



Semantic Web Grundlagen und RDF

Marko Harasic

Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme

harasic@inf.fu-berlin.de

Warum reicht XML generell nicht aus?

- `<Buch>Dieses Buch</Buch> hat den Titel
<Titel>Semantic Web Grundlagen</Titel>`

- `<foo>Dieses Buch</foo> hat den Titel
<bar>Semantic Web Grundlagen</bar>`

- natürliche Sprache
- Mehrdeutigkeit

<Apple>

<Pear>



Apple



- Syntax – die Art und Weise, wie Worte in einem Satz zusammengesetzt wurden.
- Semantik – Informationen, die in diesem Sinne kodiert wurden.
- Pragmatik – Implikationen aus den Informationen in einem Kontext.

- unstrukturierter Text
- keine Unterstützung für Maschinenverarbeitung
- Data mining benötigt NLP
- keine/kaum Metadatenstandards

- Keyword-basiert
- hoher recall, geringe precision

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

<http://upload.wikimedia.org/math/9/b/5/9b5a523e5d9c366caf75ed1ea1767b1c.png>

<http://upload.wikimedia.org/math/b/4/3/b43cb2dbb709c4932e8dd8b9b0c491fe.png>

Bildersuche: „Apache“



- Maschinen fehlt dieser Kontext aus Begriffen und Zusammenhängen
- Kontext muss Maschinen zusätzlich bereitgestellt werden

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

Berners-Lee, Hendler, and Lassila, 2001.



Foto: W3C



Foto: Homepage

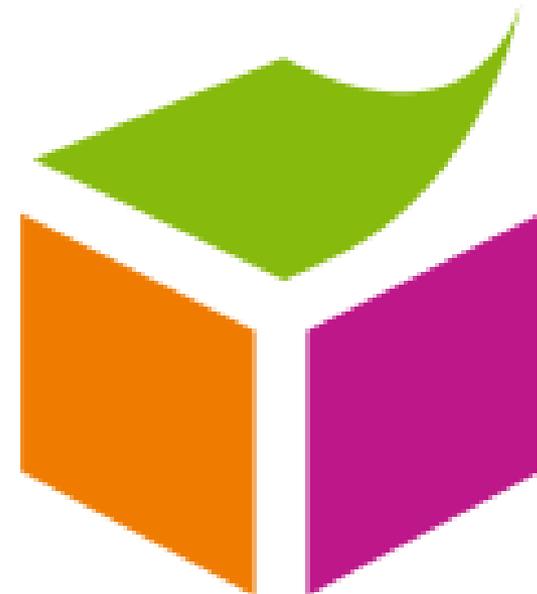


Foto: Homepage

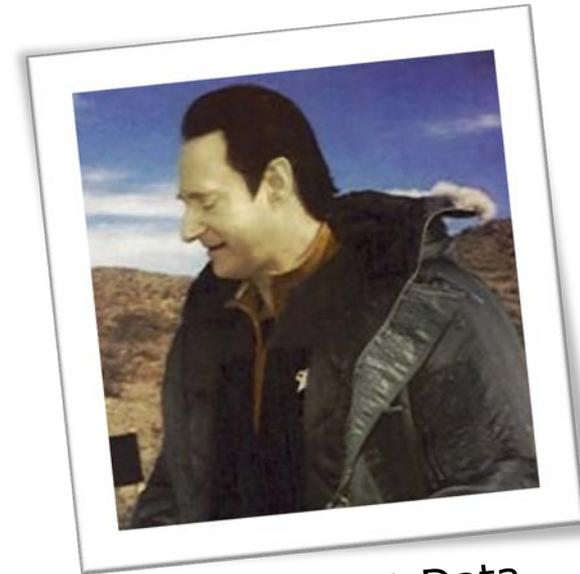
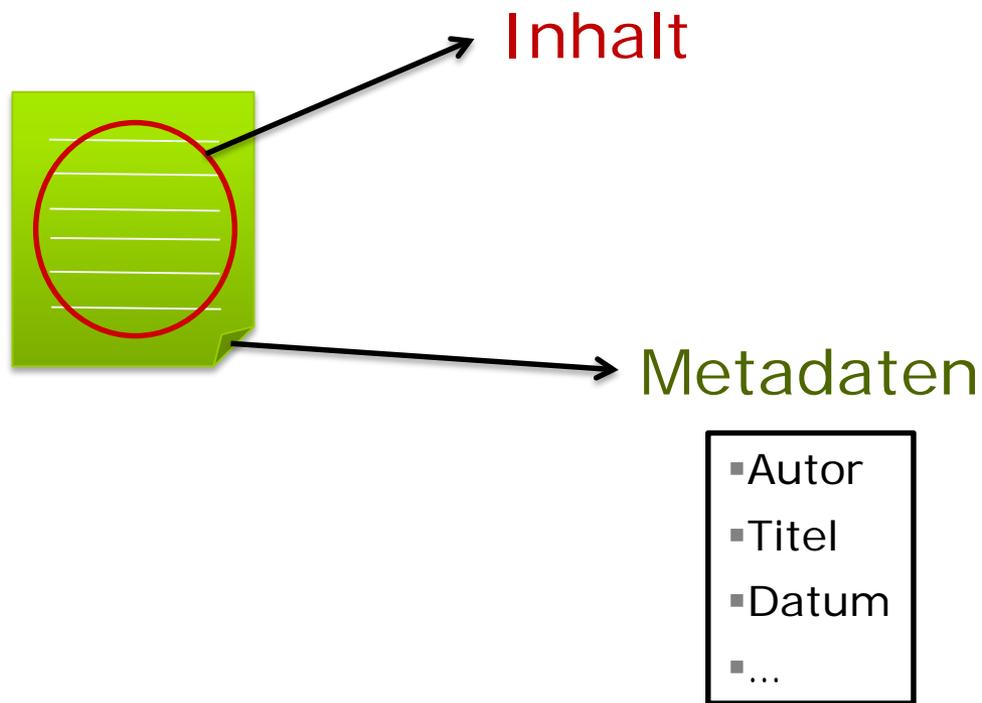
- Webinhalte und ihre Vernetzung werden für Maschinen verständlich.
- Auch komplexe Anfragen können ans Web gestellt werden.
- Beispiel: Finde alle Fußballspieler, die bei einem Verein spielen, der ein Stadion mit mehr als 40.000 Plätzen hat und die in einem Land mit mehr als 10 Millionen Einwohnern geboren wurden.

- eine Erweiterung des existierenden Web
 - + Metadaten
 - + Ontologien
 - + Reasoning
 - + intelligente Agenten

= Semantic Web



- Daten über Daten
 - beschreiben Inhalt
 - im besten Fall maschinenverarbeitbar



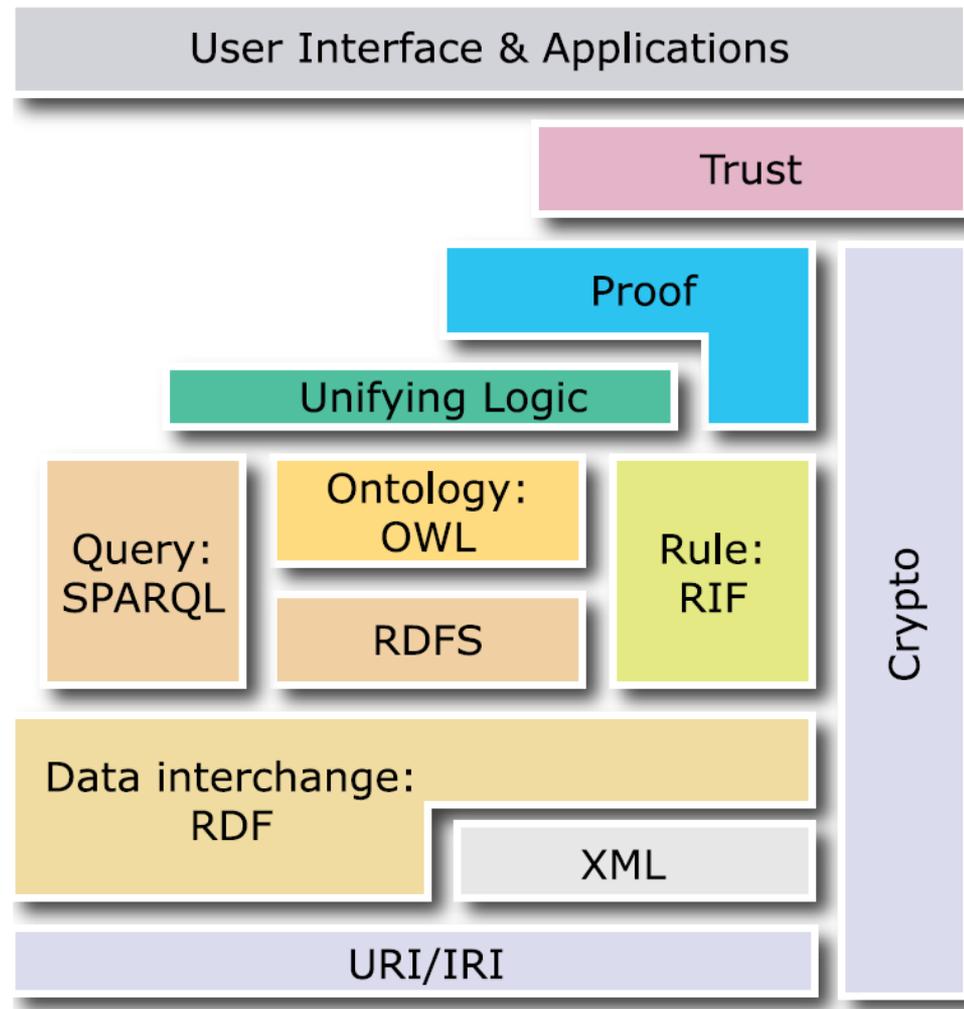
Data about Data

- Spezies: Android
- Größe: ...

http://commons.wikimedia.org/wiki/File:Data_ST.jpg

- **Damit Metadaten nutzbar sind**
 - Informationsanbieter muss sich so ausdrücken, dass Informationsnutzer ihn verstehen
 - Informationsnachfrager muss so fragen, dass er etwas finden kann
- **Gemeinsame Benutzung von Konzepten**
- **Gemeinsame Sprache**
- ***Ontologie* zur Definition einer gemeinsamen Sprache**
 - Es gibt Konzepte, die wir mit „Bank“ und „Sparkasse“ benennen
 - Es gibt ein Konzept, das wir „Geldinstitut“ nennen und das die Konzepte „Bank“ und „Sparkasse“ umfasst

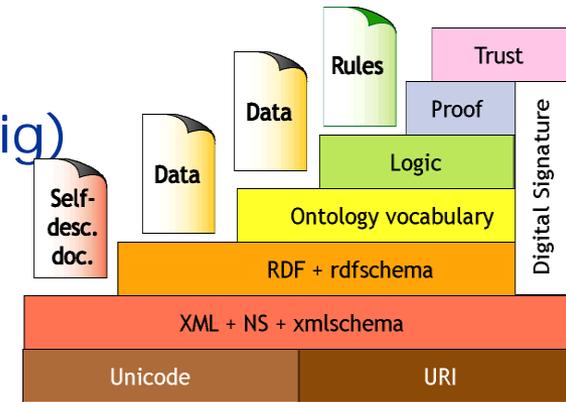
Semantic Web Stack (W3C, 2000)



Quelle: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

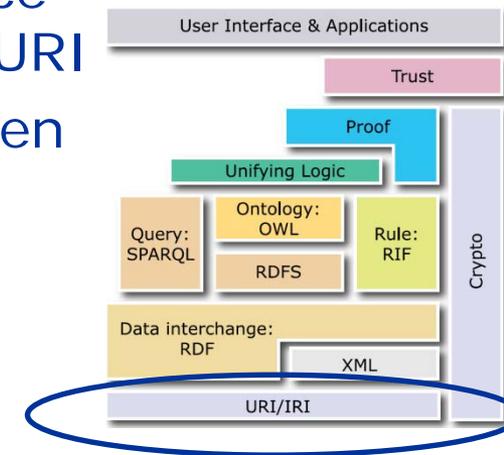
Unicode

- jedes Zeichen eigene Nummer (system-, programm- und sprachunabhängig)
- Unicode-Codierung – Zeichensätze für fast jede natürliche Sprache



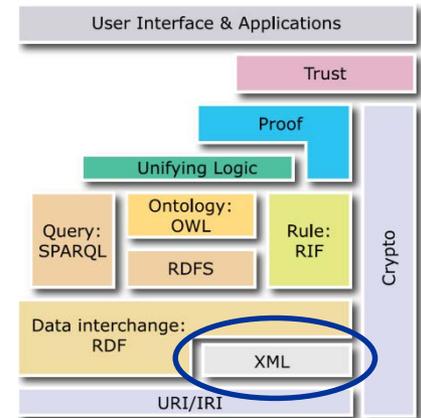
URI – Uniform Resource Identifier

- eindeutige Identifikation einer Quelle/Ressource → jedes beliebige Objekt verfügt über einen URI
- Mechanismus um Daten verteilt zu repräsentieren Untergruppe von URIs
- Syntax vom W3C standardisiert



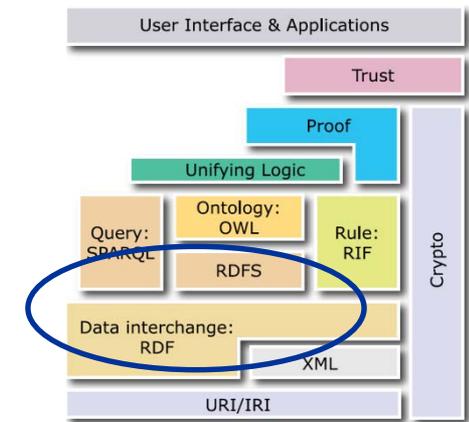
XML + Namensräume + XML-Schema

- hierarchisch strukturierte, medienneutrale Daten
- Vokabular kann mit XML-Schema definiert werden
- Bedeutung des Vokabulars mit Namensräumen festgelegt
- XML-Daten können mit XLink verlinkt werden: Links können Namen, aber keinen Namensraum haben
- ⇒ maschinenverarbeitbare verlinkte Daten, Links jedoch nicht maschinenverarbeitbar



RDF + Namensräume + RDF-Schema

- Web als Menge vernetzter Ressourcen
- Vokabular für Beziehungen kann mit RDF-Schema definiert werden
- Bedeutung des Vokabulars wird mit Namensräumen festgelegt
- RDF Modell bietet eine syntaxunabhängige Darstellung
- ⇒ maschinenverarbeitbares Netzwerk von Beziehungen

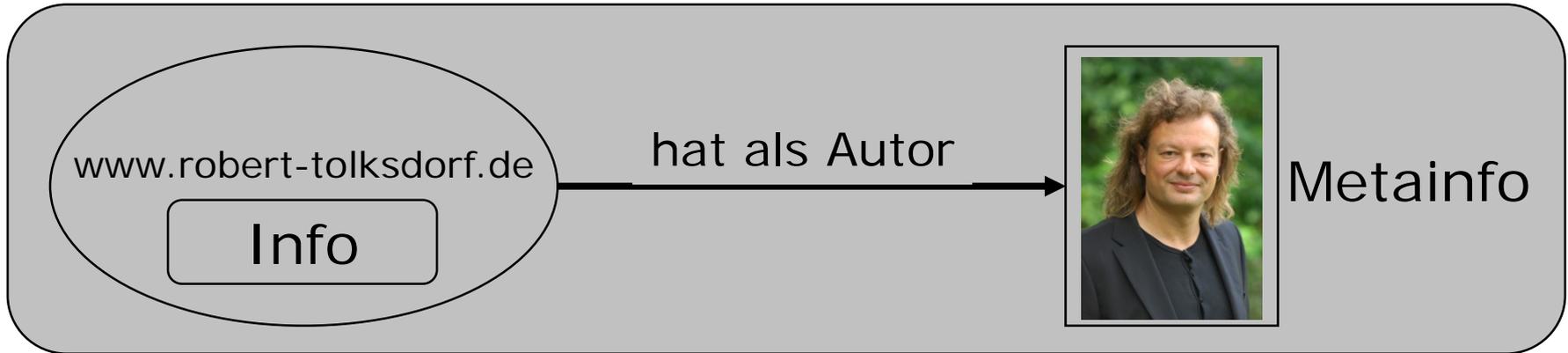


- RDF – W3C Recommendation seit 1999
- verschiedene Versionen:
 - kompakt und lesbar
 - für maschinelle Verarbeitung
- Tripel setzen bel. Web-Ressourcen URI-s und URI-o miteinander in Beziehung:

<URI-s, URI-p, URI-o>

URI-s steht zu URI-o in der Beziehung URI-p

- Informationen und Metainformationen:



- In RDF als Satz ausgedrückt:

"www.robert-tolksdorf.de	Subjekt
hat als Autor	Prädikat
Robert Tolksdorf"	Objekt

- `<?xml version="1.0"?>`

`<RDF xmlns=`

`"http://www.w3.org/1999/02/22-rdf-syntax-ns"`

`xmlns:s="http://description.de/schema/">`

`<Description about=`

`"http://www.robert-tolksdorf.de">`

`<s:Autor>Robert Tolksdorf</s:Autor>`

`</Description>`

`</RDF>`

Subjekt

Prädikat

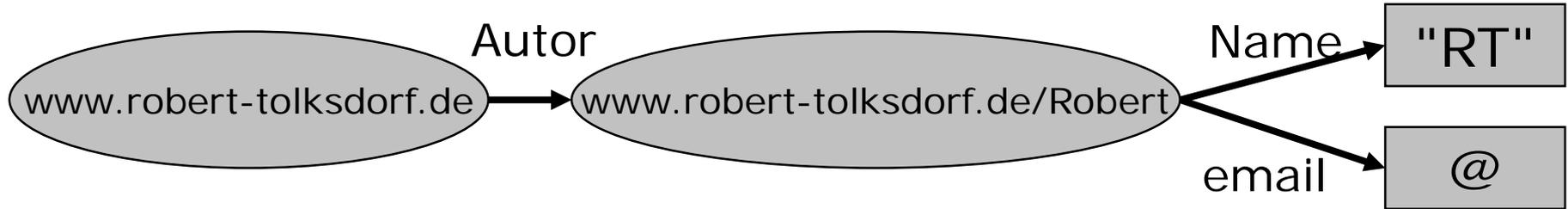
Objekt

- Aus so explizit gemachten und maschinenverständlich repräsentierten Aussagen können Tools und Dienste inhaltliche Schlüsse ziehen

- In `Autor` können keine weiteren Elemente stehen, also auch als XML-Attribut repräsentierbar:

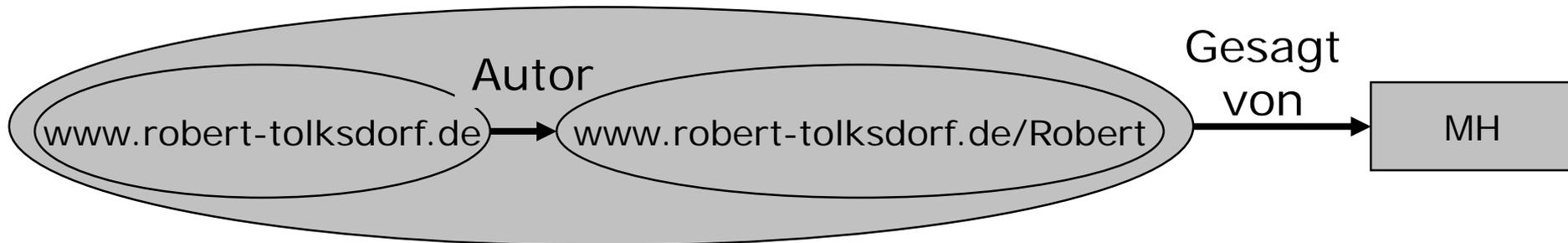
```
<?xml version="1.0"?>
<RDF xmlns=
  "http://www.w3.org/1999/02/22-rdf-syntax-ns"
  xmlns:s="http://description.de/schema/">
  <Description
    about="http://www.robert-tolksdorf.de"
    s:Autor="Robert Tolksdorf"
    s:Erzeugt="10.11.2001"/>
</RDF>
```

- Objekte können selber auch Subjekte sein:



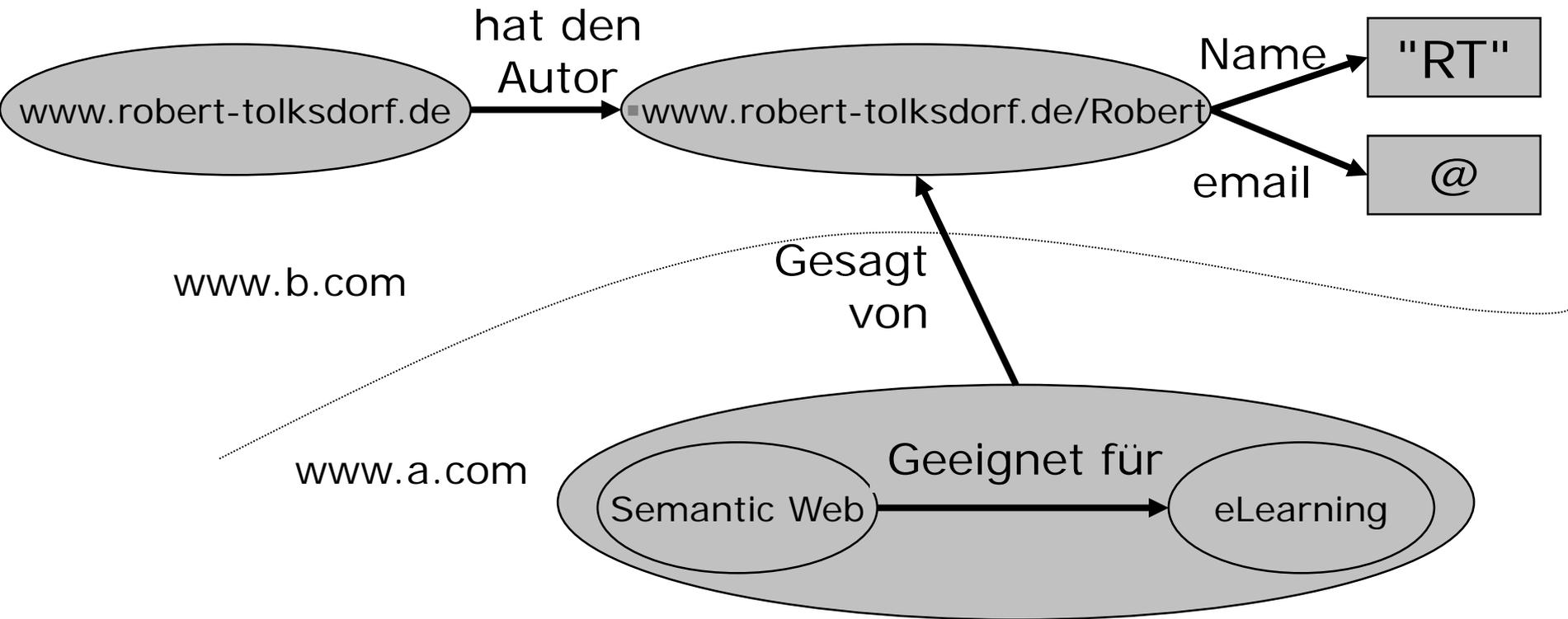
- `<RDF ... >`
 `<Description about=`
 `"http://www.robert-tolksdorf.de/Robert"`
 `s:Name="Robert Tolksdorf"`
 `s:email="mail@robert-tolksdorf.de" />`
 `<Description`
 `about="http://www.robert-tolksdorf.de">`
 `<s:Autor resource=`
 `"http://www.robert-tolksdorf.de/Robert" />`
 `</Description>`
 `</RDF>`

- „Markus Luczak-Rösch sagt ` Robert Tolksdorf ist der Autor seiner Homepage´ “



- ```
<RDF ... >
 <Description>
 <subject resource=
 "http://www.robert-tolksdorf.de" />
 <predicate resource=
 "http://description.de/schema/Autor" />
 <object>Robert Tolksdorf</rdf:object>
 <type resource=
 "http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement" />
 <s:gesagtVon>Marko Harasi c</s:gesagtVon>
 </Description>
</rdf:RDF>
```

- Semantic Web: Verteiltes Geflecht aus getypten Beziehungen zwischen Konzepten



- „RDF-Welt“: Gerichteter Graph
  - Knoten (Ressourcen)
  - Kanten (Properties)
- Ressourcen (RDF Resource)
  - Alles worüber man Aussagen machen kann
  - Identifiziert durch URIs (qualified URIs = URI + fragment identifier)
  - Aussagen sind auch Ressourcen
- Eigenschaften/Beziehungen (RDF Property)
  - Verbinden Ressourcen miteinander oder Ressourcen zu Werten (RDF Literal)
- Aussagen (RDF Statement)
  - (Subjekt, Prädikat, Objekt)
  - “Resource has Property with Value”

- Fragment identifier (eindeutig im Dokument)
- Abkürzung der vollständigen URI einer Ressource
- Vollständiger Name zusammengesetzt aus:
  - Base URI (xml:base = ...)
  - #
  - Wert von rdf:ID
- **Beispiel**
  - `http://www.example.com/products#item123`

- Mengenobjekte (geordnet, ungeordnet, mit Duplikaten, ohne Duplikaten, *offen*)
- Ermöglichen Aussagen über mehrere Ressourcen
- Platzhalter für komplexe Mengenobjekte (vs. Blank Node)

- 3 Typen von Containern

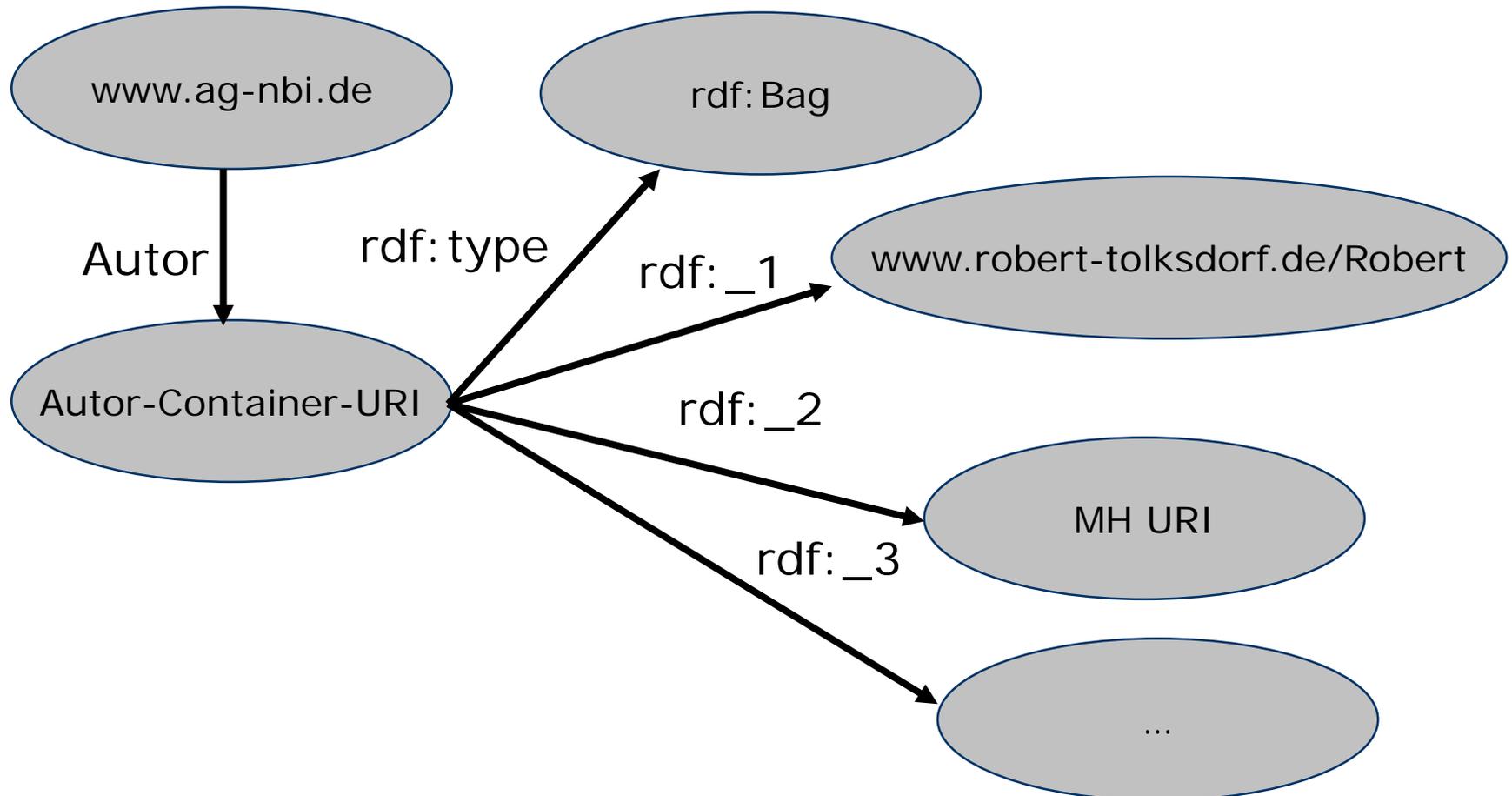
- Bag                      ungeordnete Liste              rdf: Bag
- Sequence              geordnete Liste              rdf: Seq
- Alternative              eindeutiger Wert              rdf: Alt

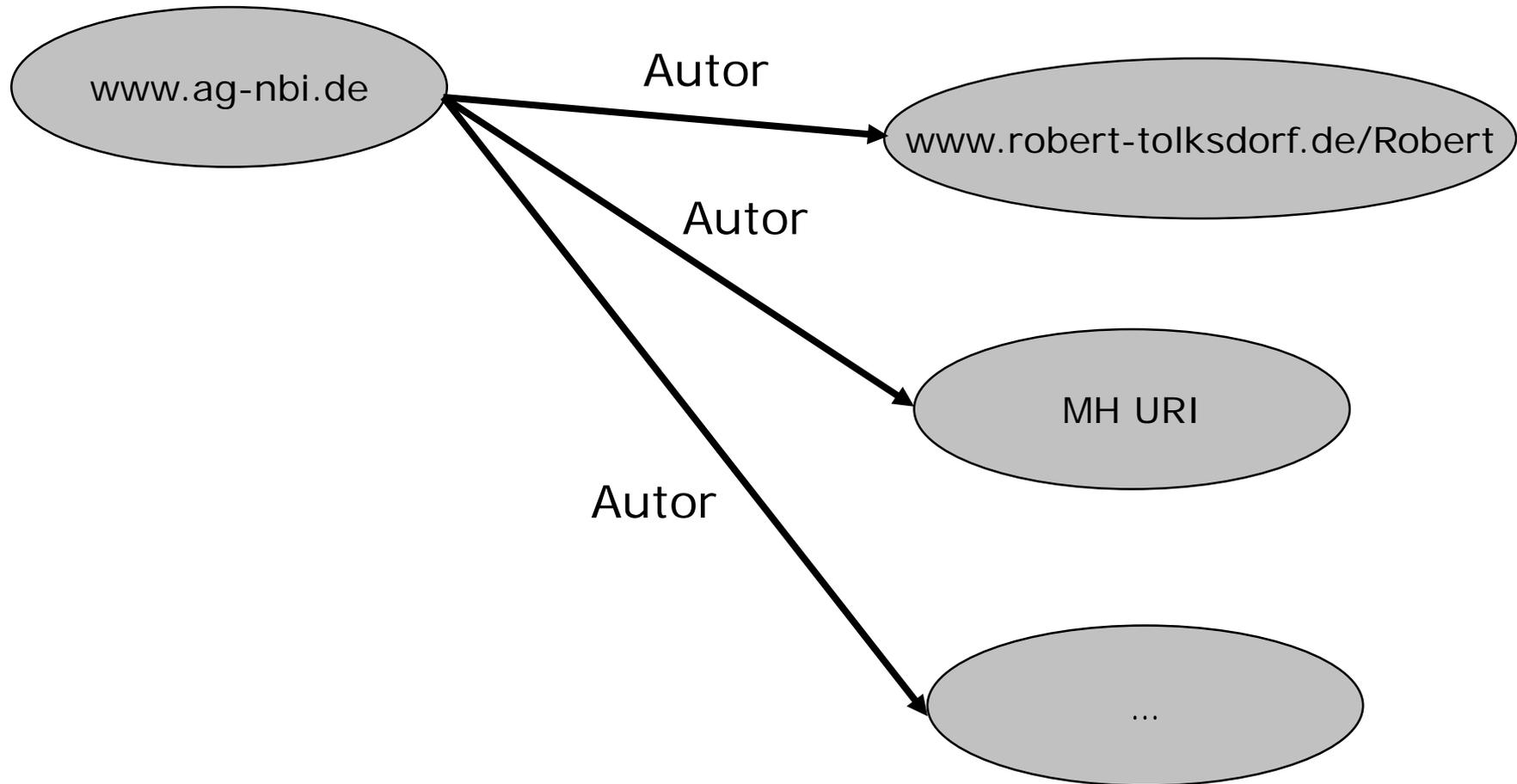
- Containers sind auch RDF Ressourcen

- Semantik: offene Mengen

- unbekannt ob weitere Elemente zu der Menge gehören

- Aussage: die Webseite „www.ag-nbi.de“ wurde erstellt von Robert Tolksdorf und MLR und ...





# Multimengen (Bags)

- Ungeordnete Liste von Werten, Duplikate möglich
- Eigenschaft hat mehrere Werte, die Elemente der Menge
  - z.B. Mitglieder einer Gruppe, Dateien in einem Verzeichnis
- `<RDF ... >`

```

<Description
 about="http://www.ag-nbi.de">
 <s: Autor>
 <Bag>
 <li resource=
 "http://www.robert-tolksdorf.de/Robert" />
 <li resource=
 "http://www.ag-nbi.de/root" />
 </Bag>
 </s: Autor>
</Description>
</RDF>

```

# Liste (Sequence)

- Geordnete Liste von Werten, Duplikate möglich
- Eigenschaft hat mehrere Werte, die Elemente der Menge, deren Reihenfolge wichtig ist
  - z.B. Buch/Artikelautoren, Punkte in einer Tagesordnung
- `<RDF ... >`

```

<Description
 about="http://www.fu-berlin.de">
 <s:Fachbereiche>
 <Seq ID="fachbereiche">
 <li resource=
 "http://www.bio-chem-pha.fu-berlin.de/" />
 <li resource=
 "http://www.ewi-psy.fu-berlin.de/" />
 ...
 </Seq>
 </s:Fachbereiche>
</Description>
</RDF>

```

- Liste von Werten
- Eigenschaft hat einen Wert, der aus der Auswahl stammt
  - z.B. document home and mirrors, mailing-list moderators
- `<RDF ...>`
  - `<Description`
    - `about="http://x.org/packages/X11">`
    - `<s: DistributionSite>`
      - `<Alt>`
        - `<li resource="ftp://ftp.x.org" />`
        - `<li resource="ftp://ftp.cs.purdue.edu" />`
      - `</Alt>`
    - `</s: DistributionSite>`
  - `</Description>`
- `</RDF>`

- **about**

- direkte Angabe des URI

```
<rdf:Description
```

```
 rdf:about="http://www.example.org/index.html">
```

```
 <externs:creation-date>
```

```
 August 16, 1999
```

```
 </externs:creation-date>
```

```
</rdf:Description>
```

- **aboutEach**

- URI eines Containers
- Property auf alle Elemente angewendet

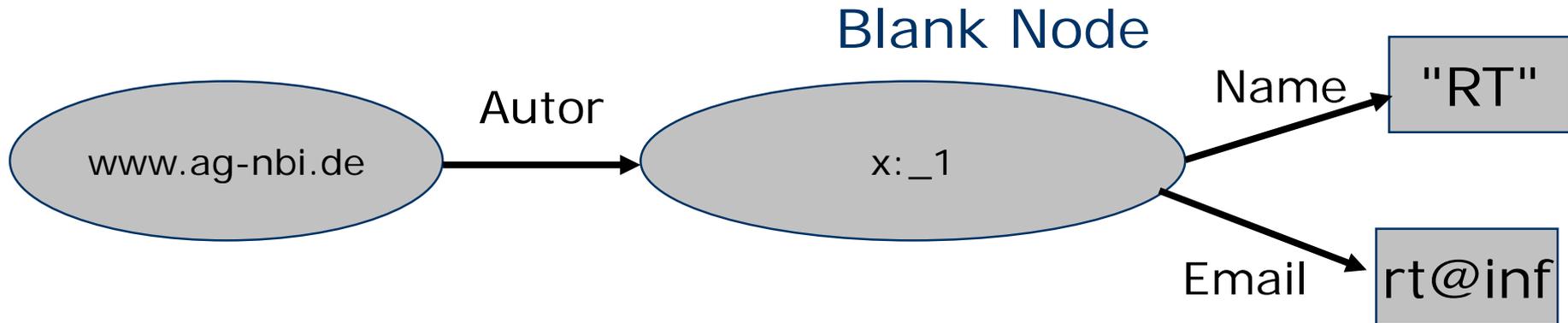
- **aboutEachPrefix**

- URI Präfix
- Eigenschaft auf alle Ressourcen mit dem Präfix angewendet

- Ähnlich zu Containern, aber geschlossen:
  - *Alle* Elemente einer Menge sind spezifiziert
- Zugriff auf einzelne Mengenelemente
  - rekursiv
  - first (erstes Element)
  - rest (restliche Elemente)
  - nil (leere Menge)

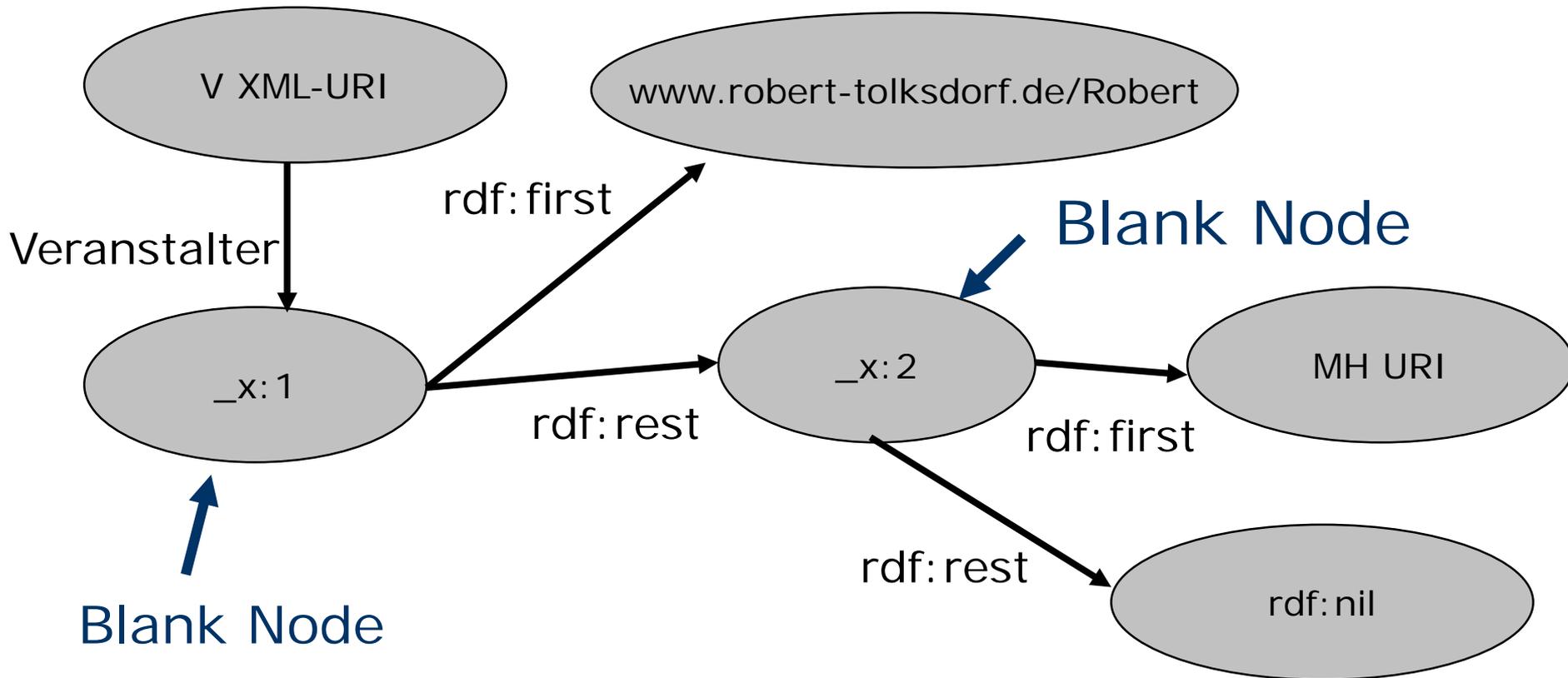
# RDF Blank Nodes

- Anonyme Ressourcen (haben keine URI)
- Platzhalter für komplexe Objekte
- Ressourcen von unbekanntem Typ



- Aussage: die Webseite „www.ag-nbi.de“ wurde erstellt von jemanden/etwas mit dem Namen „RT“ und der Email-Adresse „rt@inf“

- Aussage: Die Vorlesung „XML-Technologien“ wird veranstaltet von Robert Tolksdorf und Markus Luczak-Rösch



- Datenmodell
  - Graphenstruktur:
    - Knoten (Ressourcen, Werte)
    - Kanten (Properties)
- Verschiedene syntaktische Formate
  - RDF/XML Syntax
  - N3
  - ...

```
<rdf:RDF>
```

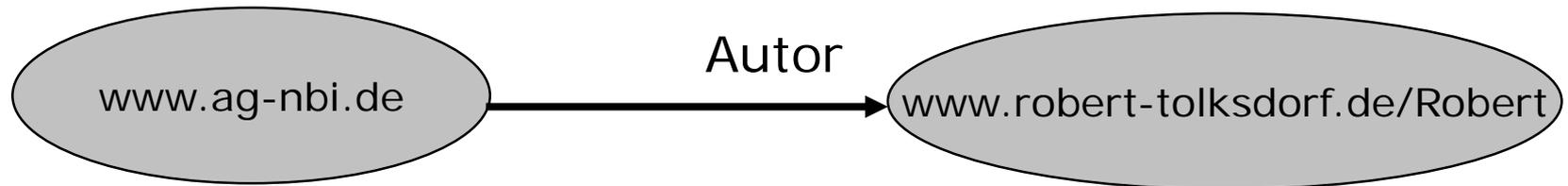
```
<rdf:Description about="http://www.ag-nbi.de">
```

```
<Autor
```

```
 rdf:resource="http://robert.tolksdorf.de/Robert" />
```

```
</rdf:Description>
```

```
</rdf:RDF>
```



```
<rdf:RDF >
```

```
<rdf:Description about="http://www.ag-nbi.de" >
```

```
 <Autor >
```

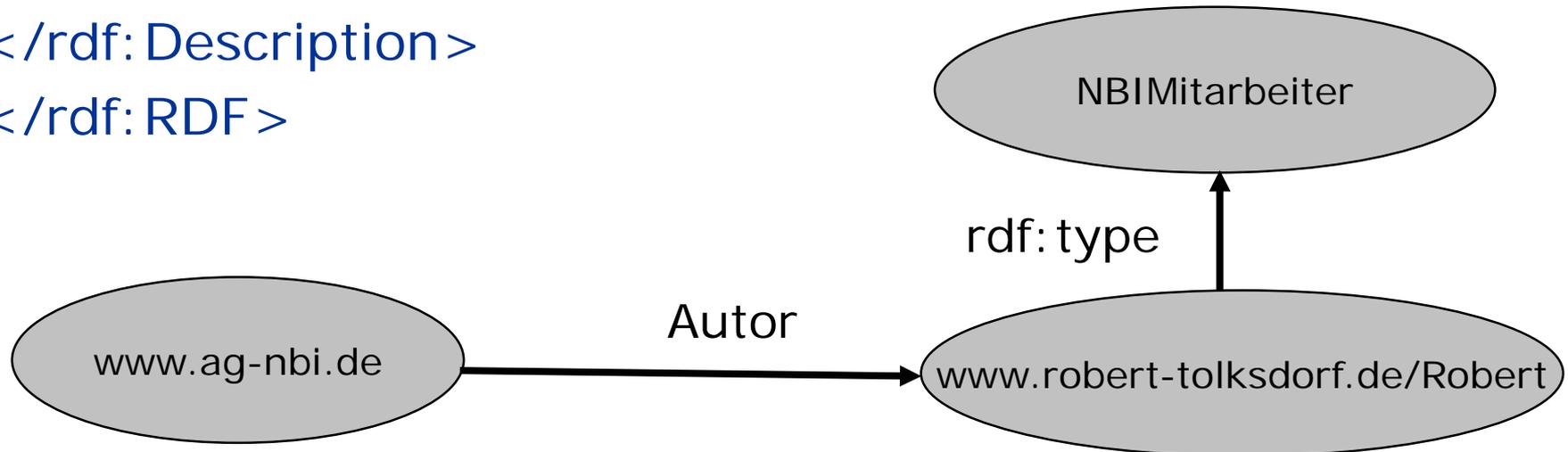
```
 <NBIMitarbeiter
```

```
 rdf:resource="http://robert.tolksdorf.de/Robert"/ >
```

```
 </Autor >
```

```
</rdf:Description >
```

```
</rdf:RDF >
```



- RDF
  - Sprache für die Darstellung von Aussagen im Web
  - definiert ein Datenmodell:
    - Ressourcen minimal typisiert
    - Semantik der Ressourcen minimal spezifiziert
- Notwendig
  - Erweiterung von RDF für die Beschreibung von semantisch komplexere Vokabularien

- Grundlegende RDF Mechanismen:  
einfache Aussagen auf vielfältige Weise treffen
- RDF Schema:  
einige Typen von Aussagen für nützliche  
Modellierungsaussagen in Schemas:
  - „Jede Webseite hat einen Autor“
  - „Webseiten sind elektronische Dokumente“

- Elektronischen Dokumente bilden eine Klasse:  

```
<rdf:Description rdf:ID="electronicDocument">
 <rdf:type rdf:resource=
 "http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```
- Web-Seiten sind elektronische Dokumente  

```
<rdf:Description rdf:ID="webPage">
 <rdf:type rdf:resource=
 "http://www.w3.org/2000/01/rdfschema#Class"/>
 <rdfs:subClassOf rdf:resource="#electronicDocument"/>
</rdf:Description>
```
- Web-Seiten haben eine URL  

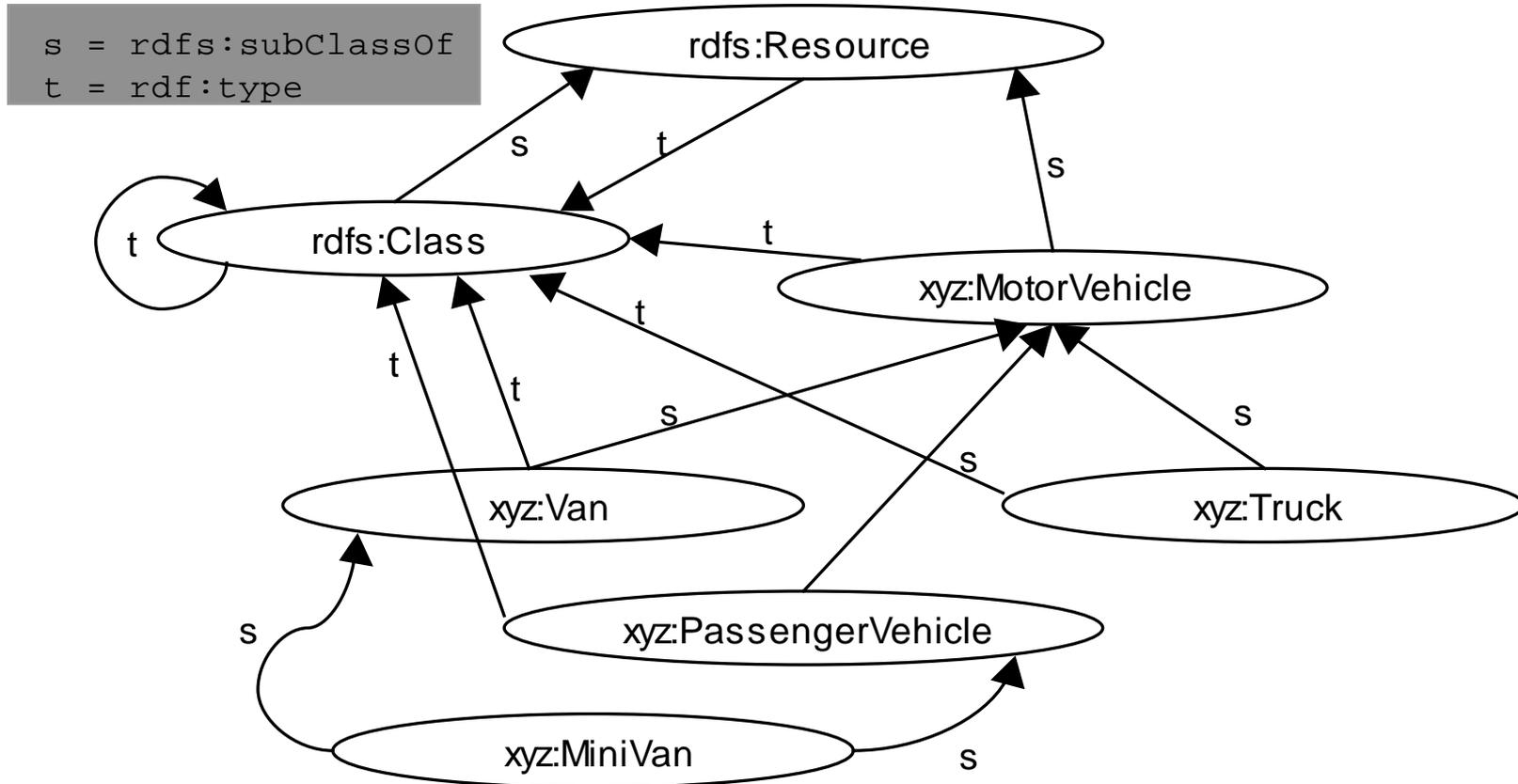
```
<rdf:Property rdf:ID="URL">
 <rdfs:domain rdf:resource="#webPage"/>
 <rdfs:range rdf:resource=
 "http://www.w3.org/2001/XMLSchema#anyURI"/>
</rdf:Property>
```

- `rdfs:Resource`  
Alles, was durch RDF Sätze beschrieben werden kann
- `rdf:type`  
Eigenschaft aller Dinge, die Klasse oder Typ angibt
  - Nutzerdefiniert:  

```
<rdf:Description rdf:ID="item10245">
 <rdf:type
 rdf:resource="http://www.example.com/terms/Tent"/>
</rdf:Description>
```
  - Vorgegeben:  

```
<rdf:Description rdf:ID="MotorVehicle">
 <rdf:type
 rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

- `rdfs:Class`  
Dinge, die Typen oder Klassen repräsentieren, also von anderen Dingen abstrahieren  
`<rdfs:Class rdf:ID="MotorVehicle"/>`
- `rdfs:subClassOf`  
Eigenschaft einer Klasse, die Generalisierung angibt  
`<rdfs:Class rdf:ID="MiniVan" >`  
`<rdfs:subClassOf rdf:resource="#Van"/>`  
`<rdfs:subClassOf rdf:resource="#PassengerVehicle"/>`  
`</rdfs:Class>`



[Bild: <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>]

- `rdfs:Literal`

### Die Klasse aller Werte

- Plain literals: UNICODE-Zeichenketten

```
<rdfs:label xml:lang="en" >
```

```
PowerSystemResource
```

```
</rdfs:label >
```

- Typed literals: spezifiziert den Datentyp eines Literals

```
<name rdf:datatype="&xsd:string" >RT</name >
```

- `rdf:Property`  
Alle Ressourcen, die Eigenschaften sind
- `rdfs:range`  
Einschränkende Eigenschaft Wertebereich
- `rdfs:domain`  
Einschränkende Eigenschaft Herkunftsbereich

```
<rdf:Description ID="registeredTo">
<rdf:type resource=
 "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdfs:domain rdf:resource="#MotorVehicle"/>
<rdfs:range rdf:resource="#Person"/>
</rdf:Description>
```

```
<rdf:Property rdf:ID="rearSeatLegRoom">
<rdfs:domain rdf:resource="#PassengerVehicle"/>
<rdfs:domain rdf:resource="#Minivan"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
```

- `rdfs:subPropertyOf`  
Eigenschaft ist Spezialisierung einer (oder mehrerer) anderer Eigenschaften

```
<rdf:Description ID="biologicalParent" >
 <rdf:type resource=
 "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/ >
</rdf:Description >
```

```
<rdf:Description ID="biologicalFather" >
 <rdf:type resource=
 "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/ >
 <rdfs:subPropertyOf
 rdf:resource="#biologicalParent"/ >
</rdf:Description >
```

- `rdfs:label`  
Menschenlesbarer Name der Ressource
- `rdfs:comment`  
Menschenlesbare Beschreibung der Ressource  

```
<rdfs:Class rdf:ID="PowerSystemResource" >
 <rdfs:label xml:lang="en" >
 PowerSystemResource
 </rdfs:label >
 <rdfs:comment >"A power system component that can be
either an individual element such as a switch or a set of
elements such as a substation. PowerSystemResources
that are sets could be members of other sets. [...]"
 </rdfs:comment >
</rdfs:Class >
```

- `rdfs:seeAlso`  
Verweist auf Ressource, die weitere Informationen über das Subjekt liefern kann

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns="http://xmlns.com/foaf/0.1/" >
```

```
<Person >
```

```
<name>Dan Brickley</name >
```

```
<workplaceHomepage rdf:resource="http://www.w3.org/" />
```

```
<homepage rdf:resource="http://rdfweb.org/people/danbri/" />
```

```
<rdfs:seeAlso rdf:resource="http://.../danbri-foaf.rdf"/>
```

```
</Person >
```

```
</rdf:RDF >
```

▪[<http://www.w3.org/2001/sw/Europe/talks/xml2003/slide3-3.html>]

- `rdfs:isDefinedBy`

Ressource, die das Subjekt definiert, z.B. ein Schema

```
<rdfs:Class
```

```
 rdf:about="http://jibbering.com/vocabs/image/#Area"
```

```
 rdfs:label="Area" rdfs:comment="An Area of an image." >
```

```
<rdfs:isDefinedBy
```

```
 rdf:resource="http://jibbering.com/vocabs/image/" />
```

```
</rdfs:Class>
```

▪ [\[http://jibbering.com/vocabs/image/index.rdf\]](http://jibbering.com/vocabs/image/index.rdf)

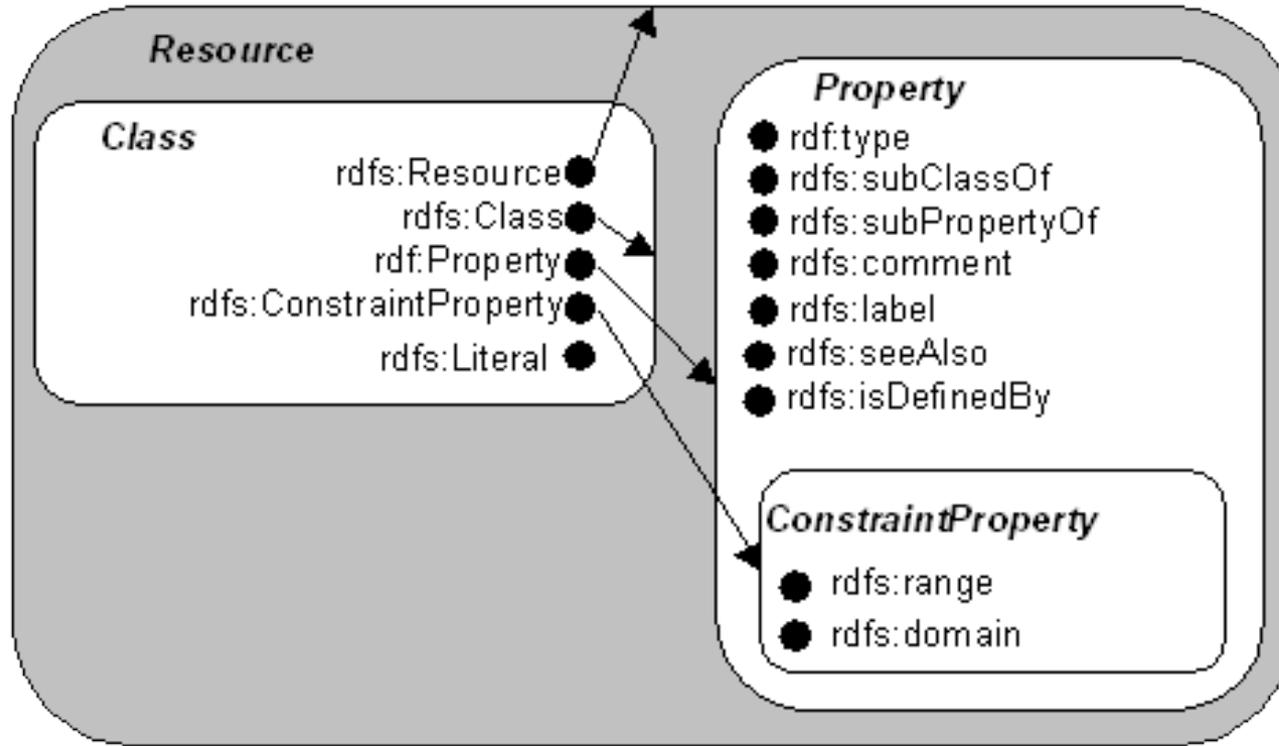


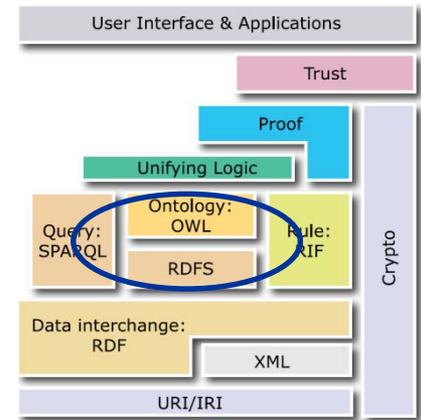
Abb.: © 2001 W3C

- Z.B.:
  - `rdf:Statement` ist vom Typ `rdfs:Class`
  - Die Property `rdf:type` ist eine Subklasse der Klasse `rdfs:Property`

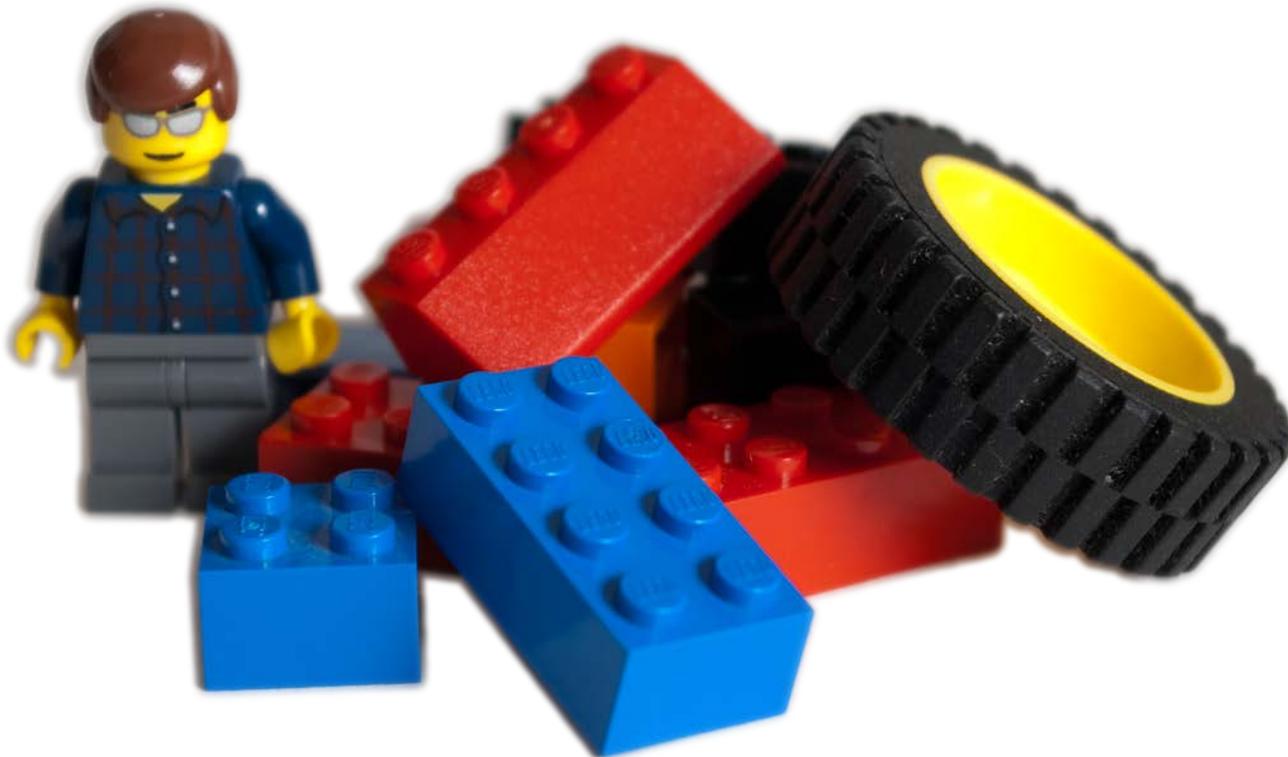
[Bild: <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>]

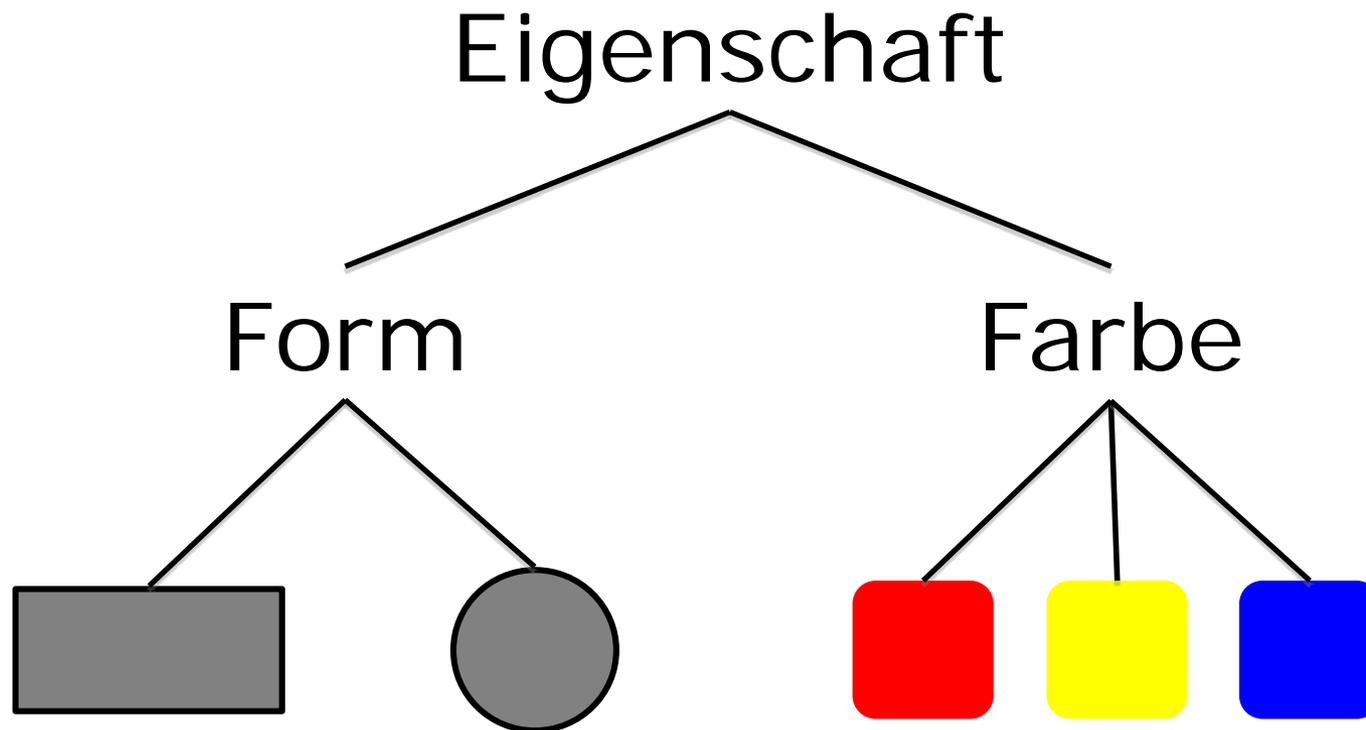
## Ontologien

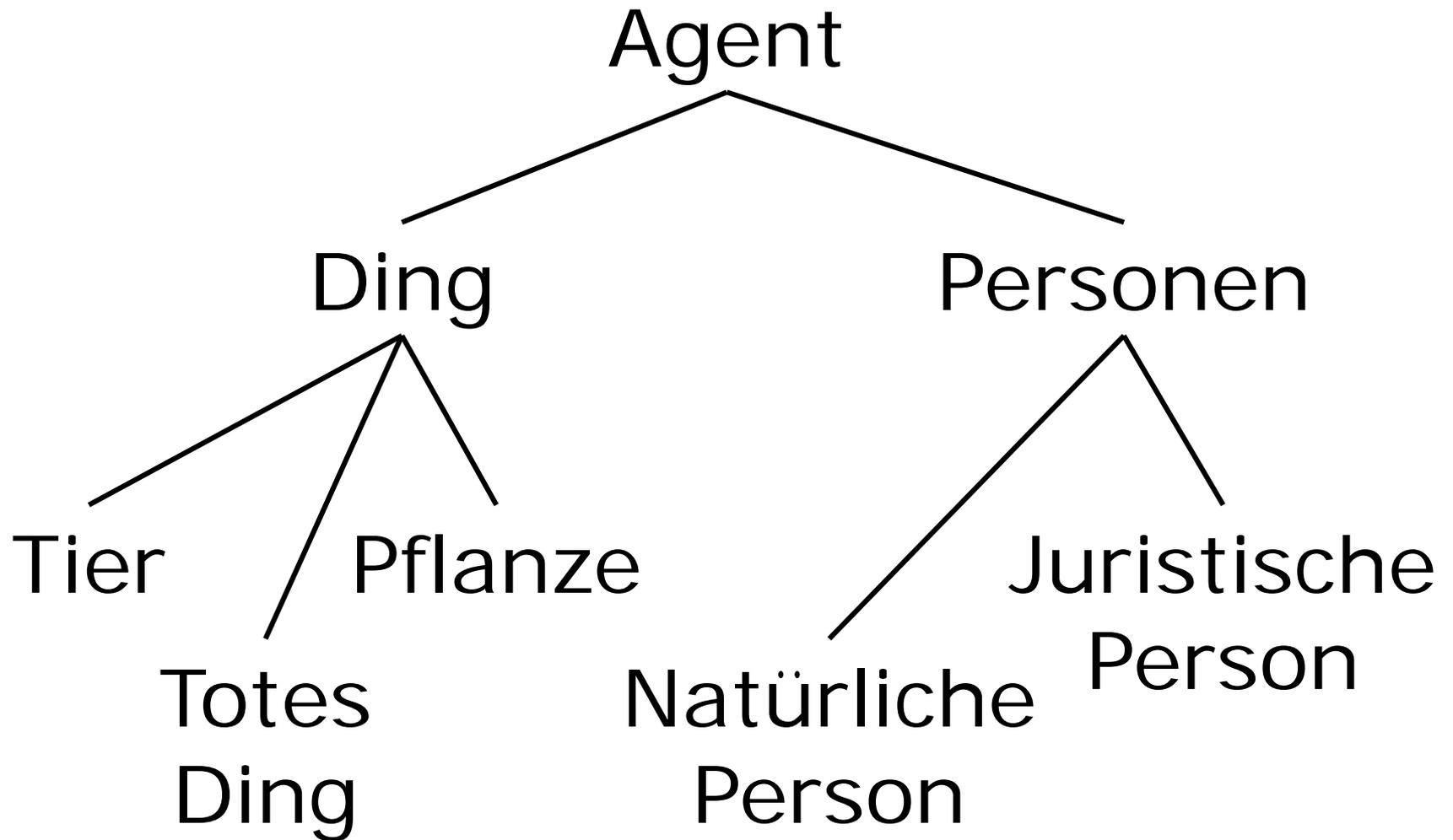
- Vokabulare
- Begriffsbeziehungen (Unterklasse, Untereigenschaft, Wertebereiche, ..., selbstdefinierte)
- Sprache für Web-Ontologien:
  - OWL – Web Ontology Language
    - Erweiterte Beschreibungsmöglichkeiten
    - In unterschiedlichen Komplexitäten (OWL-Lite, OWL-DL, OWL-Full)
    - mittlerweile OWL 2 mit feinerer Unterscheidung der Komplexität

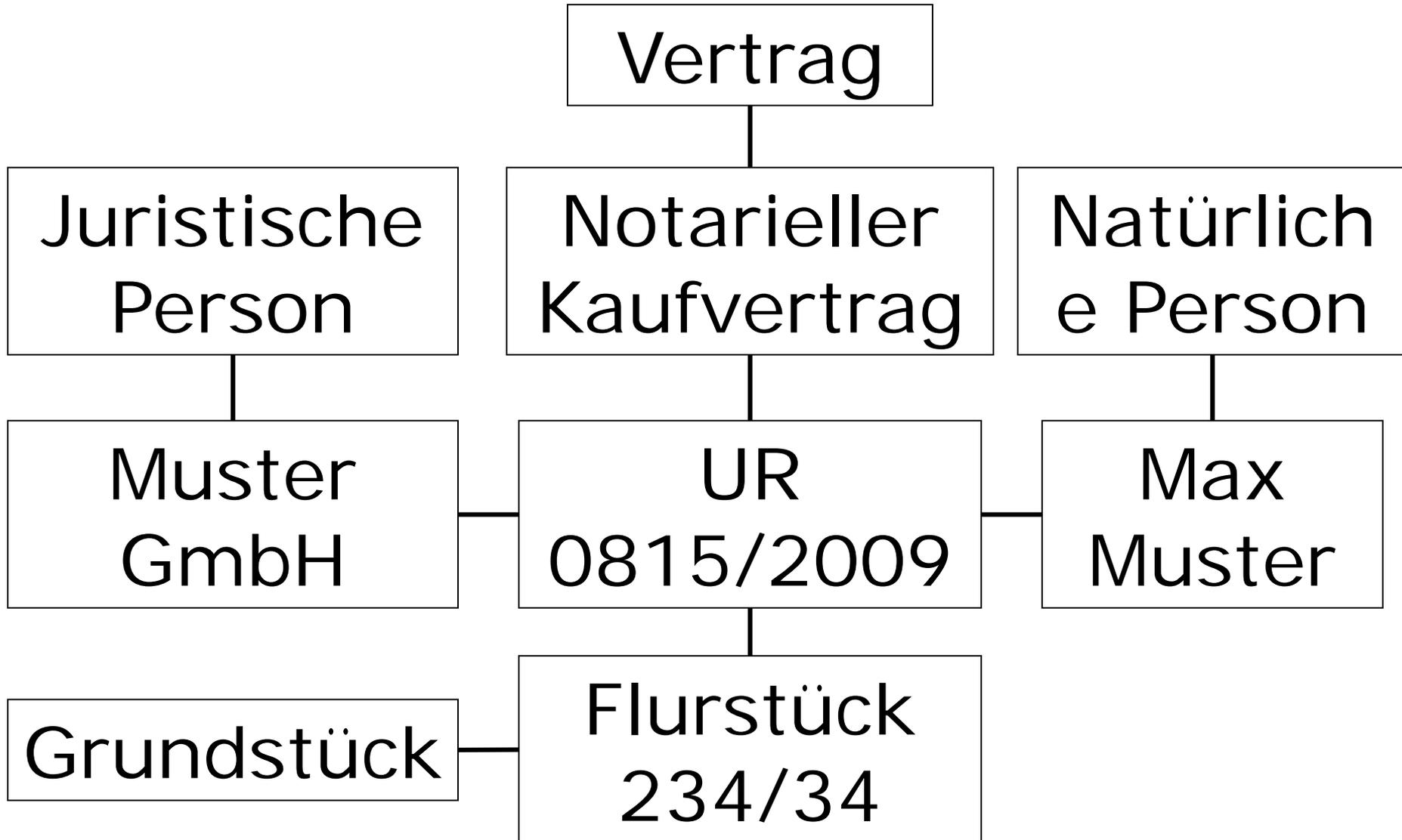


# Eine Domäne



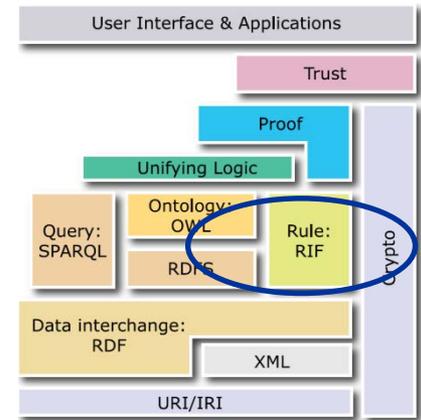




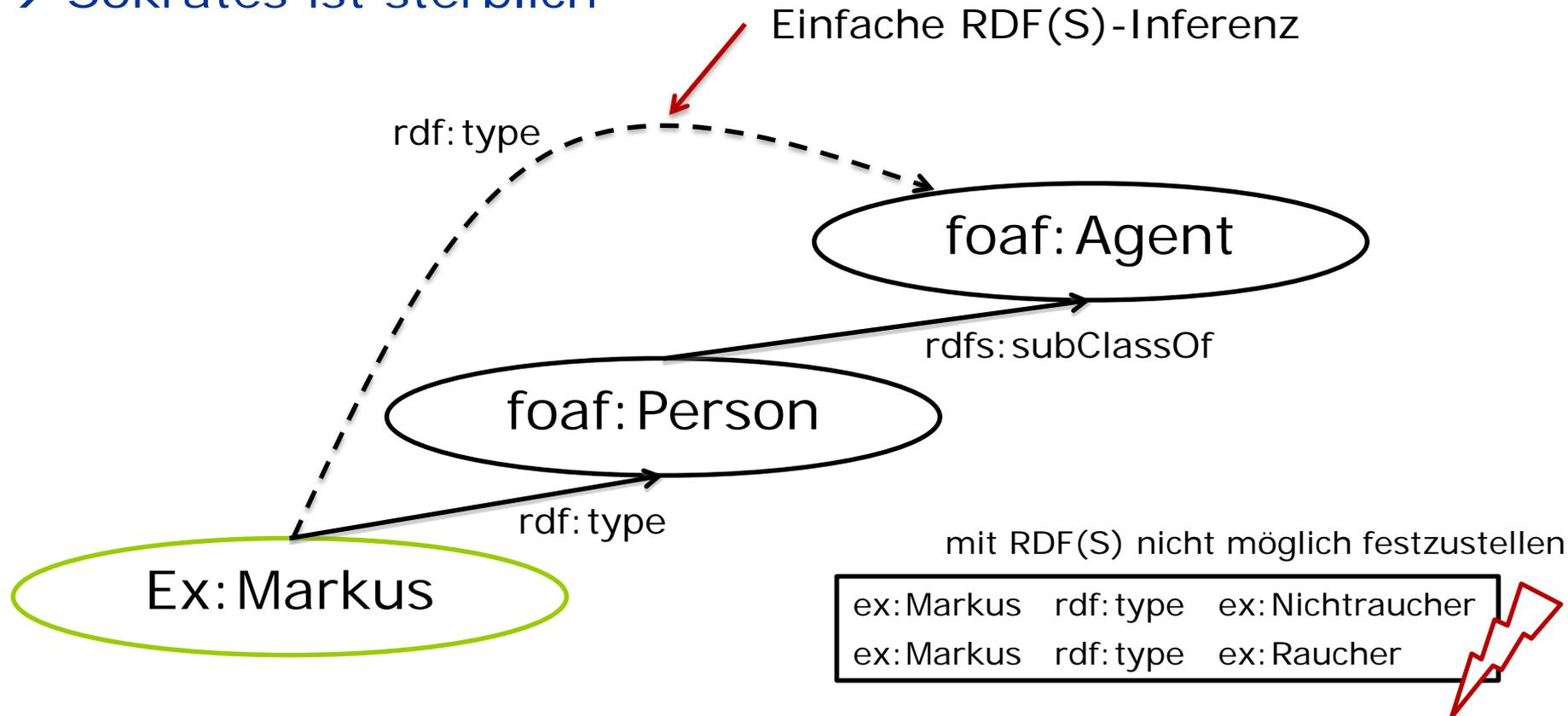


## Regelsprachen

- bilden die Grundlage für das logische schließen auf Basis semantischer Daten
- früher SWRL (echte Regelsprache für OWL) als Teil des Layer Cakes
- heute RIF als ein Austauschformat zwischen unterschiedlichen Regelsystemen



- Alle Menschen sind sterblich
  - Sokrates ist ein Mensch
- Sokrates ist sterblich



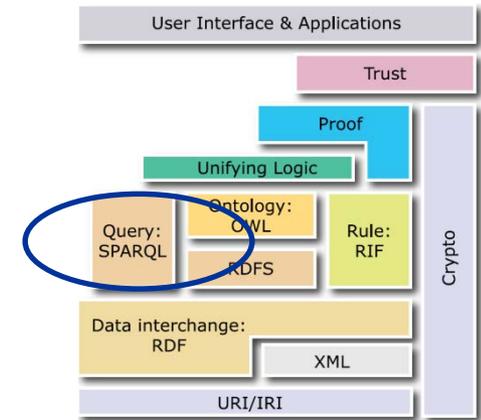
## Inferenz (cont.)

- Klassenäquivalenz
- Unterklassenbeziehung
- Klassendisjunktheit
- Globale Konsistenz
- Klassenkonsistenz
- Instanzüberprüfung
- Klasseninstanzen

## Anfragesprache SPARQL

- Dient zur Abfrage von Instanzdaten in einer RDF-Datenbank
- „Gib mir alle Menschen, die vor 1900 in Berlin geboren wurden“

- ```
SELECT ?name ?birth ?death ?person
WHERE {
  ?person dbpedia2:birthPlace <http://dbpedia.org/resource/Berlin> .
  ?person dbo:birthDate ?birth .
  ?person foaf:name ?name .
  ?person dbo:deathDate ?death
  FILTER (?birth < "1900-01-01"^^xsd:date) .
}
ORDER BY ?name
```



- Graph Pattern als Anfragemuster

```
SELECT DISTINCT ?player {
?s foaf:page ?player .
?s rdf:type <http://dbpedia.org/ontology/SoccerPlayer> .
?s dbpedia2:position ?position .
?s <http://dbpedia.org/property/clubs> ?club .
?club <http://dbpedia.org/ontology/capacity> ?cap .
?s <http://dbpedia.org/ontology/birthPlace> ?place .
?place ?population ?pop.
OPTIONAL {?s <http://dbpedia.org/ontology/number> ?tricot.}
Filter (?population in (<http://dbpedia.org/property/populationEstimate>,
<http://dbpedia.org/property/populationCensus>,
<http://dbpedia.org/property/statPop> ))
Filter (xsd:int(?pop) > 10000000 ) .
Filter (xsd:int(?cap) > 40000 ) .
Filter (?position = "Goalkeeper"@en || ?position =
<http://dbpedia.org/resource/Goalkeeper_%28association_football%29> ||
?position = <http://dbpedia.org/resource/Goalkeeper_%28football%29>)
} Limit 1000
```

Beispiel: Finde alle Fußballspieler, die bei einem Verein spielen, der ein Stadion mit mehr als 40.000 Plätzen hat und die in einem Land mit mehr als 10 Millionen Einwohnern geboren wurden.

SPARQL Ergebnis (nicht RDF)

```
<sparql xmlns=http://www.w3.org/2005/sparql-results#
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation=http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd>
  <head>
    <variable name="player"/>
  </head>
  <results distinct="false" ordered="true">
<result>
  <binding name="player">
    <uri>http://en.wikipedia.org/wiki/Petar_Radenkovi%C4%87</uri>
  </binding>
</result>
<result>
  <binding name="player">
    <uri>http://en.wikipedia.org/wiki/Michal_Vorel</uri>
  </binding>
</result>
...
</results>
</sparql>
```

Logik

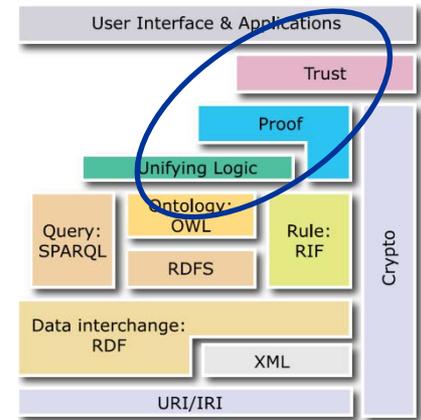
- Semantik auf logischer Basis
- Ableitungsregeln

Proof

- Konsistenz
- Ableitung (Inferenz)

Trust

→ Immer noch in der Forschung





Semantic Web → Beispiel

Organisatorisch:

- Stellenanbieter nutzen gemeinsames kontrolliertes Vokabular für die Annotierung von Stellenangeboten
- Stellensuchende nutzen gleiches Vokabular für Stellengesuche/Bewerberprofile

Technisch:

- Einfache Annotation → Reichere Annotation → Ersatz von Freitext durch RDF
- Stellenangebote direkt auf der Web-Seite des Unternehmens
- Semantische Suchmaschinen:
 - sammeln Informationen
 - Vergleich auf Basis von semantischen Informationen (Semantic Matching)

- Mit RDF und Bezug auf gemeinsames Vokabular (z.B. abgeleitet von HR-XML)

```
<html>
```

```
<head>
```

```
<rdf:RDF xmlns:rdf="...#" xmlns:jpp="...#">  
  <jpp:JobPosting  
    rdf:about="http://www.example.org/jp1.html"/>  
</rdf:RDF>
```

```
</head>
```

```
<body>
```

```
  ...Job posting in free text...
```

```
</body>
```

```
</html>
```

- Suchmaschinen können so Stellenangebote identifizieren

```
<html>
```

```
<head>
```

```
<rdf:RDF xmlns:rdf="...#" xmlns:jpp="...#" xmlns:skills="...#">  
  <jpp:JobPosting  
    rdf:about="http://www.example.org/jp1.html"/>  
  <jpp:requiredCompetence>  
    <skills:Java>  
      <skills:hasCompetenceLevel rdf:resource="...#expert"/>  
    </skills:Java>  
  </jpp:requiredCompetence>  
</rdf:RDF>
```

```
</head>
```

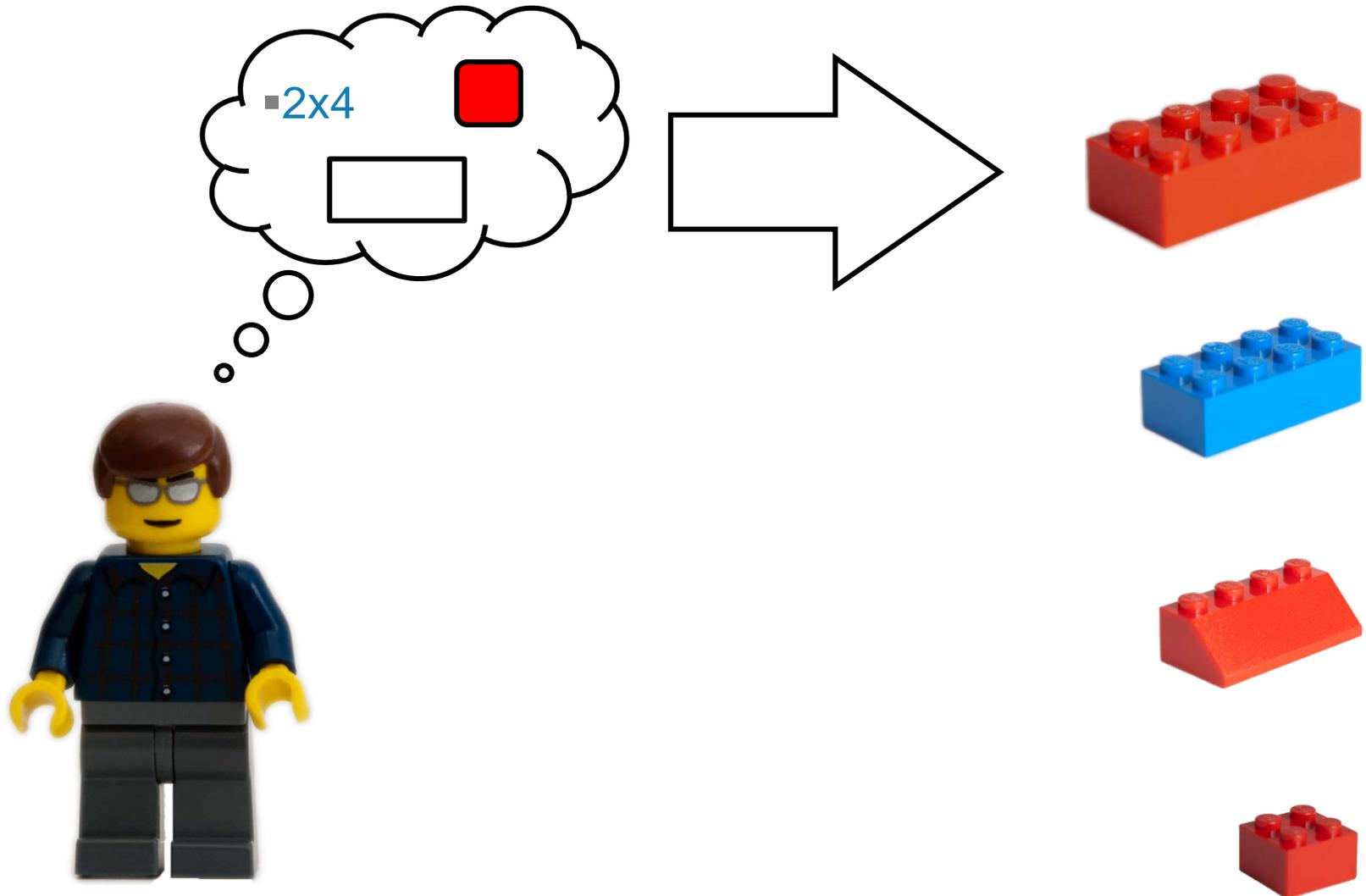
```
<body>
```

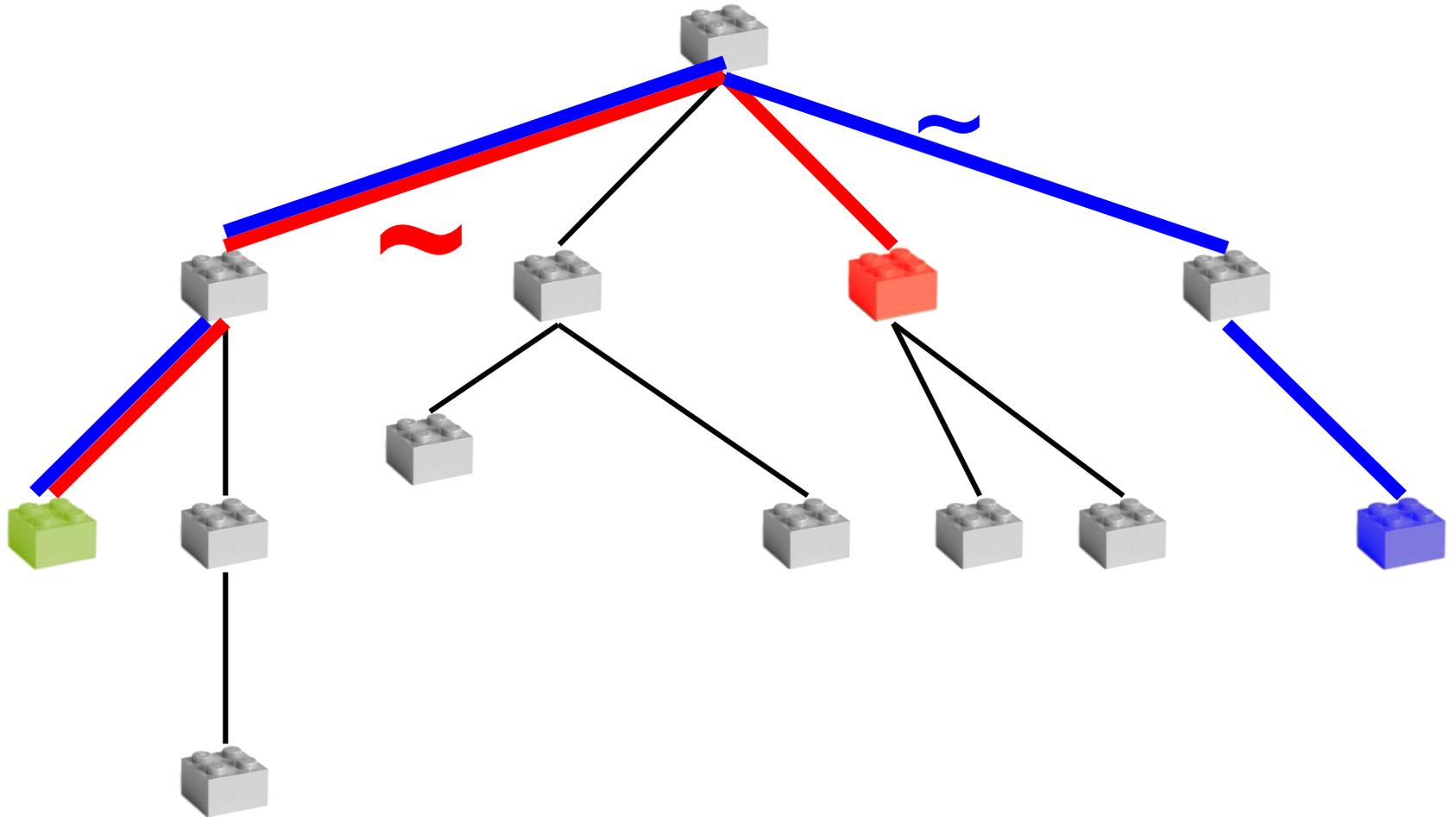
```
  ... Job posting in free text ...
```

```
</body>
```

```
</html>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="...#" xmlns:jpp="...#" xmlns:skills="...#">
  <jpp:JobPositionPosting rdf:about="#JobPositionPostingId-inf-44">
    <jpp:hasHiringOrganisation>
      <org:Organisation>
        <org:name>Freie Universität Berlin</org:name>
      </org:Organisation>
    </jpp:hasHiringOrganisation>...
    <jpp:requiredCompetence>
      <skills:Java>
        <skills:hasCompetenceLevel rdf:resource="...#expert"/>
      </skills:Java>
    </jpp:requiredCompetence>...
  </jpp:JobPositionPosting>...
</rdf:RDF>
```





Matching



$\sim 1,0$



$\sim 0,9$



$\sim 0,4$



XML

- XML heute omnipräsent, wenn auch nicht immer sichtbar

RDF

- HTML-Seiten und XML-Dokumente werden erstellt, aber noch nicht so viel RDF
- neue HTML-Versionen integrieren RDF als sogenanntes RDFa
 - Jedes HTML-Element kann dann RDF-Meta-Informationen haben

- Genutzte HTML Attribute

- rel
 - Beziehung zwischen zwei Ressourcen, ein Prädikat
- rev
 - Beziehung zwischen zwei Ressourcen, ein Prädikat
- href
 - Die Resource die in der Beziehung steht, das Objekt
- src
 - Die eingebettete Resource die in der Beziehung steht, das Objekt
 - t.html: `<a xmlns:cc="http://creativecommons.org/licenses/"
rel="cc:license"
href="http://creativecommons.org/licenses/by/nc-nd/3.0/">`
- Verarbeitbares *RDF Tripel*:
 - Subject: `http://t.html>`
 - Prädikat: `cc:licence`
 - Objekt: `http://creativecommons.org/licenses/by/nc-nd/3.0/`

3 Levels von Markup im Web

