

GRUNDLAGEN SEMANTIC WEB

Lehrveranstaltung im WS09/10
Seminar für Computerlinguistik
Universität Heidelberg

Dr. Sebastian Rudolph
Institut AIFB
Universität Karlsruhe

AGENDA



- Vorstellung des Dozenten
- Organisatorisches zur Vorlesung
- Was ist das "Semantic Web"?

AGENDA

- **Vorstellung des Dozenten**
- Organisatorisches zur Vorlesung
- Was ist das "Semantic Web"?

VORSTELLUNG DER DOZENTEN



Dr. Sebastian Rudolph

- 1995 – 2000 Studium Lehramt
Mathematik/Physik/Informatik,
TU Dresden
- 2000 – 2003 Stipendiat im Graduiertenkolleg 334,
TU Dresden
- 2003 – 2005 wissenschaftlicher Mitarbeiter an der
Professur für die Psychologie des
Lehrens und Lernens, TU Dresden
- 2006 Promotion in Mathematik, TU Dresden
- seit 2006 als Postdoc, seit 2008 als Projektleiter am AIFB



Themen:

formale Aspekte der
Wissensverarbeitung
Logik
Komplexitätstheorie
Formale Begriffsanalyse
NLP
E-Learning

AGENDA

- Vorstellung des Dozenten
- **Organisatorisches zur Vorlesung**
- Was ist das "Semantic Web"?

ORGANISATORISCHES: ZEIT UND ORT



- 5.- 9. Oktober 2009, täglich
 - vormittags, 9:00 - 13:00
 - nachmittags, 14:00 - 16:00 (außer Mo.)
 - Vorlesung & Übung flexibel im Wechsel
- Ort: Im Neuenheimer Feld 306, SR 14
- Webseite:
http://semantic-web-grundlagen.de/wiki/GSW_WS09/10

ORGANISATORISCHES: INHALT



Einleitung und Ausblick

XML und URIs

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

OWL - Semantik und Reasoning

OWL 2

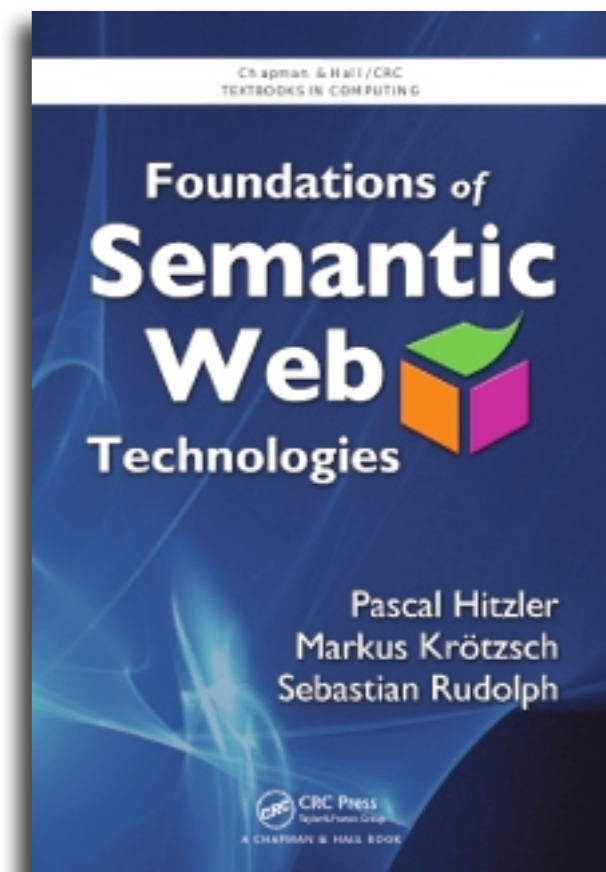
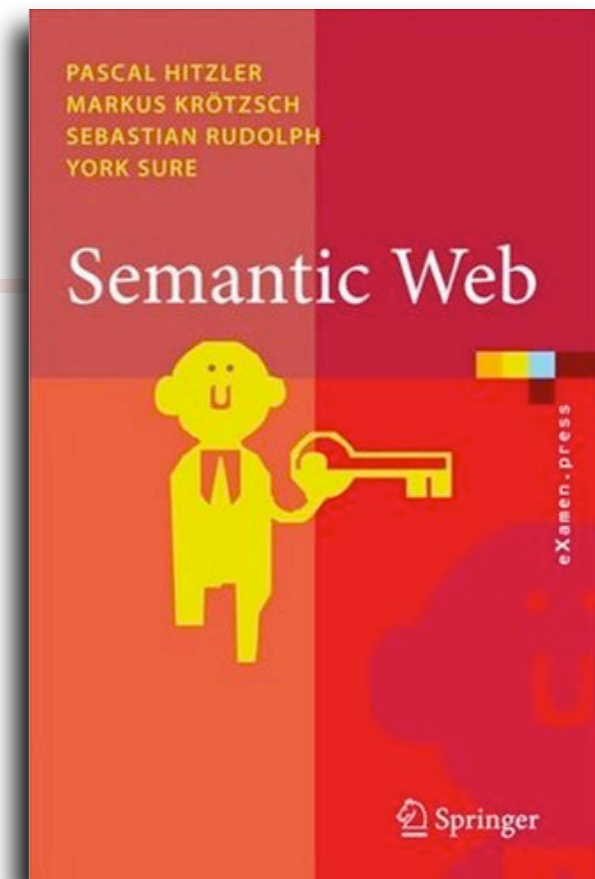
SPARQL - Syntax und Intuition

Semantik von SPARQL und konjunktive Anfragen

Semantic Web und Regeln

LITERATUR

- Hitzler, Krötzsch, Rudolph, Sure
"Semantic Web. Grundlagen."
Springer-Verlag
- Hitzler, Krötzsch, Rudolph
"Foundations of Semantic Web
Technologies"
CRC Press



AGENDA



- Vorstellung des Dozenten
- Organisatorisches zur Vorlesung
- **Was ist das "Semantic Web"?**

DAS WEB



Das Web flankiert den Übergang von der Industrie- zur Informationsgesellschaft und bietet die Infrastruktur für eine neue Qualität des Umgangs mit Information hinsichtlich Beschaffung wie auch Bereitstellung.



- hohe Verfügbarkeit
- hohe Aktualität
- geringe Kosten

Kommerzialisierung in allen Größenordnungen



ebay Einloggen oder Neu anmelden

Kategorien ▾ Motors Express Shops

zurück Kategorie: Computer > Apple > MacBook / MacBook Pro > MacBook Pro 15"

Apple Macbook Pro 15" 2,33 GHZ!!!! glossy

Bieter oder Verkäufer dieses Artikels? [Einloggen](#) zur Statusabfrage

 [Größeres Bild](#)

Aktuelles Gebot: **EUR 1.450,00**

Ihr Maximalgebot: EUR [Bieten >](#)
(Geben Sie mindestens EUR 1.460,00 ein)

Angebotsende: **54 Minuten 18 Sekunden**
(23.10.07 17:48:17 MESZ)

Versandkosten: **EUR 12,00**
Versicherter Versand
Service nach: [Deutschland](#)

Versand nach: Weltweit
Artikelstandort: Hamburg, Deutschland
Übersicht: [36 Gebot\(e\)](#)
Höchstbietender: [m***](#) (23 ★) 

Weitere Möglichkeiten: [Diesen Artikel beobachten](#)

Lassen Sie sich benachrichtigen per [Instant Messenger](#)
[An einen Freund senden](#)

Angebots- und Zahlungsdetails: [Anzeigen](#)

amazon.de  WUNSCHZETTEL | [MEIN KONTO](#) | [HILFE](#) | [IMPRESSUM](#)

HOME MEIN SHOP **BÜCHER** ENGLISH BOOKS ELEKTRONIK & FOTO MUSIK DVD KAUFEN & LEIHEN SOFTWARE GAMES KÜCHE, HAUS & GÄRTEN SPIELWAREN & KINDERWELT SPORT & FREIZEIT UHREN BABY **NEU** SCHUHE & HANDTASCHEN

ERWEITERTE SUCHE | STÖBERN | BESTSELLER | NEUHEITEN | Hörbücher | Taschenbücher | Fachbücher | Preis-Hits | Bücher verkaufen

Suche



Semantic Web. Grundlagen (eXamen.press) (Taschenbuch)
von [Pascal Hitzler](#) (Autor), [Markus Krötzsch](#) (Autor), [Sebastian Rudolph](#) (Autor), [York Sure](#) (Autor)

Preis: EUR 24,95 **Kostenlose Lieferung.** [Siehe Details.](#)

Verfügbarkeit: Dieser Artikel ist noch nicht erschienen. Reservieren Sie sich Ihr Exemplar jetzt und Sie erhalten es pünktlich zum Erscheinungstermin. Verkauf und Versand durch **Amazon.de**. Geschenkverpackung verfügbar. Zustellung durch **DHL**.

Noch 4 Tage bis zum Erscheinungstermin von [Harry Potter Band 7](#). Sichern Sie sich jetzt [Ihr Exemplar mit Liefergarantie -- sonst geschenkt!](#)

Preis: EUR 24,95
Vorbestellbar
Verkauf und Versand durch **Amazon.de**

Menge:

[Jetzt vorbestellen](#)

oder

[Loggen Sie sich ein](#), um 1-Click® einzuschalten.

[Auf meinen Wunschzettel](#)

[Auf die Hochzeitsliste](#)

[Einem Freund weitersagen](#)

[Größeres Bild](#)

[Verleger: So können Kunden in diesem Buch suchen.](#)

[Bewerten](#)
[Frage](#)
[Zu meinen bevorzugten Verkäufern hinzufügen](#)
[Andere Artikel des Verkäufers](#)

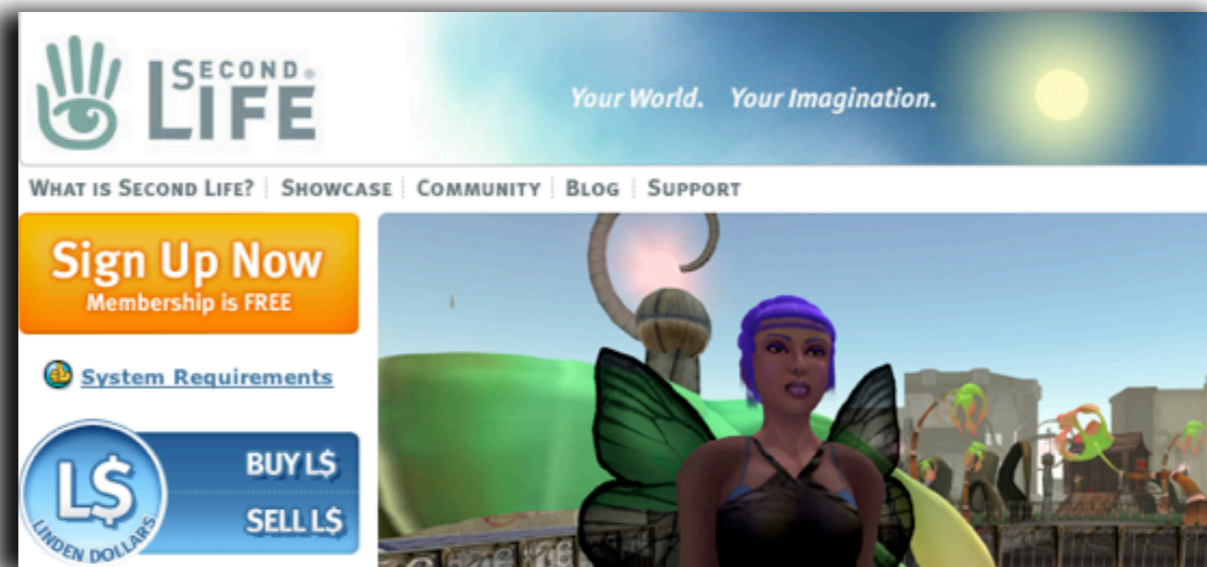
Sicher kaufen

1. **Sehen Sie sich das Bewertungsprofil des Verkäufers an**
Bewertungspunkte: 29 | 100% Positiv
[Bewertungskommentare lesen](#)
2. **Informieren Sie sich über den Käuferschutz**
Lesen Sie unsere [Tipps zum sicheren Kauf](#)

DAS WEB

weitere Lebensbereiche werden "webisiert":

- Behörden, Verwaltung (eGovernment)
- Ausbildung (eLearning, eEducation)
- Sozialkontakte (Social-Networking-Plattformen, Partnerbörsen)
- Alltag?



WARUM SEMANTIC WEB?



Exkurs: Syntax vs. Semantik

- **Syntax**
(von grch. συνταξις – *Zusammenstellung, Satzbau*) steht für die (normative) Struktur von Daten, d.h. sie charakterisiert, was "wohlgeformte" Daten sind.
- **Semantik**
(grch. σημαντικός – *zum Zeichen gehörend*) steht für die Bedeutung von Daten, d.h. sie charakterisiert beispielsweise, welche inhaltliche Schlussfolgerungen sich ziehen lassen.

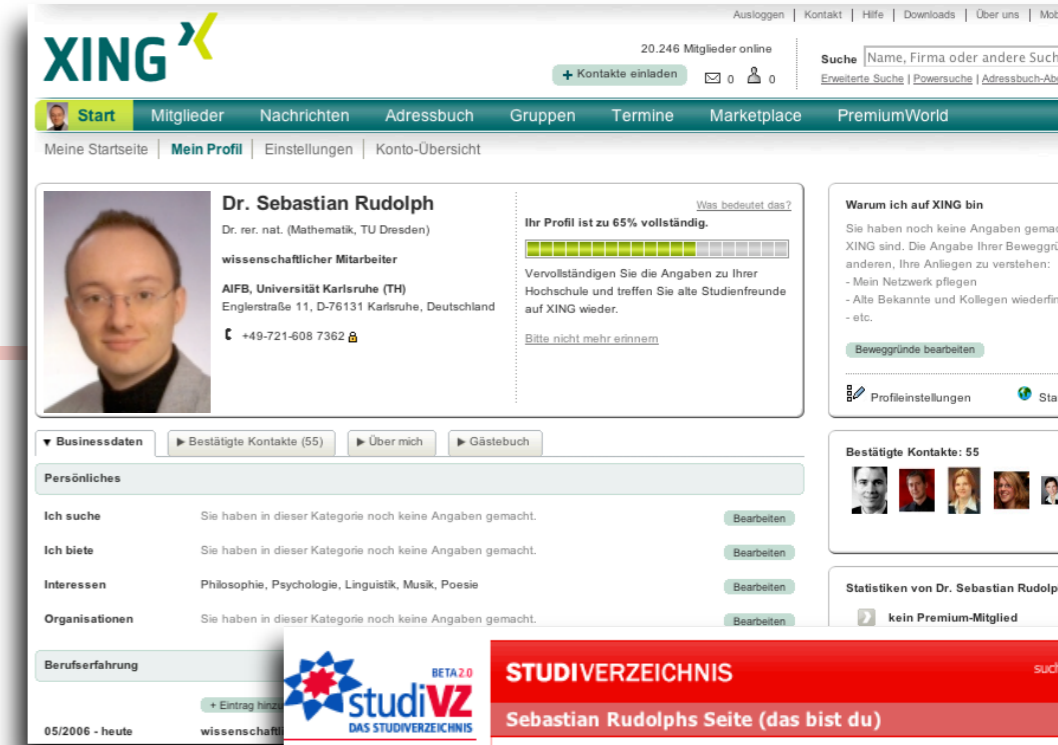
$4+)= ($
syntaktisch falsch
--

$3+4=12$
syntaktisch richtig
semantisch falsch

$3+4=7$
syntaktisch richtig
semantisch richtig

PROBLEME DES WEB

- Fülle an Informationen
- ausgerichtet auf Menschen als Endnutzer
 - Erfassen der Bedeutung einer Webseite
 - Unabhängig von konkreter Repräsentation
 - Bilden von Zusammenhängen



XING 20.246 Mitglieder online

Suche [Name, Firma oder andere Suchkriterien] [Erweiterte Suche](#) [Powersuche](#) [Adressbuch-Ab](#)

Start Mitglieder Nachrichten Adressbuch Gruppen Termine Marketplace PremiumWorld

Meine Startseite **Mein Profil** Einstellungen Konto-Übersicht

Dr. Sebastian Rudolph
Dr. rer. nat. (Mathematik, TU Dresden)
wissenschaftlicher Mitarbeiter
AIFB, Universität Karlsruhe (TH)
Englerstraße 11, D-76131 Karlsruhe, Deutschland
+49-721-608 7362

Ihr Profil ist zu 65% vollständig.

Vervollständigen Sie die Angaben zu Ihrer Hochschule und treffen Sie alte Studienfreunde auf XING wieder.

Bitte nicht mehr erinnern

Businessdaten Bestätigte Kontakte (55) Über mich Gästebuch

Persönliches

Ich suche Sie haben in dieser Kategorie noch keine Angaben gemacht. [Bearbeiten](#)

Ich biete Sie haben in dieser Kategorie noch keine Angaben gemacht. [Bearbeiten](#)

Interessen Philosophie, Psychologie, Linguistik, Musik, Poesie [Bearbeiten](#)

Organisationen Sie haben in dieser Kategorie noch keine Angaben gemacht. [Bearbeiten](#)

Berufserfahrung [+ Eintrag hinzufügen](#)

05/2006 - heute wissenschaftl.



studivZ BETA 2.0 DAS STUDIERVERZEICHNIS

Leute finden

Start [Leute finden](#)

Meine Seite [ändern](#)

Meine Freunde

Meine Fotos

Meine Gruppen

Nachrichtendienst

Mein Account

Privatsphäre

Klartext: Schluß mit der Gerüchteküche: studivZ war, ist und bleibt kostenlos.

Top Suche [Anzeige](#)

Arbeit

Englisch Lernen

Abo

Krankenversicherung

Zeitschrift

STUDIERVERZEICHNIS Sebastian Rudolphs Seite (das bist du)

Verbindung

Information

Account

Name: Sebastian Rudolph

Mitglied seit: 05.03.09

Letztes Update: 03.09.09

Allgemeines [\[bearbeiten \]](#)

Hochschule: Uni Karlsruhe

Status: Hochschüler

Geschlecht: männlich

Geburtsdatum: 09.03.1975

Heimatstadt: 09517

Letzte Schule: TU

Kontakt [\[bearbeiten \]](#)

Skype: [\[bearbeiten \]](#)

Persönliches [\[bearbeiten \]](#)

Lieblingszitat: [\[bearbeiten \]](#)

Arbeit [\[bearbeiten \]](#)

Firma: AIFB

Position / Job-Titel: [\[bearbeiten \]](#)

Du bist auf 7 Fotos verlinkt

Meine Seite bearbeiten

Mein Foto bearbeiten

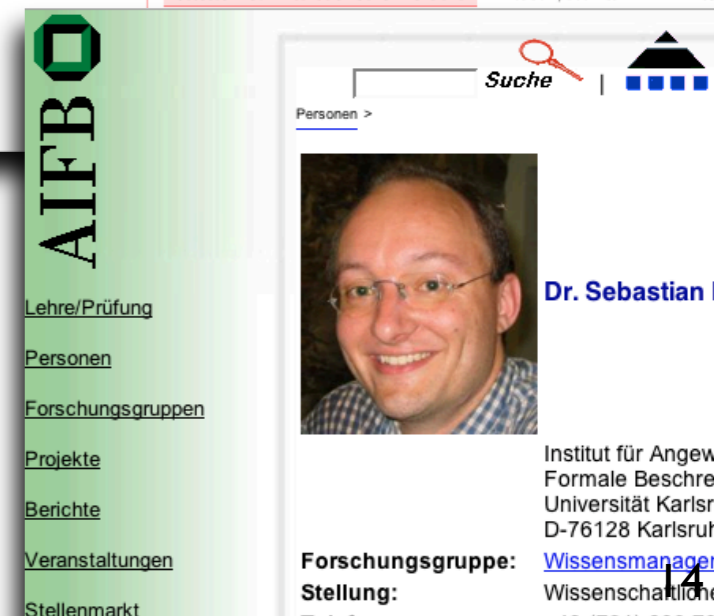
Privatsphäre-Einstellungen

Sebastian ist gerade ...

Halte deine Freunde auf dem Laufenden. [\[bearbeiten \]](#)

Freunde (gleiche Hochschule)

Sebastian hat 2 Freunde an der Uni Karlsruhe



AIFB

Suche [Suche](#)

Personen >

Dr. Sebastian Rudolph

Institut für Angewandte Formale Beschreibende Logik
Universität Karlsruhe
D-76128 Karlsruhe

Forschungsgruppe: [Wissensmanagement](#)

Stellung: Wissenschaftlicher Mitarbeiter

Lehre/Prüfung

Personen

Forschungsgruppen

Projekte

Berichte

Veranstaltungen

Stellenmarkt

PROBLEME DES WEB

- **Lokalisierung** von Information problematisch
- heutige Suchmaschinen gut, aber stichwortbasiert
- wünschenswert:
inhaltliche,
semantische Suche



PROBLEME DES WEB



- **Heterogenität** der vorhandenen Information auf verschiedensten Ebenen:
 - Zeichenkodierung (z.B. ASCII vs. Unicode)
 - verwendete natürliche Sprachen
 - Anordnung von Information auf Webseiten
- *Informationsintegration*

Semantic Web Technologies I & II: Intelligente Systeme im WWW

Winter 2007/08

Dozenten: PD Dr. Pascal Hitzler,
Dr. Sebastian Rudolph

Betreuer: M.Sc. Markus Krötzsch

Umfang:

2+1 SWS (Vorlesung+Übung),
4.5 Leistungspunkte

Zeit & Ort:

Vorlesung: wöchentlich Mittwoch 11:30 bis 13:00



위키백과

우리 모두의 백과사전
둘러보기

- 대문
- 사용자 모임
- 요즘 화제

문서 토론 편집 0

시맨틱 웹

위키백과 — 우리 모두의 백과사전

시맨틱 웹(Semantic Web)은 인터넷과 같은 분산환경에서 리소스 각종 화일, 서비스 등)에 대한 사이의 관계-의미 정보(Semantic Web)가 처리할 수 있는

PROBLEME DES WEB

- **implizites Wissen**, d.h. Informationen, sind nicht explizit spezifiziert, folgen aber aus der Kombination gegebener Daten
- formallogische Methoden erforderlichlich
- *automatisches Schlussfolgern*



PROBLEME DES WEB

Lösungsansätze:

- I. Ad hoc: Verwendung von KI-Methoden zur Auswertung bestehender unstrukturierter Informationen im Web
- II. A priori: Strukturierung der Web-Informationen zur Erleichterung der automatisierten Auswertung:
→ **Semantic Web**

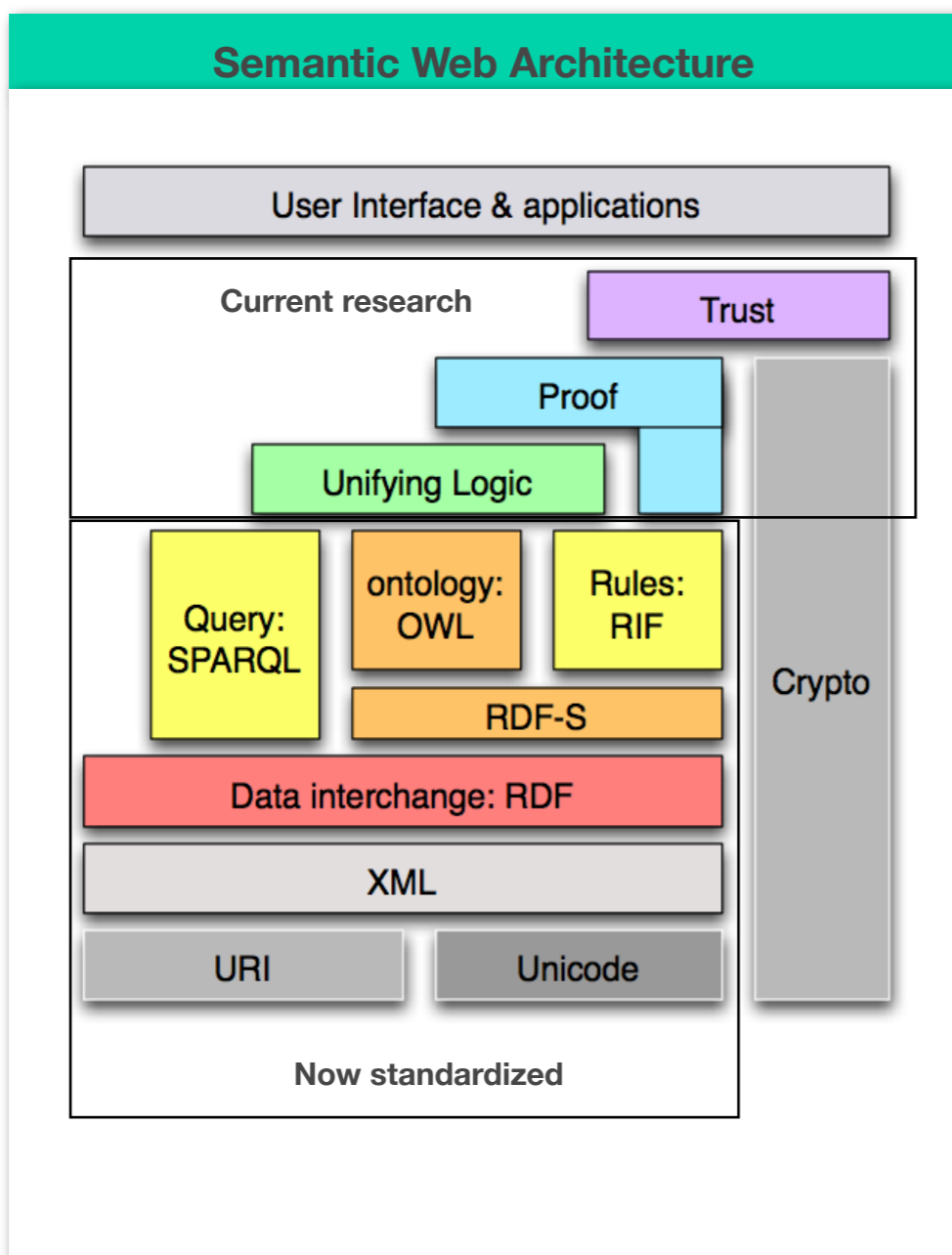
SEMANTIC WEB - ZUTATEN



Zwei essentielle Voraussetzungen zur Realisierung:

1. offene Standards zur Beschreibung von Informationen
 - klar definiert
 - flexibel
 - erweiterbar
2. Methoden zur Gewinnung von Informationen aus derlei Beschreibungen

SEMANTIC WEB - STANDARDS



Standardization Semantic Web

1994	<ul style="list-style-type: none"> • First public presentation of the Semantic Web idea
1998	<ul style="list-style-type: none"> • Start of standardization of data model (RDF) and a first ontology languages (RDFS) at W3C
2000	<ul style="list-style-type: none"> • Start of large research projects about ontologies in the US and Europe (DAML & Ontoknowledge)
2002	<ul style="list-style-type: none"> • Start of standardization of a new ontology language (OWL) based on research results
2004	<ul style="list-style-type: none"> • Finalization of the standard for data (RDF) and ontology (OWL)
2006	<ul style="list-style-type: none"> • Standardization of a query language (SPARQL, 6. April 2006) • Ongoing work on rule languages (SWRL, DL-safe rules, RIF) • Extension of OWL to OWL 1.1 / 2.0 • Ontology language of OMG based on UML (ODM)

XML UND URIs

Dr. Sebastian Rudolph

Einleitung und Ausblick

XML und URIs

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

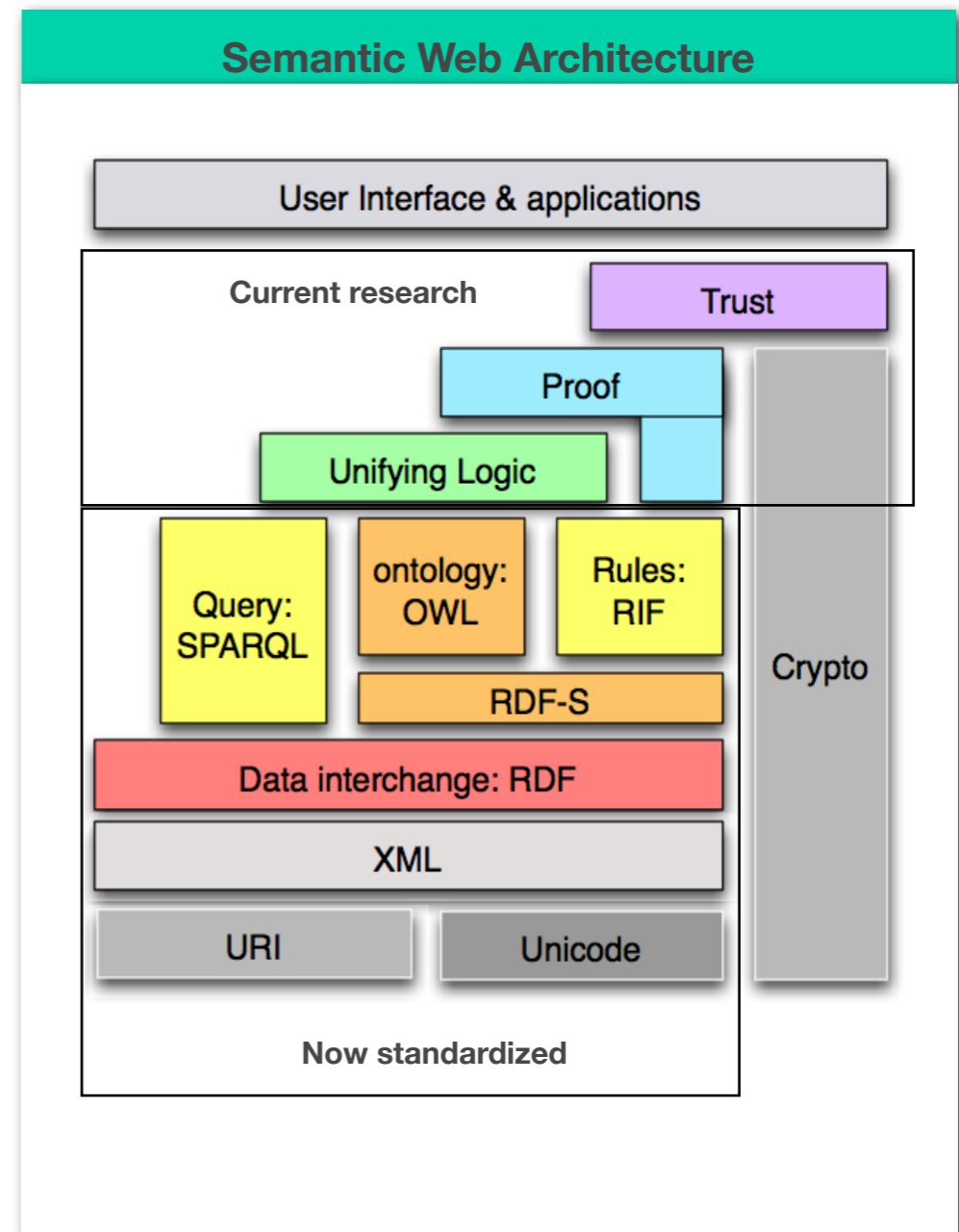
OWL - Semantik und Reasoning

SPARQL - Syntax und Intuition

Semantik von SPARQL und konjunktive Anfragen

OWL 1.1 - Syntax und Semantik

Semantic Web und Regeln



XML UND URIs

Dr. Sebastian Rudolph

Einleitung und Ausblick

XML und URIs

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

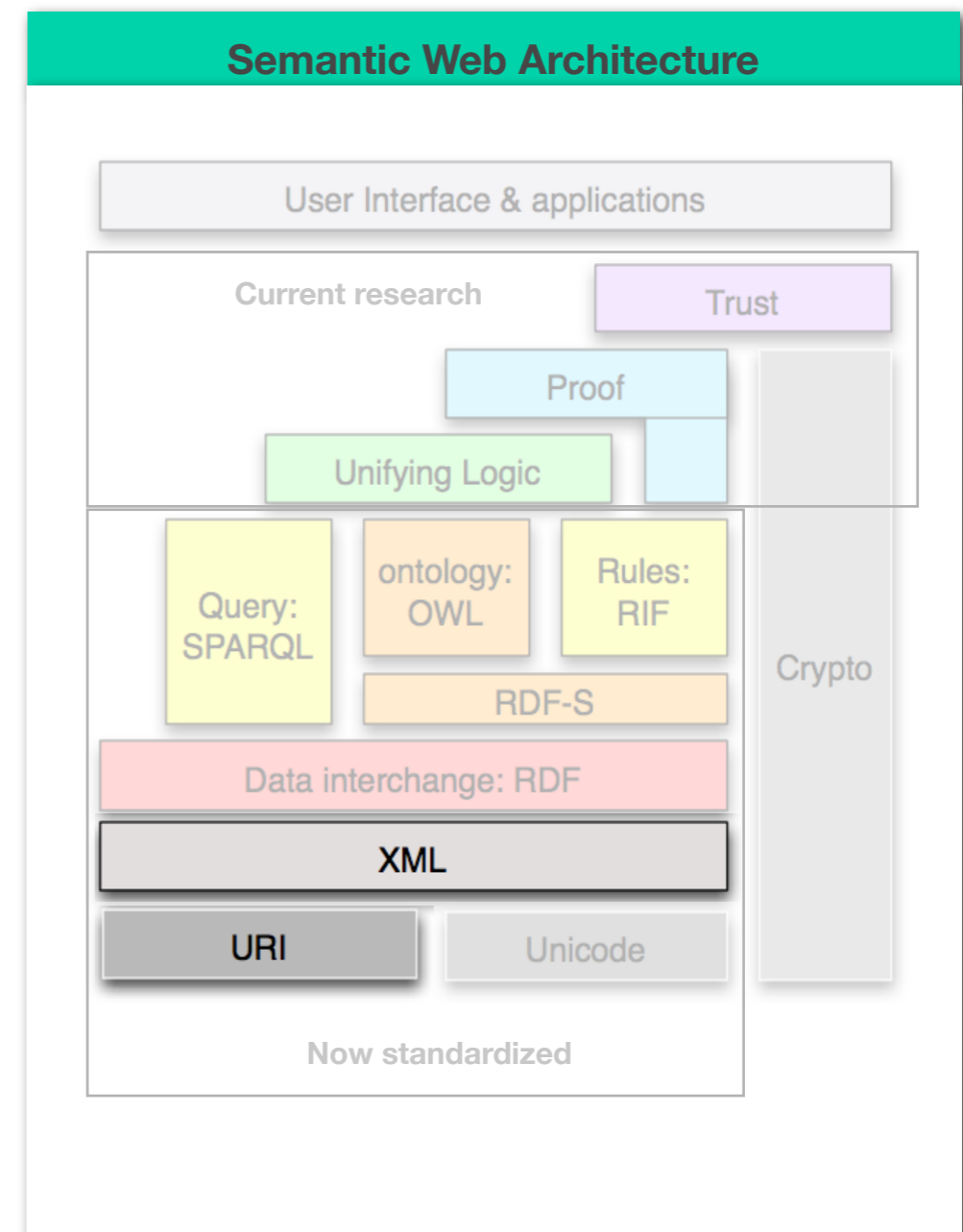
OWL - Semantik und Reasoning

SPARQL - Syntax und Intuition

Semantik von SPARQL und konjunktive Anfragen

OWL 1.1 - Syntax und Semantik

Semantic Web und Regeln



AGENDA



- XML - Motivation/Idee
- XML - Syntax
- URIs
- Namensräume

AGENDA



- XML - Motivation/Idee
- XML - Syntax
- URIs
- Namensräume

ANNOTATION MIT MARKUPSPRACHEN

- Grundidee des Markup: versehen von (unstrukturiertem) Text mit zusätzlicher Information (bzw. Struktur)
- synonym: *auszeichnen*, auch: *annotieren* von Text
- Text = Daten
Zusatzinformation = *Metadaten*

ANNOTATION MIT MARKUPSPRACHEN

- häufige Markup-Strategie: Einschließen des zu annotierenden Textes in sogenannte *tags* (engl.: Etikett, Schild):

<Tag-Bezeichner>...	Text	...</Tag-Bezeichner>
<i>öffnendes Tag</i>		<i>schließendes Tag</i>
- Zusatzinformation wird von verarbeitenden Programmen gelesen und interpretiert

ANNOTATION MIT MARKUPSPRACHEN



- prominentestes Beispiel: HTML
tags kodieren Darstellungsinformationen:
<i>Dieses Buch</i> hat den Titel Semantic Web Grundlagen.
- Darstellung im Browser:
Dieses Buch hat den Titel **Semantic Web Grundlagen**.
- Strategie auch geeignet zur inhaltlichen
Annotation, z.B.:
<Buch>Dieses Buch</Buch> hat den Titel <Titel>Semantic Web
Grundlagen</Titel>.

ANNOTATION MIT MARKUPSPRACHEN

- Verschachtelung von Tags erlaubt:

```
<Vorlesung>  
  <Titel>  
    XML und URIs  
  </Titel>  
  <Dozent>  
    <Titel>  
      Dr.  
    </Titel>  
    <Vorname>  
      Sebastian  
    </Vorname>  
    <Nachname>  
      Rudolph  
    </Nachname>  
  </Dozent>  
</Vorlesung>
```


ANNOTATION MIT MARKUPSPRACHEN



- Verschachtelung von Tags erlaubt:

```
<Vorlesung>
```

```
<Titel>
```

```
XML und URIs
```

```
</Titel>
```

```
<Dozent>
```

```
<Titel>
```

```
Dr.
```

```
</Titel>
```

```
<Vorname>
```

```
Sebastian
```

```
</Vorname>
```

```
<Nachname>
```

```
Rudolph
```

```
</Nachname>
```

```
</Dozent>
```

```
</Vorlesung>
```

ANNOTATION MIT MARKUPSPRACHEN



<Vorlesung>

<Titel>

XML und URIs

</Titel>

<Dozent>

<Titel>

Dr.

</Titel>

<Vorname>

Sebastian

</Vorname>

<Nachname>

Rudolph

</Nachname>

</Dozent>

</Vorlesung>

ANNOTATION MIT MARKUPSPRACHEN



Vorlesung

```
<Titel>  
  XML und URIs  
</Titel>  
<Dozent>
```

```
<Titel>  
  Dr.  
</Titel>  
<Vorname>  
  Sebastian  
</Vorname>  
<Nachname>  
  Rudolph  
</Nachname>
```

```
</Dozent>
```

ANNOTATION MIT MARKUPSPRACHEN

Vorlesung

Titel

XML und URIs

<Dozent>

<Titel>

Dr.

</Titel>

<Vorname>

Sebastian

</Vorname>

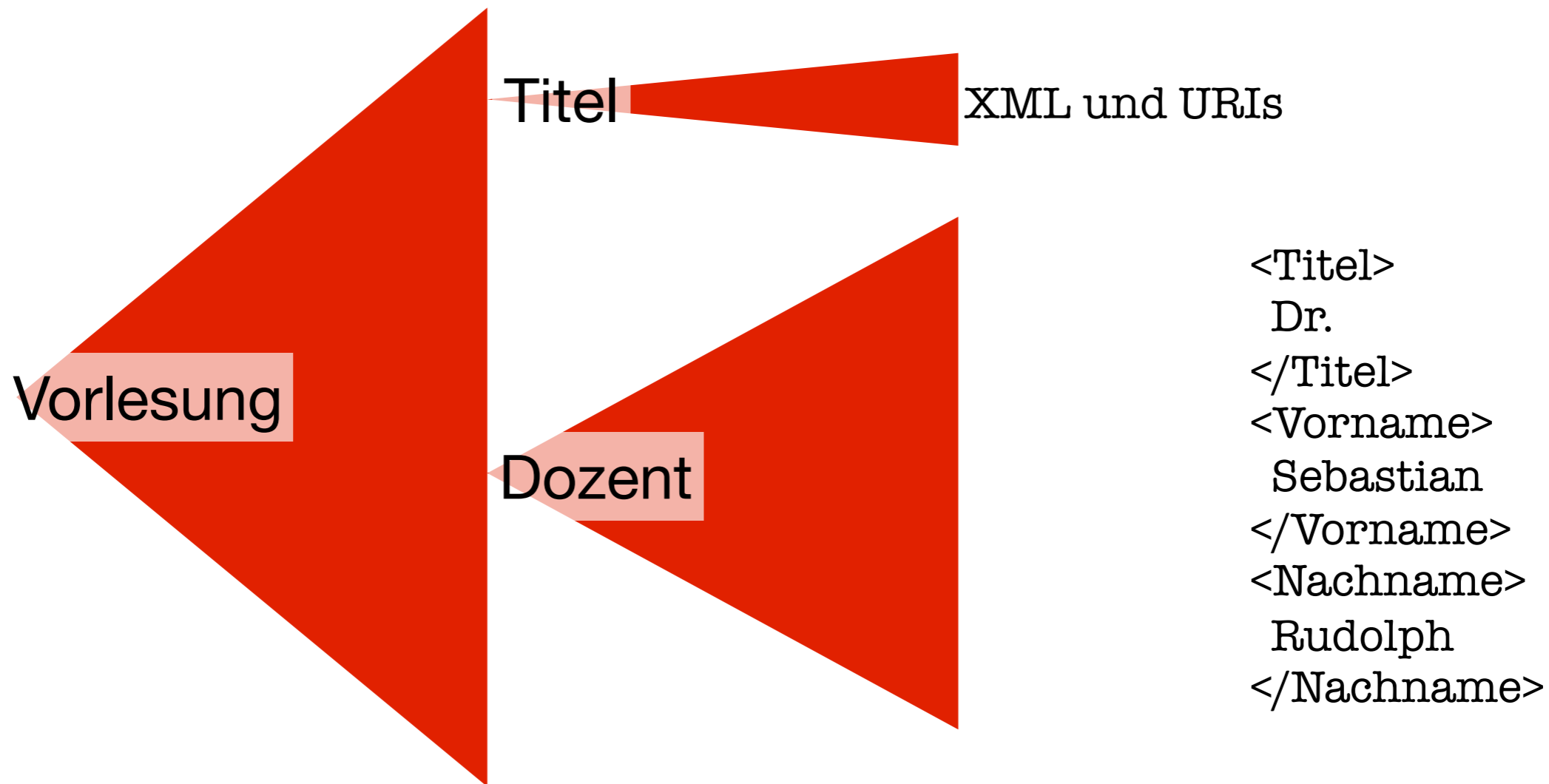
<Nachname>

Rudolph

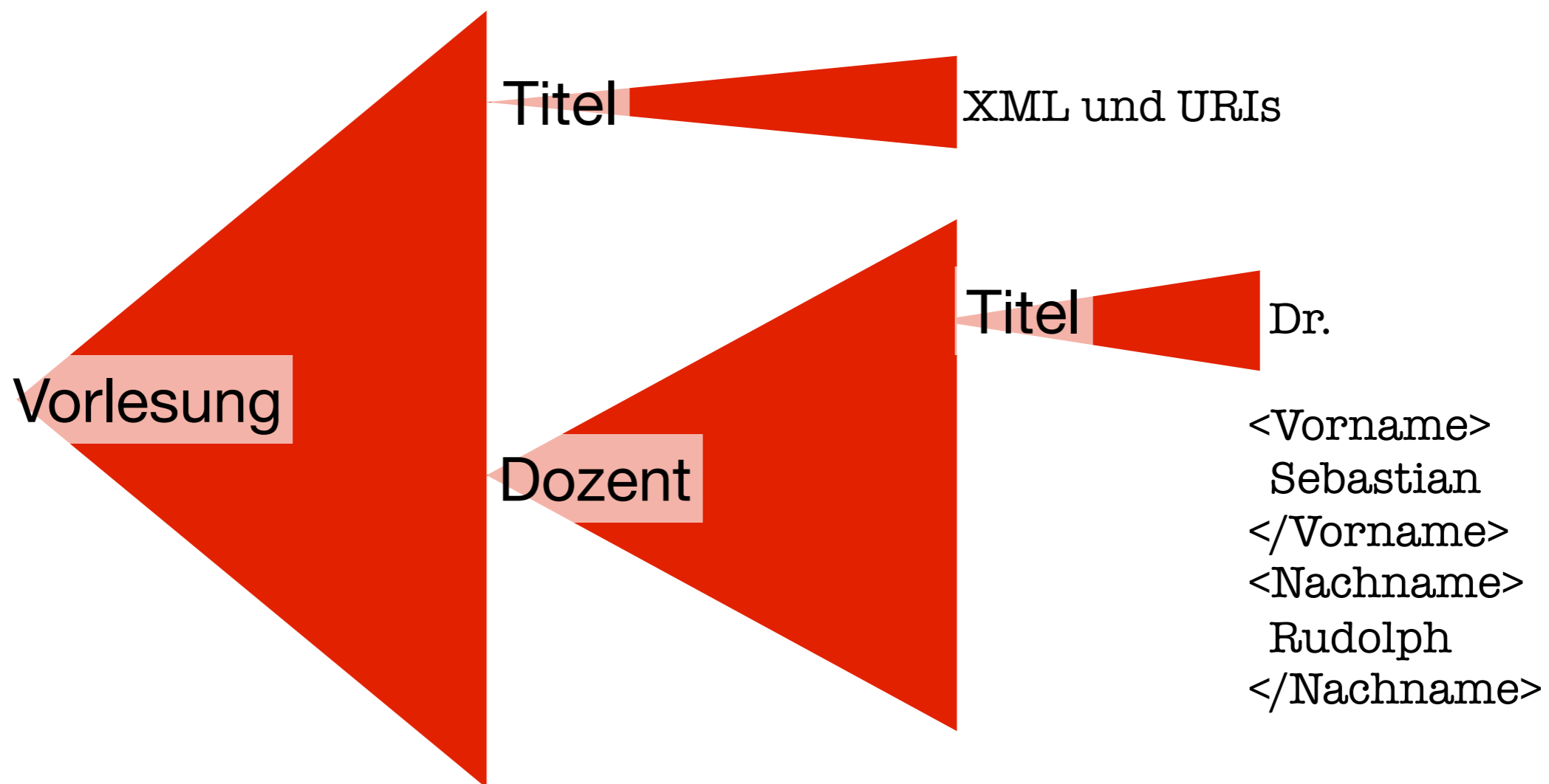
</Nachname>

</Dozent>

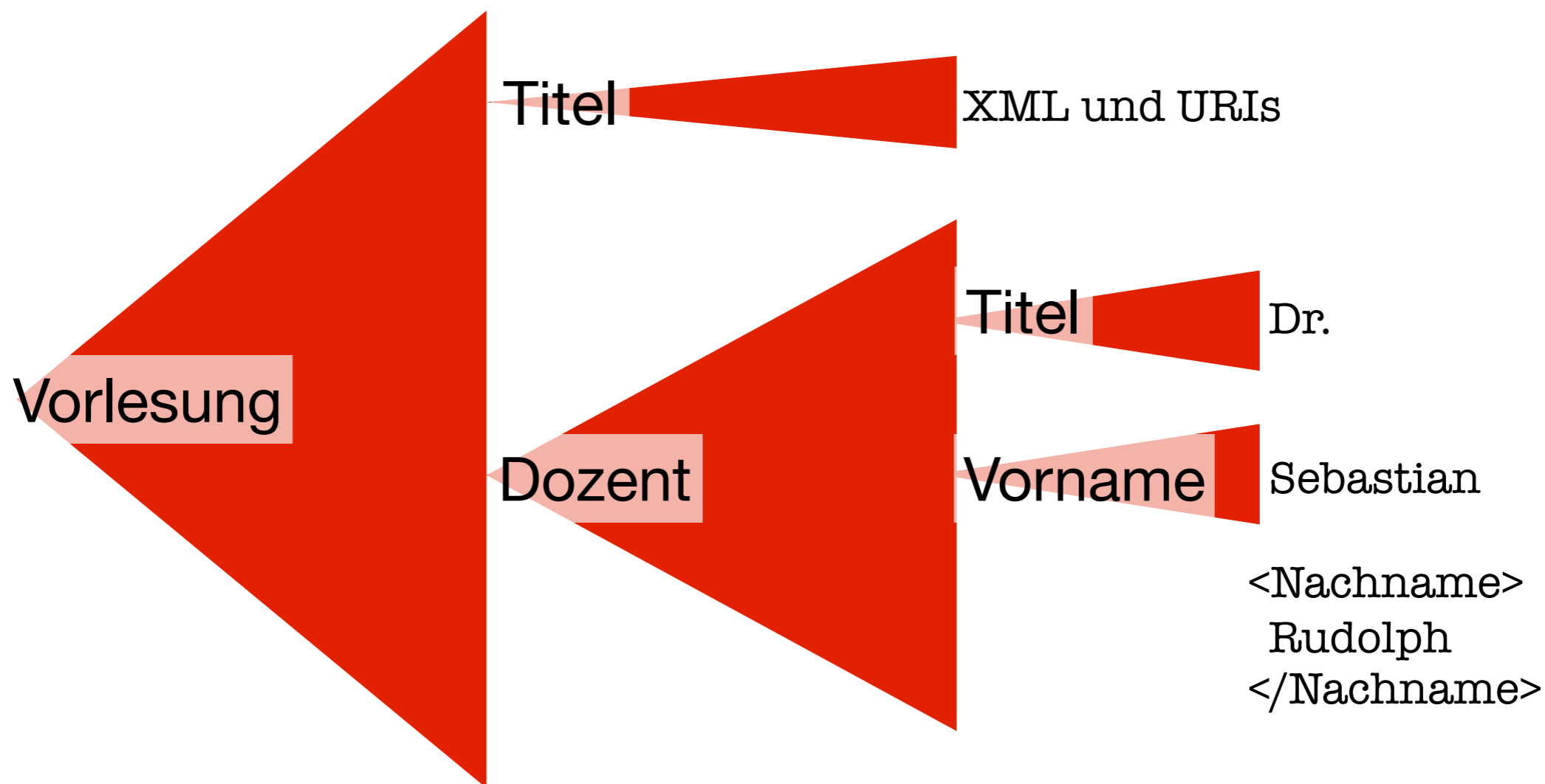
ANNOTATION MIT MARKUPSPRACHEN



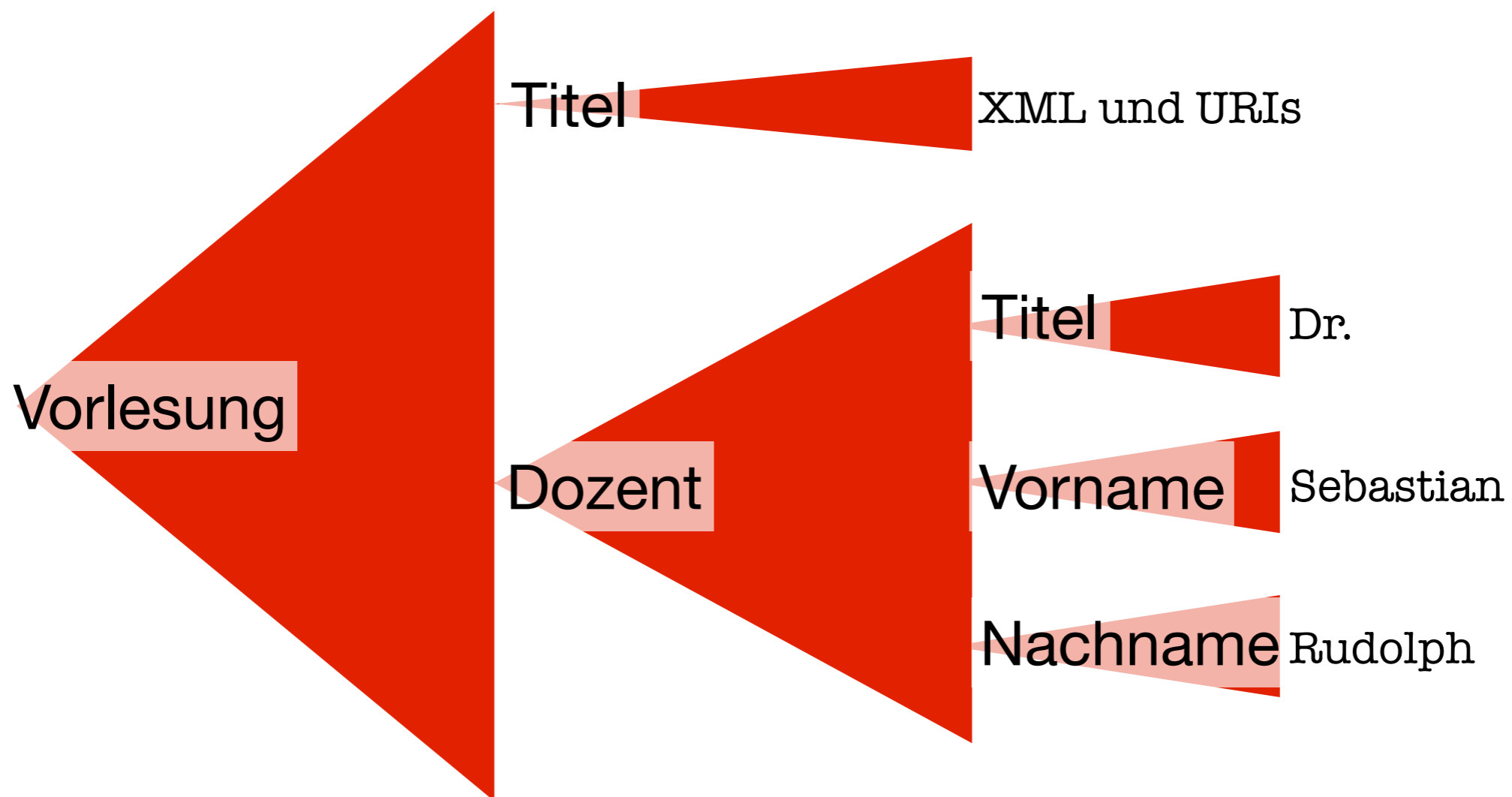
ANNOTATION MIT MARKUPSPRACHEN



ANNOTATION MIT MARKUPSPRACHEN

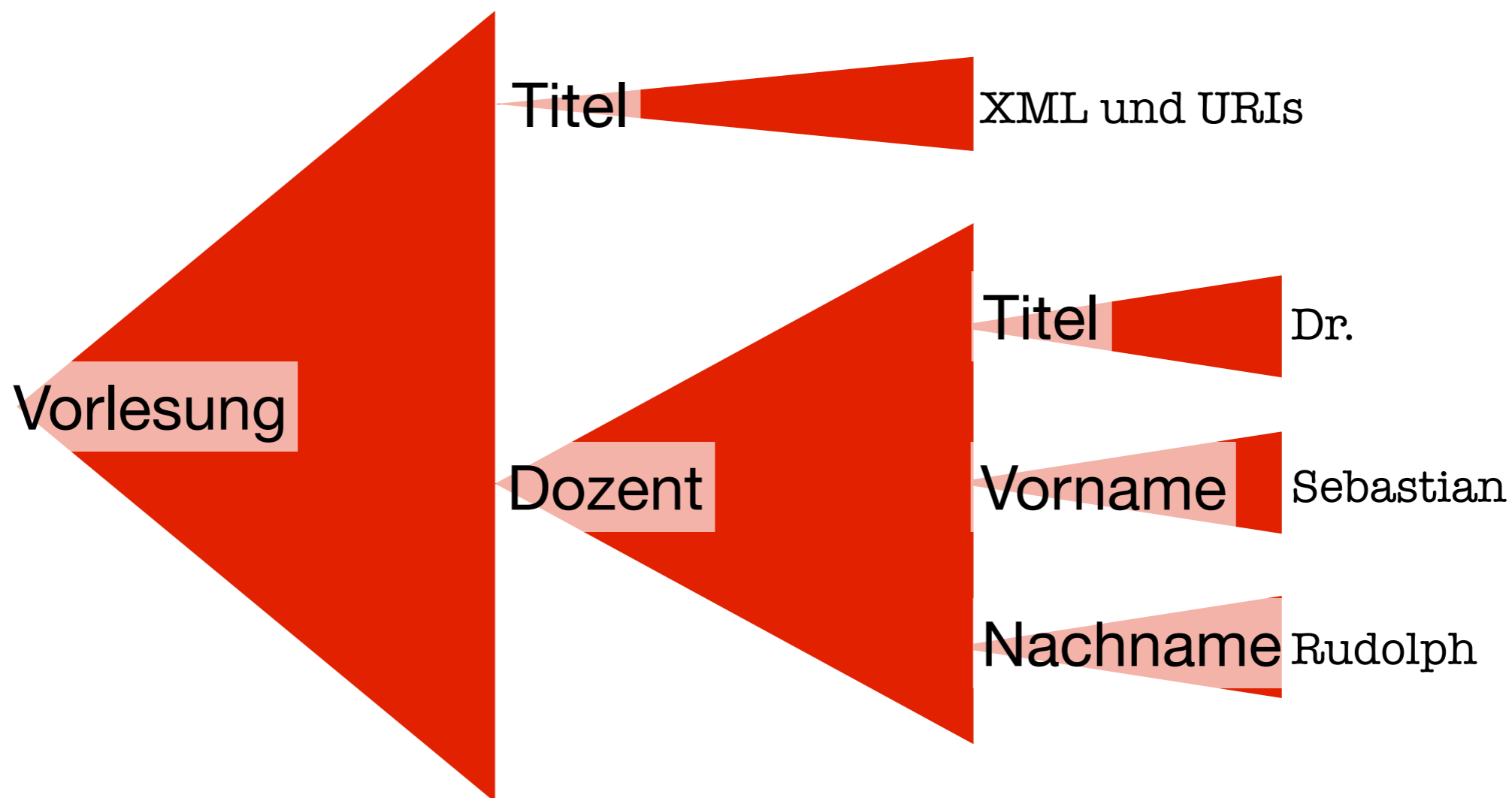


ANNOTATION MIT MARKUPSPRACHEN



ANNOTATION MIT MARKUPSPRACHEN

Baumstruktur



AGENDA



- XML - Motivation/Idee
- XML - Syntax
- URIs
- Namensräume

XML



- eXtensible Markup Language
- Ursprung: strukturierter Text ($\text{HTML4.0} \in \text{XML} \subset \text{SGML}$)
- Web-Standard (W3C) zum Datenaustausch:
 - Ein- und Ausgabedaten von Anwendungen können mittels XML beschrieben werden
 - Industrie muss sich nur noch auf standardisierte Beschreibung (= Vokabular) einigen
- Komplementärsprache zu HTML:
 - HTML beschreibt die Präsentation
 - XML beschreibt den Inhalt

XML-SYNTAX (I) PRÄAMBEL



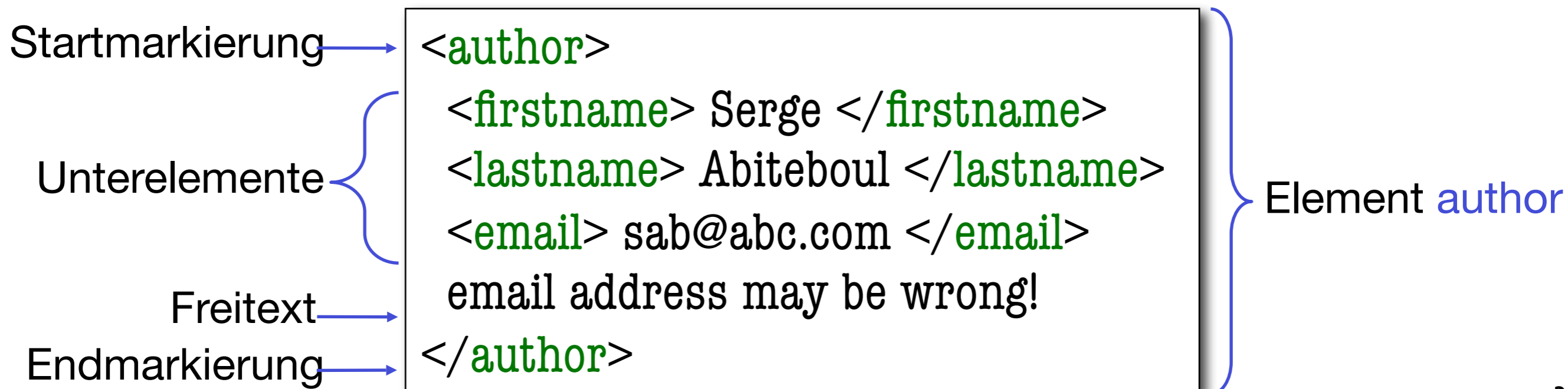
- XML-Dokument ist Textdokument
- beginnt mit Deklaration, die Versionsnummer des verwendeten Standards und optional die Zeichenkodierung enthält, z.B.:

```
<?xml version="1.0" encoding="utf-8"?>
```

XML-SYNTAX (2) – XML-ELEMENT



- XML-Element (engl. element):
 - Beschreibung eines Objekts, die durch passende Markierungen (tags) geklammert ist
 - Inhalt eines Elements: Text und/oder weitere Elemente (beliebige Schachtelung möglich)
 - Leere Elemente: `<year></year>` kurz: `<year/>`
 - "äußerstes" Element heißt Wurzelement (nur eines pro XML-Dokument)




XML-SYNTAX (3) – XML-ATTRIBUTE

AIFB 

- XML-Attribut (engl. attribute):
 - Name-Zeichenkettenwert-Paar in Start- oder selbstschließendem Tag
 - Assoziiert mit einem Element
 - Alternative Möglichkeit, Daten zu beschreiben

Attribut **email**




```
<author email="sab@abc.com">  
  <firstname> Serge </firstname>  
  <lastname> Abiteboul </lastname>  
</author>
```

XML-SYNTAX (3) – XML-ATTRIBUT

AIFB 

- XML-Attribut (engl. attribute):
 - Name-Zeichenkettenwert-Paar in Start- oder selbstschließendem Tag
 - Assoziiert mit einem Element
 - Alternative Möglichkeit, Daten zu beschreiben

Attribut **email**



```
<author email="sab@abc.com">  
  <firstname> Serge </firstname>  
  <lastname> Abiteboul </lastname>  
</author>
```

Weitere denkbare Beschreibung derselben Daten:

```
<author firstname="Serge" lastname="Abiteboul" email="sab@abc.com"/>
```

XML vs. HTML



- HTML: festes Vokabular (Menge von tags) und Semantik (die Darstellung von Text)
- XML: freie Bezeichner zur Beschreibung von anwendungsspezifischer Syntax und Semantik
- XML_CSGML

```

<h1> Bib </h1>
<p>
  <i> Foundations of Databases </i>
  Serge Abiteboul
  <br> Addison Wesley, 1997
<p>
  ...
  
```

HTML

```

<Bib id="01">
  <paper id="012">
    <title> Foundations of Databases </title>
    <author>
      <firstname> Serge </firstname>
      <lastname> Abiteboul </lastname>
    </author>
    <year> 1997 </year>
    <publisher> Addison Wesley </publisher>
  </paper>
  ...
</Bib>
  
```

XML

AGENDA



- XML - Motivation/Idee
- XML - Syntax
- URIs
- Namensräume

URIs - IDEE



- URI = Uniform Resource Identifier
- dienen zur weltweit eindeutigen Bezeichnung von Ressourcen
- Ressource kann jedes Objekt sein, was (im Kontext der gegebenen Anwendung) eine klare Identität besitzt (z.B. Bücher, Orte, Menschen, Verlage, Beziehungen zwischen diesen Dingen, abstrakte Konzepte usw.)
- in bestimmten Domänen ähnliches bereits realisiert: ISBN für Bücher

URIs - SYNTAX



- Erweiterung des URL-Konzeptes; nicht jede URI bezeichnet aber ein Webdokument (umgekehrt wird als URI für Webdokumente häufig deren URL verwendet)
- Beginnt mit dem sogenannten URI-Schema das durch ":" vom nachfolgenden Teil getrennt ist (z.B.: http, ftp, mailto)
- häufig hierarchisch aufgebaut

SELBSTDEFINIERTER URIs



- nötig, wenn für eine Ressource (noch) keine URI existiert (bzw. bekannt ist)
- Strategie zur Vermeidung von (ungewollten) Überschneidungen:
Nutzung von `http`-URIs einer eigenen Webseite
- ermöglicht auch Ablegen einer Dokumentation zur URI an dieser Stelle

BESCHREIBENDES VS. BESCHRIEBENES



- Trennung von URI für Ressource und deren Dokumentation durch URI-Referenzen (durch "#" angehängte Fragmente) oder content negotiation
- z.B.: als URI für Shakespeares "Othello"
<http://de.wikipedia.org/wiki/Othello>
nicht geeignet, besser
<http://de.wikipedia.org/wiki/Othello#URI>

AGENDA



- XML - Motivation/Idee
- XML - Syntax
- URIs
- Namensräume

XML-NAMENSRAÜME: MOTIVATION



- XML-Dokumente besitzen Element- und Attributnamen (“Markup Vocabulary”) mit allgemeiner Gültigkeit
- Eine XML-Anwendung basiert auf allgemeiner Interpretation dieser Namen
- Ein XML-Dokument soll Markup-Vokabular aus mehreren ‘Dictionaries’ enthalten können. (Erinnerung: XML-Dokument muss keine DTD haben.)
- Namespaces zur Vermeidung von Namenskonflikten.

XML-NAMENSRAÜME



- XML Namespaces sind ähnlich zu Modul-Konzepten in Programmiersprachen
- Disambiguierung von Tag-Namen durch Verwendung unterschiedlicher “Prefixe”
- Ein Prefix wird vom lokalen Namen separiert durch ein “:”, so entstehen prefix:name Tags
- Namespace-Bindungen werden von manchen Werkzeugen ignoriert, sog. “flache Namespaces”

NAMENSRAUM-BINDUNGEN



- Prefixe werden belegt mit Namespace URIs, indem ein Attribut `xmlns:prefix` bei dem relevanten Element oder einem seiner Vorgängerelemente eingefügt wird: `prefix:name1, ..., prefix:namen`
- Der Wert des `xmlns:prefix`-Attributes ist eine URI, welche (für XML Schemata) auf eine Beschreibung auf eine Beschreibung der Namespace Syntax verweisen kann aber nicht muss
- Ein Element kann Bindings nutzen für mehrere (unterschiedliche) Namespaces durch Verwendung separater Attribute `xmlns:prefix1, ..., xmlns:prefixm`

BEISPIEL: OHNE NAMENS RÄUME



```
<Vorlesung>
  <Titel>
    XML und URIs
  </Titel>
  <Dozent>
    <Titel>
      Dr.
    </Titel>
    <Vorname>
      Sebastian
    </Vorname>
    <Nachname>
      Rudolph
    </Nachname>
  </Dozent>
</Vorlesung>
```

Titel ist
mehrdeutig
verwendeter
TagName

ZWEI VERSCHIEDENE NAMENSRÄUME



```
<Vorlesung xmlns:lv="http://www.semantic-web-Grundlagen/Lehrveranstaltungen"
           xmlns:person="http://www.semantic-web-Grundlagen/Person" >
  <lv:Titel>
    XML und URIs
  </lv:Titel>
  <lv:Dozent>
    <person:Titel>
      Dr.
    </person:Titel>
    <person:Vorname>
      Sebastian
    </person:Vorname>
    <person:Nachname>
      Rudolph
    </person:Nachname>
  </lv:Dozent>
</lv:Vorlesung>
```

Titel wurde
disambiguiert
durch
Verwendung der
Prefixe lv und
person

EINFÜHRUNG IN RDF

Dr. Sebastian Rudolph

Einleitung und Ausblick

XML und URIs

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

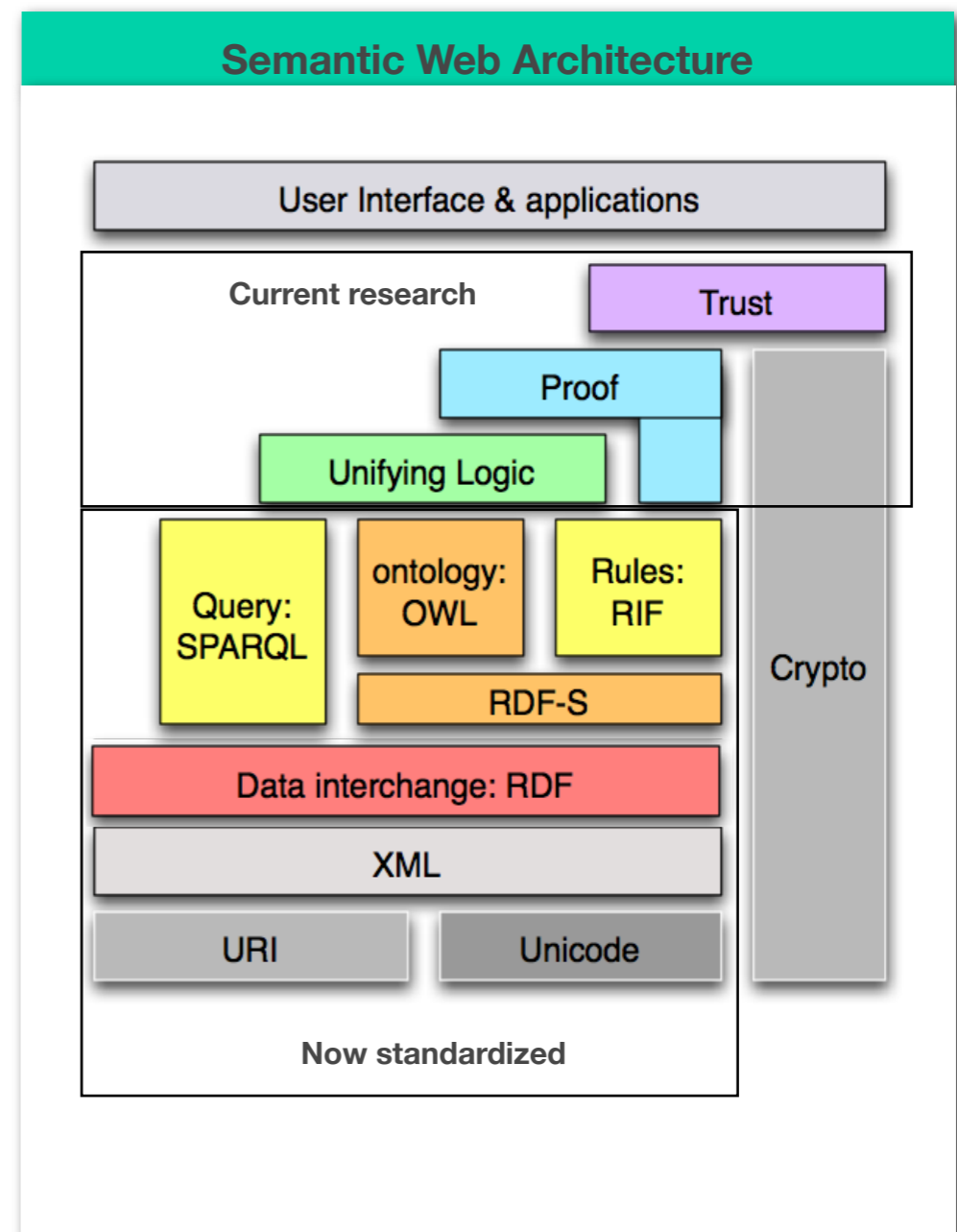
OWL - Semantik und Reasoning

SPARQL - Syntax und Intuition

Semantik von SPARQL und konjunktive Anfragen

OWL 1.1 - Syntax und Semantik

Semantic Web und Regeln



EINFÜHRUNG IN RDF

Dr. Sebastian Rudolph

Einleitung und Ausblick

XML und URIs

Einführung in RDF

RDF Schema

Logik - Grundlagen

Semantik von RDF(S)

OWL - Syntax und Intuition

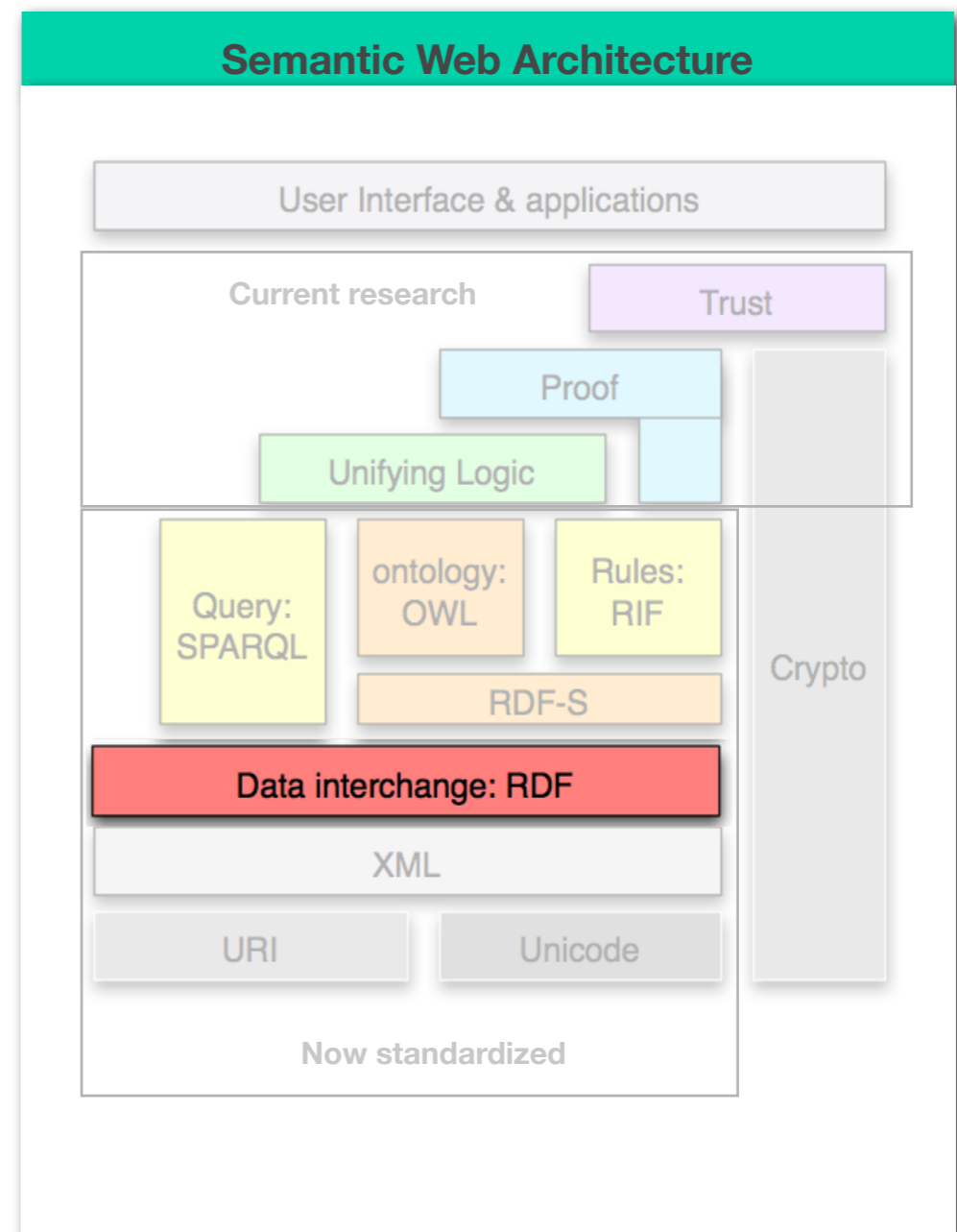
OWL - Semantik und Reasoning

SPARQL - Syntax und Intuition

Semantik von SPARQL und konjunktive Anfragen

OWL 1.1 - Syntax und Semantik

Semantic Web und Regeln



AGENDA



- Motivation
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- Datentypen
- mehrwertige Beziehungen
- leere Knoten
- Listen

AGENDA



- **Motivation**
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- Datentypen
- mehrwertige Beziehungen
- leere Knoten
- Listen

UNZULÄNGLICHKEITEN VON XML



- Tag-Namen ambig (durch Namespaces und URIs behebbbar)
- Baumstruktur nicht optimal für
 - intuitive Beschreibung der Daten
 - Informationsintegration
- Beispiel: wie kodiert man in einem Baum den Fakt:
"Das Buch 'Semantic Web - Grundlagen' wird beim Springer-Verlag verlegt"?

MODELLIERUNGSPROBLEME IN XML

AIFB 

- "Das Buch 'Semantic Web - Grundlagen' wird beim Springer-Verlag verlegt"

```
<Verlegt>  
<Verlag>Springer-Verlag</Verlag>  
<Buch>Semantic Web - Grundlagen</Buch>  
</Verlegt>
```

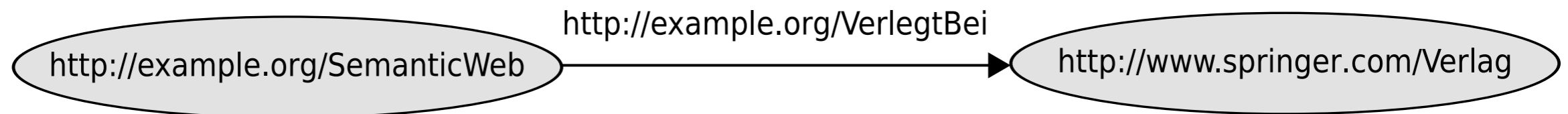
```
<Verlag Name="Springer-Verlag">  
<Verlegt Buch="Semantic Web - Grundlagen/">  
</Verlag>
```

```
<Buch Name="Semantic Web - Grundlagen">  
<Verleger Verlag="Springer-Verlag">  
</Buch>
```

etc.

RDF: GRAPHEN STATT BÄUME

- Lösung: Darstellung durch (gerichtete Graphen)



AGENDA



- Motivation
- **RDF-Datenmodell**
- Syntax für RDF: Turtle und XML
- Datentypen
- mehrwertige Beziehungen
- leere Knoten
- Listen

ALLGEMEINES ZU RDF



- “Resource Description Framework”
- W3C Recommendation
(<http://www.w3.org/RDF>)
- RDF ist ein Datenmodell
 - ursprünglich: zur Angabe von Metadaten für Web-Ressourcen, später allgemeiner
 - kodiert strukturierte Informationen
 - universelles, maschinenlesbares Austauschformat

BESTANDTEILE VON RDF-GRAPHEN

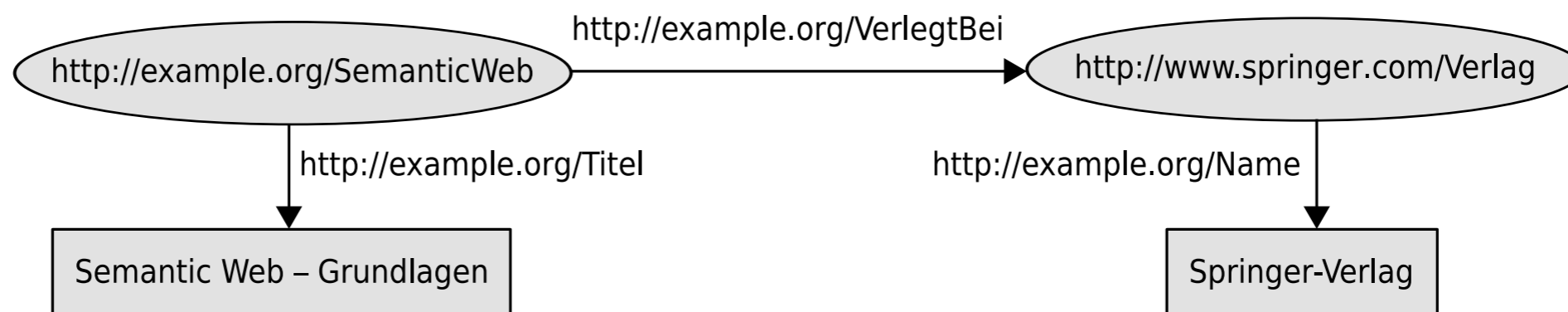


- URIs
 - zur eindeutigen Referenzierung von Ressourcen
 - bereits im Rahmen von XML behandelt
- Literale
 - beschreiben Datenwerte, denen keine separate Existenz zukommt
- leere Knoten
 - erlauben Existenzaussagen über ein Individuum mit gewissen Eigenschaften, ohne es zu benennen

LITERALE



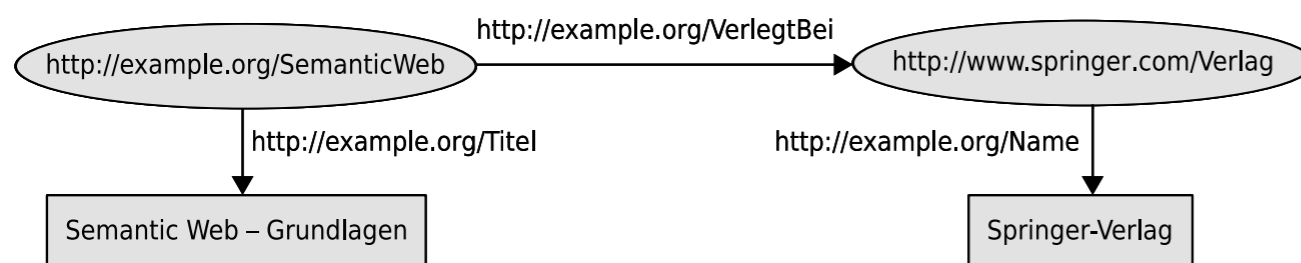
- zur Repräsentation von Datenwerten
- Darstellung als Zeichenketten
- Interpretation erfolgt durch *Datentyp*
- Literale ohne Datentyp werden wie Zeichenketten behandelt



GRAPH ALS MENGE VON TRIPELN



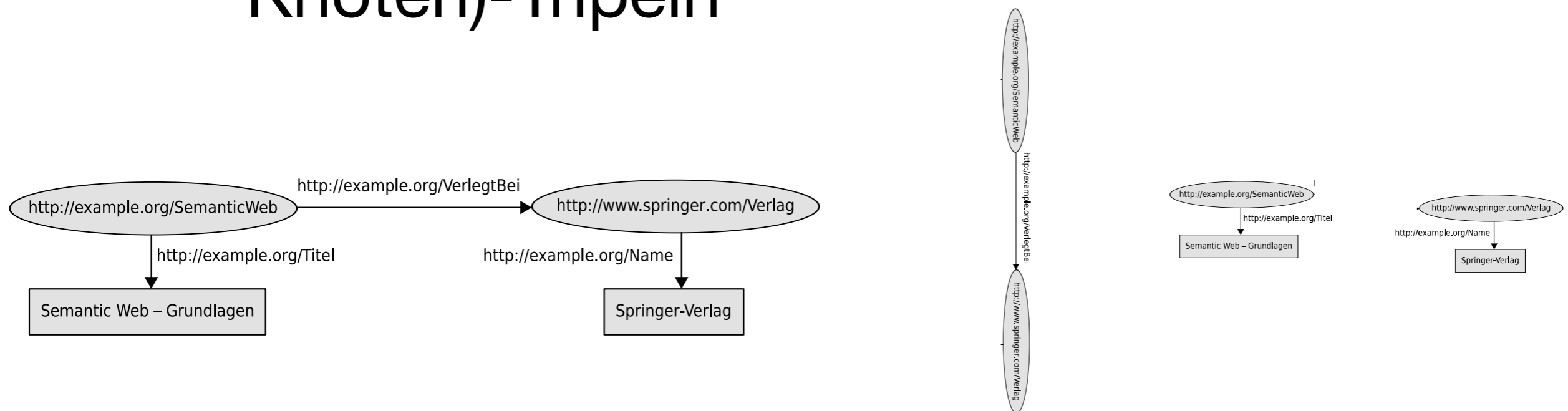
- verschiedene Darstellungsmöglichkeiten für Graphen
- hier verwendet: Liste von (Knoten-Kante-Knoten)-Tripeln



GRAPH ALS MENGE VON TRIPELN



- verschiedene Darstellungsmöglichkeiten für Graphen
- hier verwendet: Liste von (Knoten-Kante-Knoten)-Tripeln



RDF-TRIPEL



- Bestandteile eines RDF-Tripels



- angelehnt an linguistische Kategorien, aber nicht immer stimmig
- erlaubte Belegungen:
Subjekt : URI oder leerer Knoten
Prädikat: URI (auch Propertys genannt)
Objekt : URI oder leerer Knoten oder Literal
- Knoten- und Kantenbezeichner eindeutig, daher ursprünglicher Graph aus Tripel-Liste rekonstruierbar

AGENDA



- Motivation
- RDF-Datenmodell
- **Syntax für RDF: Turtle und XML**
- Datentypen
- mehrwertige Beziehungen
- leere Knoten
- Listen

EINFACHE SYNTAX FÜR RDF



- direkte Auflistung der Tripel:
 - N3: "Notation 3" - umfangreicher Formalismus
 - N-Triples: Teil von N3
 - Turtle: Erweiterung von N-Triples (Abkürzungen)
- Syntax in Turtle:
 - URIs in spitzen Klammern
 - Literale in Anführungszeichen
 - Tripel durch Punkt abgeschlossen
 - Leerzeichen und Zeilenumbrüche außerhalb von Bezeichnern werden ignoriert

TURTLE SYNTAX: ABKÜRZUNGEN



- **Beispiel**

```
<http://example.org/SemanticWeb>
  <http://example.org/VerlegtBei> <http://springer.com/Verlag> .
<http://example.org/SemanticWeb>
  <http://example.org/Titel> "Semantic Web - Grundlagen" .
<http://springer.com/Verlag>
  <http://example.org/Name> "Springer-Verlag" .
```

- auch in Turtle können Abkürzungen für Präfixe festgelegt werden:

```
@prefix ex: <http://example.org/> .
@prefix springer: <http://springer.com/> .

ex:SemanticWeb    ex:VerlegtBei    springer:Verlag .
ex:SemanticWeb    ex:Titel        "Semantic Web - Grundlagen" .
springer:Verlag   ex:Name        "Springer-Verlag" .
```

TURTLE SYNTAX: ABKÜRZUNGEN



- mehrere Tripel mit gleichem Subjekt kann man zusammenfassen:

```
@prefix ex: <http://example.org/> .
```

```
@prefix springer: <http://springer.com/> .
```

```
ex:SemanticWeb    ex:VerlegtBei    springer:Verlag ;
```

```
                  ex:Titel        "Semantic Web - Grundlagen" .
```

```
springer:Verlag  ex:Name          "Springer-Verlag", "Springer" .
```

- ebenso Tripel mit gleichem Subjekt und Prädikat:

```
@prefix ex: <http://example.org/> .
```

```
ex:SemanticWeb ex:Autor ex:Hitzler, ex:Krötzsch, ex:Rudolph, ex:Sure ;
```

```
                  ex:Titel "Semantic Web - Grundlagen" .
```

XML-SYNTAX VON RDF



- Turtle intuitiv gut lesbar und maschinenverarbeitbar
- aber: bessere Tool-Unterstützung und Programmbibliotheken für XML
- daher: XML-Syntax am verbreitetsten

XML-SYNTAX VON RDF



- wie in XML werden Namensräume eingesetzt, um Tagnamen zu disambiguieren
- RDF-eigene tags haben einen festgelegten Namensraum, der Bezeichner ist standardmäßig 'rdf'

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ex="http://example.org/">

  <rdf:Description rdf:about="http://example.org/SemanticWeb">
    <ex:VerlegtBei>
      <rdf:Description rdf:about="http://springer.com/Verlag">
        </rdf:Description>
      </ex:VerlegtBei>
    </rdf:Description>

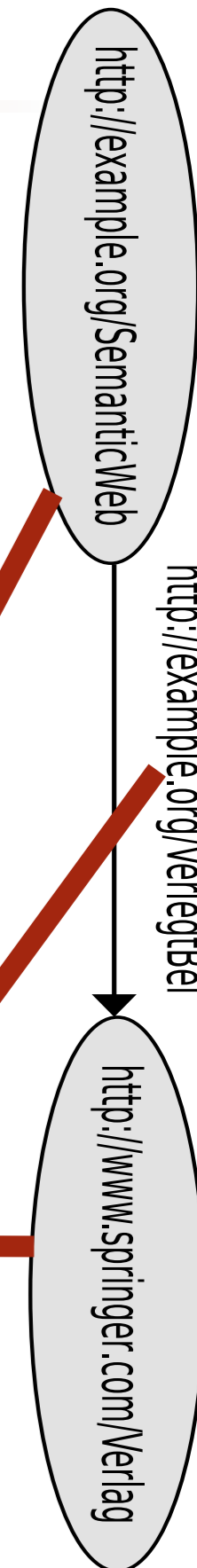
  </rdf:RDF>
```

XML-SYNTAX VON RDF



- Das `rdf:Description`-Element kodiert das Subjekt (dessen URI wird als Wert des zugehörigen `rdf:about`-Attributs angegeben).
- Jedes geschachtelt im `rdf:Description`-Element enthaltene Element steht für ein Prädikat (dessen URI ist der Elementname), das wiederum das Tripel-Objekt als `rdf:Description`-Element enthält.

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">  
  <ex:VerlegtBei>  
    <rdf:Description rdf:about="http://springer.com/Verlag">  
      </rdf:Description>  
    </ex:VerlegtBei>  
</rdf:Description>
```

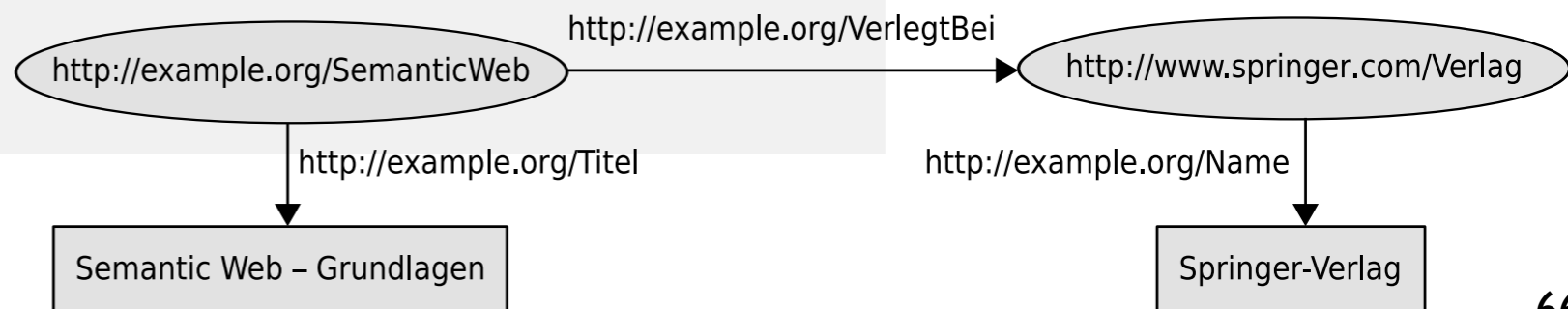


XML-SYNTAX VON RDF



- ungetypte Literale können als Freitext in das Prädikatelement eingeschlossen werden
- Verkürzte Darstellung erlaubt:
 - ein Subjekt enthält mehrere Property-Elemente
 - eine Objekt-Description dient als Subjekt für ein weiteres Tripel

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">  
  <ex:Titel>Semantic Web - Grundlagen</ex:Titel>  
  <ex:VerlegtBei>  
    <rdf:Description rdf:about="http://springer.com/Verlag">  
      <ex:Name>Springer-Verlag</ex:Name>  
    </rdf:Description>  
  </ex:VerlegtBei>  
</rdf:Description>
```

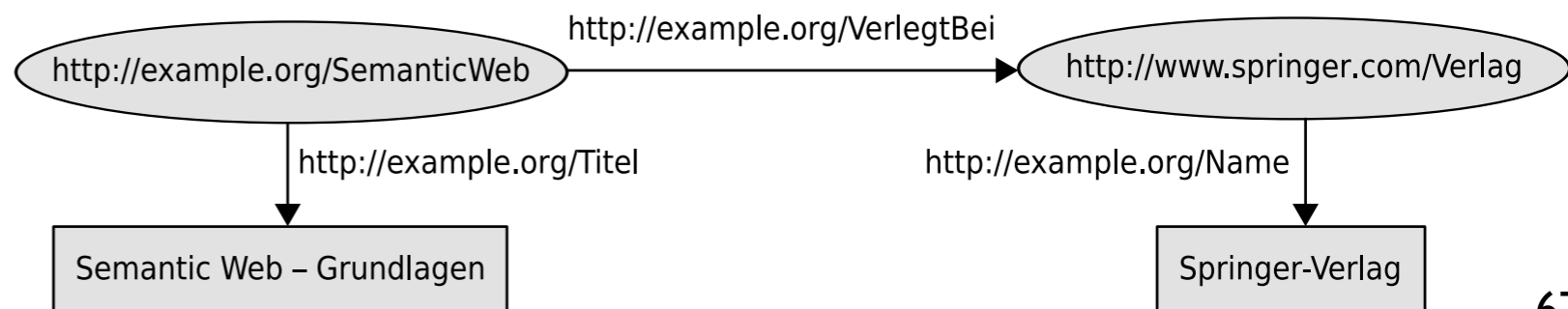


XML-SYNTAX VON RDF



- Alternative (aber semantisch gleichwertige) Darstellung für Literale als XML-Attribute
- Attributnamen sind dann die Property-URIs
- Angabe von Objekt-URIs als Wert des `rdf:resource`-Attributs innerhalb eines Property-Tags

```
<rdf:Description rdf:about="http://example.org/SemanticWeb"  
    ex:Titel= "Semantic Web - Grundlagen">  
  <ex:VerlegtBei rdf:resource="http://springer.com/Verlag" />  
</rdf:Description>  
<rdf:Description rdf:about="http://springer.com/Verlag"  
    ex:Name="Springer-Verlag" />
```



RDF/XML-SYNTAX: KOMPLIKATIONEN



- Namensräume sind essentiell (nicht nur Abkürzung), da in XML-Elementen und -Attributen keine Doppelpunkte zulässig, die keine Namensräume kodieren
- Problem: in XML keine Namensräume in Attributwerten möglich (würde im Sinne eines URI-Schemas interpretiert), also z.B. verboten:

```
rdf:about="ex:SemanticWeb"
```

- "Workaround" via XML-Entitäten:

Deklaration:

```
<!ENTITY ex 'http://example.org/'>
```

Verwendung:

```
rdf:resource="&ex;SemanticWeb"
```

RDF/XML-SYNTAX: BASIS-URIs



- Arbeit mit Basis-URIs:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xml:base="http://www.example.org/" >

  <rdf:Description rdf:about="SemanticWeb">
    <ex:VerlegtBei rdf:resource="http://springer.com/Verlag" />
  </rdf:Description>

</rdf:RDF>
```

- Erkennung relativer URIs an Abwesenheit eines Schemateils

AGENDA



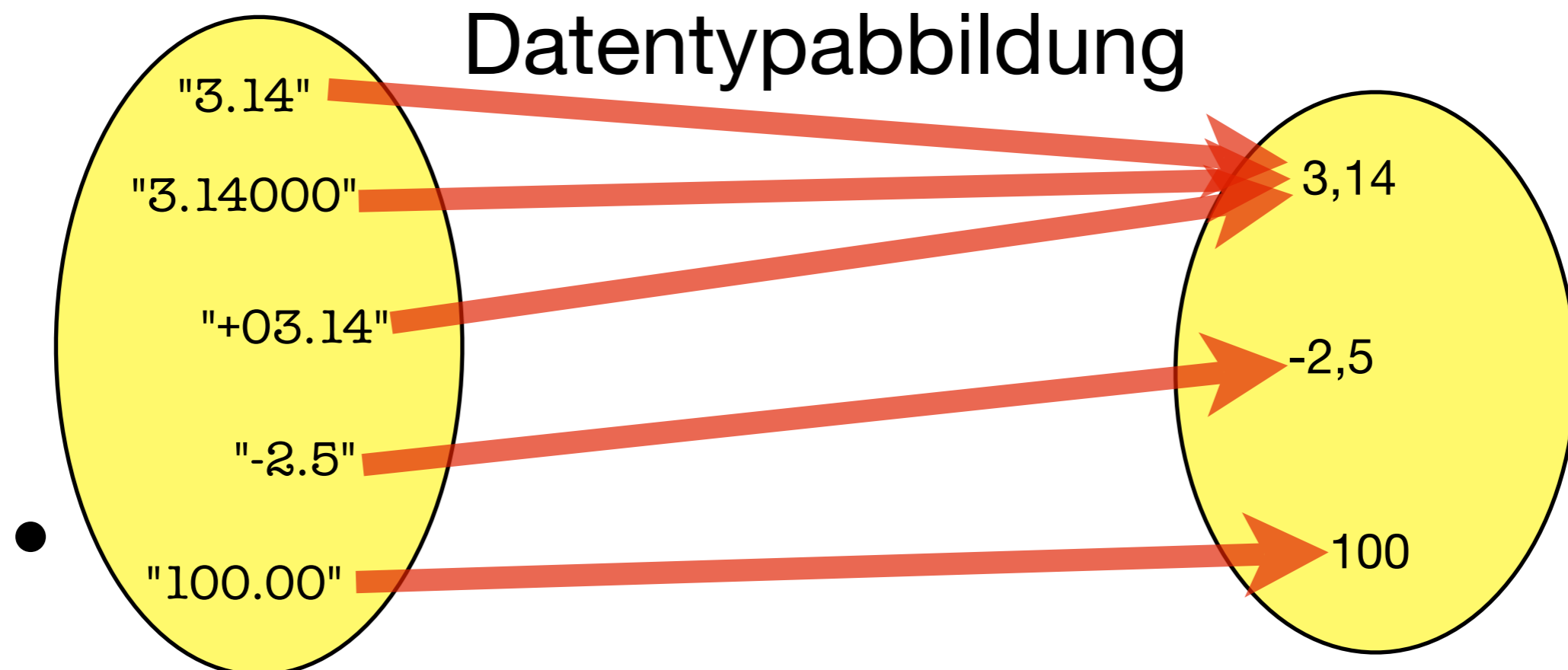
- Motivation
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- **Datentypen**
- mehrwertige Beziehungen
- leere Knoten
- Listen

DATENTYPEN - ABSTRAKT

- Beispiel: xsd:decimal

lexikalischer Bereich

Wertebereich



- bzgl. xsd:decimal gilt "3.14" = "+03.14"
bzgl. xsd:string nicht!

DATENTYPEN IN RDF

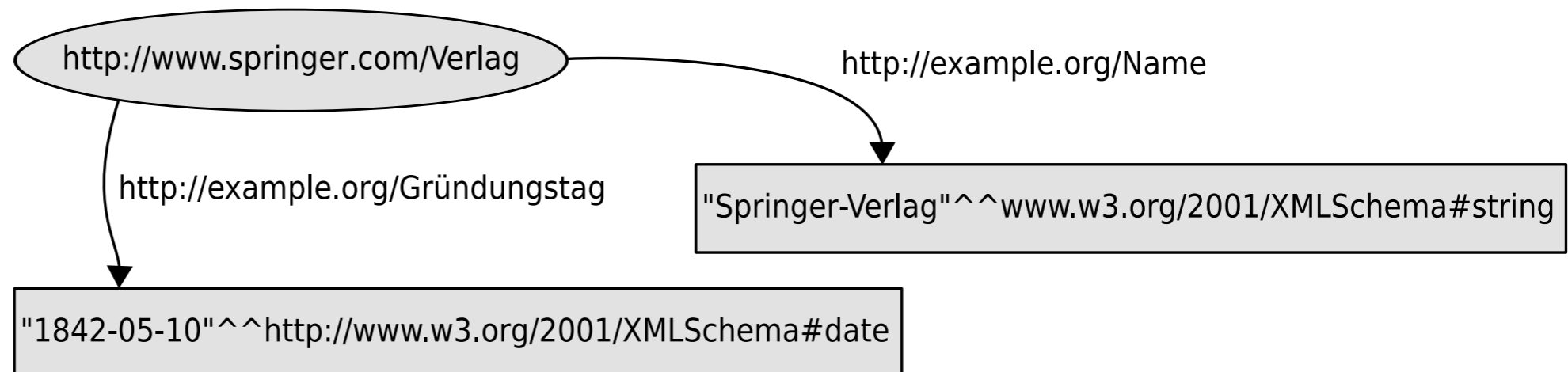


- Bisher: Literale ungetypt, wie Zeichenketten behandelt (also z.B.: "02" < "100" < "11" < "2")
- Typung erlaubt besseren (semantischen = bedeutungsgemäßen) Umgang mit Werten
- Datentypen werden durch URIs identifiziert und sind im Prinzip frei wählbar
- häufig: Verwendung von xsd-Datentypen
- Syntax:
"Datenwert"^^Datentyp-URI

DATENTYPEN IN RDF - BEISPIEL



- Graph:



- Turtle:

```

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://springer.com/Verlag>
  <http://example.org/Name> "Springer-Verlag"^^xsd:string ;
  <http://example.org/Gründungstag> "1842-05-10"^^xsd:date .
  
```

- XML:

```

<rdf:Description rdf:about="http://springer.com/Verlag">
  <ex:Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Springer-Verlag
  </ex:Name>
  <ex:Gründungstag
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    1842-05-10
  </ex:Gründungstag>
</rdf:Description>
  
```


DER VORDEFINIERTER DATENTYP



- `rdf:XMLLiteral` ist einziger vordefinierter Datentyp in RDF
- bezeichnet beliebige (balancierte) XML-Fragmente
- in RDF/XML besondere Syntax zur eindeutigen Darstellung:

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:Titel rdf:parseType="Literal">
    <b>Semantic Web</b>
    <br />
    Grundlagen
  </ex:Titel>
</rdf:Description>
```

SPRACHANGABEN UND DATENTYPEN



- Sprachinformationen beeinflussen nur ungetypte Literale
- Beispiel:
 - XML

```
<rdf:Description rdf:about="http://springer.com/Verlag">  
  <ex:Name xml:lang="de">Springer-Verlag</ex:Name>  
  <ex:Name xml:lang="en">Springer Science+Business Media</ex:Name>  
</rdf:Description>
```

- Turtle

```
<http://springer.com/Verlag> <http://example.org/Name>  
  "Springer-Verlag"@de, "Springer Science+Business Media"@en .
```

SPRACHANGABEN UND DATENTYPEN



- nach RDF-Spezifikation sind demnach die folgenden Literale unterschiedlich:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
<http://springer.com/Verlag>      <http://example.org/Name>  
    "Springer-Verlag",  
    "Springer-Verlag"@de,  
    "Springer-Verlag"^^xsd:string .
```

- ...werden aber häufig (intuitionsgemäß) als gleich implementiert.

AGENDA



- Motivation
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- Datentypen
- **mehrwertige Beziehungen**
- leere Knoten
- Listen

MEHRWERTIGE BEZIEHUNGEN



- Kochen mit RDF:
"Für die Zubereitung von Chutney benötigt man 450g grüne Mango, einen Teelöffel Cayennepfeffer, ..."

- erster Modellierungsversuch:

```
@prefix ex: <http://example.org/> .  
ex:Chutney ex:hatZutat "450g grüne Mango", "1TL Cayennepfeffer"
```

- nicht zufriedenstellend: Zutaten samt Menge als Zeichenkette. Suche nach Rezepten, die grüne Mango beinhalten, so nicht möglich.

MEHRWERTIGE BEZIEHUNGEN



- Kochen mit RDF:
"Für die Zubereitung von Chutney benötigt man 450g grüne Mango, einen Teelöffel Cayennepfeffer, ..."
- zweiter Modellierungsversuch:


```
@prefix ex: <http://example.org/> .
ex:Chutney ex:Zutat ex:grüneMango; ex:Menge "450g" ;
ex:Zutat ex:Cayennepfeffer; ex:Menge "1TL" .
```
- überhaupt nicht zufriedenstellend: keine eindeutige Zuordnung von konkreter Zutat und Menge mehr möglich.

MEHRWERTIGE BEZIEHUNGEN

- Problem: es handelt sich um eine echte dreiwertige (auch: ternäre) Beziehung (s. z.B. Datenbanken)

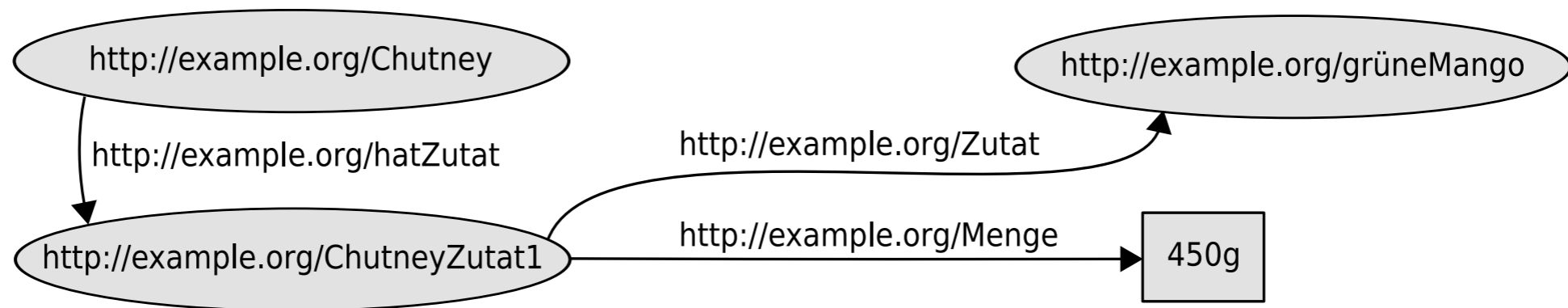
Gericht	Zutat	Menge
Chutney	grüne Mango	450g
Chutney	Cayennepfeffer	1 TL

- direkte Darstellung in RDF nicht möglich
- Lösung: Einführung von Hilfsknoten

MEHRWERTIGE BEZIEHUNGEN

- Hilfsknoten in RDF:

- als Graph



- Turtle-Syntax (mit Verwendung von `rdf:value`)

```

@prefix ex:    <http://example.org/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Chutney    ex:hatZutat    ex:ChutneyZutat1 .
ex:ChutneyZutat1  rdf:value    ex:grüneMango;
ex:Menge       "450g" .
  
```


AGENDA

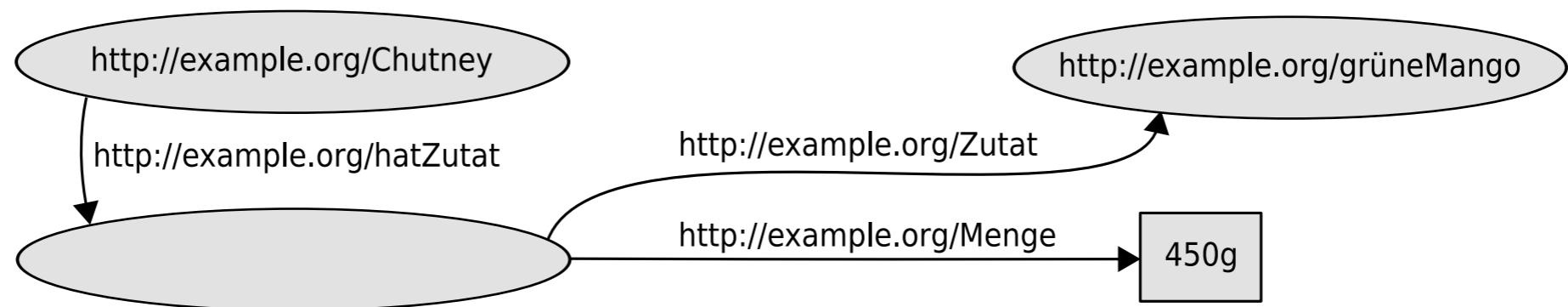


- Motivation
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- Datentypen
- mehrwertige Beziehungen
- **leere Knoten**
- Listen

LEERE KNOTEN



- leere Knoten (blank nodes, bnodes) können für Ressourcen verwendet werden, die nicht benannt werden müssen (z.B. Hilfsknoten)
- können als Existenzaussagen gelesen werden
- Syntax:
 - als Graph



LEERE KNOTEN



- Syntax:
 - RDF/XML-Syntax

```
<rdf:Description rdf:about="http://example.org/Chutney">
  <ex:hatZutat rdf:nodeID="id1" />
</rdf:Description>
<rdf:Description rdf:nodeID="id1">
  <ex:Zutat rdf:resource="http://example.org/grüneMango" />
  <ex:Menge>450g</ex:Menge>
</rdf:Description>
```

- verkürzt

```
<rdf:Description rdf:about="http://example.org/Chutney">
  <ex:hatZutat rdf:parseType="Resource">
    <ex:Zutat rdf:resource="http://example.org/grüneMango" />
    <ex:Menge>450g</ex:Menge>
  </ex:hatZutat>
</rdf:Description>
```

LEERE KNOTEN



- Syntax:

- Turtle

```
@prefix ex: <http://example.org/> .  
ex:Chutney    ex:hatZutat    _:id1 .  
_:id1        ex:Zutat      ex:grüneMango;  ex:Menge    "450g" .
```

- verkürzt

```
@prefix ex: <http://example.org/> .  
ex:Chutney    ex:hatZutat  
              [ ex:Zutat      ex:grüneMango;  ex:Menge    "450g" ] .
```

AGENDA



- Motivation
- RDF-Datenmodell
- Syntax für RDF: Turtle und XML
- Datentypen
- mehrwertige Beziehungen
- leere Knoten
- **Listen**

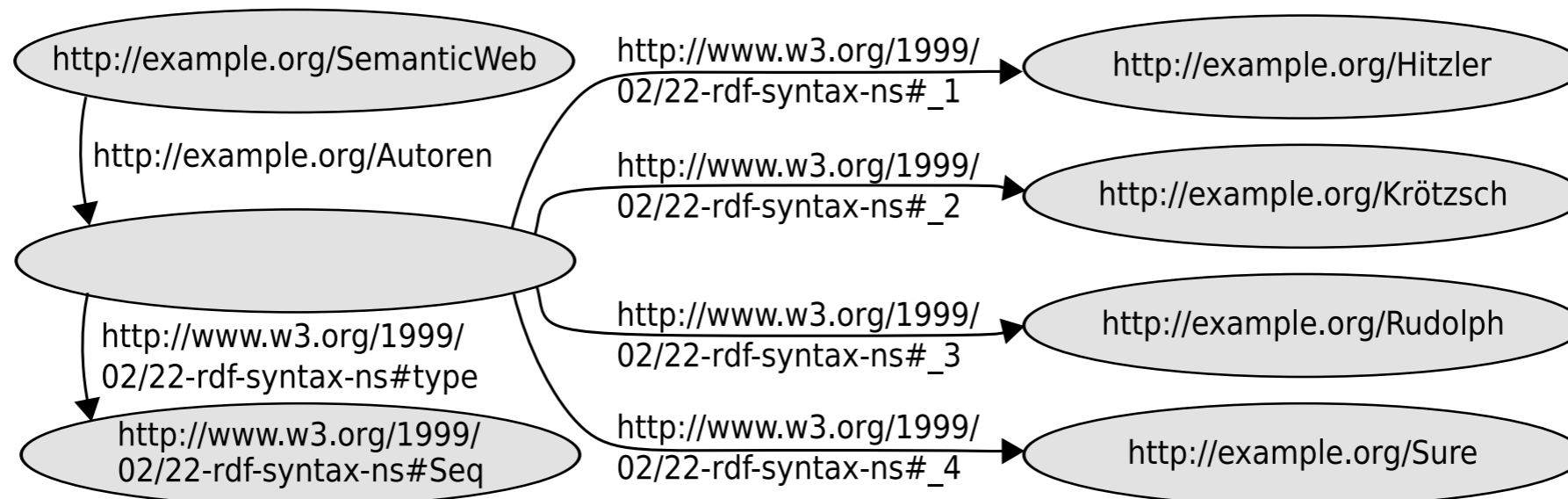
LISTEN

- allgemeine Datenstrukturen zur Aufzählung von beliebig vielen Ressourcen (Reihenfolge relevant), z.B. Autoren eines Buches
- Unterscheidung zwischen
 - offenen Listen (Container)
Hinzufügen von neuen Einträgen möglich
 - geschlossenen Listen (Collections)
Hinzufügen von neuen Einträgen nicht möglich
- Können auch mit bereits vorgestellten Ausdrucksmitteln modelliert werden, also keine zusätzliche Ausdrucksstärke!

OFFENE LISTEN (CONTAINER)



- Graph:



- verkürzt in RDF/XML:

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:Autoren>
    <rdf:Seq>
      <rdf:li rdf:resource="http://www.example.org/Hitzler" />
      <rdf:li rdf:resource="http://www.example.org/Kröttsch" />
      <rdf:li rdf:resource="http://www.example.org/Rudolph" />
      <rdf:li rdf:resource="http://www.example.org/Sure" />
    </rdf:Seq>
  </ex:Autoren>
</rdf:Description>
```

TYPEN OFFENER LISTEN

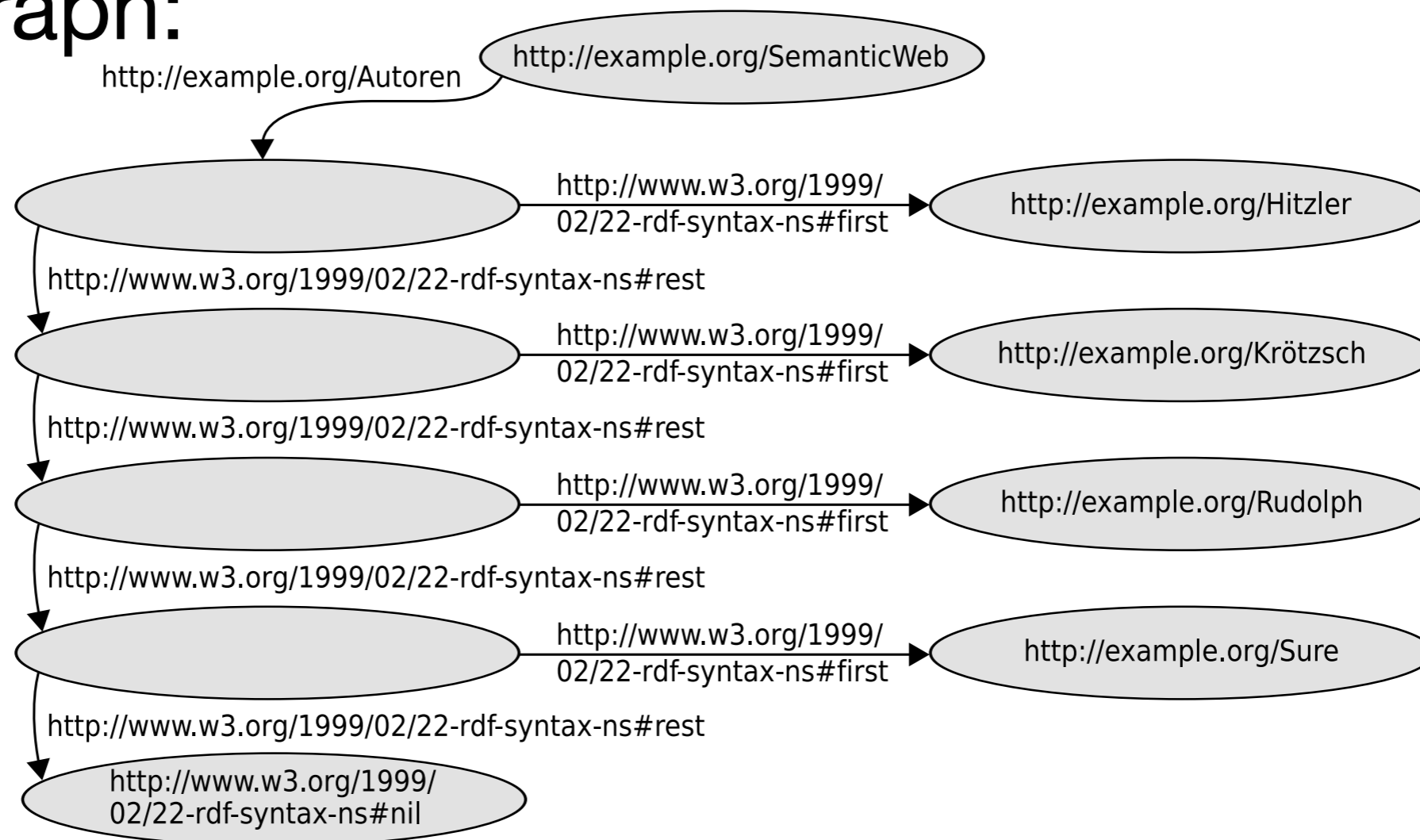


- via `rdf:type` wird dem Listen-Wurzelknoten ein Listentyp zugewiesen:
 - `rdf:Seq`
Interpretation als geordnete Liste (Sequenz)
 - `rdf:Bag`
Interpretation als ungeordnete Menge
in RDF kodierte Reihenfolge nicht von Belang
 - `rdf:Alt`
Menge alternativer Möglichkeiten
im Regelfall immer nur ein Listeneintrag relevant

GESCHLOSSENE LISTEN (COLLECTIONS)



- Graph:



- Idee: rekursive Zerlegung der Liste in Kopfelement und (möglicherweise leere) Restliste.

GESCHLOSSENE LISTEN (COLLECTIONS)



- RDF/XML-Syntax

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:Autoren rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.example.org/Hitzler">
    <rdf:Description rdf:about="http://www.example.org/Kröttsch" />
    <rdf:Description rdf:about="http://www.example.org/Rudolph" />
    <rdf:Description rdf:about="http://www.example.org/Sure" />
  </ex:Autoren>
</rdf:Description>
```

- Turtle

```
@prefix ex: <http://example.org/> .
ex:SemanticWeb    ex:Autor
                  ( ex:Hitzler    ex:Kröttsch    ex:Rudolph    ex:Sure ) .
```

VERBREITUNGSGRAD VON RDF



- heute existiert Vielzahl von RDF-Tools
- Programmier-Bibliotheken für praktisch jede Programmiersprache
- frei verfügbare Systeme zum Umgang mit großen RDF-Datenmengen (sogenannte RDF Stores oder Triple Stores)
- auch kommerzielle Anbieter (z.B. Oracle) unterstützen zunehmend RDF
- Grundlage für Datenformate: RSS 1.0, XMP (Adobe), SVG (Vektorgrafikformat)

BEWERTUNG VON RDF

AIFB 

- weitläufig unterstützter Standard für Speicherung und Austausch von Daten
- ermöglicht weitgehend syntaxunabhängige Darstellung verteilter Informationen in graphbasiertem Datenmodell
- reines RDF sehr "individuenorientiert"
- kaum Möglichkeiten zur Kodierung von Schemawissen
 - RDF Schema (nächste Vorlesung)