RDF, linked data and semantic web

Jose Emilio Labra Gayo

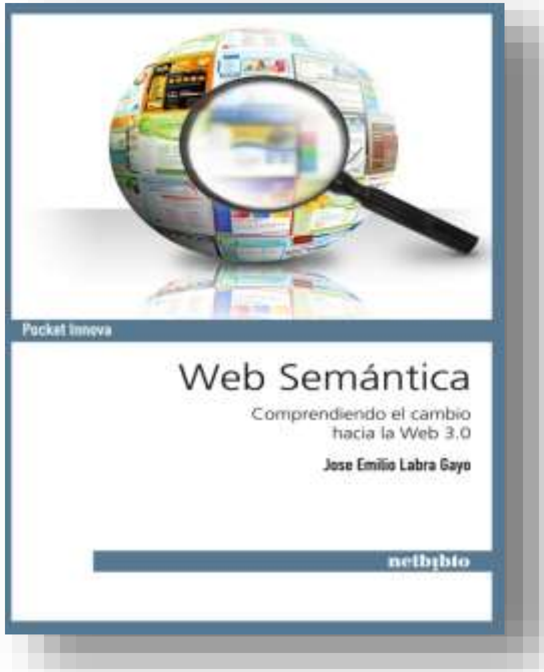
Departamento de Informática
Universidad de Oviedo

About me

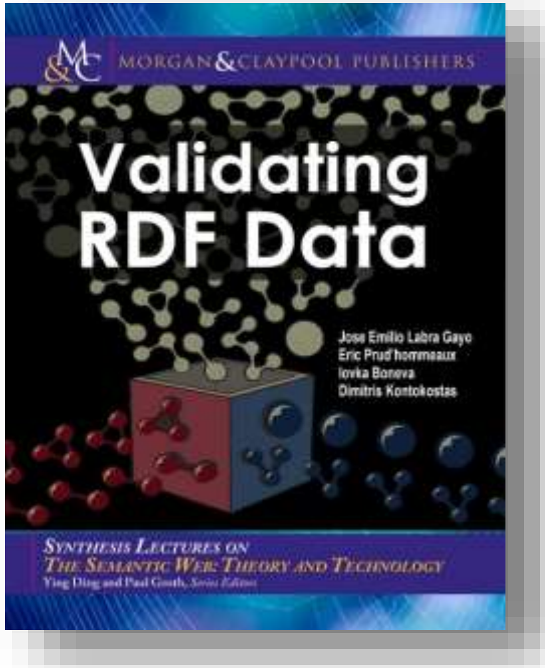
Main researcher WESO research group (WEb Semántics Oviedo)

Author of the following books:

Web Semántica (2012)



Validating RDF Data (2017)


MORGAN & CLAYPOOL PUBLISHERS
Validating RDF Data
Jose Emilio Labra Gayo
Eric Prud'hommeaux
Iovka Boneva
Dimitris Kontokostas
Synthesis Lectures on The Semantic Web: Theory and Technology
Ying Ding and Paul Groth, Series Editors

http://www.di.uniovi.es/~labra

Structure of presentation

- Why RDF?
- RDF data model
- RDF ecosystem
  - RDF applications
  - Inference systems: RDFS, OWL
  - SPARQL
- Challenges

Why RDF?

Data Web is coming!

The flood of data

Producing data is more and more easy

Open trends

- *Open Software*
- *Open Content*
- ***Open Data***
- *Open Science*
- *Open Government*

Old models are affected

- Músic, Films, finance,...
- Education
- Government ...

Why?

Reasons for governments
- Transparency
- Leadership
  - Government as catalyzer
  - Promote participation
  - New initiatives and apps

Reasons for citizens
- Data belong to us
- Created with public money
- We want better services

OK, long live to data! but…

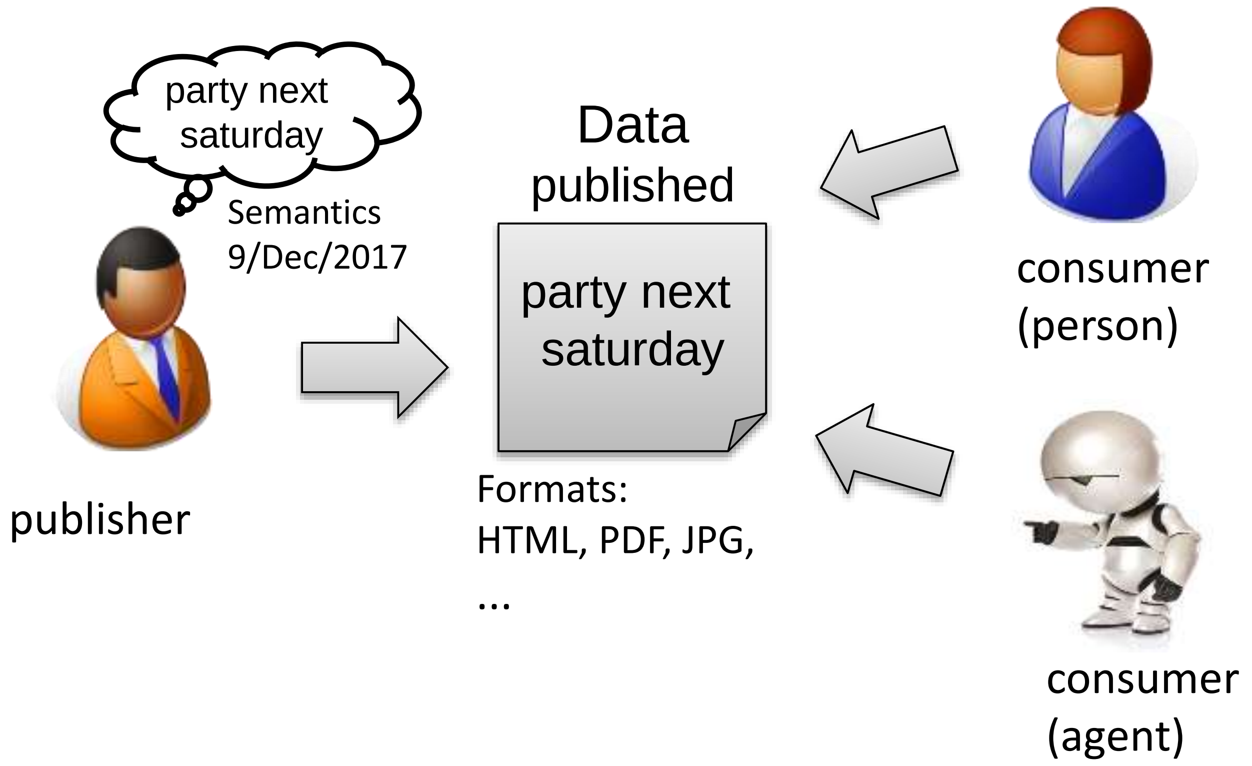
How to publish it?

Problems of current web

- It is not enough to publish data
  - It must be found
  - If not found, as if it doesn't exist
- It must be usable
  - If not usable, it is worthless
  - Reuse data in unexpected contexts

Semantic loss

- When publishing data, some semantics is lost
  - The person that wants to publish has more info about the data
  - Some info is lost during the publication process



party next saturday

Semantics 9/Dec/2017

publisher

Data published

party next saturday

Formats:
HTML, PDF, JPG, ...

consumer (person)

consumer (agent)

HTML doesn't have enough semantics

HTML is intended to publish hypertext

- HTML tags are understood by browsers
- Information inside tags = natural language
- Machines don't understand natural language *yet*

```
<p>Event:
<ul>
<li>Name: Concert</li>
<li>Date: Next saturday</li>
</ul>
</p>
```

```
<p>իրադարձություն:
<ul>
<li>տիպ:  համերգ</li>
<li>ամսաթիվ: հաջորդ շաբաթ/li>
</ul>
</p>
```

XML problem

XML goes a step forward

- Specific vocabularies have meaning in some specific context
- Specific applications can process XML documents
- XML documents are difficult to integrate if they are from different domains

```
<event>
 <name>Concert</name>
 <date>Next saturday</date>
</event>
```

```
<event>
 <name>համերգ</name>
 <date>հաջորդ շաբաթ</date>
</event>
```

```
<իրադարձություն>
 <տիպ>համերգ</տիպ>
 <ամսաթիվ>հաջորդ շաբաթ</ամսաթիվ>
</իրադարձություն>
```

Json problem

JSON is almost the same as XML

- It may be easier to parse and processby developers
- But the meaning depends on each domain
- It is even worse as there are no namespaces or validation

```
{
  "event": {
   "name": "Party" ,
   "date": "Next saturday"
 }
}
```

```
{
  "event": {
   "name": "համերգ" ,
   "date": "հաջորդ շաբաթ"
 }
```

```
{
  "իրադարձություն": {
   "տիպ": "համերգ" ,
   "ամսաթիվ": "հաջորդ շաբաթ"
 }
}
```

Towards semantic web

Semantic web = vision of the data web

Goal: Share and Reuse data between applications, and communities

Tim Berners Lee
Source: Wikipedia

From a web of documents to a web of data

Benefits

- Accessible data
  - Avoid semantic loss
  - Facilitate task automation
- Linked data
  - Data reuse
  - Application integration

The best way to use your data will be found by other people

Jo Walsh, Rufus Pollock, http://www.okfn.org/files/talks/xtech_2007/

Web features

Non centralized

Difficult to ensure data integrity and quality

Dynamic information

Information is constantly changing

Big amounts of information

Big data

3Vs: Volume, Velocity, Variety

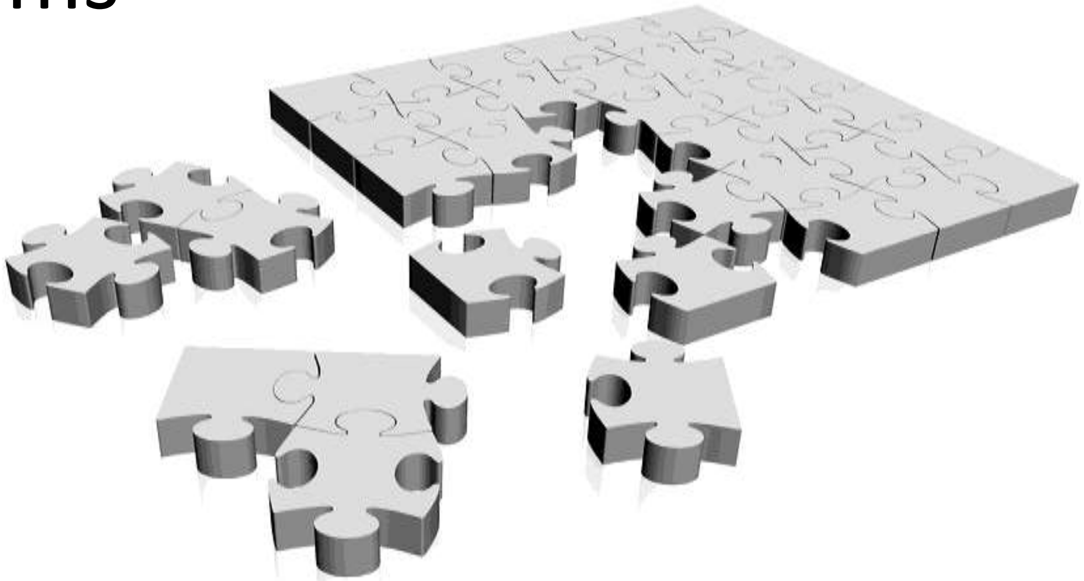
Open system

AAA lemma: Anyone can say Anything about Any topic

It is not enough to publish data…

Biggest challenge = Integration

- In general, the challenge is not to *computerize* something
- The challenge is **integrate** systems
- **Interoperability**



Publish = make something accessible

Accessibility levels

- Physical disability
- Technical disability: other environments
- Cultural and intellectual
  - Illiteracy
  - Knowledge barriers
  - Other languages…
- Accessible to machines

Star model*

http://5stardata.info/

★ Publish data
(any format)

★★ Use structured formats
(Excel instead of scanned pictures)

★★★ Non proprietary structured formats
(CSV instead of Excel)

★★★★ Use URIs to identify data
(other systems can link to our data)

★★★★★ Link to other data
(provide contextual information)

* Tim Berners-Lee, Gov 2.0 Expo 2010
http://www.youtube.com/watch?v=ga1aSJXCFe0

Unstructured formats

Black box formats: Pictures, video, audio, etc.

Binary formats: PDF, PS, etc.

They require low level techniques, pattern recognition, signal processing, etc

★ ★

Structured formats

Data have some structure

Example: Excel

- Problem with proprietary formats
- May require non-free tools

Non-proprietary formats ★ ★ ★

Use open-structured formats

Examples: CSV, HTML

Problem: Content depends on context

URIs identify data ★★★★

Use URIs to identify data

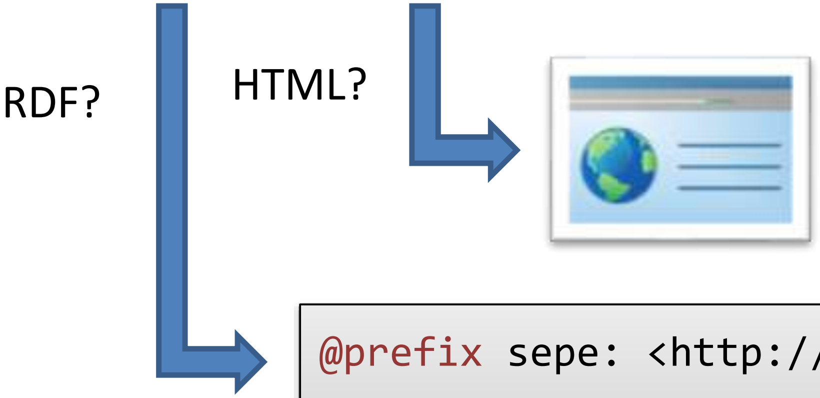
Content negotiation can provide different representations

Example

Example: RDF

★ ★ ★ ★

<http://www.sepe.es/data/unemployment/Asturias/Allande/2013/10>



```
@prefix sepe: <http://www.sepe.es/datos/>

sepe:obs1 sepe:municipality "Allande" ;
          sepe:unemployees  18 .
```

Link with other data

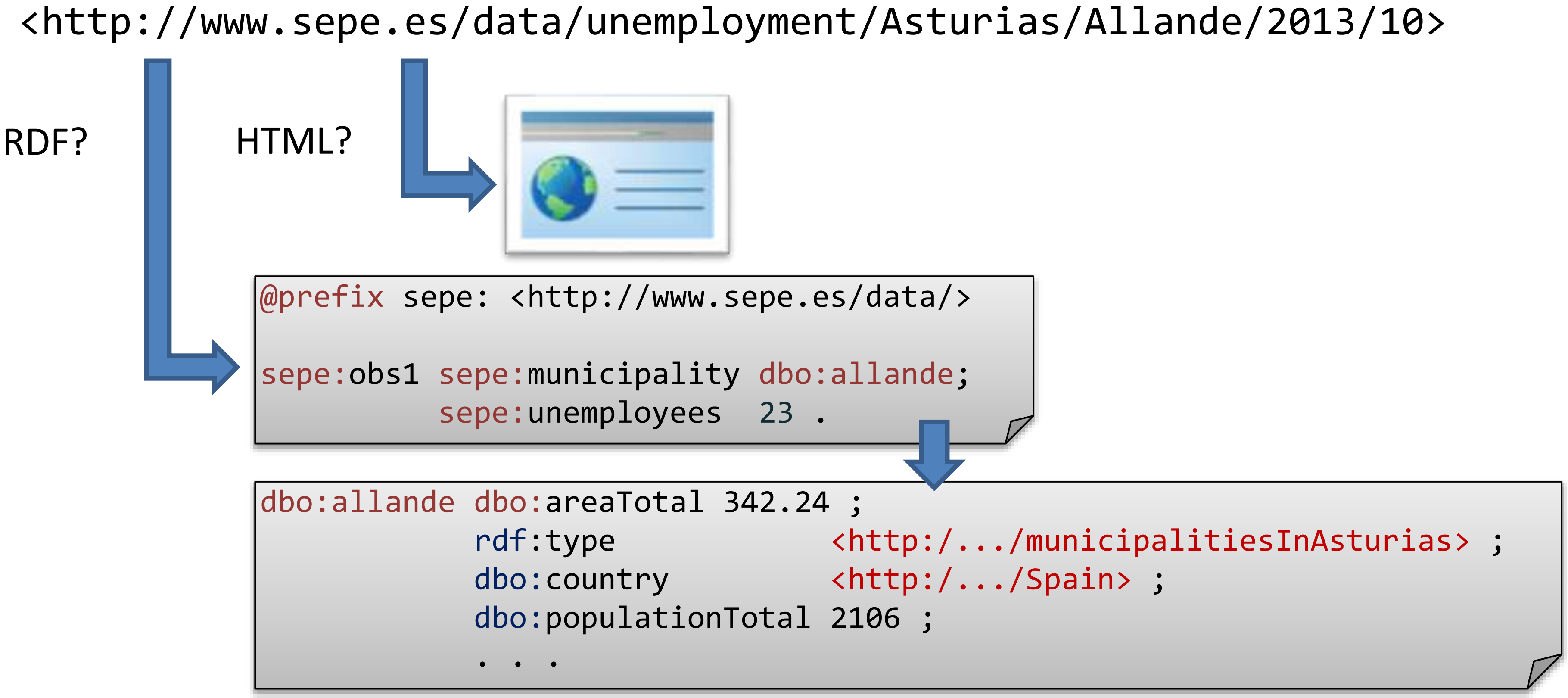
★ ★ ★ ★ ★

Representations return links to other data

It allos to:

- Reuse and find other data
- Unforeseen applications

Linked data example ★ ★ ★ ★ ★



```
<http://www.sepe.es/data/unemployment/Asturias/Allande/2013/10>
```

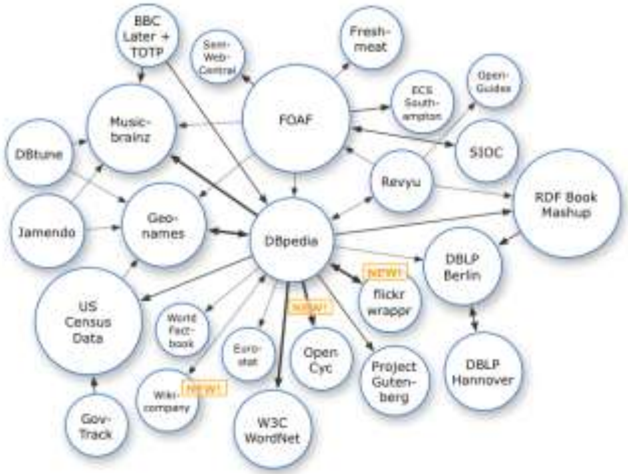
```
@prefix sepe: <http://www.sepe.es/data/>

sepe:obs1 sepe:municipality dbo:allande;
          sepe:unemployees  23 .
```

```
dbo:allande dbo:areaTotal 342.24 ;
            rdf:type            <http:/.../municipalitiesInAsturias> ;
            dbo:country         <http:/.../Spain> ;
            dbo:populationTotal 2106 ;
            . . .
```

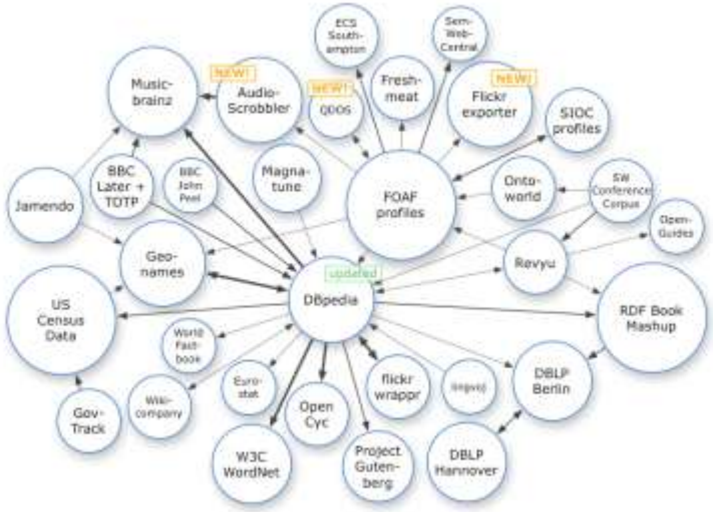
Linked Open Data principles ★ ★ ★ ★ ★

1. Use URIs to denote things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

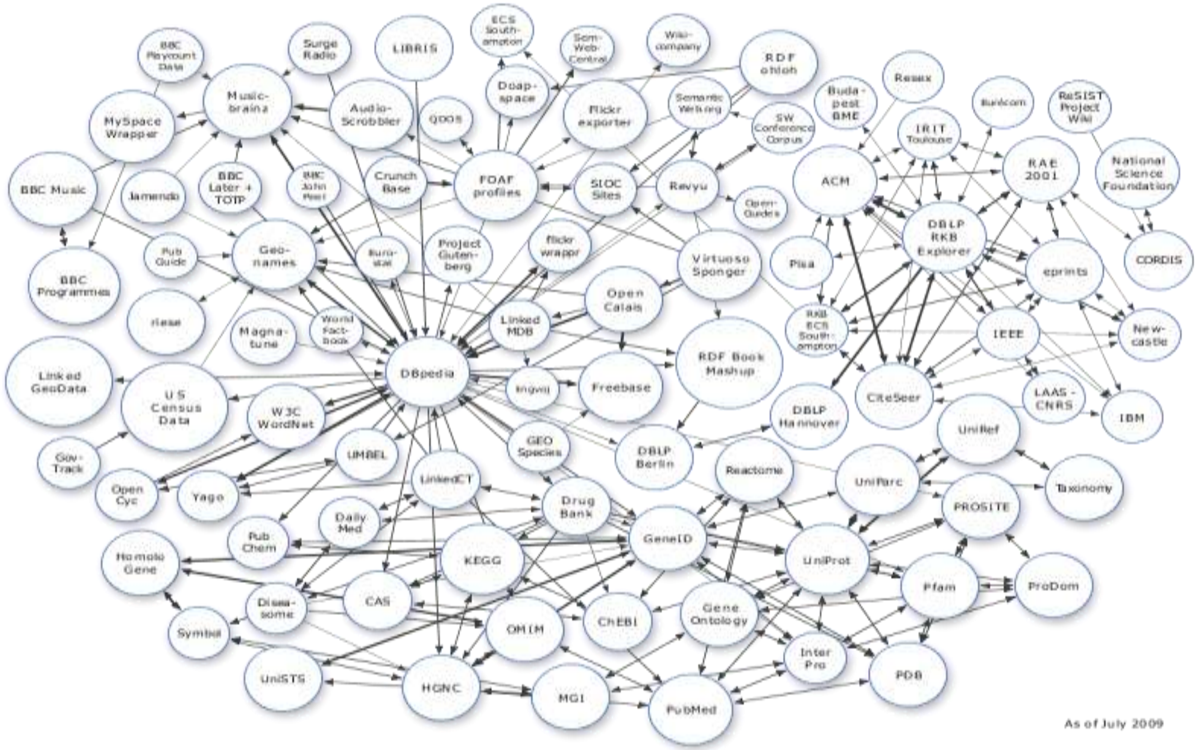
Linked open data (2007)



Linked open data (2008)

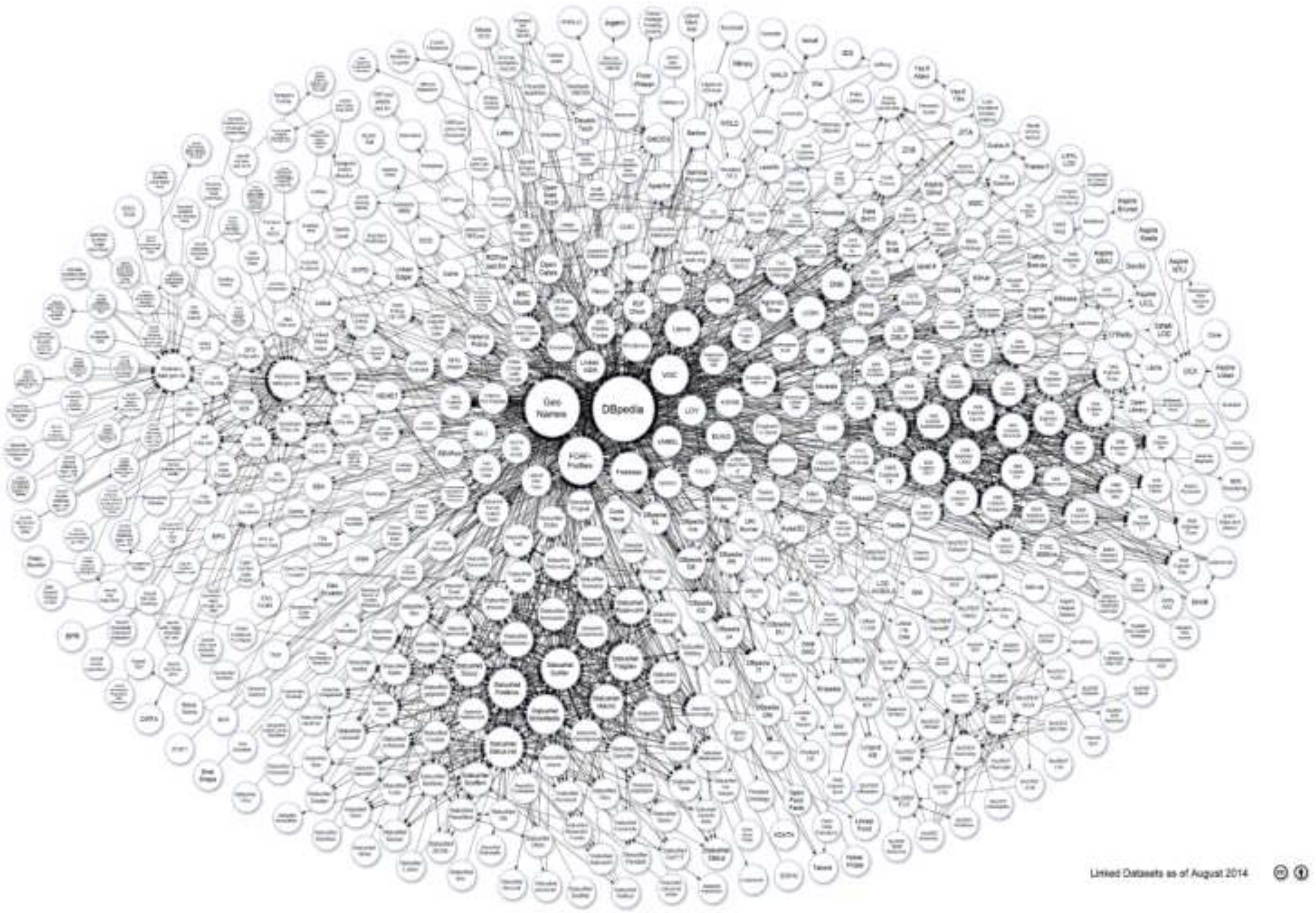


Linked open data (2009)

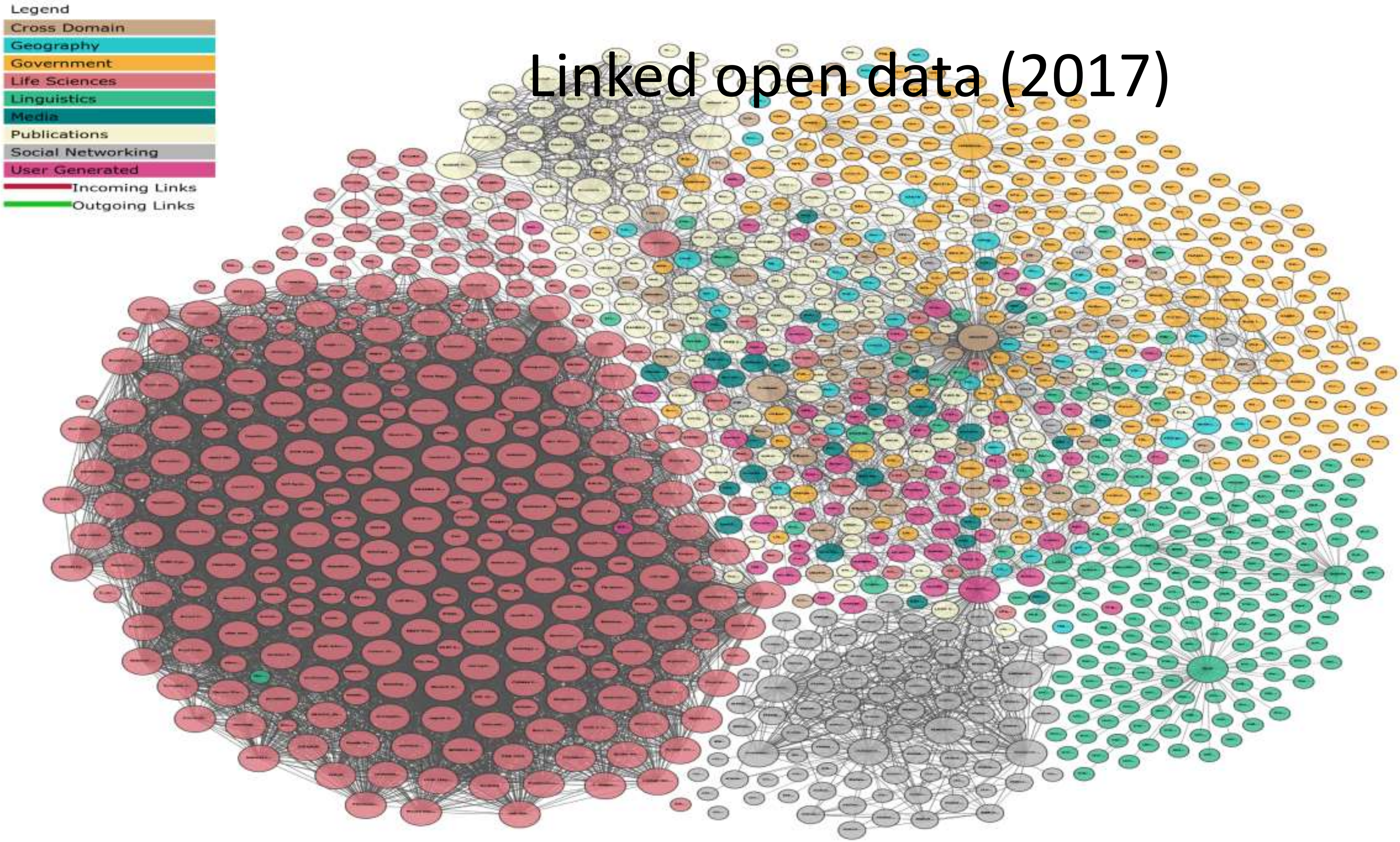


As of July 2009

Linked open data (2014)



Linked open data (2017)



Fuente: "Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. http://lod-cloud.net/"

RDF Data Model

Short history of RDF

RDF: Resource Description Framework

Around 1997 - PICS, Dublin core, Meta Content Framework

1997 1st Working draft https://www.w3.org/TR/WD-rdf-syntax-971002

- RDF/XML

1999 1st W3C Rec https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

- XML Syntax, first applications RSS, EARL

2004 - RDF Revised https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

- Emergence of SPARQL, Turtle, Linked Data

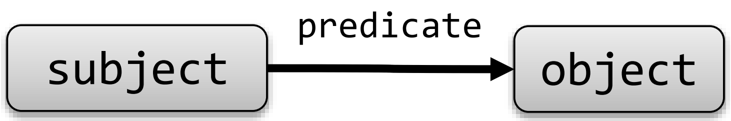
2014 - RDF 1.1 https://www.w3.org/TR/rdf11-concepts/

- SPARQL 1.1, JSON-LD

2017 - RDF validation: SHACL/ShEx

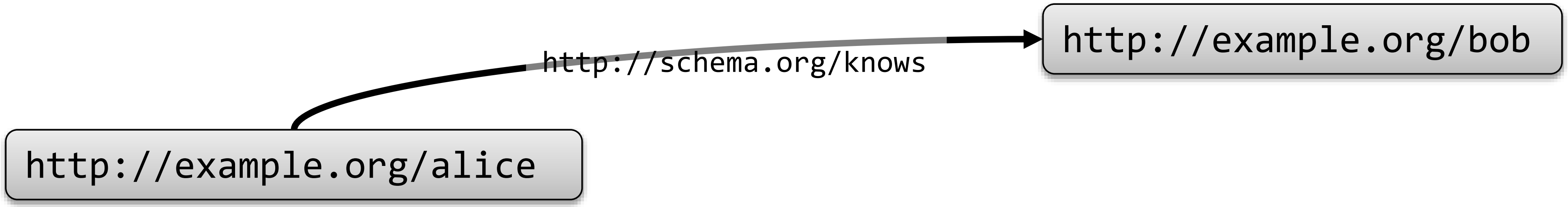
RDF Data Model

RDF is made from statements



Statment = a triple (subject, predicate, object)

Example:



subject predicate object

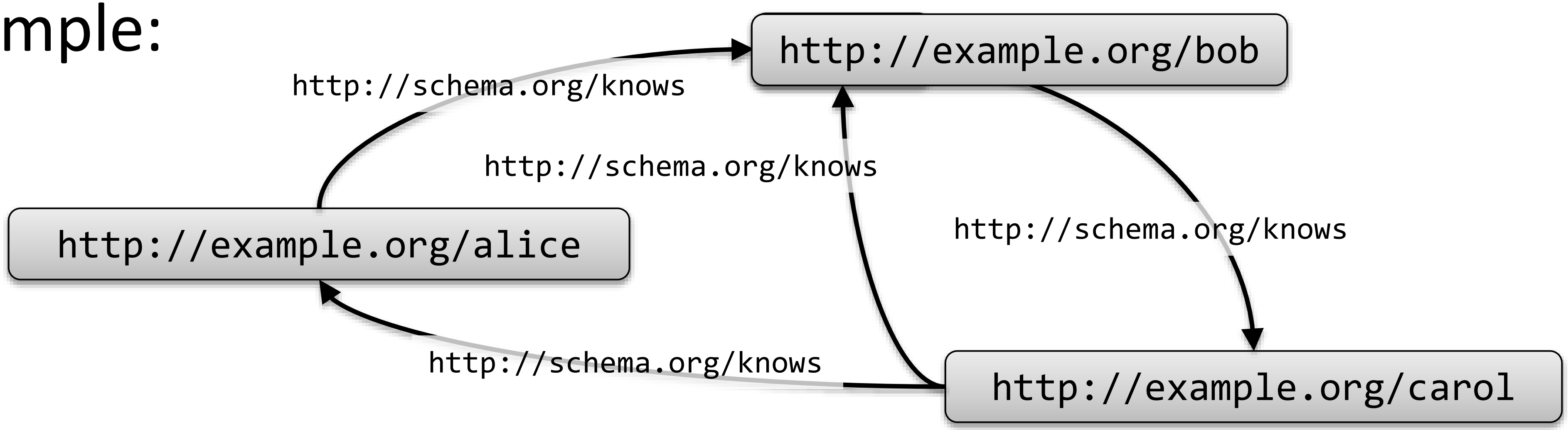
N-Triples representation

```
<http://example.org/alice> <http://schema.org/knows> <http://example.org/bob> .
```

Set of statements = RDF graph

RDF data model = directed graph

Example:



N-triples representation

```
<http://example.org/alice>   <http://schema.org/knows>       <http://example.org/bob> .
<http://example.org/bob>     <http://schema.org/knows>       <http://example.org/carol> .
<http://example.org/carol>   <http://schema.org/knows>       <http://example.org/alice> .
<http://example.org/carol>   <http://schema.org/knows>       <http://example.org/bob> .
```

subject predicate object

Turtle notation

Human readable notation that simplifies N-Triples

Allows namespace declarations

N-Triples

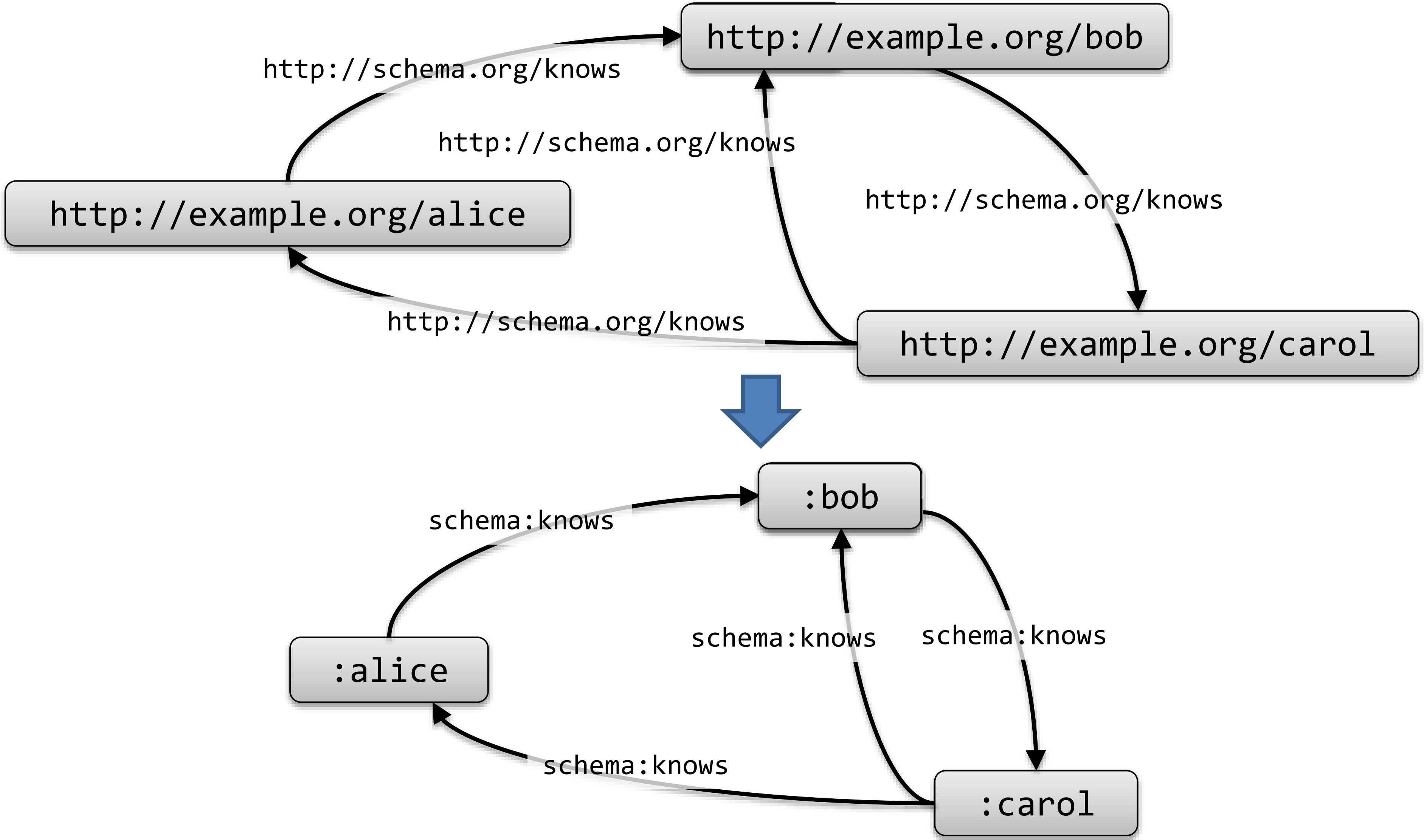
```
<http://example.org/alice>   <http://schema.org/knows>       <http://example.org/bob> .
<http://example.org/bob>     <http://schema.org/knows>       <http://example.org/carol> .
<http://example.org/carol>   <http://schema.org/knows>       <http://example.org/alice> .
<http://example.org/carol>   <http://schema.org/knows>       <http://example.org/bob> .
```

Turtle

```
prefix :       <http://example.org/>
prefix schema: <http://schema.org/>

:alice schema:knows :bob .
:bob   schema:knows :carol .
:carol schema:knows :bob .
:carol schema:knows :alice .
```

Note:
We will see later other Turtle simplifications

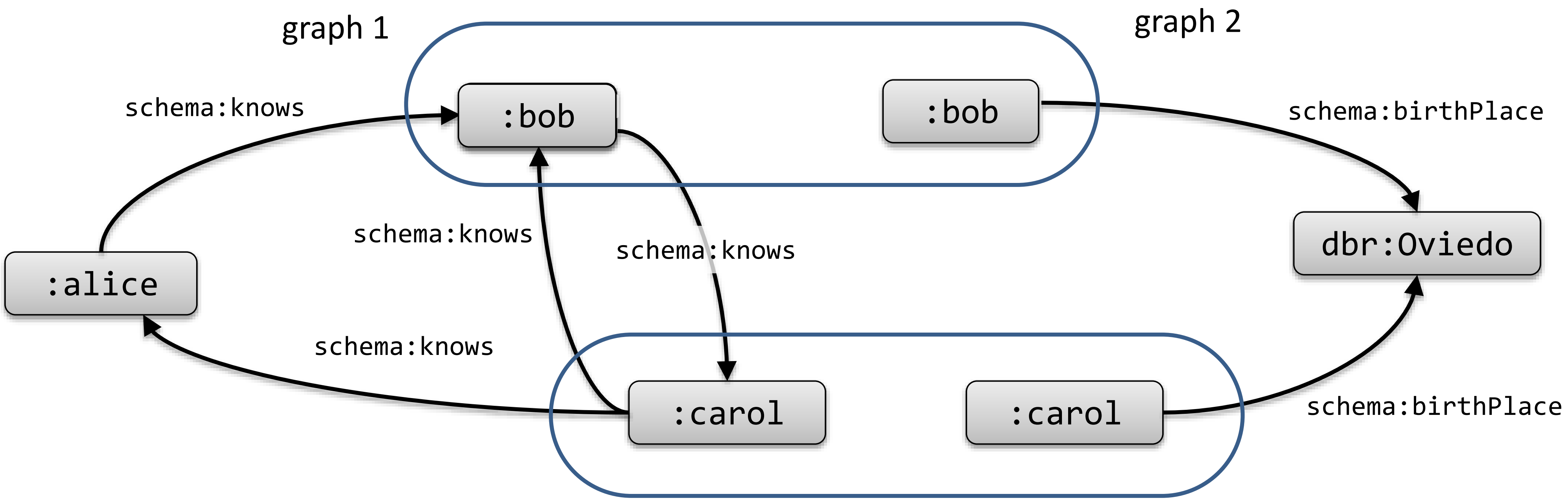
Namespaces simplification



RDF is compositional

RDF graphs can be merged to obtain a bigger graph

Automatic data integration

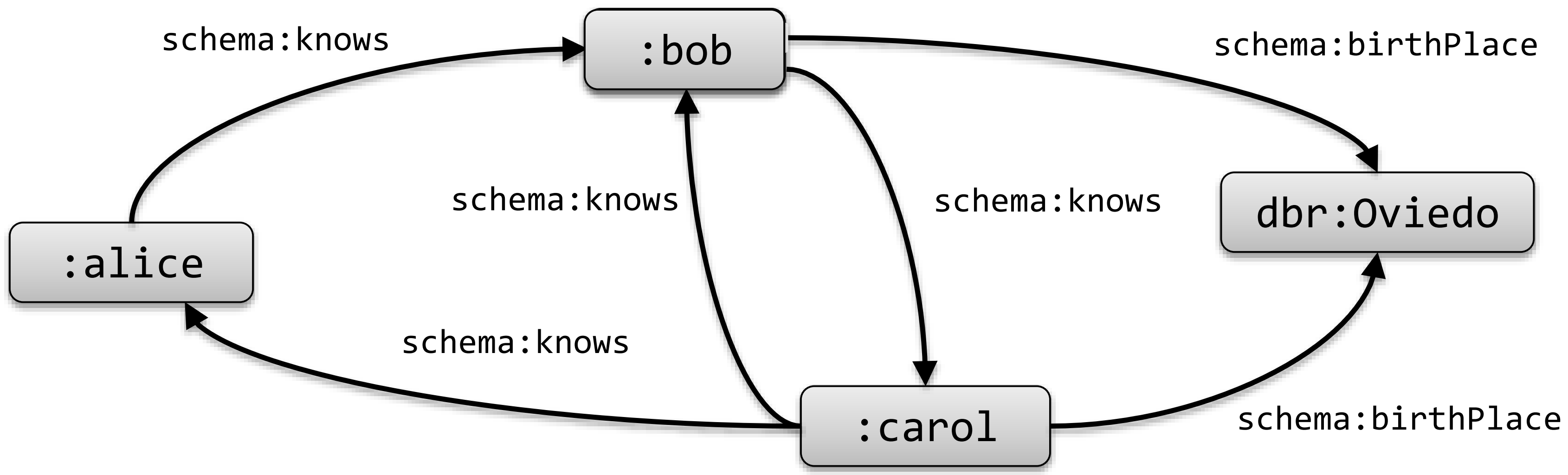


RDF is compositional

RDF graphs can be merged to obtain a bigger graph

Automatic data integration

graph 1 + graph 2



Turtle syntax

Some simplifications

prefix declarations

; when triples share the subject

```
:alice  schema:birthPlace  dbr:Oviedo .
:alice  schema:knows      :bob .
```

```
:alice  schema:birthPlace  dbr:Oviedo ;
        schema:knows       :bob .
```

, when triples share subject and object

```
:alice  schema:knows      :alice .
:alice  schema:knows      :bob .
```

```
:carol  schema:knows      :alice , :bob .
```

Turtle syntax

Exercise: simplify

```
prefix :      <http://example.org/>
prefix schema: <http://schema.org/>
prefix dbr:    <http://dbpedia.org/resource>


:alice schema:knows       :bob .
:bob   schema:knows       :carol .
:carol schema:knows       :bob .
:carol schema:knows       :alice .
:bob   schema:birthPlace dbr:Spain .
:carol schema:birthPlace dbr:Spain .
```

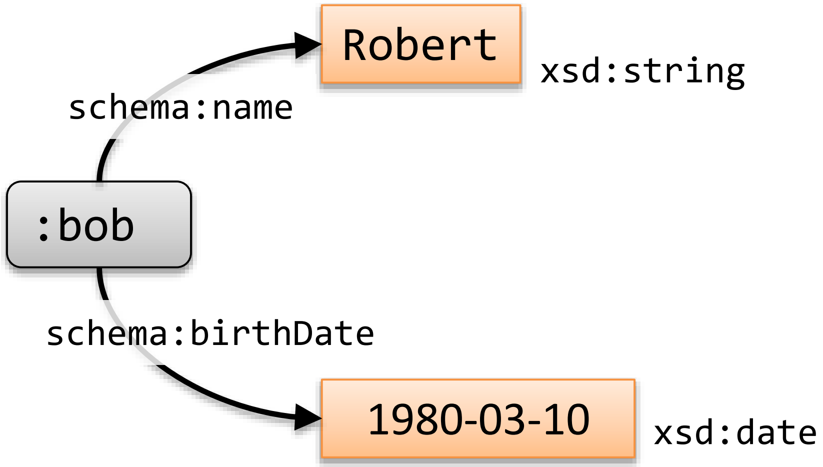
```
prefix ex:     <http://example.org/>
prefix schema: <http://schema.org/>
prefix dbr:    <http://dbpedia.org/resource>

:alice schema:knows       :bob , :carol.
:bob   schema:knows       :carol ;
       schema:birthPlace dbr:Spain .
:carol schema:knows       :bob, :alice ;
       schema:birthPlace dbr:Spain .
```

Try it: https://tinyurl.com/y9wbdycp

RDF Literals

Objects can also be literals

- Literals contain a lexical form and a datatype
- Typical datatypes = XML Schema primitive datatypes
  - If not specified, a literal has datatype `xsd:string`

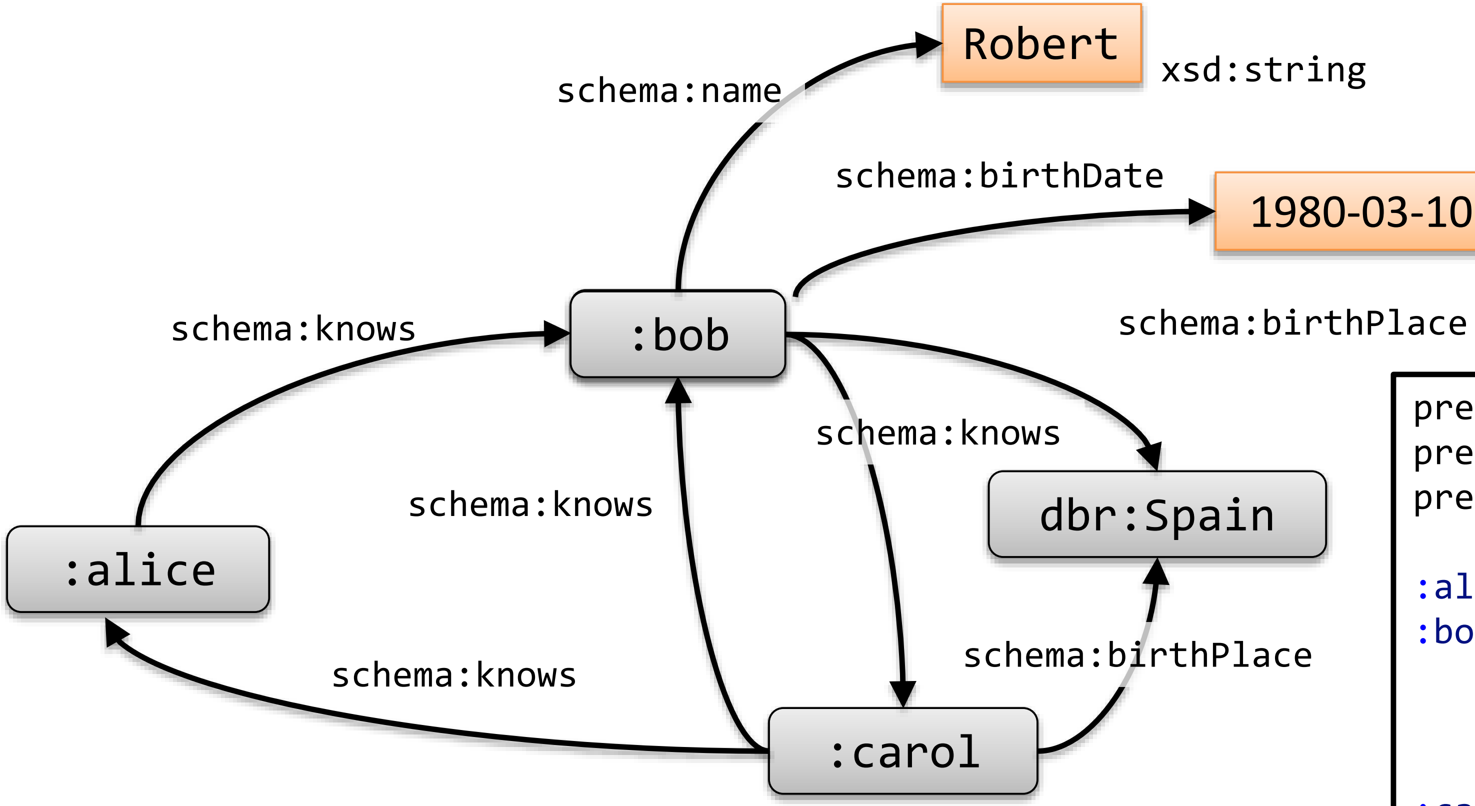


Turtle notation

```
:bob schema:name      "Robert" ;
:bob schema:birthDate "1980-03-10"^^<xsd:date>.
```

Remember...RDF is compositional

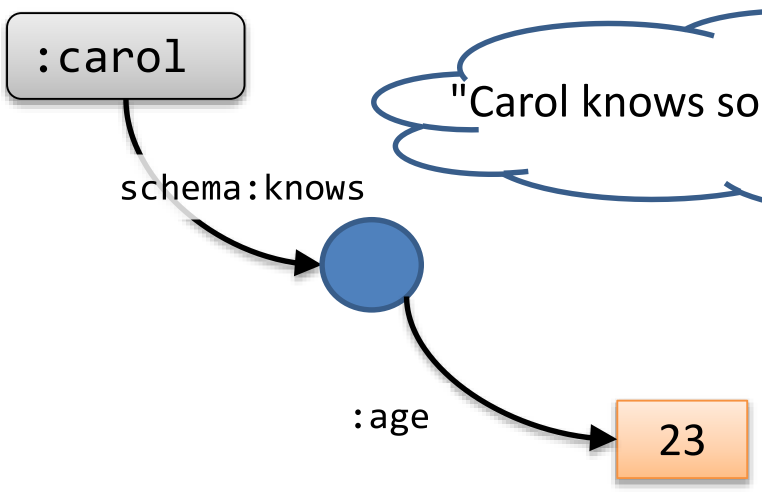
Merging previous data



```
prefix : <http://example.org/>
prefix schema: <http://schema.org/>
prefix dbr: <http://dbpedia.org/resource>

:alice schema:knows      :bob , :carol.
:bob   schema:knows      :carol ;
       schema:birthPlace dbr:Spain;
       schema:name       "Robert";
       schema:birthDate  "1980-03-10"^^<xsd:date>.
:carol schema:knows      :bob, :alice ;
       schema:birthPlace dbr:Spain .
```

Blank nodes

Subjects and objects can also be Blank nodes



"Carol knows someone whose age is 23"

Turtle notation with local identifier

```
:carol schema:knows _:x .
_:x    :age         23 .
```

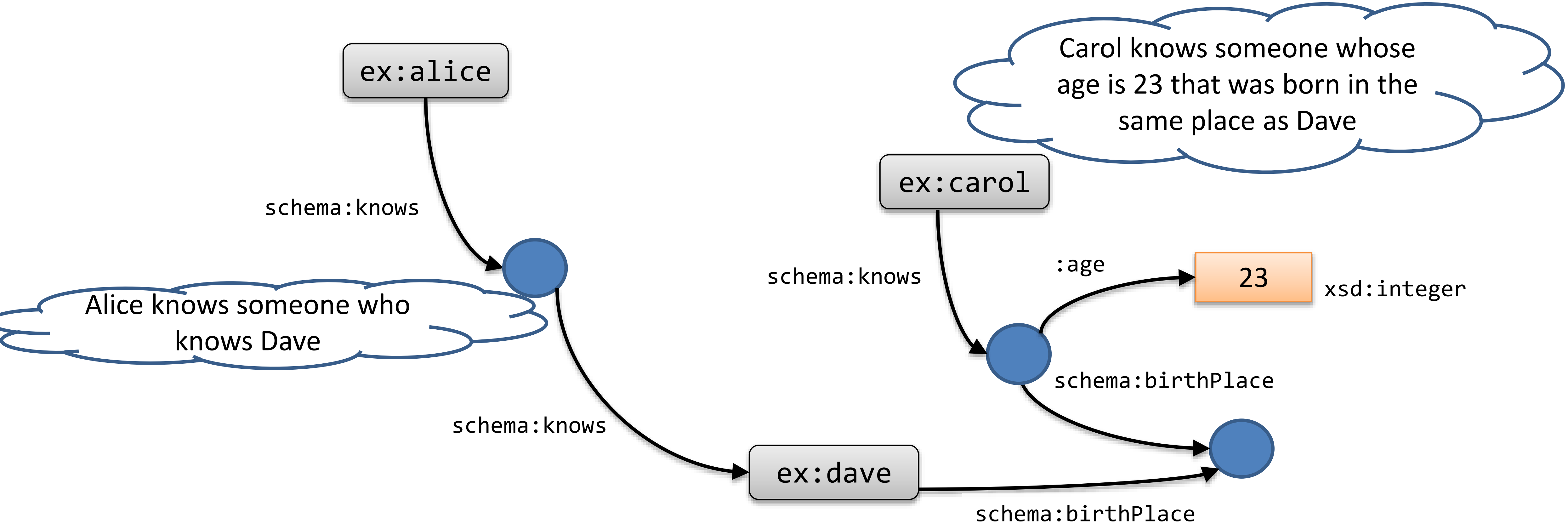
Turtle notation with square brackets

```
:carol schema:knows [ :age 23 ] .
```

Mathematical meaning:

$$\exists x(\texttt{schema:knows}(\texttt{:carol},x) \wedge \texttt{:age}(x, 23))$$

Blank nodes



Alice knows someone who knows Dave

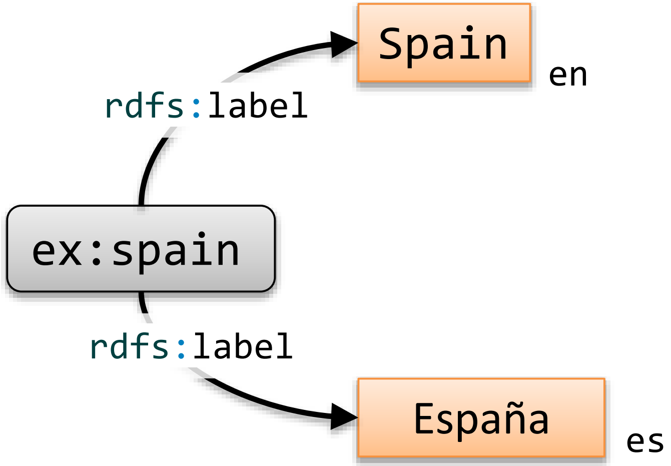
Carol knows someone whose age is 23 that was born in the same place as Dave

```
:alice schema:knows     [ schema:knows :dave ] .
:carol schema:knows     [ :age              23 ;
                          schema:birthPlace _:p ] .
:dave schema:birthPlace _:p .
```

Language tagged strings

String literals can be qualified by a language tag

- They have datatype `rdfs:langString`



Turtle notation

```
ex:spain rdfs:label "Spain"@en .
ex:spain rdfs:label "España"@es .
```

RDF data model

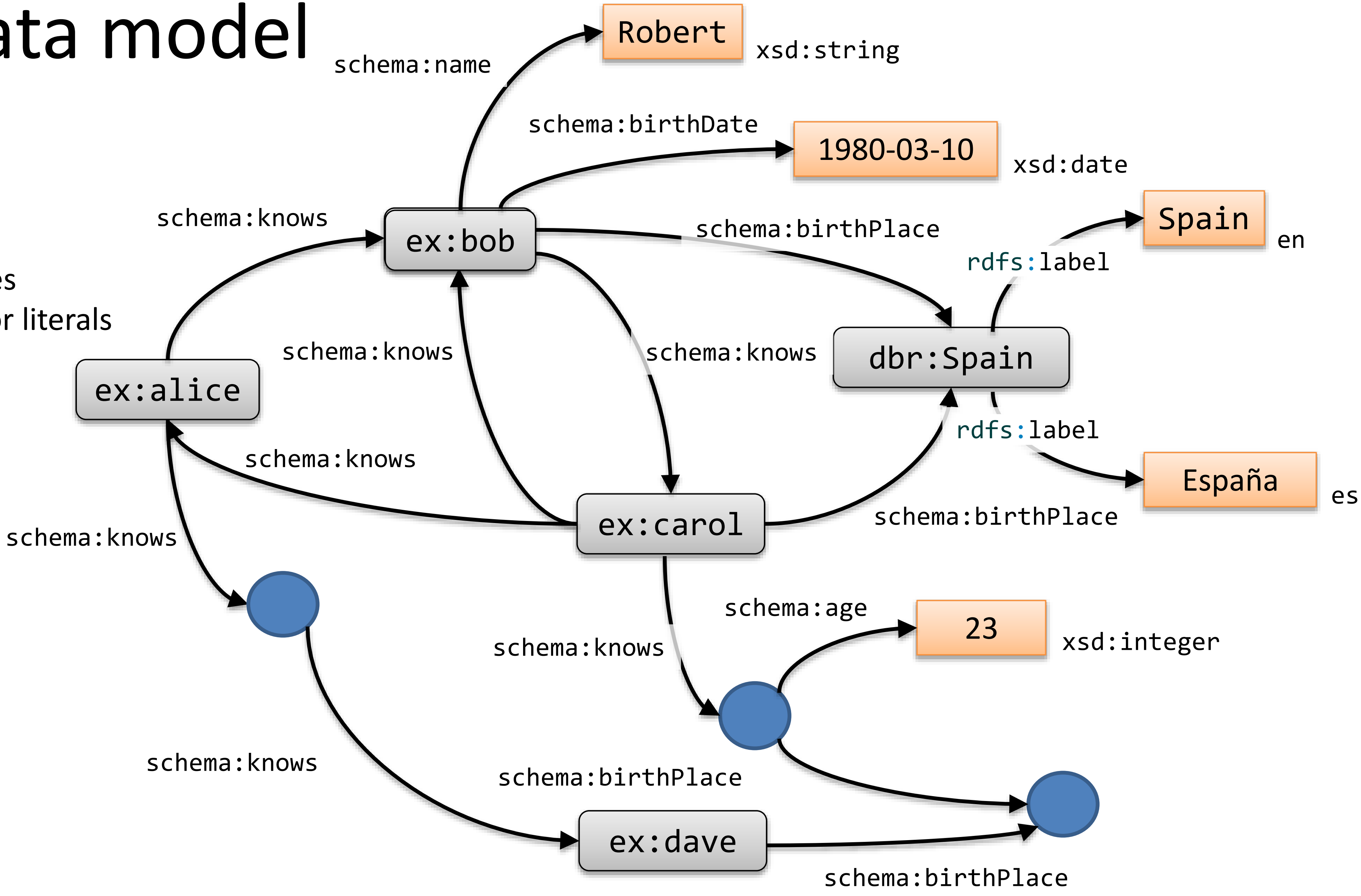
3 types of nodes

- ▭ URIs
- ● Blank nodes
- 🟧 Literals

Subjects: URIs or Blank nodes

Objects: URIs, Blank nodes or literals

Predicates always URIs



...and that's all about the RDF data model

The RDF Data model is very simple

Simple
is
better

RDF ecosystem

RDF Syntax

Shared entities and RDF vocabularies

Applications of RDF

Inference and ontologies

Query languages

RDF Validation

RDF syntax

First syntax based on XML: RDF/XML

N-Triples (enumerates all triples separated by dots)

Turtle (human readability)

JSON-LD

...other syntaxes...

....lots of syntaxes but a unique data model

RDF/XML

First syntax

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://example.org/"
         xmlns:schema="http://schema.org/">
<rdf:Description rdf:about="http://example.org/carol">
 <schema:knows>
  <rdf:Description rdf:about="http://example.org/bob">
   <schema:knows rdf:resource="http://example.org/carol"/>
   <schema:name>Robert</schema:name>
   <schema:birthDate rdf:datatype="xsd:date">1980-03-10</schema:birthDate>
  </rdf:Description>
 </schema:knows>
 <schema:knows>
  <rdf:Description rdf:about="http://example.org/alice">
   <schema:knows rdf:resource="http://example.org/bob"/>
   <schema:knows rdf:resource="http://example.org/carol"/>
  </rdf:Description>
 </schema:knows>
 <schema:knows rdf:parseType="Resource">
  <age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">23</age>
 </schema:knows>
 </rdf:Description>
</rdf:RDF>
```

N-Triples

For testing and easy parsing

...just triples separated by dots

```
<http://example.org/carol> <http://schema.org/knows>     <http://example.org/bob> .
<http://example.org/carol> <http://schema.org/knows>     <http://example.org/alice> .
<http://example.org/carol> <http://schema.org/knows>     _:x .
_:x                        <http://example.org/age>      "23"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://example.org/alice> <http://schema.org/knows>     <http://example.org/bob> .
<http://example.org/alice> <http://schema.org/knows>     <http://example.org/carol> .
<http://example.org/bob>   <http://schema.org/knows>     <http://example.org/carol> .
<http://example.org/bob>   <http://schema.org/name>      "Robert" .
<http://example.org/bob>   <http://schema.org/birthDate> "1980-03-10"^^<xsd:date> .
```

Turtle

Designed to be human-readable

```
prefix :       <http://example.org/>
prefix schema: <http://schema.org/>

:alice schema:knows :bob , :carol .
:bob   schema:knows :carol ;
       schema:name "Robert";
       schema:birthDate "1980-03-10"^^<xsd:date>.
:carol schema:knows :bob, :alice ;
       schema:knows [ :age 23 ] .
```

JSON-LD

Json for linked data

```
{
"@context" : {
"knows" : { "@id" : "http://schema.org/knows", "@type" : "@id" },
"age" : { "@id" : "http://example.org/age", "@type" : "http://www.w3.org/2001/XMLSchema#integer" },
"name" : { "@id" : "http://schema.org/name" },
"birthDate" : { "@id" : "http://schema.org/birthDate", "@type" : "xsd:date" },
"@vocab" : "http://example.org/",
"schema" : "http://schema.org/"
},
"@graph" : [
 { "@id" : "http://example.org/alice",
  "knows" : [ "http://example.org/bob", "http://example.org/carol" ] },
 { "@id" : "http://example.org/bob",
  "birthDate" : "1980-03-10",
  "knows" : "http://example.org/carol",
  "name" : "Robert" },
 { "@id" : "http://example.org/carol",
  "knows" : [ "http://example.org/bob", "http://example.org/alice", "_:x" ] },
 { "@id" : "_:x",
  "http://example.org/age" : 23 }
 ]
}
```

Other Turtle simplifications

RDF type property

Numbers

Collections

RDF type property

The rdf:type property declares the type of a resource

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix schema: <http://schema.org/> .

e:alice rdf:type schema:Person .
e:bob   rdf:type schema:Person .
```

rdf:type can be simplified as a

```
@prefix schema: <http://schema.org/> .

:alice a schema:Person .
:bob   a schema:Person .
```

Constants

Numbers and boolean values can be represented without quotes

- They are parsed as XML Schema datatypes

Datatype	Shorthand example	Lexical example
`xsd:integer`	`3`	`"3"^^xsd:integer`
`xsd:decimal`	`-3.14`	`"true"^^xsd:decimal`
`xsd:double`	`3.14e2`	`"true"^^xsd:double`
`xsd:boolean`	`true`	`"true"^^xsd:boolean`

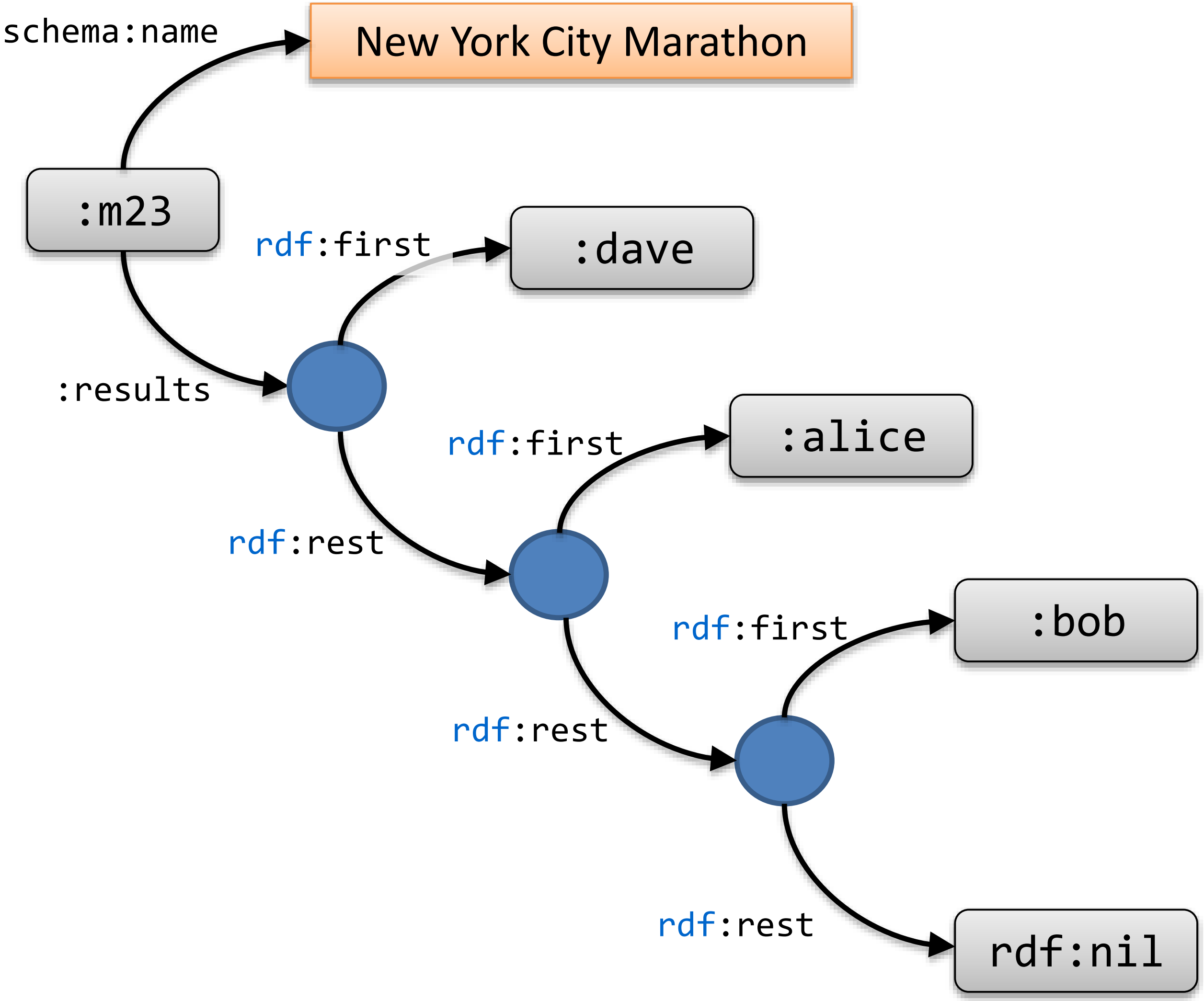
Collections

Ordered lists

```
:m23 schema:name "New York City Marathon ";
     :results ( :dave :alice :bob ) .
```

Internally, represented as linked lists

```
:m23 schema:name "New York City Marathon ";
:results _:1 .
_:1 rdf:first :dave ;
    rdf:next _:2 .
_:2 rdf:first :alice ;
    rdf:next _:3 .
_:3 rdf:first :bob ;
    rdf:next rdf:nil .
```



Shared entities and vocabularies

The use of URIs instead of plain strings facilitates:

- Merging data from heterogeneous sources
- Avoid ambiguity

Challenge: Agreeing on common entities and properties

Appearance of some popular vocabularies:

- schema.org: Joint effort from Google, Yahoo, Microsoft, Yandex
- Linked open vocabularies Project: http://lov.okfn.org/

Some popular vocabularies and namespaces

Alias	URL	Name	Some properties
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF	type, subject, predicate, object,…
rdfs:	http://www.w3.org/2000/01/rdf-schema#	RDF Schema	domain, range Class, Property subClassOf,…
owl:	http://www.w3.org/2002/07/owl#	OWL Ontologías	sameAs, intersectionOf unionOf, …
dc:	http://purl.org/dc/elements/1.1/	Dublin Core	author, date, creator, …
schema	http://schema.org/	Schema.org	name, knows, etc.
skos:	http://www.w3.org/2008/05/skos#	SKOS Simple Knowledge Organization System	broader, narrower,

Service http://prefix.cc can be used to find the most popular prefix for some URI

Applications of RDF

First applications

RDF & HTML: RDFa, Microdata

RDF to represent knowledge

RDF as an internal database

Linked data

First RDF applications

- Some initiatives proposed by W3C
  - RSS 1.0 was proposed with an RDF/XML based syntax
    - Other XML based versions were available
  - EARL: Evaluation and Reporting Language
- RDF/XML adoption was not popular

RDF & HTML

Possibilities

One resource for HTML and another for metadata in RDF

RDFa: Use HTML attributes to encode RDF triples

Microdata: New HTML5 attributes can encode metadata

RDFa

```
<p vocab="http://schema.org/"
   typeof="Book"
   about="http://example.org/book1">
The book
 <span property="name">The Spring</span> by
 <span property="author">Cervantes</span>
 was published
 <span property="datePublished"
       content="2014-05-04">
  last Saturday</span>.
</p>
```

Microdata

```
<p itemscope
   itemid="http://leer.com/libro123"
   itemtype="http://schema.org/Book">
The book
 <span itemprop="name">The Spring</span> by
 <span itemprop="author">Cervantes</span>
 was published
   <time itemprop="datePublished"
         content="2014-05-04">
  last saturday</time>.
</p>
```

RDF to represent knowledge

- Freebase: developed by Metaweb (2005)
  - Open, shared database of world's knowledge
  - Acquired by Google in 2010. It is the basis of Google knowledge graph
- DBpedia (http://dbpedia.org)
  - Extracts knowledge from Wikipedia and converts it to RDF
- Wikidata (http://wikidata.org/)
  - Free knowledge base edited collaboratively
  - Developed by Wikimedia foundation

RDF as an internal database

Specialized RDF databases (triplestores)

RDF = very flexible, easy to adapt to domain changes

Several big companies are using RDF internally

- Example: BBC, Europeana

RDF for Linked open data

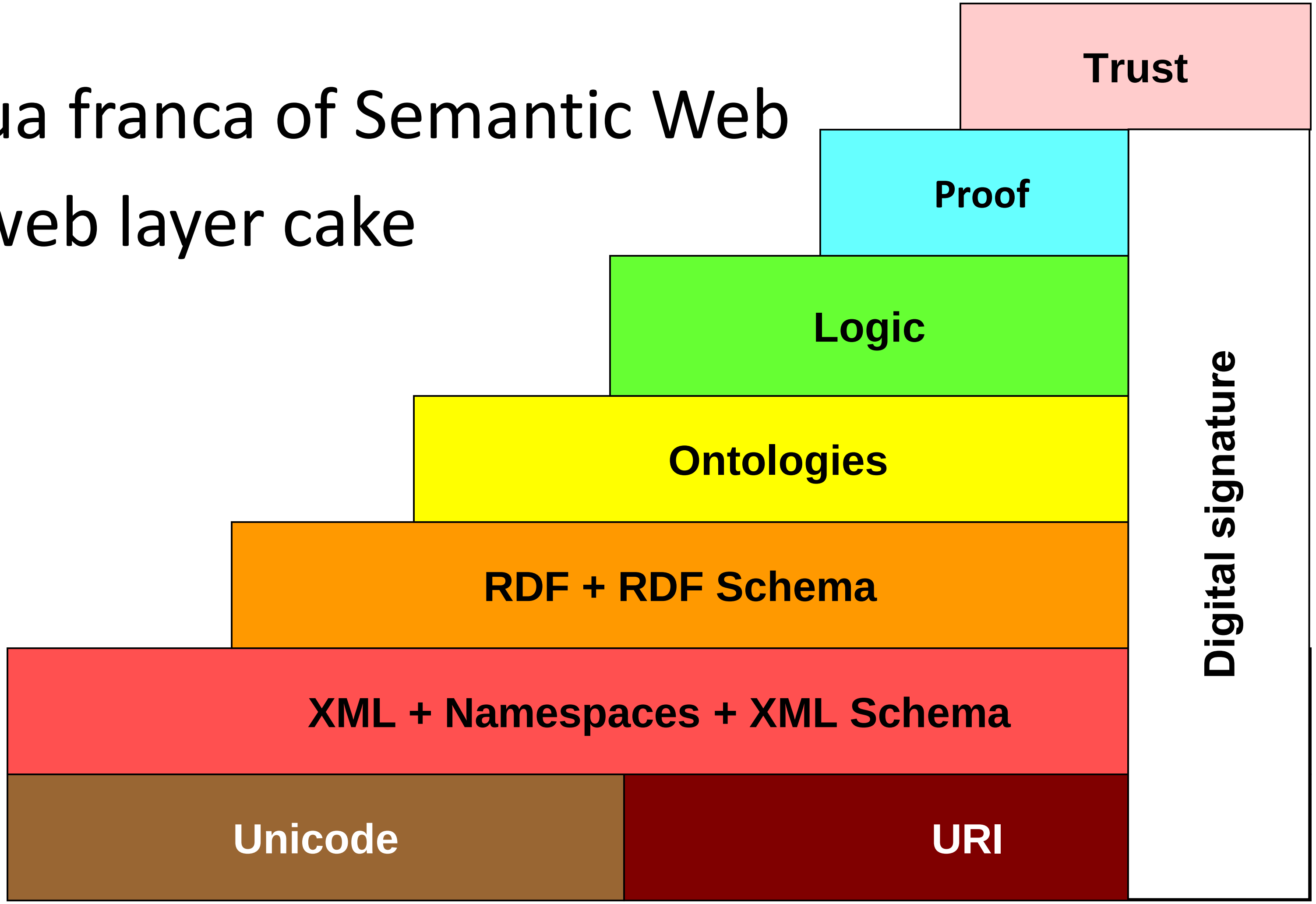
Principles proposed by Tim Berners-Lee to publish data:

1. Use URIs to denote things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

RDF as the basis of Semantic Web

RDF = lingua franca of Semantic Web

Semantic web layer cake

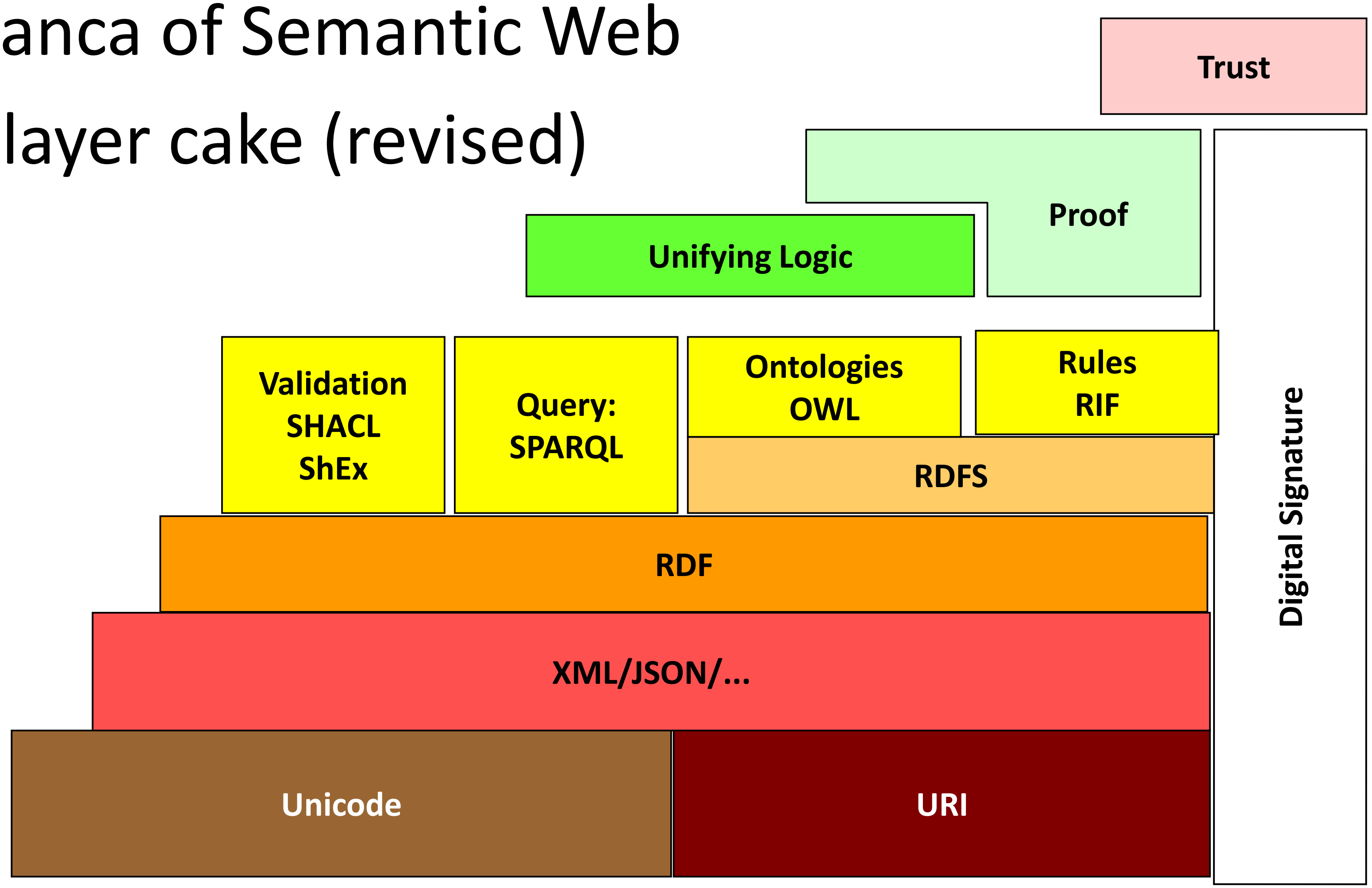


First version proposed by Tim Berners Lee, year 2000
http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

RDF as the basis of Semantic Web

RDF = lingua franca of Semantic Web

Semantic web layer cake (revised)



Trust

Proof

Unifying Logic

Validation
SHACL
ShEx

Query:
SPARQL

Ontologies
OWL

Rules
RIF

RDFS

RDF

XML/JSON/...

Unicode

URI

Digital Signature

RDFS & inferences

RDFS

Originally RDF Schema (2000)

Defines a vocabulary for common concepts

Classes: `rdfs:Class`, `rdfs:Property`, `rdfs:Literal`

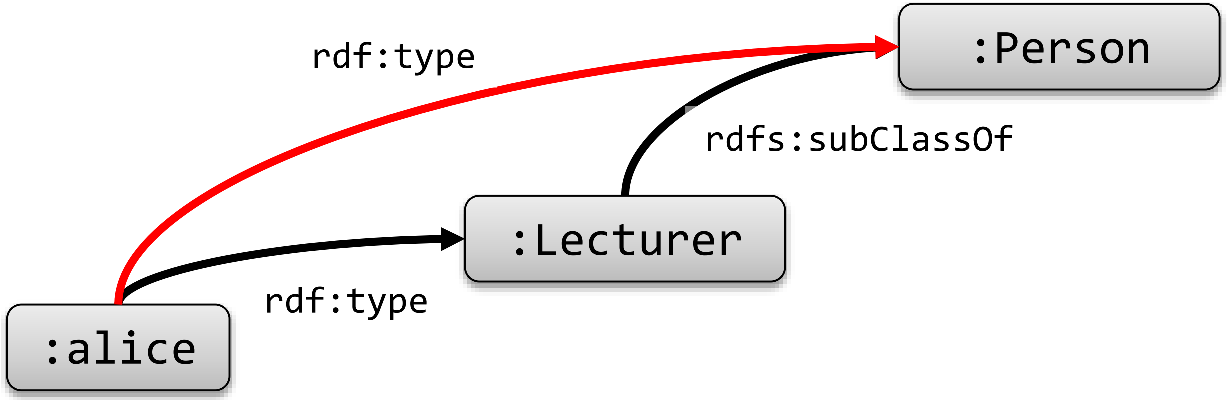
Properties: `rdfs:domain`, `rdfs:range`, `rdfs:subClassOf`, ...

RDFS

RDFS processors can infer new triples

RDFS defines several rules for inference:

IF x `rdf:type` A AND A `rdfs:subClassOf` B THEN x `rdf:type` B



OWL

Web Ontology Language.

First version (2004), OWL 2 (2009)

Based on description logics

Language to describe classes, individuals, relationships

OWL example

```
<> a owl:Ontology .

:Man a owl:Class ;
 owl:equivalentClass [
  owl:intersectionOf (:Person
   [ a owl:Restriction ;
     owl:onProperty schema:gender ; owl:hasValue schema:Male
   ] )
 ] .
:Woman a owl:Class ;
 owl:equivalentClass [
  owl:intersectionOf ( :Person
   [ a owl:Restriction ;
    owl:onProperty schema:gender ; owl:hasValue schema:Female
   ] )
 ] .
[ a owl:AllDisjointClasses ; owl:members ( :Woman :Man ) ] .

:Person owl:equivalentClass [ rdf:type owl:Class ;
  owl:unionOf ( :Woman :Man )
] .
```

Instance data

```
:alice a :Woman ;
  schema:gender schema:Female .

:bob a :Man .
```

Inferred data

```
:alice a :Person .
:bob a :Person .
:bob schema:gender schema:Male .
```

OWL

OWL can been used to describe domain ontologies

Different kinds of ontologies:

- Upper level ontologies (SUMO, WordNet, ...)
- Domain specific (example: SNOMED)

Tools to edit ontologies: Protégé editor