Cloud-Computing Seminar (CLCP) Basistechnologie: Web Services

Selina Raschack

Universität Heidelberg bei Christian Baun

11.5.2009

- Szenario:
 - Es soll ein Standardlösung für eine Dienstleistung angeboten werden
- Problem:
 - Ein Anwender möchten diesen Dienst in bestehende individuelle Geschäftsprozesse integrieren und automatisiert ihn zugreifen
- Lösungsansatz:
 - Dienst als Web Service

Agenda

- Web Services
 - Anforderungen

Web Services

- Eigenschaften
- Architektur
- Definition
- Basistechnologien
 - XMI
 - SOAP
 - WSDL
 - UDDI
- Web Services und Sicherheit
- Zusammenfassung
 - Zusammenfassung
 - Quellen
 - Fragen

Anforderungen¹ [HeZ03]

- Sicherheit
- Performance
- Wartbarkeit
- Zuverlässigkeit

¹Web Services müssen allgemeinen Anforderungen an verteilte Anwendungen genügen. Denn, Dienst-Anbieter und -Nutzer haben jeweils nur Wissen über ein begrenztes Teilsystem und auch nur entsprechenden Einfluss.

Anforderungen (2) [HeZ03]

Sicherheit

Es ist notwendig, dass (i) sensible Daten verschlüsselt werden, und (ii) Benutzer sich ggf. erst gegenüber dem Server authentifizieren, da Systeme, die über ein Netzwerk arbeiten, grundsätzlich leicht angreifbar sind.

Performance

Geschwindigkeiten eines Web Services sind insb. über die Netzinfrastruktur um den Dienst beeinflusst. Zu beachten sind (i) Durchsatz^a und (ii) Latenzzeit^b.

^aDatenmenge pro Zeiteinheit; wichtig, wenn große Dokumente übertragen werden sollen ^bbenötigte Zeit eines Datenpakets, von Knoten A nach B zu kommen; wichtig, wenn viele kleine Remote-Procedure-Control Anfragen

Anforderungen (3) [HeZ03]

Wartbarkeit

Verteilte Systeme haben i.A. die Anforderung an eine lange Laufzeit und damit verbunden einen Mechanismus, flexibel auf Änderungen reagieren zu können. Wichtige Kriterien sind dabei (i) Erweiterbarkeit und (ii) Wartbarkeit im laufenden Betrieb.

Zuverlässigkeit

Die Zuverlässigkeit bezieht sich besonders auf die Erreichbarkeit und das Verhalten im Ausfall einer Hardwarekomponente. Es sind also Mechanismen für ein *fail over* bereit zu stellen.

Eigenschaften [Gfl05]

- Programmierbarkeit
- Selbstbeschreibend
- Kapselung

- lockere Koppelung
- Ortstransparenz
- Protokolltransparenz
- Komposition

Eigenschaften (2) [Gfl05]

Programmierbarkeit

Motivation

Der Dienst ist über eine *programmierbare Schnittstelle* erreichbar. Er dient also der *Kommunikation zwischen Anwendungen* und benötigt keine graphische Oberfläche.

Selbstbeschreibend

Der automatisierten Dienstzugriff wird durch Metadaten begleitet, die zur Laufzeit ausgewertet werden können. Sie gibt ein Web Service z.B. über sich selbst Auskunft, in welchem Format Nachrichten ausgetauscht werden können oder welchen Antworttypen zu erwarten sind.

Eigenschaften (3) [Gfl05]

Kapselung

Motivation

Der Dienst ist *unabhängig*, in sich *abgeschlossen* und erfüllt eine *genau definierte* Aufgabe.

lockere Koppelung

Die Kommunikation zwischen zwei Diensten erfolg über den Austausch von Nachrichten. Dabei bleiben Implementierungsdetails verborgen und es sind keine Informationen notwendig, die über die Dauer des Datenaustausches hinweg gehen.^a

^aProblem der Erreichbarkeit: Was passiert, wenn ein Dienst eigentlich nicht mehr da ist?

Eigenschaften (4) [Gfl05]

Ortstransparenz

Motivation

Der Dienst ist zu jeder Zeit und von einem beliebigen Ort aus aktivierbar.

Protokolltransparenz

Der Dienst basiert auf der Internet-Protkollsuite. Es können damit unterschiedliche Protokolle des Application Layers^a unterstützt werden.

Komposition

Der Dienst kann (i) in weitere Teildienste untergliedert oder (ii) als Teildienst in einen komplexen Dienst integriert werden.

^aAnwendungsschicht des OSI-Referenzmodells

Architektur [KuW02][HeZ03]

Schichten:

Motivation

Web Service Applikation

Service Veröffentlichung / Service Erreichbarkeit

Service Beschreibung

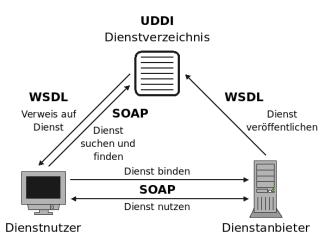
Formulieren / Verschicken von Nachrichten

Internet Transport Protokolle
Internet Netzwerk Protokolle

Rollen:



Architektur (2)



Definition²

Web Services

Ein Web Service ist eine **Dienst, der** im Rahmen einer verteilten Anwendung, eine konkrete Teilaufgabe umsetzt. Dieser Dienst kann insbesondere ortsungebunden und automatisiert angesprochen und sein Ergebnis ebenfalls ortsungebunden und automatisiert weiterverarbeitet werden.

²Vgl. auch [Jec0x]. Die im Nachfolgenden gegebene Definition dient einzig dem gemeinsamen Verständnis von Web Services im Rahmen dieses Seminars und erhebt keinen Anspruch auf Allgemeingültigkeit.

XML Basics [RoG02][HeZ03]

- XML steht für Extensible Markup Language
- Ziel: Daten kapseln und strukturieren

XML Merkmale [RoG02][HeZ03]

Wohlgeformtheit

Motivation

Die Wohlgeformtheit bezieht sich auf eine konsequente Strukturierung und ist obligatorisch. Konkret sind z.B. alle geöffneten Tags auch wieder zu schließen oder Positionsvorgaben einzuhalten.

Validierung

Xml-Dokumente können gegen *Dokumenten Typ Definitionen* validiert werden. In den DTD's sind z.B. Beschrieben, welche Elemente in welcher Reihenfolge vorkommen dürfen, wann sie ggf. vorkommen müssen oder welche Attribute erfoderlich sind. Für XML sind auch sog. XML-Schema zur Validierung möglich. Der Vorteil, diese sind selbst auch wieder in XML.

XML Arten des Zugriffs [RoG02][HeZ03]

Sequentieller Zugriff

Der XML-Baum wird von der Wurzel beginnend zeilenweise eingelesen. Dieses Verfahren bietet sich z.B. an, wenn sowieso das gesamte Dokument eingelesen und verarbeitet werden soll.

Wahlfreier Zugriff

Es kann zu einem beliebigen Zeitpunkt auf ein beliebiges Element im XML-Baum zugegriffen werden. Dazu wird zunächst das komplette Dokument in den Hauptspeicher geladen und Programmiersprachen intern abgebildet.

XML-Beispiel: Online-Bewerbungsystem



zwei Beispielregeln:

- Default Summe Frage 1 + 2 'Staatsexamen'setze Summe.
- Wenn Container 'Staatsexamen' > 50 Punkte, 'Einladen' erhöhe 1.

XML-Beispiel: Online-Bewerbungsystem (2)

Motivation

ullet Default — Summe Frage 1+2 — 'Staatsexamen'setze Summe.

XML-Beispiel: Online-Bewerbungsystem (3)

Motivation

• Wenn — Container 'Staatsexamen' > 50 Punkte, — 'Einladen'erhöhe 1.

```
<regel>
    <vergleich>
      <untersuchselemente>
        <frage>FR 00041</frage>
        <frage>FR 00042</frage>
      </untersuchselemente>
      <operator>groesser</operator>
      <punkte>50</punkte>
    </vergleich>
    <addiere>
      <container>Einladen</container>
      <optionswert>1</optionswert>
    </addiere>
  </regel>
</regeln>
```

SOAP Basics [KuW02][HeZ03]

- SOAP steht für Simple Object Access Protocol
- Ziel: Austausch von Daten über XML

SOAP Struktur: der Umschlag [KuW02][HeZ03]

SOAP Header

Motivation

Der Header enthält die verschiedenen META-Daten und macht SOAP damit von Übertragungsprotokollen unabhängig. Hier können auch Daten zur Authentifizierung verpackt werden.

SOAP Body

Der Körper enthält die eigentlichen Nutzdaten. Das können z.B. ein Methodenaufruf und die beötigten Parameter oder dessen Rückgabewert sein.

SOAP-Beispiel: Auswertung anfordern

- Dienst hat Methode: getAuswertung()
- Diese erwartet eine Bewerbungsid als Parameter
- Nutzer soll sich authentifizieren, damit er die Auswertung nur bekommt, wenn er auch berechtigt ist
- SOAP-Anfrage: Header: Authentifizierung, Body: Methodenaufruf
- SOAP-Antwort: Body: Auswertung als Rückgabewert

SOAP-Beispiel: Auswertung anfordern (2)

```
<SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlSOAP.org/SOAP/envelope">
  <SOAP-ENV:Header>
    <auth:authentifkation
      xmlns:auth="http://beispielfirma.com/basys2rules/auth">
      <auth:user>mustermann</auth:user>
      <auth:pwd>musterpasswort</auth:pwd>
    </auth:authentifkation>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <meth:getAuswertung xmlns:meth="http://beispielfirma.com/basys2rules/</pre>
      meth">
      <br/>bewerbungsid>musterid</bewerbungsid>
    </meth:getAuswertung>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP-Beispiel: Auswertung anfordern (3)

WSDL Basics [KuW02][HeZ03]

- WSDL steht f
 ür Web Service Description Language
- Ziel: Schnittstelle zwischen Dienst und Client definieren
- Merkmale:

- Schnittstellendefinition
- Beschreibung der Datentypen
- Bindungsinformationen an Transportprotokolle
- Informationen für Auffinden benutzter Dienste

WSDL Dokumentenaufbau [KuW02][HeZ03]

- Spezifikation: Definitions
 - types
 - message
 - portType
 - binding
 - service
- Optionale Elemente:
 - documentation
 - import

WSDL-Beispiel: Auswertung anfordern

- Types:
 - komplex: einen Benutzer für die Authentifizierung, eine Auswertung
 - Einfach: Bewerbungsid

```
<?xml version"1.0"?>
<xsd:schema targetNamespace="http://beispielfirma.com/basys2rules/types"</p>
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
  <xsd:element name="auswertung">
    <xsd:complexTvpe>
      <xsd:all>
        <xsd:elment name="bewerbungsid" type="xsd:string">
        <xsd:elment name="gesamtpunkte" type="xsd:integer">
      </xsd·all>
    <xsd:complexType>
  </xsd·element>
</schema>
```

• Messages:

Motivation

• getAuswertung, getAuswetungResponse

```
<message name="getAuswertung">
    <part name="bewerbungsid" type="xsd:string">
    </message>
<message name="getAuswertungResponse">
    <part name="auswertung" type="xsd:auswertung">
    </message></part
```

WSDL-Beispiel: Auswertung anfordern (3)

- portType:
 - input: getAuswertung
 - output: getAuswertungResponse

```
<portType name="auswertungsPort">
    <operation name="auswertung">
        <input message="getAuswertung">
        <output message="getAuswertungResponse">
        </operation>
    </portType>
```

Binding:

Motivation

1 Soap-binding für unsern Port auswertungsPort

```
<binding name="auswertungsBinding" type="auswertungsPort">
  <SOAP:binding transport="hhtp://schemas.xmlSOAP.org/SOAP/HTTP"
    style="rpc"/>
  <operation name="auswertung">
  <SOAP:operation style="rpc">
    <input>...</input>
    <output>...</output>
  </operation>
</binding>
```

WSDL-Beispiel: Auswertung anfordern (5)

- Services:
 - 1 Soap-binding für unsern Port auswertungsPort

```
<service name="Bewerberauswertung">
  <port name="auswertungsPort" binding="auswertungsBinding" />
    <SOAP:address location="http://beispielfirma.com/bewerberauswertung">
  </port>
</service>
```

UDDI Basics [KuW02][HeZ03]

- UDDI steht für Universal Descritpion and Discovery Interface
- Ziel: Publizieren und Auffinden von Diensten

UDDI Merkmale [KuW02][HeZ03]

Service-Broker-Architektur

Motivation

Die Service Oriented Architecture (SOA) ist eine verteilte Architektur, in der insgesamt drei agierende Instanzen (Rollen) definiert sind. (i) Service Broker oder Service Registry stellt Unterstützungsfunktionen für das Veröffentlichen und Auffinden der offerierten Webservices bereit. (ii) Service Requester, der Nutzer: findet die entsprechenden Webservices mithilfe der Service Registry und nutzt anschließend die bereit gestellten Dienste. (iii) Service Provider, der Anbieter: stellt Dienst zur Verfügung und veröffentlicht über den Service Broker.

Verzeichnis Web Service

Die abgelegten Informationen lassen sich in drei wesentliche Felder abgrenzen: (i) Kontaktionformationen (*white pages*), (ii) Service Informationen (*yellow pages*) und (iii) Technische Informationen (*green pages*).

UDDI Datenmodell [KuW02][HeZ03]

- BusinessEntity
- PublishersAssertion
- BusinesService
- BindingTemplate
- TModel

UDDI Datenmodell (2) [KuW02][HeZ03]

BusinessEntity

Motivation

Repräsentiert eine grundlegende Geschäftsinformation: (i) Kontaktinfos, (ii) Kategorisierung, (iii) Beschreibung, (iv) Beziehung zu anderen Objekten und (v) Identifier.

Publishers Assertion

Repräsentiert eine Beziehung zwischen zwei businessEntity-Strukturen.

BusinesService

Repräsentiert eine einzelne logische Klassifikation von Diensten und dient dazu, einzelne Dienstleistungen eines Unternehmens zu beschreiben.

UDDI Datenmodell (3) [KuW02][HeZ03]

BindingTemplate

Dient dazu, einen Dienst tatsächlich zugreifbar zu machen. Enthält (i) eine Beschreibung (description), (ii) Zugriffsknoten (accessPoint) und optional (iii) technische Beschreibungen.

TModel

Dient der eindeutigen Indetifikation eines Dienstes und legt die Kommunikation zwischen Web Services fest. Wird oft heran gezogen, um zu prüfen, ob ein Dienst zu den persönlichen Erfordernissen des Nutzers passt.

UDDI Nachrichten [HeZ03]

- Detailinformationen abfragen
- Suchen

- Authentifikation
- Hinzufügen / Ändern
- Löschen

Anforderungen [HeZ03]

Verschlüsselung

- Authentifizierung
- Autorisierung
- Herkunftsnachweis

Anforderungen (2) [HeZ03]

Verschlüsselung

Motivation

Ziel ist es, die angebotenen Services und Daten nicht beliebig von Nutzern abfragbar und einsehbar zu halten. Sondern diese vor unberechtigten Zugriffen zu schützen.

Authentifizierung

Beschreibt eine reine Zugangskontrolle. Meißt wird eine Kombination aus Benutzer und Passwort verwendet.

Anforderungen (3) [HeZ03]

Autorisierung

Motivation

Definiert verschiedene Rollen mit unterschiedlichen Rechten und legt damit eine detaillierte Zugriffsberechtigung fest.

Herkunftsnachweis

Zeichnet den Sender einer Nachricht aus. Stellt also insbesondere die Echtheit einer Quelle aus.

Kritikpunkte [HeZ03]

- SOAP-Nachrichten bilden auch die Struktur der Daten mit ab
- Kein einheitliches Sicherheitsnetzwerk für Client und Server gemeinsam gewährleistet
- Softwareverschlüsselung rechenintensiv
- Beziehung zwischen Kommunikationspartnern, die ggf. vorgibt, welche Sicherheitsmechanismen wie angewandt sind

Zusammenfassung

Stärken:

- Standardaufgaben mit Standardlösungen bearbeiten
- Dienste insbesondere zur automatisierten Nutzung
- Schwächen:
 - Web Services und Sicherheit
 - Einfluss auf Fehlererkennung und -behandlung
- Abgrenzung zu anderen Technologien:
 - Dienen dazu einen Dienst softwareseitig zu entkoppeln und ggf. zu outsourcen

Quellen

- [HeZ03] HEIN Mannfred, ZELLER Henner: Java Web Services Entwicklung plattformübergreifender Dienste mit J2EE, XML und SOAP, ADDISON-WESLEY, Pearson Education Deutschland GmbH, 2003
- [KuW02] KUSCHKE Michael, WÖLFEL Ludgar: Web Services kompakt, Spektrum, Akademischer Verlag, 2002
 - [RoG02] ROTTACH Thilo, GROß Sascha: XML kompakt, Spektrum, Akademischer Verlag, 2002
 - [Gfl05] Gesellschaft für Informatik EV: Web Services, http://www.gi-ev.de/no_cache/service/ informatiklexikon/informatiklexikon-detailansicht/ meldung/web-services-95/, 2005
 - [Jec0x] JECKLE Mario: Nachgestichelt: Wie schreibt man eigentlich
 Web Service?!,
 http://www.jeckle.de/webServices/index.html

Vielen Dank für die Aufmerksamkeit!