UNIVERSITÄT MANNHEIM

Semantic Web Technologies
Resource Description Framework (RDF)

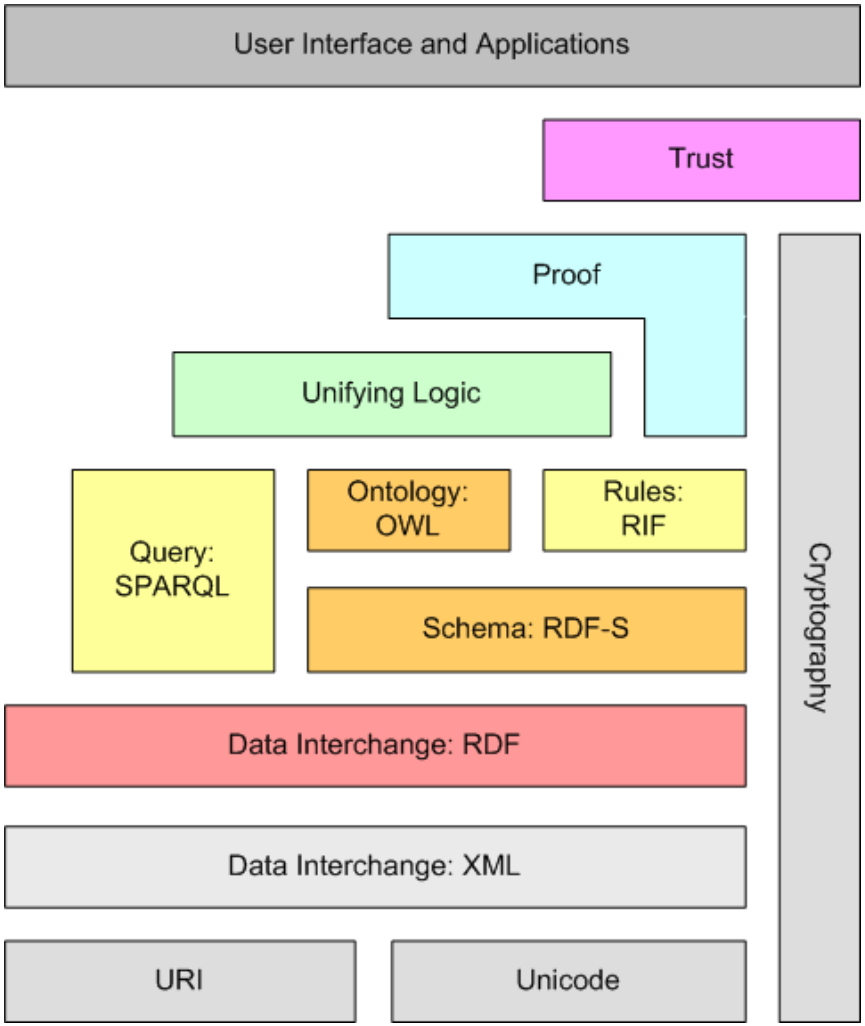
Heiko Paulheim

Semantic Web – Architecture

here be dragons...

Semantic Web Technologies (This lecture)

Technical Foundations



Berners-Lee (2009): *Semantic Web and Linked Data*
http://www.w3.org/2009/Talks/0120-campus-party-tbl/

Overview

- A brief history of RDF
- Encodings of RDF
- Semantics and principles of RDF
- Embedding RDF in HTML – RDFa, Microdata, Microformats
- RDF Tools
- Examples for RDF in the wild

History: Metadata on the Web

- Goal: more effective rating and ranking of web contents, e.g., by search engines

- Who has created this page?
- When has it been changed the last time?
- What is its topic?
- Which is the content's license?
- How does it relate to other pages?

Metadata on the Web: Dublin Core

- Developed in 1995 at a workshpo in Dublin, Ohio
- 15 predefined tags
- A widely accepted standard (ISO 15836:2009)

- May be embedded into HTML:

```
<html>
  <head profile="http://dublincore.org/documents/2008/08/04/dc-html/">
    <title>Semantic Web</title>
    <link rel="schema.DC"      href="http://purl.org/dc/elements/1.1/" >
    <meta name="DC.publisher"  content="University of Mannheim" />
    <meta name="DC.subject"    content="Semantic Web" />
    <meta name="DC.creator"    content="Heiko Paulheim" />
    <meta name="DC.relation"   content="http://www.w3.org/2001/sw/" />
    ...
  </head>
  <body>
  ...
  </body>
</html>
```

Metadata on the Web: Dublin Core

- Identifier
- Format
- Type
- Language
- Date

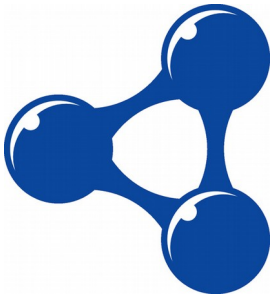
- Title
- Subject
- Coverage
- Description

- Creator
- Publisher
- Contributor
- Rights

- Source
- Relation

What is RDF?

- „Resource Description Framework“
- A W3C standard since 2004
- Description of arbitrary things



- View 1: Sentences in the form <subject, predicate, object>
  „Heiko works for University of Mannheim.“

- View 2: Directed graphs with labeled edges

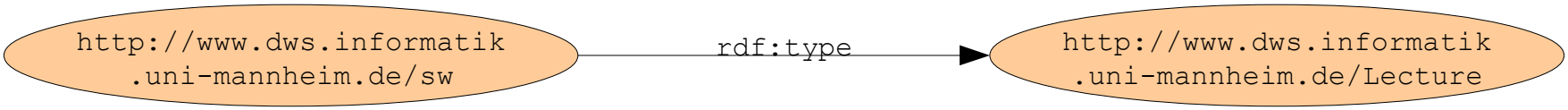
Basic Building Blocks of RDF

- Resources
  - denote things
  - are identified by a URI
  - can have one or multiple types
- Literals
  - are values like strings or integers
  - can only be objects, not subjects or predicates (graph view: they can only have ingoing edges)
  - can have a datatype or a language tag (but not both)
- Properties (Predicates)
  - Link resources to other resources and to literals

Types

- All resources (not literals) can have a type
- Types can be arbitrarily defined
- The predefined predicate `rdf:type*` defines the type of a resource

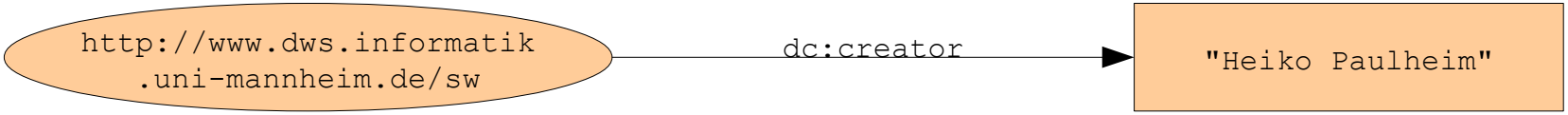
- Semantic Web is a lecture



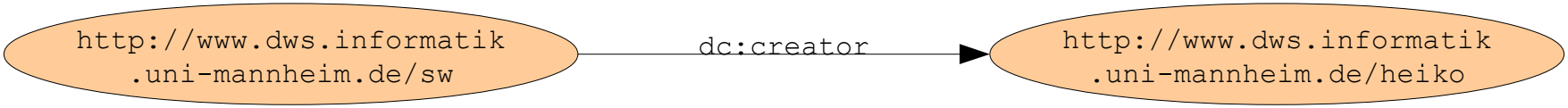
* <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

Resource vs. Literal

- A literal is an atomic value
  - can only be object
  - i.e., a literal terminates always a graph

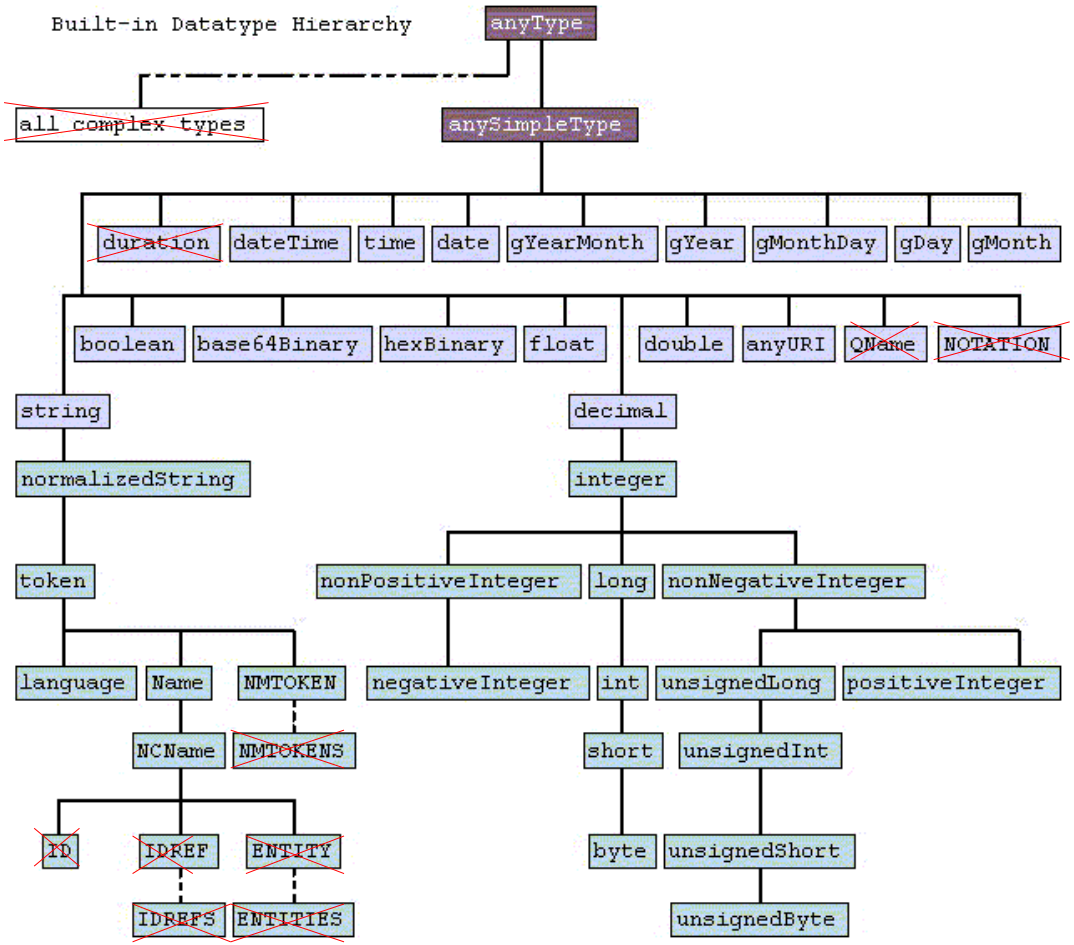


- A resource can be a subject itself



Datatypes for Literals

- (Almost) all XML Schema datatypes may be used
- Exception:
  - XML specific types
  - The underspecified type "duration"
  - sequence types



XML Schema Part 2: Datatypes Second Edition
http://www.w3.org/TR/xmlschema-2/

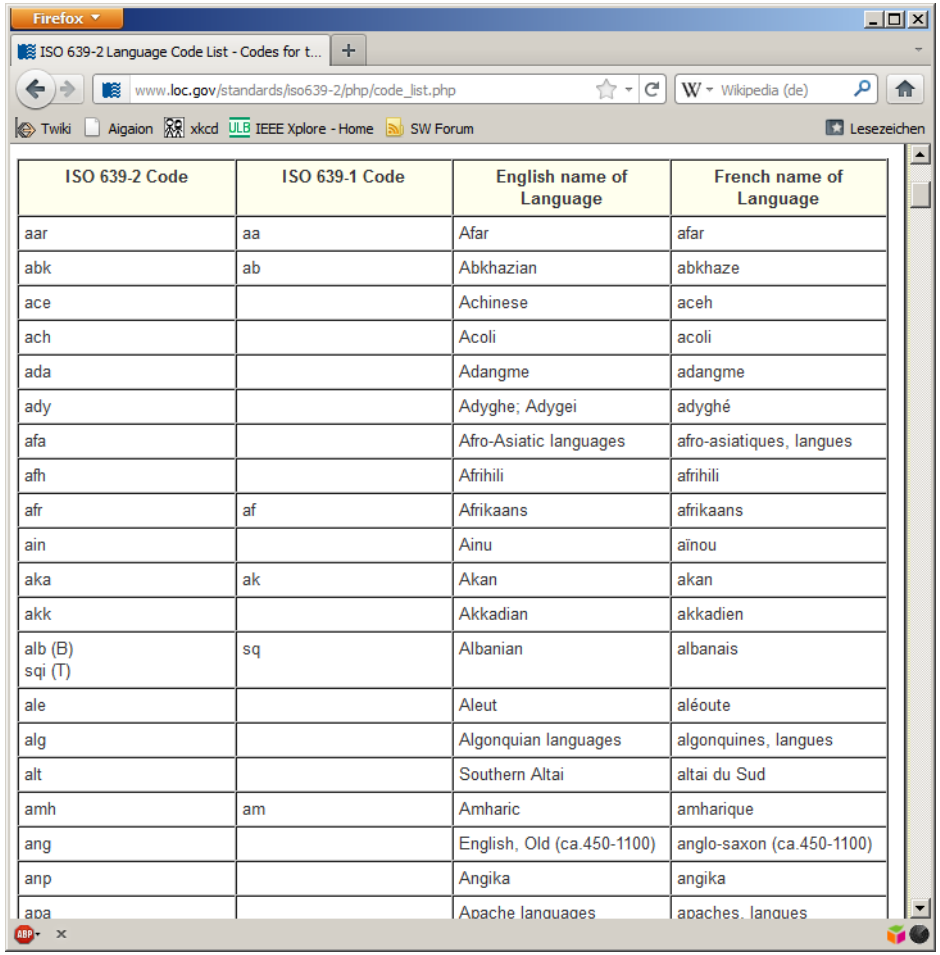
Language Tags for Literals

- Literals may be defined in different natural languages
  - `"München"@de`
  - `"Munich"@en`
- Those can be marked
- Note: the Semantic Web is multilingual!

- Language codes according to ISO 963
  - ISO 963-1 (1963): two-digit codes, 136 languages
  - ISO 963-2 (1998): three-digit Codes, 464 languages
  - if both are defined, ISO 963-1 has to be used!

http://www.loc.gov/standards/iso639-2/php/English_list.php

Language Tags for Literals



ISO 639-2 Code	ISO 639-1 Code	English name of Language	French name of Language
aar	aa	Afar	afar
abk	ab	Abkhazian	abkhaze
ace		Achinese	aceh
ach		Acoli	acoli
ada		Adangme	adangme
ady		Adyghe; Adygei	adyghé
afa		Afro-Asiatic languages	afro-asiatiques, langues
afh		Afrihili	afrihili
afr	af	Afrikaans	afrikaans
ain		Ainu	aïnou
aka	ak	Akan	akan
akk		Akkadian	akkadien
alb (B) sqi (T)	sq	Albanian	albanais
ale		Aleut	aléoute
alg		Algonquian languages	algonquines, langues
alt		Southern Altai	altai du Sud
amh	am	Amharic	amharique
ang		English, Old (ca.450-1100)	anglo-saxon (ca.450-1100)
anp		Angika	angika
apa		Apache languages	apaches, langues

Datatypes in RDF

- Examples:
- :Munich :hasName "München"@de .
  :Munich :hasName "Munich"@en .
  :Munich :hasPopulation "1356594 "^^xsd:integer .
  :Munich :hasFoundingYear "1158-01-01"^^xsd:date .

- Note: there are no default datatypes (not even “string”!)
- These are three different literals:
  - "München"
  - "München"@de
  - "München"^^xsd:string .

Triple Notation

- A W3C standard (2004)
- Triples consist of a subject, predicate, and object
- An RDF document is an *unordered* set of triples

- Simple triple:

```
<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>
<http://purl.org/dc/elements/1.1/relation>
<http://www.w3.org/2001/sw/> .
```

- Literal with language tag:

```
<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>
<http://purl.org/dc/elements/1.1/subject>
"Semantic Web"@en .
```

- Type literal:

```
<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>
<http://www.uni-mannheim.de/mhb/creditpoints>
"6"^^<http://www.w3.org/2001/XMLSchema#integer> .
```

Turtle Notation

- A simplified triple notation

- Central definition of namespaces (and a default namespace):

```
@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix : <http://www.dws.informatik.uni-mannheim.de/teaching/>
:semantic-web dc:subject "Semantic Web"@en .
```

- Triples sharing the same subject or subject+predicate:

```
:semantic-web dc:subject "Semantisches Web"@de ,
                         "Semantic Web"@en ;
              dc:creator "Heiko Paulheim".
```

- Shorthand notation for `rdf:type`:

```
:semantic-web a :lecture .
```

Notation RDF/XML

- A W3C standard since 2004
- Encodes RDF in XML
- Suitable for machine processing (plenty of XML tools!)

- Defining resources:

```
<rdf:Description rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
  <dc:creator>Heiko Paulheim</dc:creator>
</rdf:Description>
```

- Defining typed resources:

```
<rdf:Description rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
  <rdf:type rdf:resource="http://www.uni-mannheim.de/mhb/Lecture"/>
</rdf:Description>
```

- Alternative representation:

```
<mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw"
 xmlns:mhb="http://www.uni-mannheim.de/mhb/" />
```

Notation RDF/XML

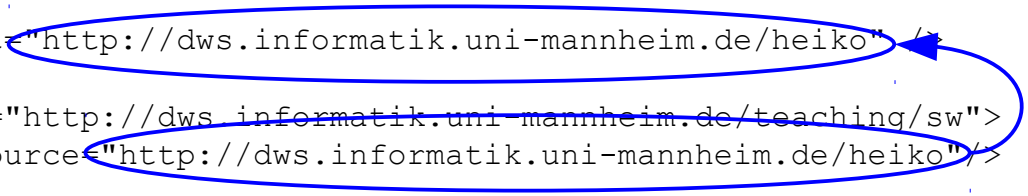
- Relations between resources by nesting

```
<mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
  <mhb:givenBy>
    <mhb:Lecturer rdf:about="http://dws.informatik.uni-mannheim.de/heiko"/>
  </mhb:givenBy>
</mhb:Lecture>
```

- Relations between resources by explicit links

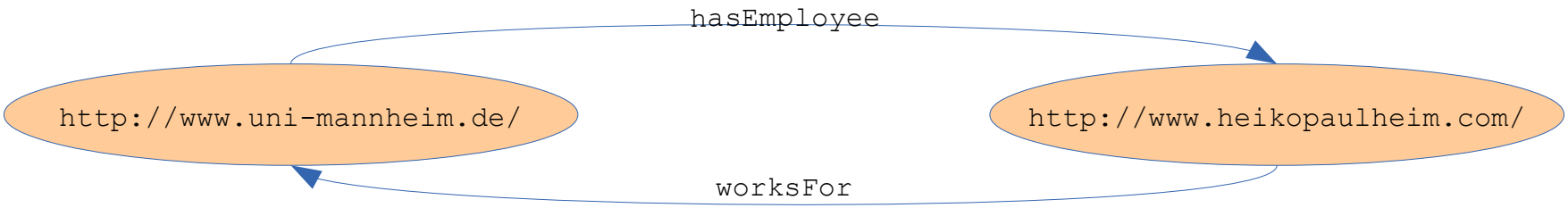
```
<mhb:Lecturer rdf:about="http://dws.informatik.uni-mannheim.de/heiko" />

<mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
  <mhb:givenBy rdf:resource="http://dws.informatik.uni-mannheim.de/heiko"/>
</mhb:Lecture>
```



Notation RDF/XML

- An RDF graph may contain cycles
- XML may not → explicit links are necessary




```
<mhb:University rdf:about="http://www.uni-mannheim.de">
  <mhb:hasEmployee>
    <mhb:UniversityMember rdf:about="http://www.heikopaulheim.com/">
      <mhb:worksFor rdf:resource="http://www.uni-mannheim.de"/>
    </mhb:UniversityMember>
  </mhb:hasEmployee>
</mhb:University>
```

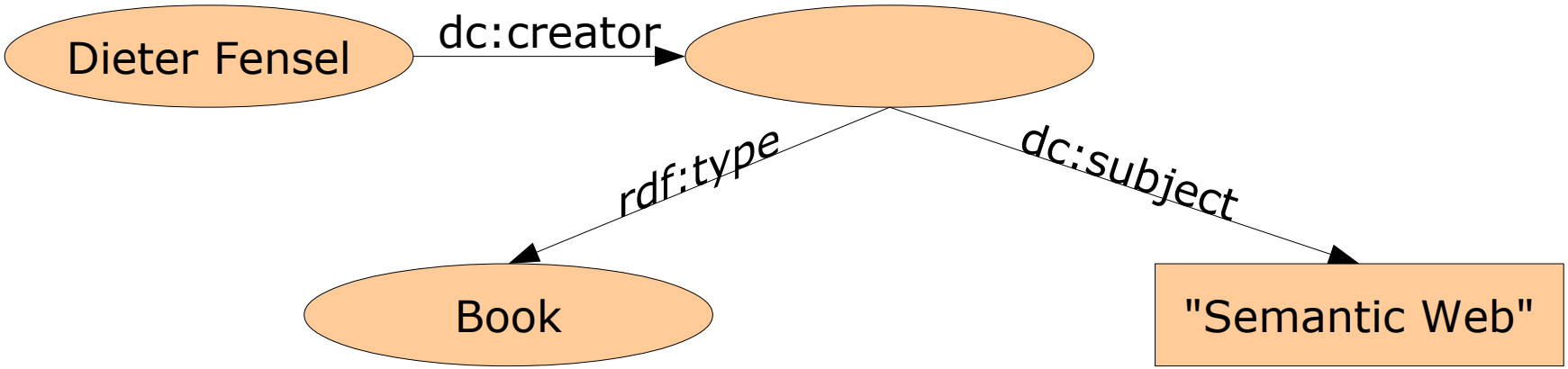
JSON-LD Notation

- JSON is popular in script programming
- JSON-LD: Standard for serializing RDF in JSON

```
{
        "@id": "http://www.heikopaulheim.com/",
        "http://dws.informatik.uni-mannheim.de/name": "Heiko Paulheim",
        "http://dws.informatik.uni-mannheim.de/teaches":
        {
                "http://http://www.w3.org/1999/02/22-rdf-syntax-ns#type":
                        "http://dws.informatik.uni-mannheim.de/Lecture",
                "http://dws.informatik.uni-mannheim.de/title":
                        "Semantic Web"
        }
}
```

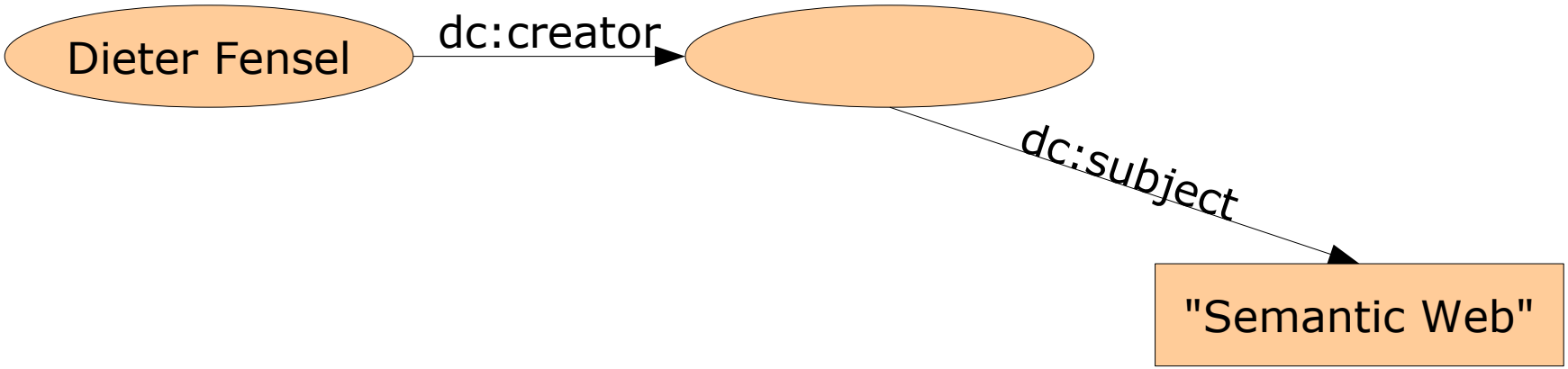
Blank Nodes

- Information that is not or cannot be specified
  - "Dieter Fensel has written a book about the Semantic Web"



Blank Nodes

- Information that is not or cannot be specified
  - "Dieter Fensel has written something about the Semantic Web."



Blank Nodes in Turtle

- Variant 1: explicitly named with an underscore

```
:Dieter_Fensel dc:creator _:x .
_:x a :Book ;
    dc:subject "Semantic Web" .
```

- Variant 2: unnamed with square brackets

```
:Dieter_Fensel dc:creator
  [ a :Book;
    dc:subject "Semantic Web" ].
```

- Notes:
  - both are equivalent
  - changing blank node names does not change the semantics!

Application of Blank Nodes: n-ary Predicates

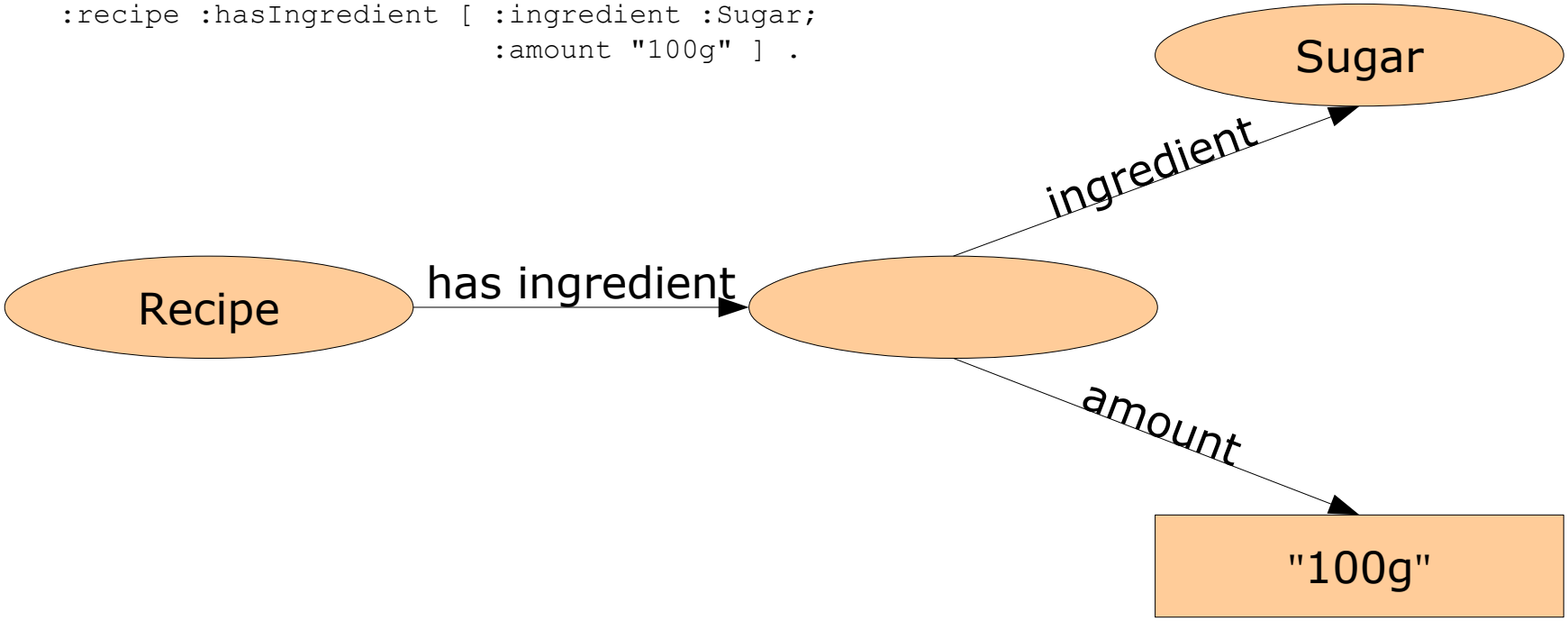
- RDF predicates always connect a subject and an object
  - i.e., in the sense of predicate logic, they are binary predicates

```
  :Heiko :works_for :UniMannheim .
⇔          works_for(Heiko, UniMannheim) .
```

- Sometimes, n-ary predicates are needed
  - has_ingredient(Recipe, Sugar, 100g)

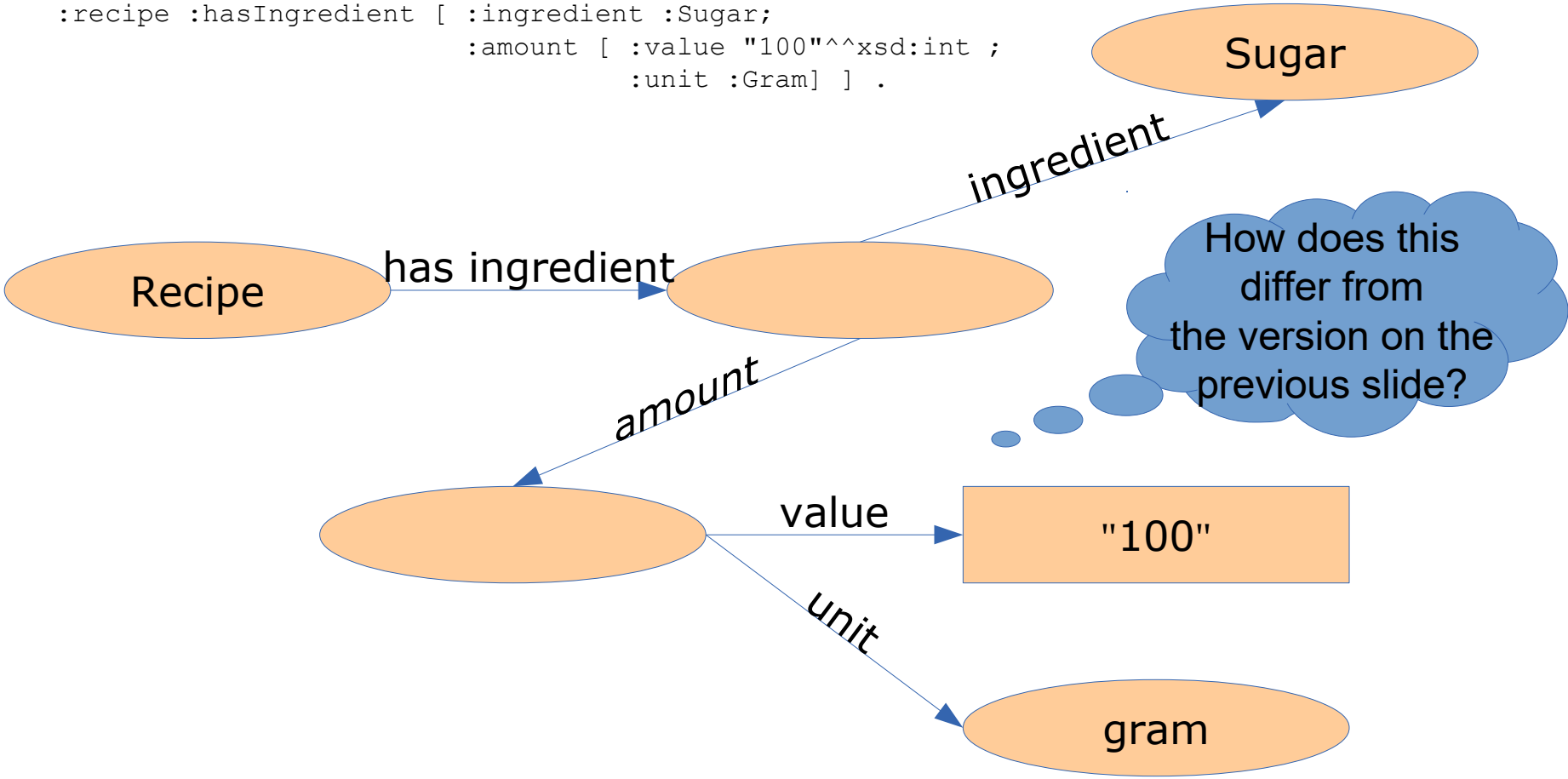
Application of Blank Nodes: n-ary Predicates

```
:recipe :hasIngredient [ :ingredient :Sugar;
                         :amount "100g" ] .
```



Application of Blank Nodes: n-ary Predicates

```
:recipe :hasIngredient [ :ingredient :Sugar;
                         :amount [ :value "100"^^xsd:int ;
                                   :unit :Gram] ] .
```



Semantic Principles of RDF

- On the Web, "anybody can say anything about anything"
  - This is called the AAA principle (Allemang & Hendler)

- This principle also holds for the Semantic Web

Semantic Principles of RDF

- One thing can have multiple names

```
:Munich :capitalOf :Bavaria .
:München :capitalOf :Bayern .
```

- On the semantic web, there is not just one name for each thing
  - this is called the *Non-unique name assumption*

- This means: Just that two things have different names does not mean that they are different!

RDF: Intuition and Actual Semantics

- Let us consider the following example:

```
:Peter :fatherOf :Julia ,
                 :Mary .
```

- How many children does Peter have?

- Intuitively, we assume that Julia and Mary are two different persons
- However, this is not trivial for a machine
  - (and the assumption may even be wrong)

Semantic Principles of RDF

- We (probably) do not know all the contents of the Semantic Web
- Therefore, there may be more information on a resource than what we have

- This principle is called "Open World Assumption"

RDF: Intuition and Actual Semantics

- Let us consider this example again:

```
:Peter :fatherOf :Julia ,
                 :Mary .
```

- How many children does Peter have?

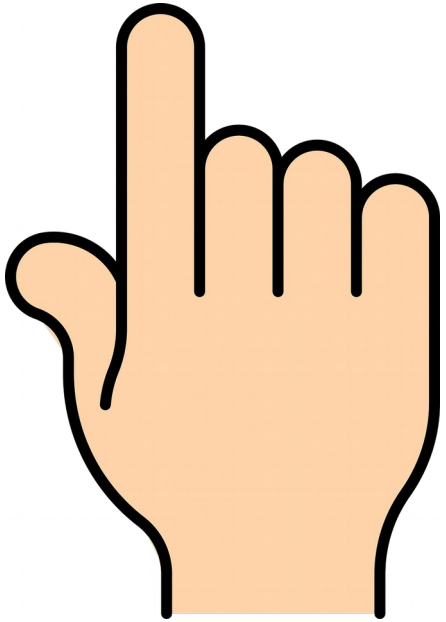
- Intuition says: two children
- However, he could also have three or more
  (oder also just one, as we have learned just a minute ago)

RDF: Intuition and Actual Semantics

- Both
  - Non-unique Name Assumption and
  - Open World Assumption

  will re-occur quite a few times in this lecture

- Hint: consider those two whenever something seems weird when interpreting RDF data



RDF and HTML

- The Semantic Web uses RDF
- The “classic” Web uses HTML

- Does that mean that each information has to be encoded twice?
  - once for humans, once for machines

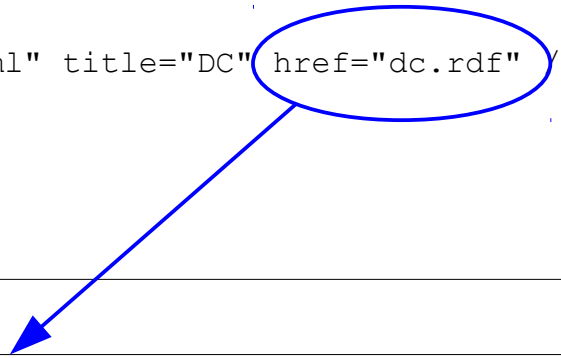
```
<html>
  …
  <b>Dr. Mark Smith</b>
  <i>Physician</i>
  Main St. 14
  Smalltown
  Mon-Fri 9-11 am
  Wed 3-6 pm
  …
</html>
```

```
:p a :Physician .
:p :hasDegree "Dr." .
:p :hasName "Mark Smith" .
:p :hasAddress :a .
:a :street "Main Street" .
:a :number "14"^^xsd:int .
:a :city "Smalltown" .
:p :hasOpeningHours [
  a rdf:Bag ;
  [ :day :Monday;
    :from "9"^^xsd:int;
    :to "11"^^xsd:int;
  ]
  ...
```

Using RDF and HTML Together – Variant 1

- Explicit reference to a RDF version
  - an agent stumbling on the HTML page can download the RDF data file

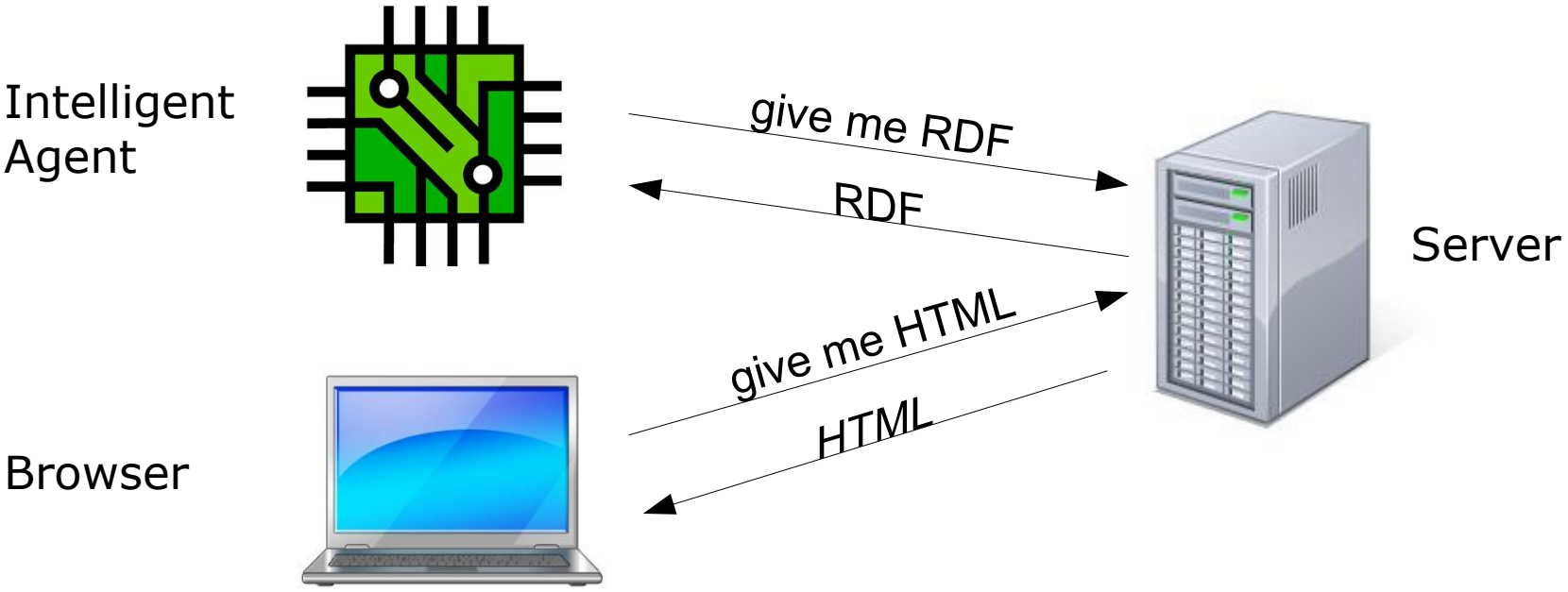
```
<html>
  <head>
    <link rel="meta" type="application/rdf+xml" title="DC" href="dc.rdf" />
  </head>
  <body>
  ...
  </body>
</html>
```



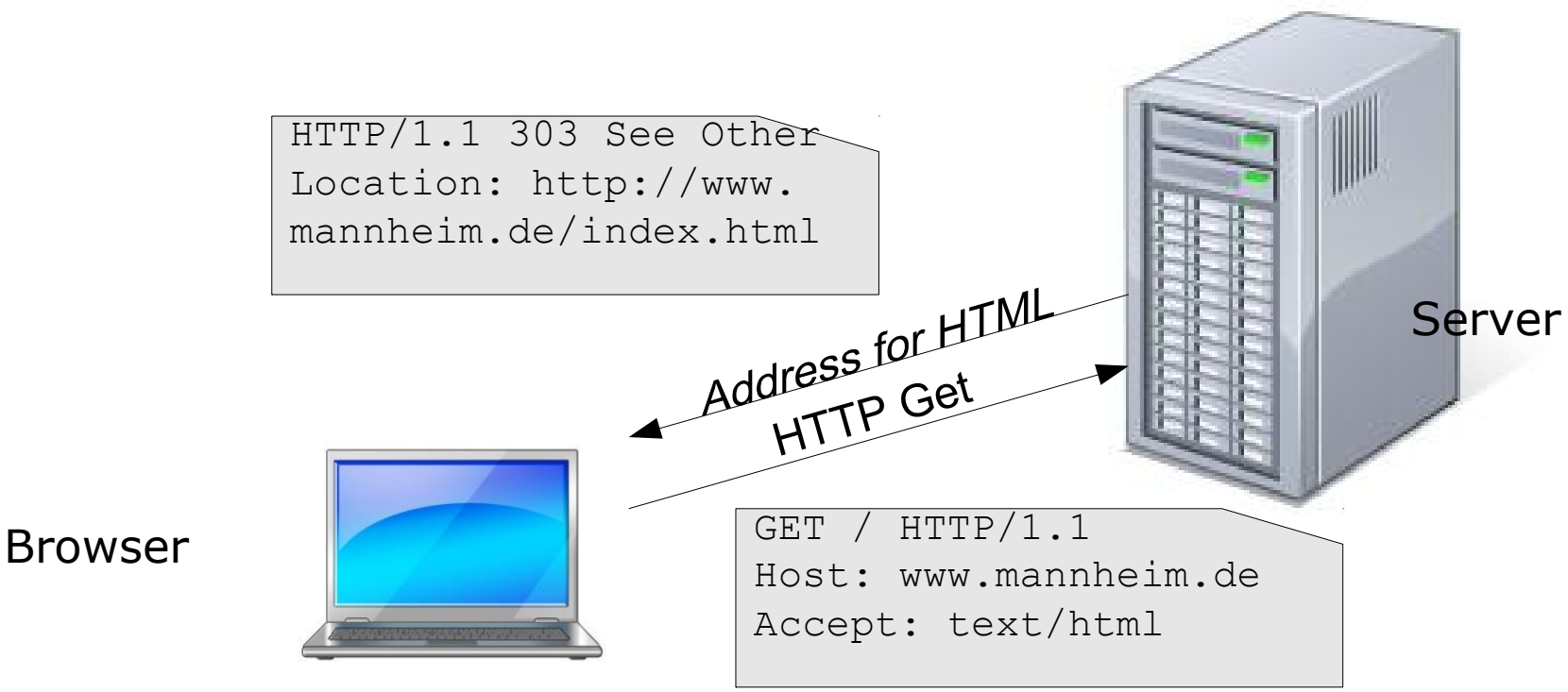
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.dws.informatik.uni-mannheim.de/mhb/sw">
    <dc:publisher>University of Mannheim</dc:publisher>
    <dc:subject>Semantic Web</dc:subject>
    <dc:creator>Heiko Paulheim</dc:creator>
    <dc:relation rdf:resource="http://www.w3.org/2001/sw/" />
  </rdf:Description>
</rdf:RDF>
```

Using RDF and HTML Together – Variant 2

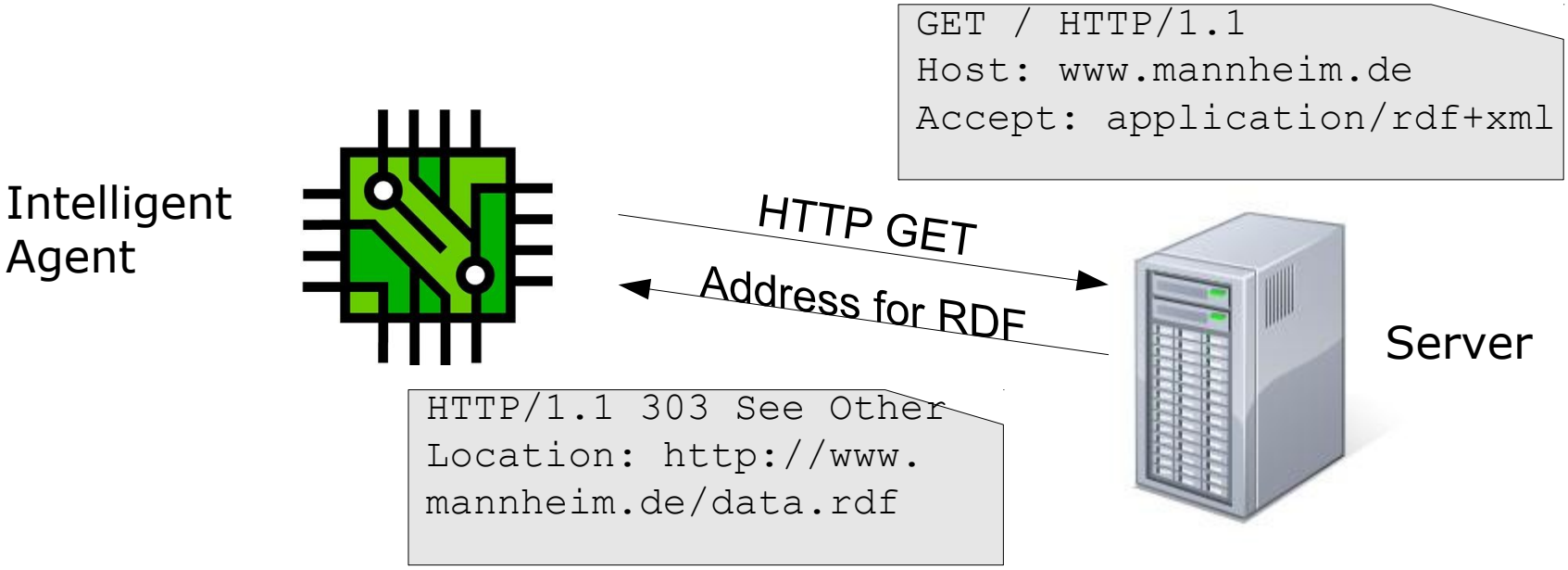
- Content Negotiation



Content Negotiation in Detail



Content Negotiation in Detail



Content Negotiation: MIME Types

- MIME: Multipurpose Internet Mail Extensions
- Original purpose: classifying e-mail attachements
  - Text, PDF, ..
- First version: 1996
- Administrated by IANA

- Important MIME types for the Semantic Web
  - application/rdf+xml
  - text/turtle
  - text/n3
  - application/json
  - application/sparql-query
  - application/sparql-results+xml

Using RDF and HTML Together

- Link to RDF Document
  - Can be done with a simple HTML editor
  - No special server configuration needed
- Content Negotiation
  - Requires particular server setup
  - *One* URI can be used for different representations

- Both cases require
  - two different representatoins
  - “double bookkeeping”

  → Potential source of inconsistencies!

RDF in Attributes (RDFa)

- Idea of RDFa
  - Why not encode HTML and RDF in *one* document
  - The essential information only has to be encoded once

- RDFa combines XHTML with RDF

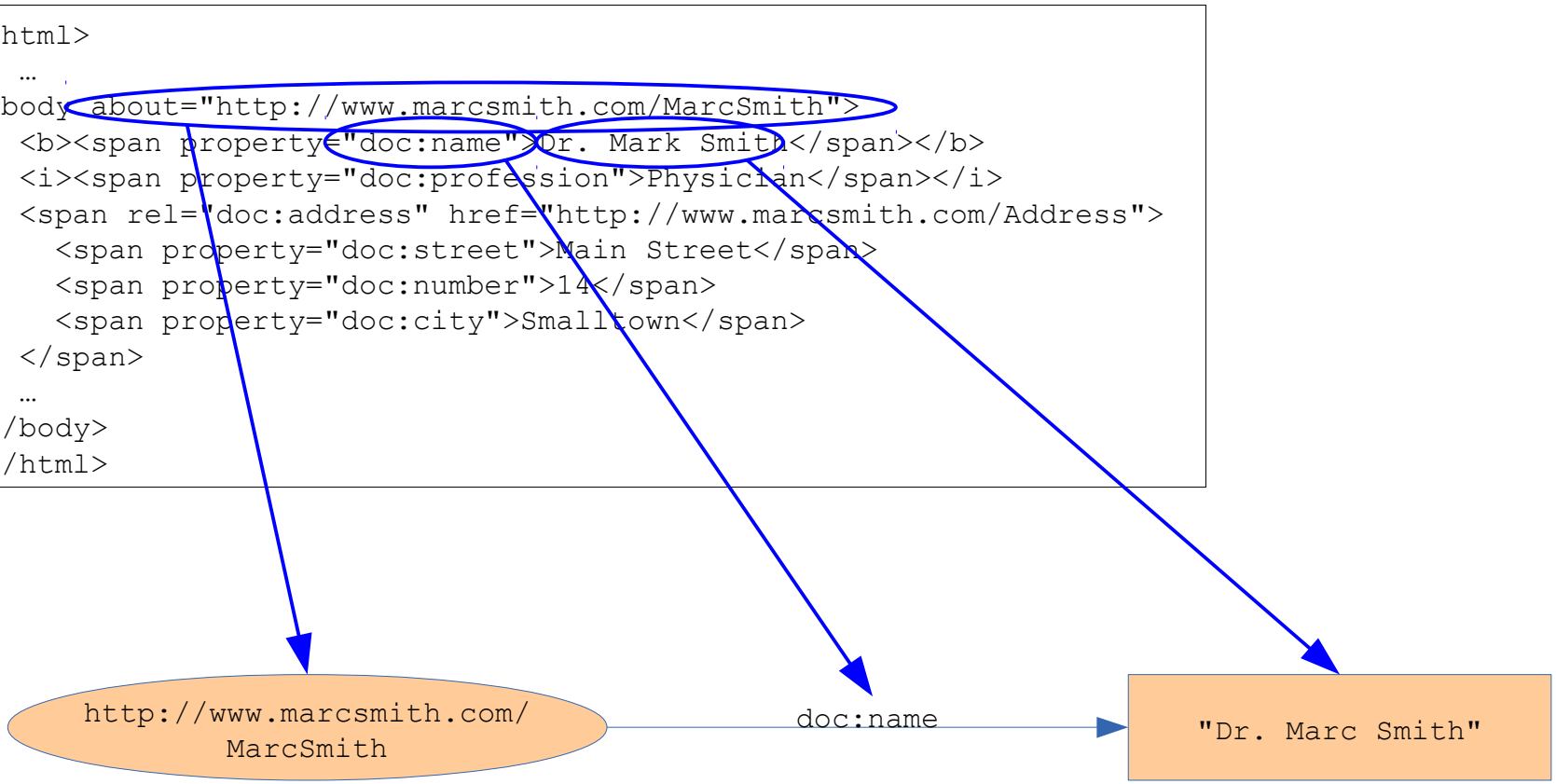
```
<html>
  …
<body about="http://www.marcsmith.com/MarcSmith">
  <b><span property="doc:name">Dr. Mark Smith</span></b>
  <i><span property="doc:profession">Physician</span></i>
  <span rel="doc:address" href="http://www.marcsmith.com/Address">
    <span property="doc:street">Main Street</span>
    <span property="doc:number">14</span>
    <span property="doc:city">Smalltown</span>
  </span>
  …
</body>
</html>
```

RDFa Language Constructs

- about="http://foo.bar/aSubject"
  - Defines the subject of a page or section
- property = "http://foo.bar/aProperty"
  - Defines a relation
  - Contents of the tag are interpreted as a literal
- rel = "http://foo.bar/aRelation"
  - Defines a relation to another resource
- href = "http://foo.bar/aResource"
  - Defines a relation's object
  - can be the subject of a resource again
- typeof = "http://foo.bar/aType"
  - defines a resource's type

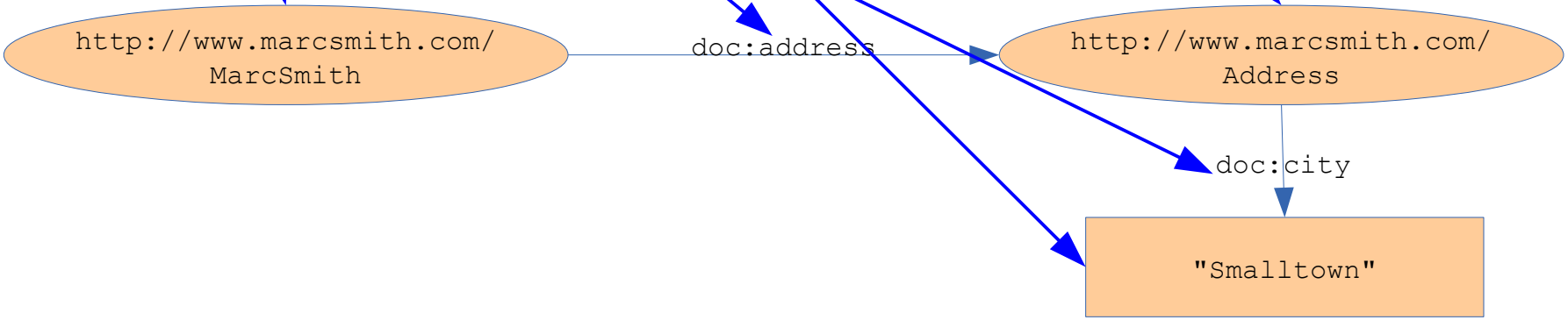
RDF in Attributes (RDFa)

```
<html>
  …
<body about="http://www.marcsmith.com/MarcSmith">
  <b><span property="doc:name">Dr. Mark Smith</span></b>
  <i><span property="doc:profession">Physician</span></i>
  <span rel="doc:address" href="http://www.marcsmith.com/Address">
    <span property="doc:street">Main Street</span>
    <span property="doc:number">14</span>
    <span property="doc:city">Smalltown</span>
  </span>
  …
</body>
</html>
```



RDF in Attributes (RDFa)

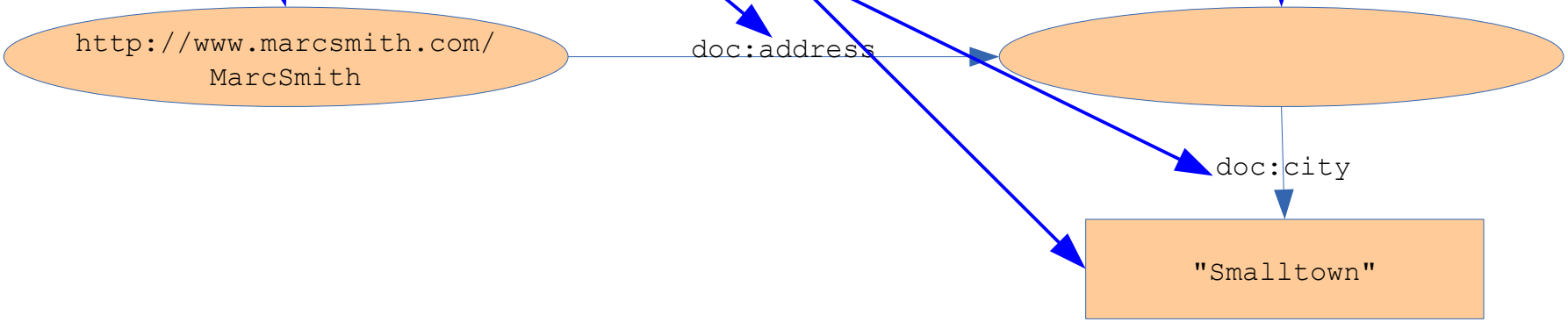
```
<html>
  …
<body about="http://www.marcsmith.com/MarcSmith">
  <b><span property="doc:name">Dr. Mark Smith</span></b>
  <i><span property="doc:profession">Physician</span></i>
  <span rel="doc:address" href="http://www.marcsmith.com/Address">
    <span property="doc:street">Main Street</span>
    <span property="doc:number">14</span>
    <span property="doc:city">Smalltown</span>
  </span>
  …
</body>
</html>
```



RDF in Attributes (RDFa)

```
<html>
  …
<body about="http://www.marcsmith.com/MarcSmith">
  <b><span property="doc:name">Dr. Mark Smith</span></b>
  <i><span property="doc:profession">Physician</span></i>
  <span rel="doc:address">
    <span property="doc:street">Main Street</span>
    <span property="doc:number">14</span>
    <span property="doc:city">Smalltown</span>
  </span>
  …
</body>
</html>
```

Relations without "href" become blank nodes!



Alternative to RDFa: Microdata

- Adding structured information to web pages
  - By marking up contents
  - Arbitrary vocabularies are possible
  - Introduced with HTML5
- Similar to RDFa


```
<div itemscope
itemtype="http://schema.org/PostalAddress">
 <span itemprop="name">Data and Web Science Group</span>
 <span itemprop="addressLocality">Mannheim</span>,
 <span itemprop="postalCode">68131</span>
 <span itemprop="addressCountry">Germany</span>
</div>
```

Alternative to RDFa: Microdata

- Markup can be extracted to RDF
  - See W3C Interest Group Note: Microdata to RDF [1]


```
<div itemscope
itemtype="http://schema.org/PostalAddress">
 <span itemprop="name">Data and Web Science Group</
```

```
_:1 a <http://schema.org/PostalAddress> .
_:1 <http://schema.org/name> "Data and Web Science Group" .
_:1 <http://schema.org/addressLocality> "Mannheim" .
_:1 <http://schema.org/postalCode> "68131" .
_:1 <http://schema.org/adressCounty> "Germany" .
```

[1] http://www.w3.org/TR/microdata-rdf/

Alternative to RDFa: Microdata

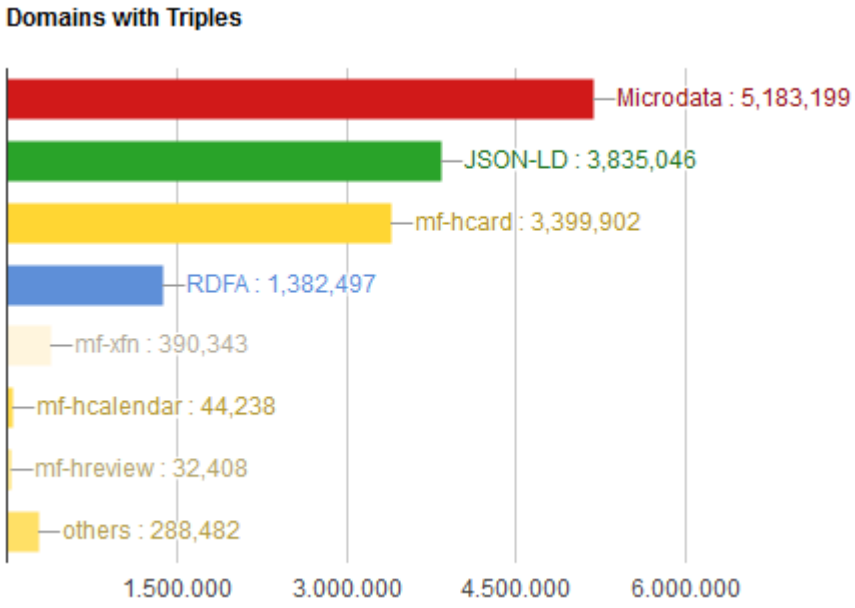
- Commonalities
  - Arbitrary classes/predicates are possible
  - Although Microdata is mainly used with schema.org

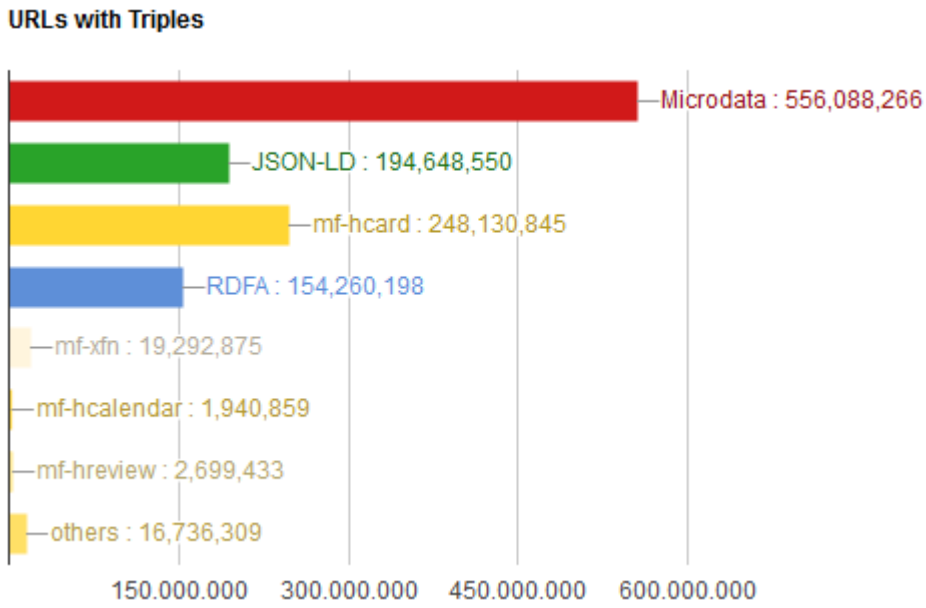
- Differences
  - Microdata is slightly less expressive
  - No URIs, only blank nodes
  - No cycles in the RDF graph
  - No reification (see later)

RDFa, MicroFormats, and Microdata

- MicroFormats: fixed vocabularies for persons, addresses, etc.

- WebDataCommons: Large-Scale Extraction of RDFa, MicroFormats, and Microdata from the Web





http://webdatacommons.org/structureddata/2018-12/stats/stats.html

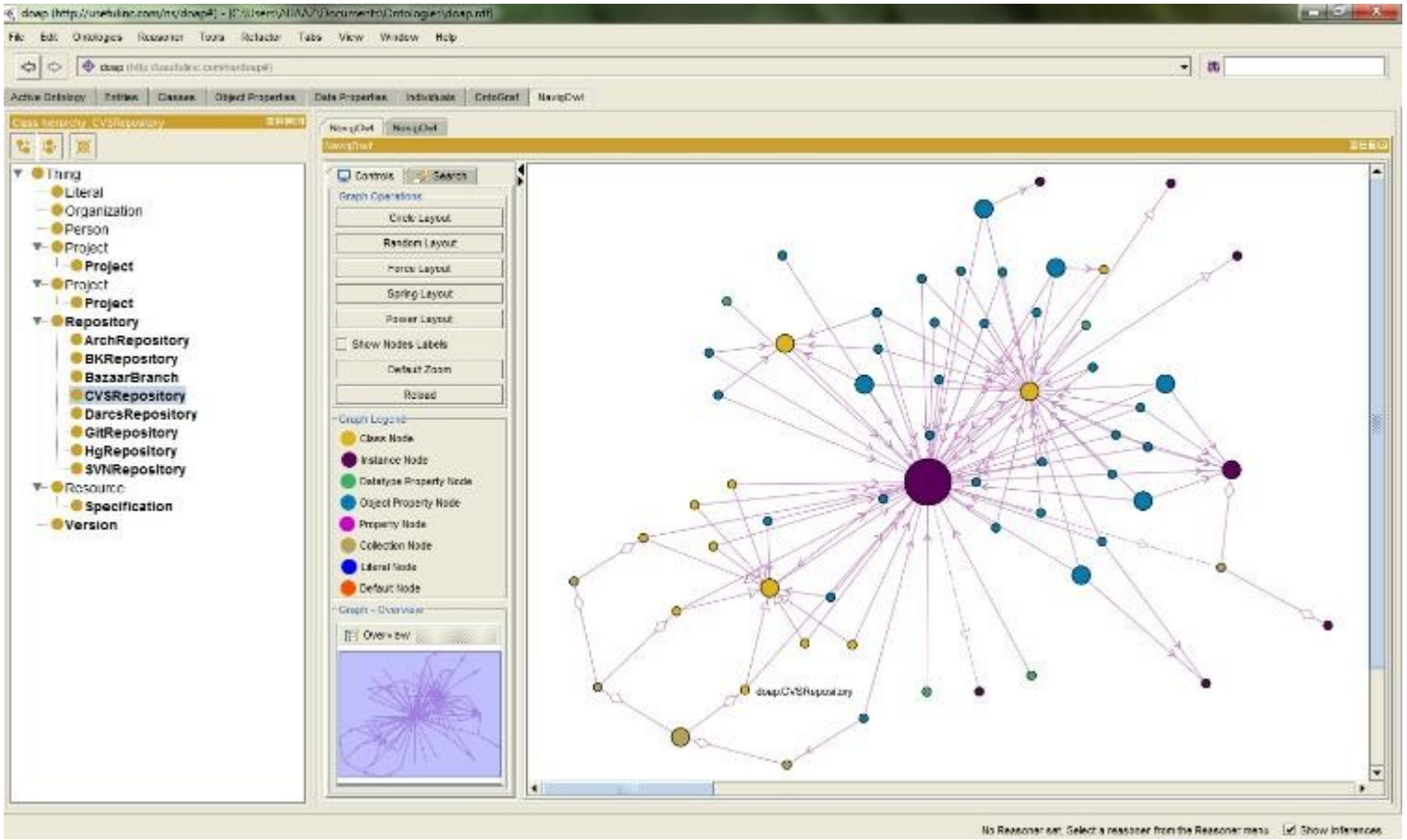
RDF Tools: Storage

- RDF is often stored in relational databases
- Different storage strategies
  - single triple table
  - one table per class
  - one table per property
- Strategies differ
  - by requirements of (disk) space
  - by query response time for different query types

- Examples: Virtuoso, Sesame, ...

RDF Tools: Visualization

- Mostly graph-based visualization tools



RDF Tools: Validation

- W3C RDF Validator:
  - W3C RDF Validator: http://www.w3.org/RDF/Validator/
  - Output of RDF/XML and graphs

- EasyRDF Validator:
  - http://www.easyrdf.org/converter
  - Understands and converts a variety of notations

RDF Tools: Programming, Reasoning

- Programming Frameworks
  - for developing RDF-based applications
  - e.g., JENA, RDFReactor, ...

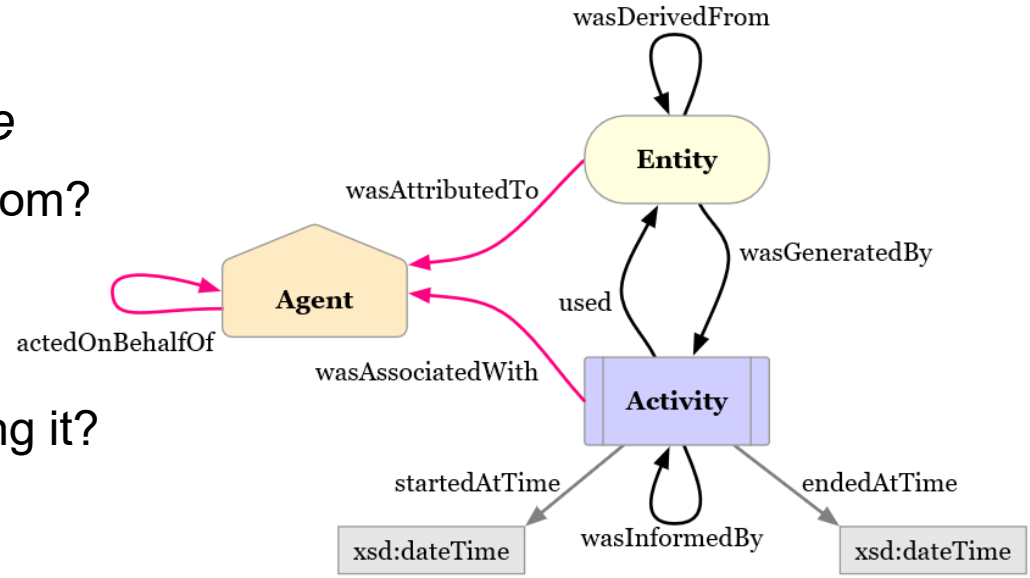
- Reasoners
  - can draw logical conclusions from RDF graphs
  - can answer queries on RDF graphs

- Both will be covered in separate lectures

Metadata for RDF

- Recap: Dublin Core was designed as Metadata for the Web
- On the Semantic Web, we may have metadata as well

- Most prominently: *provenance*
  - Where does the data come from?
  - Who created it?
  - When was it created?
  - What was the process creating it?
  - …



https://www.w3.org/TR/prov-o/

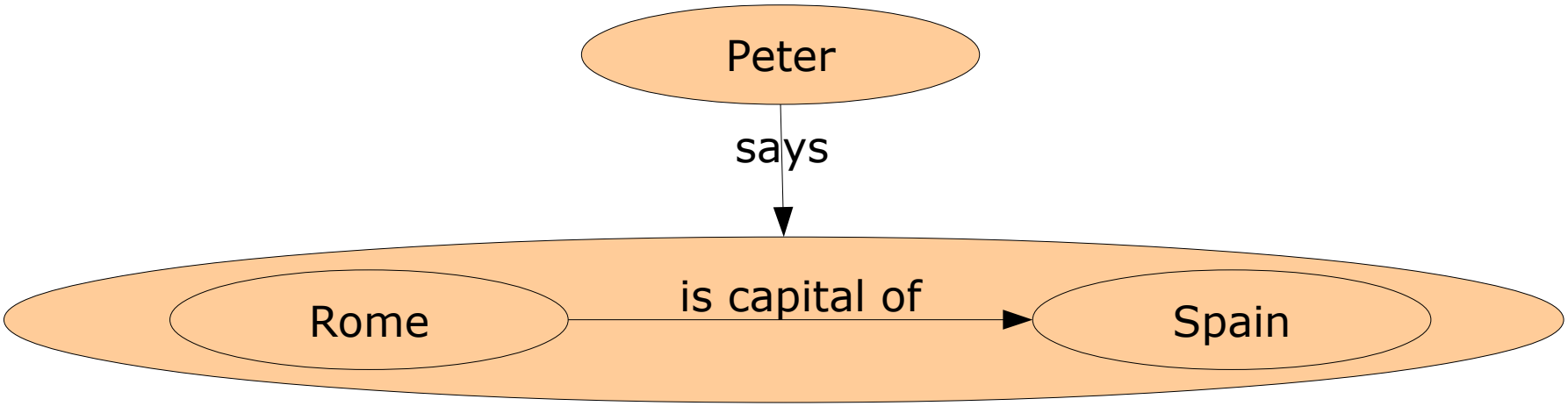
Reification

- Latin *res* ("Thing"), *facere* ("make")
  - an Explication
  - making a statement, an opinion etc. the subject of a statement
- In RDF: Statements about statements

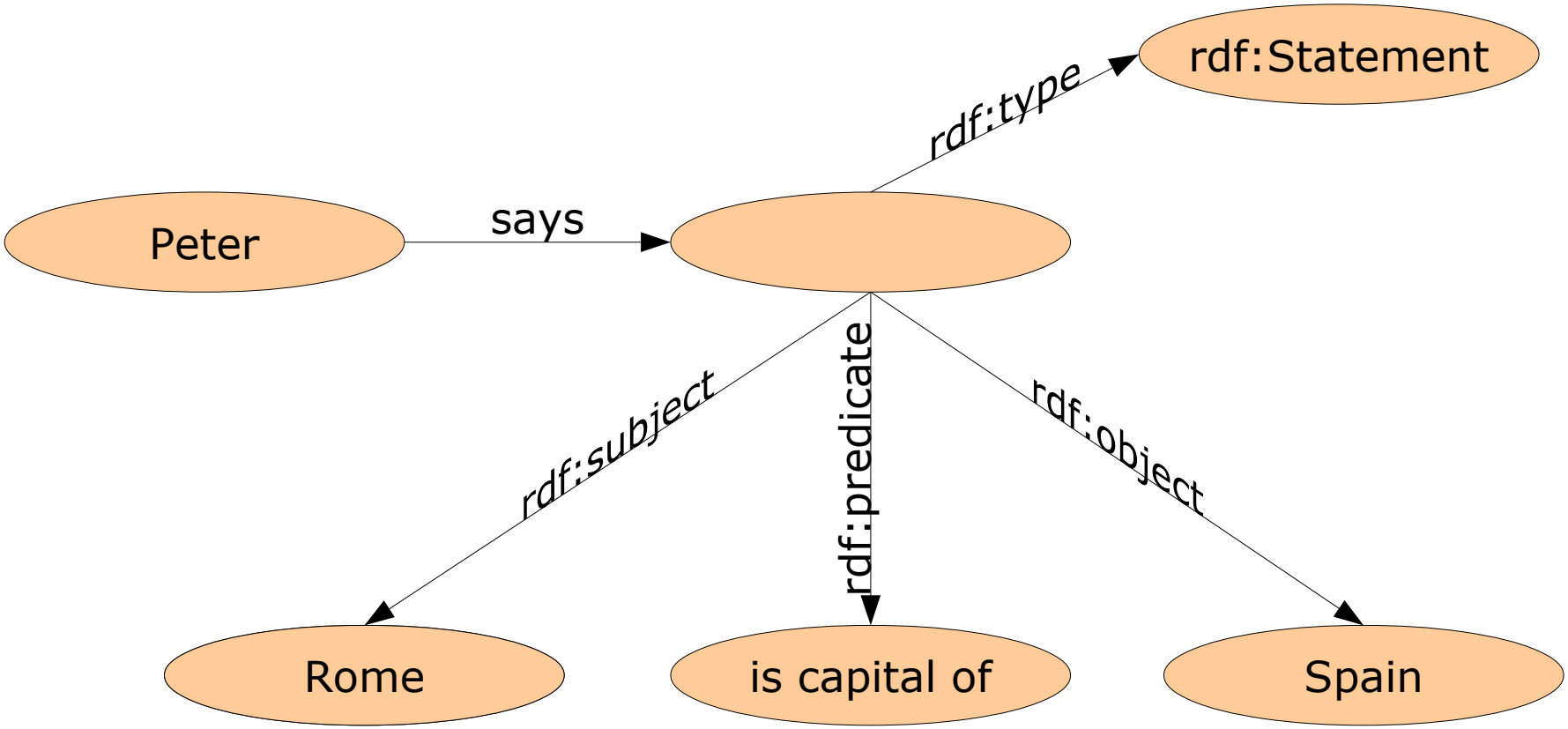
- "Peter says that Rome is the capital of Spain."

  Implementation:
  - RDF Statements are considered resources themselves
  - Can be subject or object of other statements
- Reification can have multiple levels
  - “Peter says that Wikipedia states that Rome is the capital of Spain.”

Reification in RDF



Implementing Reification as Standard RDF



Encoding Reification in Turtle

- Variant 1: Named Statement (with URI)

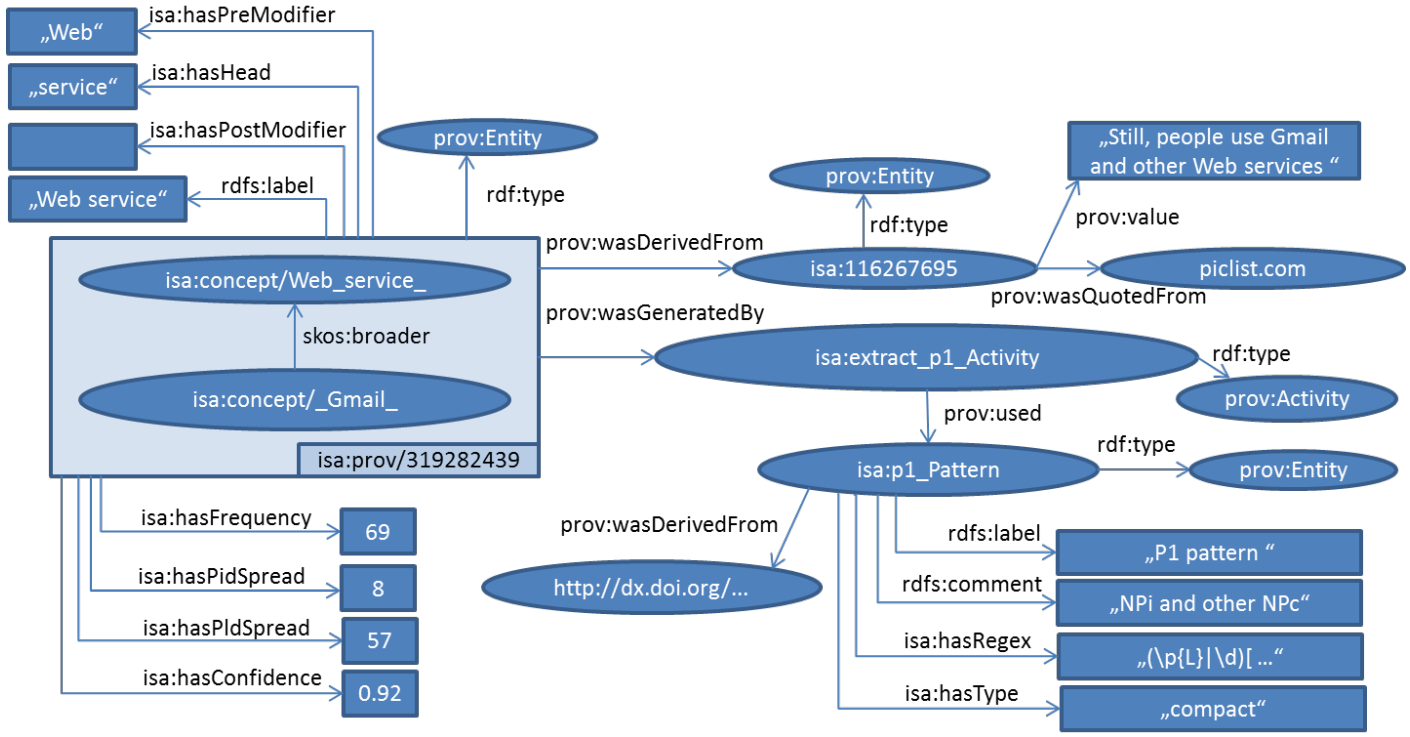
```
:triple1 rdf:type rdf:Statement ;
         rdf:subject :Rome ;
         rdf:predicate :isCapitalOf ;
         rdf:object :Spain .
:Peter :says :triple1 .
```

- Variant 2: Unnamed Statement (Blank Node)

```
:Peter :says [
       a rdf:Statement ;
       rdf:subject :Rome ;
       rdf:predicate :isCapitalOf ;
       rdf:object :Spain .
] .
```

Reification in the Wild

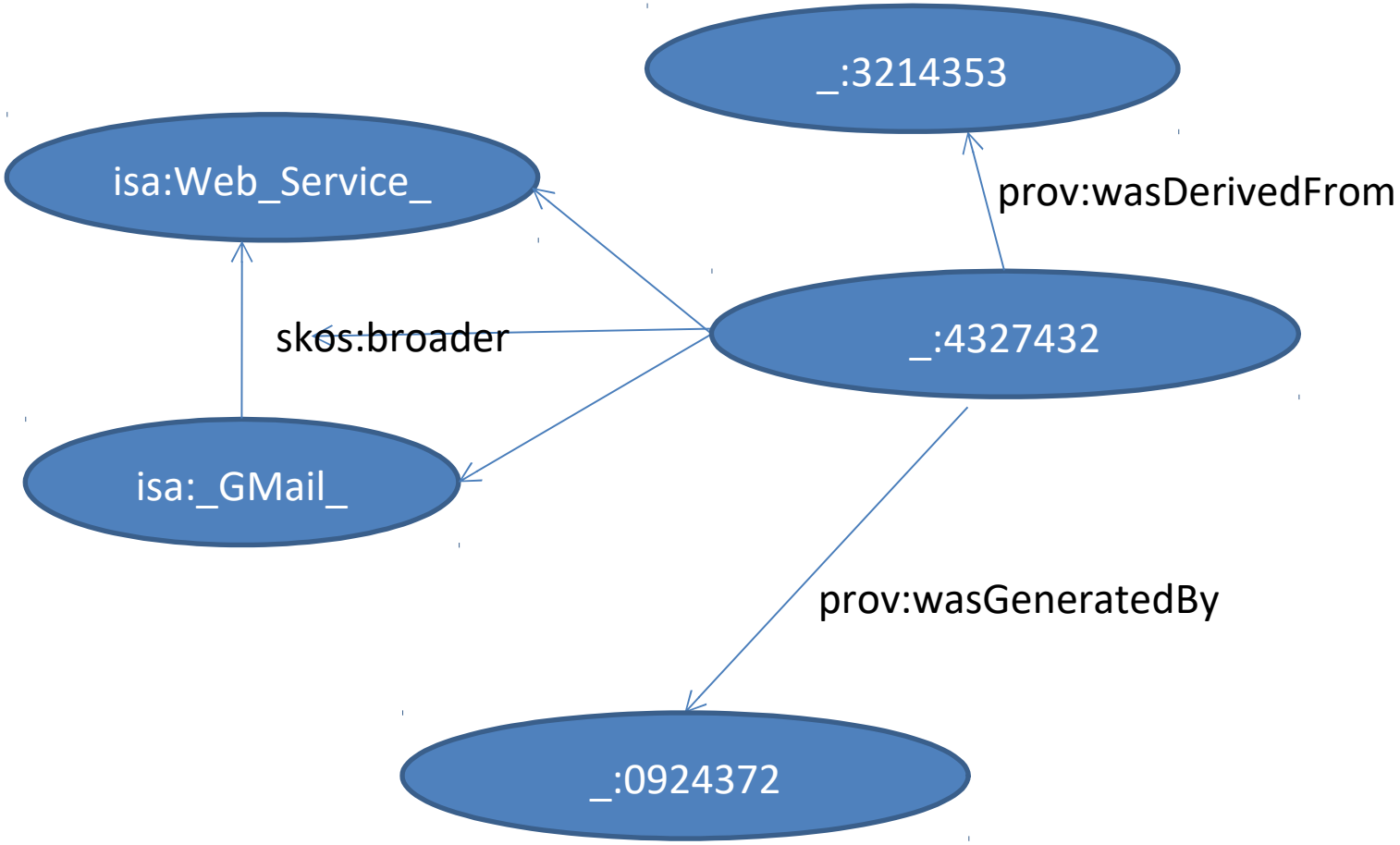
- Example: the WebIsALOD dataset



http://webisa.webdatacommons.org/

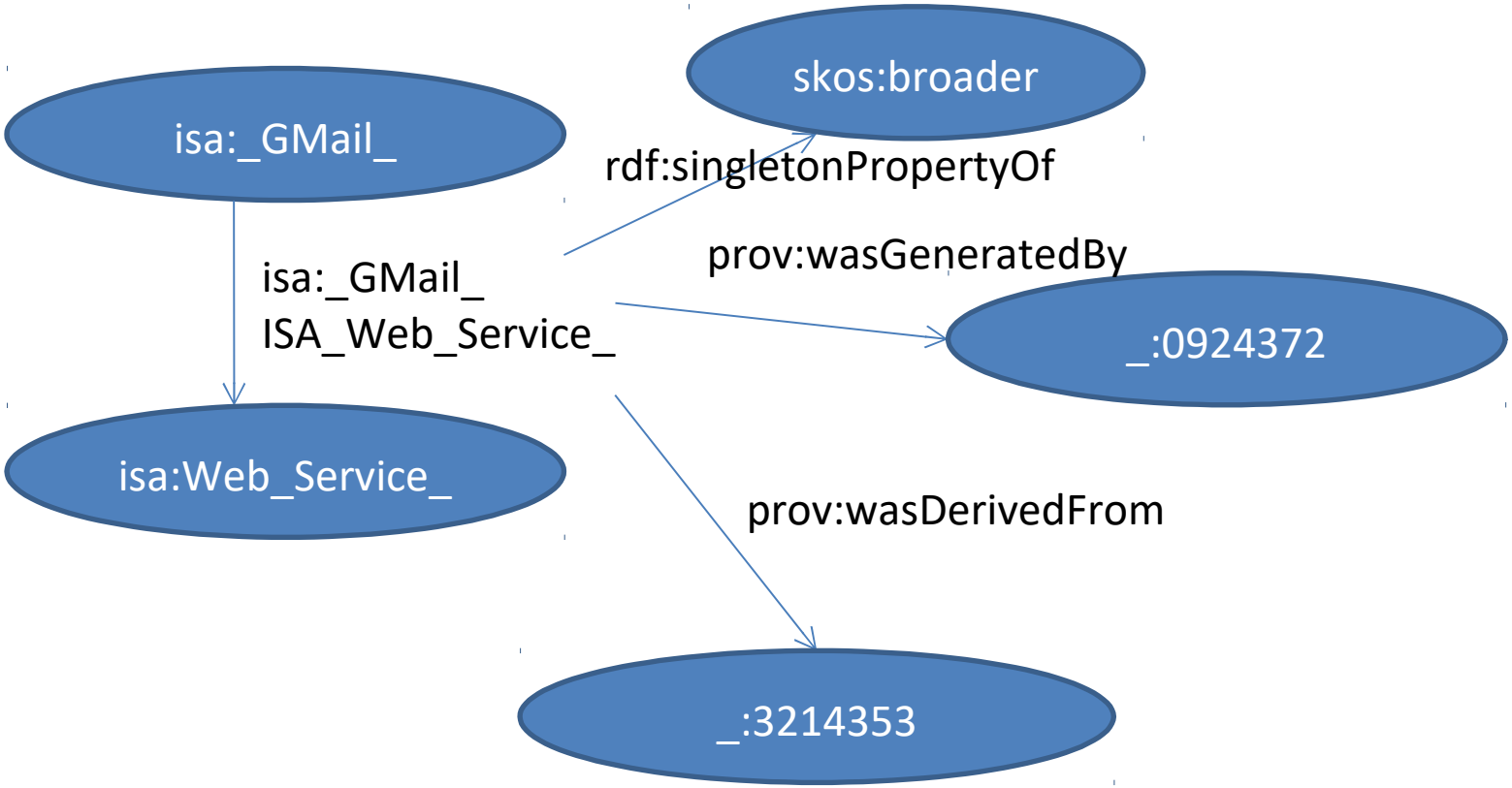
Design Decisions: Provenance in WebIsALOD

- Variant 1: RDF Reification



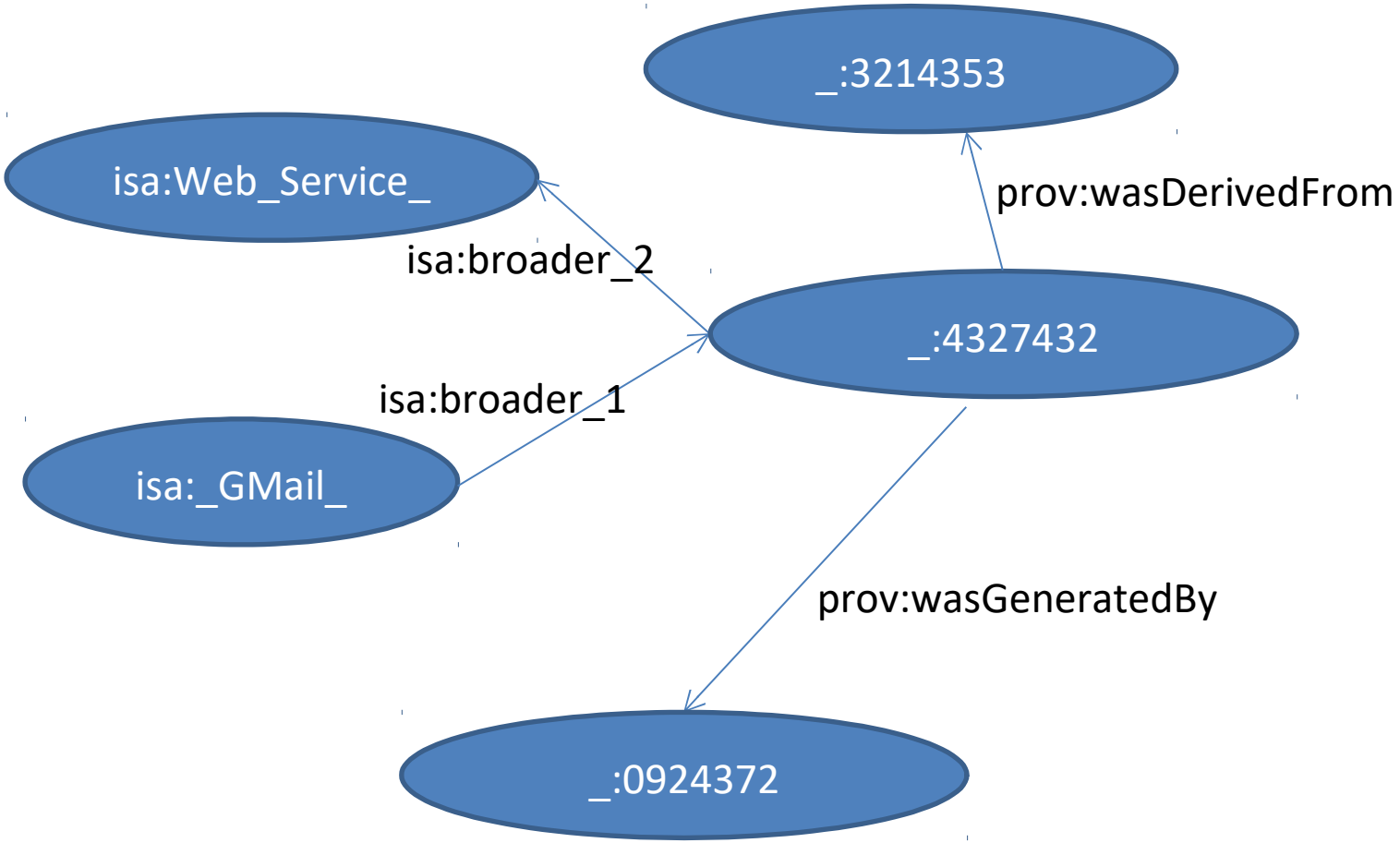
Design Decisions: Provenance in WebIsALOD

- Variant 2: Singleton Property



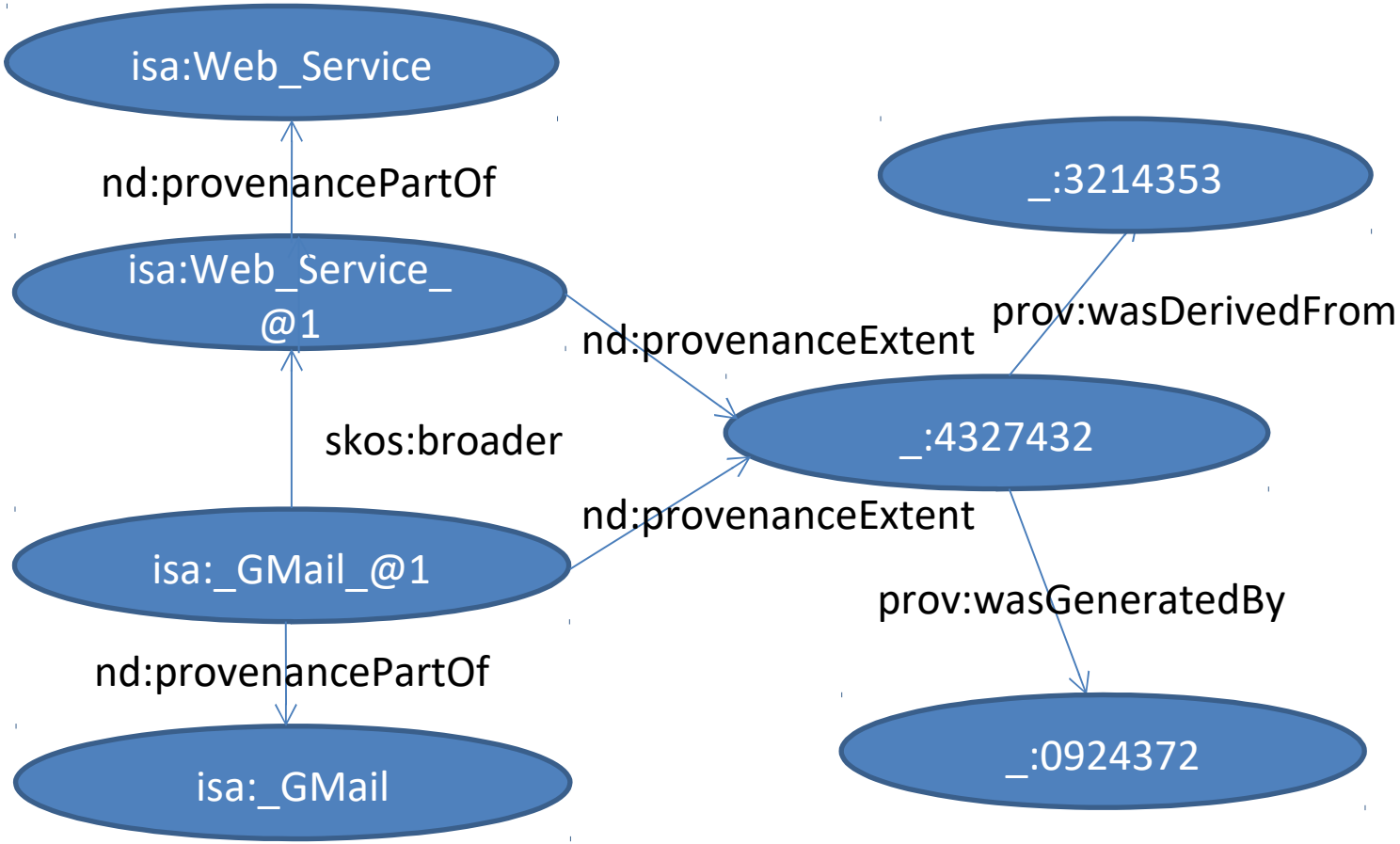
Design Decisions: Provenance in WebIsALOD

- Variant 3: n-ary relations



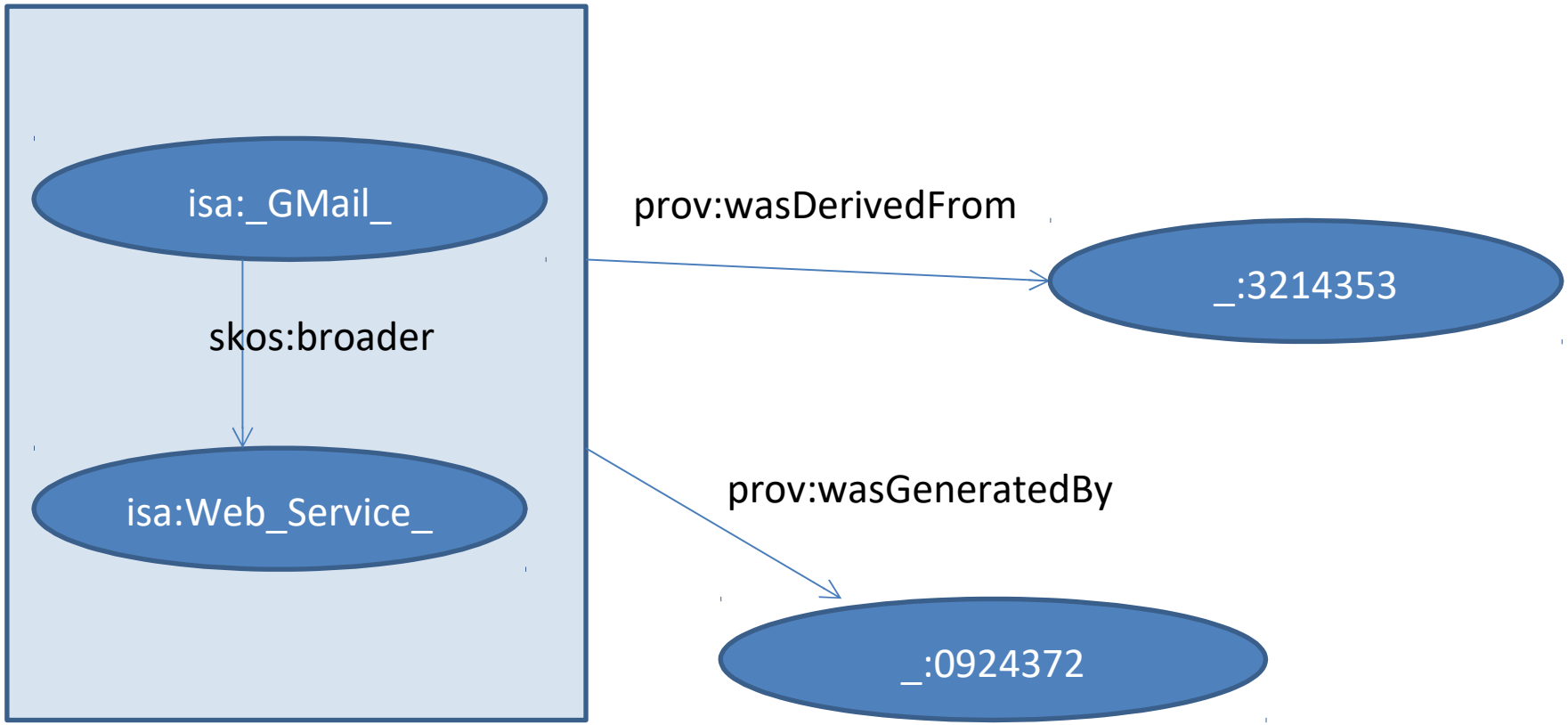
Design Decisions: Provenance in WebIsALOD

- Variant 4: NdFluents



Design Decisions: Provenance in WebIsALOD

- Variant 5: RDF Graphs



Design Decisions: Provenance in WebIsALOD

- Challenge 1: Verbosity
  - WebIsALOD has 400M core statements

RDF Graphs are least verbose

- Challenge 2: Usability
  - Querying should not be overly complex
  - (see later)

NDFluents and reification lead to longer queries

- Challenge 3: Understandability
  - We should keep the learning curve low

Reification and RDF Graphs are standardized

- Challenge 4: Scalability
  - Support by RDF store should be considered

Virtuoso recommends RDF Graphs

Wrap Up

- RDF is a language for describing arbitrary things
  - interpretation: set of statements or directed graph
  - Notations: RDF/XML, Turtle
- Special language constructs
  - Blank nodes
  - Reification and its variants
- Semantics
  - Non-unique name assumption
  - Open world assumption
- Embedding in HTML is possible
- Large set of tools is available

A Critical Look in the Rear View Mirror

- Is RDF more powerful than XML?

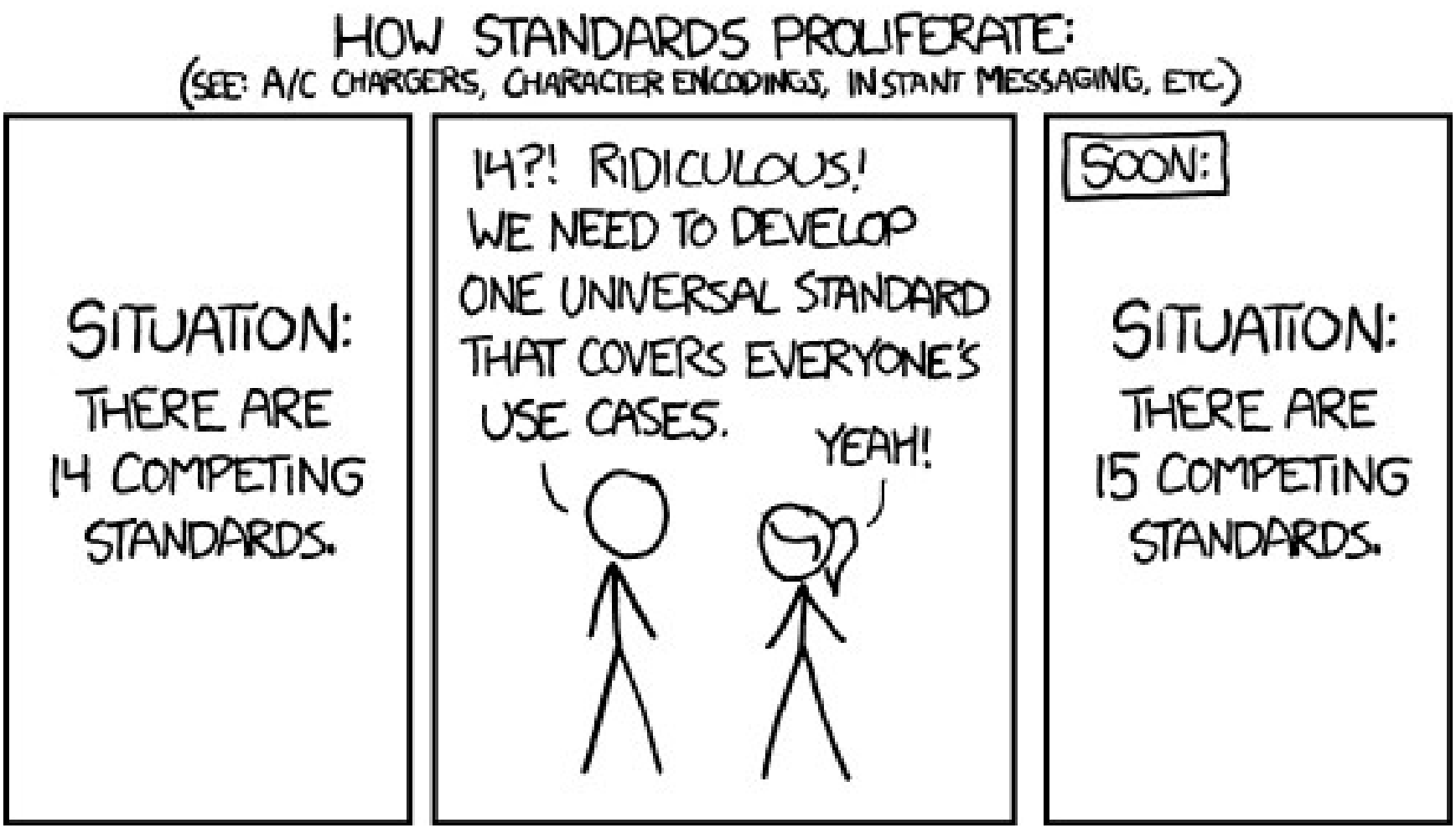
- XML is a markup language for information
- In XML, arbitrary elements and attributes can be defined
- XML tag names are meaningless for a computer

- RDF is a markup language for information
- In RDF, arbitrary classes and predicates can be defined
- RDF class and predicate names are meaningless for a computer

A Critical Look in the Rear View Mirror

- So, why did we spend an entire lecture on RDF?

A Critical Look in the Rear View Mirror



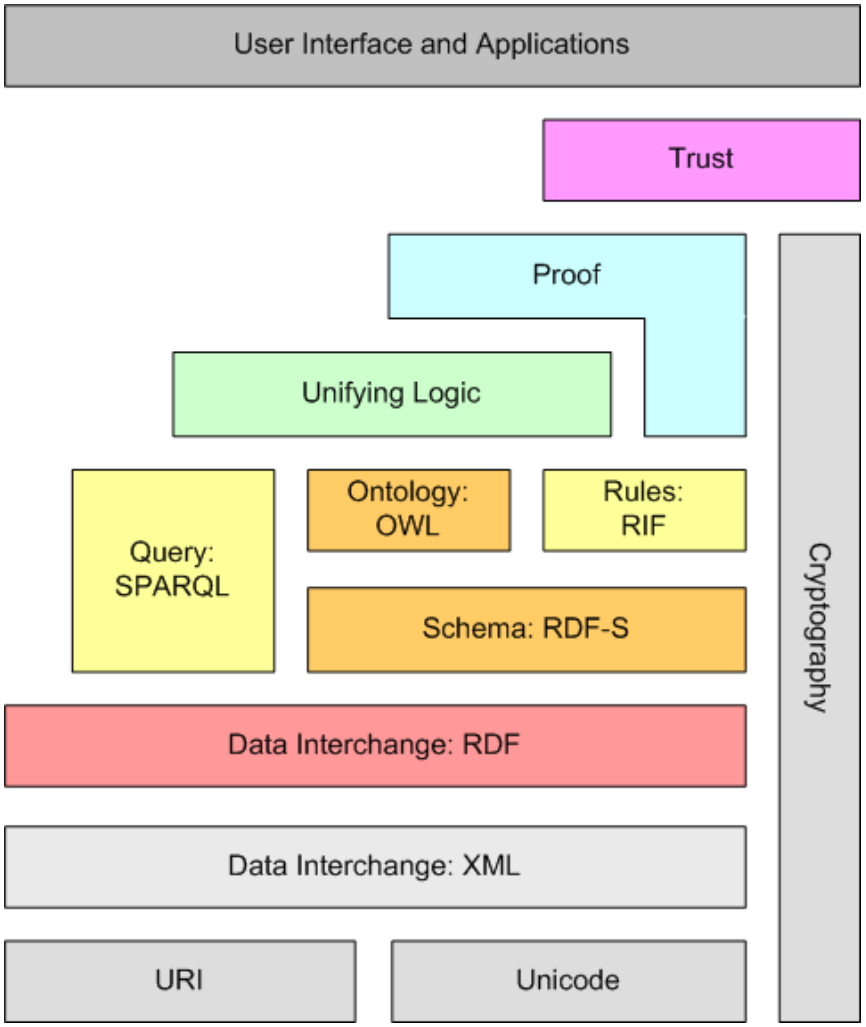
http://xkcd.com/927/

Semantic Web – Architecture

here be dragons...

Semantic Web Technologies (This lecture)

Technical Foundations



Berners-Lee (2009): *Semantic Web and Linked Data*
http://www.w3.org/2009/Talks/0120-campus-party-tbl/

Questions?