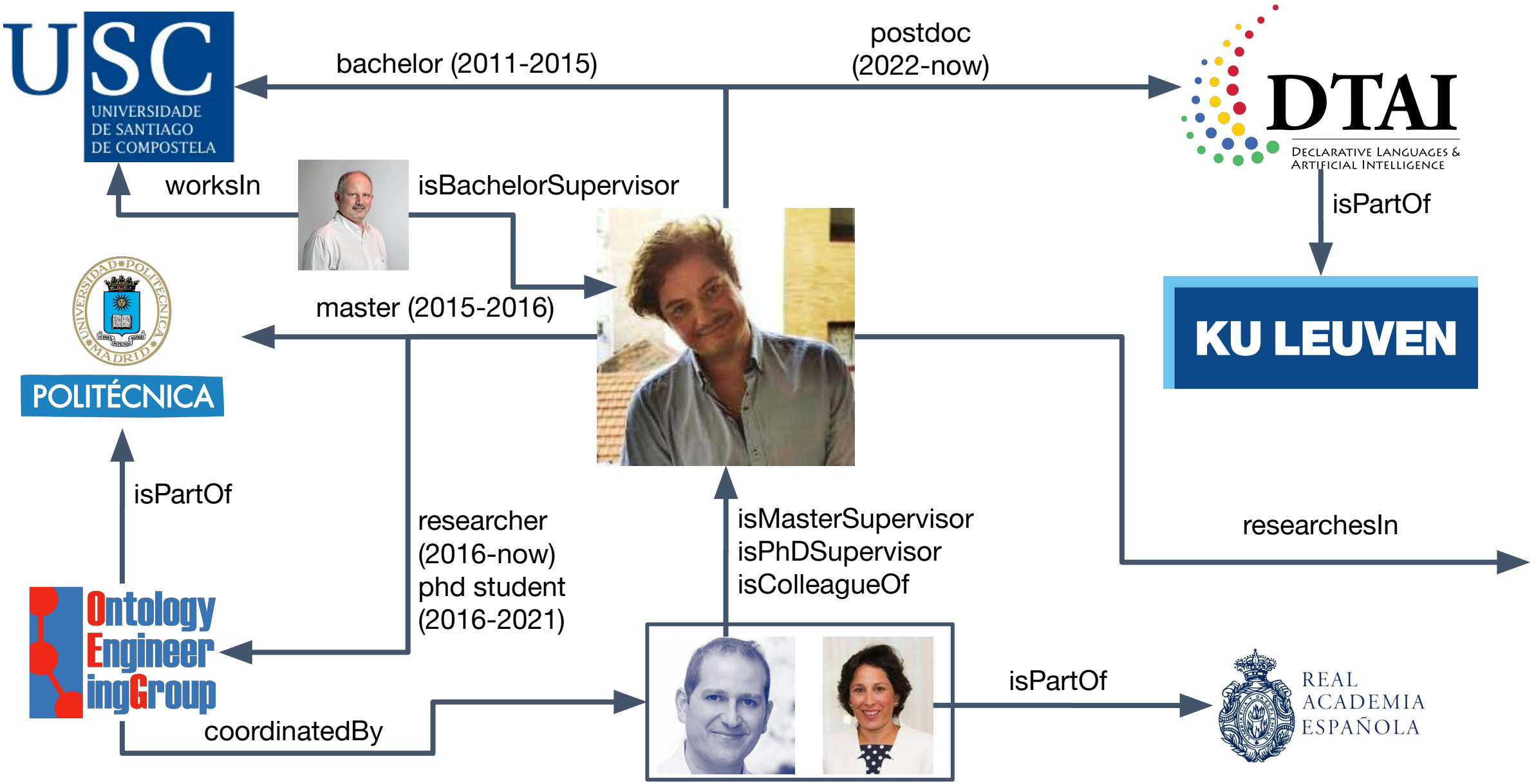
Knowledge Graphs and Ontologies

A Practical Introduction to the Semantic Web

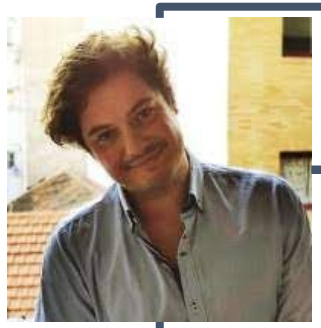
David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

david.chaves@upm.es dchavesf 08/11/2022 USC-CiTIUS

Why am I here?


USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
bachelor (2011-2015)
postdoc (2022-now)
DTAI
Declarative Languages & Artificial Intelligence
worksIn
isBachelorSupervisor
isPartOf
KU LEUVEN
master (2015-2016)
POLITÉCNICA
isPartOf
researcher (2016-now) phd student (2016-2021)
isMasterSupervisor
isPhDSupervisor
isColleagueOf
researchesIn
Ontology Engineering Group
coordinatedBy
isPartOf
REAL ACADEMIA ESPAÑOLA

Who am I?



researchesIn

Semantic Web
Semantic Data Integration
Knowledge Graphs
Declarative Mapping Rules

PhDThesis (June 2021)

Knowledge Graph Construction from Heterogeneous Data Sources exploiting Declarative Mapping Rules

chairOf (2019-now)

Knowledge Graph Construction
W3C Community Group

gaveTutorialAt

organizes (2019-now)

KGC Workshop

SEM4TRA

coordinates (2018-2021)

Open Summer of Code

mainAuthorOf

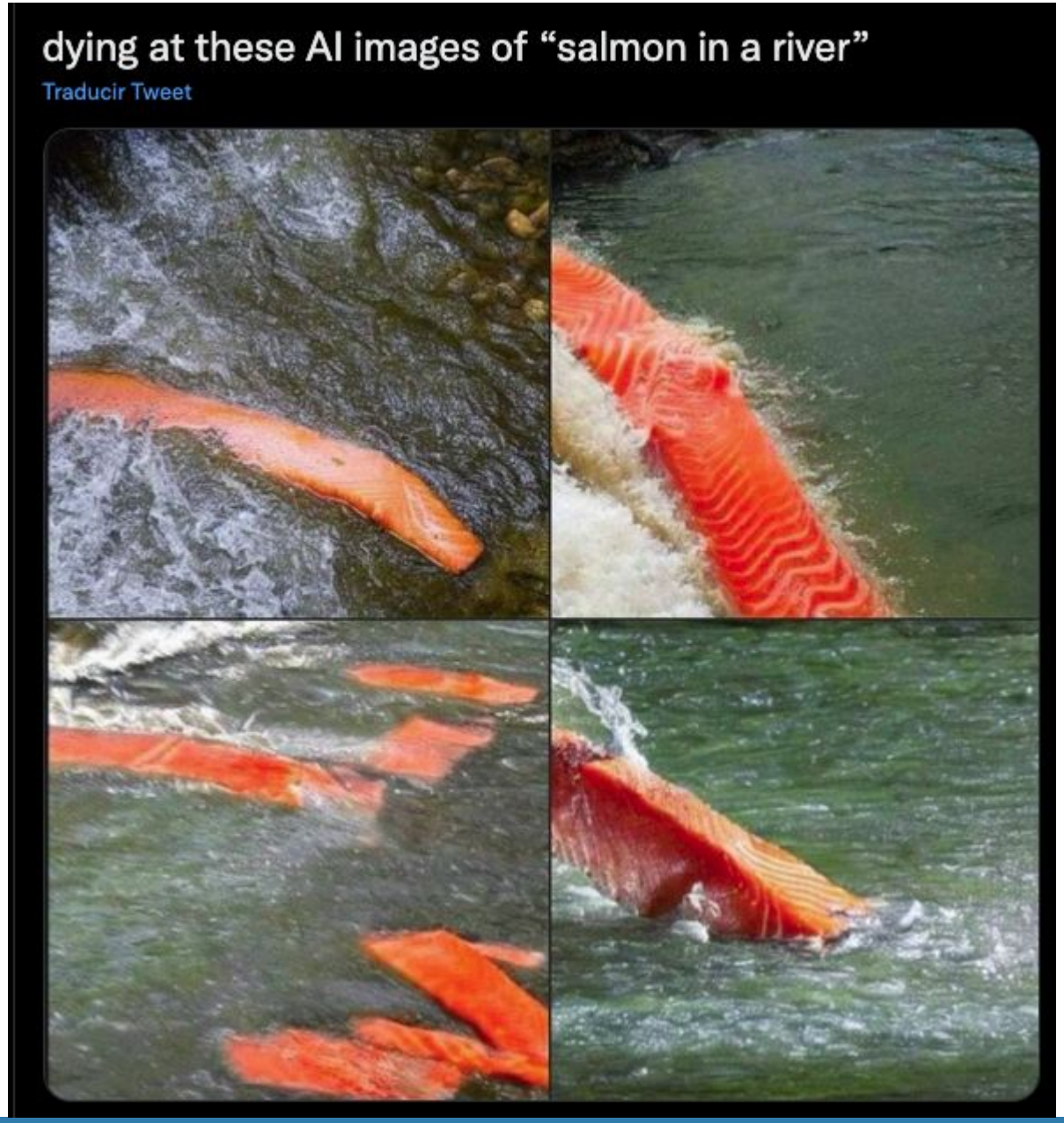
datos.gob.es
reutiliza la información pública

ESWC19

ISWC 2020
NOVEMBER 1-6

ESWC22

The problem of the salmon in a river



http://dbpedia.org/resource/Salmon
http://dbpedia.org/resource/Salmon_as_food

License

- This work is licensed under the license CC BY-NC-SA 4.0 International
  - http://purl.org/NET/rdflicense/cc-by-nc-sa4.0

- You are free:
  - to Share — to copy, distribute and transmit the work
  - to Remix — to adapt the work

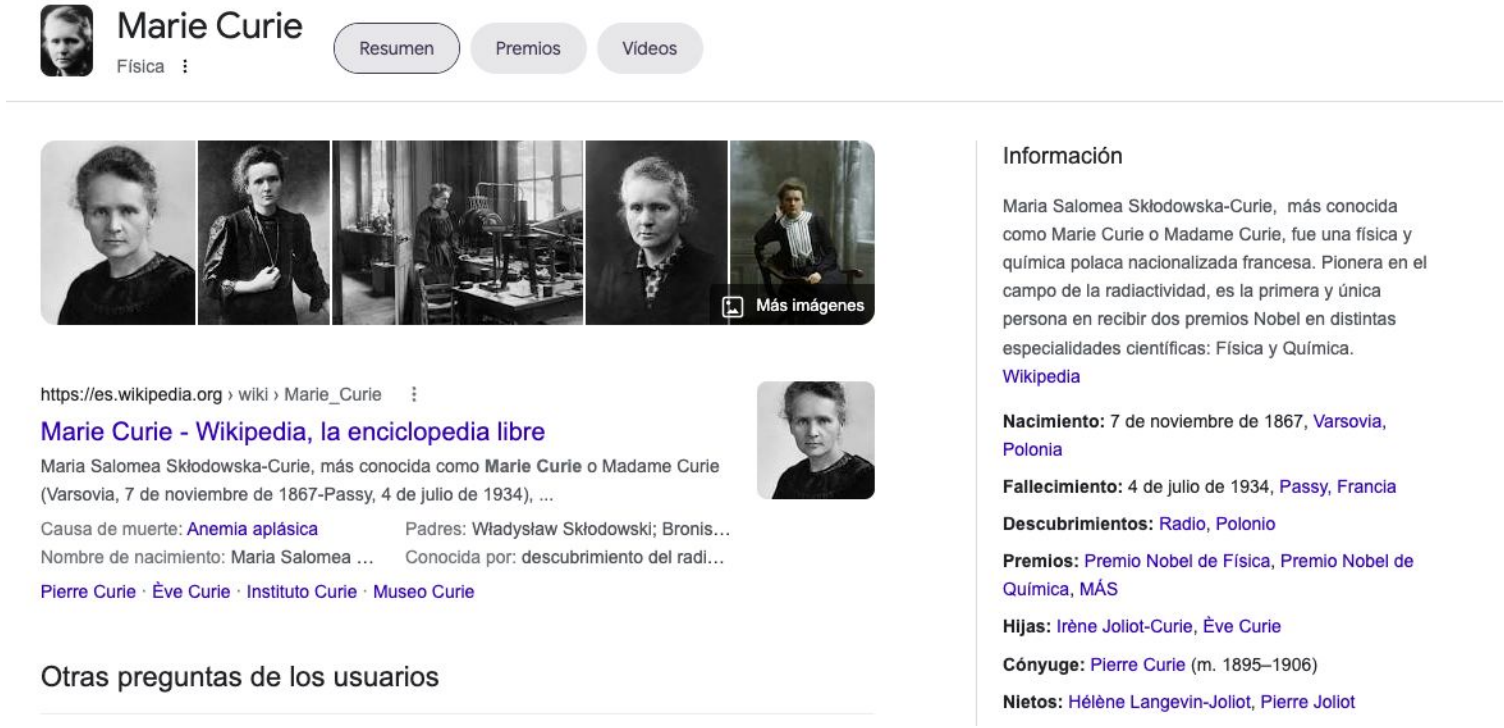
- Under the following conditions
  - Non-commercial – You cannot use it for commercial purposes, nor for training inside a commercial company
  - Attribution — You must attribute the work by inserting
    - a credits slide stating: These slides are partially based on “Knowledge Graphs
    - and Ontologies: A Practical Introduction to the SW” by David Chaves-Fraga
  - Share-alike

These slides are partially based on:

- “Semantic Web, Linked Data and Knowledge Graphs” course by Oscar Corcho, Raúl García-Castro, Daniel Garijo, David Chaves-Fraga, Paola Espinoza (Universidad Politécnica de Madrid)
- “Data and Knowledge Graphs” course Anastasia Dimou, David Chaves-Fraga (KULeuven)

Table of Contents

- When do we start talking about Knowledge Graphs?
  - Google's announcement of the Google Knowledge Graph in 2012
    - https://www.blog.google/products/search/introducing-knowledge-graph-things-not/
    - "Things, not strings"
  - Followed by Airbnb, Amazon, eBay, Facebook, IBM, LinkedIn, Microsoft, Uber, etc.



facebook

Google

LinkedIn

ontotext

SEMANTIC WEB COMPANY

aws

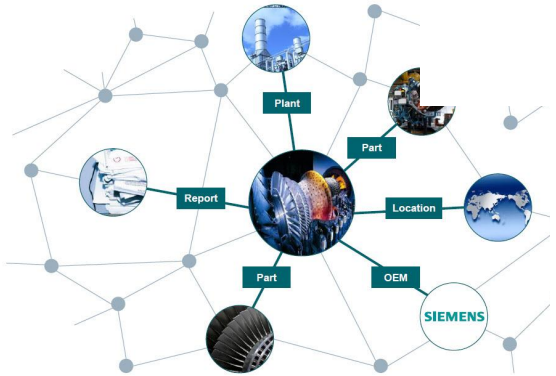
Stardog

SANOF

Why (Knowledge) Graphs?

Benefits of using knowledge graphs for data representation

- The world is entities and relations!
- Intelligible domain model instead of complex (physical) data model
- Schema-on-read instead of complex schema migration for extensions
- Easy integration of multiple data sources (schemas) and types (structured, unstructured, …)
- Formal semantic representation enables inference and machine processing



MPIWG
Max Planck Institute for the History of Science

THE BRITISH MUSEUM

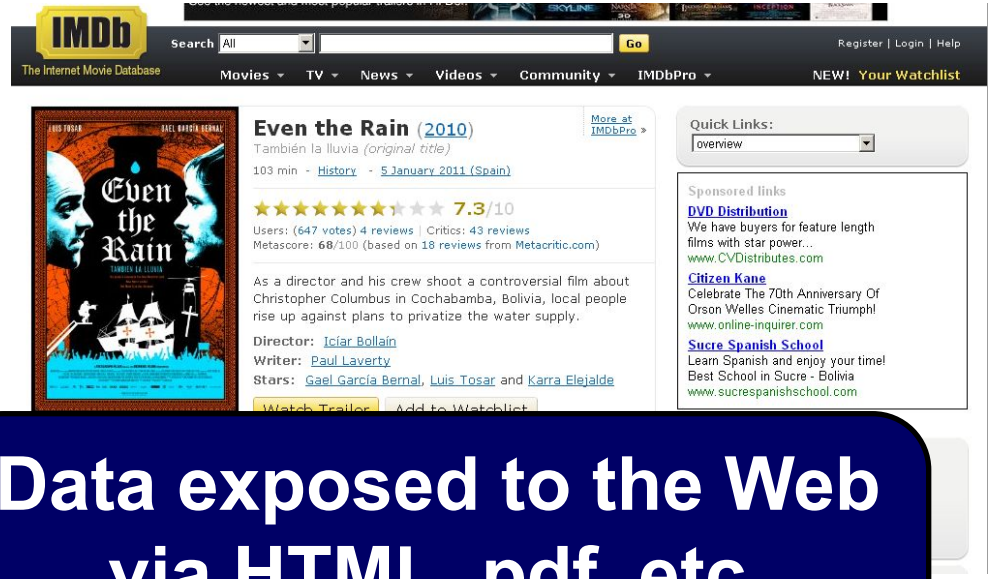
poolparty

ebay

Uber

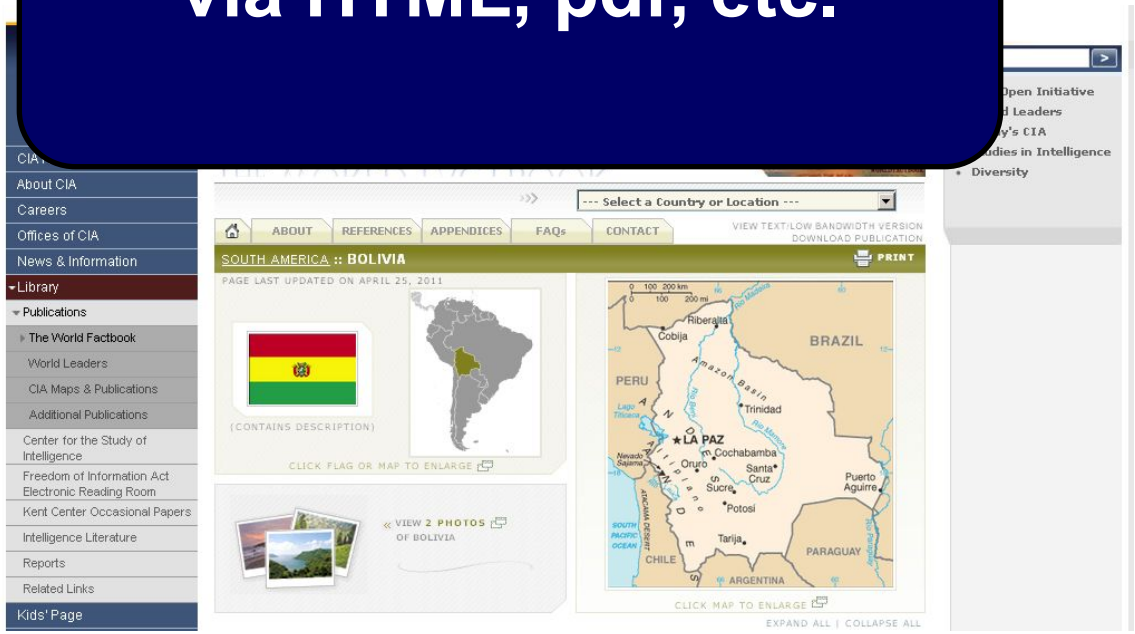
Adapted from Mariano Rodriguez Muro from KGB2019 Keynote ESWC2019

MovieDB



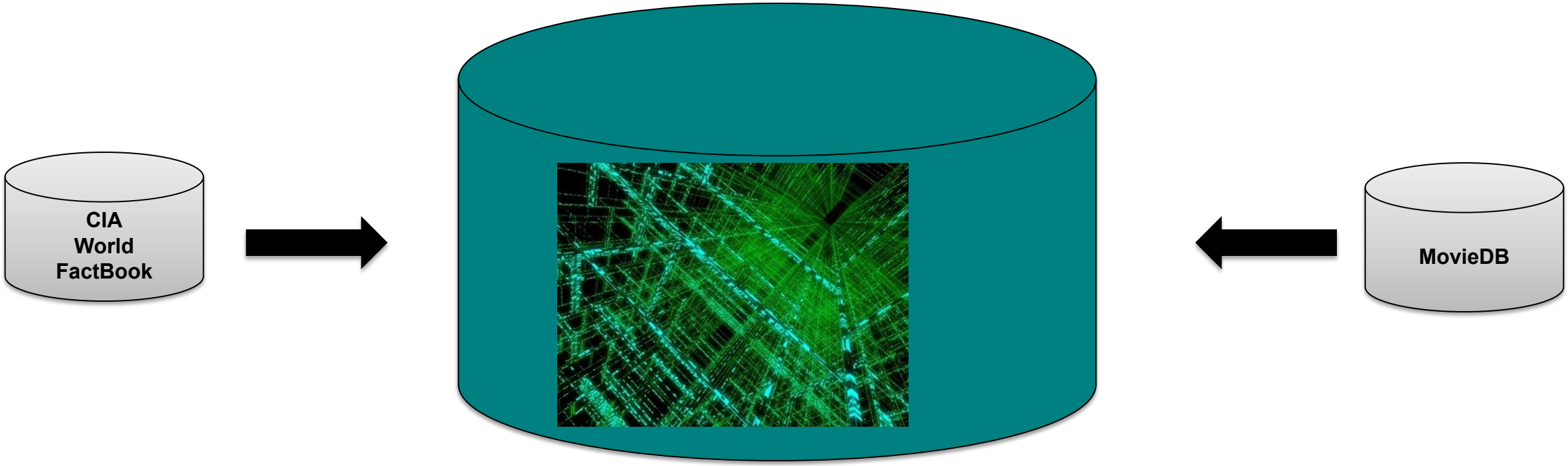
Data exposed to the Web via HTML, pdf, etc.

CIA World FactBook



© Slide adapted from "5min Introduction to Linked Data"- Olaf Hartig

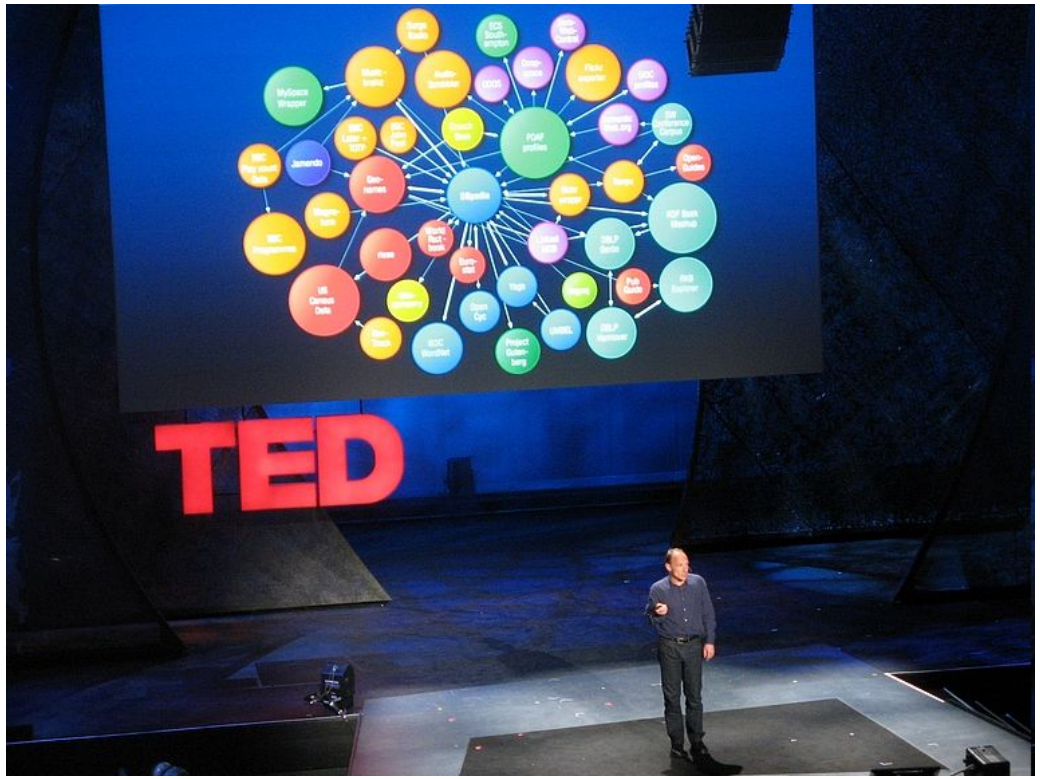
- Use the Web like a single global database
  - Move from a Web of documents to a **Web of Data**



Linked Data: Four principles (Tim Berners Lee, 2006)

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

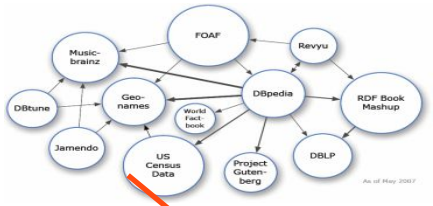
http://www.w3.org/DesignIssues/LinkedData.html



https://youtu.be/OM6XIICm_qo

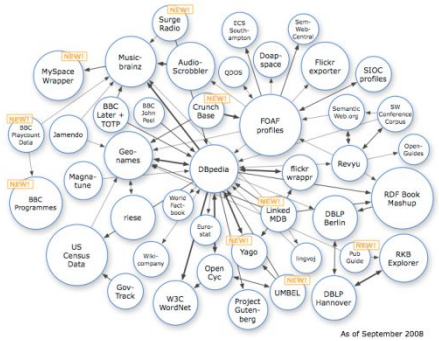
http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html

Linked Open Data evolution

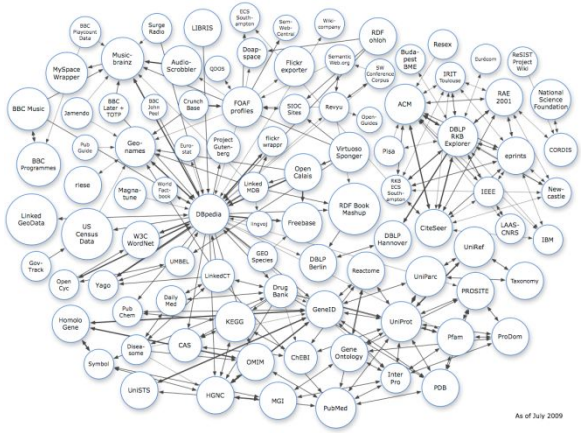


2007

2008

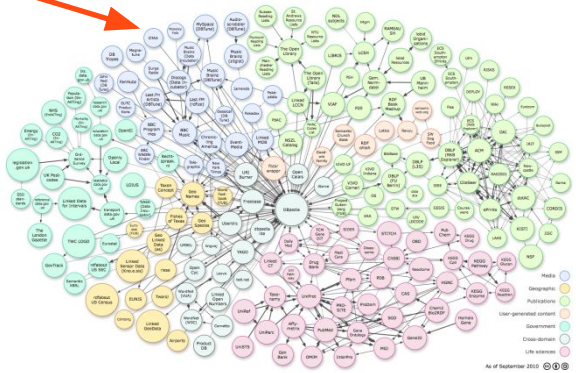


2009

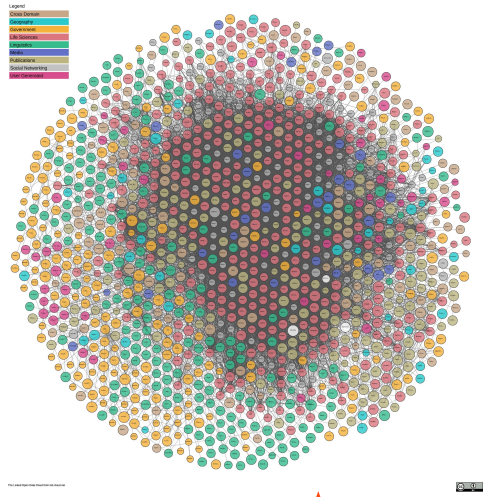


- UK Government
- US Government
- BBC
- Open Calais
- Freebase
- NY Times
- CNET
- DBpedia
- ….

2010



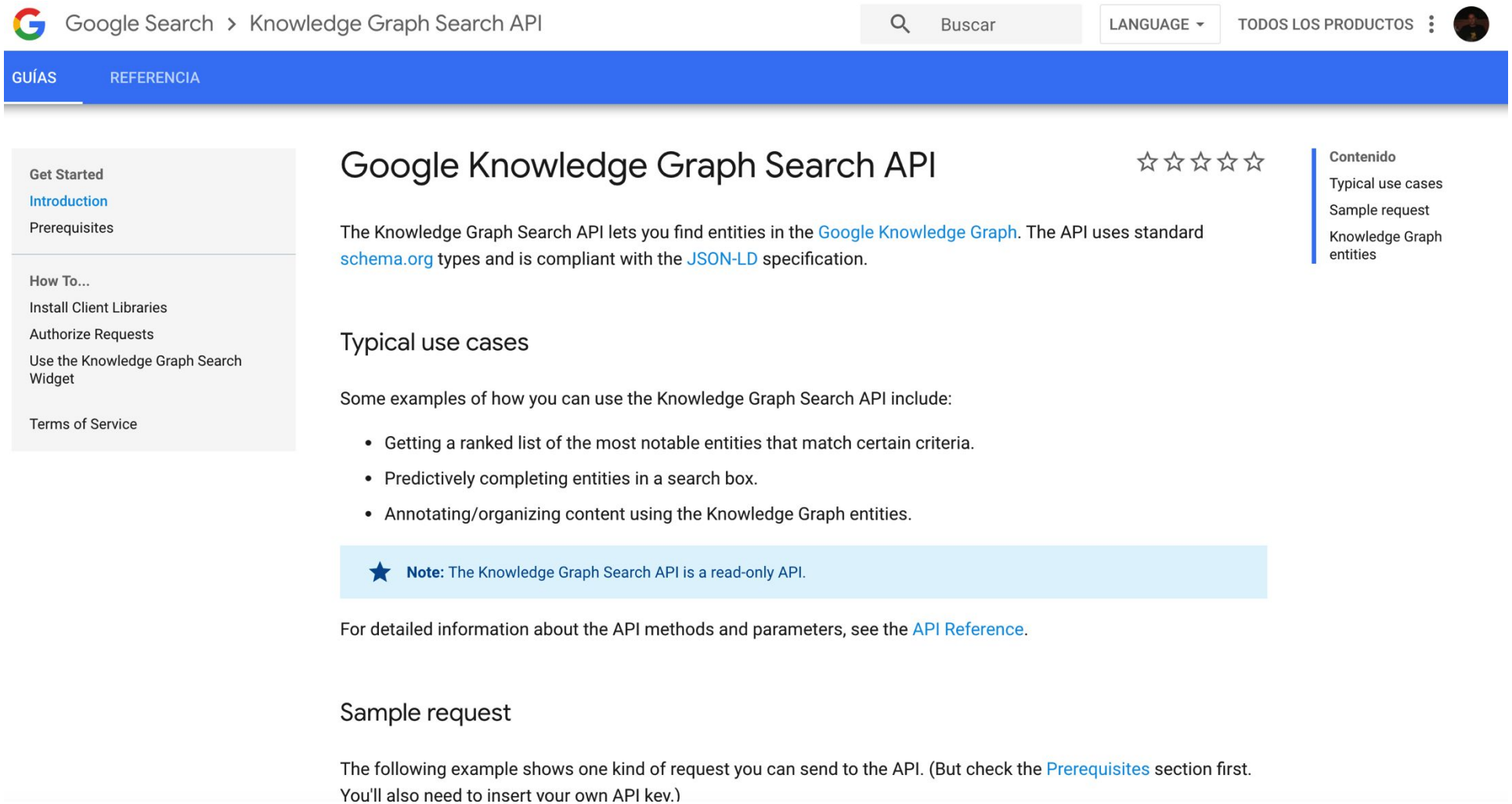
2018



"The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the Web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications."

Hendler, J., Berners-Lee, T., and Miller, E.
Integrating Applications on the Semantic Web, 2002,
http://www.w3.org/2002/07/swint.html

https://developers.google.com/knowledge-graph/
https://www.youtube.com/watch?v=mmQl6VGvX-c

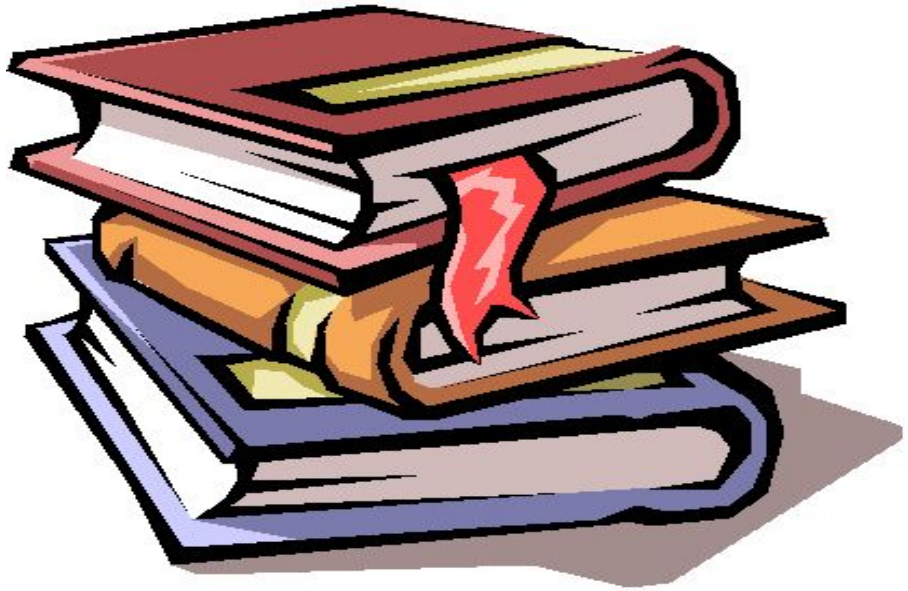


General-purpose open and domain-specific KGs

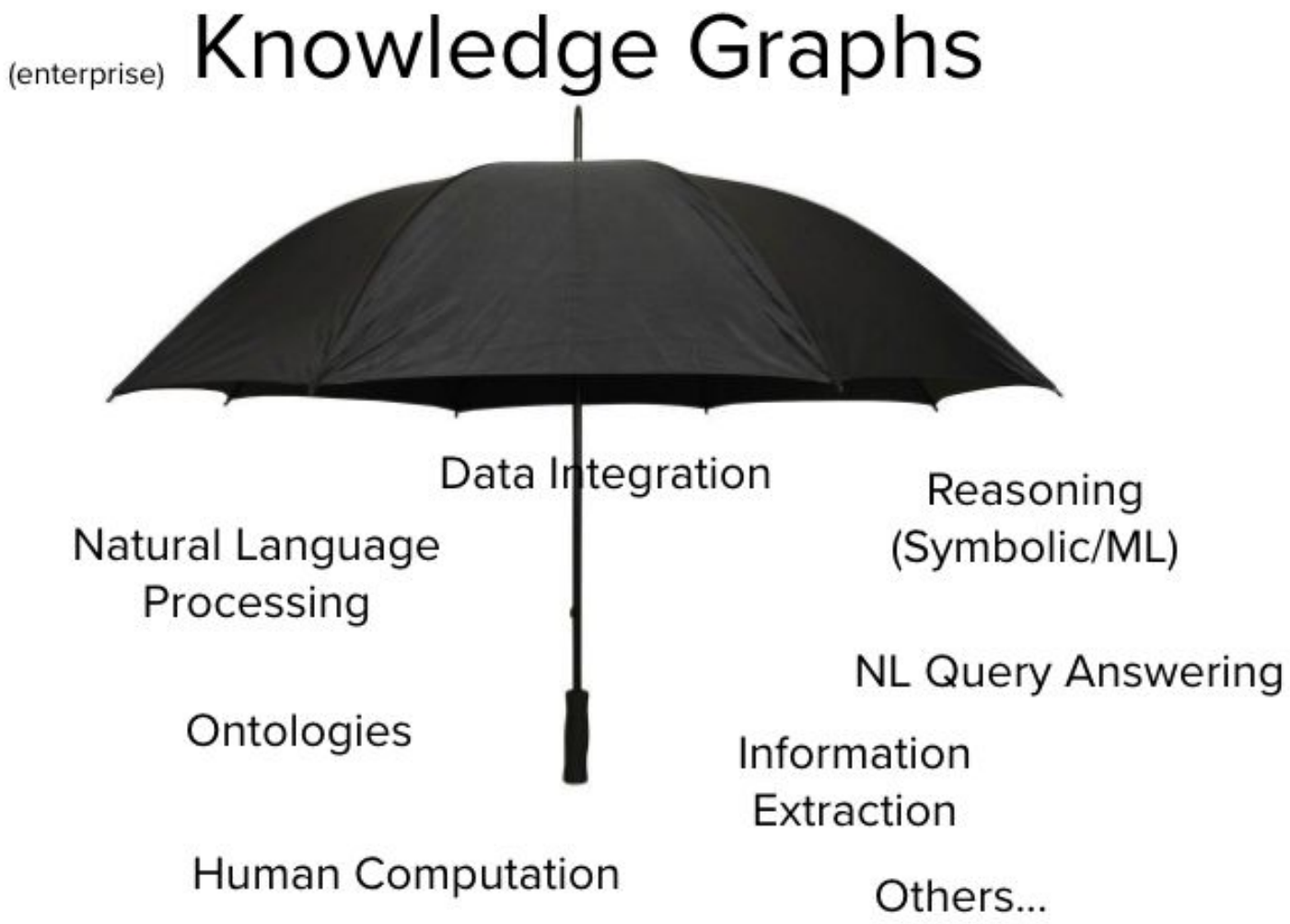
- Some of the well-known ones:
  - DBpedia and YAGO
  - Wikidata (and the earlier Freebase)
  - BabelNet
  - OpenCyc
- And many domain-specific ones
  - Media
  - Government
  - Publications
  - Geography
  - Life Sciences
  - Cultural heritage
  - Law

https://www.youtube.com/watch?v=TJfrNo3Z-DU

Recent references about Knowledge Graphs



- Hogan A, Blomqvist E, Cochez M, d'Amato C, de Melo G, Gutiérrez C, Labra Gayo JE, Kirrane S, Neumaier S, Polleres A, Navigli R, Ngonga Ngomo AC, Rashid SM, Rula A, Schmelzeisen L, Sequeda J, Staab S, Zimmermann A (2020) **Knowledge Graphs**. https://arxiv.org/abs/2003.02320
- Ji S, Pan S, Cambria E, Marttinen P, Yu PS (2020) **A Survey on Knowledge Graphs: Representation, Acquisition and Applications** https://arxiv.org/abs/2002.00388
- Gutiérrez C, Sequeda J (2021) **Knowledge Graphs**. Communications of the ACM 64(3):96-104. https://cacm.acm.org/magazines/2021/3/250711-knowledge-graphs/fulltext
- Guilin Qi, Huajun Chen, Kang Liu, Haofen Wang, Qiu Ji, and Tianxing Wu (2021) **Knowledge Graph**. Springer. *To appear*



Extracted from Mariano Rodriguez Muro from KGB2019 Keynote ESWC2019

What are we going to learn today?

Knowledge is more important than data

Symbolic AI still has something to say

See the Web as a big database with structured resources

Semantic Web technologies are not (so) difficult to learn

How to create, validate, manage and query RDF KGs

Successful and real use-cases with real companies

Benefits of structure your knowledge and data as RDF

How KGs help AI

Fork the repository (and the clone it)
https://github.com/dachafra/kg-tutorial

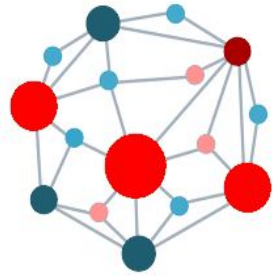
Good to have installed in your local machine
Git, Python

An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) **RDF, OWL and SPARQL**
3) **RDF APIs and Triplestores**
4) Hands-On 1: Querying Wikidata

Hands-On 2: Constructing and validating your RDF graph (50min)

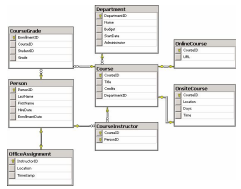
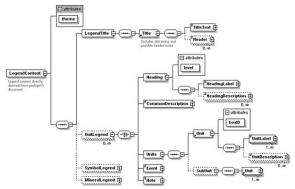
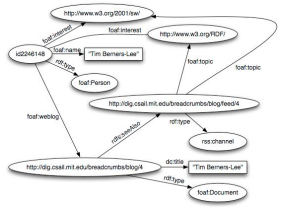
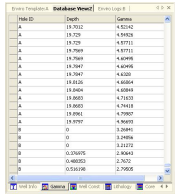
1) Tools for Developing Ontologies and KGs
2) RML (from CSV/XML/JSON to RDF)
3) SHACL (RDF validation)



Hands-On 3: Examples of Exploiting KGs for AI (30 min)

1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project

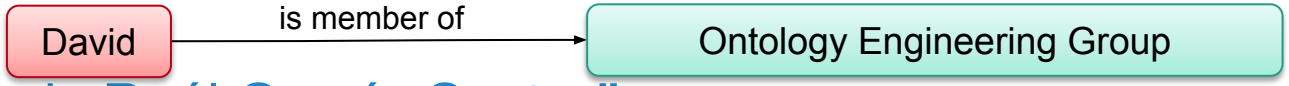
- RDF: Resource Description Framework

	Database	XML/JSON	RDF(S)	
Schema				RDF Schema OWL
Data				RDF

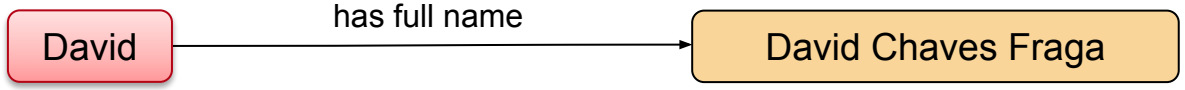
- W3C Recommendation
  - Model
  - Syntax
  - Semantics

- Also known as triples
  - [Subject, Predicate, Object]

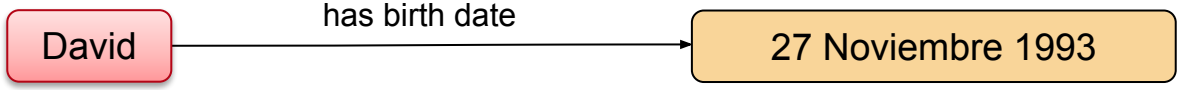
- "Raúl is a member of the Ontology Engineering Group"
  - [David, is member of, Ontology Engineering Group]



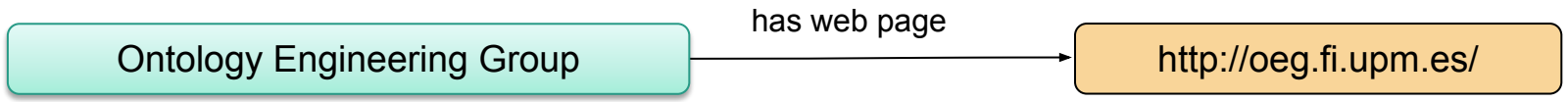
- "Raúl's full name is Raúl García Castro"
  - [David, has full name, Raúl García Castro]



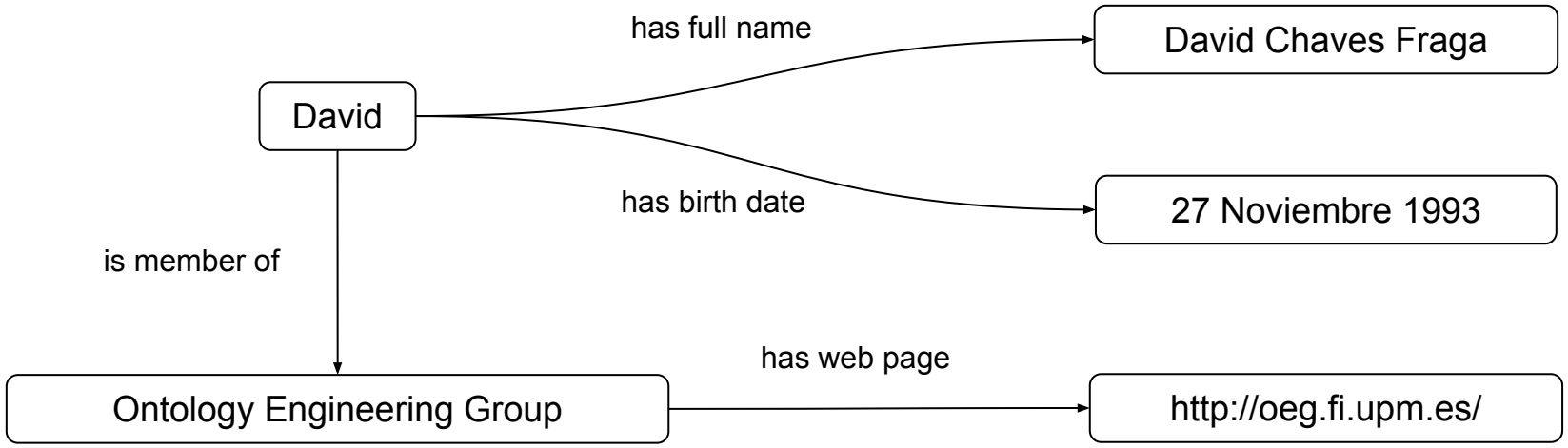
- "Raúl was born on December 26th 1975"
  - [David, was born, 26 December 1975]



- "The homepage of the Ontology Engineering Group is http://www.oeg-upm.net/"
  - [Ontology Engineering Group, has web page, http://oeg.fi.upm.es/]

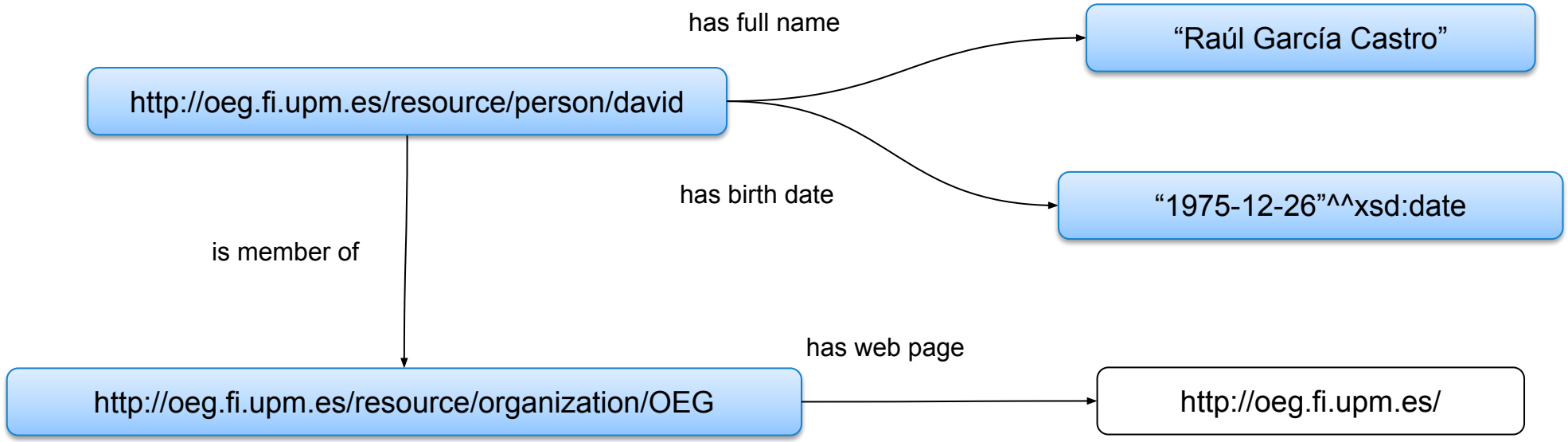


- RDF graphs are sets of triples



- A set of RDF graphs is an RDF Dataset (in RDF1.1)
  - There is a default graph
  - And zero or more named graphs

- Triple objects may be resources or literals
  - **Subjects and predicates are always resources**
- Literals may have an XML Schema datatype associated to them
  - Default: xsd:string
  - RDF1.1 defines a couple of extra ones
    - rdf:langString (e.g., “Spain”@en)
    - rdf:HTML and rdf:XMLLiteral



RDF literals. Common datatypes (I)

A list of the RDF-compatible XSD types, with short descriptions"

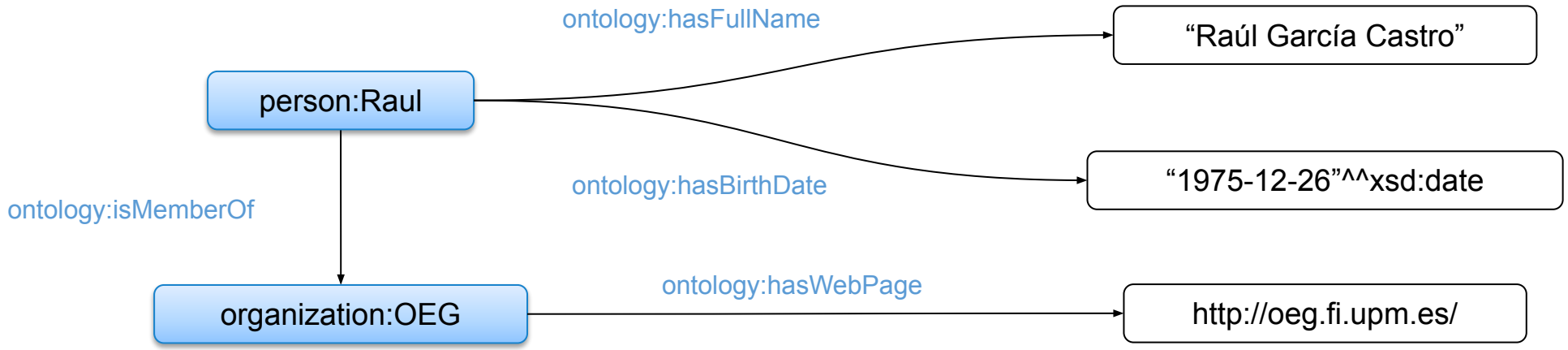
	Datatype	Value space (informative)
Core types	xsd:string	Character strings (but not all Unicode character strings)
	xsd:boolean	true, false
	xsd:decimal	Arbitrary-precision decimal numbers
	xsd:integer	Arbitrary-size integer numbers
IEEE floating-point numbers	xsd:double	64-bit floating point numbers incl. ±Inf, ±0, NaN
	xsd:float	32-bit floating point numbers incl. ±Inf, ±0, NaN
Time and date	xsd:date	Dates (yyyy-mm-dd) with or without timezone
	xsd:time	Times (hh:mm:ss.sss…) with or without timezone
	xsd:dateTime	Date and time with or without timezone
	xsd:dateTimeStamp	Date and time with required timezone
Recurring and partial dates	xsd:gYear	Gregorian calendar year
	xsd:gMonth	Gregorian calendar month
	xsd:gDay	Gregorian calendar day of the month
	xsd:gYearMonth	Gregorian calendar year and month
	xsd:gMonthDay	Gregorian calendar month and day
	xsd:duration	Duration of time
	xsd:yearMonthDuration	Duration of time (months and years only)
	xsd:dayTimeDuration	Duration of time (days, hours, minutes, seconds only)

RDF literals. Common datatypes (II)

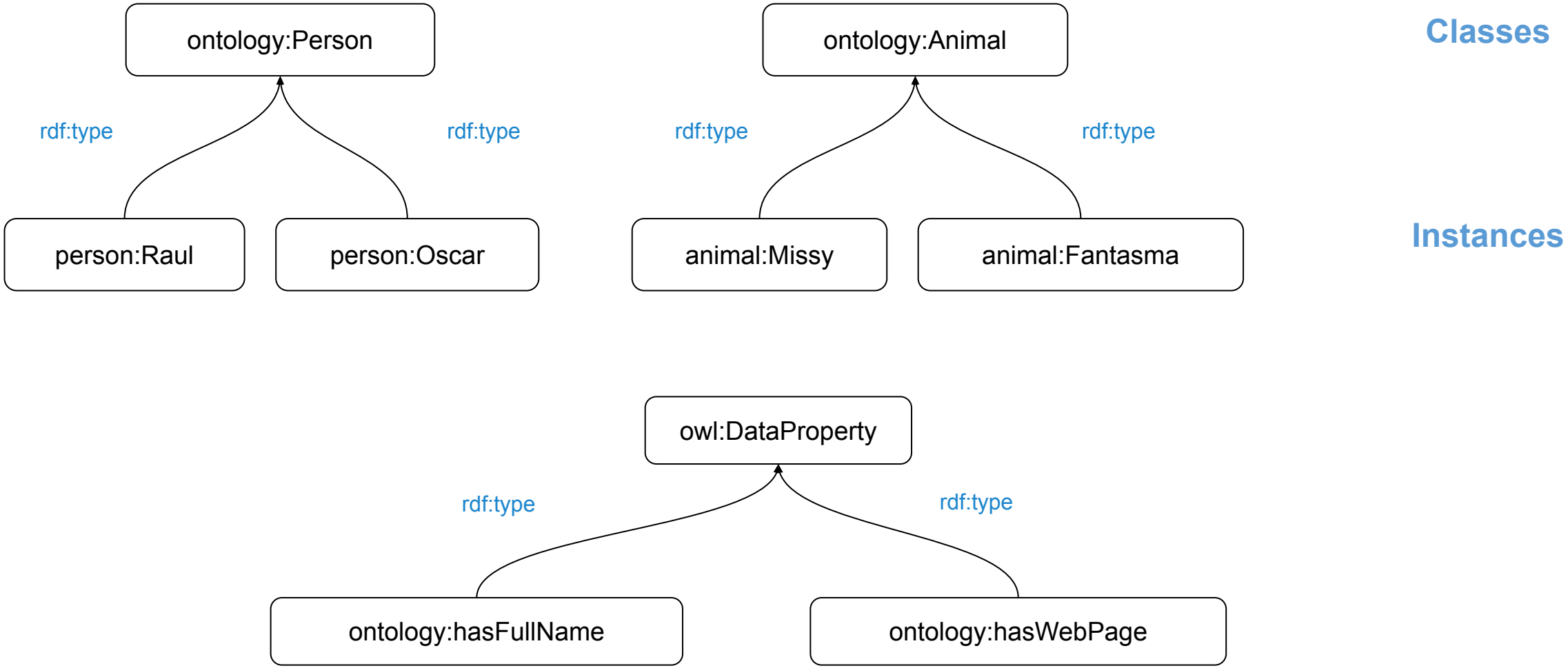
Limited-range integer numbers	xsd:byte	-128…+127 (8 bit)
	xsd:short	-32768…+32767 (16 bit)
	xsd:int	-2147483648…+2147483647 (32 bit)
	xsd:long	-9223372036854775808…+9223372036854775807 (64 bit)
	xsd:unsignedByte	0…255 (8 bit)
	xsd:unsignedShort	0…65535 (16 bit)
	xsd:unsignedInt	0…4294967295 (32 bit)
	xsd:unsignedLong	0…18446744073709551615 (64 bit)
	xsd:positiveInteger	Integer numbers >0
	xsd:nonNegativeInteger	Integer numbers ≥0
	xsd:negativeInteger	Integer numbers <0
	xsd:nonPositiveInteger	Integer numbers ≤0
Encoded binary data	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary data
Miscellaneous XSD types	xsd:anyURI	Absolute or relative URIs and IRIs
	xsd:language	Language tags per [BCP47]
	xsd:normalizedString	Whitespace-normalized strings
	xsd:token	Tokenized strings
	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
	xsd:NCName	XML NCNames

- URIs under a *namespace* are called vocabularies

Prefix	URI
person	http://oeg.fi.upm.es/resource/person/
organization	http://oeg.fi.upm.es/resource/organization/
ontology	http://oeg.fi.upm.es/def/people#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#

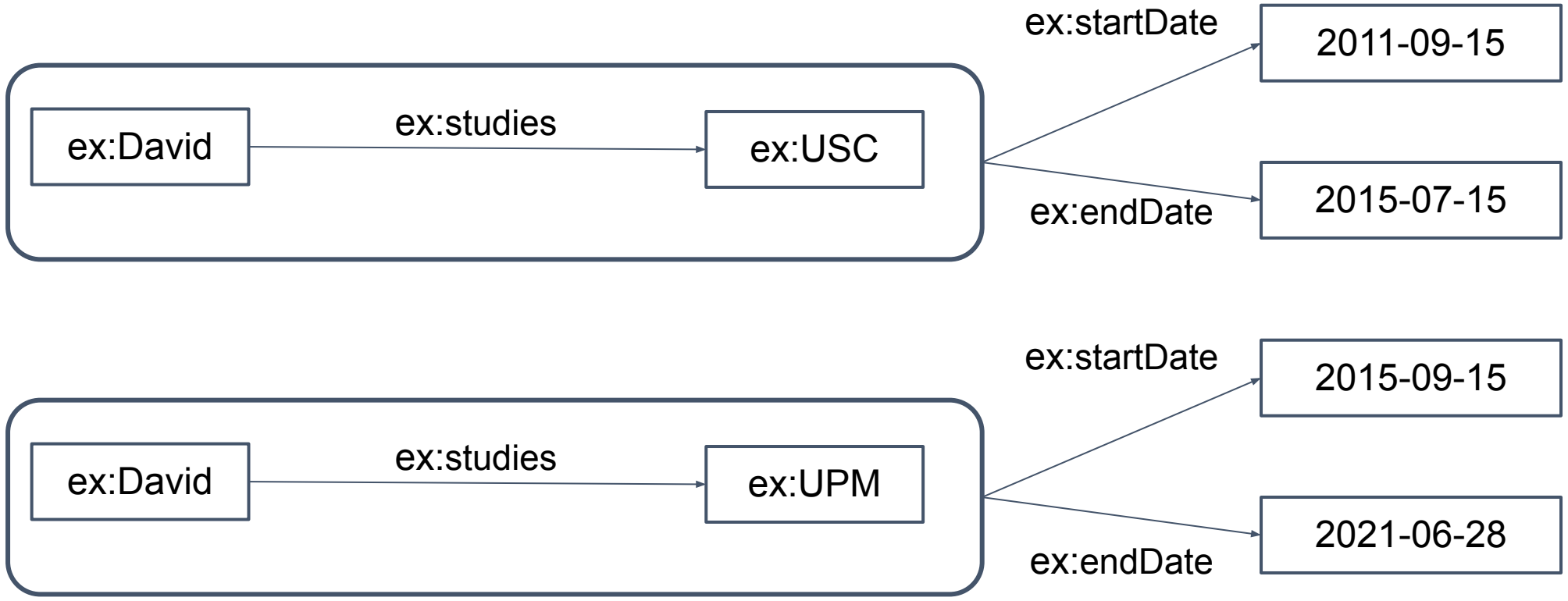


- ***rdf:type*** is used to associate resources to their corresponding categories/classes



Statements about Statements (or RDF Reification)

"David studied in USC from 2011-09-15 to 2015-07-15 and in UPM from 2015-09-15 to 2021-06-28"



RDF reification: It allows expressing **beliefs (and other modalities), confidence models and metadata about data**

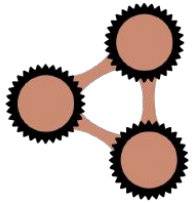
Triples that include a **triple as a subject or an object** are known as RDF-star triples

An RDF-star graph is a **set of RDF-star triples.**

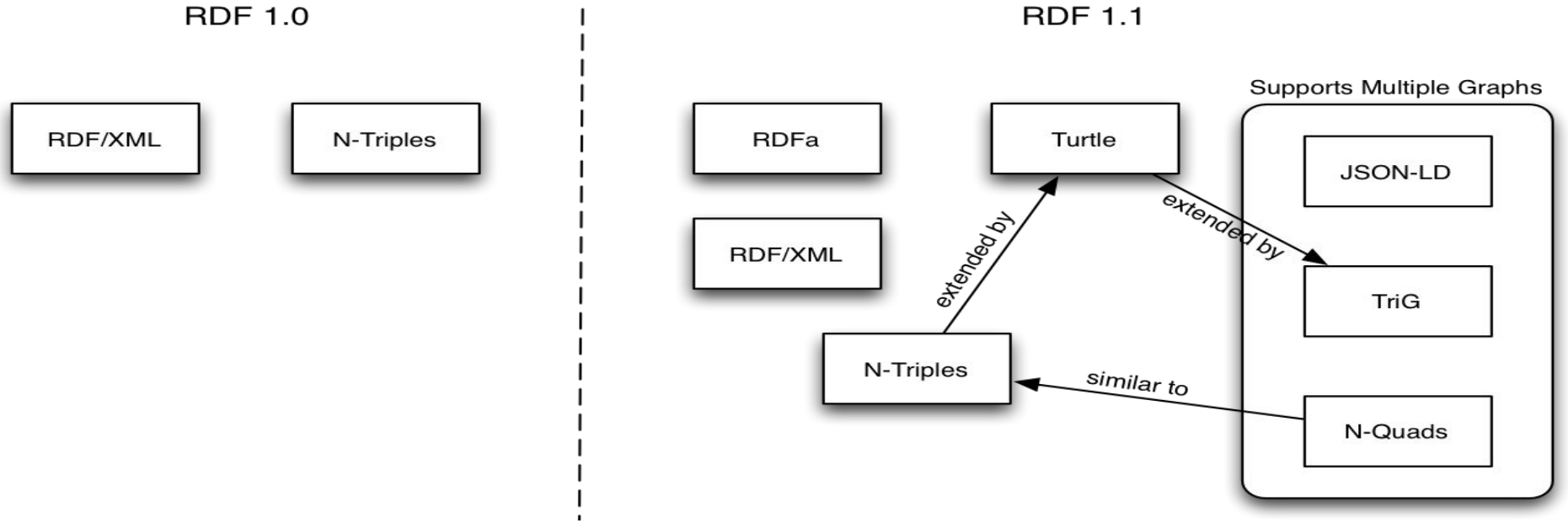
SPARQL-star extends SPARQL to query RDF-star graphs

```
<< ex:David ex:studies ex:USC >> ex:startDate "2011-09-15" .
<< ex:David ex:studies ex:USC >> ex:endDate "2015-07-15" .
<< ex:David ex:studies ex:UPM >> ex:startDate "2015-09-15" .
<< ex:David ex:studies ex:UPM >> ex:endDate "2021-06-28" .
```

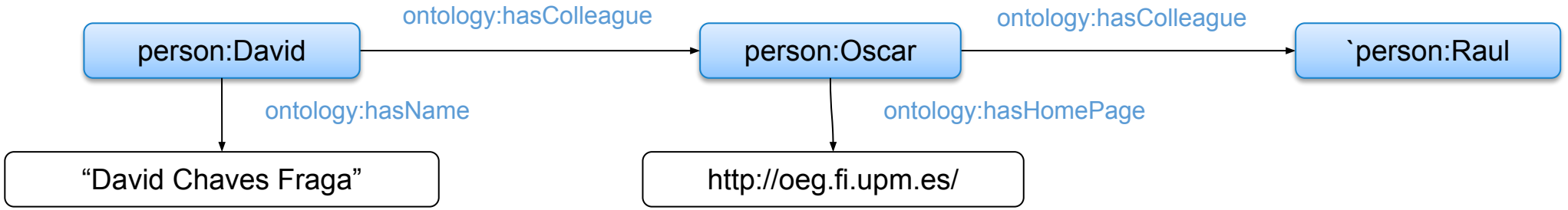





- Different syntaxes (some of them recently approved – February 2014)
  - **RDF/XML** (www.w3.org/TR/rdf-syntax-grammar/)
  - **Turtle** (http://www.w3.org/TR/turtle/)
  - **N-Triples** (http://www.w3.org/TR/n-triples/)
  - **TriG** (http://www.w3.org/TR/trig/)
  - **RDFa** (http://www.w3.org/TR/xhtml-rdfa-primer/)
  - **JSON-LD** (http://www.w3.org/TR/json-ld/)
  - **N-Quads** (http://www.w3.org/TR/n-quads/)



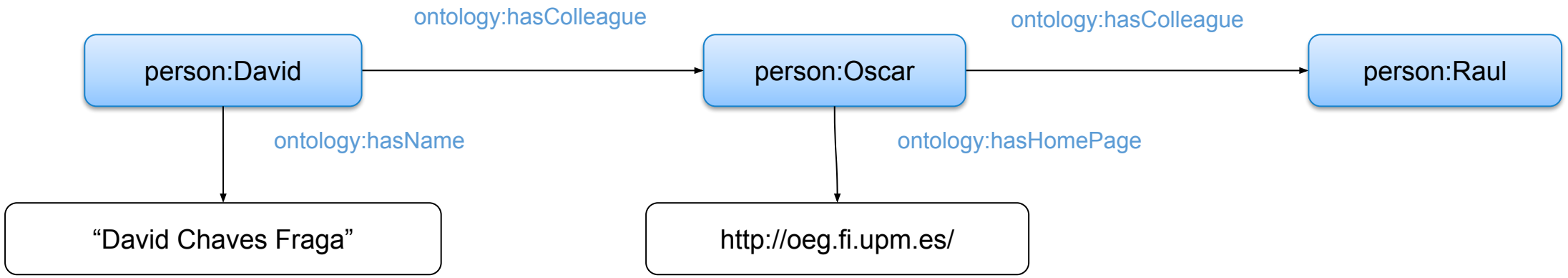
- **Important: the order of triples inside a document is not relevant**



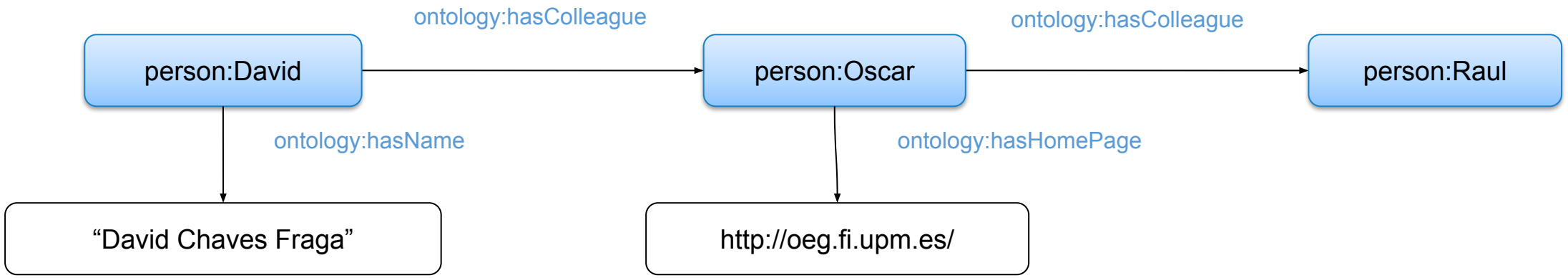

```
<?xml version="1.0"?>
   <rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ontology="http://oeg.fi.upm.es/def/people#"
        xmlns="http://oeg.fi.upm.es/resource/person/"
        xml:base="http://oeg.fi.upm.es/resource/person/" >

     <rdf:Property rdf:about="http://oeg.fi.upm.es/def/people#hasHomePage" />
     <rdf:Property rdf:about="http://oeg.fi.upm.es/def/people#hasColleague" />
     <rdf:Property rdf:about="http://oeg.fi.upm.es/def/people#hasName" />

     <rdf:Description rdf:about="Raul"/>
     <rdf:Description rdf:about="Oscar">
       <ontology:hasColleague rdf:resource="Oscar"/>
       <ontology:hasHomePage>http://www.oeg-upm.net/</ontology:hasHomePage>
     </rdf:Description>
     <rdf:Description rdf:about="David">
       <ontology:hasColleague rdf:resource="Oscar"/>
       <ontology:hasName>David Chaves Fraga</ontology:hasName>
     </rdf:Description>
   </rdf:RDF>
```



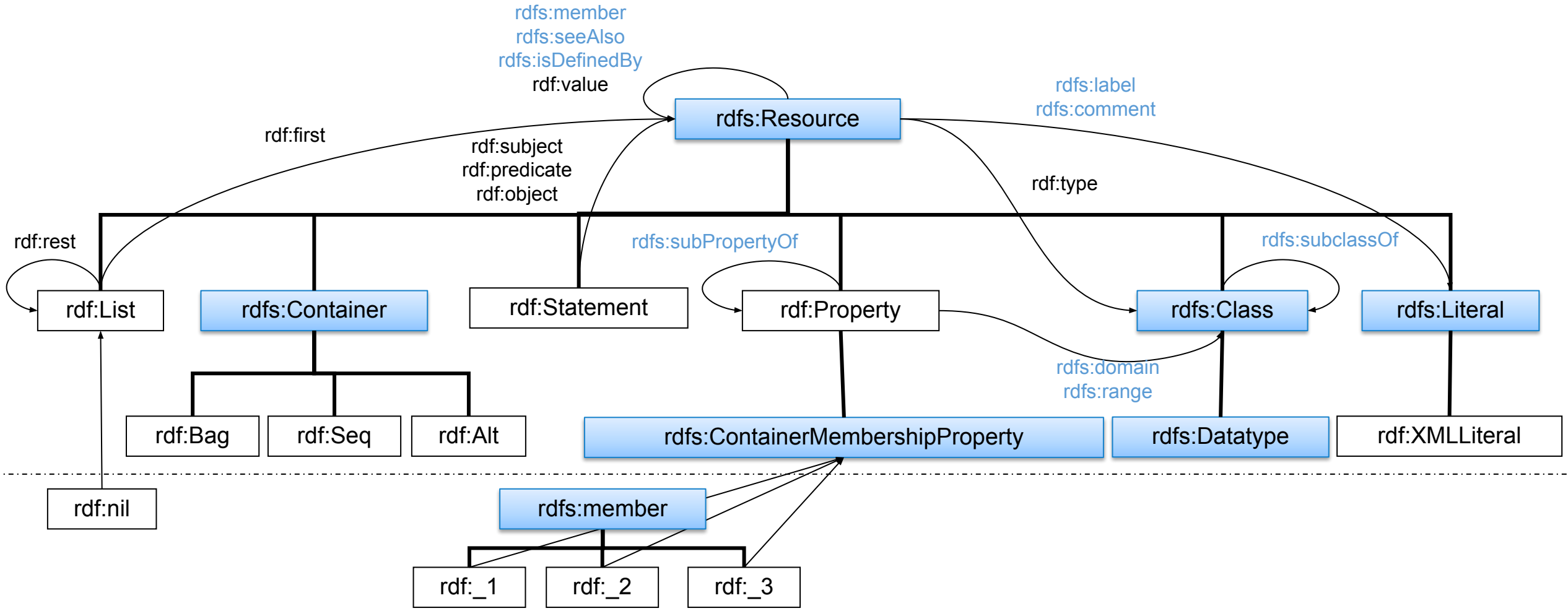
<http://example.org/Oscar> <http://example.org/hasColleague> <http://example.org/Raul> .
<http://example.org/Oscar> <http://example.org/hasHomePage> <http://oeg.fi.upm.es/>.
<http://example.org/David> <http://example.org/hasColleague> <http://example.org/Oscar> .
<http://example.org/David> <http://example.org/hasName> “David Chaves Fraga”.



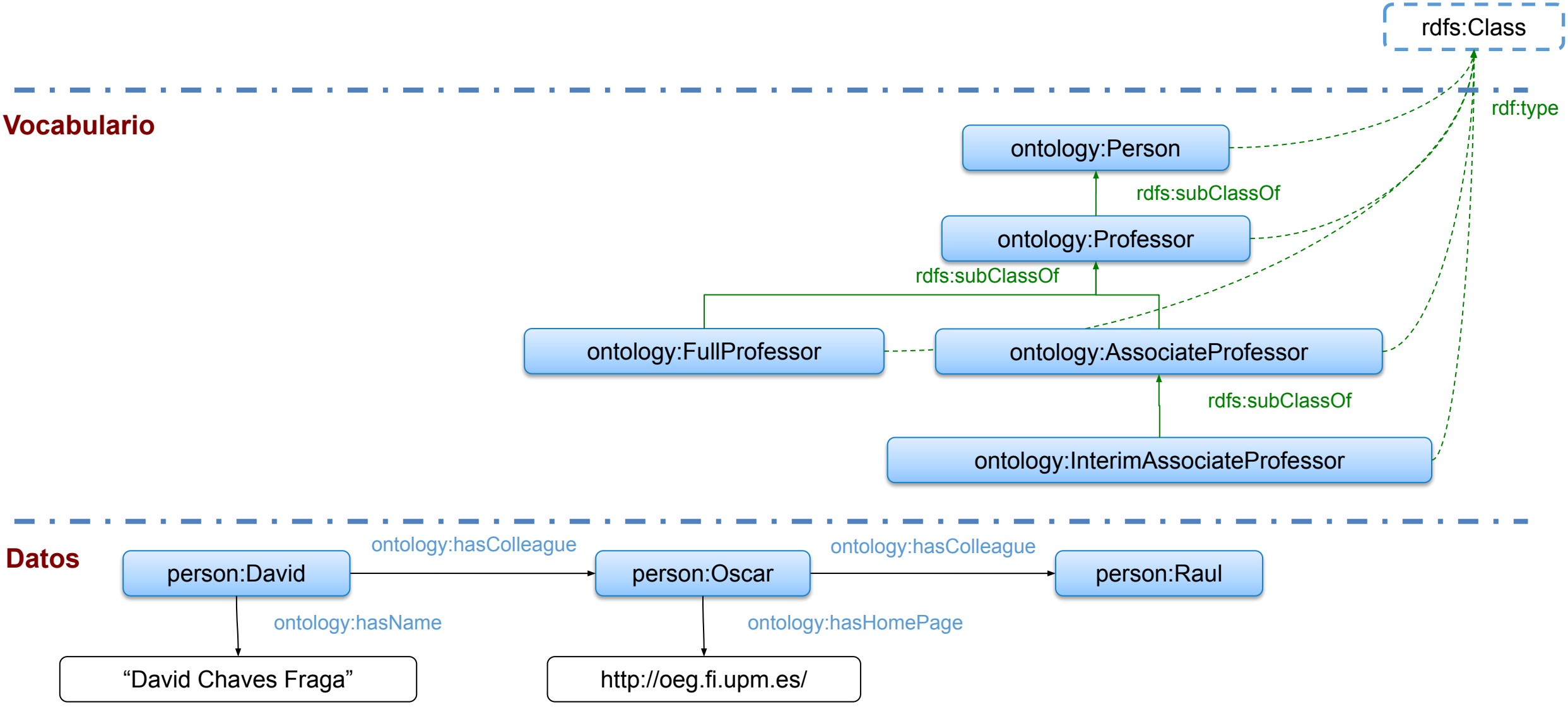

```
@base  <http://oeg.fi.upm.es/resource/person/ >
@prefix ontology: <http://oeg.fi.upm.es/def/people#>
:Oscar      ontology:hasColleague :Raul ;
      ontology:hasHomePage <http://oeg.fi.upm.es/>.
:David ontology:hasColleague :Oscar ;
      ontology:hasName “David Chaves Fraga”.
```

```
1  <http://ejemplo.es/recurso/Victor> <http://xmlns.com/foaf/0.1/age> 25 “years” .
2  “Maria” <http://xmlns.com/foaf/0.1/age> 25 .
3  <http://ejemplo.es/recurso/Maria> <http://xmlns.com/foaf/0.1/age> 25 .
4  <http://ejemplo.es/recurso/Victor> <http://xmlns.com/foaf/0.1/isMarried> .
5  <http://ejemplo.es/recurso/Victor> http://xmlns.com/foaf/0.1/livesIn “Madrid”.
6  <http://ejemplo.es/recurso/M> <http://xmlns.com/foaf/0.1/fn> “Maria alias “Mary” ” .
7  <http://ejemplo.es/recurso/M> <http://xmlns.com/foaf/0.1/fn> “Maria”^^xsd:string .
```

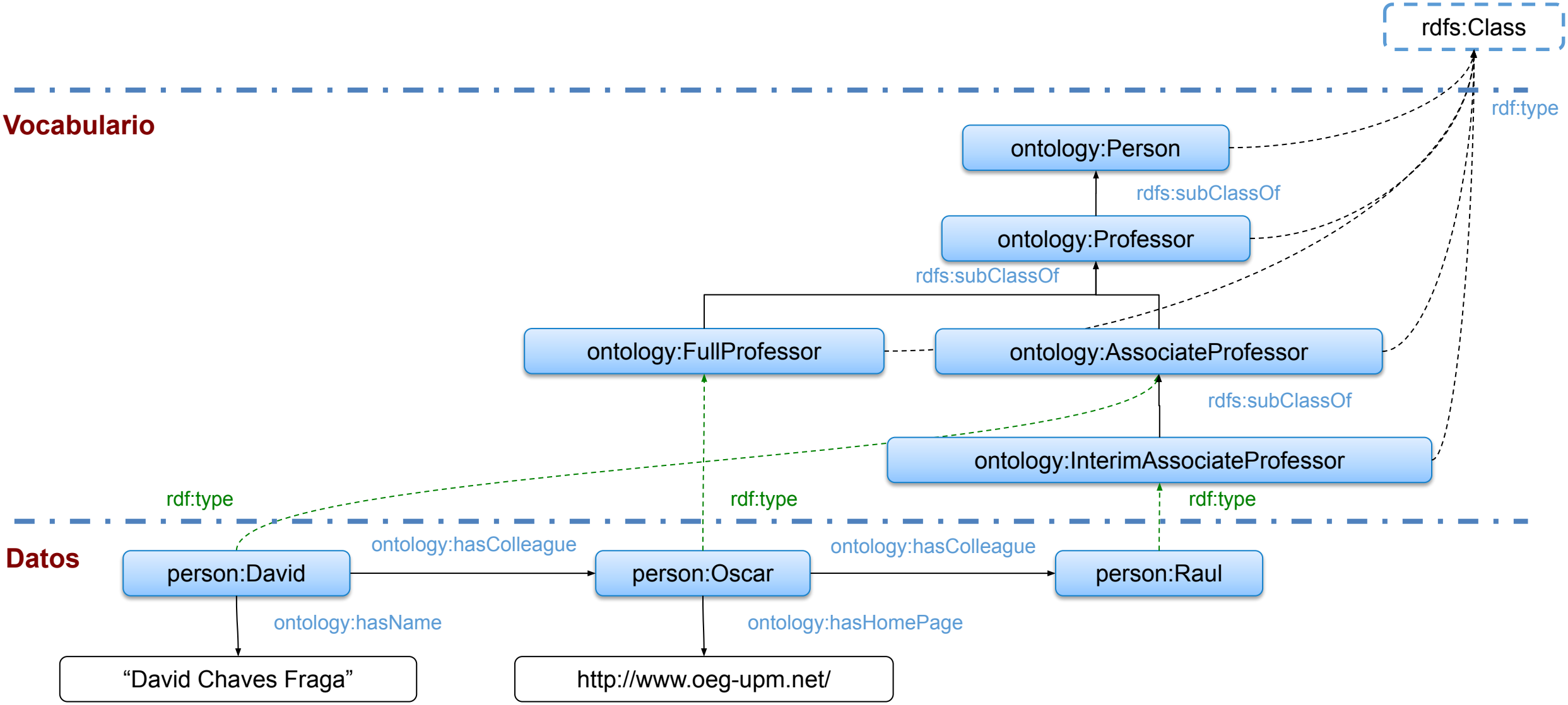
- Allows describing classes and properties
- Adds restrictions to models

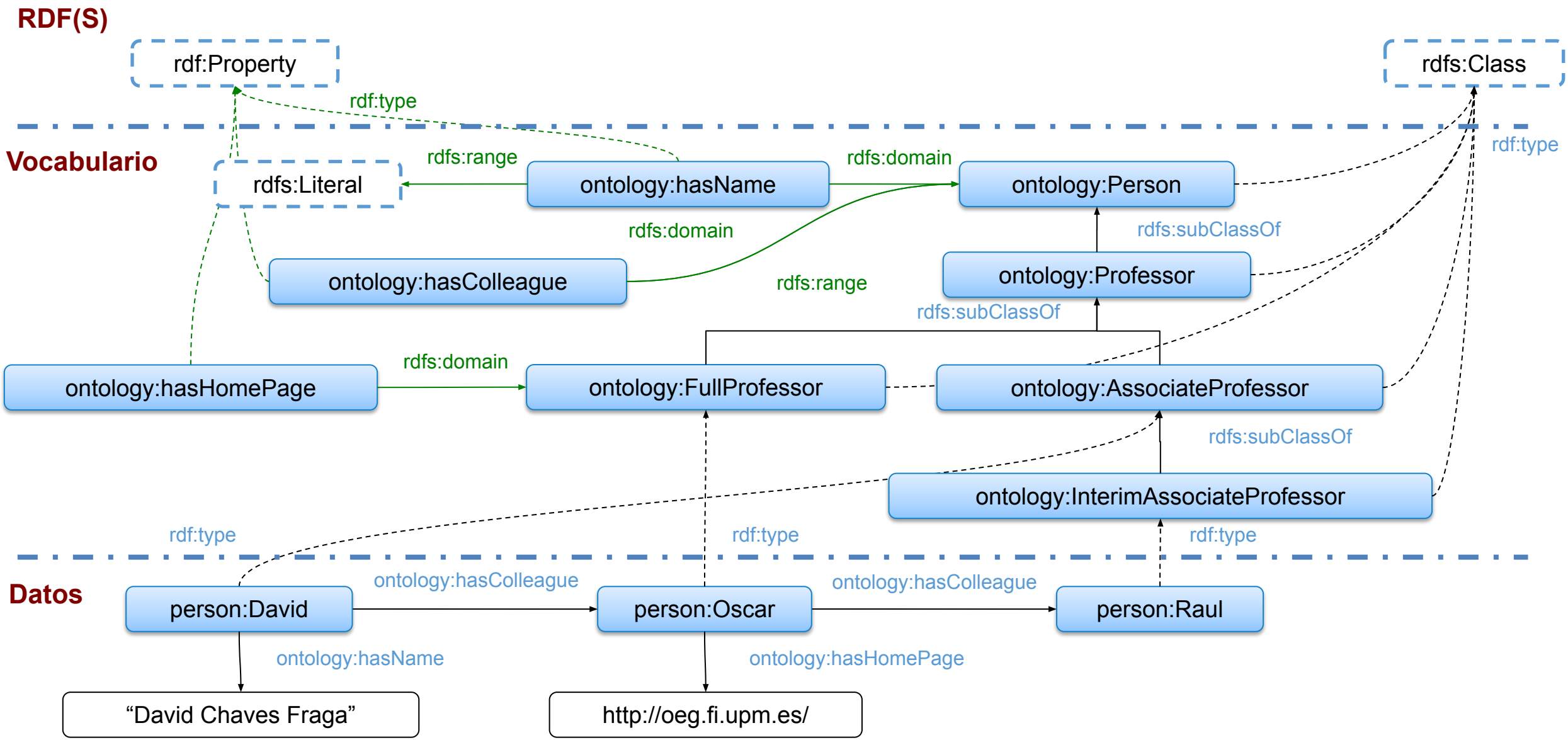


RDF(S)


rdfs:Class
rdf:type
Vocabulario
ontology:Person
rdfs:subClassOf
ontology:Professor
rdfs:subClassOf
ontology:FullProfessor
ontology:AssociateProfessor
rdfs:subClassOf
ontology:InterimAssociateProfessor
Datos
person:David
ontology:hasColleague
person:Oscar
ontology:hasColleague
person:Raul
ontology:hasName
ontology:hasHomePage
“David Chaves Fraga”
http://oeg.fi.upm.es/

RDF(S)


rdfs:Class
rdf:type
Vocabulario
ontology:Person
rdfs:subClassOf
ontology:Professor
rdfs:subClassOf
ontology:FullProfessor
ontology:AssociateProfessor
rdfs:subClassOf
ontology:InterimAssociateProfessor
rdf:type
rdf:type
rdf:type
Datos
ontology:hasColleague
ontology:hasColleague
person:David
person:Oscar
person:Raul
ontology:hasName
ontology:hasHomePage
“David Chaves Fraga”
http://www.oeg-upm.net/


RDF(S)
rdf:Property
rdfs:Class
rdf:type
Vocabulario
rdfs:range
rdfs:domain
rdfs:Literal
ontology:hasName
ontology:Person
rdf:type
rdfs:domain
rdfs:subClassOf
ontology:hasColleague
ontology:Professor
rdfs:range
rdfs:subClassOf
rdfs:domain
ontology:hasHomePage
ontology:FullProfessor
ontology:AssociateProfessor
rdfs:subClassOf
ontology:InterimAssociateProfessor
rdf:type
rdf:type
rdf:type
Datos
ontology:hasColleague
ontology:hasColleague
person:David
person:Oscar
person:Raul
ontology:hasName
ontology:hasHomePage
“David Chaves Fraga”
http://oeg.fi.upm.es/

- RDFS is too weak to describe resources in sufficient detail
  - No localised range and domain constraints
    - Can't say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can't say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No boolean operators
    - Can't say or, not, etc.
  - No transitive, inverse or symmetrical properties
    - Can't say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical
- Difficult to provide reasoning support
  - No "native" reasoners for non-standard semantics
  - May be possible to reason via FOL axiomatisation

W3C Recommendation (February 2004) / Built on top of RDF(S)

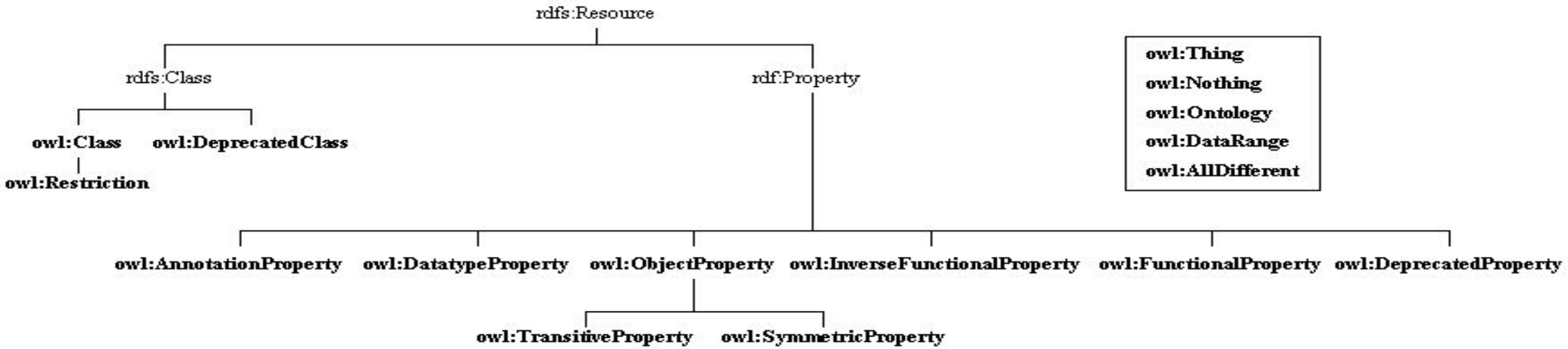
3 layers:

- OWL Lite
  - A small subset of primitives
  - Easier for frame-based tools to transition to
- OWL DL
  - Description logic
  - Decidable reasoning
- OWL Full
  - RDF extension, allows metaclasses

Several syntaxes:

- Abstract syntax
- Manchester syntax
- RDF/XML

Class taxonomy of the OWL KR ontology



SPARQL Query Language

(or how to query RDF data)

query	result
SELECT queries	XML, JSON, TSV, CSV
ASK queries	true or false in XML or JSON
CONSTRUCT queries	RDF graph constructed with a template
DESCRIBE queries	RDF graph with data about resources

```
SELECT/CONSTRUCT/DESCRIBE/ASK …
WHERE {
    <graph-pattern>
}
```

A SELECT query has two main components:
- a list of selected **variables** and
- a **WHERE** clause with the graph patterns to match

```
SELECT <variables>
WHERE {
    <graph-pattern>
}
```

```
SELECT ?s ?p ?o
WHERE {
    ?s ?p ?o .
}
```

```
SELECT ?p ?o
WHERE {
    ex:Katerina%20Stefanidi ?p ?o.
}
```

p	o
rdf:type	foaf:Person
ex:score	"4.85"^^xsd:decimal
foaf:name	"Katerina Stefanidi"@en
ex:country	ex:EL

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
SELECT ?s ?o
WHERE {
    ?s <http://xmlns.com/foaf/0.1/name> ?o.
}
```

s	o
ex:Anzhelika%20Sidorova	"Anzhelika Sidorova"@en
ex:Sandi%20Morris	"Sandi Morris"@en
ex:Katerina%20Stefanidi	"Katerina Stefanidi"@en
ex:Holly%20Bradshaw	"Holly Bradshaw"@en
ex:Alysha%20Newman	"Alysha Newman"@en
ex:Angelica%20Bengtsson	"Angelica Bengtsson"@en

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
SELECT ?s ?p
WHERE {
    ?s ?p "Katerina Stefanidi".
}
```

s	p

```
SELECT ?s ?p
WHERE {
    ?s ?p "Katerina Stefanidi"@en.
}
```

s	p
ex:Katerina%20Stefanidi	foaf:name

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
SELECT ?s
WHERE {
    ?s ?p1 4.80;
       ?p2 <http://ex.com/UK> .
}
```

s
ex:Holly%20Bradshaw

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>
SELECT ?s ?o
WHERE {
    ?s ex:score ?o.
    FILTER (?o < 4.90).
}
```

s	o
ex:Holly%20Bradshaw	"4.80"^^xsd:decimal
ex:Alysha%20Newman	"4.80"^^xsd:decimal
ex:Angelica%20Bengtsson	"4.80"^^xsd:decimal
ex:Katerina%20Stefanidi	"4.85"^^xsd:decimal

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

FILTER - regex

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?s ?o
WHERE {
   ?s foaf:name ?o.
   FILTER regex(?o, "A").
}
```

s	o
ex:Anzhelika%20Sidorova	"Anzhelika Sidorova"@en
ex:Alysha%20Newman	"Alysha Newman"@en
ex:Angelica%20Bengtsson	"Angelica Bengtsson"@en

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

OPTIONAL (1)

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?s ?o ?c
WHERE { ?s foaf:name ?o.
   OPTIONAL { ?s ex:country ?c }.}
```

s	o	c
ex:Alysha%20Newman	"Alysha Newman"@en	ex:CA
ex:Angelica%20Bengtsson	"Angelica Bengtsson"@en	ex:SE
ex:Anzhelika%20Sidorova	"Anzhelika Sidorova"@en	ex:RU
ex:Holly%20Bradshaw	"Holly Bradshaw"@en	ex:UK
ex:Katerina%20Stefanidi	"Katerina Stefanidi"@en	ex:EL
ex:Sandi%20Morris	"Sandi Morris"@en	

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      ex:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?s ?o ?c
WHERE {
    ?s foaf:name ?o.
    { ?s  ex:country  ?c }
     UNION
    { ?s  foaf:country  ?c }
}
```

```
ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        ex:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       foaf:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson foaf:country <http://ex.com/SE>.
```

s	o	c
ex:Anzhelika%20Sidorova	"Anzhelika Sidorova"@en	ex:RU
ex:Katerina%20Stefanidi	"Katerina Stefanidi"@en	ex:EL
ex:Holly%20Bradshaw	"Holly Bradshaw"@en	ex:UK
ex:Alysha%20Newman	"Alysha Newman"@en	ex:CA
ex:Angelica%20Bengtsson	"Angelica Bengtsson"@en	ex:SE
ex:Sandi%20Morris	"Sandi Morris"@en	ex:US

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?s ?n ?o
WHERE {
   ?s a foaf:Person .
   ?s foaf:name ?n .
   FILTER NOT EXISTS { ?s ex:country ?o }
}
```

s	o	c
ex:Holly%20Bradshaw	"Holly Bradshaw"@en	ex:UK
ex:Sandi%20Morris	"Sandi Morris"@en	ex:US

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        foaf:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?o (COUNT(?s) AS ?total)
WHERE {
    ?s a ?o.
}
GROUP BY ?o
```

o	total
foaf:Person	"6"^^xsd:integer

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        foaf:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?o (COUNT(?s) AS ?total)
WHERE {
    ?s foaf:name ?n ; ex:score ?o.
}
GROUP BY ?o
HAVING (?o < 4.85)
```

o	total
"4.80"^^xsd:decimal	"3"^^xsd:integer

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        foaf:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>
PREFIX foaf:
<http://xmlns.com/foaf/0.1/>

CONSTRUCT { ?s ex:name ?o }
WHERE {
    ?s a ?c.
    ?s foaf:name ?o.
}
```

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        foaf:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
PREFIX ex: <http://ex.com/>

PREFIX foaf:
<http://xmlns.com/foaf/0.1/>


CONSTRUCT { ?s ex:name ?o }

WHERE {

    ?s a ?c.

    ?s foaf:name ?o.

}
```

```
ex:Anzhelika%20Sidorova  rdf:type foaf:Person.
ex:Sandi%20Morris        rdf:type foaf:Person.
ex:Katerina%20Stefanidi  rdf:type foaf:Person.
ex:Holly%20Bradshaw      rdf:type foaf:Person.
ex:Alysha%20Newman       rdf:type foaf:Person.
ex:Angelica%20Bengtsson rdf:type foaf:Person.

ex:Anzhelika%20Sidorova  ex:score "4.95"^^xsd:decimal.
ex:Sandi%20Morris        ex:score "4.90"^^xsd:decimal.
ex:Katerina%20Stefanidi  ex:score "4.85"^^xsd:decimal.
ex:Holly%20Bradshaw      ex:score "4.80"^^xsd:decimal.
ex:Alysha%20Newman       ex:score "4.80"^^xsd:decimal.
ex:Angelica%20Bengtsson ex:score "4.80"^^xsd:decimal.

ex:Anzhelika%20Sidorova  foaf:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        foaf:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  foaf:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      foaf:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       foaf:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson foaf:name "Angelica Bengtsson"@en.

ex:Anzhelika%20Sidorova  ex:country <http://ex.com/RU>.
ex:Sandi%20Morris        foaf:country <http://ex.com/US>.
ex:Katerina%20Stefanidi  ex:country <http://ex.com/EL>.
ex:Holly%20Bradshaw      foaf:country <http://ex.com/UK>.
ex:Alysha%20Newman       ex:country <http://ex.com/CA>.
ex:Angelica%20Bengtsson ex:country <http://ex.com/SE>.
```

```
ex:Anzhelika%20Sidorova  ex:name "Anzhelika Sidorova"@en.
ex:Sandi%20Morris        ex:name "Sandi Morris"@en.
ex:Katerina%20Stefanidi  ex:name  "Katerina Stefanidi"@en.
ex:Holly%20Bradshaw      ex:name "Holly Bradshaw"@en.
ex:Alysha%20Newman       ex:name "Alysha Newman"@en.
ex:Angelica%20Bengtsson ex:name "Angelica Bengtsson"@en.
```

SPARQL stores/endpoints

Amazon Neptune https://aws.amazon.com/neptune/

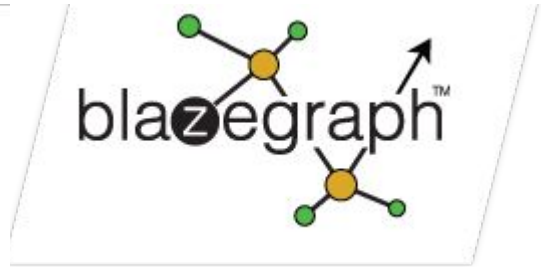
BlazeGraph https://blazegraph.com/

GraphDB https://graphdb.ontotext.com/

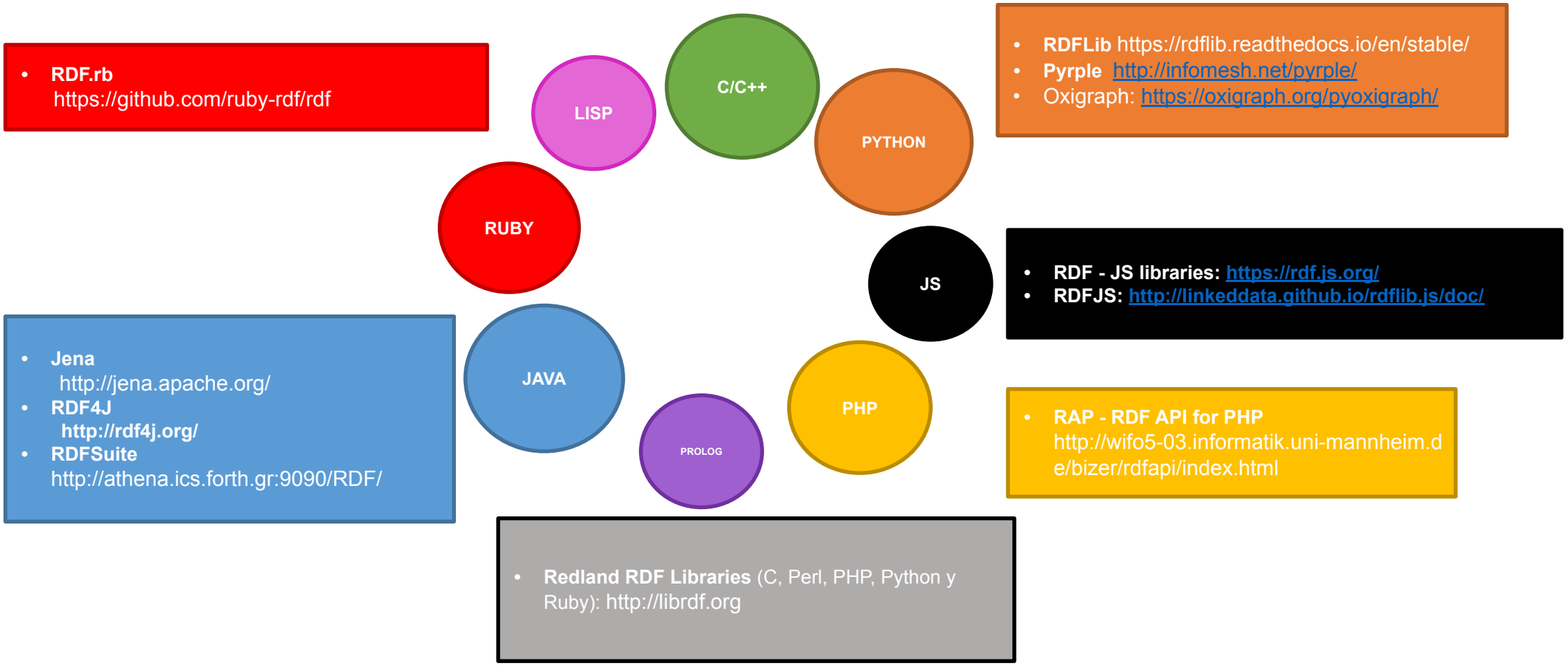
OpenLink Virtuoso https://lod.openlinksw.com/sparql

Oracle Graph Database https://www.oracle.com/be/database/graph/

Stardog https://www.stardog.com/platform/



Main RDF API



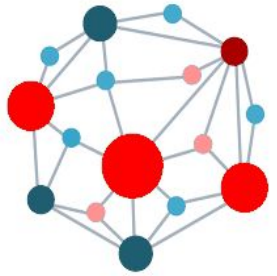
Full list available at
https://www.w3.org/2001/sw/wiki/Tools

An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) RDF, OWL and SPARQL
3) RDF APIs and Triplestores
4) **Hands-On 1: Querying Wikidata**

Hands-On 2: Constructing and validating your RDF graph (50min)

1) Tools for Developing Ontologies and KGs
2) RML (from CSV/XML/JSON to RDF)
3) SHACL (RDF validation)



Hands-On 3: Examples of Exploiting KGs for AI (30 min)

1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project

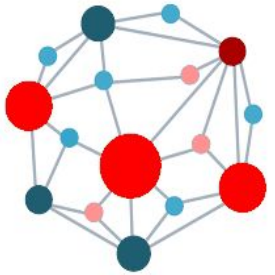
Wikidata-Queries Examples

An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) RDF, OWL and SPARQL
3) RDF APIs and Triplestores
4) Hands-On 1: Querying Wikidata

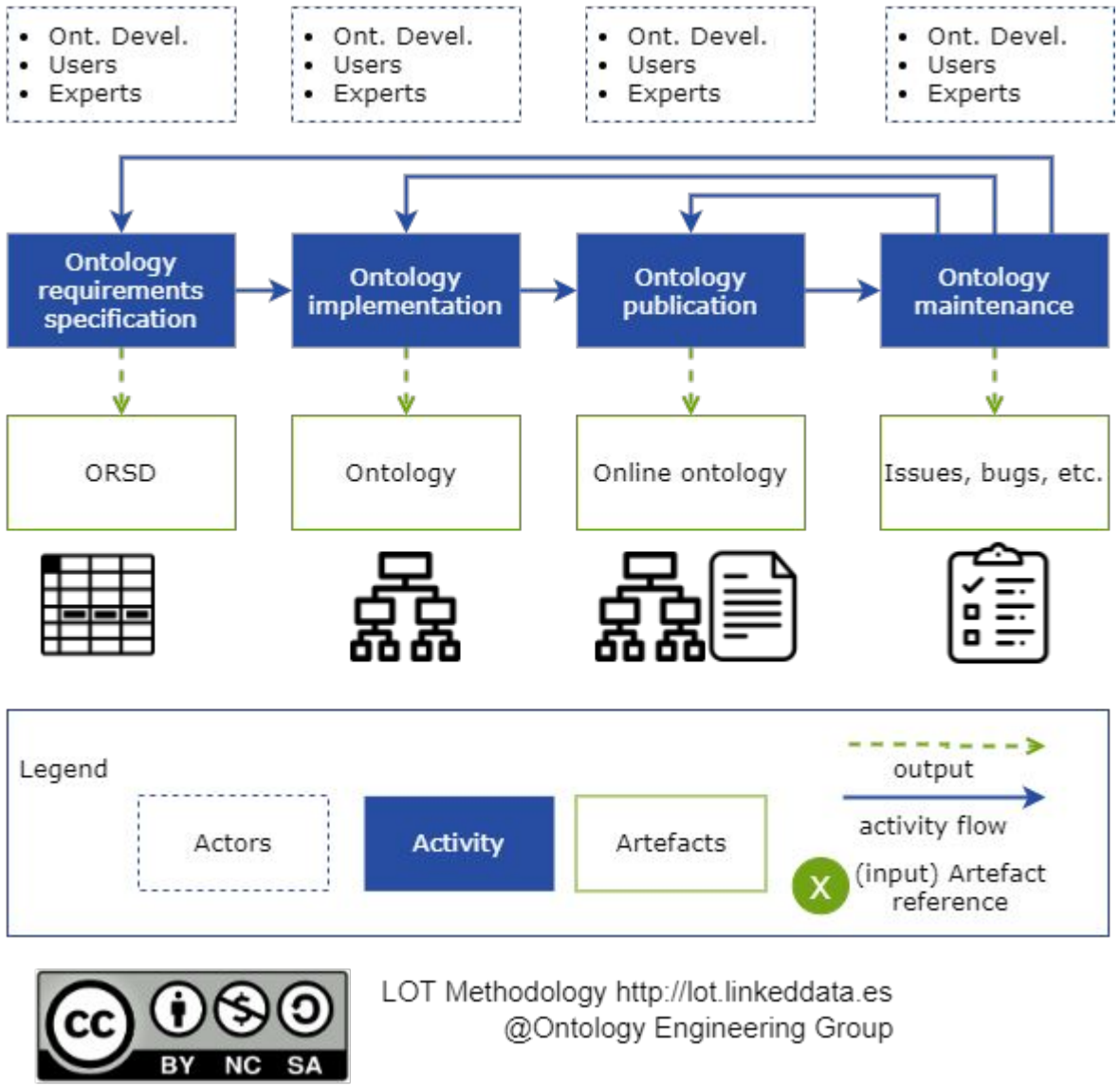
Constructing and validating your RDF graph (50min)

1) Tools for Developing Ontologies and KGs
2) RML (from CSV/XML/JSON to RDF) and SHACL (RDF validation)
3) Hands-On 2: Construct a KG as a knowledge engineer



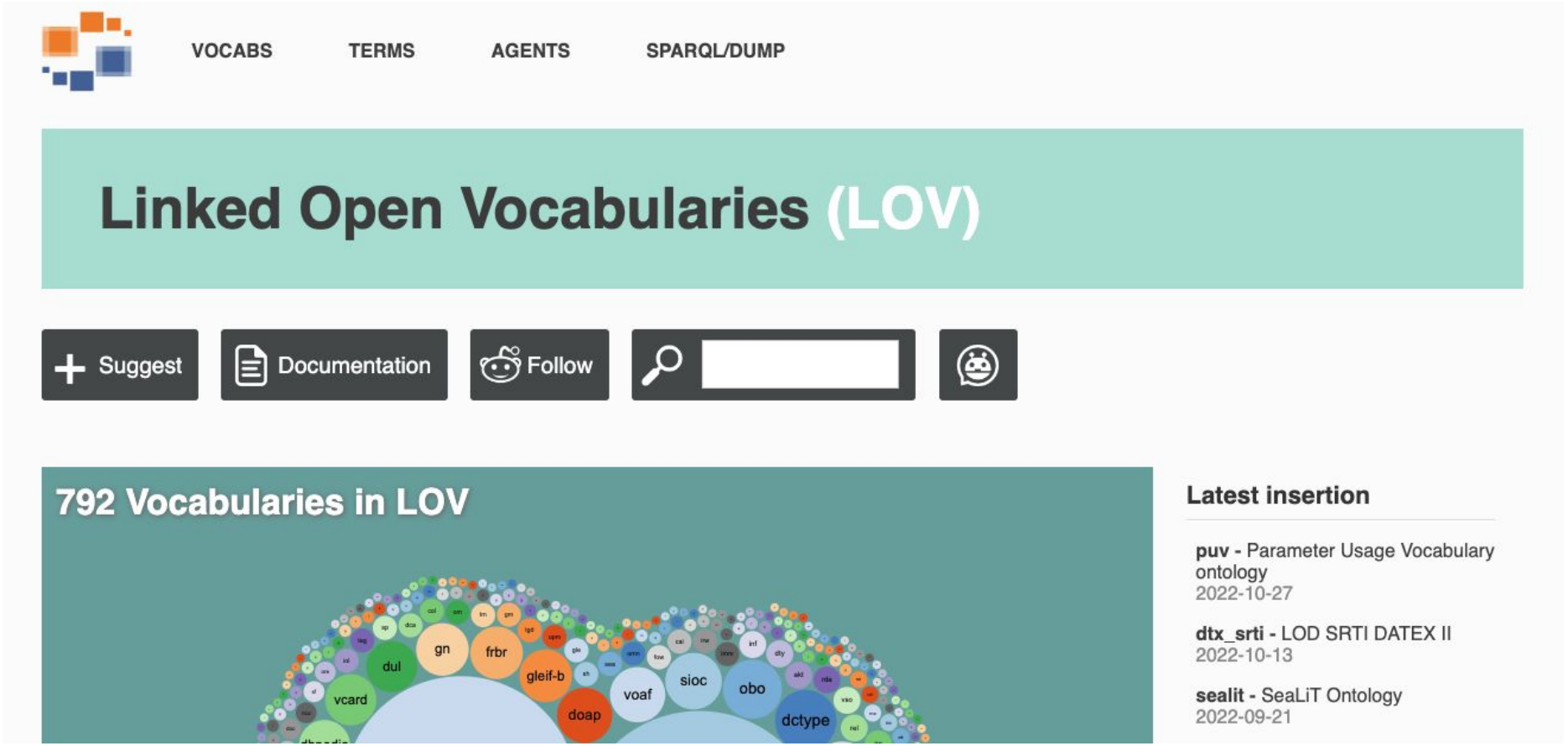
Hands-On 3: Examples of Exploiting KGs for AI (30 min)

1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project



The Linked Open Terms Methodology:
https://lot.linkeddata.es/

LOV: Linked Open Vocabularies

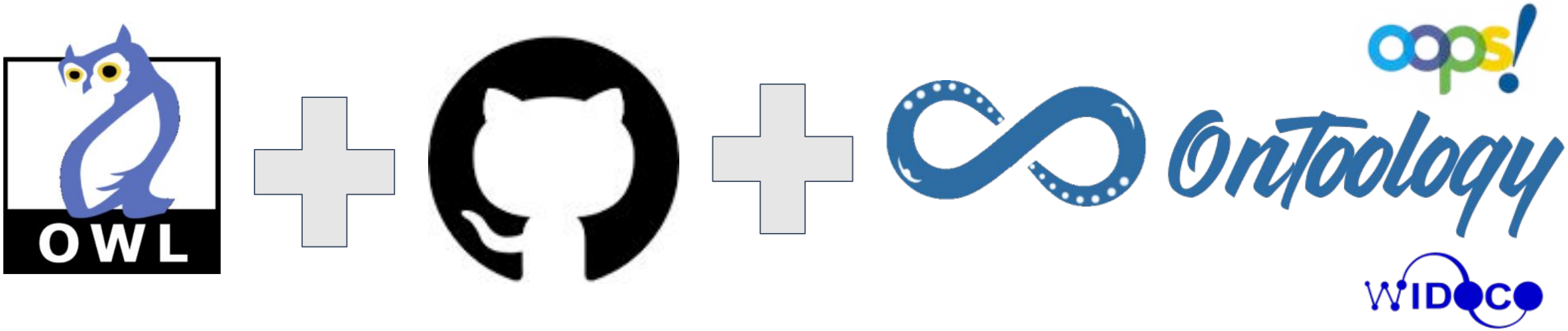


- Reuse vocabularies is a must!
- Resource: https://lov.linkeddata.es/

- Use diagrams.net instead of protégé (de-facto standard)
- Conceptualization of the ontology through human friendly user interface
- Translation from XML to OWL
- All info: https://chowlk.linkeddata.es/

Ontoology automate parts of the c**ollaborative ontology development process**. Given a repository with an owl file, OnToology will survey it and produce **diagrams, a complete documentation and validation based on common pitfalls**. OnToology also creates an issue in Github with an evaluation summary.

Let's try: http://ontoology.linkeddata.es/

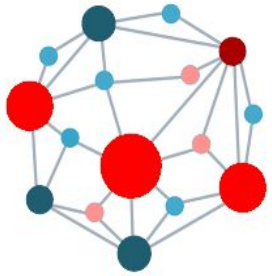


An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) RDF, OWL and SPARQL
3) RDF APIs and Triplestores
4) Hands-On 1: Querying Wikidata

Hands-On 2: Constructing and validating your RDF graph (50min)

1) Tools for Developing Ontologies and KGs
2) **RML (from CSV/XML/JSON to RDF)**
3) **SHACL (RDF validation)**

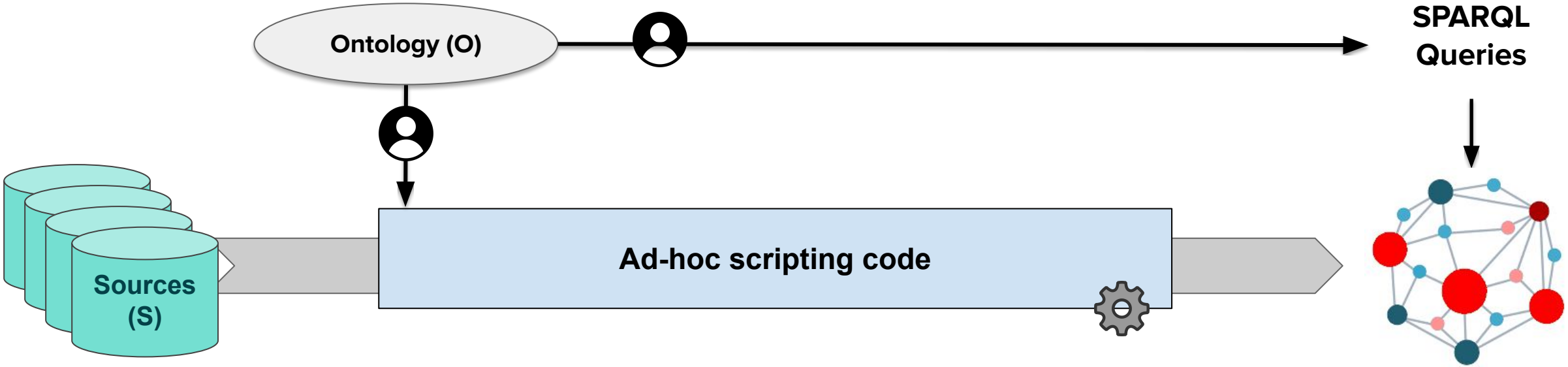


Hands-On 3: Examples of Exploiting KGs for AI (30 min)

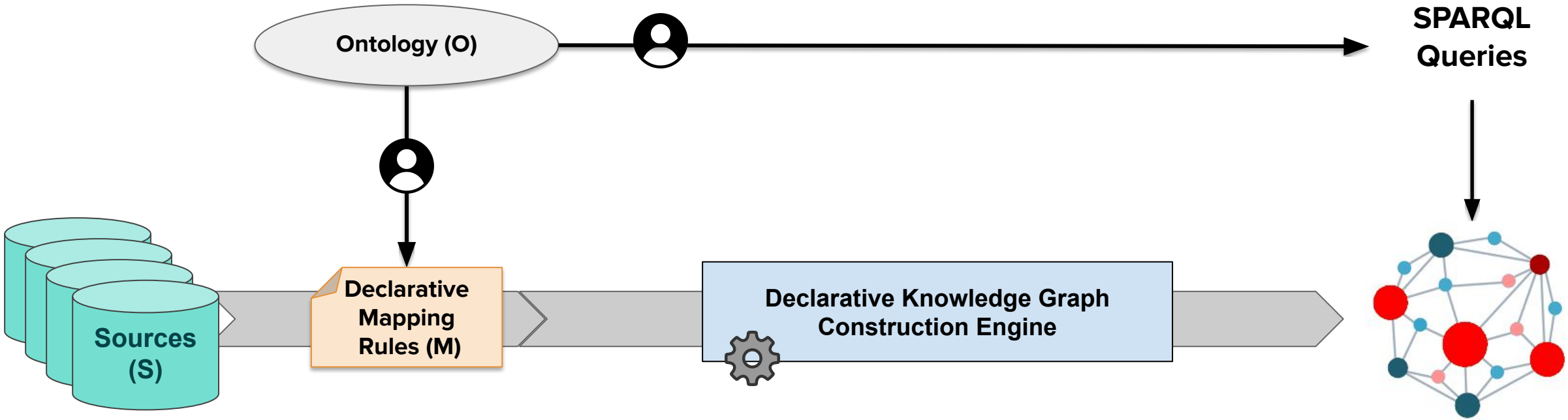
1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project

RML
The RDF Mapping Language
(or how to transform raw data into RDF)

Knowledge Graph Construction: Scripting-based



Efficient	Scalable	Maintainable	Robust	Reproducible
🤷🏽	🤷🏽	🙆	🙆	🙆



Efficient Scalable Maintainable Robust Reproducible

Example Input CSV files

EMP.csv

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	10

DEPT.csv

DEPTNO	DNAME	LOC
10	APPSERVER	NEW YORK

Desired RDF Output

```
<http://data.example.com/employee/7369> rdf:type ex:Employee.
<http://data.example.com/employee/7369> ex:name "SMITH".
<http://data.example.com/employee/7369> ex:department <http://data.example.com/department/10>.

<http://data.example.com/department/10> rdf:type ex:Department.
<http://data.example.com/department/10> ex:name "APPSERVER".
<http://data.example.com/department/10> ex:location "NEW YORK".
```

Triples Map Example

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
    sources:
    s:
    po:


 TriplesMap2:
    sources:
    s:
    po:
```

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.com/ns#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<#TriplesMap1>
    rml:logicalSource [ … ];
    rr:subjectMap [ … ];
    rr:predicateObjectMap [ … ];
    rr:predicateObjectMap [ … ];
    …
.


<#TriplesMap2>
    rml:logicalSource [ … ];
    rr:subjectMap [ … ];
    rr:predicateObjectMap [ … ];
    rr:predicateObjectMap [ … ];
    …
.
```

Logical Table: Example

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s:
   po:


 TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s:
   po:
```

```
<#TriplesMap1>
    rml:logicalSource [
       rml:source "EMP.csv";
       rml:referenceFormulation ql:CSV  ];
    rr:subjectMap [ … ];
    rr:predicateObjectMap [ … ];
    rr:predicateObjectMap [ … ];
    …
.



<#TriplesMap2>
    rml:logicalSource [
       rml:source "DEPT.csv";
       rml:referenceFormulation ql:CSV; ];
    rr:subjectMap [ … ];
    rr:predicateObjectMap [ … ];
    rr:predicateObjectMap [ … ];
    …
.
```

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s:
   po:
      - [a, ex:Employee]

 TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s:
   po:
      - [rdf:type, ex:Department]
```

```
<#TriplesMap1>
   rml:logicalSource [
      rml:source "EMP.csv";
      rml:referenceFormulation ql:CSV  ];
   rr:subjectMap [
      rr:class ex:Employee ];
   rr:predicateObjectMap [ … ];
   rr:predicateObjectMap [ … ];
   …
.



<#TriplesMap2>
   rml:logicalSource [
      rml:source "DEPT.csv";
     rml:referenceFormulation ql:CSV; ];
   rr:subjectMap [
      rr:class ex:Department ];
   rr:predicateObjectMap [ … ];
   rr:predicateObjectMap [ … ];
   …
.
```

Subject Map: Example

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s: http://data.example.com/employee/$(EMPNO)
   po:
      - [a, ex:Employee]

 TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s: http://data.example.com/department/$(DEPTNO)
   po:
      - [rdf:type, ex:Department]
```

```
<#TriplesMap1>
  rml:logicalSource [ .... ];
  rr:subjectMap [
    rr:template "http://data.example.com/employee/{EMPNO}";
    rr:termType rr:IRI; rr:class ex:Employee
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ … ] rr:objectMap [ … ] ];
```

```
<#TriplesMap2>
  rml:logicalSource [ ....];
  rr:subjectMap [
    rr:template "http://data.example.com/department/{DEPTNO}";
    rr:termType rr:IRI; rr:class ex:Department
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ … ] rr:objectMap [ … ] ];
```

Predicate Object Map

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s: http://data.example.com/employee/$(EMPNO)
   po:
      - [a, ex:Employee]

 TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s: http://data.example.com/department/$(DEPTNO)
   po:
      - [rdf:type, ex:Department]
      - [ex:name, $(DNAME)]
      - [ex:location, http://locations.org/$(LOC)]
```

```
<#TriplesMap2>
  rr:logicalTable [ rr:tableName "DEPT" ];
  rr:subjectMap [
    rr:template "http://data.example.com/employee/{DEPTNO}";
    rr:termType rr:IRI; rr:class ex:Department;
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ rr:constant ex:name;
      rr:termType rr:IRI ]
    rr:objectMap [ rr:reference "DNAME";
      rr:termType rr:Literal ]
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ rr:constant ex:location;
      rr:termType rr:IRI
      ];
    rr:objectMap [
      rr:template "http://locations.org/{LOC}";
      rr:termType rr:Literal ]
  ];
.
```

RefObjectMap: Example

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s: http://data.example.com/employee/$(EMPNO)
   po:
      - [a, ex:Employee]
      - p: ex:department
        o:
          - mapping: TriplesMap2
            condition:
              function: equal
              parameters:
                - [str1, $(DEPTID)]
                - [str2, $(DEPTNO)]
 TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s: http://data.example.com/department/$(DEPTNO)
   po:
      - [rdf:type, ex:Department]
```

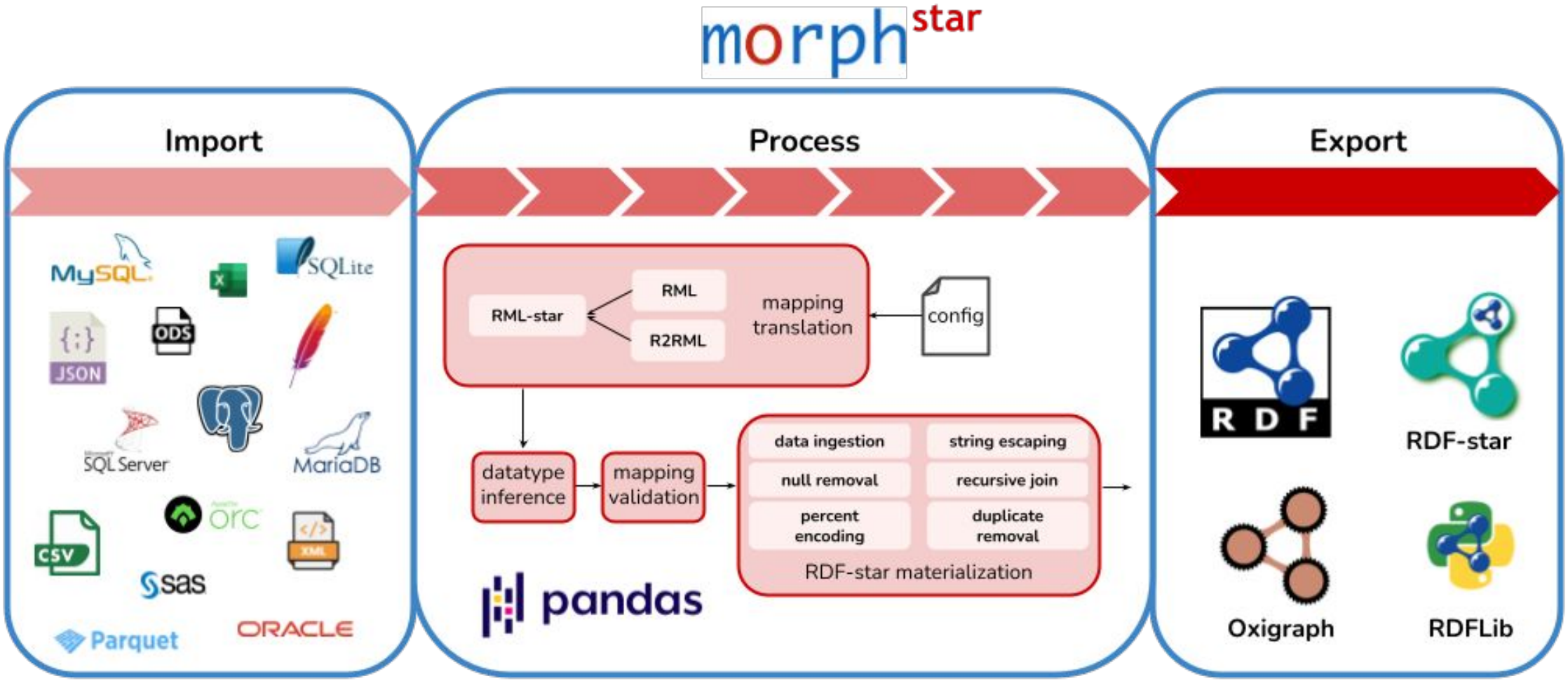
```
<#TriplesMap1>
  rr:logicalSouce [...];
  rr:subjectMap [
    rr:template "http://data.example.com/employee/{EMPNO}";
    rr:termType rr:IRI; rr:class ex:Employee
  ];
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:constant ex:department; rr:termType rr:IRI
    ]
    rr:objectMap [
      rr:parentTriplesMap <#TriplesMap2>;
      rr:joinCondition [
        rr:child "DEPTID";  rr:parent "DEPTNO";   ];
    ]
  ];
```

```
<#TriplesMap2>
  rr:logicalSource [ … ];
  rr:subjectMap [
    rr:template "http://data.example.com/department/{DEPTNO}";
    rr:termType rr:IRI; rr:class ex:Department
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ … ] rr:objectMap [ … ] ];
```

```
prefixes:
rr: <http://www.w3.org/ns/r2rml#>.
ex: <http://example.com/ns#>.
rml: <http://semweb.mmlab.be/ns/rml#>.

mappings:
 TriplesMap1:
   sources:
     - [EMP.csv~csv]
   s: http://data.example.com/employee/$(EMPNO)
   po:
      - [a, ex:Employee]
      - p: ex:department
        o:
          - mapping: TriplesMap2
            condition:
              function: equal
              parameters:
                - [str1, $(DEPTID)]
                - [str2, $(DEPTNO)]
```

```
TriplesMap2:
   sources:
     - [DEPT.csv~csv]
   s: http://data.example.com/department/$(DEPTNO)
   po:
      - [rdf:type, ex:Department]
      - [ex:name, $(DNAME)]
      - [ex:location, http://locations.org/$(LOC)]
```


morph star
Import
Process
Export
MySQL
SQLite
JSON
ODS
SQL Server
MariaDB
CSV
orc
XML
sas
Parquet
ORACLE
RML-star
RML
R2RML
mapping translation
config
datatype inference
mapping validation
data ingestion
string escaping
null removal
recursive join
percent encoding
duplicate removal
RDF-star materialization
pandas
R D F
RDF-star
Oxigraph
RDFLib

morph

https://github.com/oeg-upm/morph-kgc/

https://pypi.org/project/morph-kgc/

https://morph-kgc.readthedocs.io/

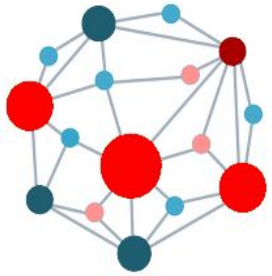
https://short.upm.es/umdvm

An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) RDF, OWL and SPARQL
3) RDF APIs and Triplestores
4) Hands-On 1: Querying Wikidata

Hands-On 2: Constructing and validating your RDF graph (50min)

1) Tools for Developing Ontologies and KGs
2) RML (from CSV/XML/JSON to RDF)
3) **SHACL (RDF validation)**



Hands-On 3: Examples of Exploiting KGs for AI (30 min)

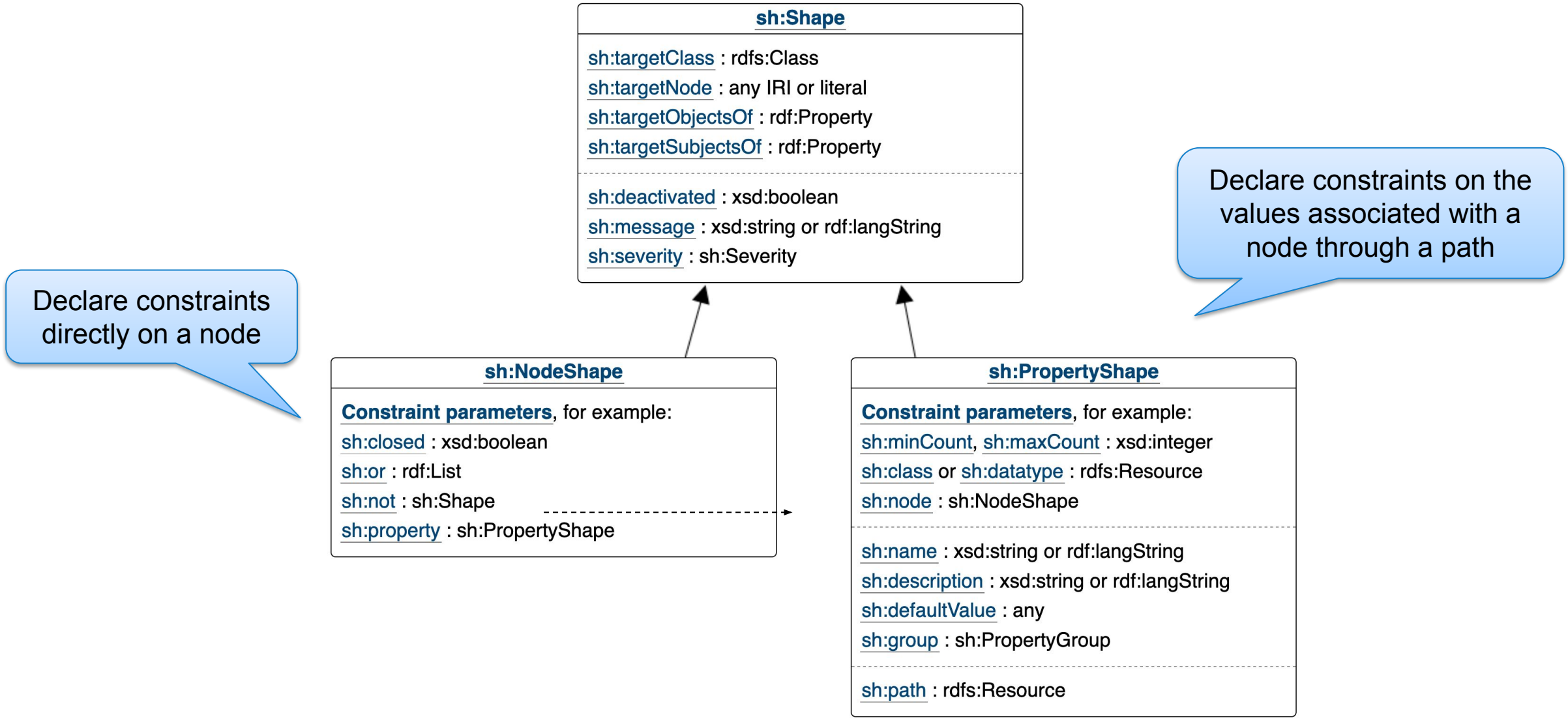
1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project

SHACL
Shapes Constraint Language
(or how to validate your RDF data against constraints)

Playground: https://shacl-playground.zazuko.com/

Main classes in the SHACL vocabulary

sh: http://www.w3.org/ns/shacl#



- Specify constraints that need to be met with respect to focus nodes
- *sh:property* associates a shape with a property shape

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:property ex:PersonShape-name .
```

Data graph

```
1867 schema:CivicStructure
9 schema:Museum
0 schema:CollegeOrUniversity
```

Shapes graph

```
ex:CivicStructureShape
  a sh:NodeShape ;
  sh:targetClass schema:CivicStructure .

ex:MuseumShape
  a sh:NodeShape ;
  sh:targetClass schema:Museum .

ex:UniversityShape
  a sh:NodeShape ;
  sh:targetClass schema:CollegeOrUniversity .
```

Validation report

```
[
  a sh:ValidationReport ;
  sh:conforms true
] .
```

- Specify constraints that need to be met with respect to nodes that can be reached from the focus node
- Reachable nodes are specified using *sh:path* either with a property IRI or a SHACL property path

```
ex:PersonShape-name
   a sh:PropertyShape ;
   sh:path ex:name ;
   sh:minCount 1.
```

```
ex:CivicStructureShape
  a sh:NodeShape ;
  sh:targetClass schema:CivicStructure ;
  sh:property ex:UPRNShape .

ex:MuseumShape
  a sh:NodeShape ;
  sh:targetClass schema:Museum ;
  sh:property ex:UPRNShape .

ex:UniversityShape
  a sh:NodeShape ;
  sh:targetClass schema:CollegeOrUniversity ;
  sh:property ex:UPRNShape .

ex:UPRNShape
  a sh:PropertyShape ;
  sh:path ec:uprn .
```

Constraint components

- Associated with (node or property) shapes to declare constraints
- A shape may have several constraint components
- A constraint component may have multiple values (all constraints apply)
- Constraint components:
  - Value type
  - Cardinality
  - Value range
  - String-based
  - Language-based
  - Property pair
  - Logical
  - Shape-based
  - Closed shapes

- Specify the set of values that a node can have

Property	Description
sh:datatype	Specifies that values must be literals with some datatype
sh:class	Specifies that values must be SHACL instances of some class
sh:nodeKind	Possible values: sh:BlankNode, sh:IRI, sh:Literal, sh:BlankNodeOrIRI, sh:BlankNodeOrLiteral, sh:IRIOrLiteral
sh:in	Enumerates the value nodes that a property is allowed to have
sh:hasValue	A node must have a given value

- Specify restrictions on the minimum and maximum number of distinct value nodes
- The default cardinality for property shapes is {0,unbounded}

Property	Description
sh:minCount	Restricts minimum number of value nodes. If not defined, there is no restriction (no minimum)
sh:maxCount	Restricts maximum number of value nodes. If not defined, there is no restriction (unbounded)

- Specify conditions on the string representation of value nodes

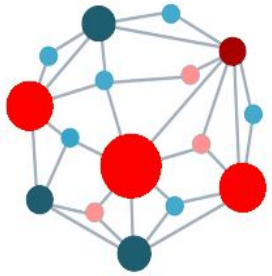
Property	Description
sh:minLength	The minimum string length; if 0, there is no length restriction but fails with blank nodes
sh:maxLength	The maximum string length
sh:pattern	Regular expression that must be matched (SPARQL REGEX)
sh:flags	Optional string of regular expression flags
sh:languageIn	Allowed language tags
sh:uniqueLang	No pair of value nodes may use the same language tag

An introduction to the Semantic Web (1,5h)

1) What is a KG? Why KGs now?
2) RDF, OWL and SPARQL
3) RDF APIs and Triplestores
4) Hands-On 1: Querying Wikidata

Hands-On 2: Constructing and validating your RDF graph (50min)

1) Tools for Developing Ontologies and KGs
2) RML (from CSV/XML/JSON to RDF)
3) SHACL (RDF validation)



Hands-On 3: Examples of Exploiting KGs for AI (30 min)

1) Entity recognition with KGs
2) RDF generation from text with OpenAI
3) Semantic Table Interpretation
4) The Drugs4Covid Project

Semantic Answer Type, Entity, and Relation Linking Task (SMART2022) 3rd

Organizer: Nandana Mihindukulasooriya, Mohnish Dubey, Alfio Gliozzo, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Ricardo Usbeck, Gaetano Rossiello and Debayan Banerjee

The challenge, SMART 2022, consists of three tasks Semantic Answer Type prediction task, Entity and Relation Linking tasks. Given a question in natural language, the three tasks in this challenge are to predict the answer type for a given question, identify entities and relations in the questions using a target ontology (e.g., DBpedia or Wikidata). This is a continuation of SMART2020, and SMART2021 challenges.

Website: https://smart-task.github.io/

DBpedia Spotlight

https://www.dbpedia-spotlight.org/

FαʃCθη 2.0

https://labs.tib.eu/falcon/falcon2/

Knowledge Base Construction from Pre-trained Language Models (LM-KBC) NEW

Organizer: Sneha Singhania, Tuan-Phong Nguyen, Simon Razniewski

Pre-trained language models (LMs) have advanced a range of semantic tasks and have also shown promise for knowledge extraction from the models itself. Although several works have explored this ability in a setting called probing or prompting, the viability of knowledge base construction from LMs has not yet been explored. In this challenge, participants are asked to build actual knowledge bases from LMs, for given subjects and relations. In crucial difference to existing probing benchmarks like LAMA (Petroni et al., 2019), we make no simplifying assumptions on relation cardinalities, i.e., a subject-entity can stand in relation with zero, one, or many object-entities. Furthermore, submissions need to go beyond just ranking the predictions, and materialize outputs, which are evaluated by established KB metrics of precision and recall. The challenge comes with two tracks: (i) a BERT-type LM track with low computational requirements, and (ii) an open track, where participants can use any LM of their choice.

Website: http://lm-kbc.github.io

Knowledge Graph Generation from Text

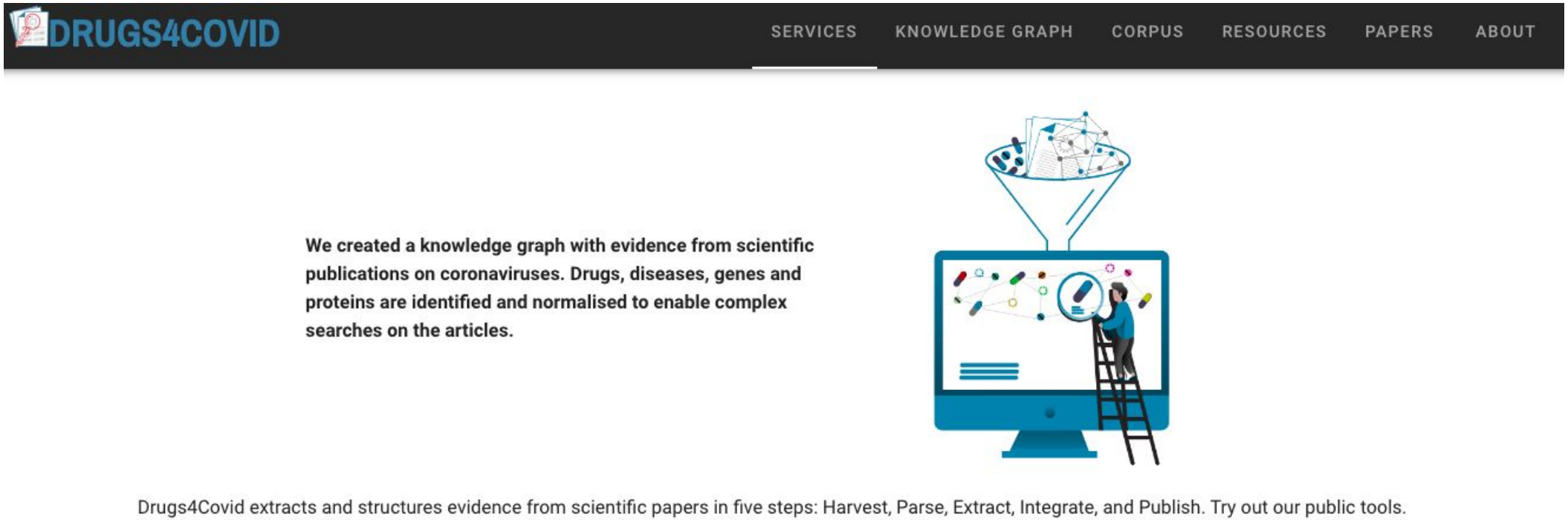
Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022) 4th

Organizer: Nora Abdelmageed, Jiaoyan Chen, Vincenzo Cutrona, Vasilis Efthymiou, Oktie Hassanzadeh, Madelon Hulsebos, Ernesto Jimenez-Ruiz, Juan F. Sequeda, Kavitha Srinivas

This challenge aims at benchmarking systems dealing with the tabular data to KG matching problem, so as to facilitate their comparison on the same basis and the reproducibility of the results.

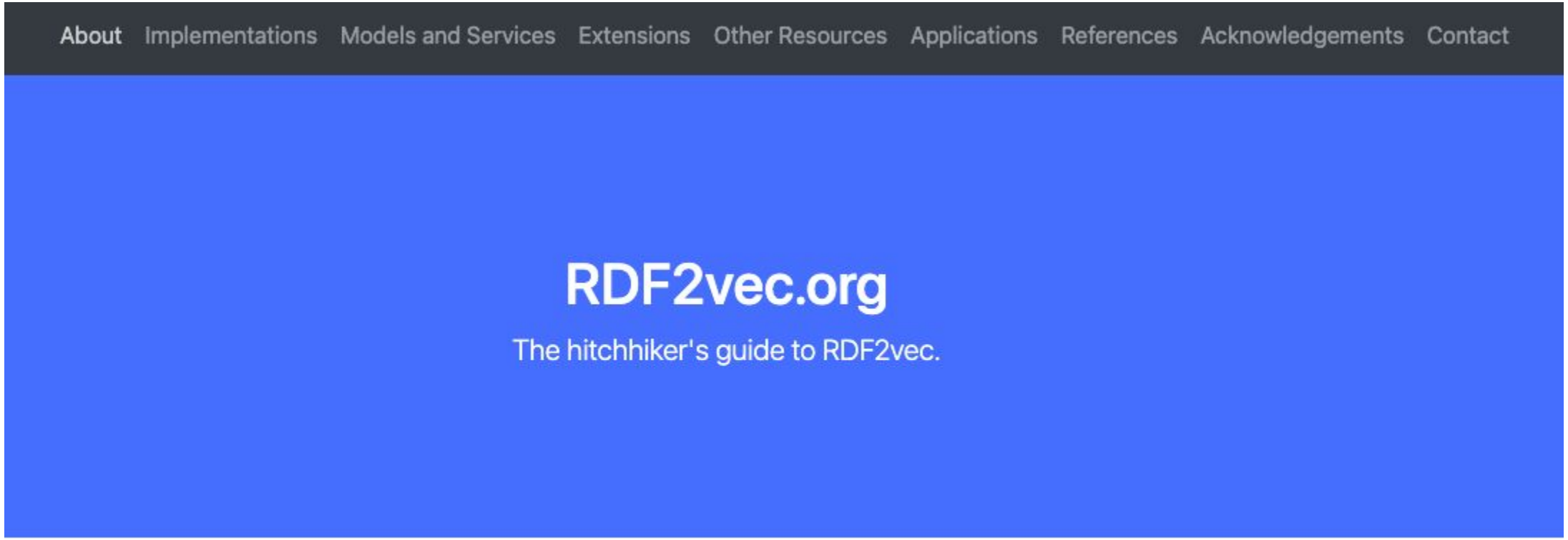
Website: https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

https://dbpedia.mtab.app/mtab



https://drugs4covid.oeg.fi.upm.es/

https://drugs4covid.oeg.fi.upm.es/services/bio-qa



http://rdf2vec.org/

Knowledge Graphs and Ontologies

A Practical Introduction to the Semantic Web

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

david.chaves@upm.es dchavesf 08/11/2022 USC-CiTIUS