Web Service Toolkit for PHP5

Free and Open Source Software Conference

| 24. Juni 2006 | Stefan Marr, Falko Menge

- Institute at University of Potsdam
    - □ In Germany between Potsdam and Berlin
- founded in 1998 by Prof. Dr. Hasso Plattner
- IT Systems Engineering
    - □ Bachelor and Master courses
    - □ Software Engineers
   - □ 350 students

    http://www.hpi.uni-potsdam.de

1. Basics on Web Services

2. Web Service Toolkit for PHP5

   ■ Extended Reflection API and Annotations with PHP

   ■ WSDL-Generator and Adapter-Generator

   ■ SOAP-Handler-Chains and WS-Security for PHP5

   ■ RESTful Web Services

3. Administration Tool and Policy PlugIn

4. Live demo of the toolkit

- **Realizes Functionality**
- **Used Technologies**
  - □ SOAP
  - □ WSDL
  - □ UDDI
  - □ HTTP
  - □ REST
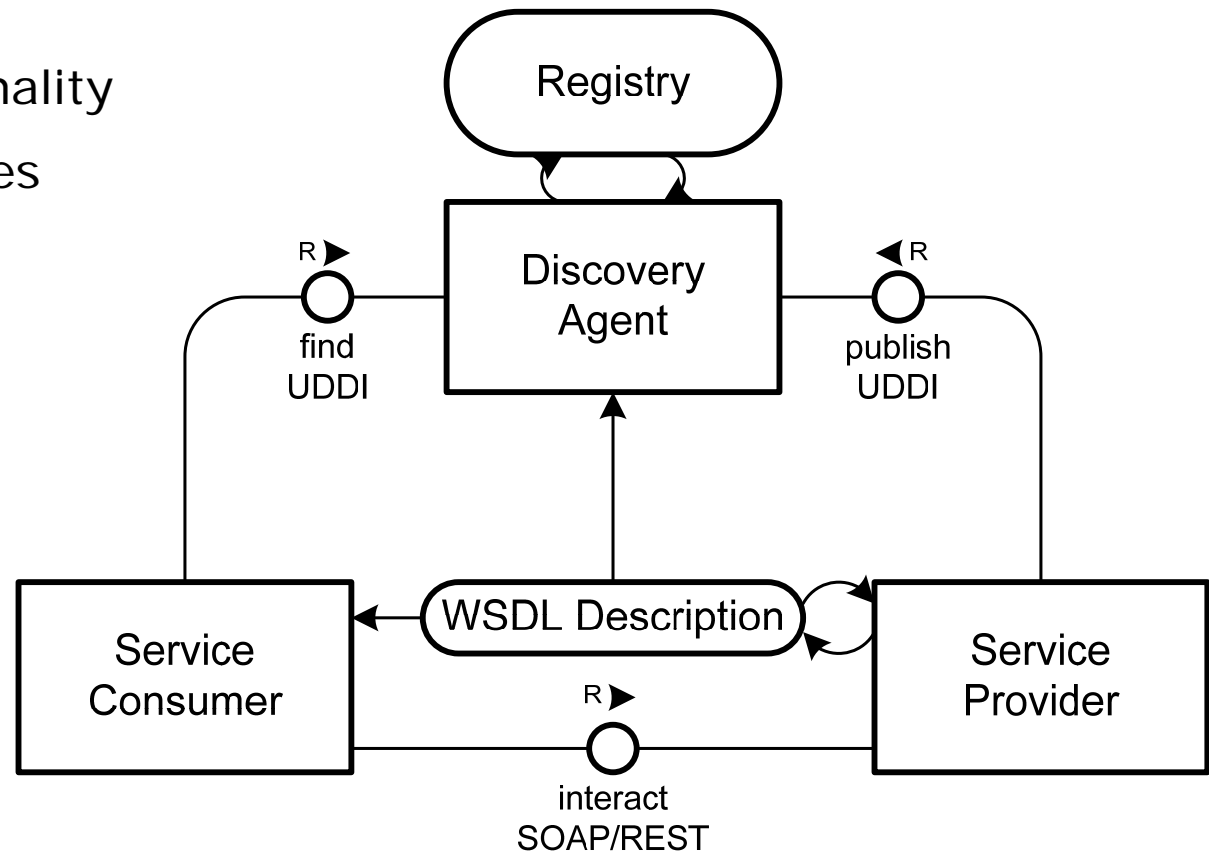
- **B2B Integration, connection to client applications,...**

- ■ Example Web Service
    - □ using SOAP Protocol

- ■ Information about Lectures
    - □ Name of a lecture
    - □ Abstract and VideoStream URL

- ■ Content integration on portals
    - □ For reselling
      by premium offers
    - □ To be used by
      business partners

Lecture
Id : int Name : string Duration : int StreamUrl : string Abstract : string

queryLectures
getAllLectures() : Lecture[] getLecture(in id : int) : Lecture updateOrAddLecture(in lecture : Lecture) : bool getLecturesByAuthor(in author : string) : Lecture[]

Current Situation

Steps for creating web services without tool support

- Create XMLSchema for data types

- Write WSDL service description

- own wrapping for Document/Literal

- Include documentation

- Build SOAP server script

- And you don't got WS-Security support, so far

- PHP 5.1 Extension

- Usage of PHP build-in Parser

- Structural information at runtime
    - Classes
    - Methods
    - PHP Extensions

- But only very few type information
    - PHP is less/dynamic typed language

- Additional information in your source code

- Developed from source code comments
    - e.g. Pre- and Post-conditions given in comments
    - Or conceptual properties

- Support in Microsoft .NET and Java 5 already available
    - But not for PHP

- Possible usage scenarios
    - Marking or configuring of classes/methods…
    - Aspect-Oriented Programming

PHPDoc as basis for annotations

- PHPDoc widespread, used in many projects for documentation
- Therefore good starting point
- Comparable development in Java with XDoclets

```php
/**
 * @myAnnotation paramA paramB
 */
class Lecture {
    /**
     * @return int
     */
    public function getId() {  return intval($this->id);  }
    /**
     * @param string $value
     */
    public function setName($value) {  $this->name = $value;  }
}
```

Extended Reflection API

- Extending the Reflection API of PHP5.1
  - □ by PHP-Classen/PHP-Code

- Adding Annotation-Mechanism
- Typing of
  - □ Parameters
  - □ Return values
  - □ Attributes

Lecture
Id : int Name : string Duration : int StreamUrl : string Abstract : string

- Types as objects for easy usage
  - □ e.g. generate XMLSchema

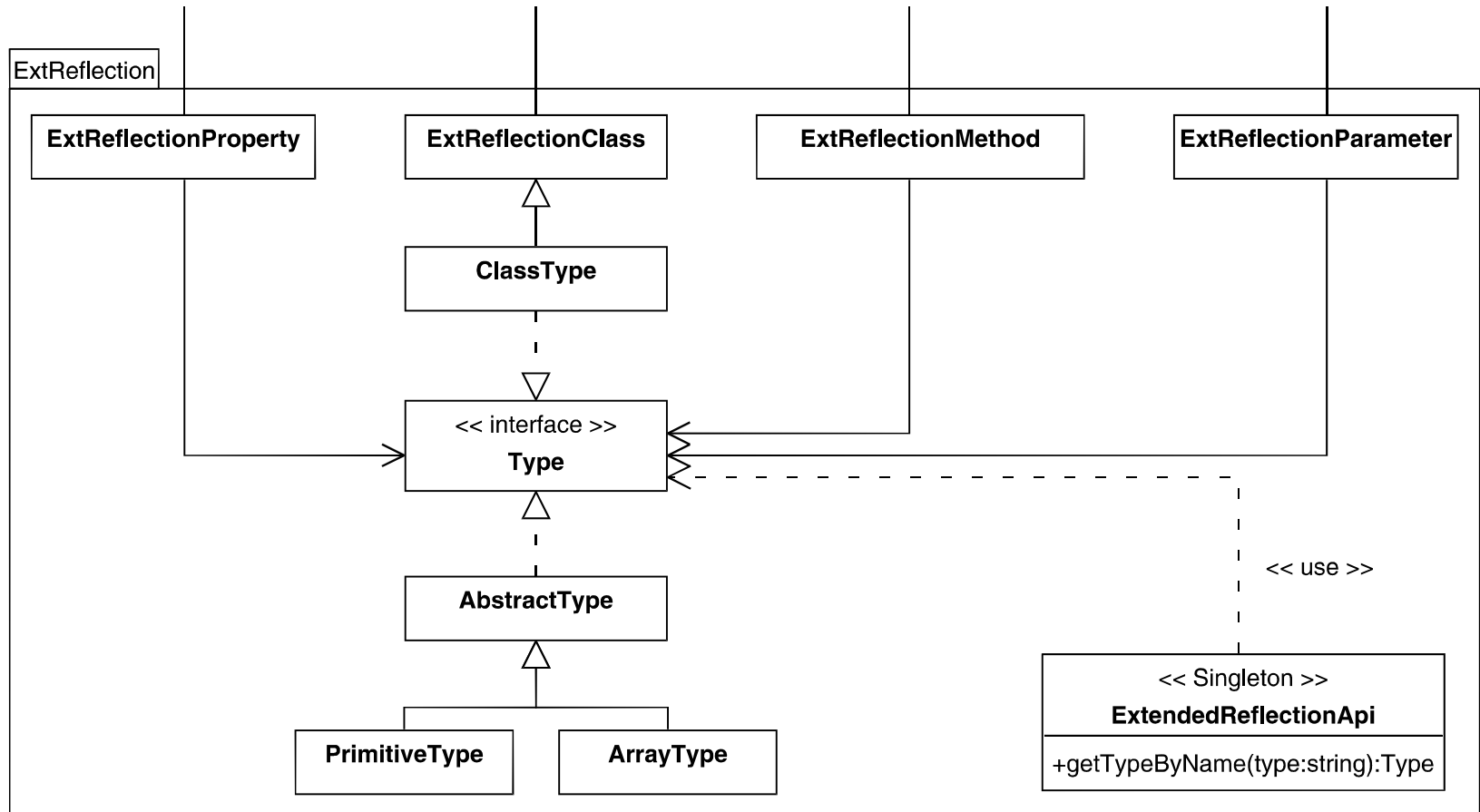| **queryLectures** |
| --- |
| |
| getAllLectures() : Lecture[]
getLecture(in id : int) : Lecture
updateOrAddLecture(in lecture : Lecture) : bool
getLecturesByAuthor(in author : string) : Lecture[] |

The PHP Extension

ExtReflection API

PHP5 Reflection Extension

Reflection   ReflectionException   << interface >> Reflector

ReflectionProperty   ReflectionClass   ReflectionFunction   ReflectionParameter
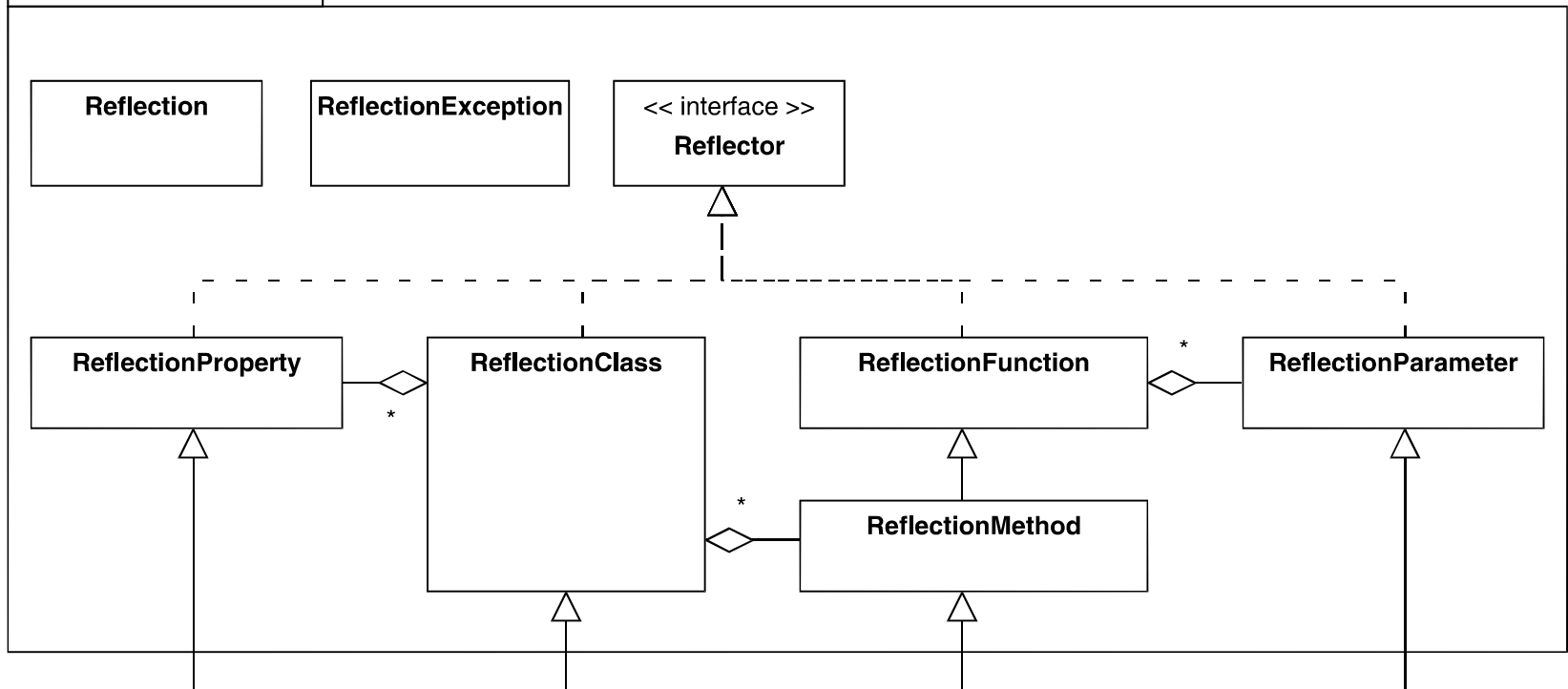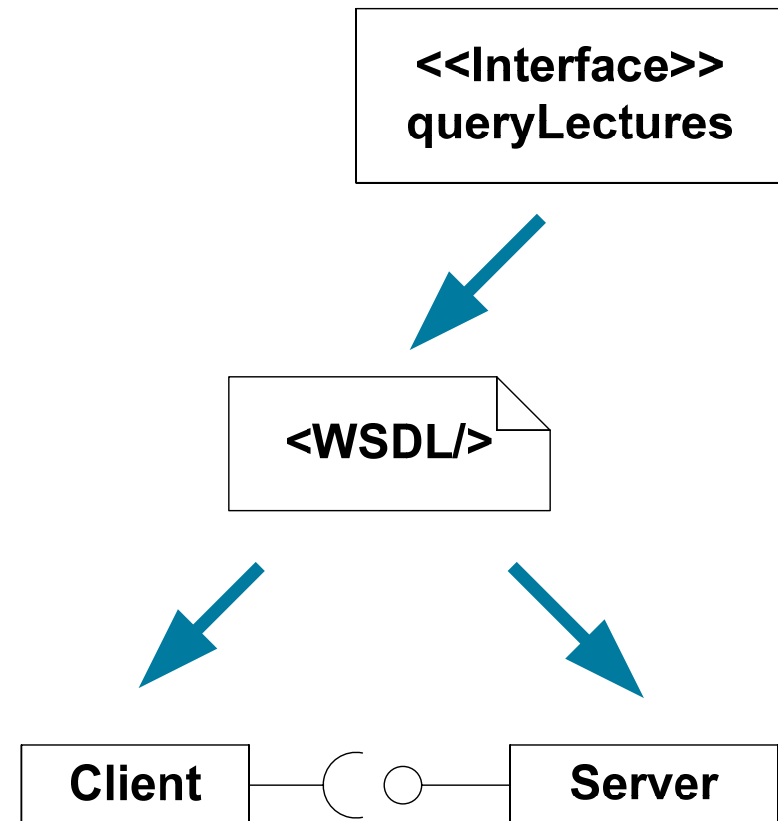
ReflectionMethod

Extended Reflection API

■ Extended with Annotations and a type system

# Web Services Description Language

- Platform independent language for describing interfaces

- Description contains
  - Interface
  - Method signatures
  - Data types

- Used with SOAP

- In WSDL 2.0 additional aspects for usage with REST

**<<Interface>>
queryLectures**

<WSDL/>

Client ── **Server**

WSDL-Generator

- Generates WSDL from PHP classes or collection of functions
- WSDL 1.1
  - conforms to the WS-I Basic Profile by 95%
  - Supports RPC Encoded / Literal
  - And Document Literal / Wrapped
- Implemented as DOMDocument
- Tested with phpt test cases
  - Partially used from SOAP Extension
- Adapter Generator
  - for document-wrapped Binding
  - Classes for Un/wrapping of arguments and return values

queryLectures

WSDLGenerator

<WSDL/>
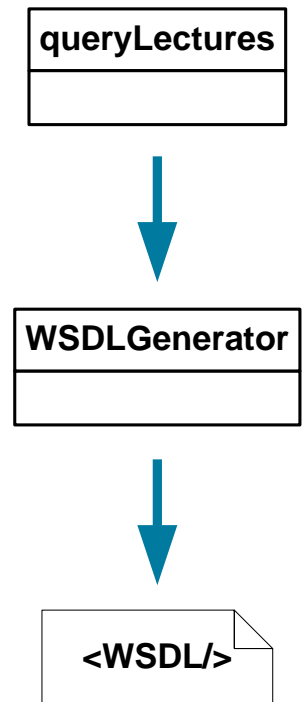
Agenda

1. Allgemeines zu Web Services

2. Web Service Toolkit für PHP5

   ■ Extended Reflection API und Annotations für PHP

   ■ WSDL-Generator und Adapter-Generator

   ■ SOAP-Handler-Chains and WS-Security for PHP5

   ■ RESTful Web Services

3. Administration Tool and Policy PlugIn
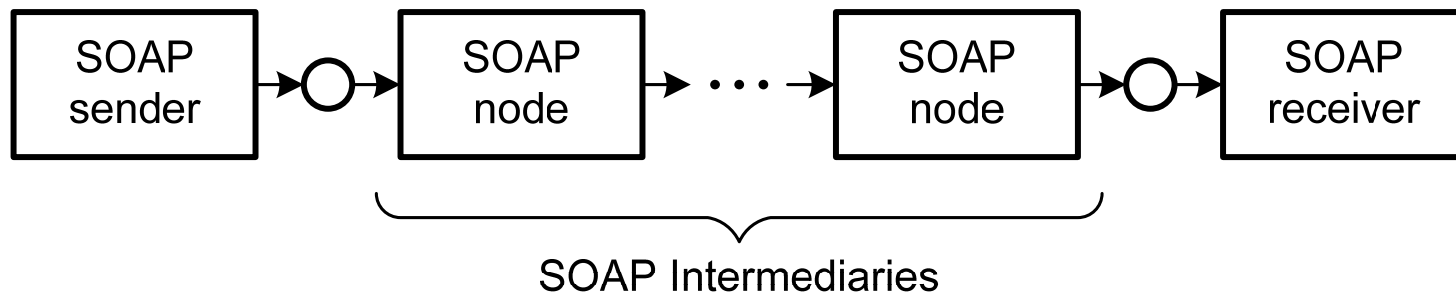
4. Live demo of the toolkit

- Basics are already implemented for PHP
    - SOAP: PHP5 Extension, PEAR SOAP, NuSOAP
    - UDDI: PEAR-Paket
    - WSDL: Generator

- over 20 posibile WS-* Standards for
    - Security
    - Synchronisation
    - Sessioning
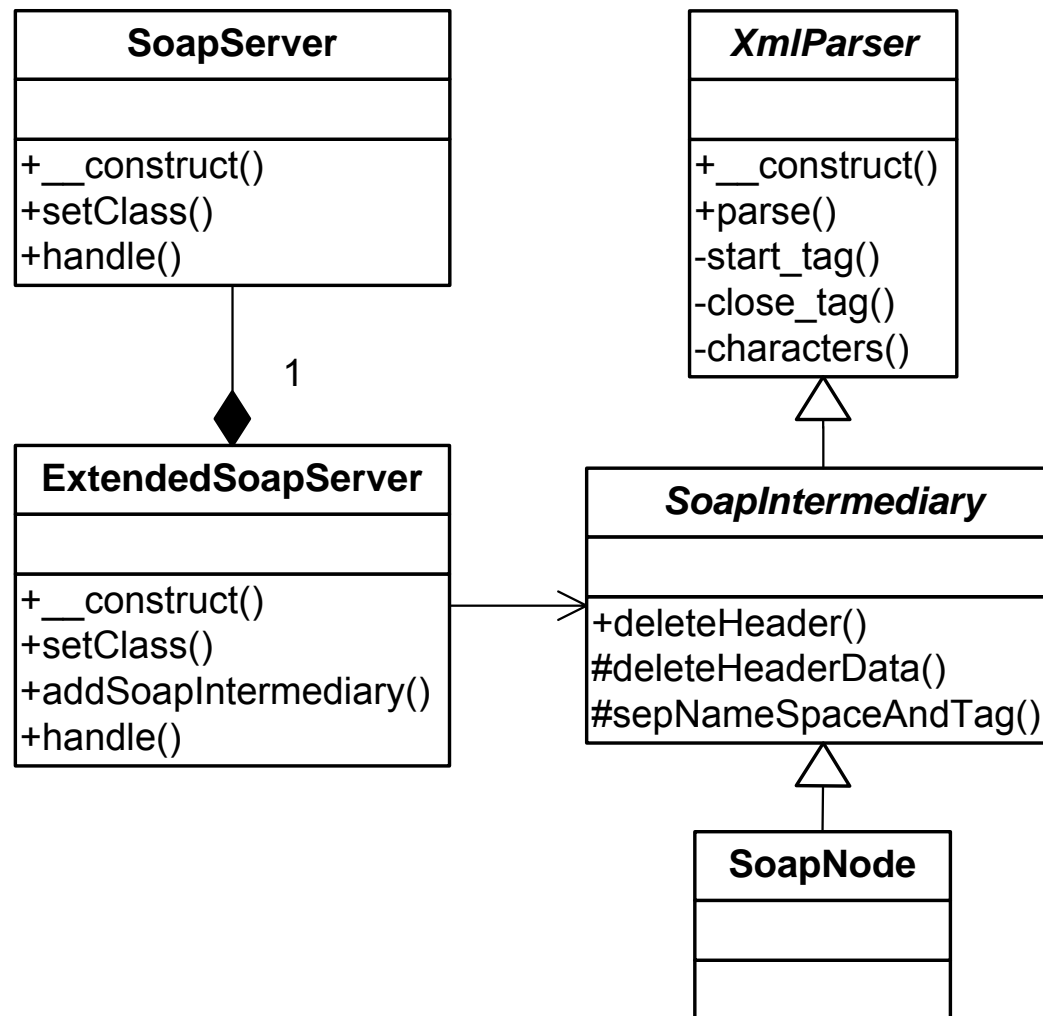- Standards are adding additonal elements to the SOAP header

SOAP-Intermediaries

- SOAP-Standard describes intermediaries

SOAP Intermediaries

- Intermediaries are working on SOAP message before reaching Ultimate Receiver

- additional features indipendent from final Web Service

- Implementation pattern: Handler-Chain

Handler-Chain-Mechanism for PHP5

WS-Security for PHP5

- Security concepts for Web Services:
    - confidentiality   => SSL (SOAP via HTTPS)
    - authentication   => WS-Security + Token Profiles
    - authorization   => Application
- Security components currently left: WS-Security
- WS-Security defines SOAP-Header-Element for security related data
- Different profiles specifies several authentication mechanisms
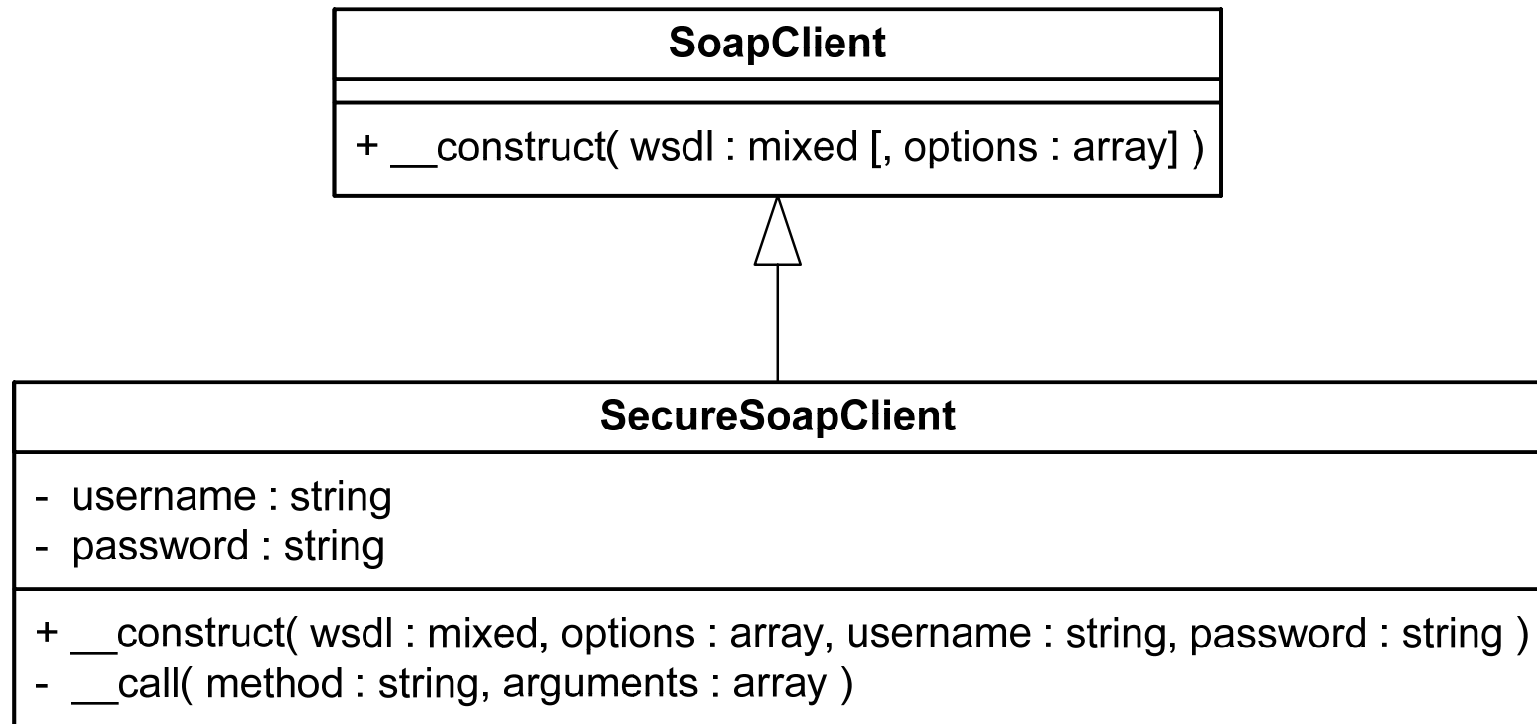- First profile to be used with PHP: Username Token Profile 1.0

Username Token Profile 1.0 (Server)
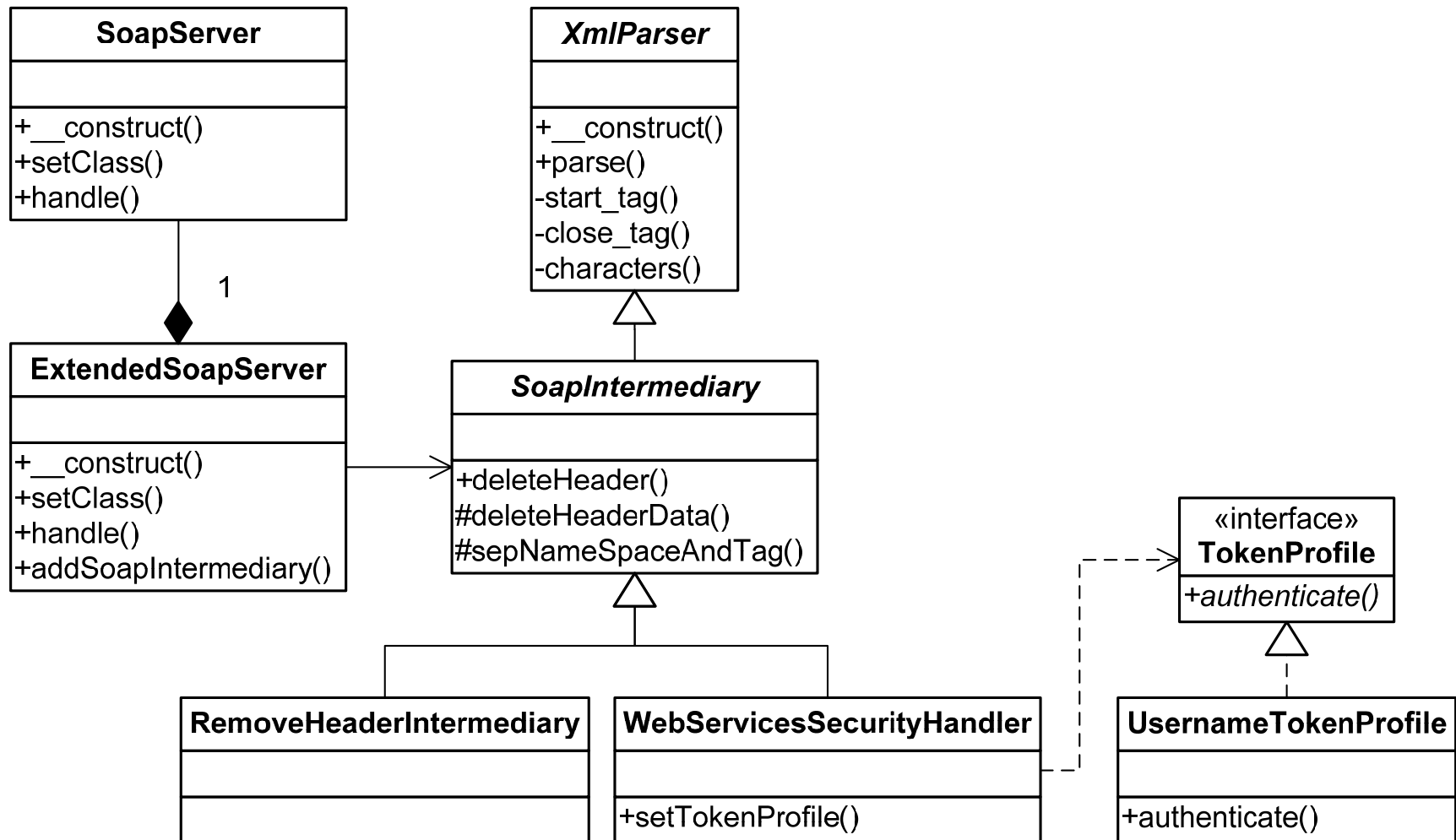
Username Token Profile 1.0 (Client)

```
┌─────────────────────────────────────────────────────────┐
│                      SoapClient                          │
├─────────────────────────────────────────────────────────┤
├─────────────────────────────────────────────────────────┤
│ + __construct( wsdl : mixed [, options : array] )        │
└─────────────────────────────────────────────────────────┘
```

```
┌─────────────────────────────────────────────────────────────────────────────┐
│                          SecureSoapClient                                     │
├─────────────────────────────────────────────────────────────────────────────┤
│ -  username : string                                                          │
│ -  password : string                                                          │
├─────────────────────────────────────────────────────────────────────────────┤
│ + __construct( wsdl : mixed, options : array, username : string, password : string ) │
│ -  __call( method : string, arguments : array )                               │
└─────────────────────────────────────────────────────────────────────────────┘
```

Representational State Transfer

- Alternative to SOAP Web Services

- Resource oriented approach

- HTTP-REST

  □ Just use HTTP Methods

    □ GET, POST, PUT, DELETE

  □ Payload semantic of messages instead of RPC semantic

- Implemented with a **RemoteFacade**

  □ Mapping from URIs to PHP methods

  □ Custom De/Serializer for create PHP objects

  □ REST-Server configuration generated from code annotations

- Security by TLS or DigestAuth (RFC2617)

Administration Tool & Policy Plug-In

- **Administration Tool**
    - automated creation of Web Services from existing applications
    - Classes used as web services identified by annotations
    - Administration via a Web-Frontend
        - select classes and methods
        - generates SOAP- and REST-Server scripts
        - generates WSDL file and Adapter-Classes
        - SOAP-Server optional with WS-Security
- **Policy Plug-In for WSDL-Generator**
    - filters methods, to be hidden from web services
    - Adds comments to WSDL Ports from source code documentation
    - Enables admin to edit published comments
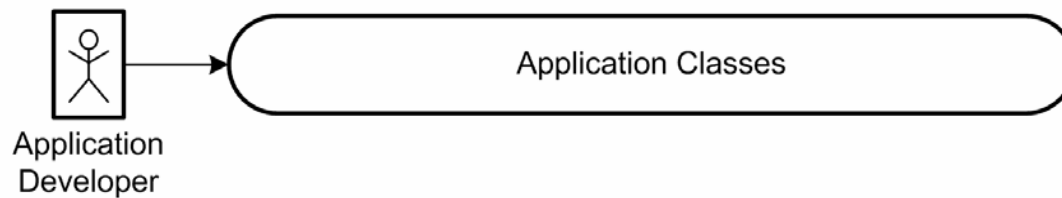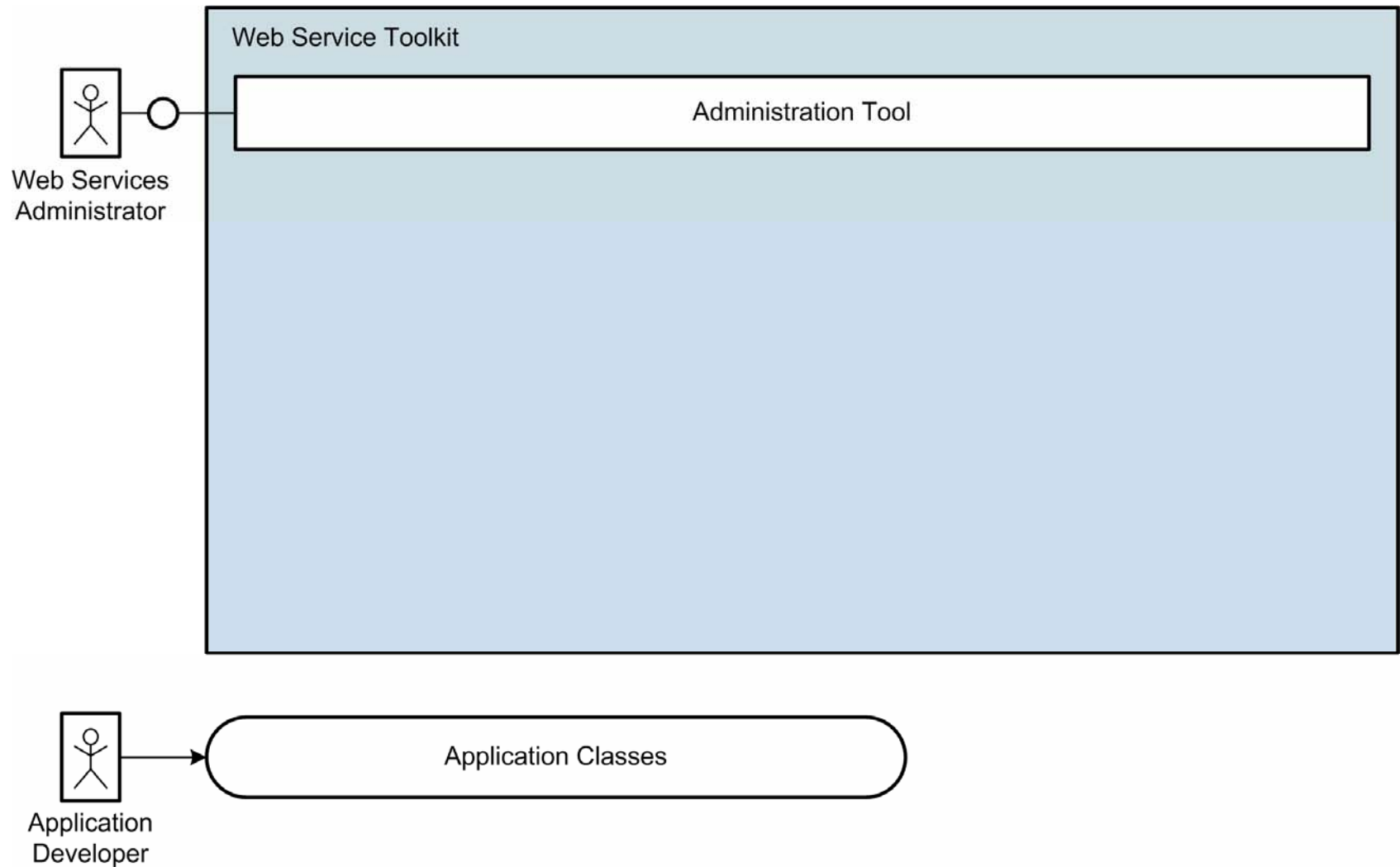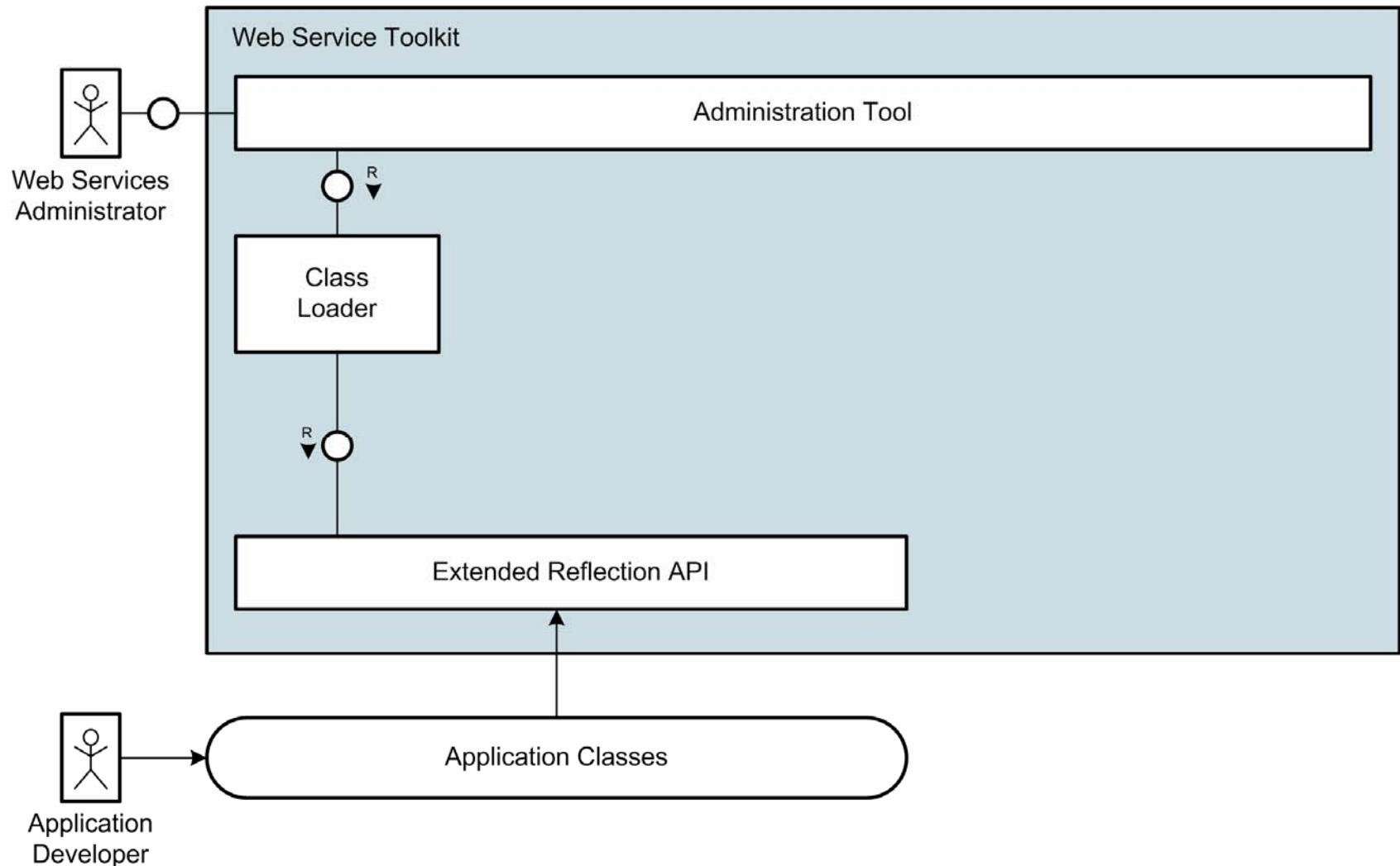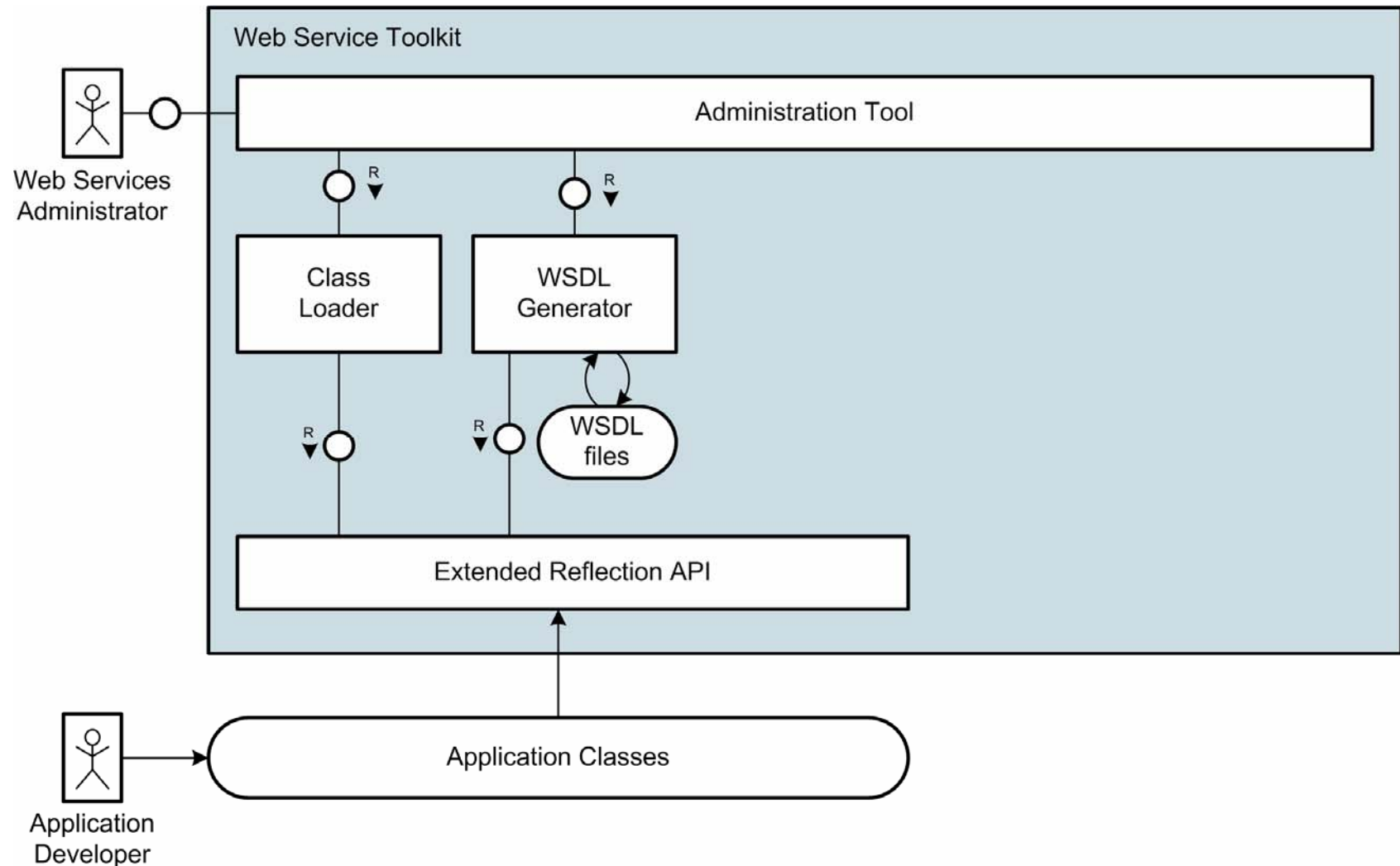
Application Developer → Application Classes

Web Service Toolkit

Administration Tool

Web Services
Administrator

Application Classes

Application
Developer

Livedemo des Toolkits

■ Extended Reflection API with type information

■ PHP with Annotations

■ WS-I conform WSDL-Generator

■ document/literal-Adapter-Generator

■ Handler-Chain-Mechanism for SOAP processing

■ implementation of WS-Security and Username Token Profile

■ RESTful Web Services

■ Administration Tool for comfortable creation und management of SOAP- und REST-Services
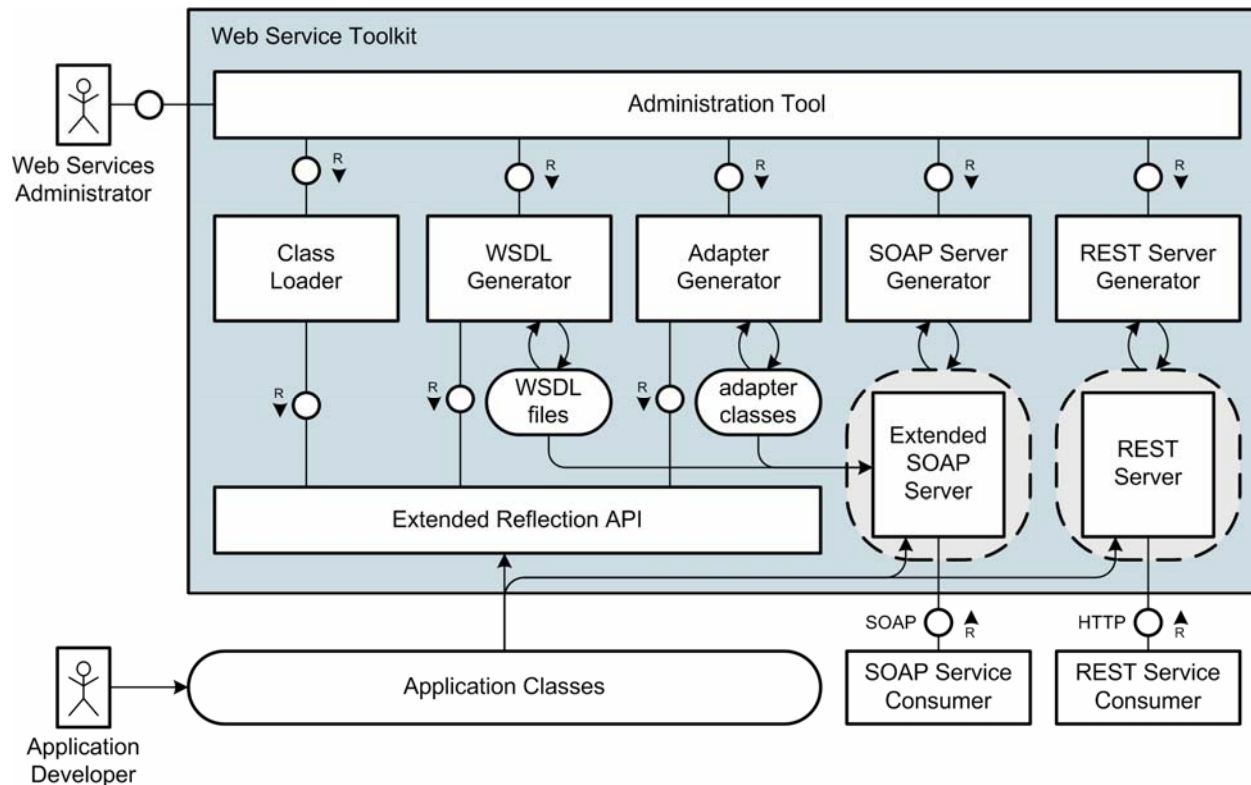
Additional Information and Toolkits download at:
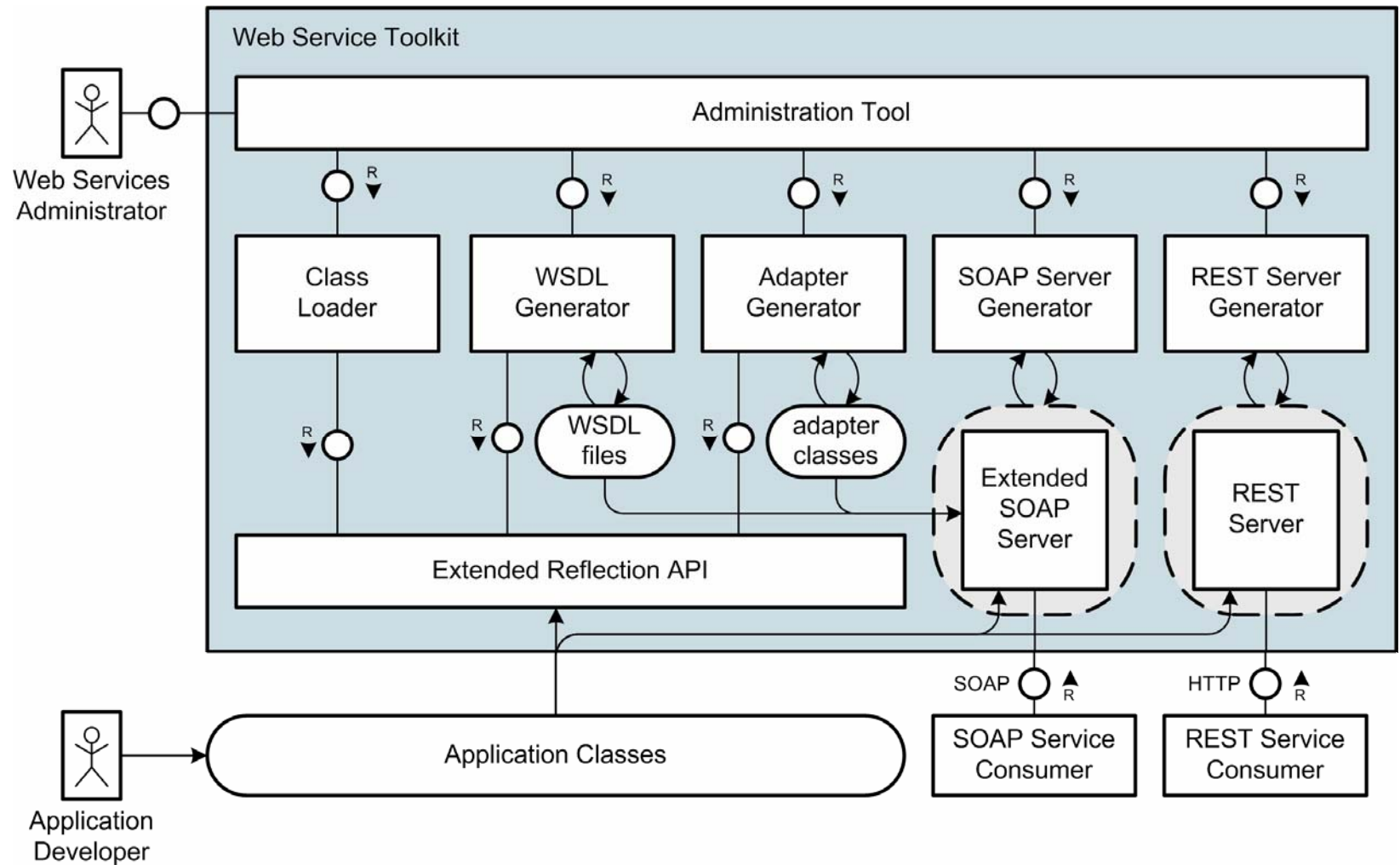
http://toolslave.com/froscon.html

Stefan Marr, Falko Menge | 24. Juni 2006

Web Service Toolkit for PHP5

The Core Web Services Protocol Stack

Discovery	UDDI (Itself a Web Service)
Description	WSDL, WSFL/XLANG, others to come
Access	SOAP, SOAP with attachments, XML-RPC, REST
Transfer	HTTP, SMTP, FTP, others
Transport	TCP/IP, UDP, others

Web Services Description Language (WSDL)

A WSDL document uses the following elements:

- **Types**– a container for data type definitions using some type system (such as XSD).

- **Message**– an abstract, typed definition of the data being communicated.

- **Operation**– an abstract description of an action supported by the service.

- **Port Type**– an abstract set of operations supported by one or more endpoints.

- **Binding**– a concrete protocol and data format specification for a particular port type.

- **Port**– a single endpoint defined as a combination of a binding and a network address.

- **Service**– a collection of related endpoints

Source: W3C.org

Adapter-Generator

```
/**
 * @param int $id
 * @return Lecture
 */
public function getLecture($id) {…}
```



```
public function getLecture($param) {
    return array("Lecture"
        => $this->target->getLecture($param->id));
}
```

```
<wsse:Security>

    <wsse:UsernameToken>

        <wsse:Username>Stefan</wsse:Username>

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

            weYI3nXd8LjMNVksCKFV8t3rgHh3Rw==

        </wsse:Password>

        <wsse:Nonce>WScqanjCEAC4mQoBE07sAQ==</wsse:Nonce>

        <wsu:Created>2006-06-24T11:00:00Z</wsu:Created>

    </wsse:UsernameToken>

</wsse:Security>
```
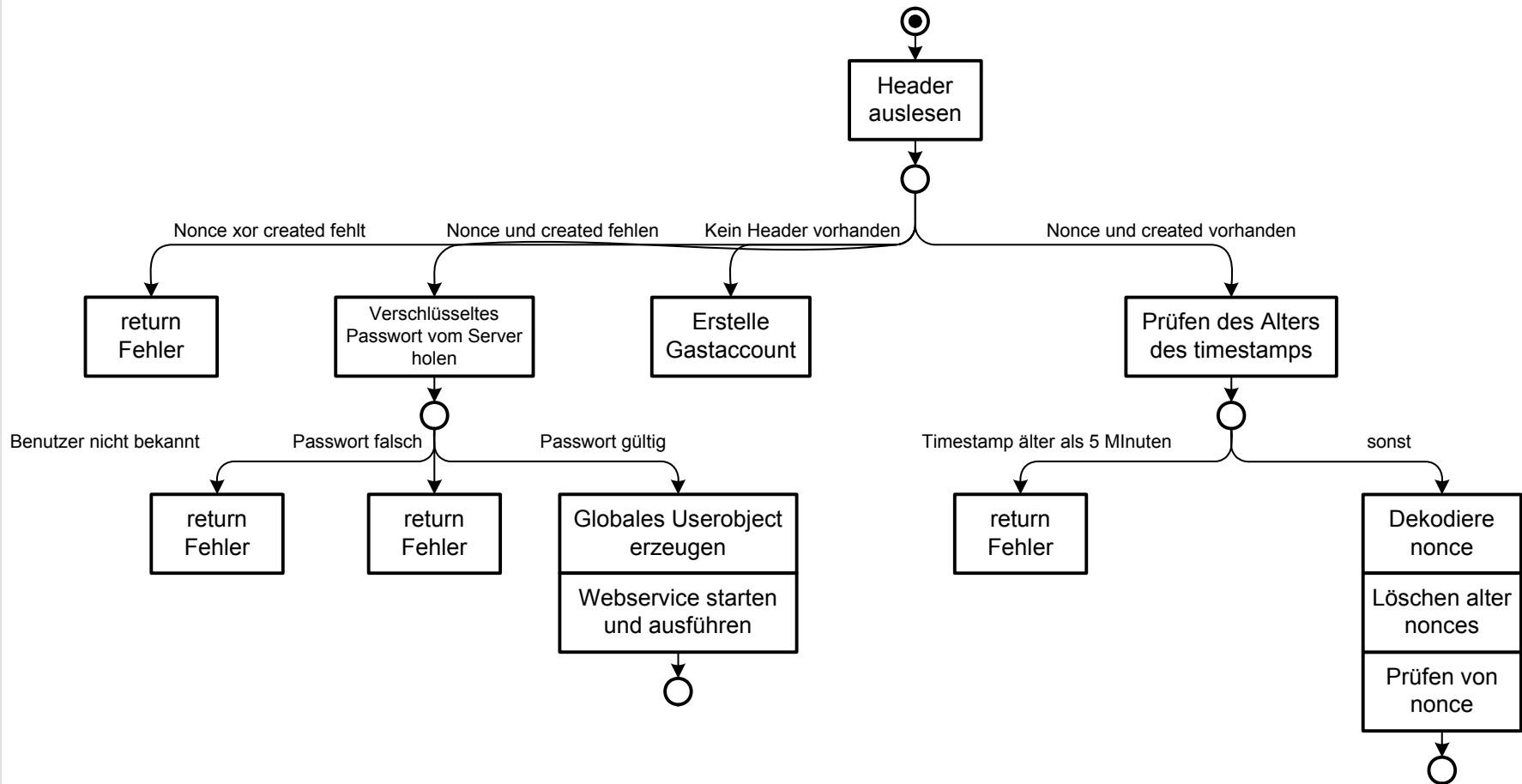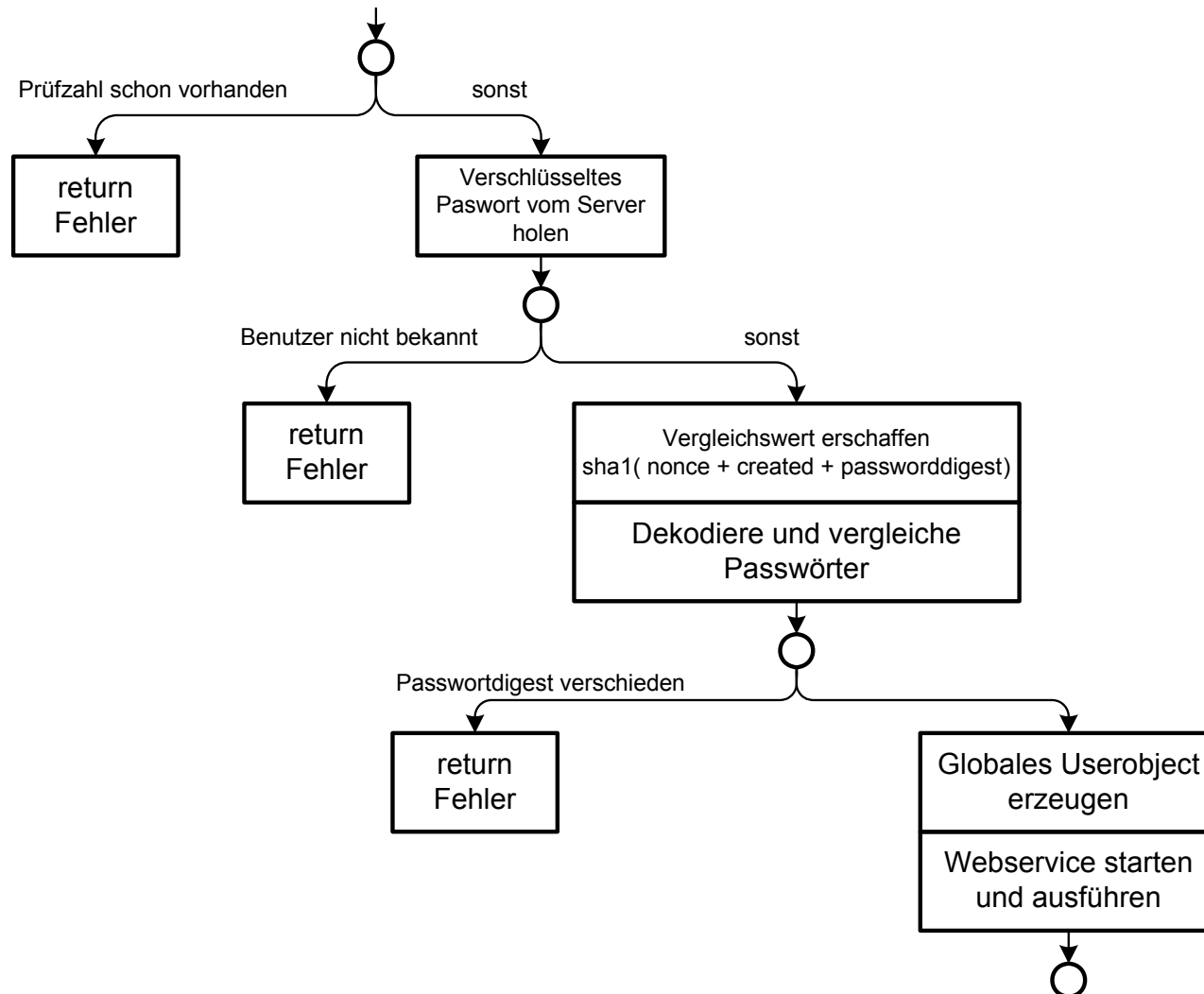
Prüfzahl schon vorhanden — sonst

return Fehler

Verschlüsseltes Paswort vom Server holen

Benutzer nicht bekannt — sonst

return Fehler

Vergleichswert erschaffen
sha1(nonce + created + passworddigest)

Dekodiere und vergleiche Passwörter

Passwortdigest verschieden

return Fehler

Globales Userobject erzeugen

Webservice starten und ausführen

White Papers

[1] Einführung in XML Web Services, Oktober 2005

[2] Web Services Facade for PHP5 – Konzeption, Januar 2006

[3] Web Service Toolkit für PHP5, März 2006
      zum Download unter http://toolslave.com/froscon.html

Weitere Quellen

[4] RESTful Web Services, Januar 2006
      http://www.stefan-marr.de/artikel/restful-web-services.html

[5] SOAP Version 1.2 specification, W3C Recommendation World
      Wide Web Consortium (W3C), Juni 2003. http://www.w3.org/TR/soap/

[6] Web Services Description Language (WSDL) 1.1, W3C Note World
      Wide Web Consortium (W3C), März 2001. http://www.w3.org/TR/wsdl/

[7] Username Token Profile 1.0, OASIS, Januar 2004
      http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf