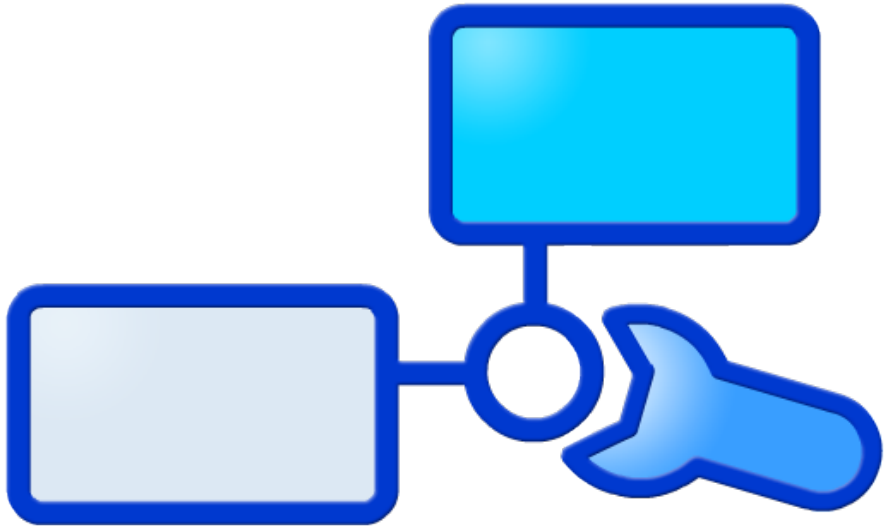
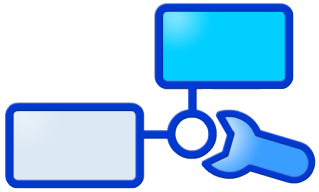


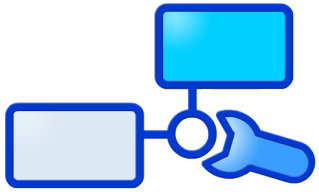
InstantSVC

The PHP Web Services Builder

| 4 September 2007 | Falko Menge

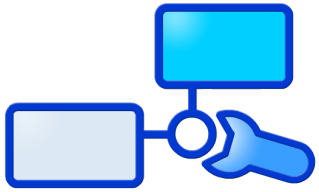


Web Services?



Agenda

- Motivation
- Brief Introduction to Web Service Technologies
- Web Services with PHP5
- Core Components of InstantSVC
  - Extended Reflection API and Annotations for PHP
  - WSDL Generator and Adapter Generator
  - SOAP Handler Chains and WS-Security for PHP5
  - RESTful Web Services
- Administration Front-End
- Live Demo



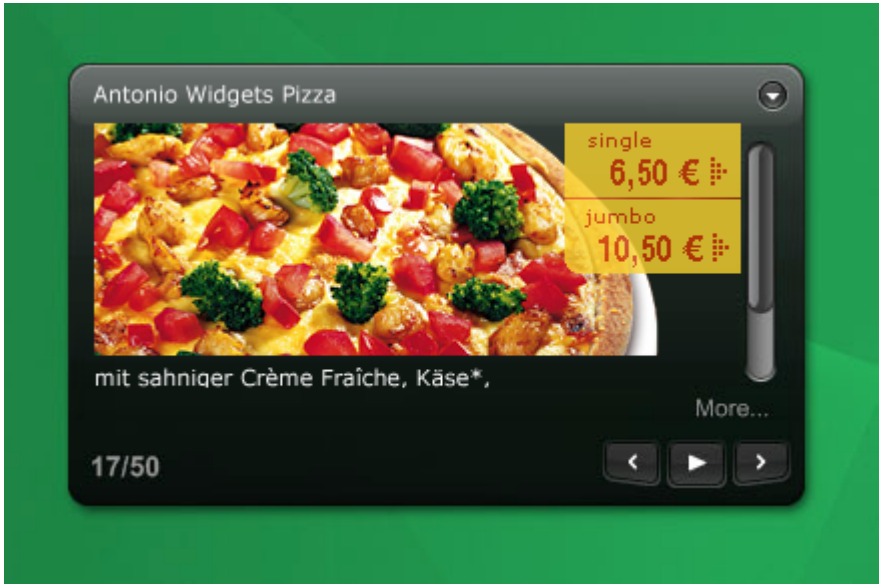
Services?

- Blog with Web 2.0 stuff
  - Feeds, Publishing API, User Contribution, Single-Sign On,…
- Shops
  - Catalog of Goods, Buying API, Availability Checks
  - Intention: Close the Media Gap
- Enterprise Applications
  - Finance, Warehouse Management, Human Resources,…
- Services provided by specialists
  - Used to build applications upon

Services: For Whom?

Enterprise People

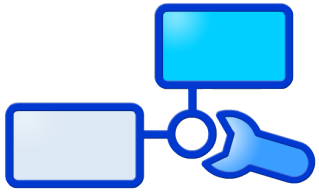
- Portals and Mesh-Ups
  - Personalization



Innovators

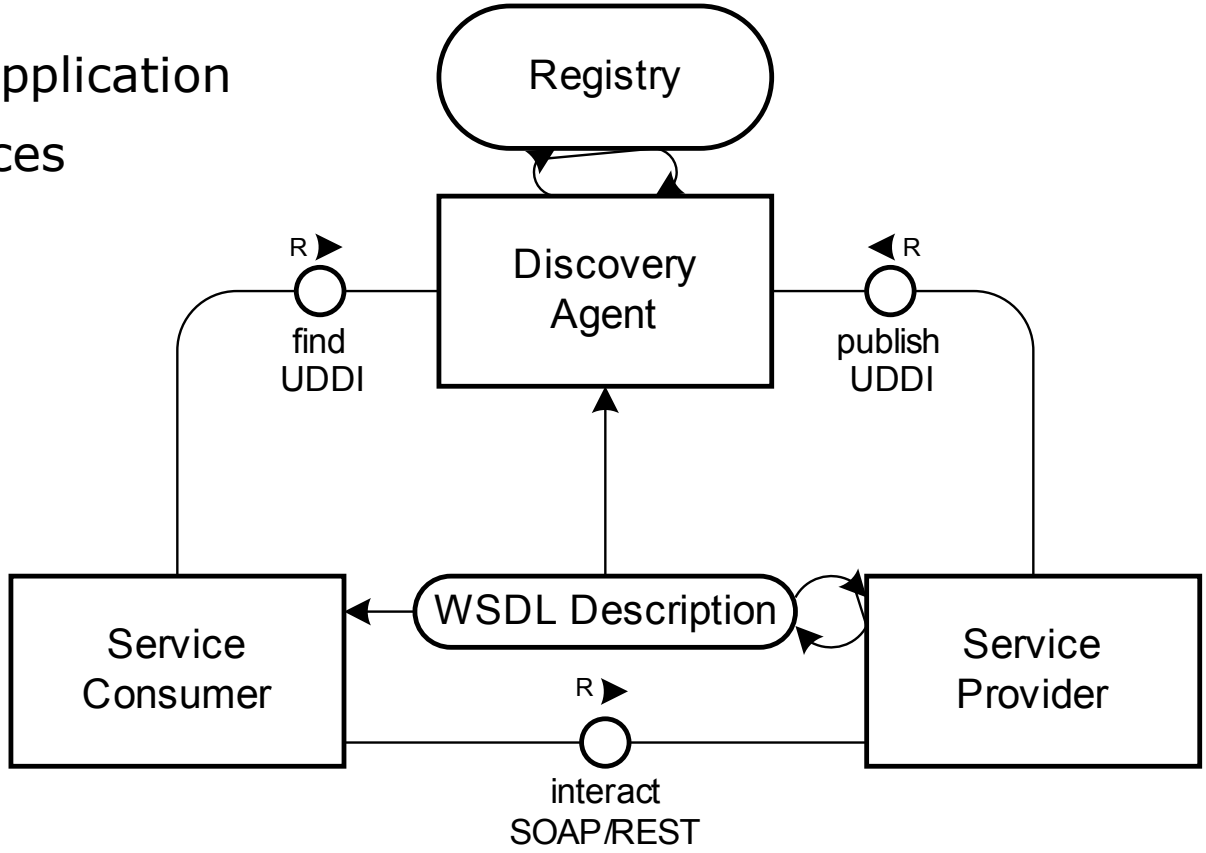
- It is cool
- Everybody wanne have it
- Is it Flickr, del.icio.us, Digg, Google Maps, Ebay?

- Likely, it’s not!
- You did not imagine what's possible with your services
- They do!

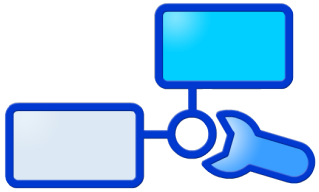


Web Services

- Functions of an application exposed as services
- Technologies:
  - HTTP
  - SOAP
  - WSDL
  - UDDI
- Alternative:
  - REST

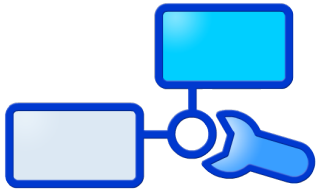


- Powerful concept for cross-platform integration

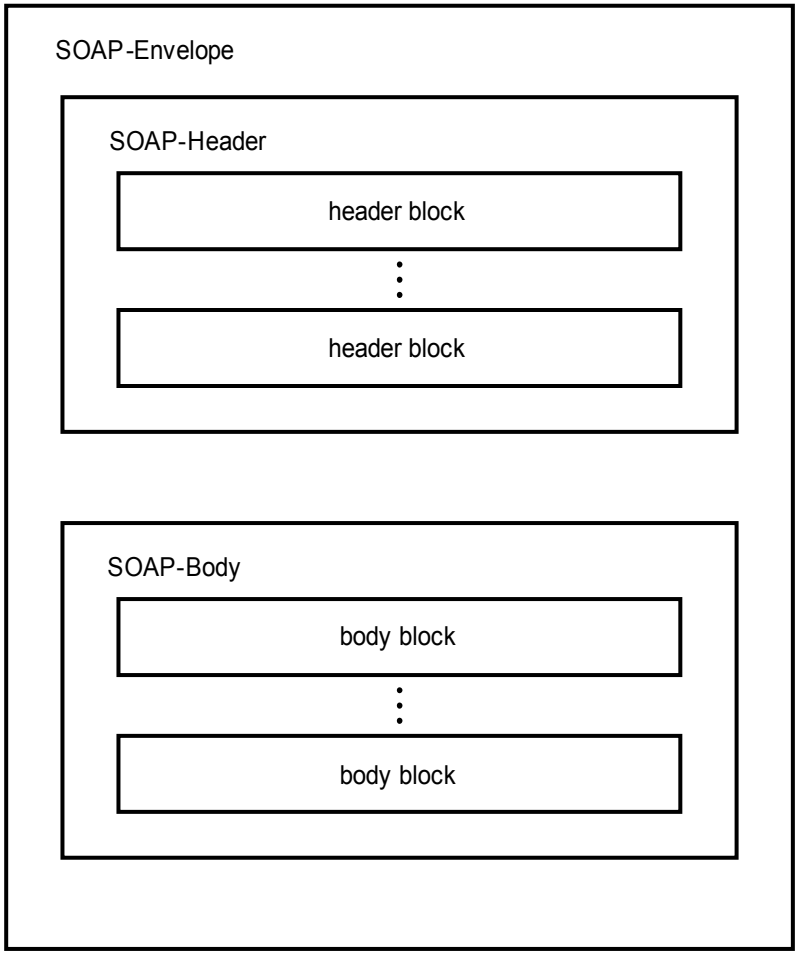


The Core Web Services Protocol Stack

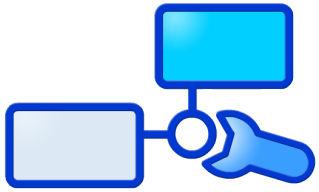
Discovery	UDDI (Itself a Web Service)
Description	WSDL, WSFL/XLANG, others to come
Access	SOAP, SOAP with Attachments, XML-RPC, REST
Transfer	HTTP, SMTP, FTP, others
Transport	TCP/IP, UDP, others



SOAP – The Messaging Protocol




```
<?xml version="1.0" ?>
<Envelope>
        <Header>
                ...
        </Header>
        <Body>
                ...
        </Body>
</Envelope>
```



Web Services Description Language (WSDL)

```
<definitions>
    <types />
    <message />
    <portType>
        <operation />
    </portType>
    <binding />
    <service>
        <port />
    </service>
</definitions>
```

→ Container for data type definitions using some type system (such as XML Schema)

→ Abstract, typed definition of the data being communicated

→ Abstract set of operations

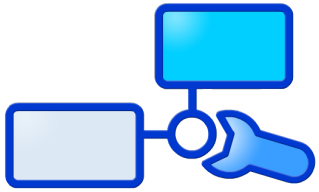
→ Abstract description of an action supported by the service

→ Concrete protocol and data format specification for a particular port type

→ Collection of related endpoints

→ Single endpoint defined as a combination of a binding and a network address

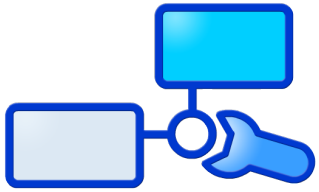
source: W3C



Web Services with PHP5

Steps for creating Web Services using the SOAP extension of PHP5:

- Create XMLSchema for data types
- Write WSDL service description
- Own wrapping for Document/Literal
- Include documentation
- Build SOAP server script
- No support for additional WS-* standards (e.g. WS-Security)

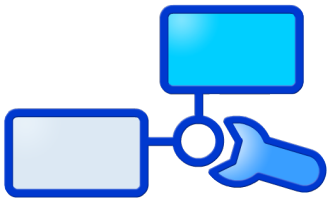


The Vision

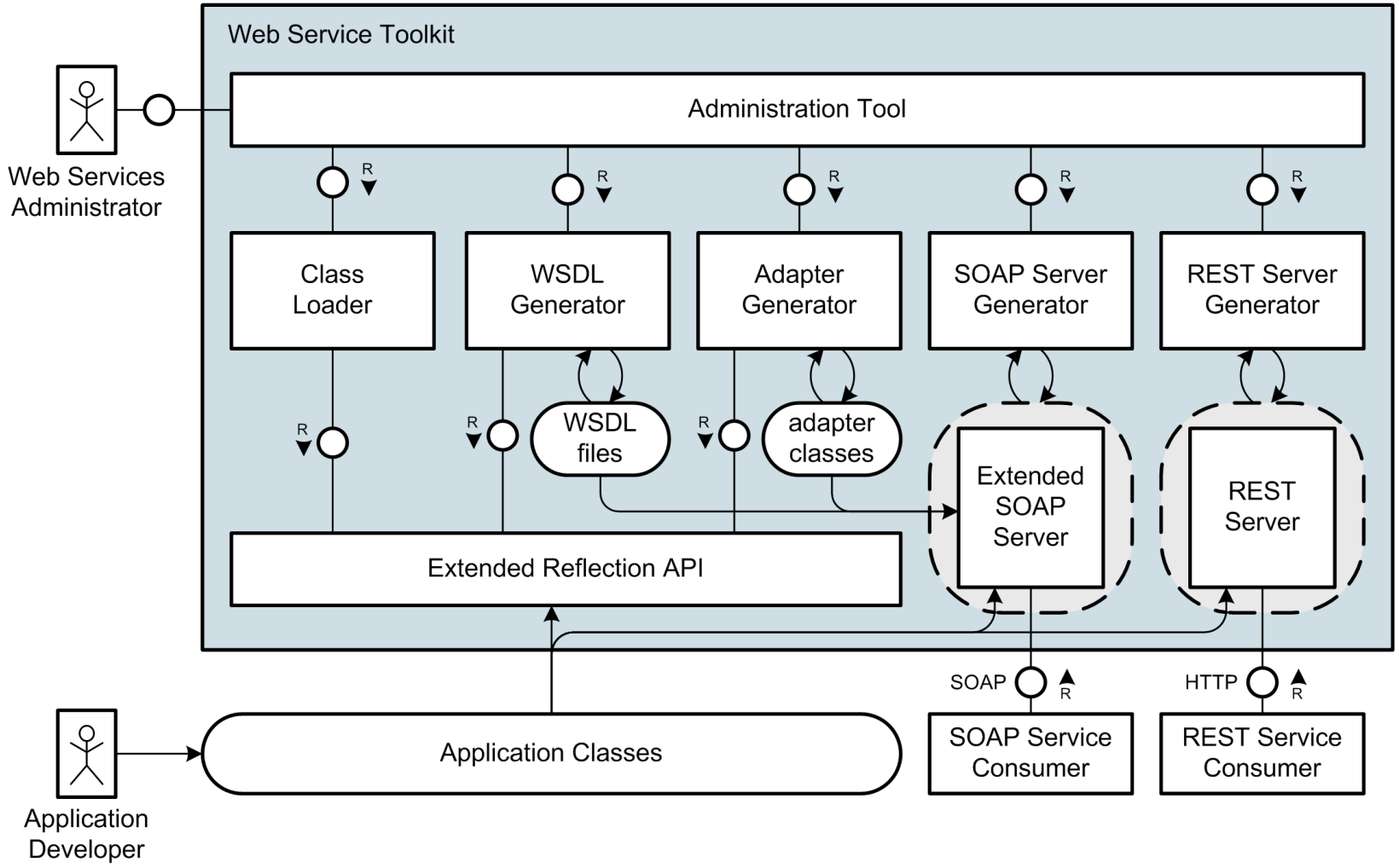
- Consuming Web Services already supported by tools
- Providing Web Services is still to much work

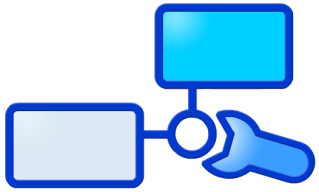
- On other platforms like Java EE or .NET:
  - Extensive tool support
  - Annotations
  - Deployment to Application Servers

- Vision:
  - Generation of Web Services for existing applications
  - Configuration instead of programming
  - Complete automation of the process



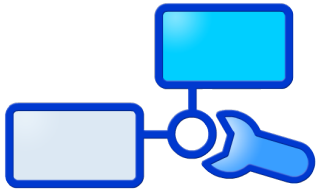
InstantSVC
The PHP Web Services Builder





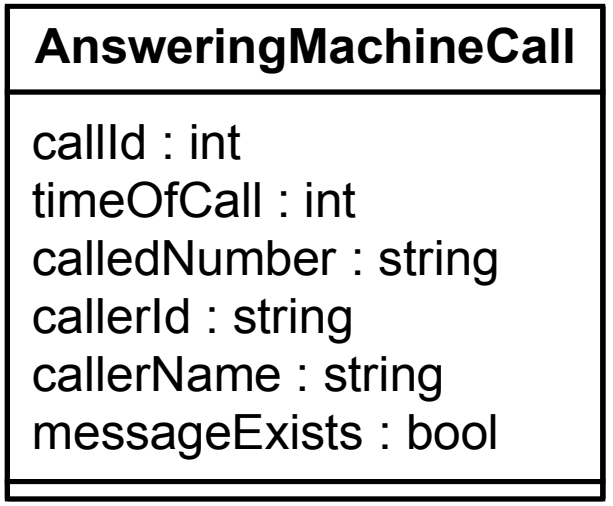
About the Project

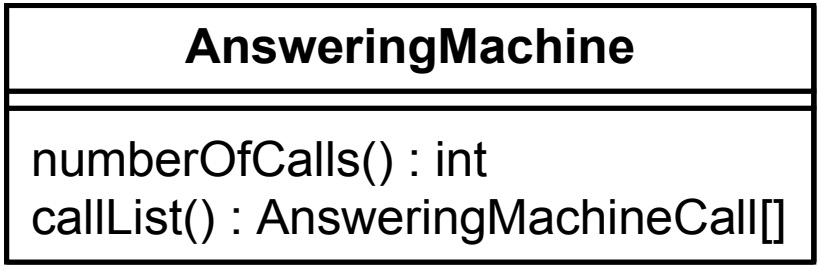
- Base project developed by 6 HPI students since october 2005
  - G. Gabrysiak, Ch. Hartmann, M. Perscheid, M. Sprengel
  - Today here: Stefan Marr and Falko Menge
- Project presented at the FrOSCon 2006
- Open steps: contribute base to eZ Components
- Additional work
  - Access Control in Service Oriented Architectures
  - Implementation of Task-Role Base Access Control based on ServiceMix/Java and InstantSVC

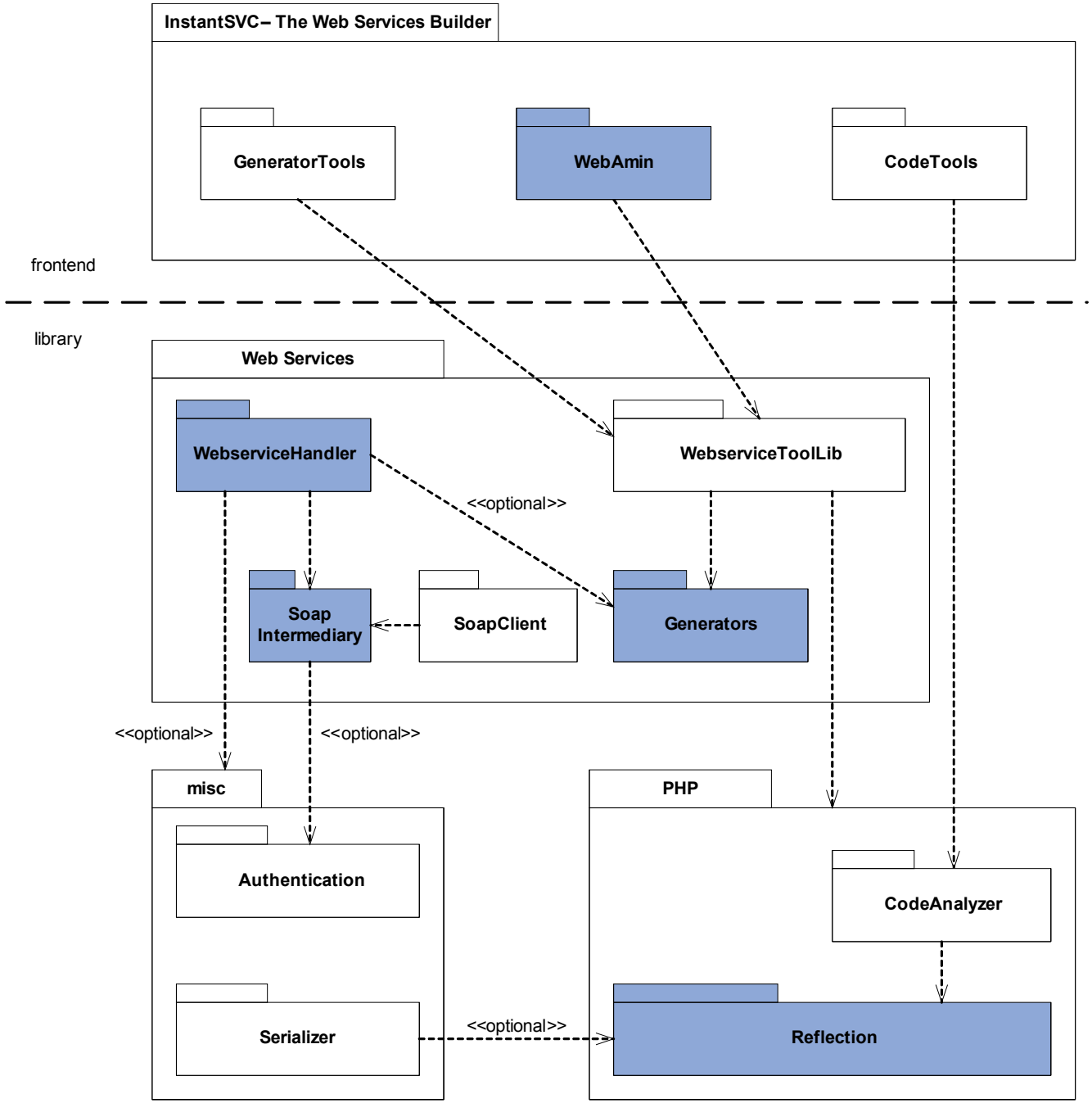


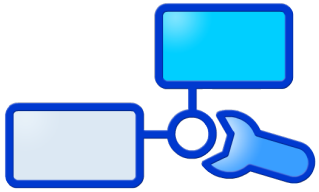
Running Example

- Application: Answering Machine
  - Number of Calls
  - List of Calls
- Example Web Service
  - using SOAP Protocol



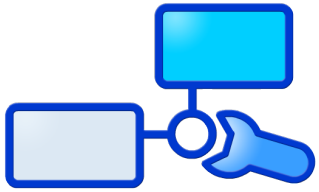



InstantSVC– The Web Services Builder
GeneratorTools
WebAmin
CodeTools
frontend
library
Web Services
WebserviceHandler
WebserviceToolLib
<<optional>>
Soap Intermediary
SoapClient
Generators
<<optional>>
<<optional>>
misc
PHP
Authentication
CodeAnalyzer
Serializer
<<optional>>
Reflection



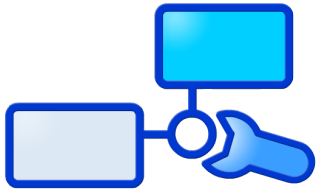
Reflection API

- PHP 5.1 Extension
- Usage of PHP build-in Parser
- Structural Information at Runtime
  - Classes
  - Methods
  - PHP Extensions
- But only very little Information about Types
  - PHP is a dynamically typed Language



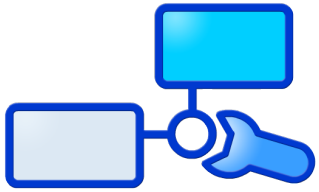
Annotations

- Additional Information in the Source Code
- Developed from Source Code Comments like:
  - Pre and Post Conditions given in Comments
  - Conceptual Properties
- Support in .NET and Java 5 already available
  - But not in PHP
- Possible Usage Scenarios
  - Marking or Configuring of Classes/Methods…
  - Aspect-Oriented Programming



Extended Reflection API

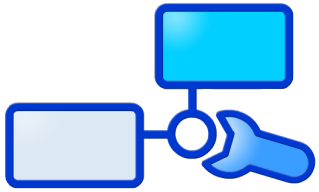
- Extends the Reflection API of PHP 5.1
  - Implemented in PHP
- Adding Annotation Mechanism
- Typing
  - Parameters
  - Return Values
  - Attributes
- Types represented as Objects for easy Usage
  - XMLSchema for Types



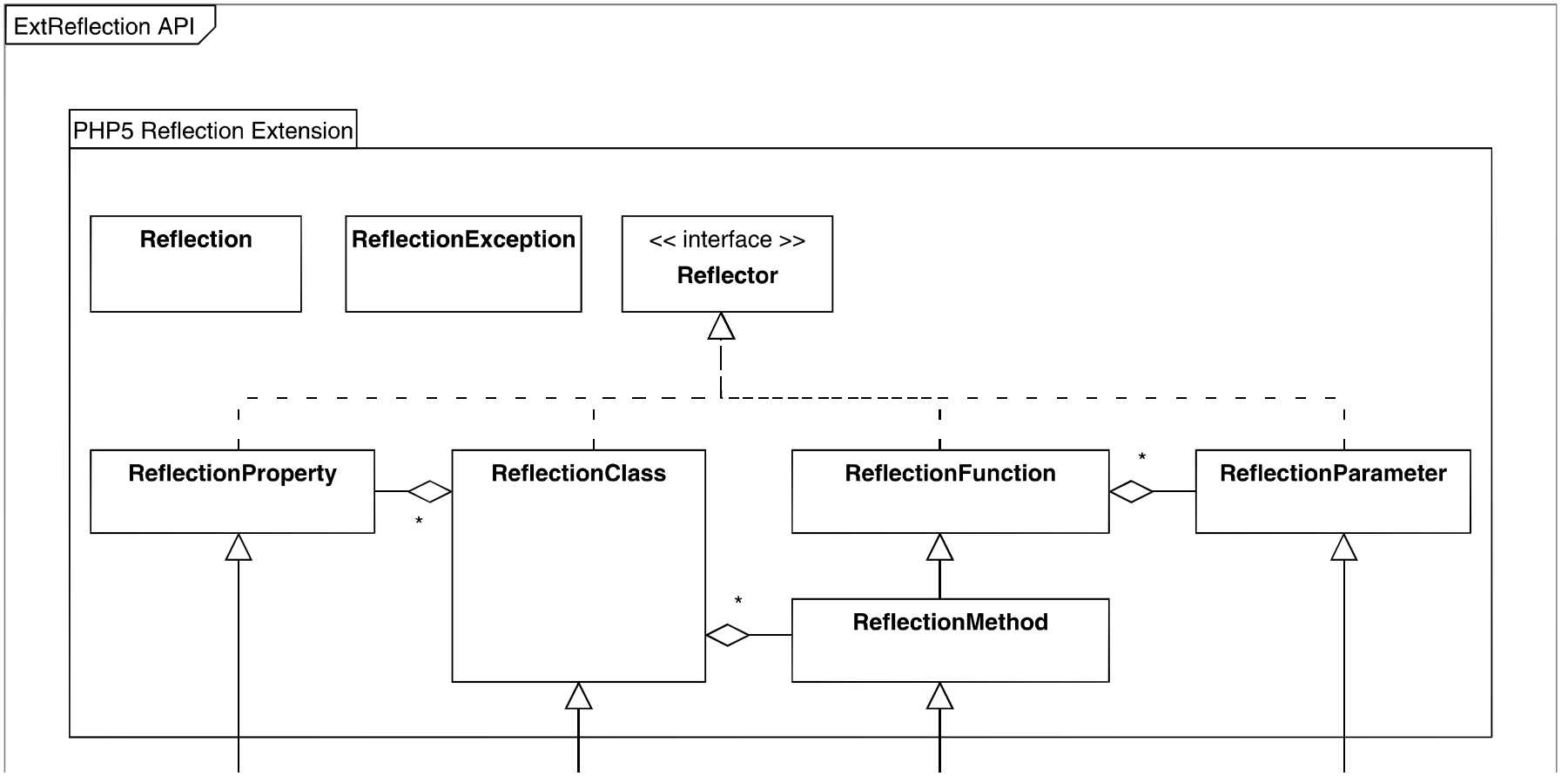
Annotations based on PHPDoc

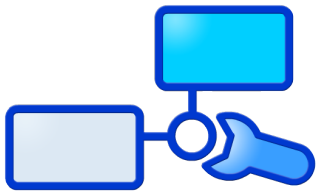
- PHPDoc widely used in many Projects for Documentation
- Good Starting Point for Annotations
- Comparable Development in Java with XDoclet

```
/**
* @myAnnotation paramA paramB
*/
class AnsweringMachineCall {
   /**
    * @return int
    */
   public function getCallId() {  return intval($this->callId);  }
   /**
    * @param string $value
    */
   public function setCallerName($value) {  $this->callerName = $value;  }
}
```

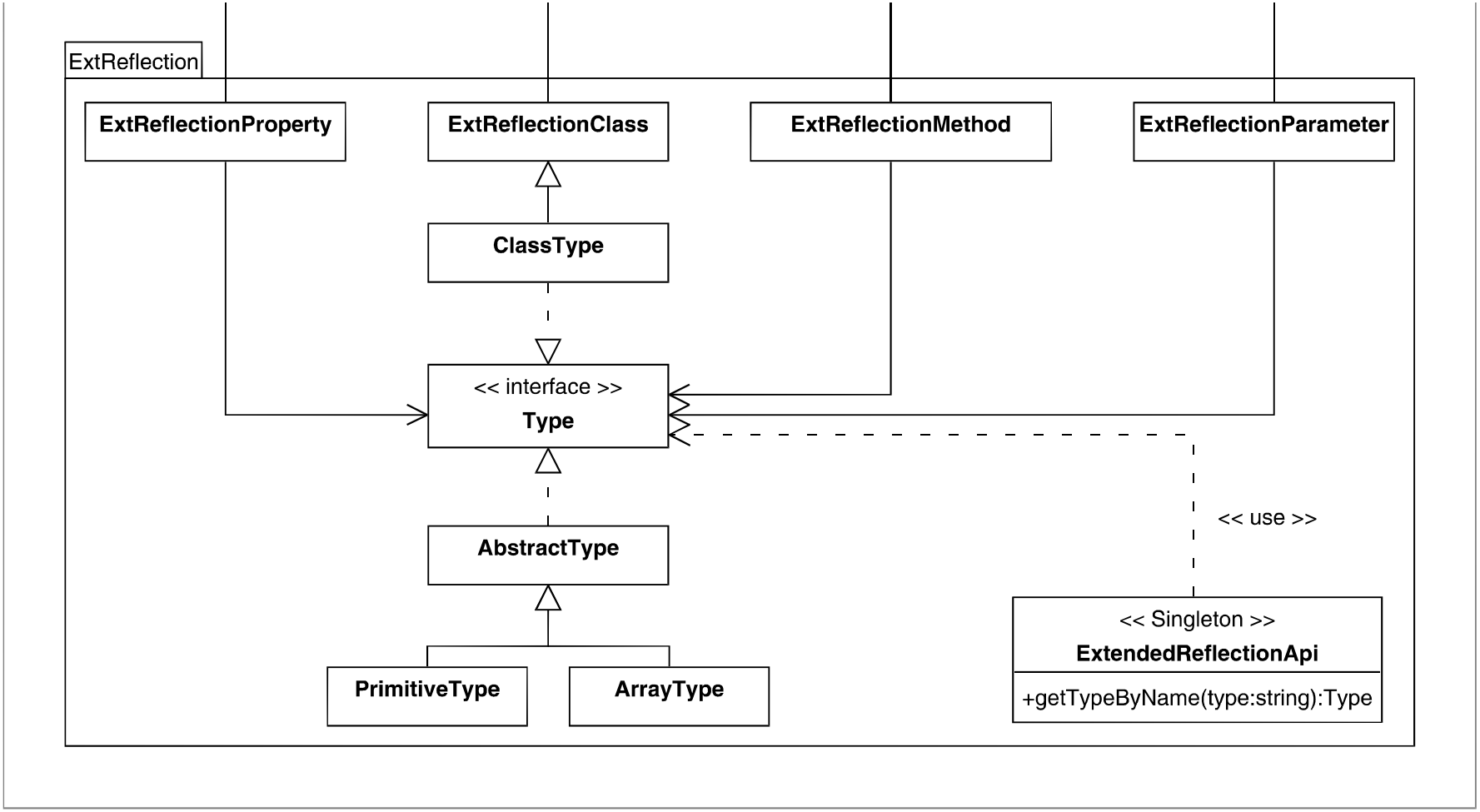


The PHP Extension

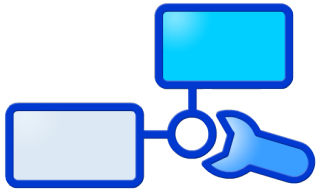




Extended Reflection API

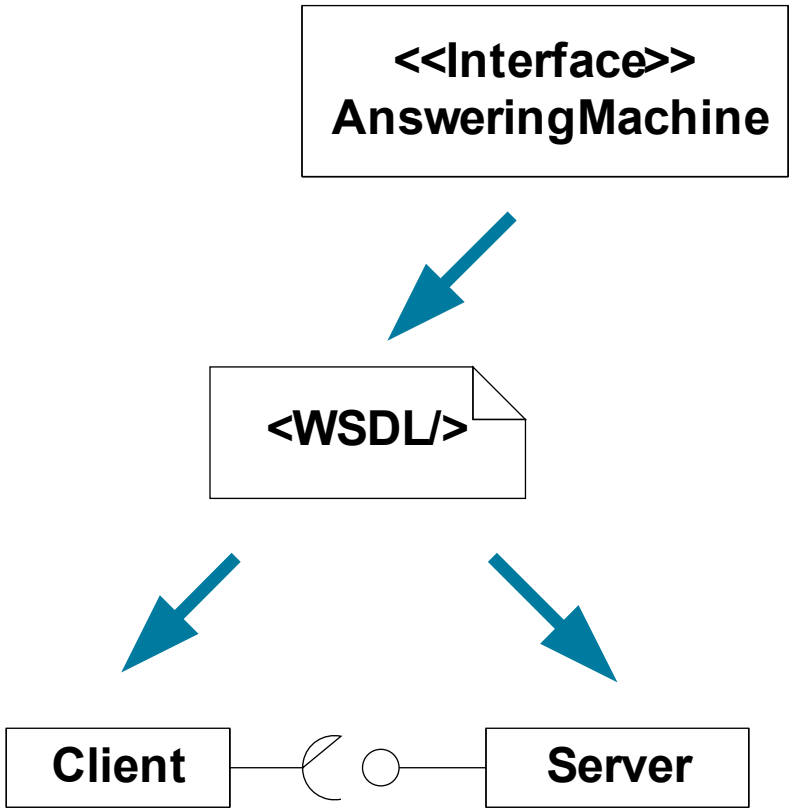


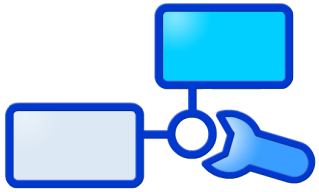
- Extended with Annotations and a type system



Web Services Description Language (WSDL)

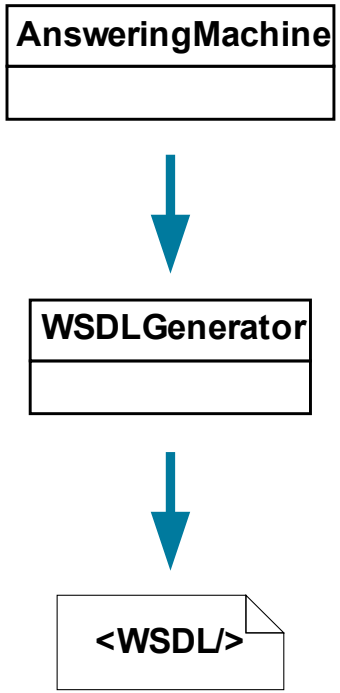
- Platform independent language for describing interfaces
- Description contains
  - Interface
  - Method signatures
  - Data types

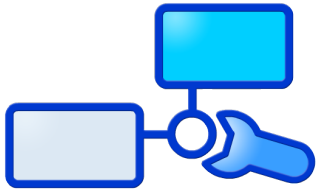




WSDL Generator

- Generates WSDL from PHP Classes or Collections of Functions
- WSDL 1.1
  - Supports RPC Encoded / Literal and Document Literal / Wrapped
  - Conforms to the WS-I Basic Profile by 95%
- Uses DOM-API
- Tested with phpt Test Cases
  - Reused from SOAP Extension
- Adapter Generator
  - For document-wrapped Binding
  - Classes for Un/wrapping of Arguments and Return Values

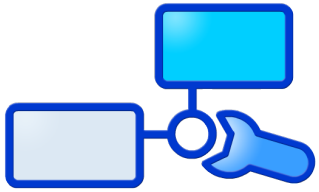




Document/Literal Adapter Generator

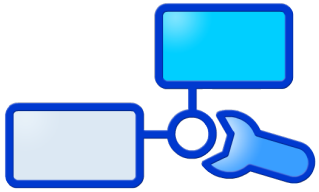
```
/**
 * @param int $id
 * @return Lecture
 */
public function getLecture($id) {…}
```

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



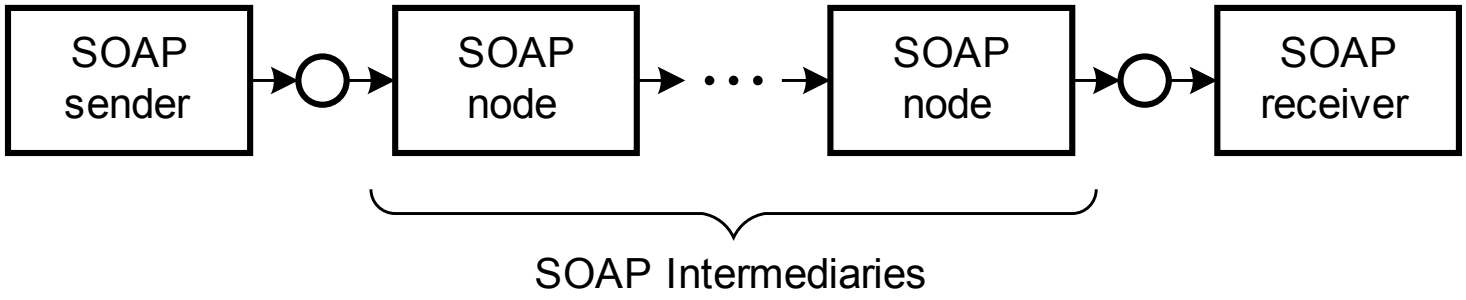
Other Web Service Standards

- Basics are already implemented for PHP
  - SOAP: PHP5 Extension, PEAR SOAP, NuSOAP
  - UDDI: PEAR Package
  - WSDL: Generator
- Over 20 additional WS-* Standards for
  - Security
  - Synchronisation
  - Sessioning
- Standards add new Elements to SOAP Header

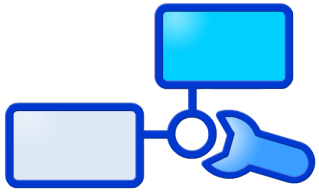


SOAP Intermediaries

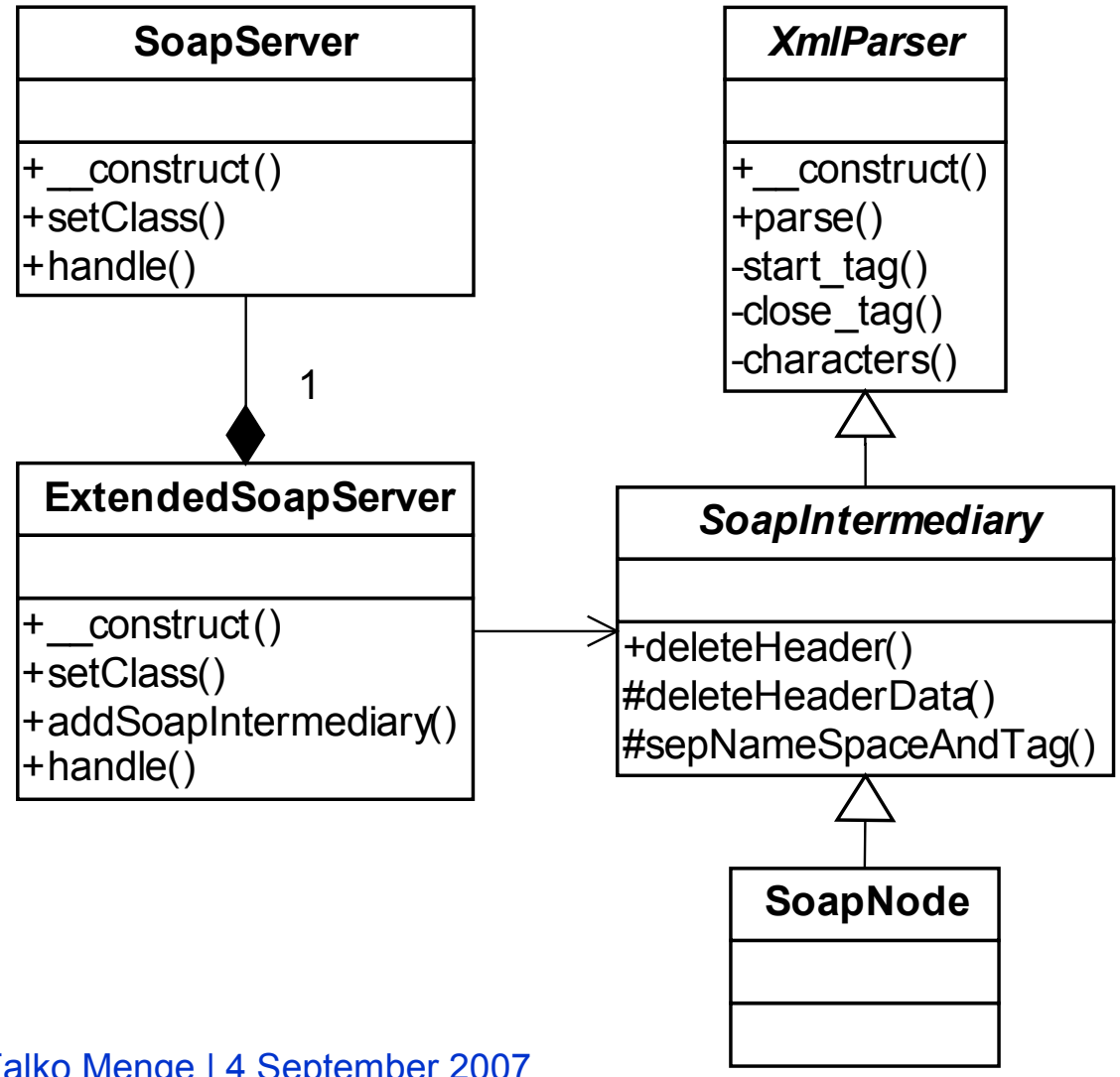
- SOAP Standard describes Intermediaries

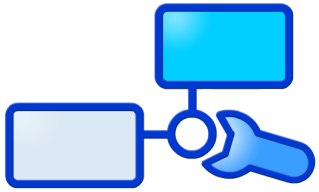


- Intermediaries are working on SOAP Message before reaching Ultimate Receiver
- Additional Features independent of final Web Service
- Implementation Pattern: Handler Chain



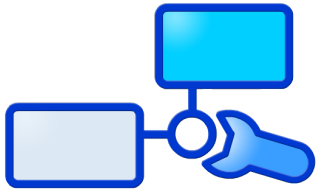
Handler Chain Mechanism





WS-Security for PHP5

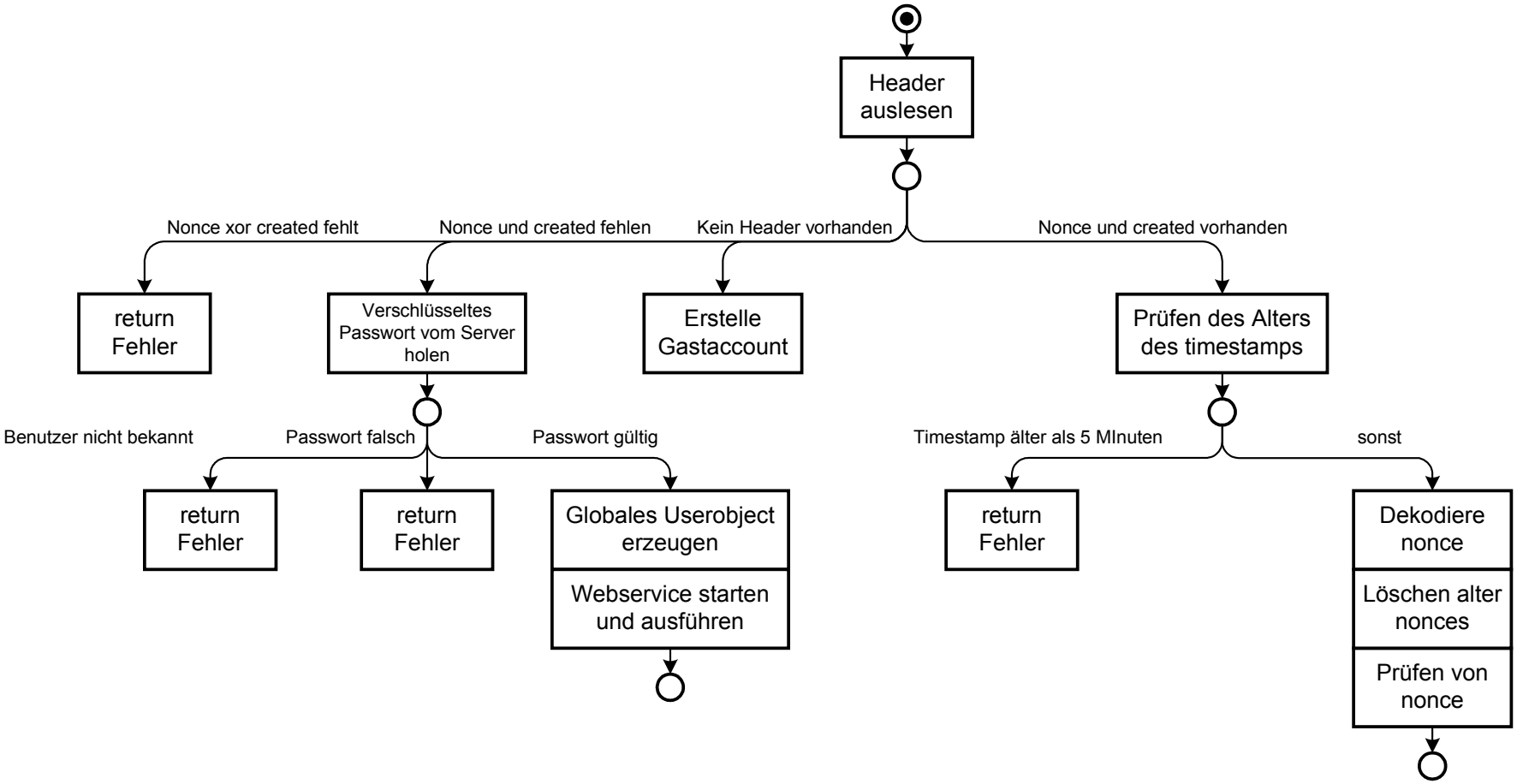
- Security Concepts for Web Services:
  - Confidentiality => SSL (SOAP via HTTPS)
  - Authentication => **WS-Security + Token Profiles**
  - Authorisation => Application
- WS-Security defines SOAP Header Element for security-related Data
- Different Profiles specify several Authentication Mechanisms
- First Profile implemented:
  - Username Token Profile 1.0



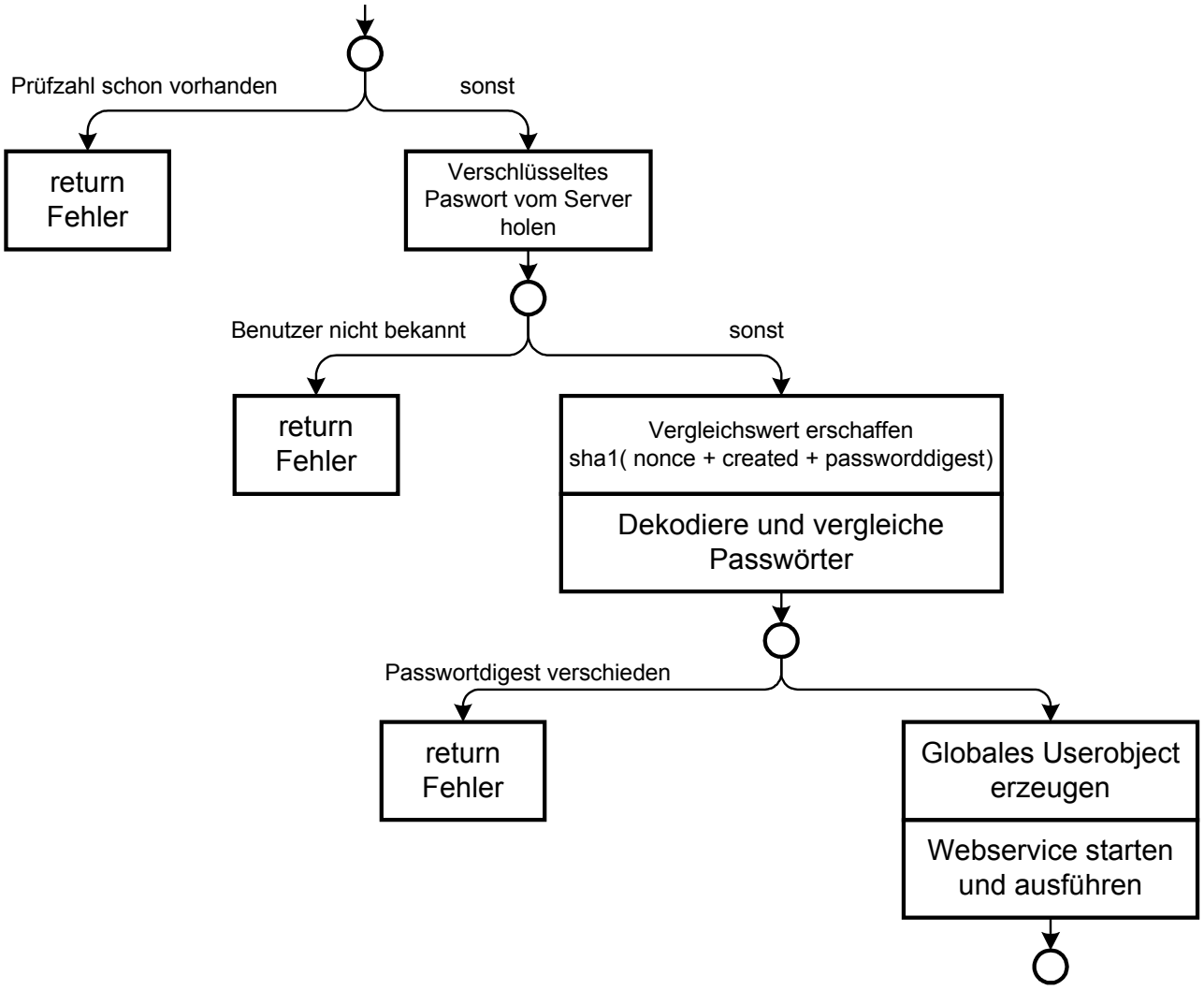
WS-Security
Username Token Profile 1.0

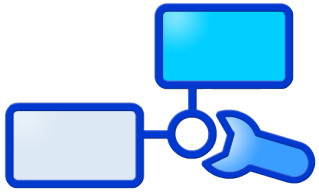
```
<wsse:Security>
    <wsse:UsernameToken>
        <wsse:Username>Stefan</wsse:Username>
        <wsse:Password Type="...#PasswordDigest">
            weYI3nXd8LjMNVksCKFV8t3rgHh3Rw==
        </wsse:Password>
        <wsse:Nonce>WScqanjCEAC4mQoBE07sAQ==</wsse:Nonce>
        <wsu:Created>2006-06-24T11:00:00Z</wsu:Created>
    </wsse:UsernameToken>
</wsse:Security>
```

Username Token Profile (1/2)

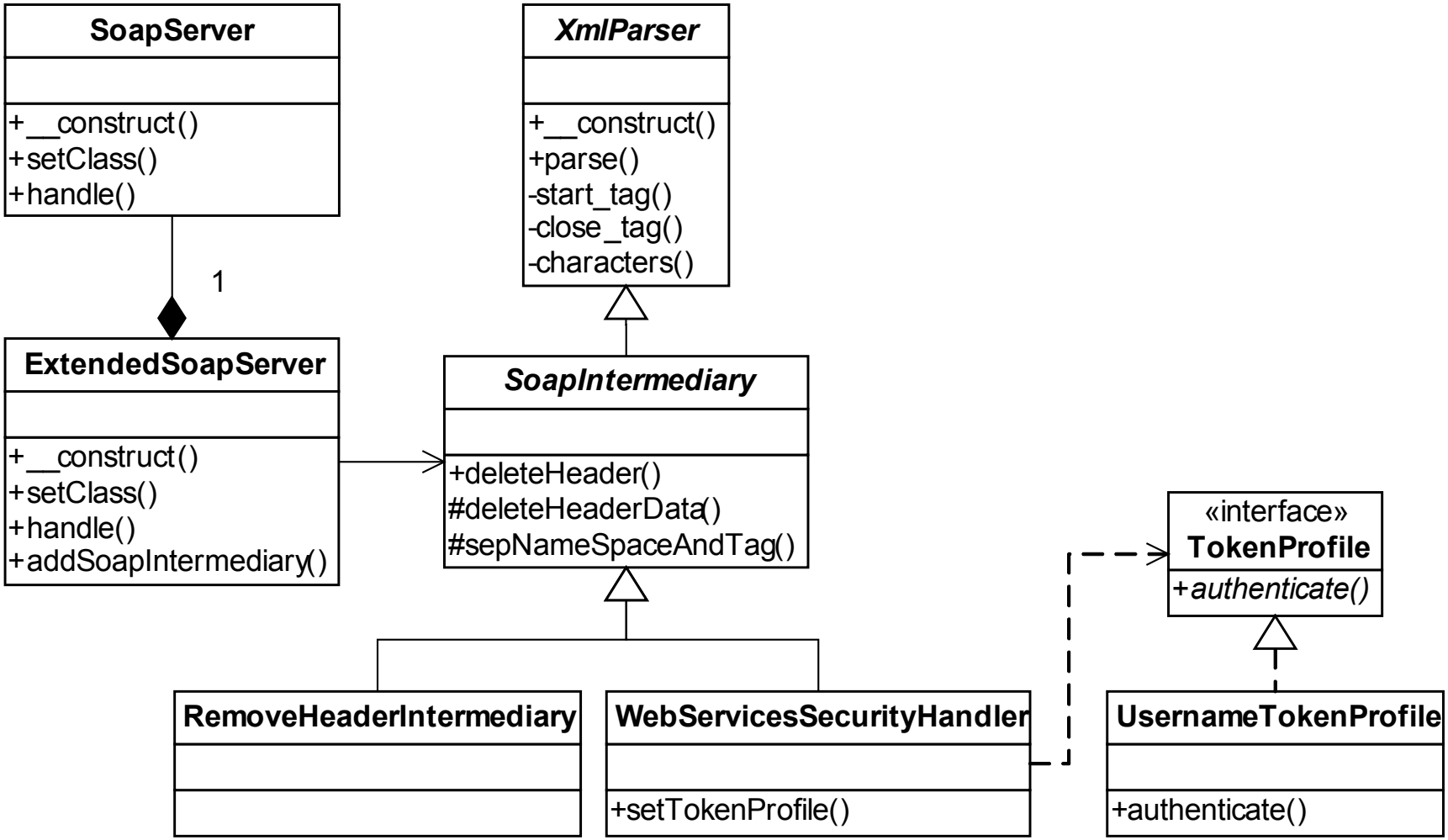


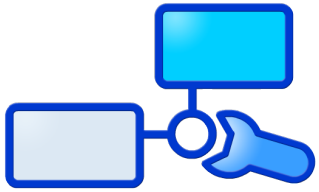
Username Token Profile (2/2)



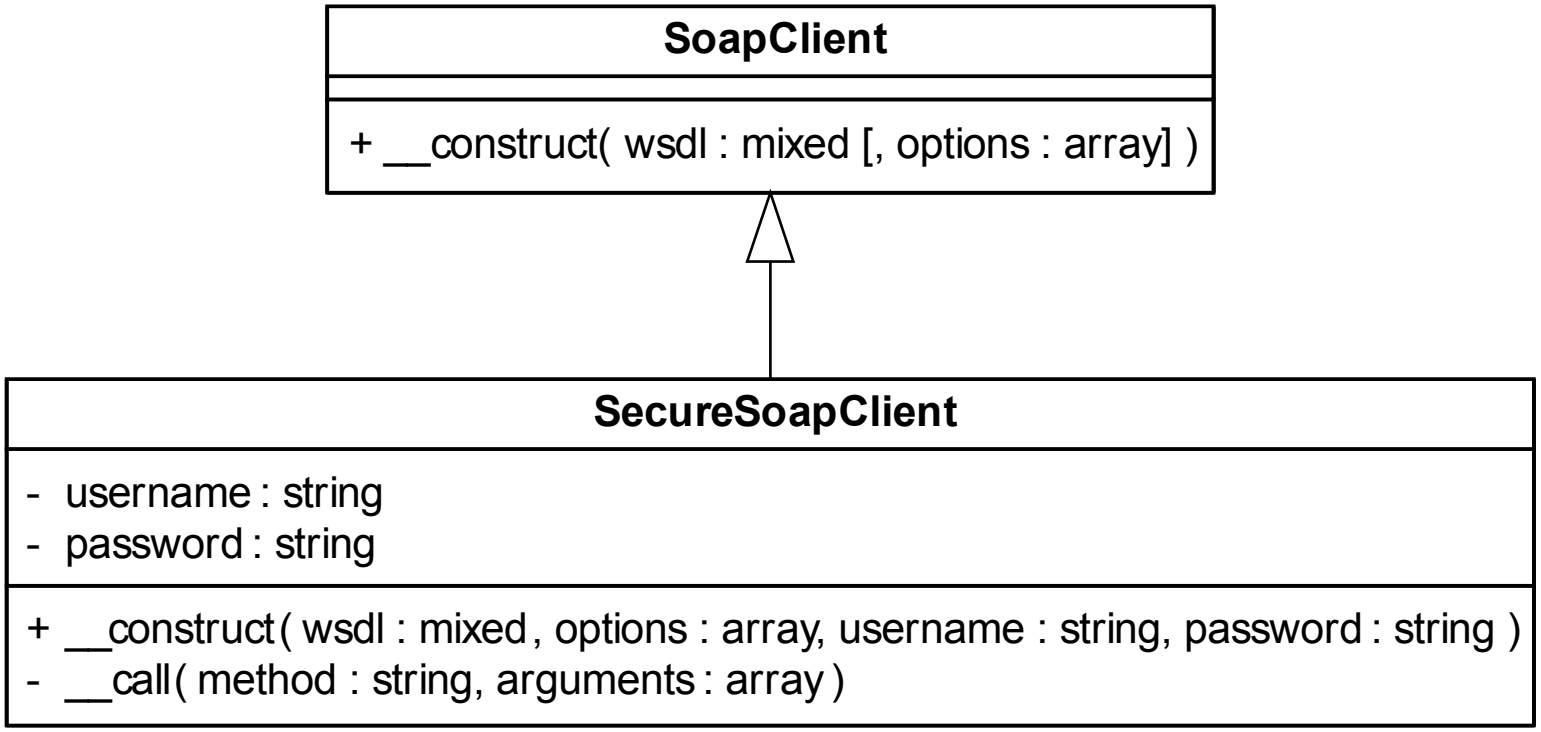


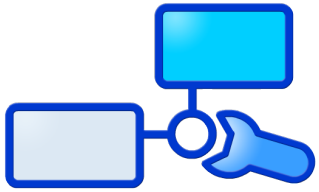
Username Token Profile 1.0 (Server)





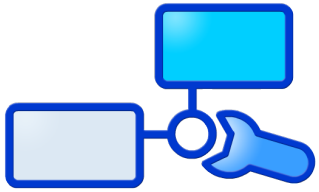
Username Token Profile 1.0 (Client)





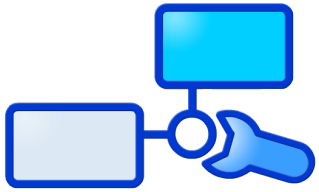
Representational State Transfer (REST)

- Alternative to SOAP Web Services
- Resource-oriented Approach
- HTTP-REST
  - Just uses HTTP Methods
    - GET, POST, PUT, DELETE
  - Messages with Payload Semantic instead of RPC Semantic
- Implemented using a Remote Facade
  - Mapping from URIs to PHP methods
  - Custom De/Serialiser for PHP Objects
  - REST Server Configuration generated from Code Annotations
- Security by TLS or DigestAuth (RFC2617)



Administration Front-End

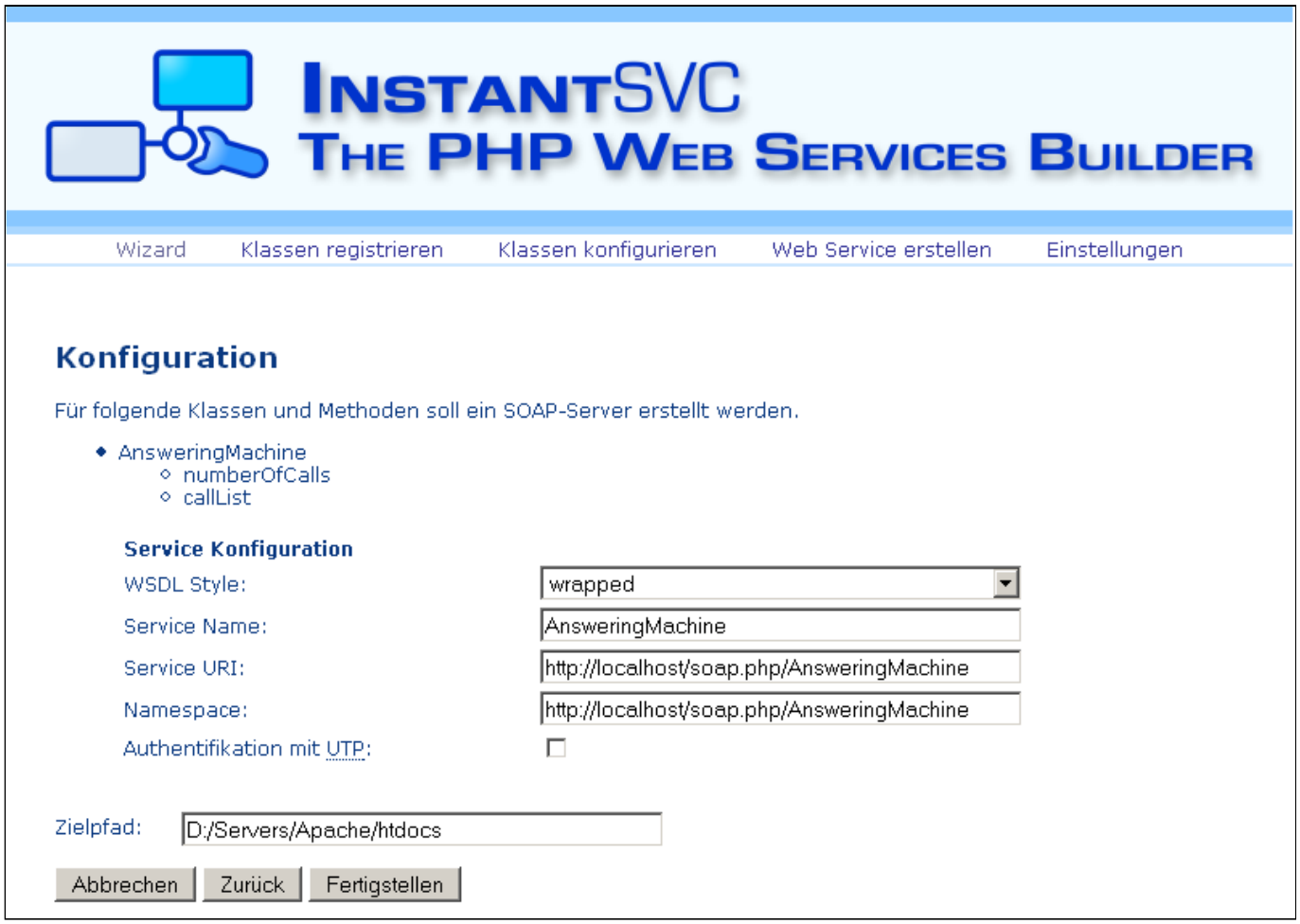
- Administration Front-End
  - Automated Creation of Web Services from existing Applications
  - Annotations identify Classes to be used
  - Administration via Web Browser
    - Classes and Methods selected by User
    - Generates SOAP and REST Server Scripts
    - Generates WSDL File and Adapter Classes
    - SOAP Server with WS-Security



Administration Tool & Policy Plug-In

- Administration Tool
  - Automated Creation of Web Services from existing Applications
  - Annotations identify Classes to be used
  - Administration via Web Frontend
    - Classes and Methods selected by User
    - Generates SOAP and REST Server Scripts
    - Generates WSDL File and Adapter Classes
    - SOAP Server 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 Administrator to edit published Comments

Administration Front-End



InstantSVC
The PHP Web Services Builder

Wizard | Klassen registrieren | Klassen konfigurieren | Web Service erstellen | Einstellungen

Konfiguration

Für folgende Klassen und Methoden soll ein SOAP-Server erstellt werden.

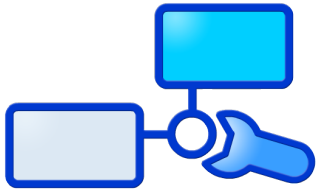
- AnsweringMachine
  - numberOfCalls
  - callList

Service Konfiguration

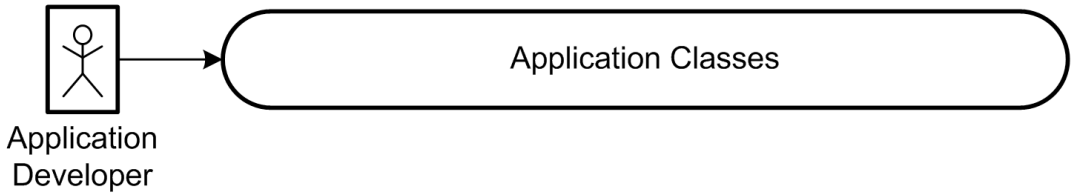
WSDL Style:	wrapped
Service Name:	AnsweringMachine
Service URI:	http://localhost/soap.php/AnsweringMachine
Namespace:	http://localhost/soap.php/AnsweringMachine
Authentifikation mit UTP:	☐

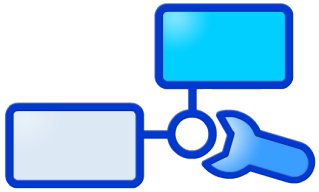
Zielpfad: D:/Servers/Apache/htdocs

Abbrechen | Zurück | Fertigstellen

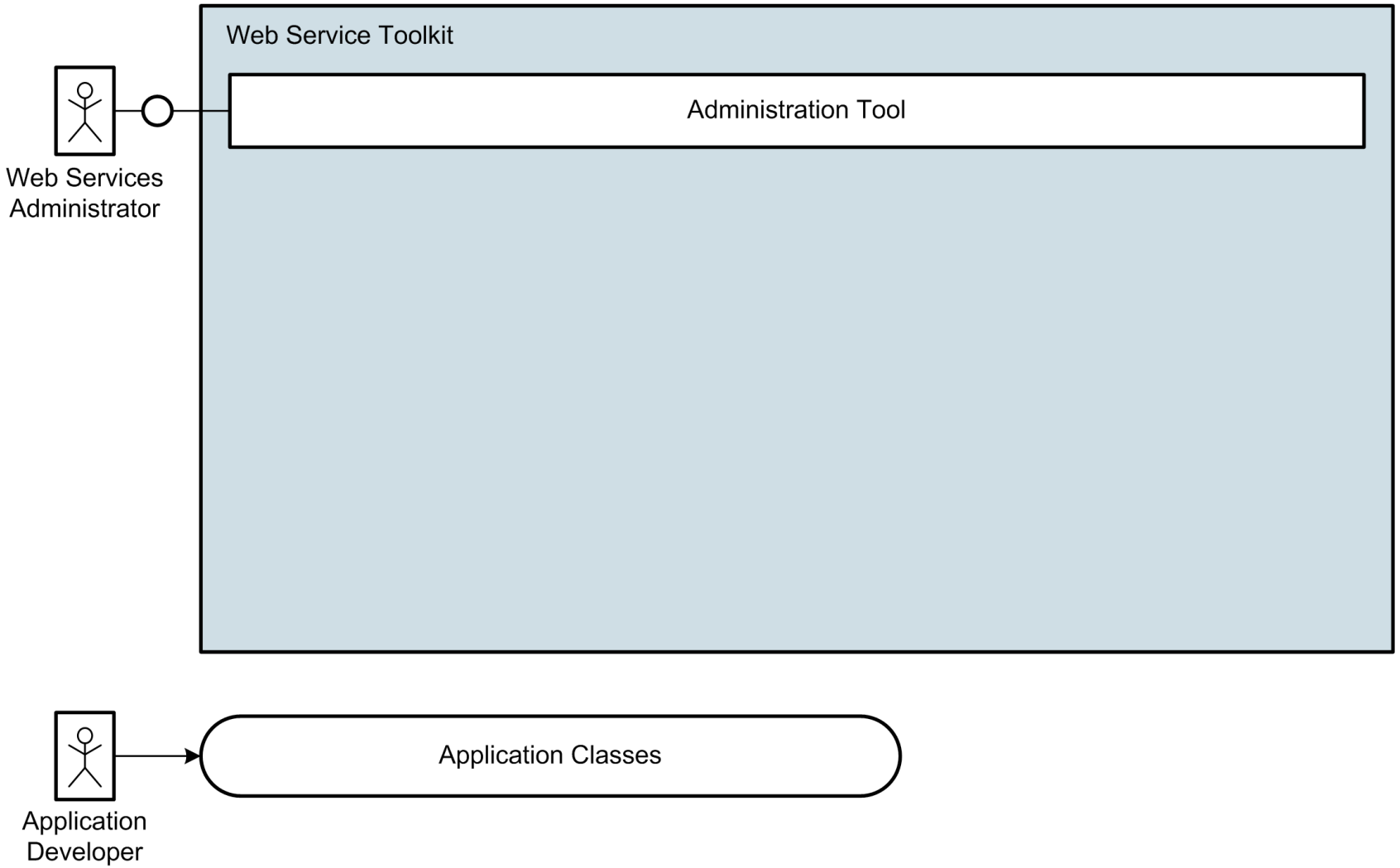


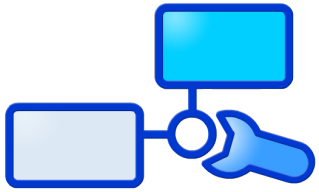
Interaction Overview



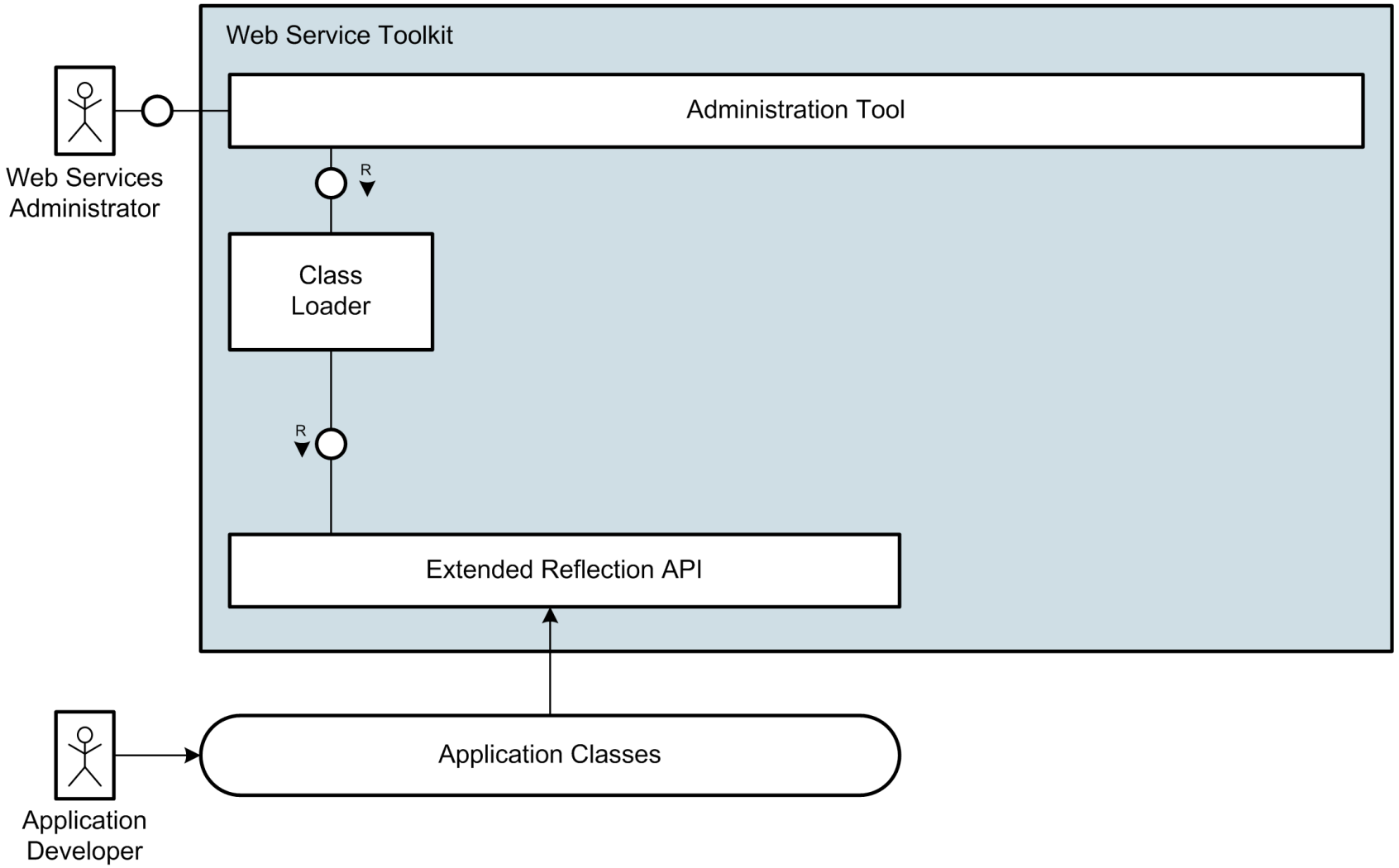


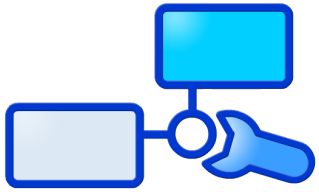
Interaction Overview



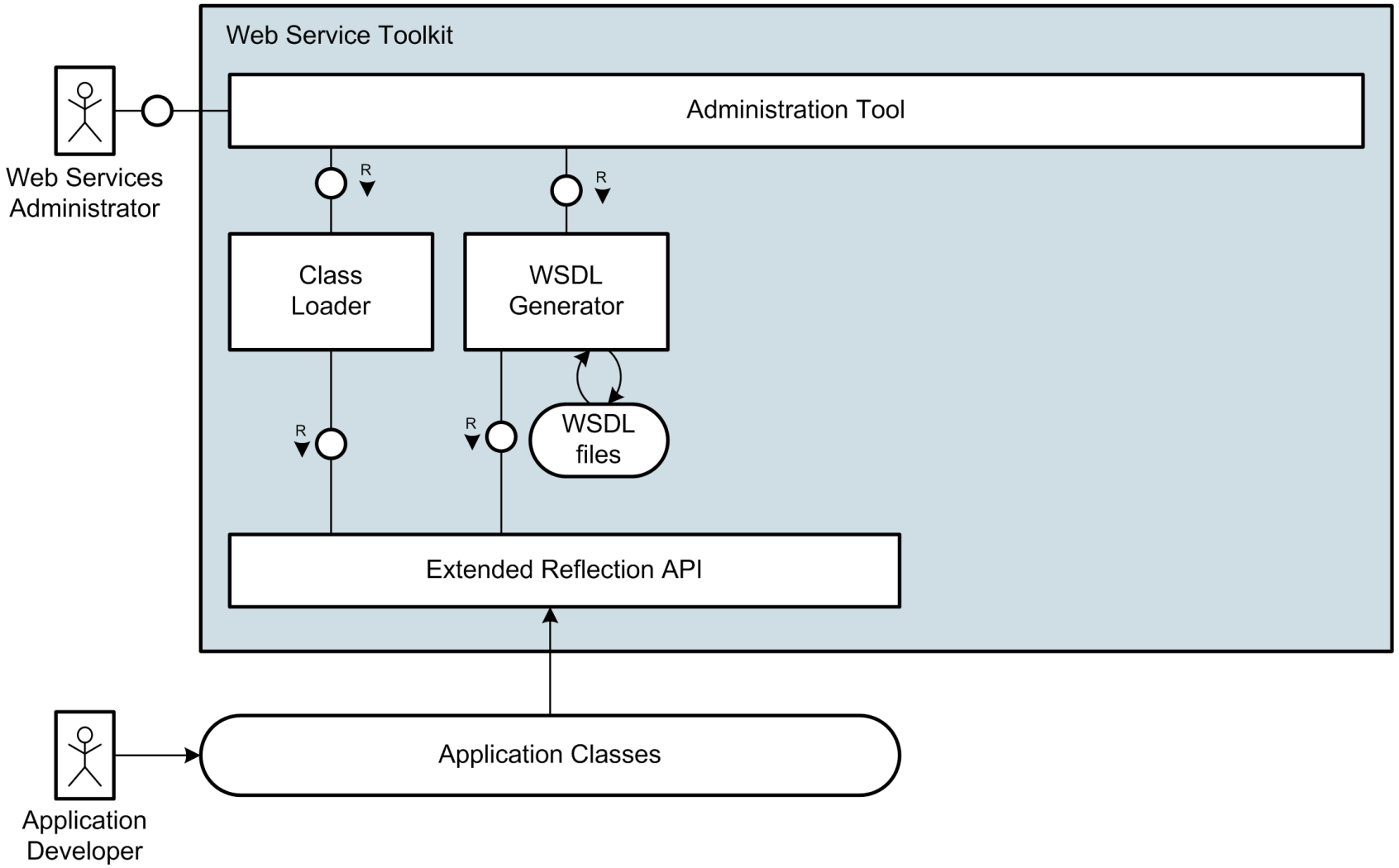


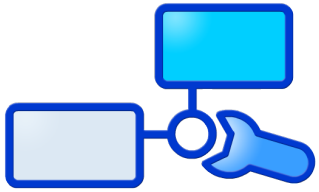
Interaction Overview



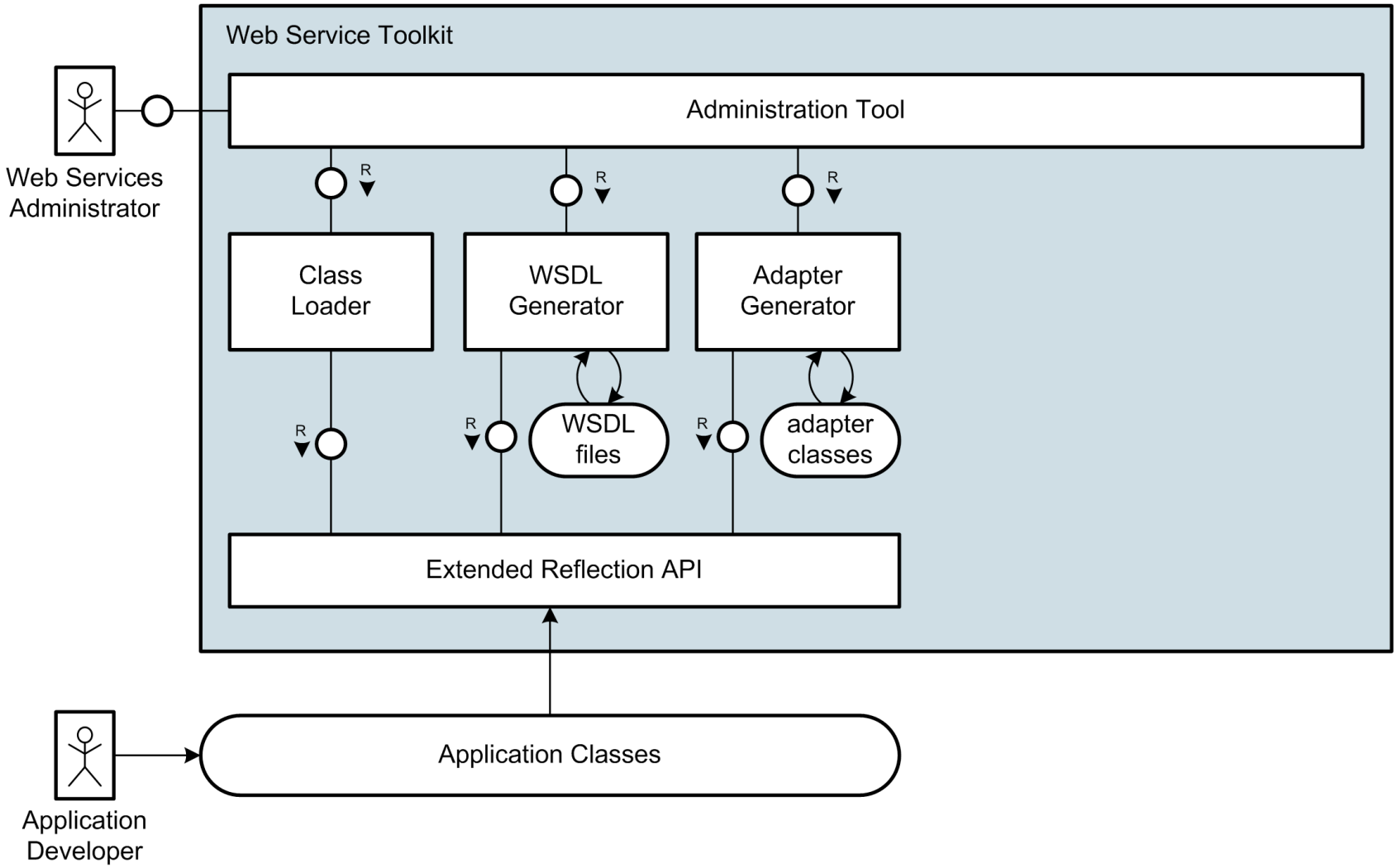


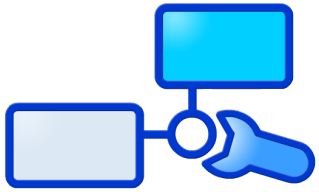
Interaction Overview



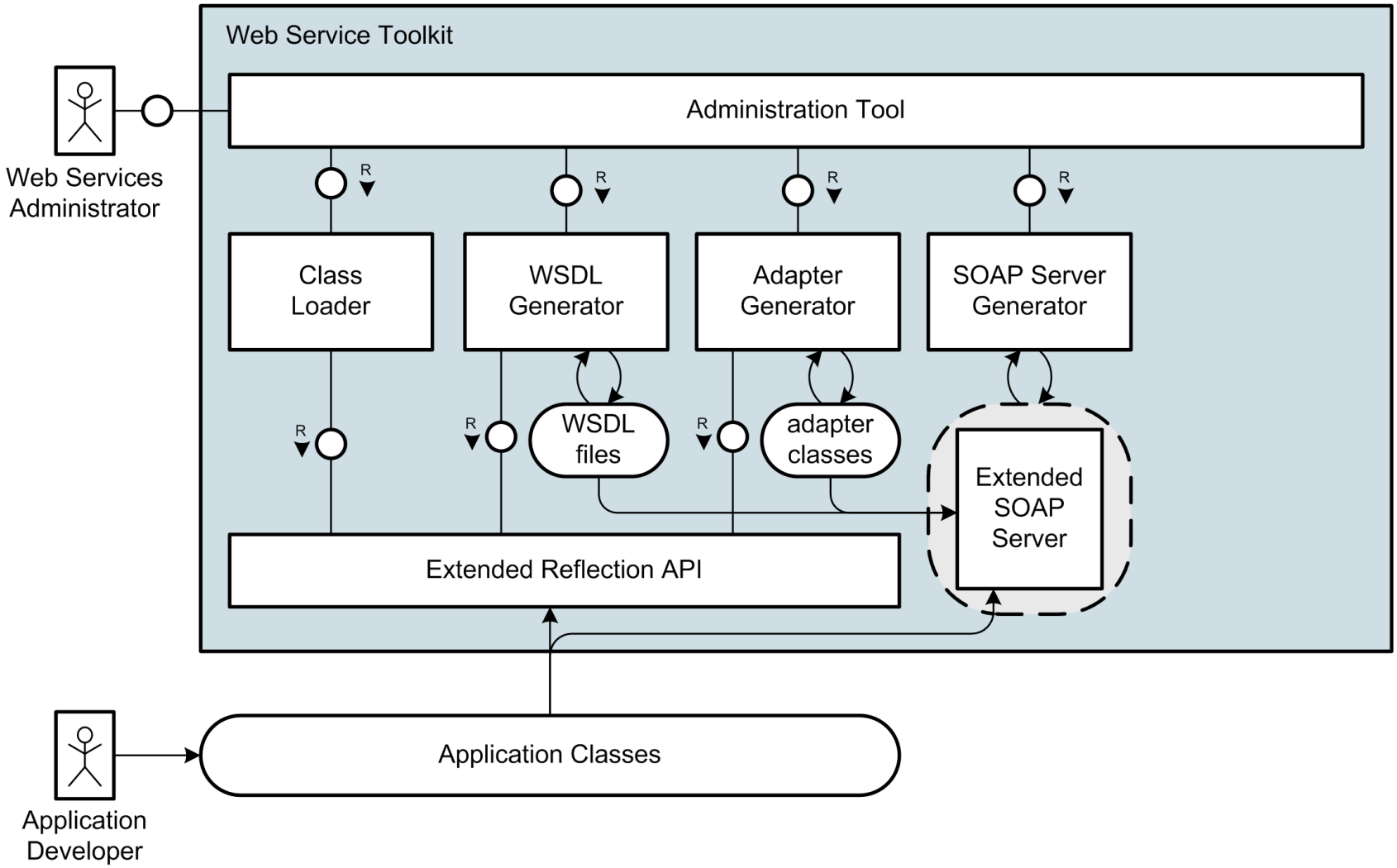


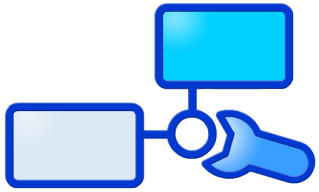
Interaction Overview



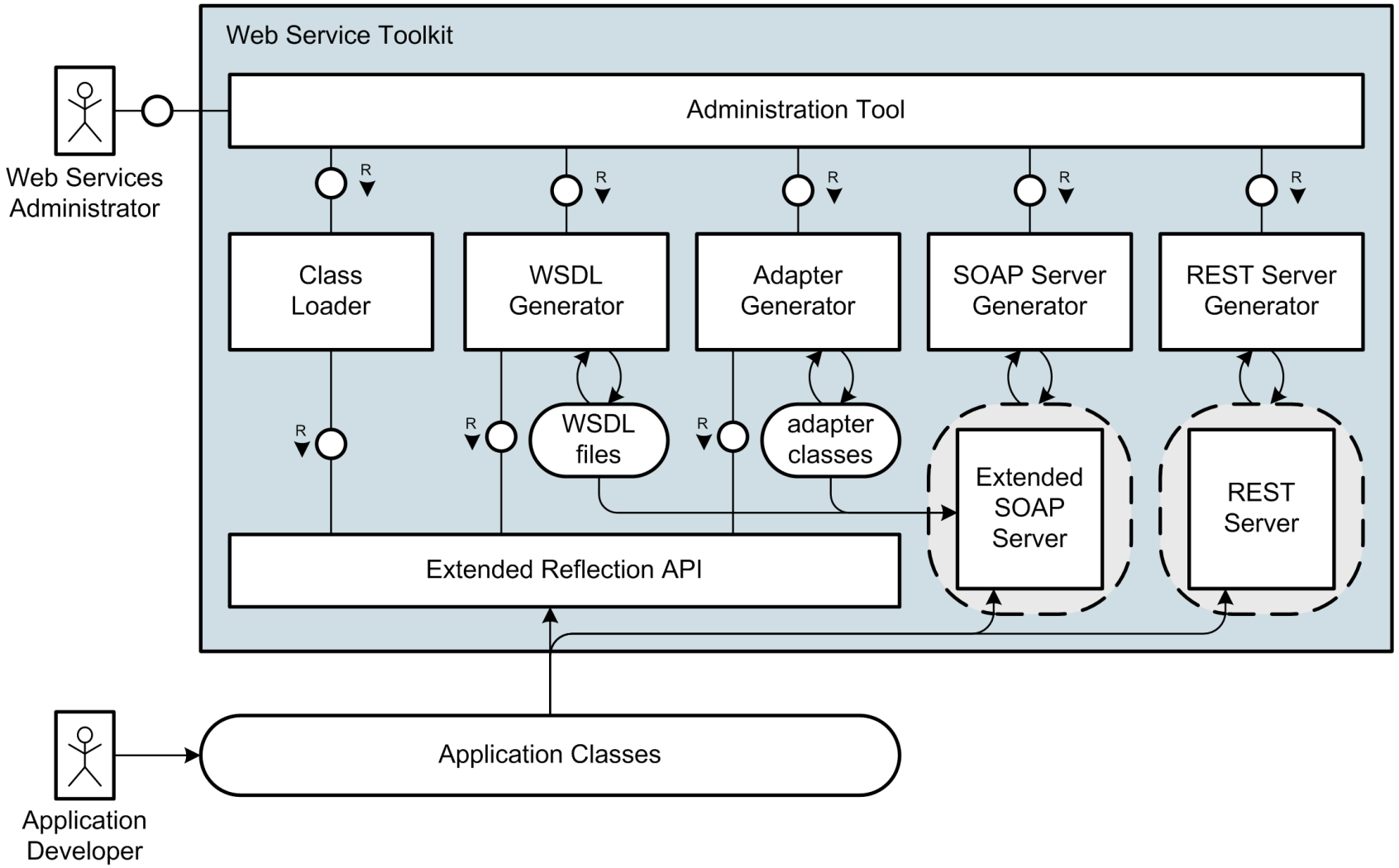


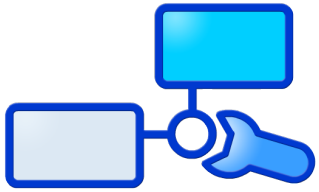
Interaction Overview



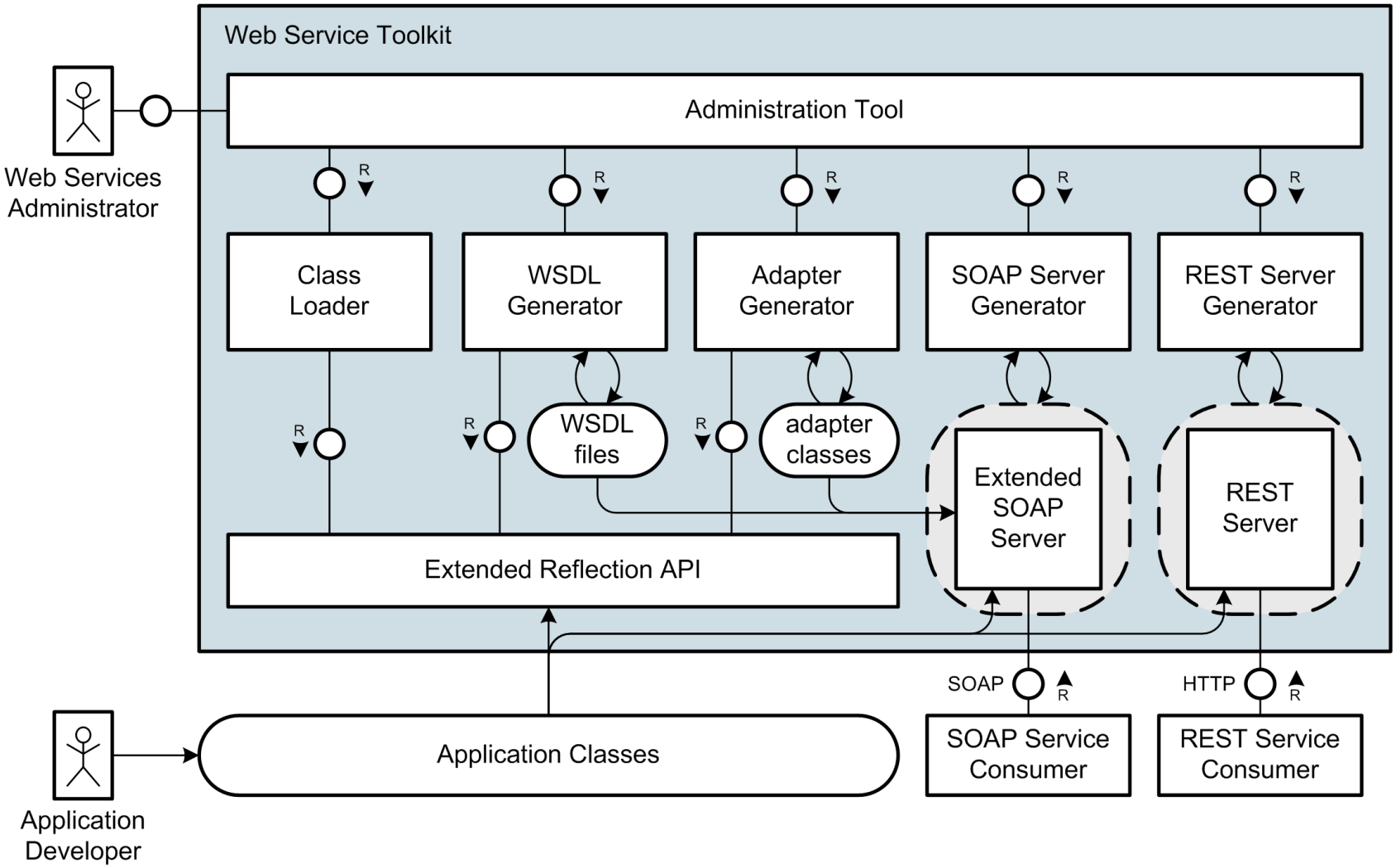


Interaction Overview

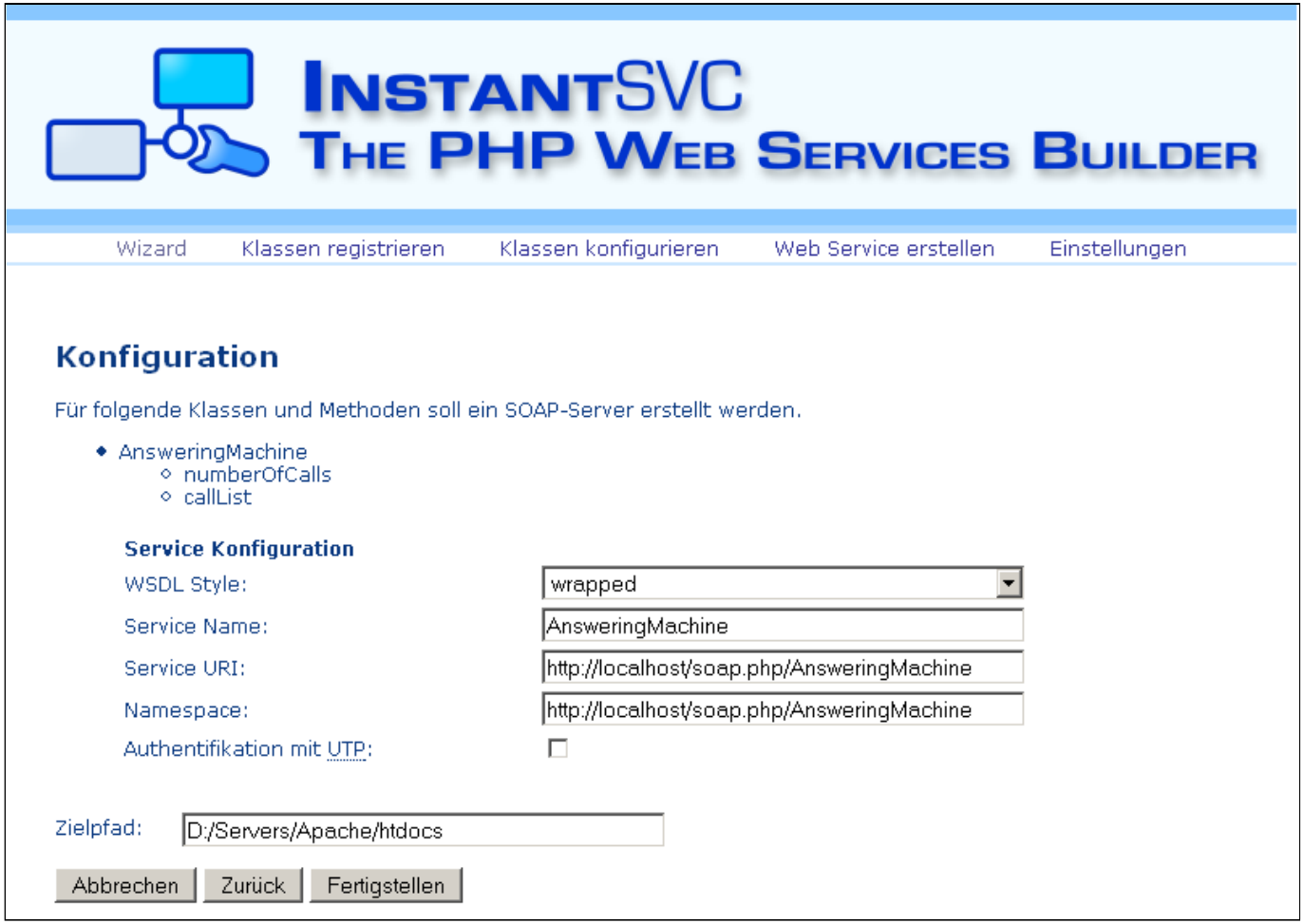


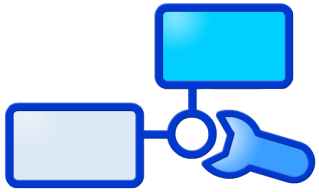


Interaction Overview



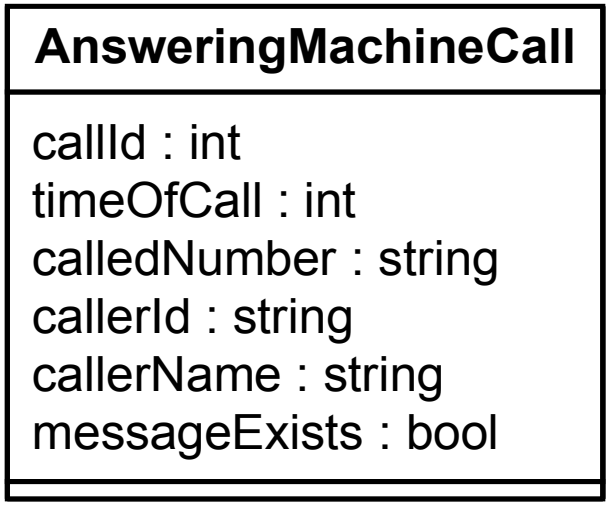
Live Demo

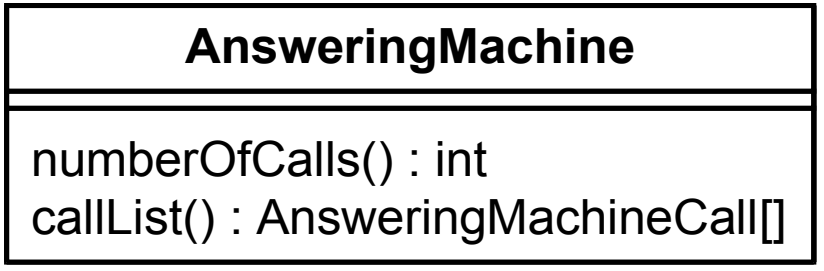

INSTANTSVC
THE PHP WEB SERVICES BUILDER
Wizard
Klassen registrieren
Klassen konfigurieren
Web Service erstellen
Einstellungen
Konfiguration
Für folgende Klassen und Methoden soll ein SOAP-Server erstellt werden.
AnsweringMachine
numberOfCalls
callList
Service Konfiguration
WSDL Style:
wrapped
Service Name:
AnsweringMachine
Service URI:
http://localhost/soap.php/AnsweringMachine
Namespace:
http://localhost/soap.php/AnsweringMachine
Authentifikation mit UTP:
Zielpfad:
D:/Servers/Apache/htdocs
Abbrechen
Zurück
Fertigstellen

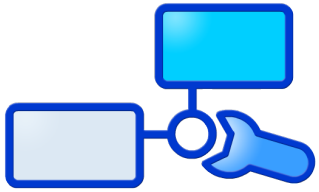


Example Application

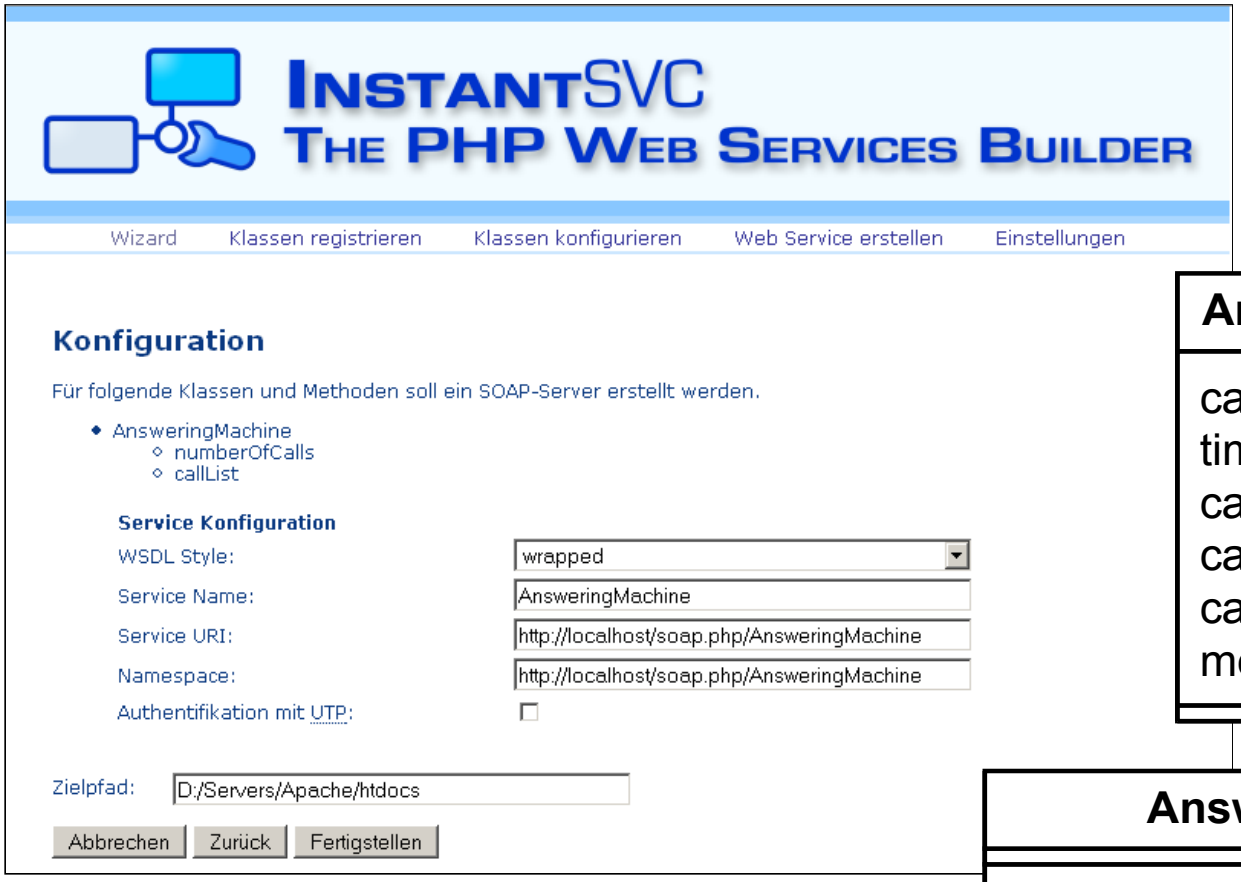
- Application: Answering Machine
  - Number of Calls
  - List of Calls
- Example Web Service
  - using SOAP Protocol

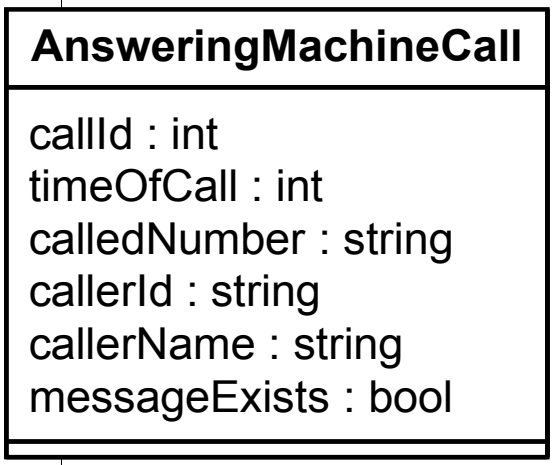


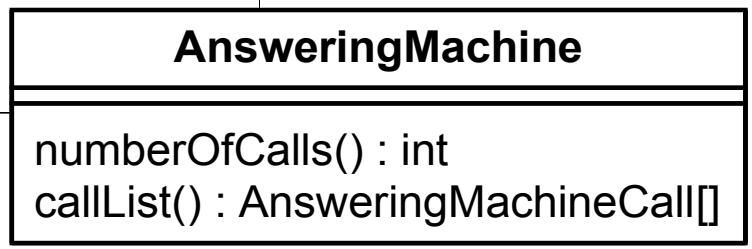




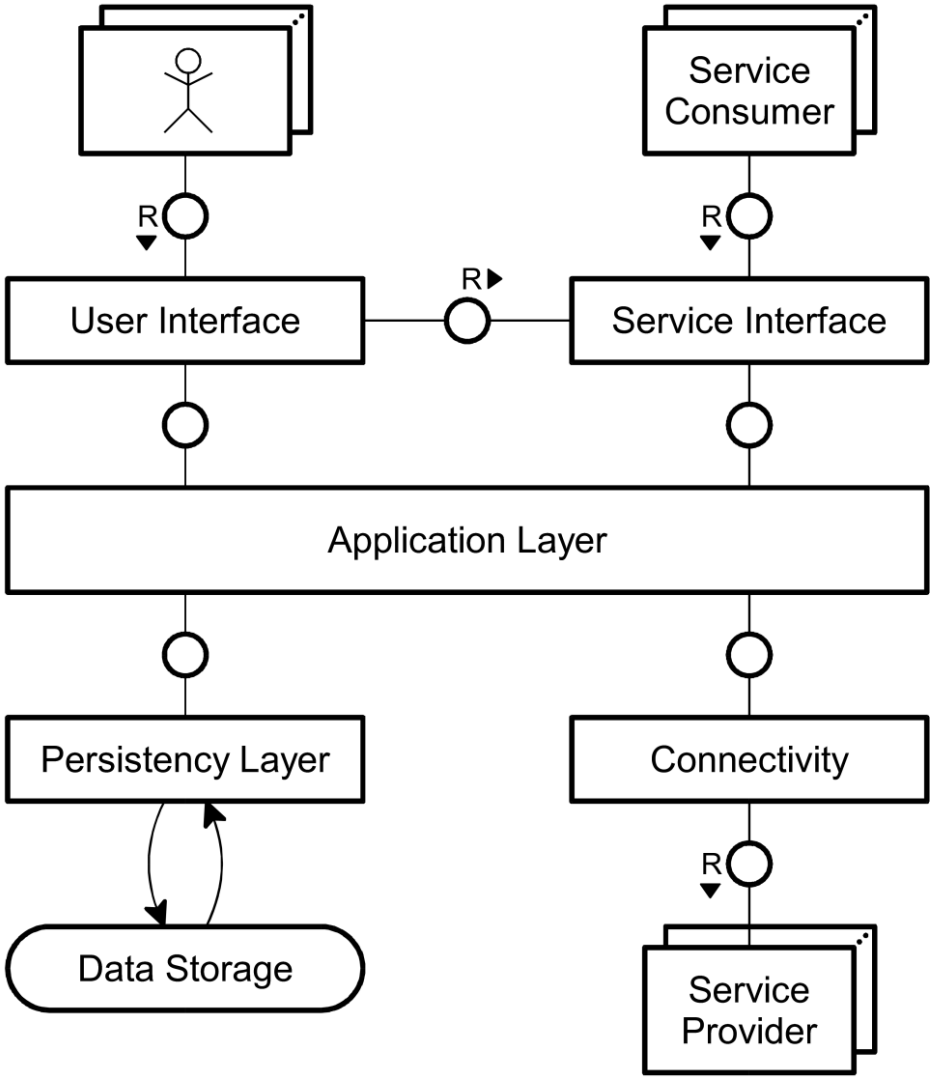
Live Demo

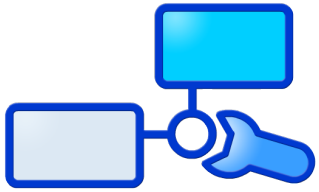

INSTANTSVC
THE PHP WEB SERVICES BUILDER
Wizard
Klassen registrieren
Klassen konfigurieren
Web Service erstellen
Einstellungen
Konfiguration
Für folgende Klassen und Methoden soll ein SOAP-Server erstellt werden.
AnsweringMachine
numberOfCalls
callList
Service Konfiguration
WSDL Style:
wrapped
Service Name:
AnsweringMachine
Service URI:
http://localhost/soap.php/AnsweringMachine
Namespace:
http://localhost/soap.php/AnsweringMachine
Authentifikation mit UTP:
Zielpfad:
D:/Servers/Apache/htdocs
Abbrechen
Zurück
Fertigstellen


AnsweringMachineCall
callId : int
timeOfCall : int
calledNumber : string
callerId : string
callerName : string
messageExists : bool


AnsweringMachine
numberOfCalls() : int
callList() : AnsweringMachineCall[]

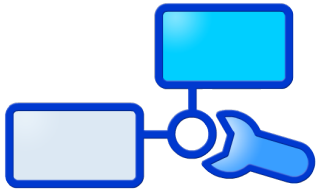
Proposed Application Architecture





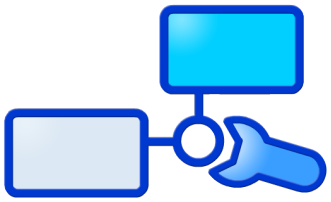
Summary

- Extended Reflection API with Information about Types
- Annotations for PHP
- WSDL Generator conform to WS-I Basic Profile
- Document/Literal Adapter Generator
- Handler Chain Mechanism for SOAP Processing
- Implementation of WS-Security and Username Token Profile
- RESTful Web Services
- Administration Tool for convenient Creation und Management of SOAP and REST Services

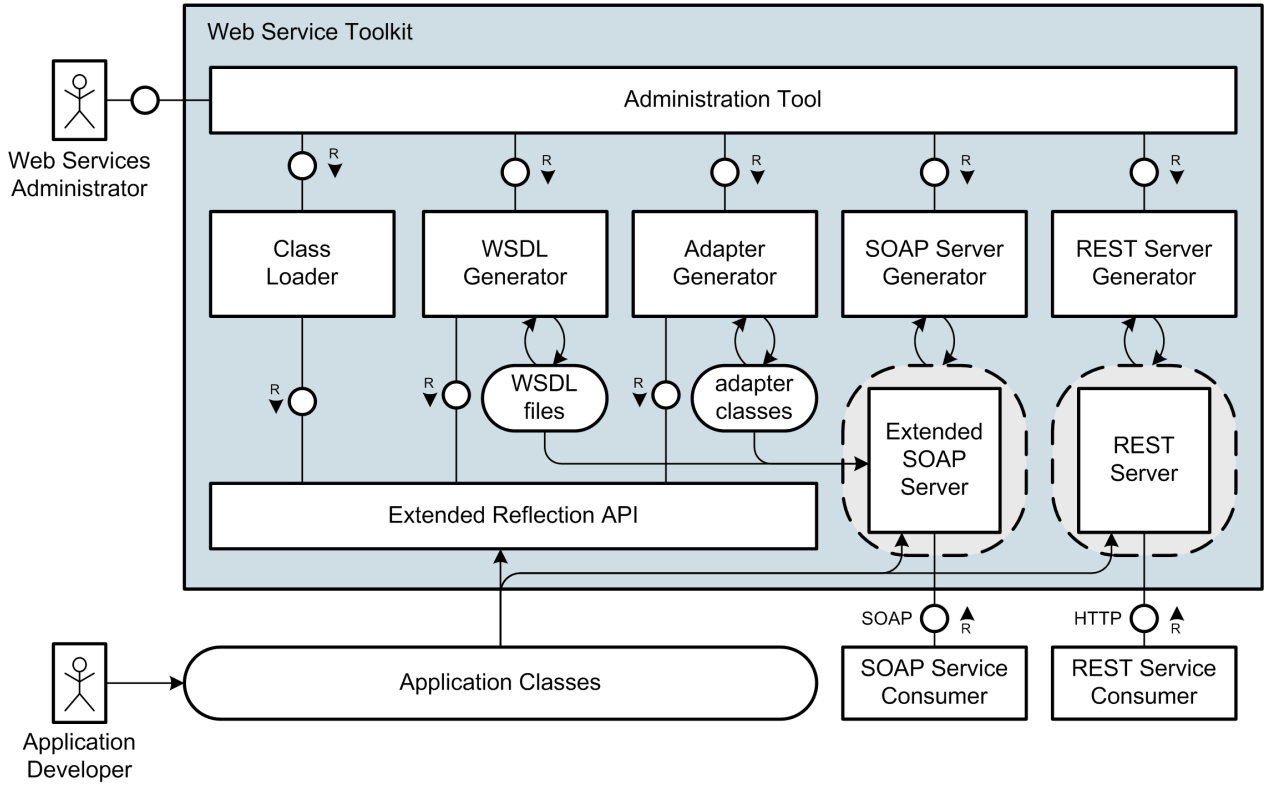


Future of the Project

- Developed by 6 HPI Students since October 2005
  - Stefan Marr
  - Christoph Hartmann
  - Michael Perscheid
  - Martin Sprengel
  - Gregor Gabrysiak
  - Falko Menge
- Project presented at the FrOSCon 2006 (Free and Open Source Software Conference)
- Next Step: Contribution to eZComponents



InstantSVC
The PHP Web Services Builder



Further Information and Download at:

http://instantsvc.sourceforge.net