @license @lastchange**

➔ **Optional:**

● **@deprecated @internal @see @since @uses @version**

Classes and Attributes

```
/** dbNews *****/
/**
 * dbNews - represents and sets elements of the table news
 *
 * Projekt: Teletask, Konzepte und Methodern der Web-Programmierung
 *
 * @package dbTables
 * @author Stefan Marr <stefan.marr@hpi.uni-potsdam.de>
 * @author Sebastian Böttner <sebastian.boettner@hpi.uni-potsdam.de>
 * @copyright 2005 by Andreas Meyer, Sebastian Böttner, Stefan Marr
 * @license http://example.org/examplelicense.php Example License
 */
```

- ➔ **Similar to Header Block Comments (same Tags)**
 - **Optical differences for distinguishing**

```
/**
 * @var string
 */
private $zeugs = 'something important';
```

- ➔ **There must be one comment for each variable**
 - **At least datatype must be present**
 - **Description optional**

Methods and Functions

```
//=====
/**
 * constructor-method of class dbNews
 *
 * This is the constructor used to instanciate dbNews objects in
 * dbMoreThanOneNews. Will be filled with a Data Record from Teletask_news
 *
 * @param integer $id
 * @param array<string,string> $data
 */
public function __construct($id = -1, $data = null) {
    $this->_tableName = news;
    $this->_primaryKey = 'id';
    parent::__construct($id, $data);
}
```

- ➔ **Short description**
- ➔ **Optional long description**
- ➔ **At least @param and @return must be present if existent**
- ➔ **//end of functionName if method spans more than 15 lines**

Control Structures

```
//=====
/**
 * shows how to use brackets, whitespaces and indentations
 */
public function styleExample() {
    if ($a == $b || !($c > $d + $e)) {
        //do something
    } //end if
    elseif ($f) {
        //do something
    } //end elseif
    else {
        //do something
    } //end else

    for|foreach|while (...) {
        //do
    } //end for|foreach|while

    do {

    } while ($g);

    switch ($h) {
        case ...:
            ...;
            break;
        default:
            ...;
    } //end switch
} //end of styleExample
```

➔ **//end of structure comment if structure spans more than 15 lines**

References

- ➔ **[UTP10] Web Services Security - UsernameToken Profile 1.0**
OASIS Standard 200401, March 2004
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>

- ➔ **[WSS11] Web Services Security: SOAP Message Security 1.1**
Working Draft - 07 November 2005
<http://www.oasis-open.org/committees/download.php/15251/oasis-wss-soap-message-security-1.1.pdf>

- ➔ **[PHPMAN] PHP.net Manual**
<http://www.php.net/manual/en/ref.soap.php>
<http://www.php.net/manual/en/language.oop5.reflection.php>

- ➔ **[PEAR] PEAR Coding Standards**
<http://pear.php.net/manual/en/standards.php>

- ➔ **[PHPDOC] phpDocumentor tags How to use tags in DocBlocks**
http://manual.phpdoc.org/HTMLSmartyConverter/HandS/phpDocumentor/tutorial_tags.pkg.html

- ➔ **[XSD] XML Schema Part 2: Datatypes Second Edition**
<http://www.w3.org/TR/xmlschema-2/>

- ➔ **[JAVADOC] How to Write Doc Comments for the Javadoc Tool**
<http://java.sun.com/j2se/javadoc/writingdoccomments/>

- ➔ **[STYLE] Style Guide**
http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_ITS/lecturenotes/webprogrammierung/style_guide/index.html

WebService Facade for PHP5

Q & A

Andreas Meyer, Sebastian Böttner, Stefan Marr

