

# Entwicklung eines Webservice-Verzeichnisses für das UIS Baden-Württemberg

Workshop am FZI Karlsruhe  
27. Juli 2006



# Agenda

- 09:00 – 09:30 Ausgangslage, Grundlagen, Projektidee – Ziele des Workshops
- 09:30 – 10:45 Anwendungsfälle (Fixierung)
- 10:45 – 11:00 Pause
- 11:00 – 12:15 Organisation, Funktionalität und Schnittstellen (Diskussion: Anwendersicht)
- 12:15 – 13:00 Pause
- 13:00 – 14:15 Systemanforderungen (Diskussion: Technische Sicht)
- 14:30 – 14:45 Pause
- 14:45 – 15:30 Abschlussdiskussion
- 15:30 – 16:00 Zusammenfassung und weiteres Vorgehen



# Ausgangslage

- UIS Baden-Württemberg: Zunehmend standardisierte Webservices aus Gründen der Wiederverwendbarkeit eingesetzt
  - Anbieter: UIS-Berichtssystem, Fachdienste Xfaweb
  - Nutzer: Themenpark Umwelt, PaUla und Umweltportal des UINBW
  - Weitere Webservice-Schnittstellen im Rahmen von UIS-Fachdokumentenmanagement angedacht, Ziel: Integration aller Fachdokumentbestände in einen zentralen Fachdokumentenbrowser
- Planung: Ansatz flächendeckend und konsequent weiterverfolgen und eine Webservice-Infrastruktur dafür aufzubauen – drei Ebenen:
  - Wichtigsten Funktionen existierender UIS-Komponenten auch als Webservices anbieten
  - Formale Beschreibung der Dienste über WSDL vornehmen
  - **UDDI-basierter Verzeichnisdienst denkbar, aber...**

**„Wichtiger im Rahmen des UIS-Baden-Württemberg wäre die Tatsache, dass es ein zentrales und vollständiges Verzeichnis verfügbarer Services gibt“.**



# Wünsche – Ziele – Anforderungen

- Mit einer Webservice-Infrastruktur sind die Entwickler einer neuen UIS-Komponente in der Lage, über den Verzeichnisdienst festzustellen, ob ...
  - Benötigte Daten oder Leistungen über einen Webservice beschafft werden können und
  - Den Service dann ggf. nutzen.
- Für spezielle Zwecke wie
  - Die Ausgabe von Karten und
  - Die Weitergabe von Nachrichten

sollten die dafür vorgesehenen Standards (WMS, RSS) verwendet werden. Hilfreich wäre es, wenn diese Dienste in einem **erweiterten** Verzeichnisdienst aufgenommen werden könnten, um einen zentralen Einstiegspunkt für Entwickler zu erhalten.



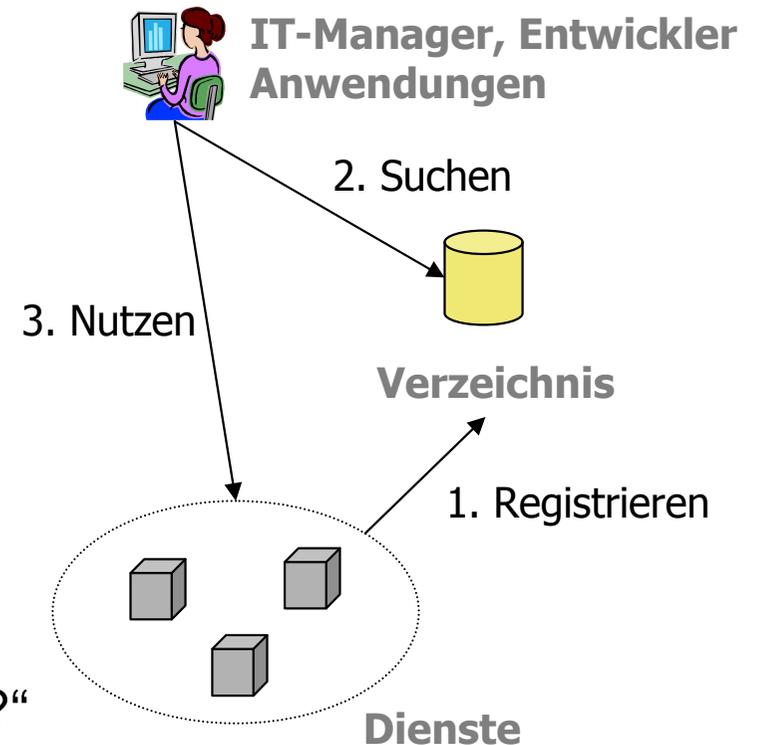
# Verzeichnisse in Service-orientierten Architekturen

## Vorzüge eines Dienstverzeichnisses:

- Integrierte Verwaltung der zur Nutzung eines Dienstes notwendigen technischen Angaben (technische Service-Metadaten)

## Probleme:

- Dienstbeschreibung und -suche anhand rein technischer Aspekte
- Nur schwache Anhaltspunkte für die fachliche Bedeutung eines Dienstes
- Beispiel: „Haben wir einen Dienst, der den Wasserpegel auf dem Neckar liefert?“
- Suchstrategien: Durchsuchen von Kommentaren in WSDL-Dokumenten? Namen von Operationen? Namen von XML-Elementen? ...



# Dienstverzeichnis UDDI

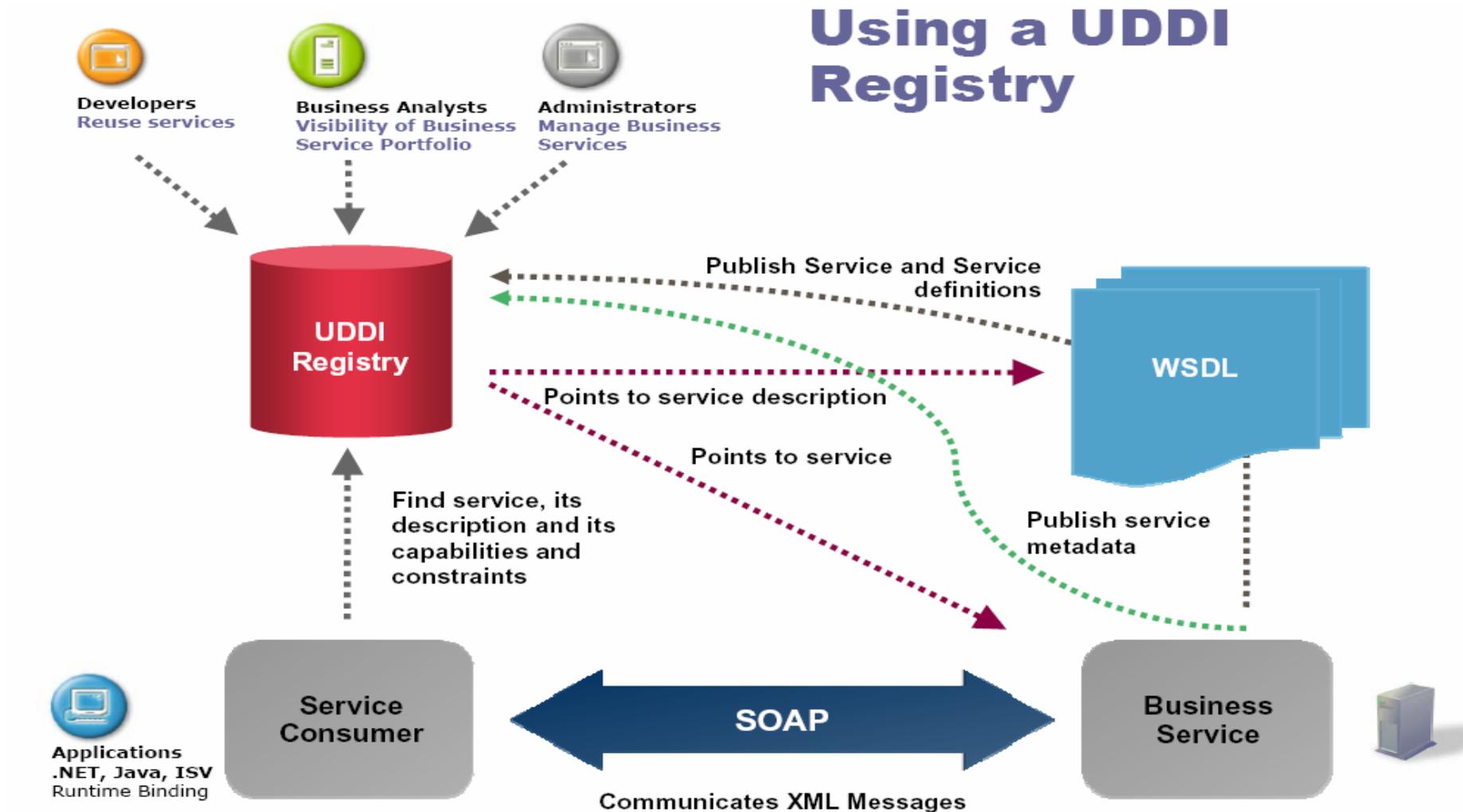


# UDDI: Überblick

- **U**niverselle, zentrale "E-Business"-Suchmaschine mit verteilter Konzeption (UDDI Business Registry)
  - Informationen über Unternehmen abrufen (**D**escription)
  - Geschäftspartner auffinden (**D**iscovery)
  - Geschäftsprozesse über Unternehmensgrenzen hinaus beschreiben und nutzbar machen (**I**ntegration)
  
- Entwicklung begann im Frühjahr 2000
  - Ariba, IBM, Microsoft
  - Erste Registry im Mai 2001



# UDDI: Nutzung



# UDDI: Überblick (2)

- Universeller Verzeichnisdienst
  - *White pages*: Informationen über Unternehmen (Namensregister, Details, Kontaktinformationen)
  - *Yellow pages*: »Branchenbuch« - Ordnung der Unternehmen nach Technologiecodes, Branchentaxonomien oder geographischem Ort
  - *Green pages*: Geschäftsmodell, technische Informationen über Nachrichtenformate, Protokolle (=> WSDL), Informationen über Geschäftsprozesse
  - *Service type registrations*: Referenzen auf Beschreibungen des jeweiligen Typs
- Primärer Zweck ist nicht unbedingt das *automatische* Auffinden



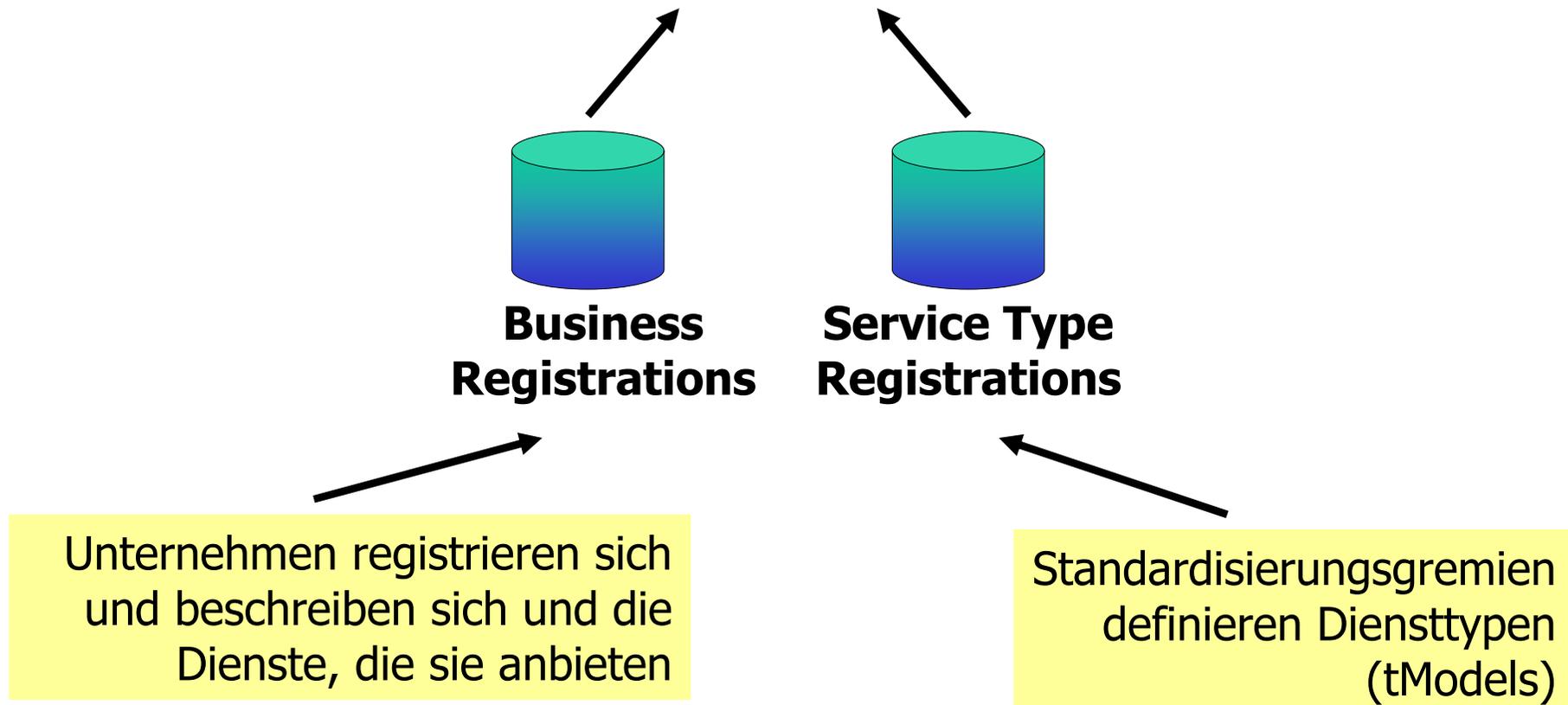
# UDDI: Überblick (3)

- Informationen über das anbietende Unternehmen
  - Auch Taxonomie nach Branche o.ä. möglich
- Beschreibung des angebotenen Dienstes
  - Nicht nur für Webservices
  - Kategorisierung des Dienstes (Thesauri) nach
    - Branche
    - Art der Dienstleistung
    - Geographischer Ort
  - Technische Schnittstelle (z.B. WSDL, aber auch natürlich sprachliche Beschreibungen)
- Protokoll zum Veröffentlichen und Nachschlagen von Dienstinformationen (Basis: SOAP)
  - Verteilte Verzeichnisse

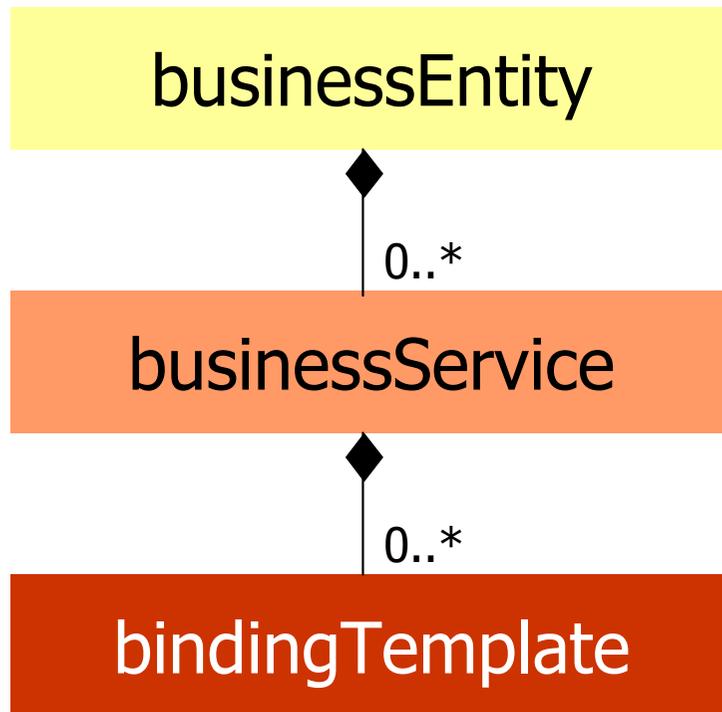


# UDDI: Registrieren von Diensten

Nutzung der Daten über Suchmaschinen oder aus Anwendungen direkt heraus, um Dienste anderer Unternehmen zu finden und in eigene Anwendungen zu integrieren



# UDDI: Datenstrukturen



Name, Beschreibung, Kontakt  
URL-Verweise auf andere businessEntities  
Zuordnung zu Taxonomien

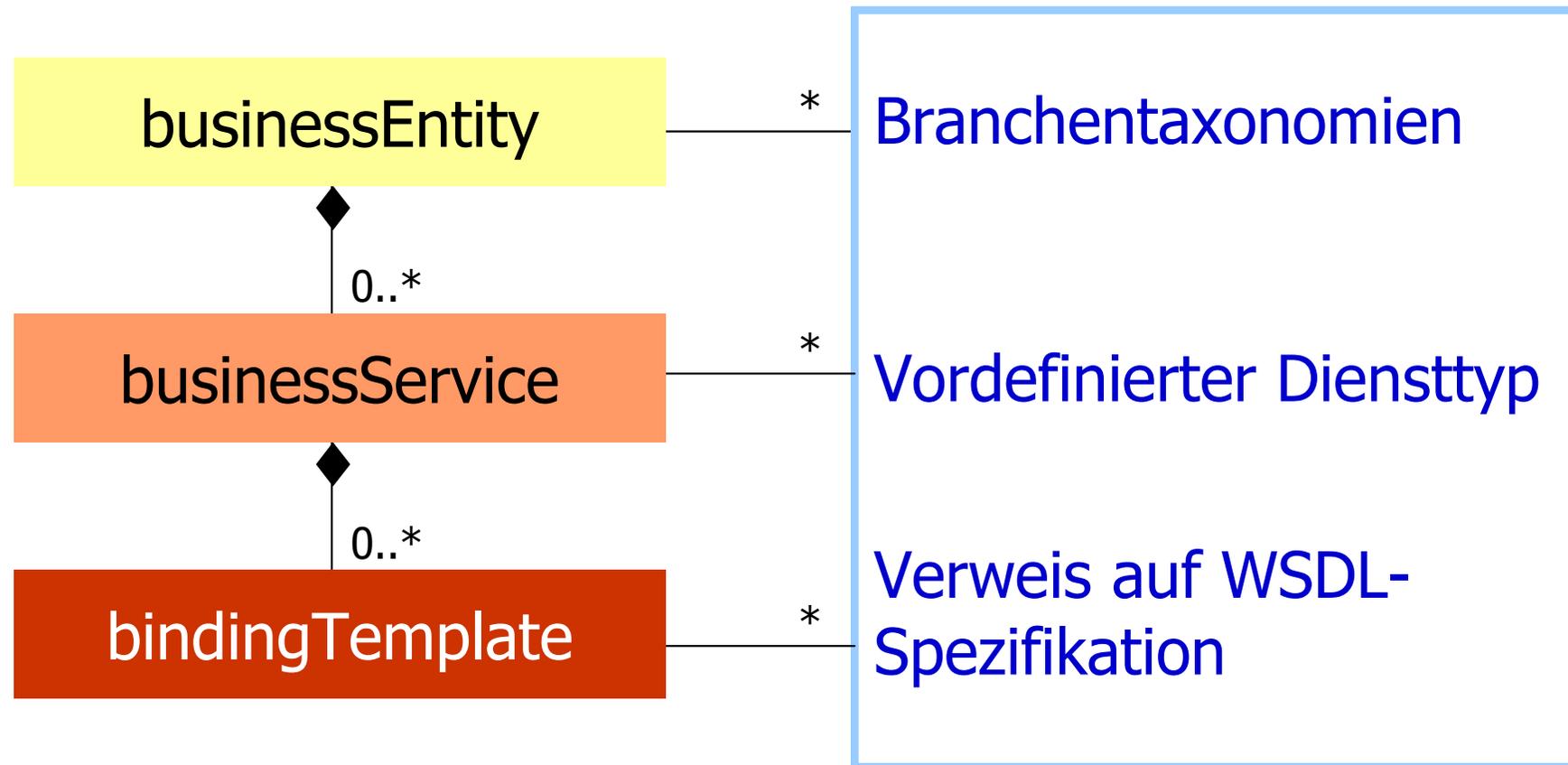
Name, Beschreibung,  
Zuordnung der Dienste zu Kategorien

Name, Beschreibung,  
Kommunikationsmöglichkeiten (http, phone)  
Zugriffspunktdefinition (Adresse)



# UDDI: Technology Models (tModel)

- tModels sind zunächst nur eindeutige Identifikatoren



# UDDI: Musterablauf Protokoll

Browse  
Suchen

find\_business  
find\_service

Drill Down  
Details abrufen

get\_businessDetail  
get\_serviceDetail

Invocation  
Dienst aufrufen

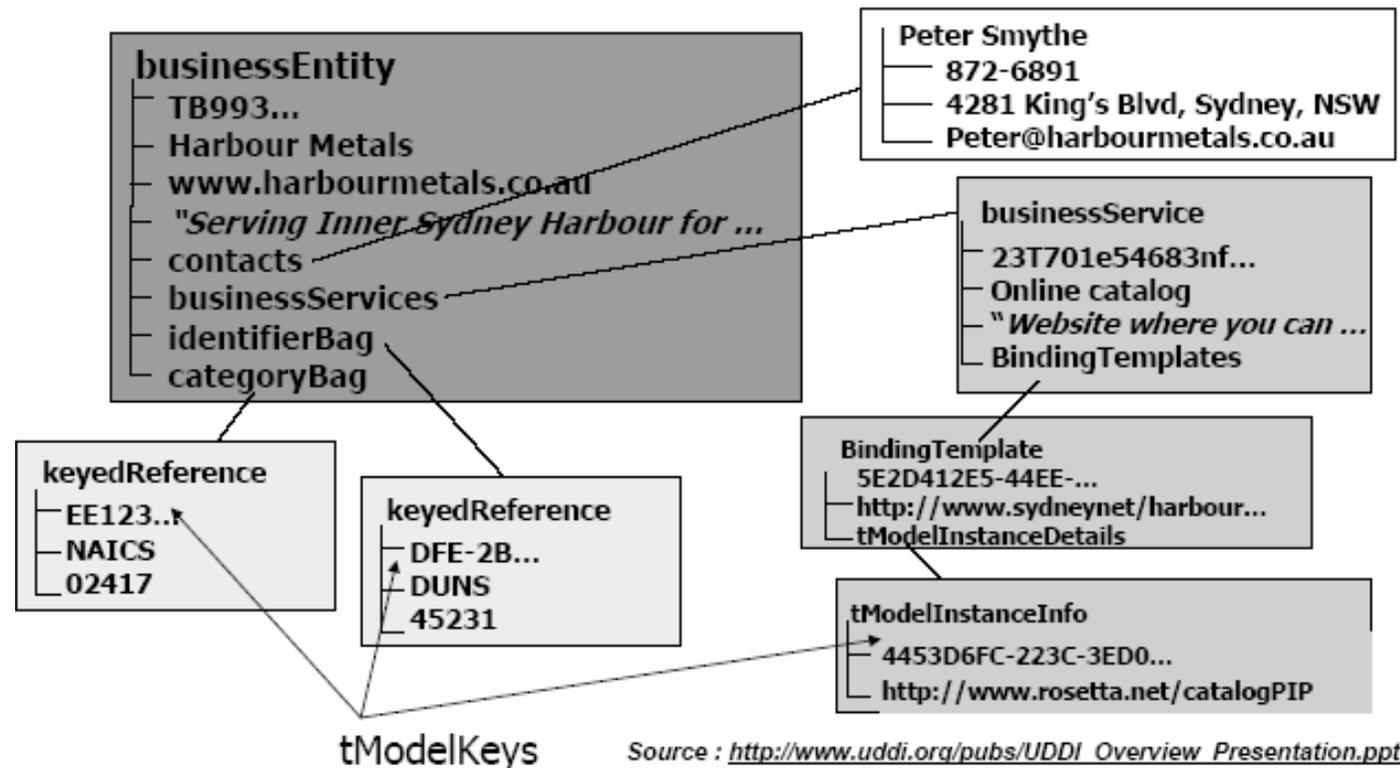
get\_bindingDetail  
liefert WSDL-Information  
anschließend SOAP



# UDDI: Beispiele

- Öffentliche UDDI-Verzeichnisse

- SAP: <http://uddi.sap.com/>
- BindingPoint: <http://www.bindingpoint.com/>
- XMethods: <http://uddi.xmethods.com/>



# UDDI: Probleme

- „UDDI has several different uses, based on the perspective of who is using it. From a business analyst's perspective, UDDI is similar to an Internet search engine for business processes. [...] **However, business users probably won't browse a UDDI registry directly, since the information stored within it is not necessarily reader friendly.** A series of marketplaces and business search portals could crop up to provide business analysts with a **more user-oriented approach** to browsing the services and businesses hosted in a UDDI registry.“

(Aus: **Java Web Services**, Tyler Jewell, David Chappell, March 2002, O'REILLY)





# Projektidee

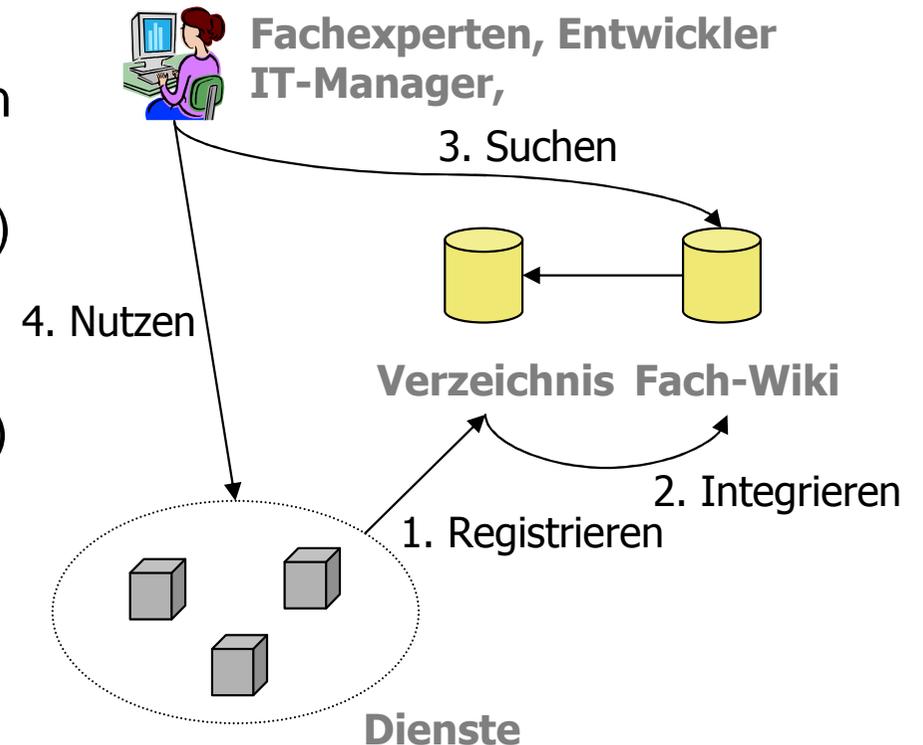


# Idee: Fachlich orientierte Dienstsuche

## ■ Ansatz: Einbettung der Service-Metadaten in eine Fachwissensstruktur

- Fachwissensstruktur in Form eines Hypertextlexikon (Fach-Wiki: „Wasser“, „Wasserpegel“, „Neckar“)
- Anreicherung der Dienstdokumentation durch Verweise in das Fachwissen („Pegeldienst“ liefert → „Wasserpegel“)
- Automatische Integration der Dokumentation in das Fachlexikon („Wasserpegel“ siehe → „Pegeldienst“)

## ■ Suchstrategie: Dienstsuche durch Browsen/Suchen in der Fachwissensstruktur



# Idee: Fachlich orientierte Dienstsuche (2)

## Vorteile des Ansatzes

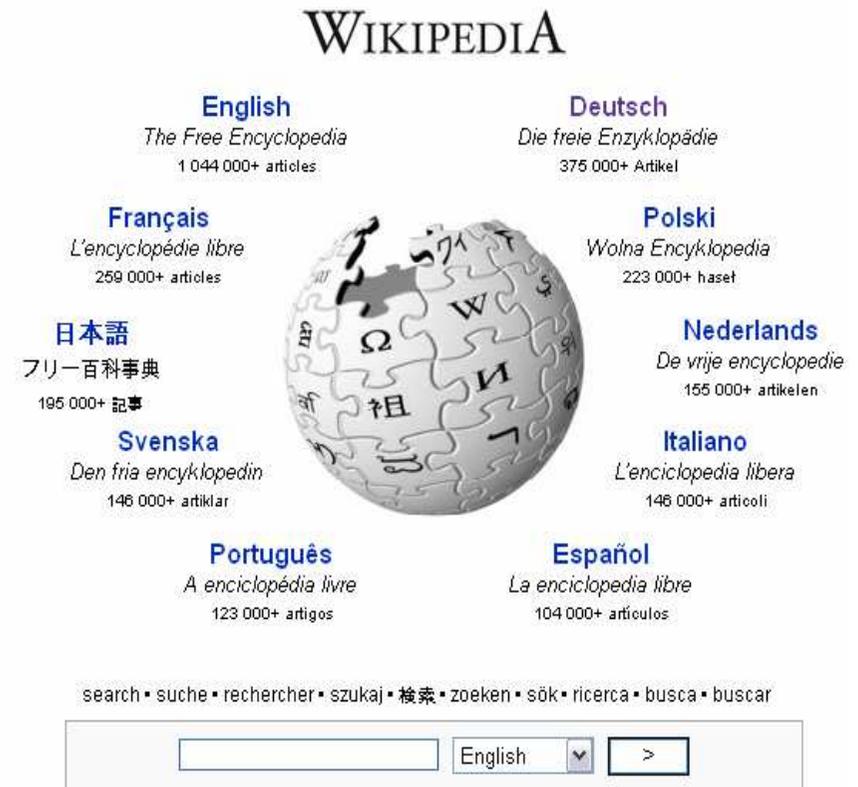
- Intuitive Nutzung
  - Natürlichsprachliche, menschenverständliche Darstellungsform der Service-Metadaten
  - Fachliche Anfragen werden möglich:
    - Haben wir bereits Dienste im Bereich „Wasser“?
    - Welche Dienste liefern „Wasserpegel“?
  - Zugang auch ohne technologischen Kenntnisse oder Ziele
- Wiederverwendbarkeit
  - Überschaubarer Einrichtungsaufwand durch den Einsatz vorhandener Komponenten und Standards (Wiki, WSDL, UDDI)
  - Rein technisches Dienstverzeichnis inbegriffen
  - Keine Mehrfacherfassung von Daten



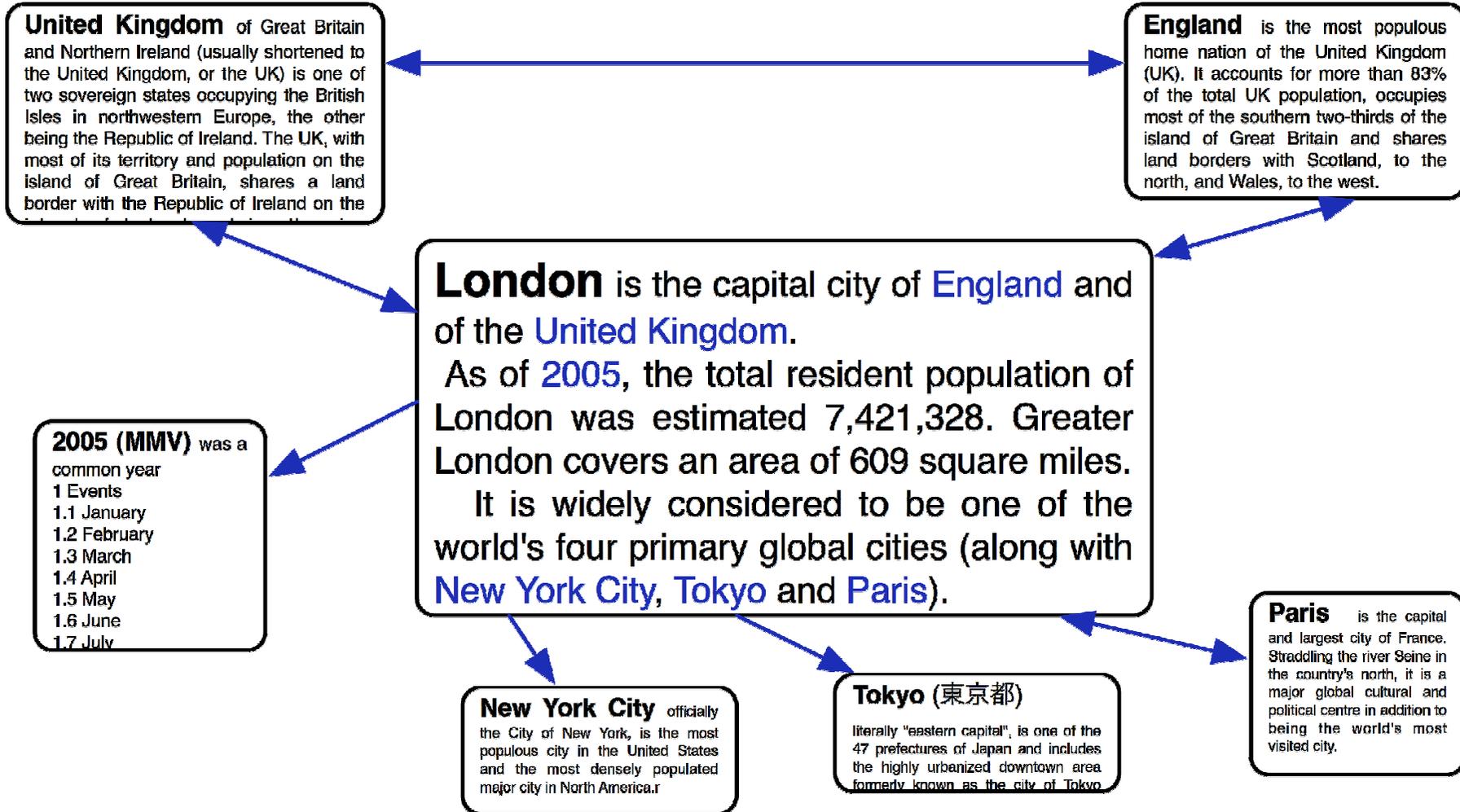
# Technologie: Wiki

## ■ Wiki

- Ein Hypertext-basiertes Lexikon von Begriffen
- Eine Seite pro Begriff
- Seitentext enthält Verweise auf andere relevante Begriffe
- Kooperative Fortentwicklung des Lexikon
- Ergebnis: Eine schnell wachsende, hochintegrierte Wissensstruktur
- Nachteil: Wissen ist nicht maschinell verarbeitbar
- [www.wikipedia.org](http://www.wikipedia.org)



# Standard WIKI



# Technologie: Semantic Wiki

## ■ Wiki

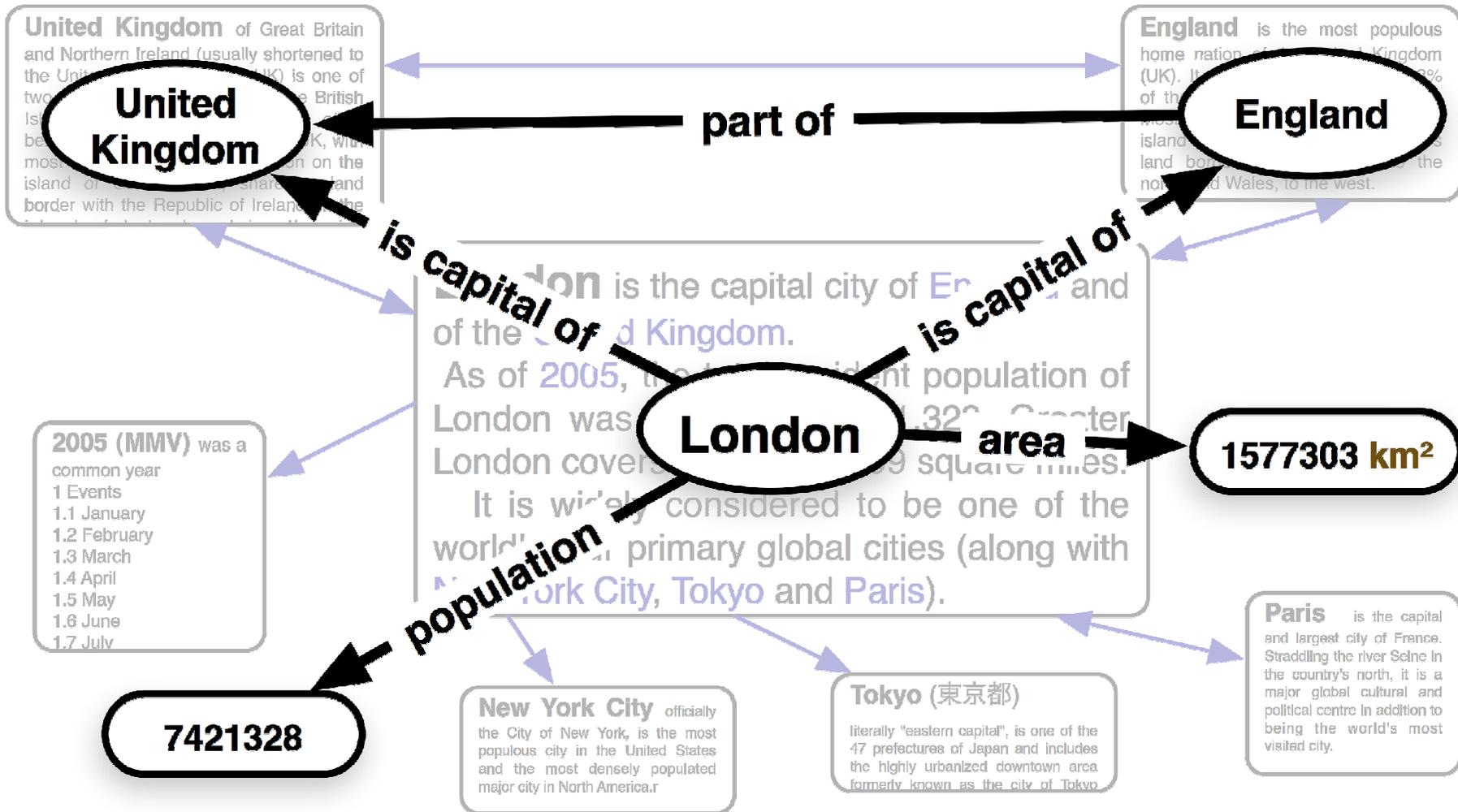
- Ein Hypertext-basiertes Lexikon von Begriffen
- Eine Seite pro Begriff
- Seitentext enthält Verweise auf andere relevante Begriffe
- Kooperative Fortentwicklung des Lexikon
- Ergebnis: Eine schnell wachsende, hoch-integrierte Wissensstruktur
- Nachteil: Wissen ist nicht maschinell verarbeitbar
- [www.wikipedia.org](http://www.wikipedia.org)

## ■ Semantic Wiki

- Maschinelle Verarb. möglich!
- Idee: Begriffe und Verweise werden typisiert
- Vorteile:
  - Strukturierte Anfragen sind möglich
  - Implizite Fakten können automatisch hergeleitet und in die Seiten automatisch integriert werden
  - Hintergrundwissen in Form von Ontologien kann hinzugefügt werden
  - Der natürlichsprachliche Zugang bleibt erhalten
- [wiki.ontoworld.org/wiki/Main\\_Page](http://wiki.ontoworld.org/wiki/Main_Page)



# Semantic Wiki



# Systemarchitektur

- Domänenorientiertes Browsen
- Volltextsuche
- Strukturierte Anfragen (per Ontologie-Server)



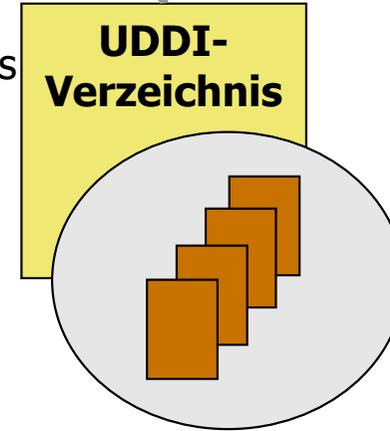
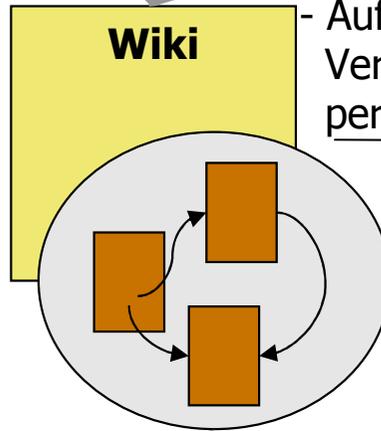
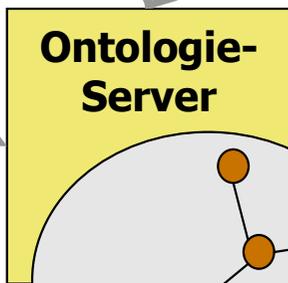
1. Generieren von annotierten Wiki-Seiten aus annotiertem WSDL

- Anlegen annotierter WSDL-Beschreibungen
- Pflege der Service-Metadaten
- Suche anhand rein technischer Kriterien

3. Schließen auf implizite Fakten

2. Gewinnung von Fakten aus annotierten Wiki-Seiten

- Aufrufen des Verzeichnisses per Link



4. Anreicherung der Wiki-Seiten mit impliziten Fakten



# Beispiel: „Pegelauskunft“ (Schritt 1)

Wiki

UDDI-Verzeichnis

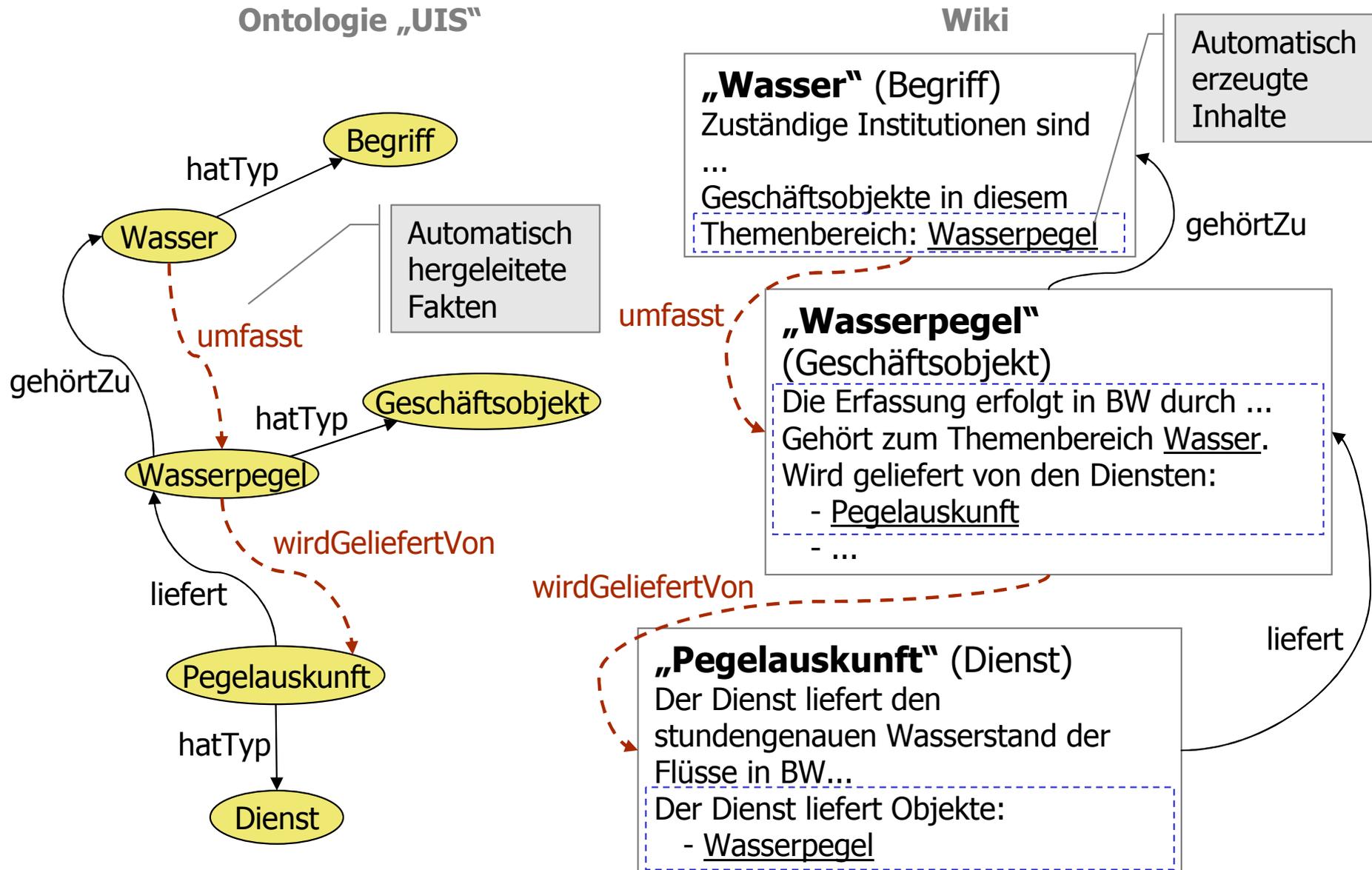
Generieren von  
annotierten Wiki-  
Seiten aus  
annotiertem WSDL

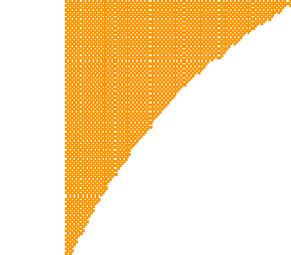
**„Pegelauskunft“ (Dienst)**  
Der Dienst liefert den stundengenauen  
Wasserstand der Flüsse in BW...  
Der Dienst liefert Objekte:  
- Wasserpegel

```
<description xmlns:wSDL=„...“>
  <interface name=„Pegelauskunft“>
    <operation name=„GibPegel“>
      ...
      <output
        element=„uis:PegelListe“>
        <documentation>
          Der Dienst liefert den
          stundengenauen Wasserstand
          der Flüsse in BW ...
          <o:ontoRef
            property=„uis:liefert“
            value=„uis:Wasserpegel“/>
        </documentation>
      </output>
    </operation>
  </interface>
</description>
```



# Beispiel: „Pegelauskunft“ (Schritte 2-4)





# Projektmethode und Ziele des Workshops



# Projektmethode und -volumen

- Methode



- Volumen

- Gesamt: 65 Personentage
- Max. 40 Personentage für Implementierung

- Priorisierung der Anforderungen notwendig!



# Ziele Workshop

- Workshop dient als Grundlage der Feinspezifikation
- Notwendig
  - Klärung, was genau unter „Dienst“ zu verstehen ist
  - Festlegen der unterstützten Anwendungsfälle
  - Festlegen der Nutzergruppen des Verzeichnisses
- Hilfreich
  1. Wie soll sich das Verzeichnis aus Sicht der verschiedenen Anwender verhalten/präsentieren?
  2. Welchen technischen Anforderungen, Schnittstellen und Randbedingungen muss das Verzeichnis genügen?

