

Web Services

XML, WSDL, SOAP und UDDI
Einblicke und Ausblicke

Finden von Web Services – UDDI Registry

Universal Description, Discovery, and Integration

- Inhalt
 - Die UDDI Organisation
 - Die Konzepte von UDDI
 - Wie funktioniert UDDI?
 - Das UDDI Datenmodell
 - Generische Daten
 - Business Entität
 - Bindungs-Template
 - Das tModell
 - UDDI SOAP API's
 - Abfrage API's
 - API's zum Publizieren

Beschreibung

Finden von Web Services – UDDI Registry

- Inhalt
 - ..
 - Einsatz Szenarios
 - Mutieren der Registry
 - Informationen abfragen
 - Einsatz von WSDL
 - UDDI für den Privatgebrauch
 - UDDI SOAP Unterstützung
 - SOAP
 - Unicode

Finden von Web Services – UDDI Registry

Einleitung

- *UDDI findet die Web Service Adresse und andere Informationen über ein Business.*
 - UDDI's Aufgabe ist es, Interessierten die nötige Informationen zur Verfügung zu stellen, damit die Dienste genutzt werden können.
 - Wie sieht dies im Alltag aus?
 - Sie schlagen vielleicht in den gelben Seiten nach
 - Sie kennen die Adresse einer Firma und deren Profil (Produkte, Dienstleistungen, News).

Finden von Web Services – UDDI Registry

Einleitung

- *UDDI ist Teil der Web Service Infrastruktur.*
 - Damit Ihre Programme oder Programme auf Ihrem Rechner mit anderen Programmen auf dem Web kommunizieren kann, muss Ihnen das andere Programm bekannt sein.
 - UDDI offeriert
 - White Pages
 - Yellow Pages
 - Green Pages
(mehr darüber später).
 - Zudem steht ein API zur Verfügung, mit dessen Hilfe die UDDI Services genutzt werden können.

Finden von Web Services – UDDI Registry

Einleitung

- *UDDI unterstützt das Auffinden von Web Service Interfaces.*
 - UDDI stellt auch Suchmöglichkeiten zur Verfügung, um Services zu finden.
 - UDDI verwendet replizierte Registrier-Datenbanken.
 - Verschiedene Hersteller (SAP, MS, ...) bieten UDDI Dienste an: *Operator Sites*.
 - Jeder Anbieter stellt seine Informationen zur Verfügung (analog zu DNS).

Finden von Web Services – UDDI Registry

Die UDDI Organisation

- *Mehr als 300 UDDI Firmen sind Mitglied der UDDI.ORG.*
 - UDDI wurde von MS, IBM und Ariba gestartet.
 - Die Working Group Members haben das Sagen.
 - Ab UDDI v3 wird die Spezifikation bei einem unabhängigen Standardisierungs-Gremium eingereicht:
 - W3C oder OASIS
 - Die Operators können sowohl produktive als auch Test-UDDI Sites anbieten.
 - Ab v3 werden die Sicherheitsanforderungen verschärft werden.

Finden von Web Services – UDDI Registry

Das UDDI zugrunde liegende Konzept

- *UDDI besteht aus Registrierung und Auffinden.*
 - Die öffentliche UDDI Registry funktioniert analog zu DNS (Internet Domain Name Service).
 - Anbieter können sich bei einem beliebigen Anbieter registrieren.
 - In regelmässigen Abständen werden die Datenbestände ausgetauscht und synchronisiert.
 - *Updates müssen am Original-Site ausgeführt werden!*
 - **Sicherheitsaspekte sind ein zentrales Thema:**
 - Informationen über den Anbieter sind nur am Original-Site vorhanden.
 - Die Autorisierung für den Ersteintrag muss beantragt werden.
 - **Die Datenqualität ist ein zentrales Thema:**
 - Ist der Anbieter echt? Wie heisst er? Besitzt er eine ID?, welche Art Service wird angeboten?...

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

- *UDDI enthält White Pages, Yellow Pages und Green Pages.*
 - UDDI Informationen gehören einer der 3 Kategorien an:
 - **White Pages:** Business Name und Adresse, Kontaktinformationen, Web Site Adresse, Data Universal Numbering System (DUNS) oder andere Nummernsysteme.
 - **Yellow Pages:** Business Typus, Ort (Adresse), Produkte, Industriezweig, Business ID, ...
 - **Green Pages :** Technische Informationen zum Business (wie kann der Dienst genutzt werden?), Pointer zur WSDL Beschreibung, ID des Services ...

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

- *UDDI Datenstrukturen sind XML Schemas.*
 - UDDI muss Klassifikations- und Identifikations-Informationen speichern.
 - IRS Industry Codes
 - Dun & Bradstreet DUNS Nummerierung
 - Produktcodes
 - Geografische Codes ...
 - Klassifikation und Identifikation werden mithilfe von *property bags* oder unstrukturierten Datensequenzen erfasst.
 - North American Industry Classification System (NAICS)
 - `,http://www.census.gov/epcd/www/naics.html`
 - ISO (geogr. Regionen..)
 - `,http://www.din.de/gremien/nas/nabdi/iso3166ma`
 - Universal Standard Products and Services Classification (UNSPSC)
 - `,http://www.unspsc.org`
 - Jeder Operator kann zusätzliche Validationen durchführen!

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell

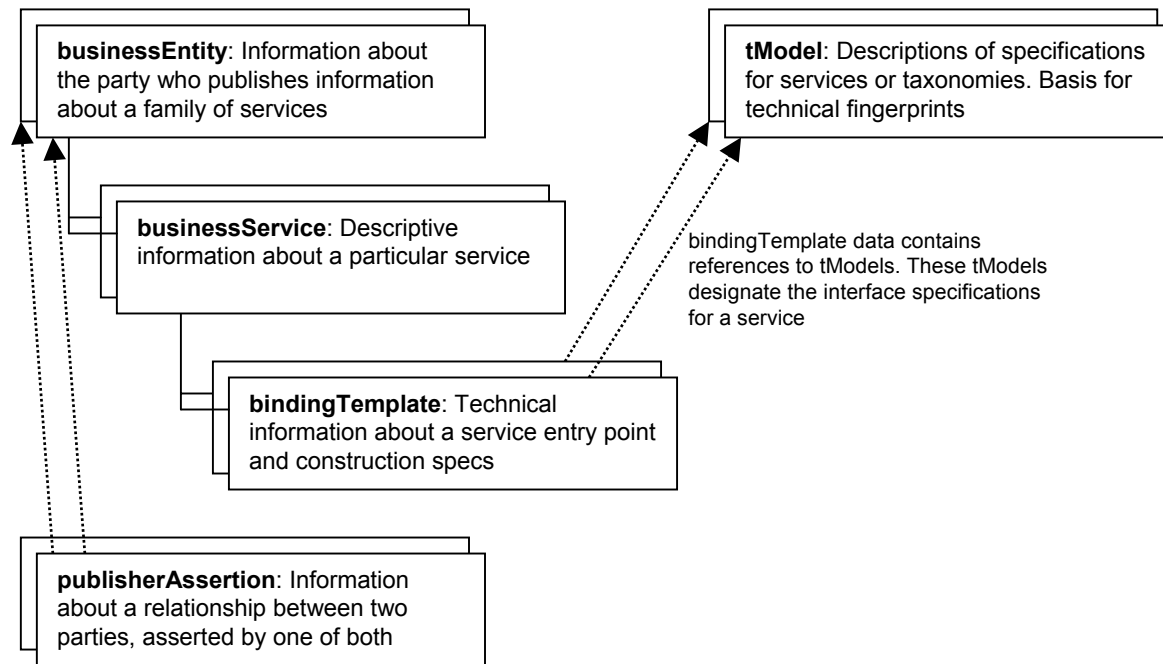
- *UDDI identifiziert fünf grundlegende Datenstrukturen.*
 - **businessEntity**: top-level Struktur, das Business beschreibend. Andere Informationen werden mit dieser Entität verkettet.
 - **businessService**: Name und Beschreibung des Services
 - **bindingTemplate**: Informationen über den Service, wie Eingangsadresse.
 - **tModel**: ein fingerprint (Sammlung von Informationen, welche den Dienst eindeutig charakterisieren). Die Information kann auch selektiv abgefragt werden (Query).
 - **publisherAssertion**: Beziehungs-Entität, welche die Beziehung zwischen mehreren businessEntity Strukturen beschreibt (Niederlassung, Abteilung, ...).

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell

- *Das UDDI Grundmodell.*
 - UDDI DataStructure-V2.00-Open-20010608.doc

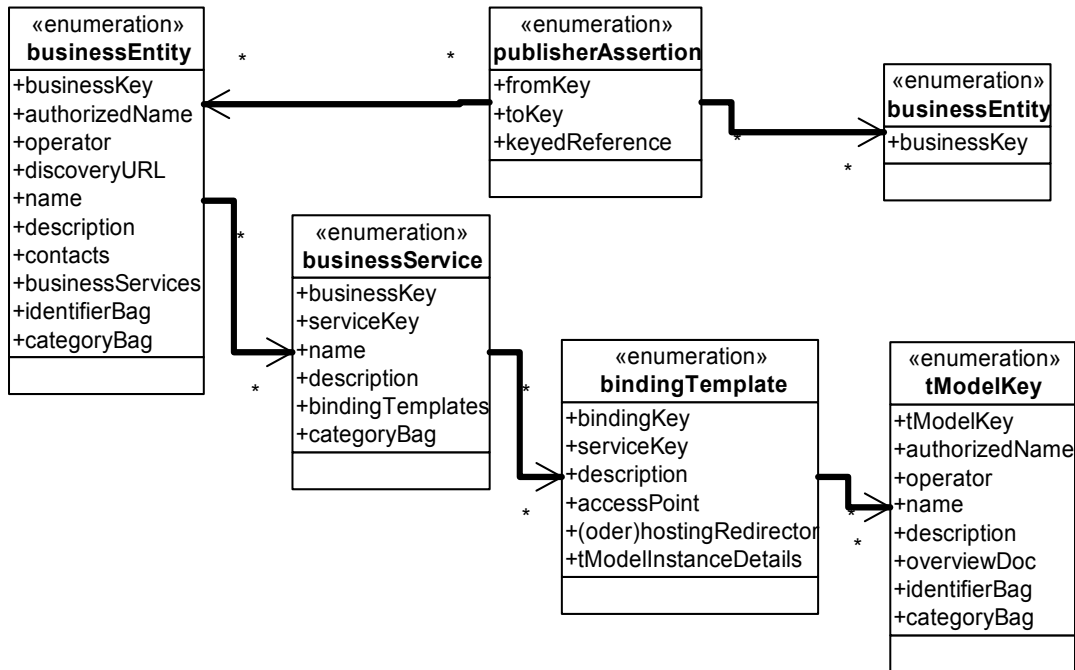


Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell

- *Das UDDI Grundmodell.*



Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Generische Daten

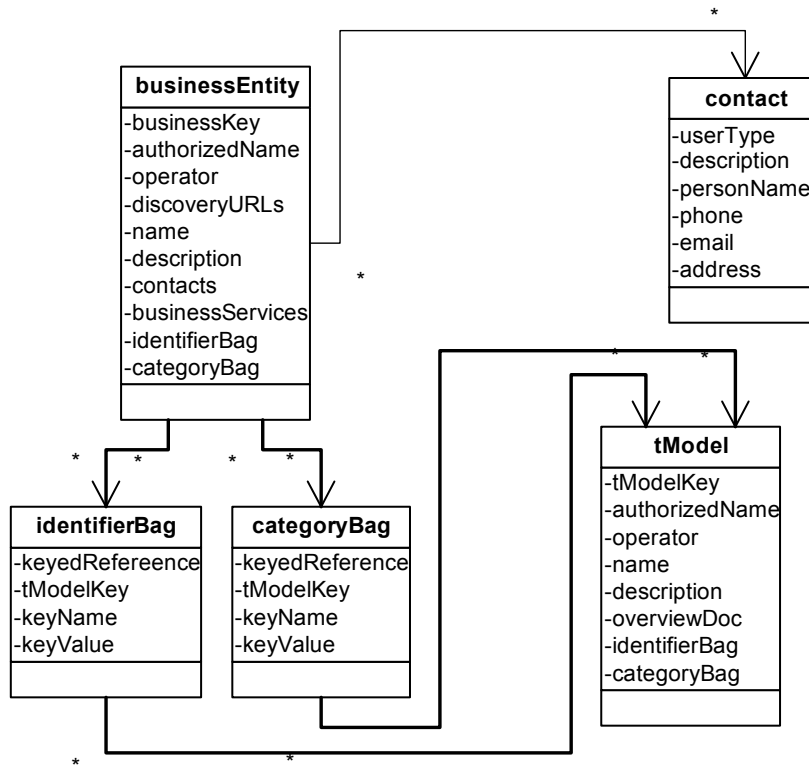
- *UDDI akzeptiert mehr oder weniger jede Art Daten.*
 - Die Datentypen sind so flexibel definierbar, dass jede Art von Daten darstellbar ist.
 - Auch die XML Schema Strukturen beschreiben keinerlei Einschränkung bezüglich der Datenspeicherung.

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Generische Daten

- *UDDI generische Daten.*



Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Business Entity

- *Die Business Entity Struktur ist die Information, mit der Abfragen in der Regel gestartet werden.*
 - Die XML Definition in Form eines XML Schemas definiert eine komplexe Sequenz mit mehreren Attributen.
 - **BusinessKey** ist eine Muss-Datenstruktur
 - Beim Kreieren der Struktur wird ihr ein eindeutiger Schlüssel (UUID) zugeordnet.
 - Die minimale Information, die zum Kreieren benötigt wird, ist der Business Name.

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Business Entity

- *Die Business Entity Struktur ist die Information, mit der Abfragen in der Regel gestartet werden.*

- *<element name = "businessEntity">
 <complexType>
 <sequence>
 <element ref = "discoveryURLs" minOccurs = "0"/>
 <element ref = "name" maxOccurs = "unbounded"/>
 <element ref = "description" minOccurs = "0"
 maxOccurs = "unbounded"/>
 <element ref = "contacts" minOccurs = "0"/>
 <element ref = "businessServices" minOccurs = "0"/>
 <element ref = "identifierBag" minOccurs = "0"/>
 <element ref = "categoryBag" minOccurs = "0"/>
 </sequence>
 <attribute ref = "businessKey" use = "required"/>
 <attribute ref = "operator"/>
 <attribute ref = "authorizedName"> </complexType>
 </element>*

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Binding Template

- *Das Binding Template enthält den Zugangspunkt (URL).*
 - Das Template enthält Informationen, die für den physischen Zugriff benötigt werden.
 - Pro Service können mehrere Zugangspunkte definiert werden.
 - Die URL kann unterschiedliche Protokolle umfassen:
 - mailto:
 - http:
 - https:
 - ftp:
 - fax:
 - phone:
 - other:
wobei jeweils gültige Ergänzungen vorhanden sein müssen
(bei mailto: eine gültige Mail-Adresse).

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – Binding Template

Das Binding Template Schema.

- *<element name = "bindingTemplate">
 <complexType>
 <sequence>
 <element ref = "description" minOccurs = "0"
 maxOccurs = "unbounded"/>
 <choice>
 <element ref = "accessPoint" minOccurs = "0"/>
 <element ref = "hostingRedirector" minOccurs = "0"/>
 </choice>
 <element ref = "tModelInstanceDetails"/>
 </sequence>
 <attribute ref = "bindingKey" use = "required"/>
 <attribute ref = "serviceKey"/>
 </complexType>
</element>*

Finden von Web Services – UDDI Registry

Wie arbeitet UDDI?

Das UDDI Datenmodell – tModel

- *Ein tModel identifiziert einen Web Service eindeutig.*
 - UDDI verwendet das tModel, um Metadaten über einen Web Service auszutauschen:
 - Web Service Beschreibung
 - Ein Pointer auf eine WSDL Beschreibung
 - UDDI kann auch nicht WSDL Pointer verarbeiten
 - UDDI v1 wurde vor der Definition von WSDL eingeführt.
 - Das tModel entspricht grob einer WSDL Beschreibung.
 - Das tModel kann auch (als Beschreibung eines Access Points) für mehrere Services benutzt werden.

Finden von Web Services – UDDI Registry

UDDI SOAP API's

- *UDDI API's lassen sich in zwei Kategorien einteilen:*
 - *solche, die sich mit dem Publizieren,*
 - *und solche, die sich mit dem Konsumieren befassen.*
 - Publizieren:
 - Eintragen, mutieren, löschen der UDDI Information
 - Diese Aktivitäten müssen autorisiert werden
 - Konsumieren:
 - Suchen von Business Informationen, Queries, ...
 - Falls SOAP eingesetzt wird, muss dies mittels HTTP **POST** geschehen.
 - **Verschiedene UDDI Versionen werden über unterschiedliche Namensräume erkennbar:**
 - **xmlns=„urn:uddi-org:api_v2“**

Finden von Web Services – UDDI Registry

UDDI SOAP API's

Inquiry API's

- *Inquiry API's erlauben Browsen und Drill Down.*
 - Inquiry API's unterstützen unterschiedliche Kriterien.
 - Start mit einer generischen Suche
 - Verfeinerung mittels weiterer API's
 - Kriterien:
 - Business Name, Kategorie, Identifikation, ...
 - Inquiry API Funktionen
 - *find_binding*
 - *find_business*
 - *find_relatedBusinesses*
 - *find_service*
 - *find_tModel*
 - *get_bindingDetail*
 - *get_businessDetail*
 - *get_businessDetailExt*
 - *get_serviceDetail*
 - *get_tModelDetail*

Finden von Web Services – UDDI Registry

UDDI SOAP API's

Inquiry API's

- *Inquiry API's Beispiel.*

– **find_binding**

- Die `find_binding` API Methode liefert eine `bindingDetail` Message, welche null oder mehrere `bindingTemplate` Strukturen enthält, welche den Kriterien der Argumenteliste genügen.

- **Syntax:**

```
– <find_binding serviceKey="uuid_key"  
  [maxRows="nn"] generic="2.0"  
  xmlns="urn:uddi-org:api_v2" >  
  [<findQualifiers/>]  
  <tModelBag/>  
  </find_binding>
```

- Referenz:

- <http://www.uddi.org/pubs/ProgrammersAPI-V2.00-Open-20010608.pdf>

Finden von Web Services – UDDI Registry

UDDI SOAP API's

Publisher API's

- *Publisher API's speichern, mutieren und löschen Registry Informationen.*
 - Falls kein UUID Key angegeben wird, nimmt UDDI an, dass es sich um die erste Speicherung, den Ersteintrag handelt.
 - Publishing API Function Reference
 - *add_publisherAssertions*
 - *delete_binding*
 - *delete_business*
 - *delete_publisherAssertions*
 - *delete_service*
 - *delete_tModel*
 - *discard_authToken*
 - *get_assertionStatusReport*
 - *get_authToken*
 - *get_publisherAssertions*
 - *get_registeredInfo*
 - *save_binding*
 - *save_business*
 - *set_publisherAssertions*
 - *save_service*
 - *save_tModel*

Finden von Web Services – UDDI Registry

UDDI SOAP API's

Inquiry API's

- *Inquiry API's Beispiel.*

- **get_registeredInfo**

- Der **get_registeredInfo** API Call wird eingesetzt, um eine Kurzliste aller **businessEntity** und **tModel** Daten zu erhalten.

- **Syntax:**

- ```
<get_registeredInfo generic="2.0"
 xmlns="urn:uddi-org:api_v2">
 <authInfo/>
</get_registeredInfo>
```

- **Argumente:**

- **authInfo**: Muss-Argument, welches den Authentifizierungs-Token enthält (**get\_authToken** API Call).

- **Rückgabe:**

- eine **registeredInfo** Message mit Business Informationen in einer oder mehreren **businessInfo** Elements und **tModel** Information in einer oder mehreren **tModelInfo** Elemente.
    - Referenz: <http://www.uddi.org/pubs/ProgrammersAPI-V2.00-Open-20010608.pdf>

# Finden von Web Services – UDDI Registry Einsatz Szenarios

- *Beispiel-Szenario.*
  1. Wahl eines Operators (Registry Host)
    - Jeder Operator hat seine eigenen Zugangsbedingungen und Autorisierungsverfahren.
  2. Bauen oder beschaffen eines UDDI Clients.
    - Viele Operators stellen ihre eigenen Clients zur Verfügung.
  3. Authentifizierungs-Token beschaffen.
  4. Registrieren der Information, soviel wie nötig.
  5. Freigabe des Authentifizierungs-Tokens.
  6. Inquiry API: Test der Suche und Abfrage
  7. tModel Angaben vervollständigen
  8. Allenfalls Business Informationen mutieren.

# Finden von Web Services – UDDI Registry

## Einsatz Szenarios

### Updaten der Registry

- *Registrierung mithilfe einer SOAP Message.*

- `POST /save_business HTTP/1.1`  
`Host: www.whoopy.com`  
`Content-Type: text/xml; charset=„utf-8“`  
`Content-Length: ...`  
`SOAPAction: „save_business“`

```
<?xml version=„1.0“ encoding=„utf-8“ ?>
<envelope
xmlns=„http://www.xmlsoap.org/soap/envelope/“>
 <body>
 <save_business generic=„2.0“
xmlns=„urn:uddi-org:api_v2“>
 <businessKey=„“></businessKey>
 <name>Whoopy</name>
 <description>Super, Toll, Whoopy
</description>
 <identifierBag> ...
 </save_business>
 </body>
</envelope>
```

Wird generiert



# Finden von Web Services – UDDI Registry

## Einsatz Szenarios

### Informationen Abfragen

- *Kunden fragen die Registry mit Hilfe von SOAP ab.*

- `POST /get_businessDetail HTTP/1.1`  
`Host: www.whoopy.com`  
`Content-Type: text/xml; charset=„utf-8“`  
`Content-Length: ...`  
`SOAPAction: „get_businessDetail“`

```
<?xml version=„1.0“ encoding=„utf-8“ ?>
<envelope
xmlns=„http://www.xmlsoap.org/soap/envelope/“>
 <body>
 <get_businessDetail generic=„2.0“
xmlns=„urn:uddi-org:api_v2“>
 <businessKey=„“>
 C90D731D-777D-....
 </businessKey>
 </get_businessDetail>
 </body>
</envelope>
```

muss stimmen  
Sonst muss man  
die Infos  
abfragen

# Finden von Web Services – UDDI Registry

## Einsatz Szenarios

### UDDI und WSDL

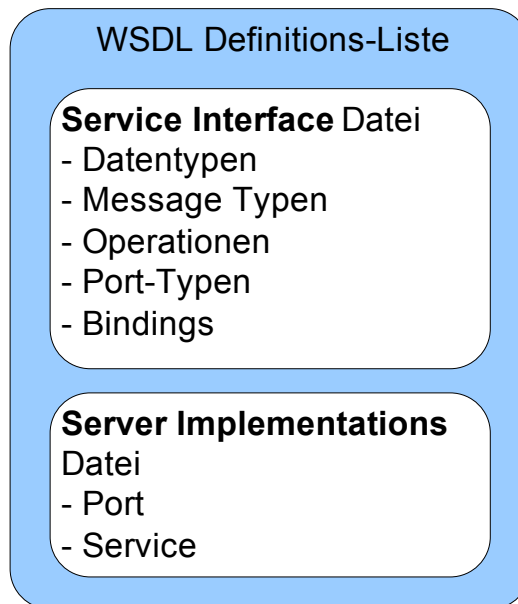
- *WSDL Informationen gehören zu den generischen Service Informationen und spezifischen Bindungsinformationen.*
  - UDDI kann leicht so eingesetzt werden, dass WSDL Spezifikationen verwendet werden können.
  - WSDL wird in UDDI auf folgende Informationen abgebildet
    - **businessService**
      - Generische Informationen über den Service
    - **bindingTemplate**
      - Enthält Netzwerk-Adressen des Services und besitzt 1..n tModels
    - **tModel**
      - Identifizieren den Dienst eindeutig

# Finden von Web Services – UDDI Registry

## Einsatz Szenarios

### UDDI und WSDL

- *WSDL Informationen können gemäss den Elementen von UDDI aufgeteilt werden.*



# Finden von Web Services– UDDI Registry

## Einsatz Szenarios

### UDDI für den Privatgebrauch

- *UDDI kann auch innerhalb der Firewall eingesetzt werden*
  - In einem privaten UDDI können Applikationen registriert und „gefunden“ werden.
    - In einer UDDI Registry können Metadaten über Applikationen gespeichert werden.
  - UDDI könnte somit als zentrale Drehscheibe eines Unternehmens eingesetzt werden, für Informationen über intern verfügbare Services, auch Integrations-Services.

# Finden von Web Services – UDDI Registry

## Einsatz Szenarios

### UDDI Support für SOAP und Unicode

- *UDDI v2 enthält SOAP über HTTP*
  - Im Beispiel weiter vorne:
    - **SOAPAction**: „**save\_business**“
  - SOAP Headers werden nicht unterstützt.
    - Diese werden einfach ignoriert.
    - Falls die SOAP Message einen „required header“ enthält, wird von UDDI ein **mustUnderstand** Fehler ausgelöst.
  - SOAP Encoding wird nicht unterstützt
    - UDDI löst eine Fehlermeldung **E\_unsupported** aus und sendet in der SOAP **errinfo** erklärenden Text.
  - UDDI Operators verwenden UTF-8 Unicode
    - Alle Requests müssen diese Codes unterstützen.



# Zugriff auf Web Services - SOAP

## Referenzen

- Artikel und Whitepapers
  - IBM und MS:  
„Using WSDL in a UDDI Registry 1.05“  
UDDI Working Draft  
Best Practice Document (June 25, 2001)  
*<http://www.uddi.org/pubs/wsdl/bestpractice-V1.05-Open-20010625.pdf>*
- Spezifikationen
  - UDDI Spezifikationen:  
*<http://www.uddi.org/specification.html>*
    - UDDI Version 2.0 Programmer's API Specification
    - UDDI Version 2.0 Data Structure Specification
    - UDDI Version 2.0 XML Schema
    - UDDI Version 2.0 Replication Specification
    - UDDI Version 2.0 XML Replication Schema
    - UDDI Version 2.0 XML Custody Schema
    - UDDI Version 2.0 Operator's Specification