



openHPI: **Web-Technologien** Webservices

Prof. Dr. Christoph Meinel

Hasso-Plattner-Institut

Universität Potsdam

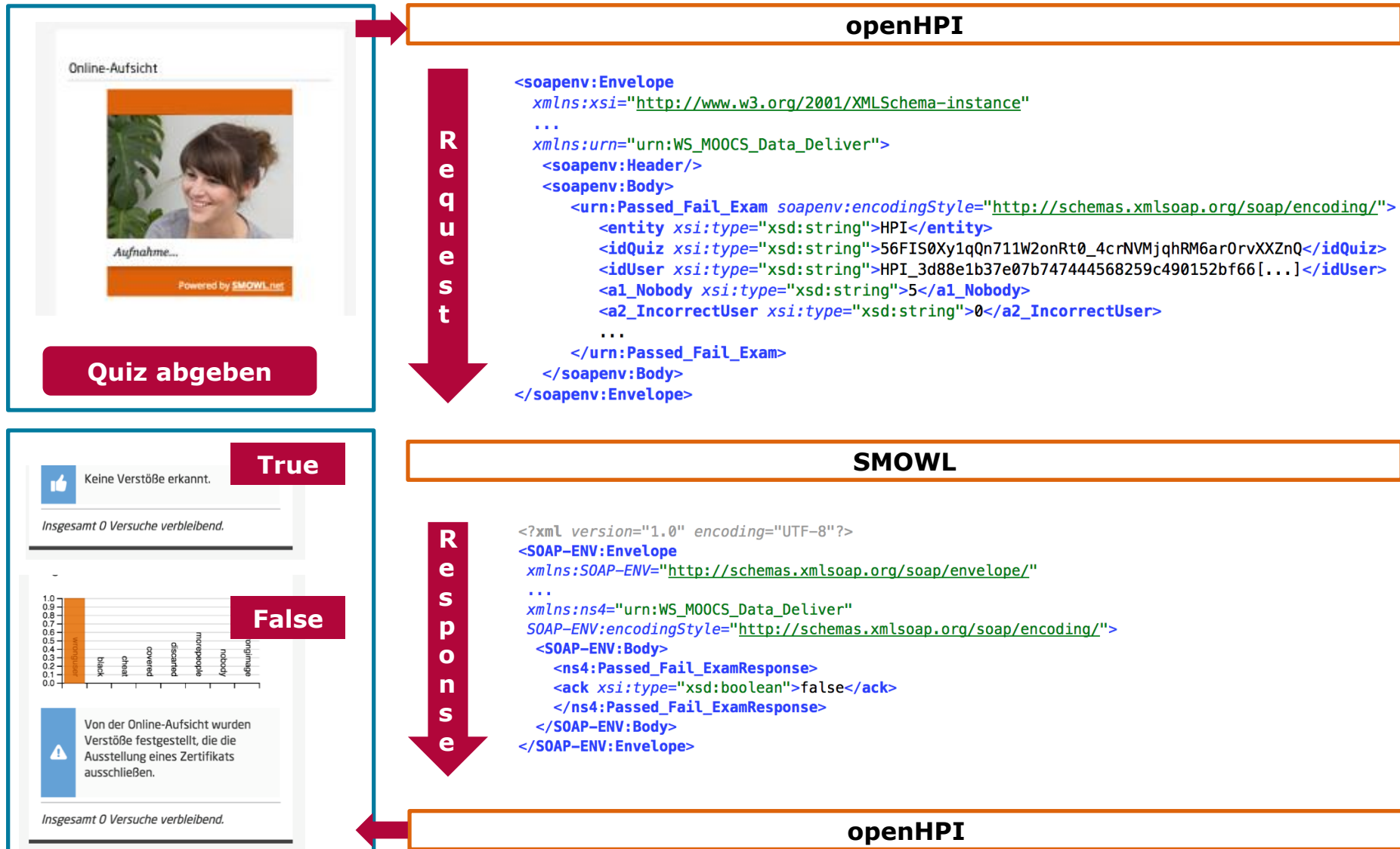
Webservices

- Allgemeiner Ansatz für Entwicklung von über das Web verteilten Anwendungen
- Zugriff auf Dienste verschiedener Anbieter
 - unabhängig von Betriebssystemen, Programmiersprachen und binären Übertragungsprotokollen
- Ermöglichen Kommunikation zwischen verteilten Anwendungen



Webservices

Beispiel: Online-Proctoring bei openHPI mit SMOWL



Webservice-Protokolle 1. Generation

SOAP – XML-basiertes Protokoll

- SOAP Envelopes – Header und Body (wie HTTP)

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

- SOAP Envelope wird als Payload im Body von HTTP-Request/HTTP-Response übermittelt
- Requests in der Regel per HTTP POST
- Aber: SOAP ist nicht auf HTTP als Transportmedium angewiesen

WSDL - Web Service Description Language

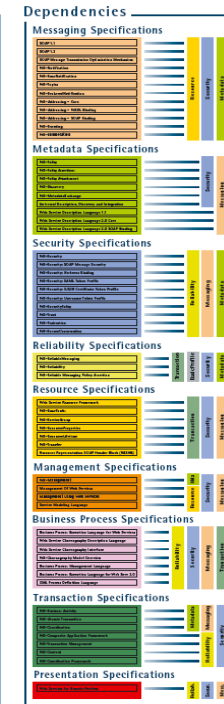
- XML-basierte Beschreibungssprache für Webservice-Schnittstellen
- Definition/Beschreibung von Methoden, Parametern, Rückgabewerten

Webservice-Protokolle 2. Generation

WS*-Protokoll-Suite Webservice-Standards der zweiten Generation

- Absicherung von Web Services
- Komposition mehrerer Web Services (Orchestrierung)
- Zuverlässige Ausführung der Web Services
- Protokolle
 - WS-Security, WS-Policy
 - WS-Coordination, WS-Federation, WS-Notification, ...
- WS* (es gibt sehr, sehr viele WS-Standards ...)
- Spielen hauptsächlich im Enterprise-Bereich eine Rolle, sind im Web weniger verbreitet

► Interoperability Issues



innoQ
innoQ Deutschland GmbH
Haldenstraße 17
D-40880 Ratingen
Phone +49 21 02 77 162 -100

innoQ
innoQ Schweiz GmbH
Gewerkestrasse 11
CH-6330 Cham
Phone +41 41 743 00 00

Webservices | openHPI | Prof. Dr. Christoph Meinel

Schlanke Webservices

Etablierung schlanker Webservices als Alternative zu den WS*-Standards für verteilte Systeme im Enterprise Bereich:
XML-RPC and **REST**

■ **XML-RPC (Remote Procedure Call)**

- Protokoll für Webservices – historischer Vorgänger von SOAP
- einfaches XML-basiertes Protokoll für Aufruf von Methoden über das Web
- Aufrufe erfolgen über HTTP POST – auszuführende Aktion werden in XML definiert

Schlanke Webservices – XML-RPC

XML-RPC example:

Transport per POST

POST /shoppingcart HTTP/1.1

Host: myServer.org

Content-Type: application/xml

<?xml version="1.0"?>

<methodCall>

<methodName> deleteItem </ methodName >

<params>

<param><value><int>244</int></value></param>

</params>

</methodCall>

**Auszuführende Aktion im
Body des Requests**

**Adressat der Aktion
wird ebenfalls im Body
des Requests definiert**

Schlanke Webservices

Etablierung schlanker Webservices als Alternative zu den WS*-Standards für verteilte Systeme im Enterprise Bereich:

XML-RPC and **REST**

■ **REST – REpresentational State Transfer**

- keine Technologie – sondern Architektur
- ROA – Resource Oriented Architecture
- benutzt komplettes HTTP Vokabular: GET, POST, PUT, ...
 - auszuführende Aktion werden direkt per HTTP definiert
- erstmals beschrieben in Dissertation Roy Fielding (2000)
- inzwischen weiter verbreitet als XML-RPC

Schlanke Webservices – REST (1/4)

REST example:

**Auszuführende Aktion
wird direkt mittels der
HTTP-Methode definiert**

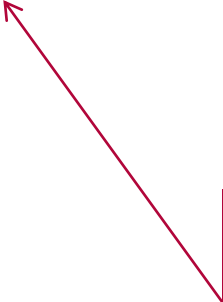


`DELETE /shoppingcart/items/244 HTTP/1.1`

`Host: myServer`

`Content-Type: application/xml`

**Adressat der Aktion
wird im Pfad
definiert**



Schlanke Webservices – REST (2/4)

REST - REpresentational State Transfer

- Web-Applikationen leben von **Web-Ressourcen** (erreichbar über ihre URL)) und ihrer Ausgestaltung, z.B. User, Course, Enrollment, ...
- REST bietet Weg, diese Ressourcen direkt mit Mitteln von HTTP zu manipulieren und zu verknüpfen
- HTTP Pfad bestimmt, welche Ressource manipuliert wird
- HTTP Methode bestimmt, wie Ressource manipuliert wird
 - POST – Ressource wird erzeugt (Create)
 - GET – Ressource wird gelesen (Read)
 - PUT – Ressource wird aktualisiert (Update)
 - DELETE – Ressource wird gelöscht (Delete)
- Im HTTP Body werden nur noch die eigentlichen Daten transportiert (POST/PUT)

Schlanke Webservices – REST (3/4)

Wiederholung: Elemente des HTTP-Protokolls (RFC 2616)

■ HTTP Request

- Methoden: GET, POST, PUT, DELETE
- Pfad: Teil der URL `https://openhpi.de/courses/1/settings`
- Request Header
- Request Body

■ HTTP Response

- Response code: 200 OK, 403 Forbidden, 404 Not Found ...
- Response Header
- Response Body

Schlanke Webservices – REST (4/4)

REST – Design Prinzipien:

- Immer passende HTTP Methode GET, POST, PUT oder DELETE nutzen
- Zustandslosigkeit (Statelessness) – Alle benötigten Daten mit Request bereitstellen
- URLs orientieren sich an Dateistrukturen
- Service registriert den MIME-Type des Requests und sendet Response im passenden MIME-Format zurück

WADL - Web Application Description Language

- XML-basierte Beschreibungssprache
- Entspricht WSDL in den SOAP basierten Webservices