

---

# Daten verknüpfen mit RDF



**Fraunhofer** Institut  
Intelligente Analyse- und  
Informationssysteme

---

Lars Bröcker, Fraunhofer IAIS

# Agenda

---

- Einführung
- Grundlagen von RDF
- RDF Schema
- RDF und Unternehmen
- Werkzeuge
- Kurzübersicht OWL

## Einführung – Was ist RDF?

---

“The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is particularly intended for representing metadata about Web **resources** [..]. However, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be *identified* on the Web, even when they cannot be directly *retrieved* on the Web.”

(RDF Primer)

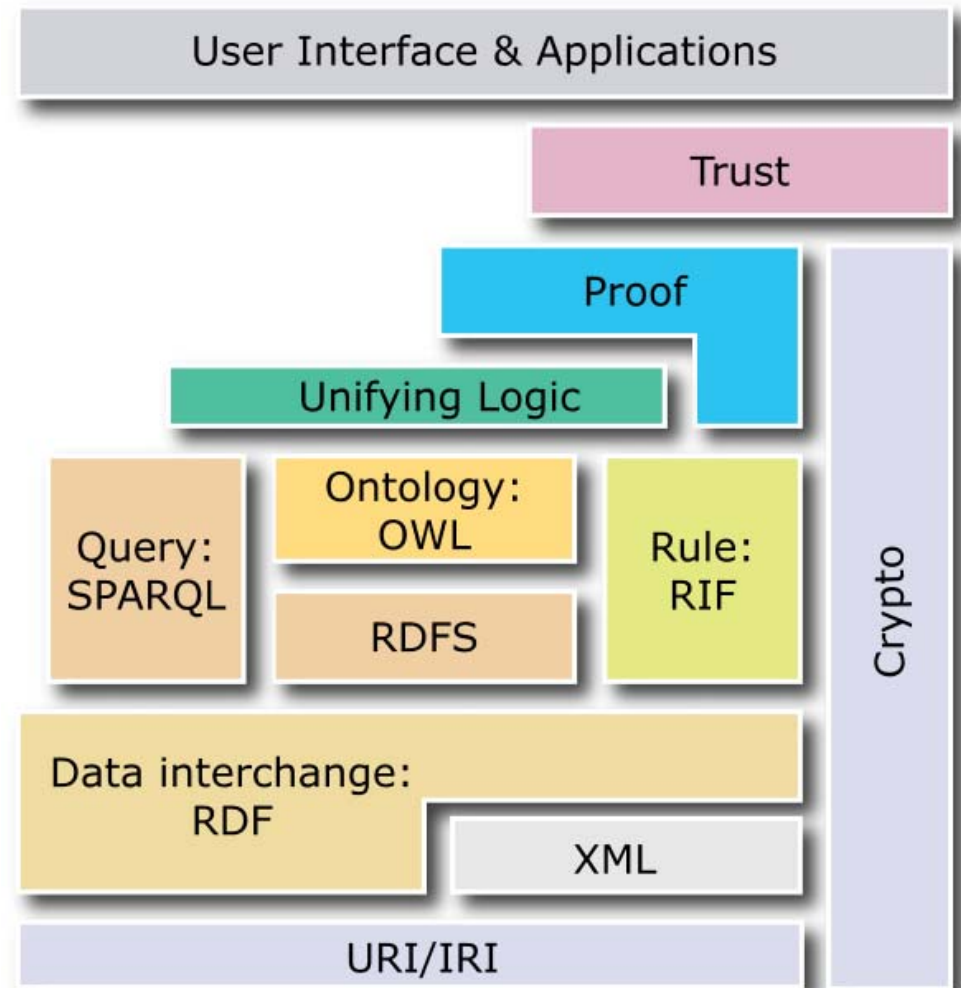
## Resource Description Framework

- ist eine Sprache, also maschinell auswertbar
- dient zum Beschreiben von Ressourcen
  - im Web
  - die im Web referenziert werden können

# Einführung – Was hat RDF mit dem Semantic Web zu tun?

## Semantic Web Stack

- RDF ist grundlegendes Austauschformat im Semantic Web
- Baut auf URI/IRI und XML auf
- SPARQL zum Abfragen von RDF-Daten
- RDFS und OWL zur Erweiterung der Modellierungsmöglichkeiten

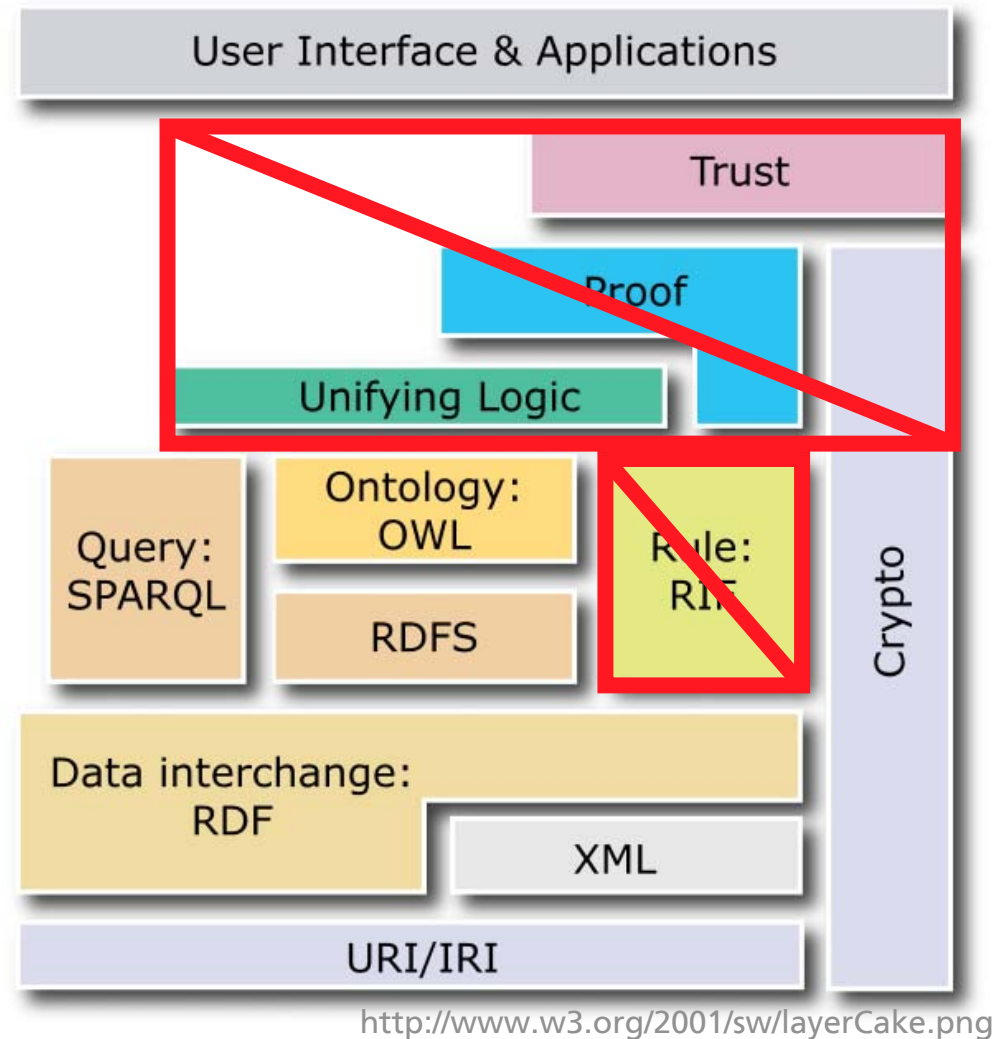


<http://www.w3.org/2001/sw/layerCake.png>

# Einführung – Was hat RDF mit dem Semantic Web zu tun?

## Semantic Web Stack

- RDF ist grundlegendes Austauschformat im Semantic Web
- Baut auf URI/IRI und XML auf
- SPARQL zum Abfragen von RDF-Daten
- RDFS und OWL zur Erweiterung der Modellierungsmöglichkeiten



# Einsatzgebiete für RDF

---

- Erfassen/Verteilen von Metadaten zu
  - Webinhalten → RSS 1.0
  - Bilddaten → Adobe XMP
- Datenhaltung, Data Warehousing, Business Intelligence
  - HP, IBM, ORACLE, SAP
- Datenbereitstellung
  - Texterschließung → OpenCalais von Reuters
  - DBPedia und Linking Open Data

---

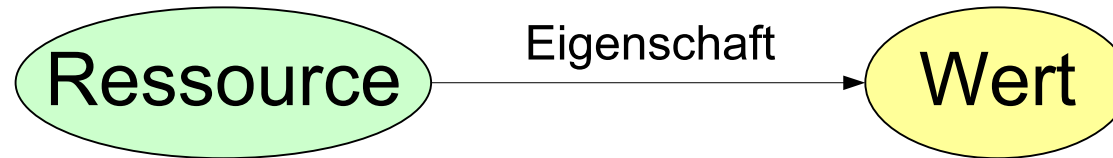
# RDF – Grundlagen



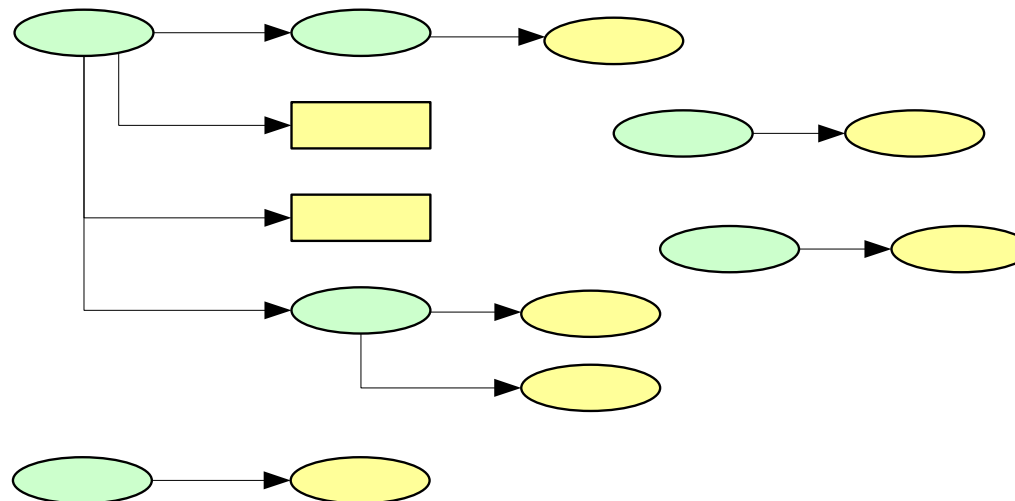
# Grundlagen von RDF

---

RDF besteht aus Aussagen der Form



RDF organisiert Aussagen als Graph



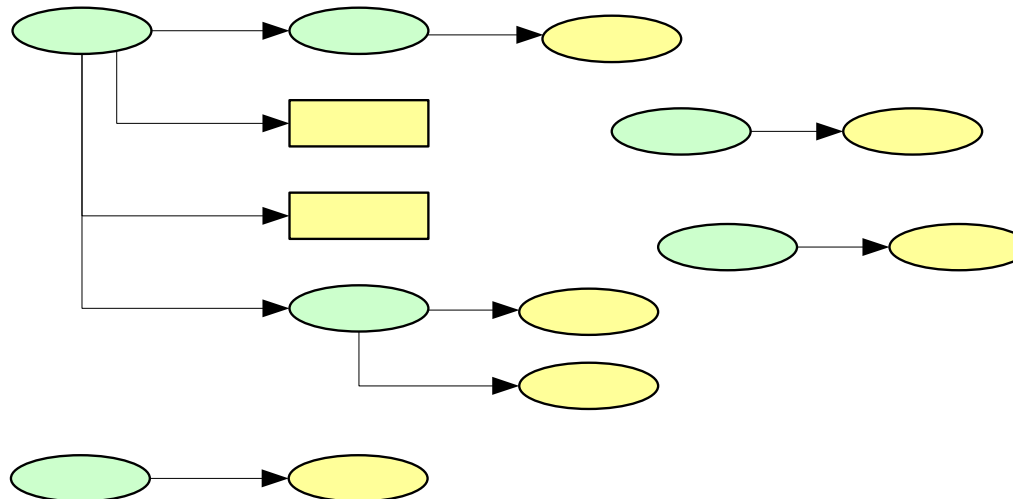
# Grundlagen von RDF

---

RDF besteht aus Aussagen der Form



RDF organisiert Aussagen als Graph



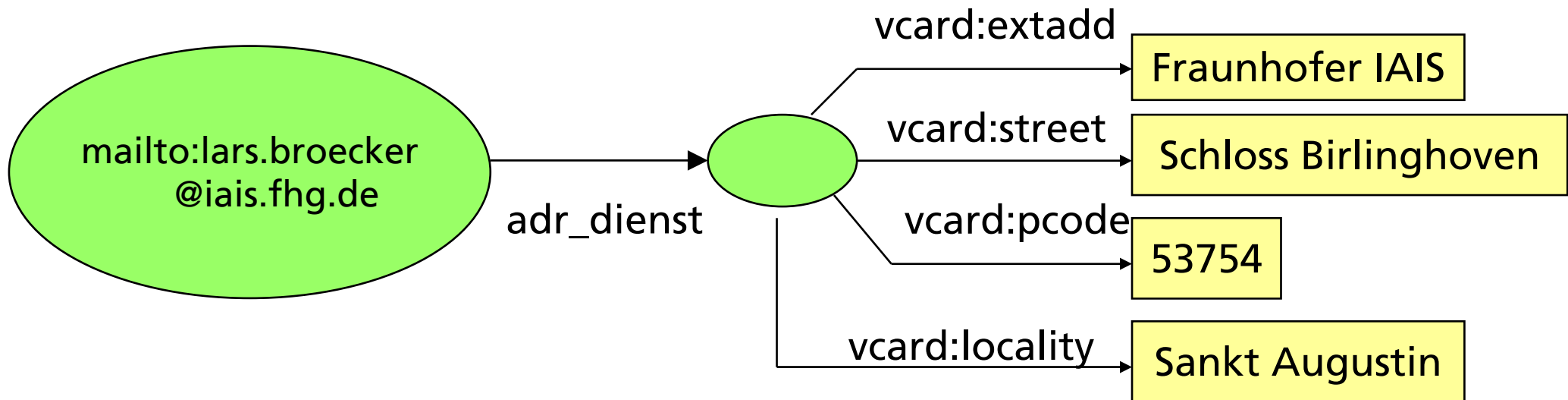
# Eigenschaften der Knoten und Kanten

---

- Eigenschaften sind immer URI
- Objekte sind entweder URI oder Literale
  - Literale typisierbar über XSD Datentypen
  - optionale Sprachangabe über ISO 3166 („de“, „en“)
- Ressourcen sind entweder URI oder Blank Nodes

# Die Welt besteht nicht nur aus Tripeln

- Adressen zum Beispiel
  - Bestandteile sollen zusammengehalten werden
  - Mehrere Adressen zu einer Person sind möglich (z.B. dienstlich/privat)
- 2 Möglichkeiten: Ressourcen oder Blank Nodes einfügen



# Funktionalitäten in RDF

---

- Namespaces
  - kürzen IRI ab
  - vereinfachen Zuordnung der IRI zu Datenquellen
- rdf:type
  - klassifiziert Ressourcen.
  - Bedeutung: X ist vom Typ Y
- Spezielle Container-Typen
  - Seq → geordnete Liste, mit Duplikaten
  - Bag → ungeordnet, mit Duplikaten
  - Alt → Auswahlliste
- Collection
  - abgeschlossene Liste

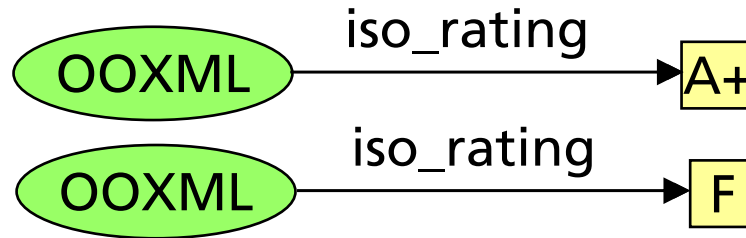
# Reifikation – Aussagen über Aussagen

---

Problem: Wie markiert man die Herkunft einer Aussage?

Microsoft: „OOXML ist toll!“

IBM: „OOXML ist nicht toll!“



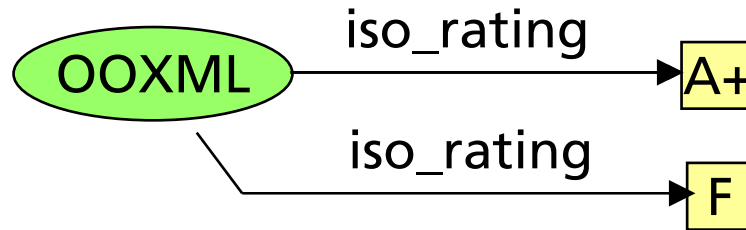
# Reifikation – Aussagen über Aussagen

---

Problem: Wie markiert man die Herkunft einer Aussage?

Microsoft: „OOXML ist toll!“

IBM: „OOXML ist nicht toll!“



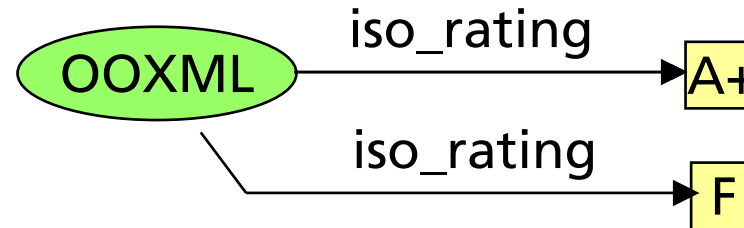
Vereinigung in einem Modell  
führt zu widersprüchlicher  
Aussage

# Reifikation – Aussagen über Aussagen

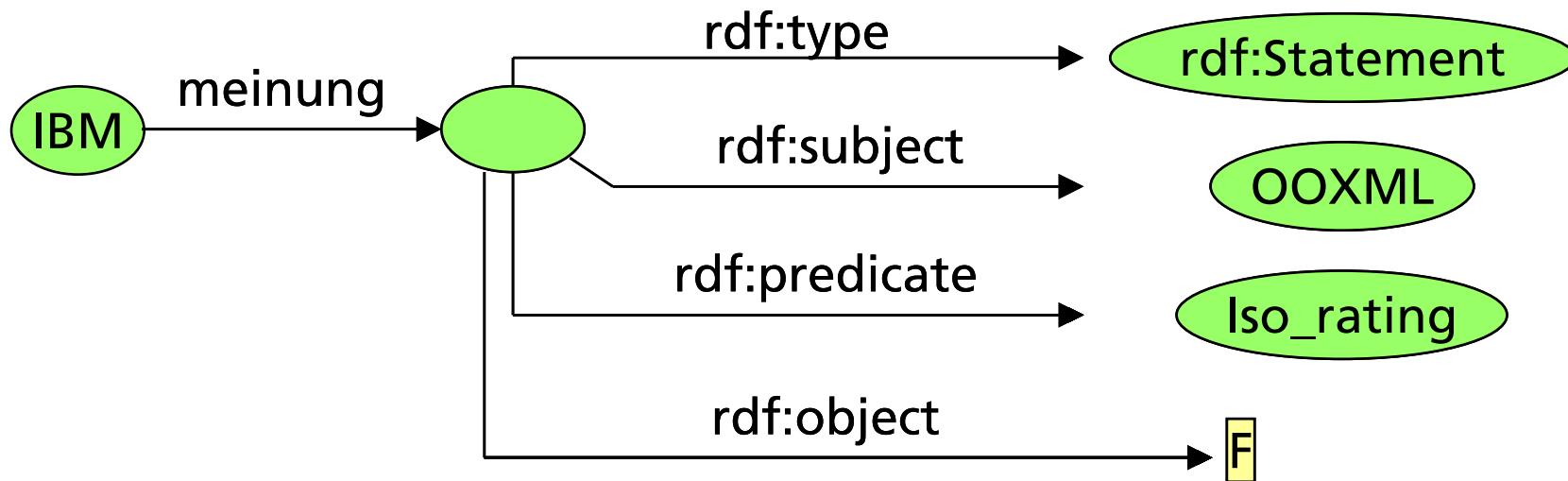
Problem: Wie markiert man die Herkunft einer Aussage?

Microsoft: „OOXML ist toll!“

IBM: „OOXML ist nicht toll!“



Zur Lösung werden die Statements geteilt





# Serialisierungen

---

- RDF/XML – XML wie man es kennt. Spitze Klammern satt
- N3 – einfaches Textformat
  - Ein Statement pro Zeile
  - Abkürzungen `rdf:type` → `a`
  - Eigenes Regelsystem, damit eigentlich mächtiger als RDF
- TURTLE
  - Subset von N3, ohne Regelsystem.
  - Eingabeformat für SPARQL

`ex:resource ex:property ex:value[^^xsd:integer] .`

# Zusammenfassung

---

- RDF ist ein Format zur Erfassung von Metadaten
- Beschrieben werden kann alles, wozu eine IRI vergeben werden kann
- Deskriptoren sind frei wählbar
- **Open World Assumption**

---

# RDF – Schema

# RDF Schema

---

RDF Schema erweitert RDF um Möglichkeiten der Definition applikations-spezifischer Vokabulare

- Definition von Typklassen
- Definition von Klassen- und Eigenschaftshierarchien
- Angabe von Definitions- und Wertemengen für Eigenschaften

# RDF Schema – Klassen und Klassenhierarchien

---

- Klassendefinition
- $X \text{ rdf:type rdfs:Class .}$
  
- Subklassenbeziehungen
- $Y \text{ rdfs:subClassOf } X .$
  
- $\text{rdfs:subClassOf}$  ist transitiv (erfordert RDFS-Inferenz)  
 $Z \text{ rdfs:subClassOf } Y. \rightarrow Z \text{ ist Subklasse von } X$
  
- Mehrere Subklassenbeziehungen möglich

# RDF Schema – Eigenschaften und Eigenschaftshierarchien

---

## Eigenschaftsklassendefinition

general\_case rdf:type RDFS:Property .

## Eigenschaftshierarchie

special\_case rdfs:subPropertyOf general\_case .

- rdfs:subPropertyOf ist transitiv (erfordert RDFS-Inferenz)
- Mehrere Beziehungen möglich

# RDF Schema – Definitions- und Wertebereiche von Properties

- `rdfs:domain` gibt den Definitionsbereich an
- `rdfs:range` gibt den Wertebereich an
- Beide können mehrfach vergeben werden
- Als Wertebereich kann auch ein typisiertes Literal angegeben werden

`has_mother rdf:type rdfs:Property .`

`has_mother rdfs:domain Person .`

`has_mother rdfs:range Person .`

`has_mother rdfs:range Female .`

Nur Personen haben Mütter,  
Mütter sind Personen und  
Frauen

# Zusammenfassung

---

- RDFS ist eine Sprache zur Definition einfacher Vokabulare
- Erlaubt Klassen- und Eigenschaftshierarchien
- Einschränkungen der erlaubten Definitions- und Wertebereiche
- Nutzung der erweiterten Sprachfunktionalitäten nur mit RDFS-API
- Wenn RDFS nicht reicht, dann wird OWL nötig



---

# RDF und Unternehmen

# Warum diese neuen Technologien?

---

- maschinell auswertbar
- offene Spezifikationen
- für die Architektur des Web gemacht
  - Service-Orientierung der Spezifikationen
  - externe Datenquellen im Web nutzen
  - Integration unterschiedlicher Vokabulare/Quellen möglich
- Metadaten sauber von Artefakten getrennt
- inkrementell implementierbar
  - bereits heute Erfahrungen sammeln
  - weitere Entwicklungen des Semantic Web Stack nutzen (Inferenz, Reasoning, Web of Trust)

# Welche Anwendungsgebiete im Enterprise-Sektor?

---

- Semantische Integration verschiedener Datenquellen
  - Wrapper für Legacy-Systeme erzeugen RDF-Views der Datenquellen
  
- Business Intelligence
  - Aggregation der Daten verschiedener Systeme
  - Regelmechanismen auf RDFS/OWL für die Auswertung
  
- Semantische Suche auf Unternehmensdaten
  - Anfragen auf alle Systeme des Unternehmens, über Web
  - Integration von Knowledge Discovery erlaubt Integration unstrukturierter Daten

---

# Werkzeuge

# Werkzeuge (Selektion)

---

## API Frameworks

- Redland (C, C#, Java, Perl, PHP, Python, Ruby)
- Jena/Joseki (Java)
- Sesame (Java)
- RAP (PHP)

## Objekt-RDF Mapper

- RDFReactor (Java)
- ActiveRDF (Ruby)

## Triple Stores

- Boca (IBM) / Open Anzo (OSS)
- Virtuoso OSS/kommerziell
- Mulgara
- Oracle 11g

---

# Kurzübersicht OWL

OWL ist eine Sprache zur Definition von Ontologien

- Constraints auf Eigenschaften
  - Werte einschränken
  - Anzahl der Vorkommen begrenzen
  - optionale/zwingend erforderliche Eigenschaften
  - neue Typen: transitiv, symmetrisch, invers, funktional, invers funktional
- Constraints auf Klassen
  - Intersection, union, complement
  - Disjunktheit
- OWL wird in RDF notiert
- OWL Lite, OWL DL, OWL Full

---

# Soweit Teil 1

## Bis hierhin Fragen?

Lars.Broecker@iais.fraunhofer.de