Web Services
mit
PHP/NuSoap und C#/.NET

Dipl.-Ing. Torsten Zachert
Geschäftsführer
teserco GmbH

torsten.zachert@teserco.de

Was ist ein Web Service ?

- Anwendung, ausführbarer Code
- Ausführung auf einem Webserver
- Funktionen sind über XML – Protokolle verfügbar
- Aufsetzen auf Internetstandard HTTP
- Aufruf einer Funktion ist so einfach, wie der Aufruf einer lokalen Funktion

Warum gibt es Web Services ?

- Internet wird programmierbar
- Ergänzung/Ablösung von COM/DCOM, CORBA/IIOP, RMI (Probleme mit Skalierung, Plattformabhängigkeit, Komplexität)
- Einigung zwischen, IBM, Sun, Microsoft
- Schaffung einer Möglichkeit, plattform- und programmiersprachenübergreifende Kommunikation zwischen verschiedenen Geschäftsprozessen zu realisieren
- Standards: HTTP, XML, SOAP und WSDL
- Vision: Neue Anwendungen werden einfach aus bereits im Netz verfügbaren Web Services „zusammengeklickt“
- UDDI – Universal Discription, Discovery and Integration
- einfache Nutzung; für Menschen lesbare Nachrichten

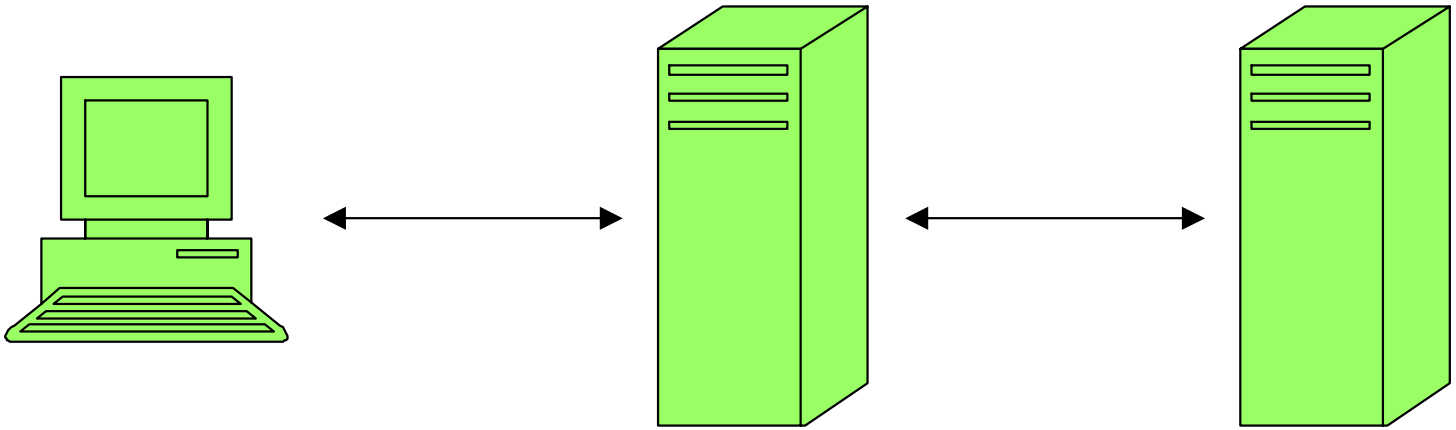
Wo sind die Einsatzmöglichkeiten ?

- B2B – Ersatz von proprietären Lösungen (z.B. EDI)
- Einsatz im Intranet – Datenaustausch zwischen verschiedenen Geschäftsbereichen
- Angebot von Funktionalitäten (z.B. Verteiltes Rechnen, Authentifizierung)
- Zentralisierung von Code-Komponenten (Nutzung durch alle Clients im Unternehmen)

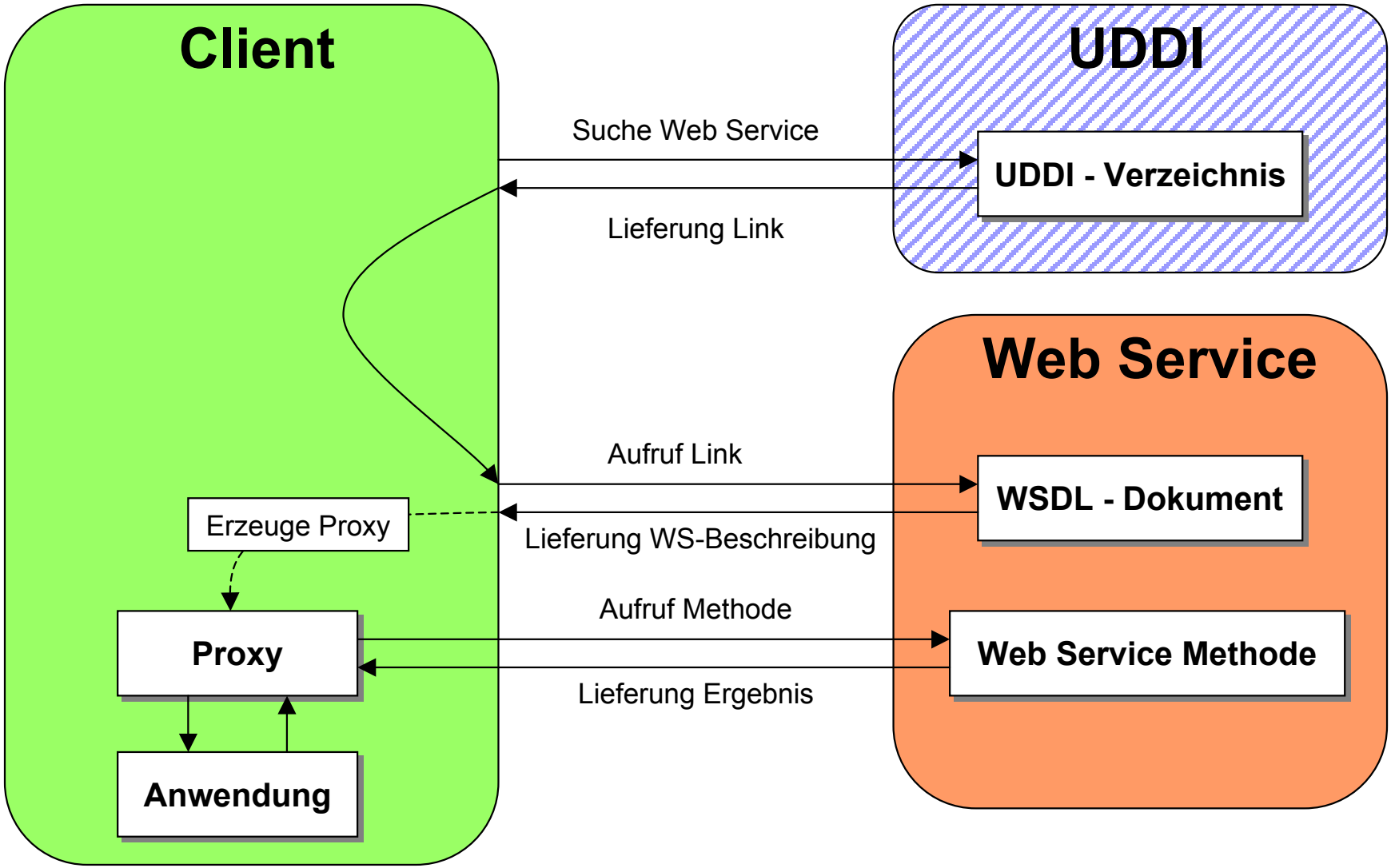
Browsing und Web Services

Browsing - Lieferung von Daten zur Anzeige

Web Services - Lieferung von Daten zur Verarbeitung mit Anwendungen



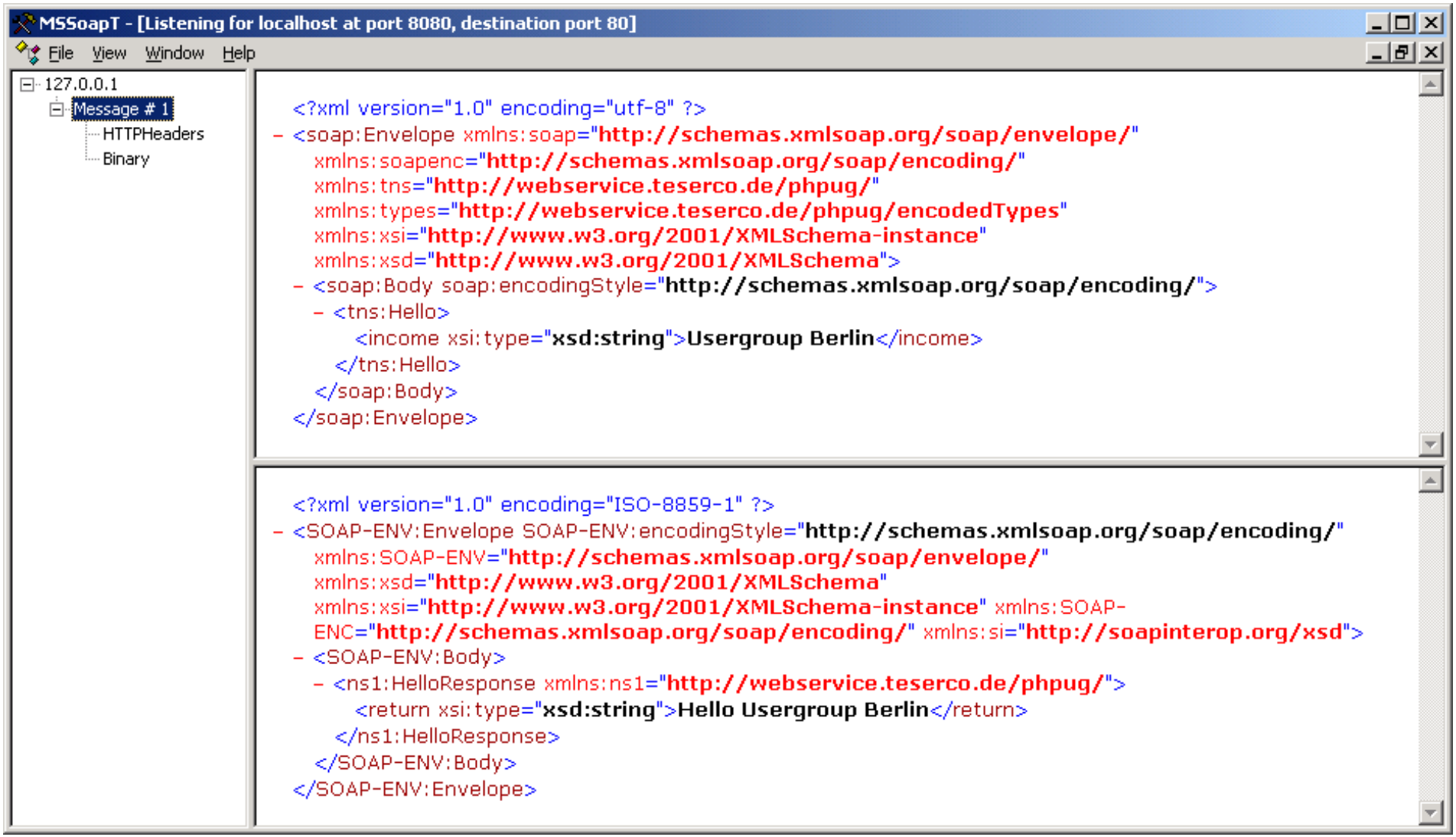
Grundlegende Architektur



WSDL - Web Service Description Language

- **Definitions**
  - Wurzelement
  - Deklaration des WS-Namens und der Namensräume
- **Types**
  - Deklaration eigener Typen (Ableitung von XML-Datentypen)
- **Message**
  - Abstrakte Typdefinitionen
- **Port Type**
  - Gruppe von Operationen
- **Binding**
  - Beschreibung der Transportprotokolle, Nachrichtenformate und Stile
- **Service**
  - Adresse des Web Service

SOAP – Simple Object Access Protocol



SOAP – Nachrichtenformate / Kodierungsformate

Nachrichtenformate

- RPC
- Document

Kodierungsformate

- encoded
- literal

am häufigsten verwendet

- RPC/encoded
- Document/Literal

Web Service Implementierungen

- PHP
  - NuSoap
  - PEAR SOAP
  - PHP 5
- Microsoft
  - .NET
  - MS SOAP
- Perl/Soap
- SUN Soap
- HP Soap
- BEA WebLogic
- Apache Axis
- Apache SOAP 2.2
- SAP::SOAP

NuSoap – eine PHP Web Service Implementierung

- Maintainer: Dietrich Ayala, Scott Nichol
- Aktueller Hauptentwickler: Scott Nichol
- Aktuelle Version: 0.6.8 Rev. 1.76
- Aktuelle Klassen
  - Basisbibliothek
  - SOAP XML Parser
  - XML Schema Parser
  - WSDL Parser
  - Server-Tools
  - Client-Tools
  - Nachrichtentransport
  - Serialisierung
  - Fehlerbehandlung

Quellen

- Interoperabilität verschiedener Web Service-Implementierungen:
  http://www.whitemesa.com/interop.htm
- NuSOAP Quelldateien: http://sourceforge.net/projects/nusoap/
- Mailing-Liste:
  https://lists.sourceforge.net/lists/listinfo/nusoap-general
- NuSOAP API Beschreibung: http://dietrich.ganx4.com/nusoap/APIDoc/
  (basiert auf Ursprungsprojekt SOAPx4)
- Beispiele: http://www.scottnichol.com
- SOAP Toolkit 3.0:
  http://msdn.microsoft.com/webservices/downloads/default.aspx
- TcpTrace: http://www.pocketsoap.com/tcpTrace/
- ASP.NET Web Matrix: http://www.asp.net/webmatrix

Aktuelle Problemstellungen

- Interoperabilität
- Transaktionssicherheit
- Web Service Security

Variante 1

- PHP/NuSOAP als Web Service
- C#/.NET als Client

Variante 1 - Der Rahmen

```
<?php
require_once('./lib/0.6.8/1.76/nusoap.php');
$server = new soap_server();
$server->soap_defencoding = 'ISO-8859-1';
```

```
/* Registrierung der Web Service-Methoden */
/* Deklaration der Web Service-Methoden */
```

```
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
  $HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>
```

Variante 1

Beispiele

Variante 1 - Interoperabilität - Datentypen

- Einfache Datentypen: leere Parameter, leere Antwort, string, integer, float, double, decimal, dateTime, boolean, base64Binary
- Komplexe Datentypen (Arrays): string, integer, float, double, decimal, dateTime, boolean, base64Binary
- Komplexe Datentypen (Structs)
- Komplexe Datentypen (Arrays von Structs)

Variante 1 – Debugging und Fehlerbehandlung

- Web Service - Debugging einschalten mit `$debug = 1`
  (Daten werden als Kommentar an Envelope angehängt)
- Web Service - Methoden mit `soap_fault()` erweitern
- Client - Proxy mit try/catch erweitern

Variante 2

- C#/.NET als Web Service
- PHP/NuSOAP als Client

Variante 2

Beispiele

Variante 2 - Interoperabilität - Datentypen

Web Service arbeitet Document/Literal

- Einfache Datentypen: leere Parameter, leere Antwort, string, integer, float, double, decimal, dateTime, boolean, base64Binary
- Komplexe Datentypen (Arrays): nur empfangen, nicht senden
- Komplexe Datentypen (Structs): nur empfangen, nicht senden
- Komplexe Datentypen (Arrays von Structs): nur empfangen, nicht senden

Web Service arbeitet RPC/Encoded

- Einfache Datentypen: leere Parameter, leere Antwort, string, integer, float, double, decimal, dateTime, boolean, base64Binary
- Komplexe Datentypen (Arrays): string, integer, float, double, decimal, dateTime, boolean, base64Binary
- Komplexe Datentypen (Structs)
- Komplexe Datentypen (Arrays von Structs)

Variante 2 – Debugging und Fehlerbehandlung

- Web Service – Methoden mit `new SoapException(...)` erweitern
- Client - `$client->fault` für Fehler auf der Seite des Web Service
- Client - `$client->getError()` für Fehler auf der Seite des Clients
- Client - `$client->debug_str` liefert alle auf der Client-Seite gesammelte Information