

Shaping Knowledge Graphs

ISWC'24 Tutorial

Jose Emilio Labra Gayo

WESO Research group

University of Oviedo, Spain



About me...

<http://labra.weso.es>

Main researcher at WESO (Web Semantics Oviedo)

Some books:

"*Web semántica*" (in Spanish), 2012

"*Validating RDF data*", 2017

"*Knowledge Graphs*", 2021

...and software:

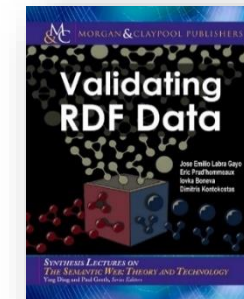
[SHacEX](#) (Scala library, implements ShEx & SHACL)

[RDFShape](#) (RDF playground)

[rudof](#) (RDF & Shapes library in Rust)

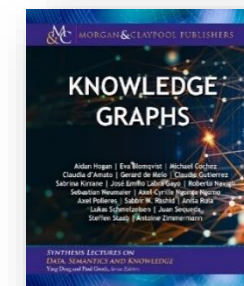


2012



2017 HTML version:

<http://book.validatingrdf.com>



2021, HTML version

<https://kgbook.org/>

Contents



Introduction to Knowledge graphs

Types of Knowledge Graphs:

RDF, Property graphs, Wikibase, RDF-Star

Shaping RDF: ShEx & SHACL

Shaping Other types of Knowledge graphs

Wikibase and Wikidata graphs

Property Graphs

RDF-Star

RDF with nodes as properties

Applications:

Inferring shapes from data, Knowledge Graphs Subsets, etc.

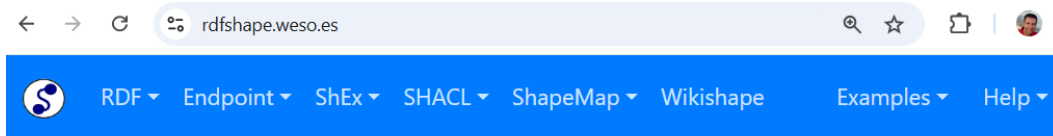
Tutorial materials

<https://github.com/validatingrdf/Tutorial>

Interactive – through the web

RDFShape: RDF playground + ShEx/SHACL support

Wikishape: Tailored to Wikibase



RDFShape

RDFShape is a playground for RDF data conversion, validation and visualization, among other features.

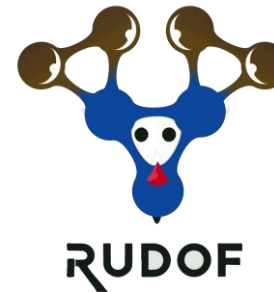
It supports the following tasks:

- [RDF](#) conversion between different formats like [Turtle](#) and [JSON-LD](#)
- RDF validation using [ShEx](#) (Shape Expressions) and [SHACL](#) (Shapes Constraint Language)
- RDF querying with [SPARQL](#)
- [RDFS](#) and [OWL](#) inference

You may jump straight in or check the examples in the navbar yourself :)

Command line

[rudof](#) (Rust): Binaries in Windows, Linux, Mac



Other command line tools:

[Shex-s](#): ShEx, WShEx

[SHACL-s](#): SHACL

[wds](#): Subsets

Interactive (Jupyter)

[rudof](#) (Rust): Python bindings

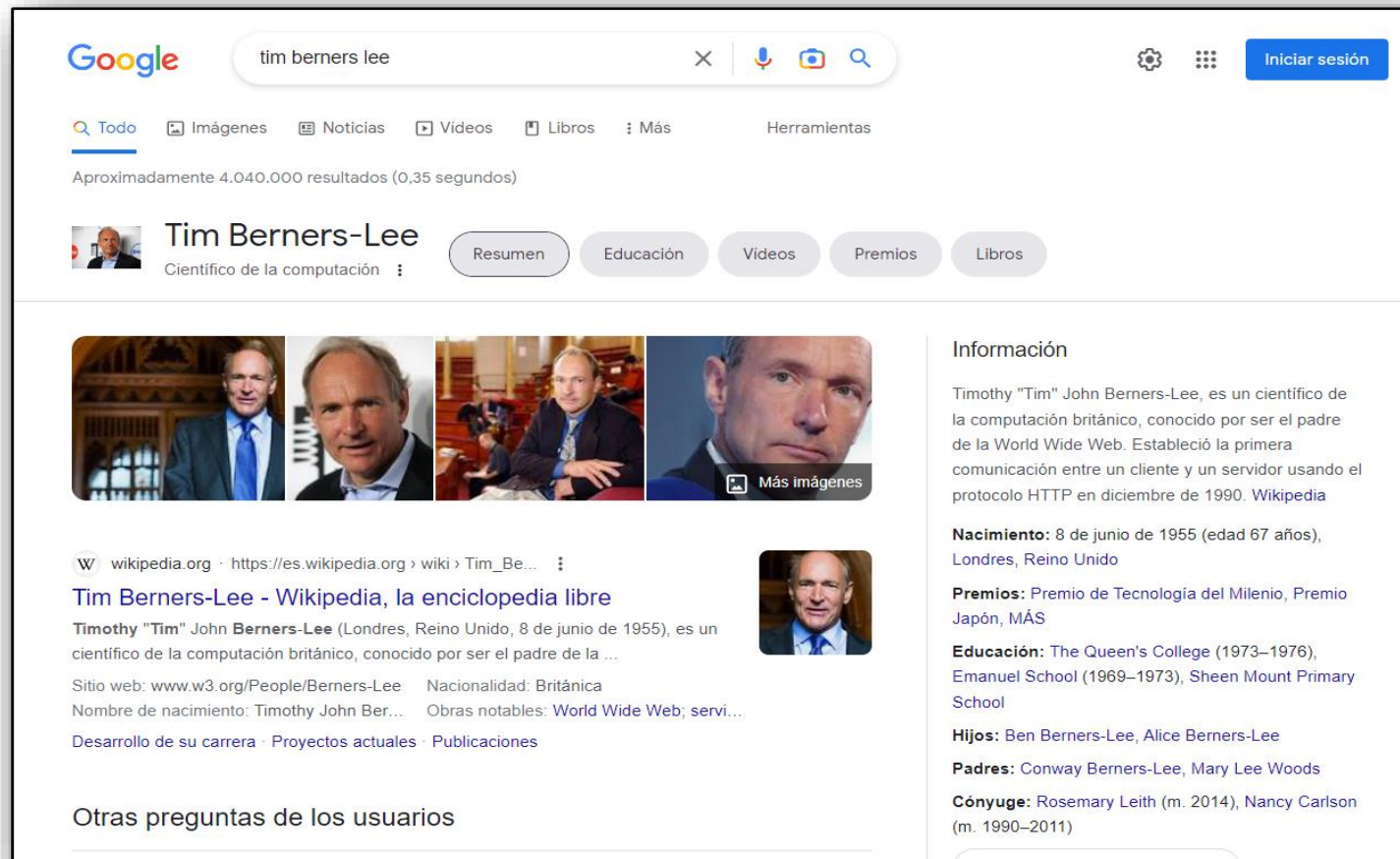
[PyShEx](#)

[PySHACL](#)

[Example Google Colab](#)

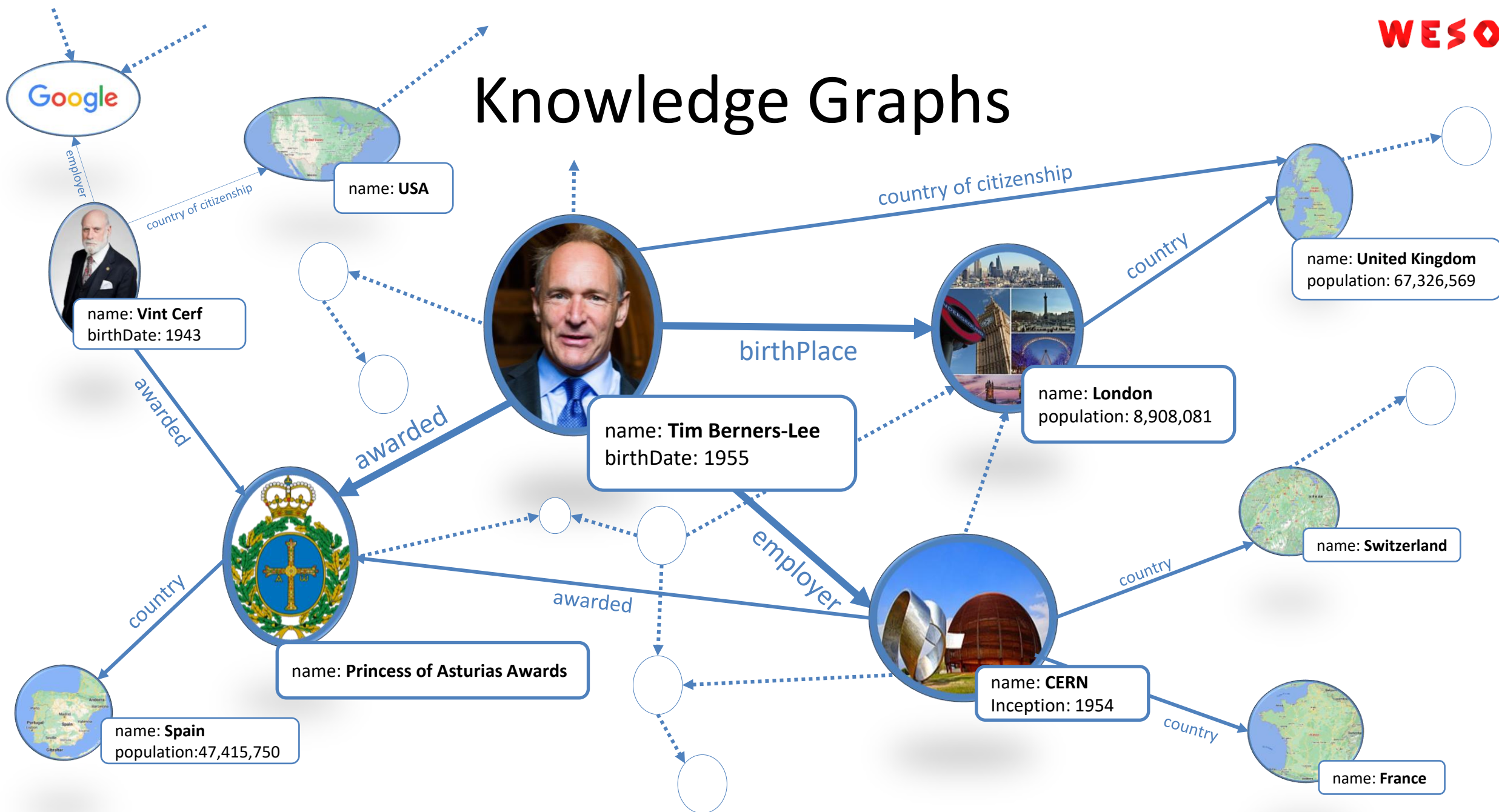
Knowledge Graphs

Current notion of Knowledge Graphs, popular after Google, 2012*



Link: <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>

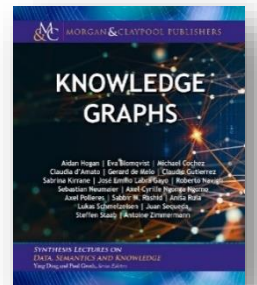
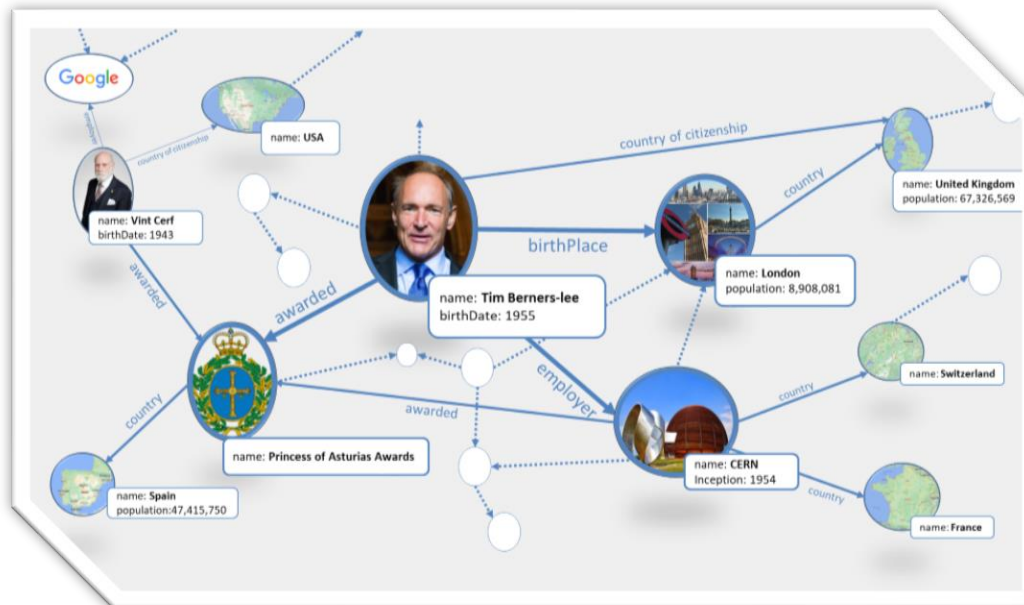
Knowledge Graphs



Knowledge Graphs

Knowledge graph = *a **graph of data***

*intended to accumulate and convey **knowledge** of the real world
whose **nodes** represent **entities** of interest and
whose **edges** represent **relations** between these entities.*



<https://kgbook.org/>

Applications of Knowledge Graphs

Improve search results

Question answering

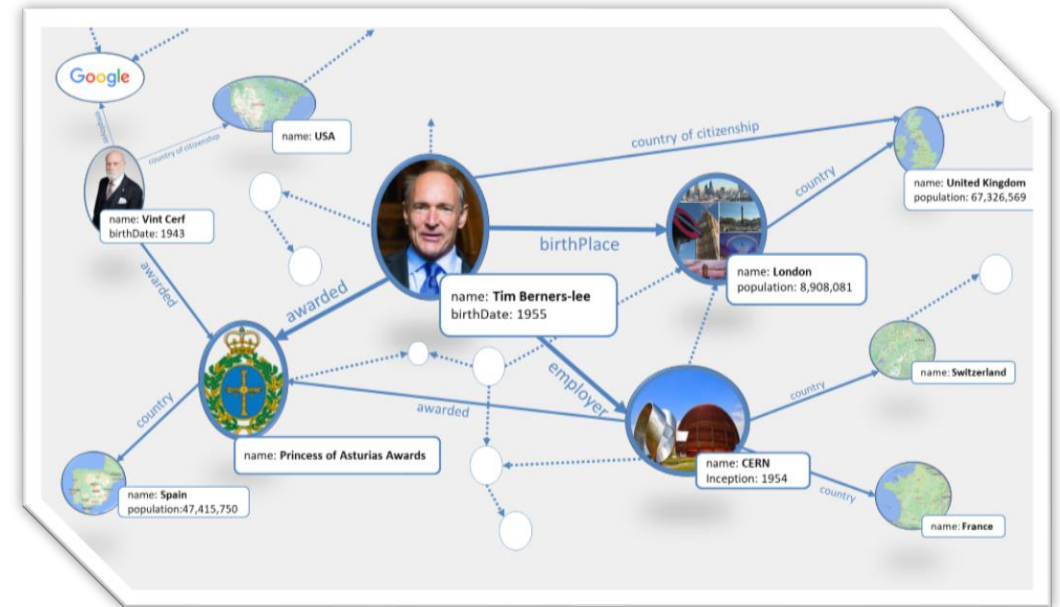
Data governance

Handling heterogeneous data

Recommender systems

Chatbots and NLP

...



Contents



Introduction to Knowledge graphs

Types of Knowledge Graphs:

RDF, Property graphs, Wikibase, RDF-Star

Shaping RDF: ShEx & SHACL

Shaping other types of Knowledge graphs:

Wikibase and Wikidata graphs

Property Graphs

RDF-Star

Applications:

Inferring shapes from data, Knowledge Graphs Subsets, etc.

Types of Knowledge Graphs

Open Knowledge Graphs

Cross-domain: Wikidata, Dbpedia, Freebase, YAGO, ...

Domain specific

Academic: Open citations, SciGraph, Microsoft Academic Knowledge Graph, ...

Life sciences: UniProt, PubChem, PDB, ...

Government: EU Knowledge graph, ...

...

Enterprise Knowledge Graphs

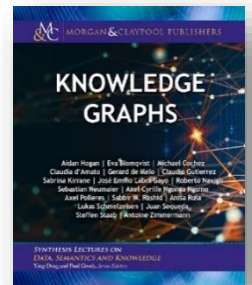
Web search: Google, Bing...

Commerce: AirBnb, Amazon, eBay, Uber,...

Social networks: Linkedin, Facebook,...

Finance: Banca d'Italia, Bloomberg, Wells Fargo, Capital One,...

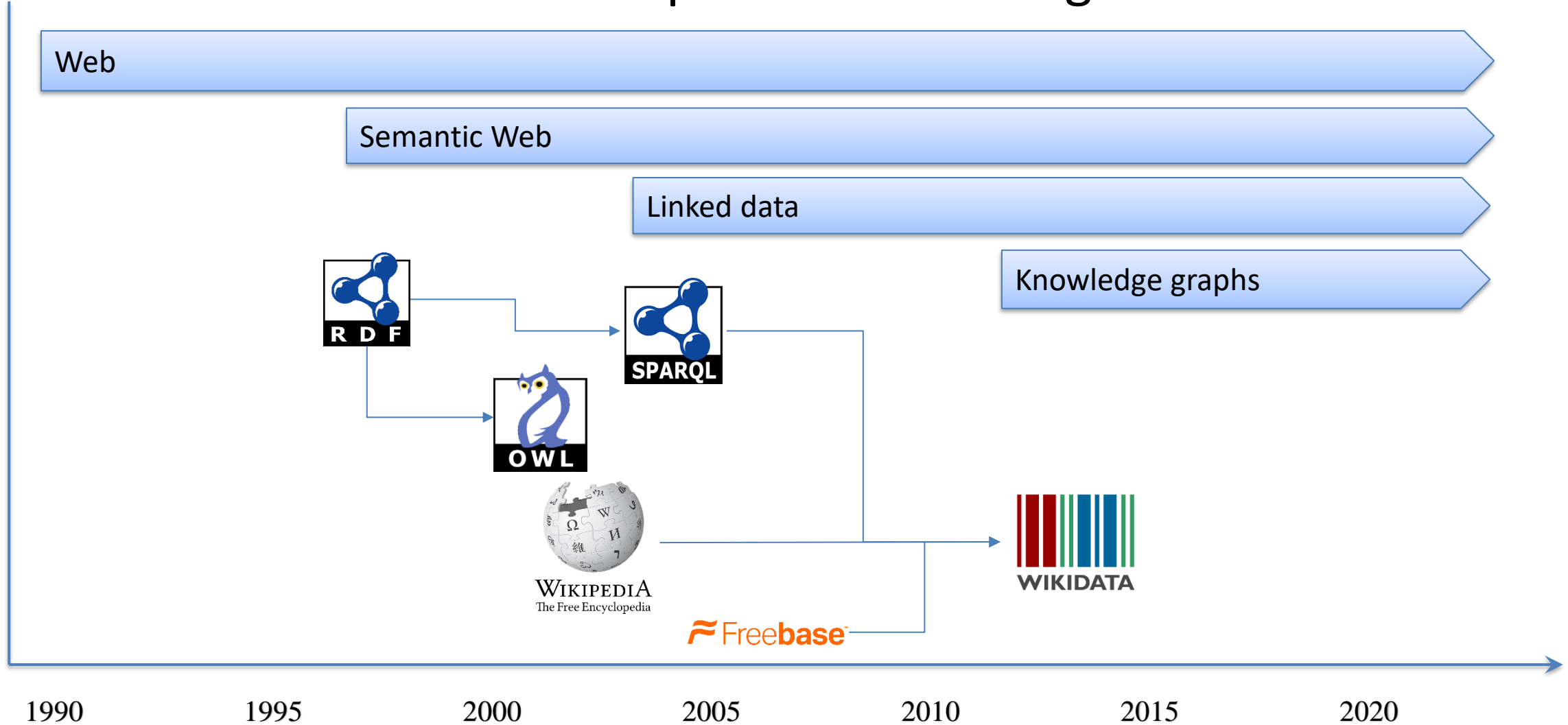
...



<https://kgbook.org/>

Evolution

Timeline with some concepts and technologies...



Knowledge Graphs models

3 popular knowledge graphs models

- RDF based
- Property graphs (Labeled property graphs)
- Wikibase graphs

RDF

Resource Description Framework

Lingua franca of the Semantic Web

1997 1st public 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/>, RDF Model and Syntax

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

2008 - SPARQL 1.0, <https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>

2014 - RDF 1.1 <https://www.w3.org/TR/rdf11-concepts/>, SPARQL 1.1, JSON-LD

2017 - SHACL 1.0 <https://www.w3.org/TR/2017/REC-shacl-20170720/>

2024 - RDF 1.2 <https://www.w3.org/TR/rdf12-concepts> Statements about triples (RDF-Star)



RDF graphs

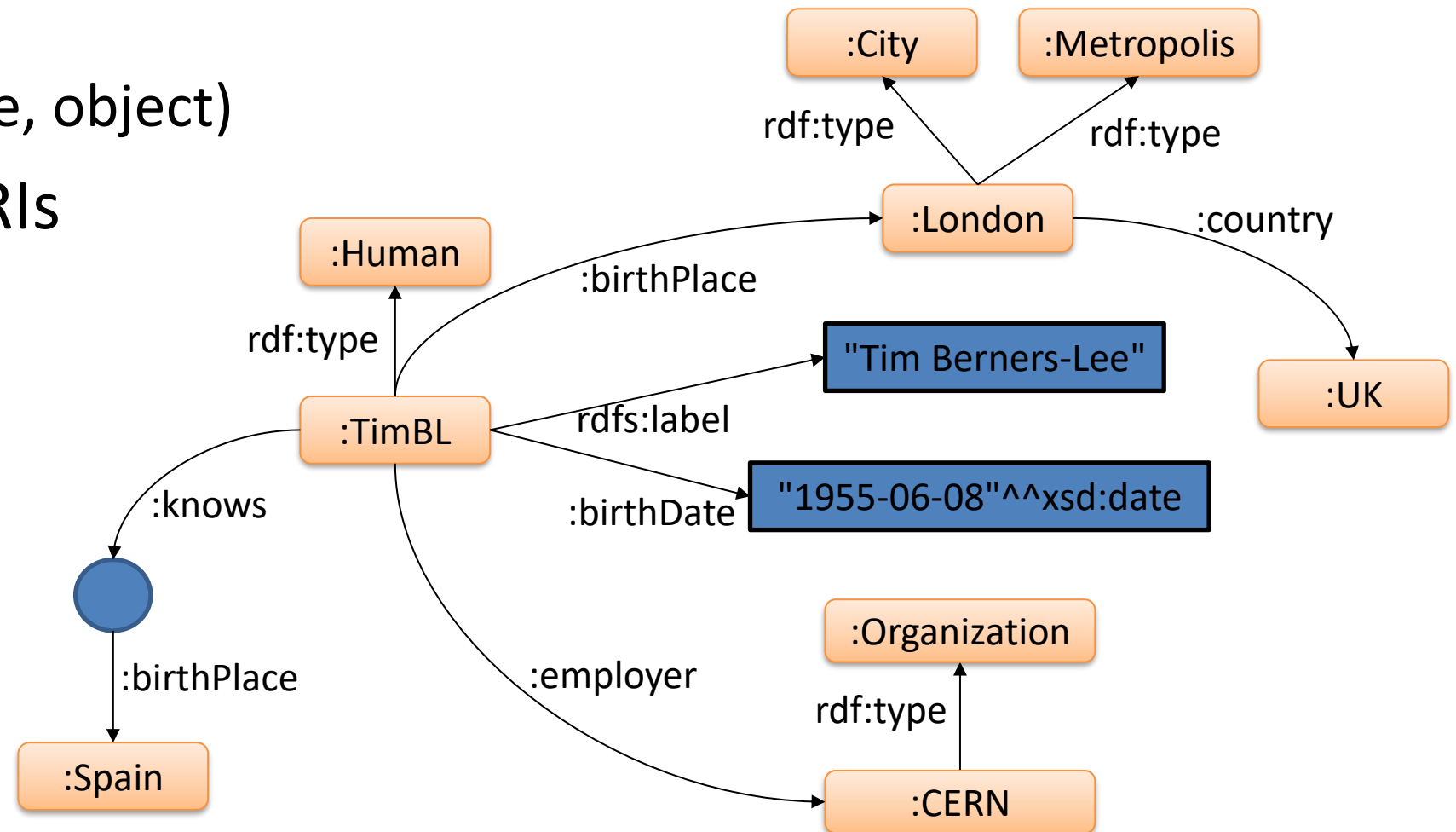
Based on triples

(subject, predicate, object)

Most nodes are URIs

Interoperability

Simple & flexible





RDF ecosystem

One data model, several syntaxes: Turtle, N-Triples, JSON-LD

Vocabularies: RDF Schema, OWL, SKOS, etc.

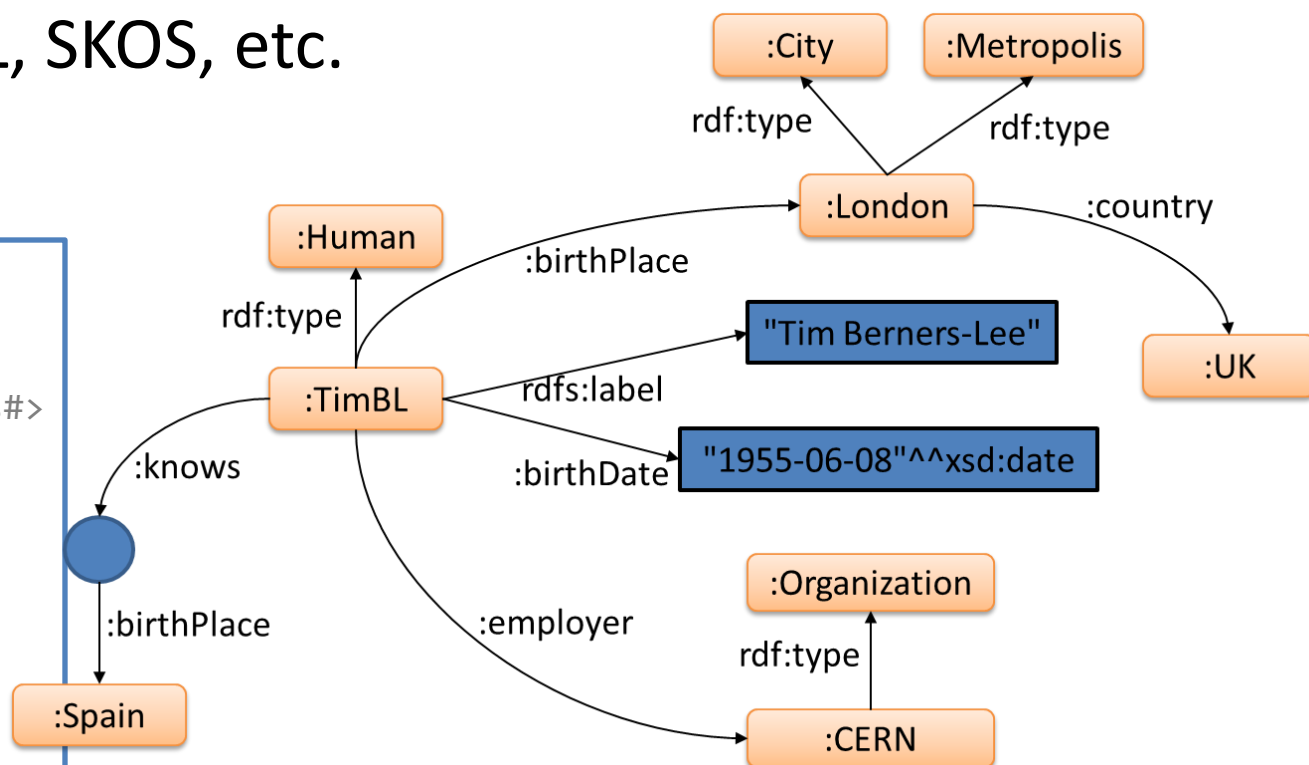
Turtle

```
prefix :      <http://example.org/>
prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
prefix xsd:   <http://www.w3.org/2001/XMLSchema#>
prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
:timbl rdfs:type      :Human ;
       :birthPlace   :london ;
       rdfs:label    "Tim Berners-Lee" ;
       :birthDate    "1955-06-08"^^xsd:date ;
       :employer     :CERN ;
       :knows        _:1 .

:london rdfs:type     :City, :Metropolis ;
       :country      :UK .

:CERN   rdfs:type     :Organization .
_:1     :birthPlace  :Spain .
```



Try it:

<https://rdfshape.weso.es/link/17313171788>



RDF ecosystem: SPARQL

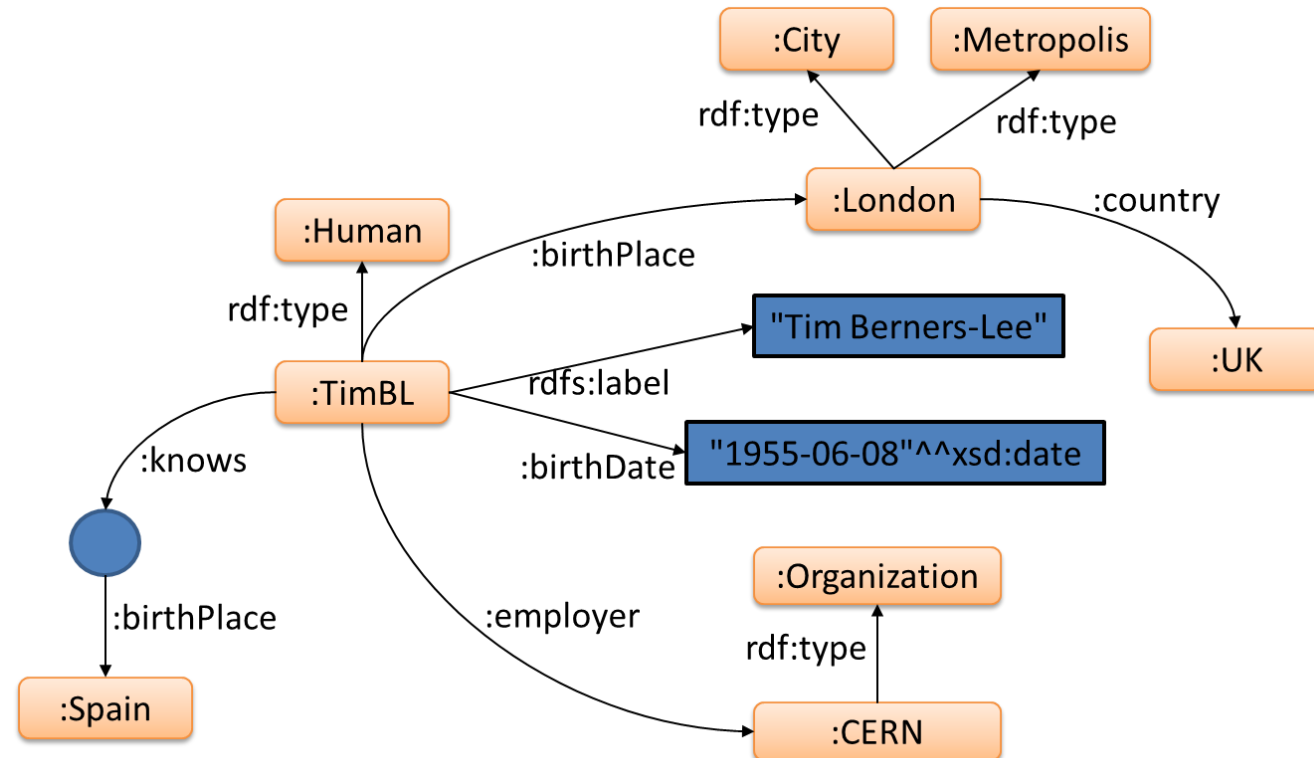
SPARQL = RDF query language and protocol

Enables the creation of SPARQL endpoints

```
select ?person ?date ?country where {  
  ?person :birthDate ?date .  
  ?person :birthPlace ?p .  
  ?p      :country ?country  
}
```

?person	?date	?country
:timbl	1955-06-08	:UK

Try it: <https://rdfshape.weso.es/link/17313175698>



RDF1.2 (RDF-Star)

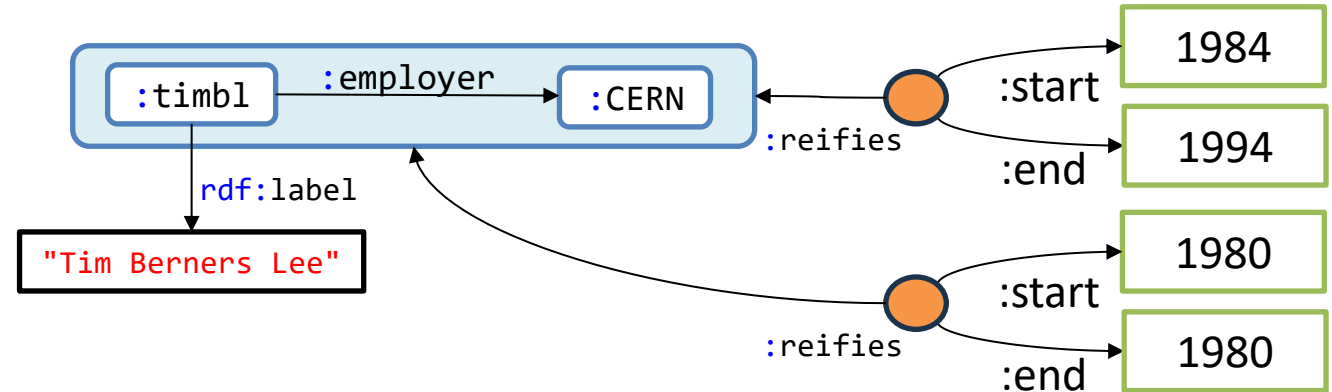
Currently under discussion (<https://github.com/w3c/rdf-star-wg/wiki>)

Add statements about triples

Reifiers

```
:timbl rdfs:label "Tim Berners Lee" ;
      :employer :CERN
      { | :start "1984" ;
          :end   "1994" | }
      { | :start "1980" ;
          :end   "1980" | } .
```

```
:timbl rdfs:label "Tim Berners Lee" ;
      :employer :CERN .
_:r rdf:reifies << :timbl :employer :CERN >> .
_:r :start "1984" ;
   :end   "1994" .
_:s rdf:reifies << :timbl :employer :CERN >> .
_:s :start "1980" ;
   :end   "1980" .
```



Property graphs

Also called *labeled property graphs*

Since ~2007, very popular model in industry

Neo4j, Amazon Neptune, Oracle, etc

Several query languages: Cypher, Gremlin, PGQL, ...

2024 - GQL has been published

Recent publication of ISO/IEC FDIS 39075

Developed by ISO/IEC JTC1 SC32 WG3: the “SQL” committee

Influenced by Cypher, PGQL, etc.

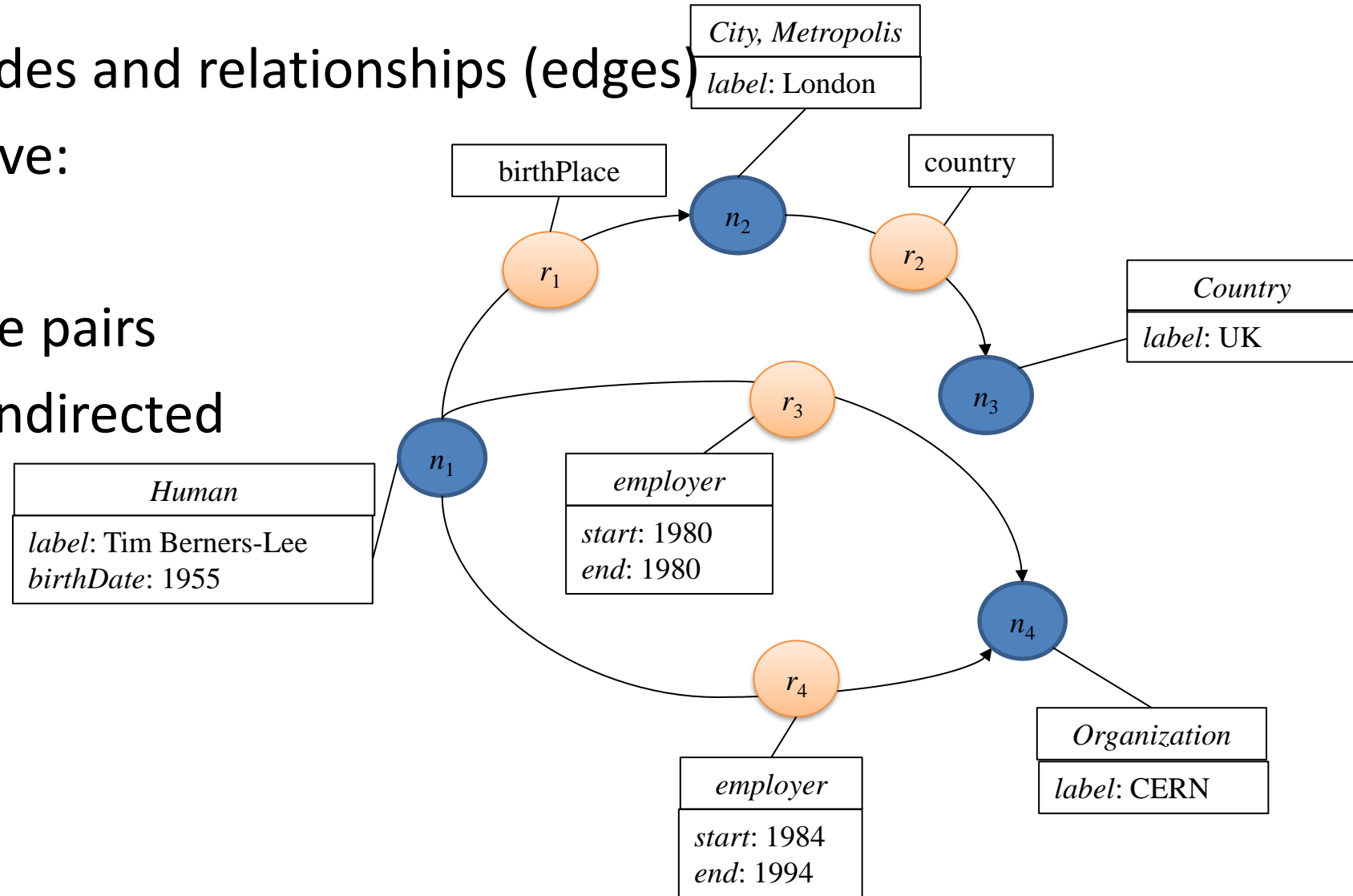
Property graphs

Graph structure with nodes and relationships (edges)

Nodes and edges can have:

- Labels
- A set of property-value pairs

Edges can be directed/undirected



Wikibase graphs

Popularized by Wikidata

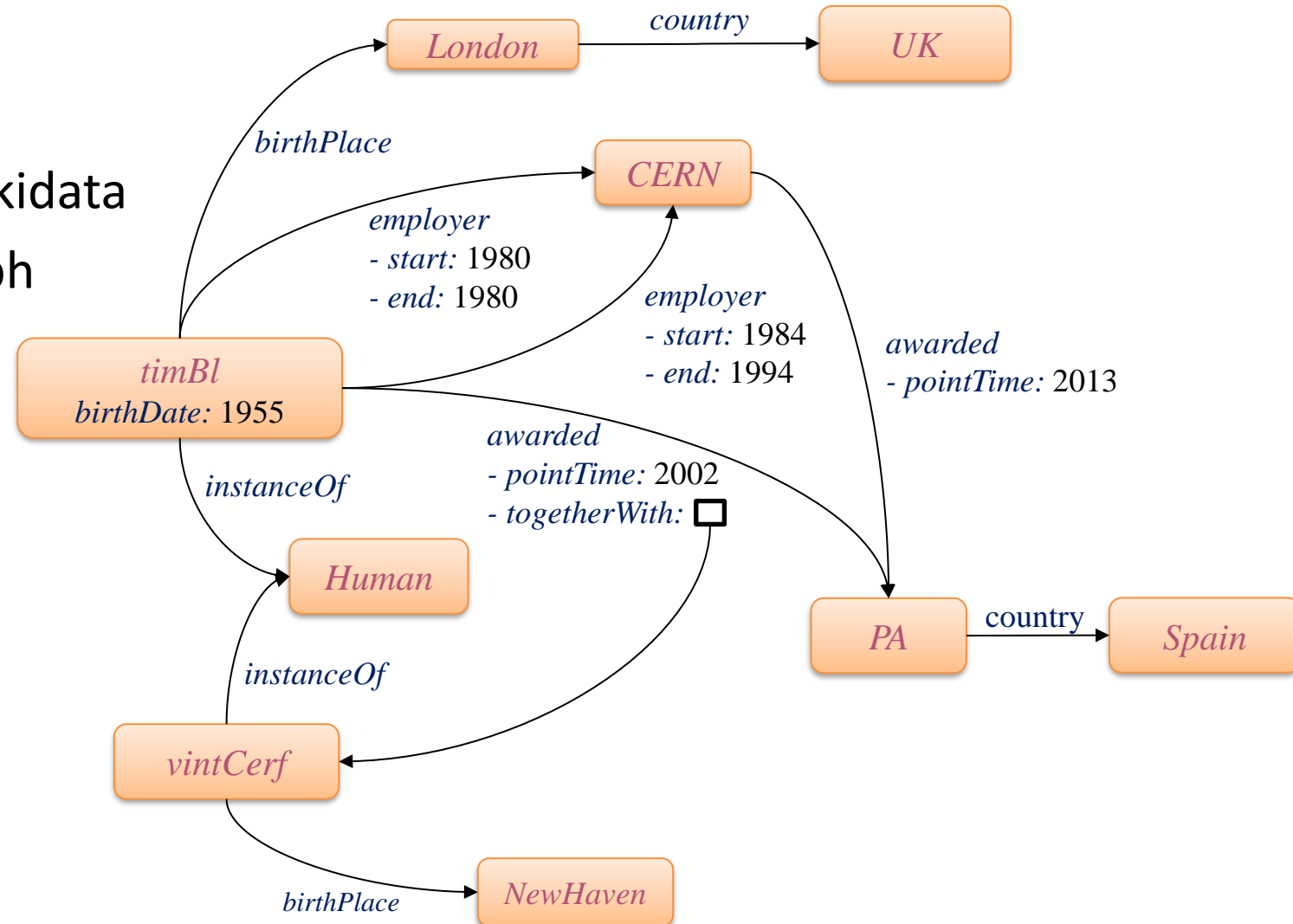
Wikibase = software supporting Wikidata

The values can be nodes in the graph

Example:

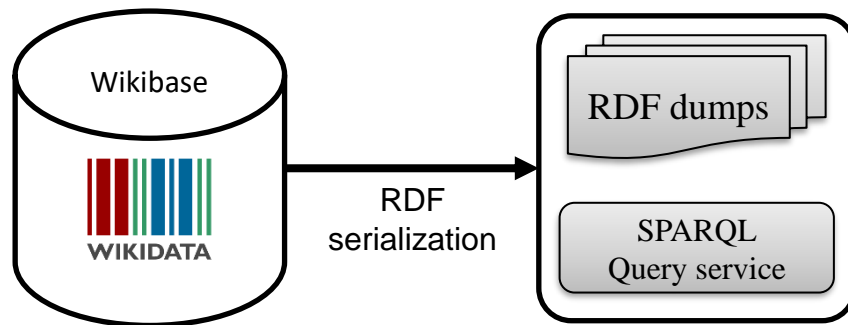
Tim Berners Lee

<http://www.wikidata.org/entity/Q80>



Wikibase graphs and SPARQL

Wikibase graphs generate RDF serializations for each item
SPARQL endpoint and Query service available



```
select ?name ?date ?country where {  
  wd:Q80 wdt:P1559 ?name .  
  wd:Q80 wdt:P569 ?date .  
  wd:Q80 wdt:P19 ?place .  
  ?place wdt:P17 ?country  
}
```

?name	?date	?country
Tim Berners-lee	1955-06-08	:UK

Contents

Introduction to Knowledge graphs

Types of Knowledge Graphs:

RDF, Property graphs, Wikibase, RDF-Star

Shaping RDF: ShEx & SHACL

Shaping other types of Knowledge graphs:

Wikibase and Wikidata graphs

Property Graphs

RDF-Star

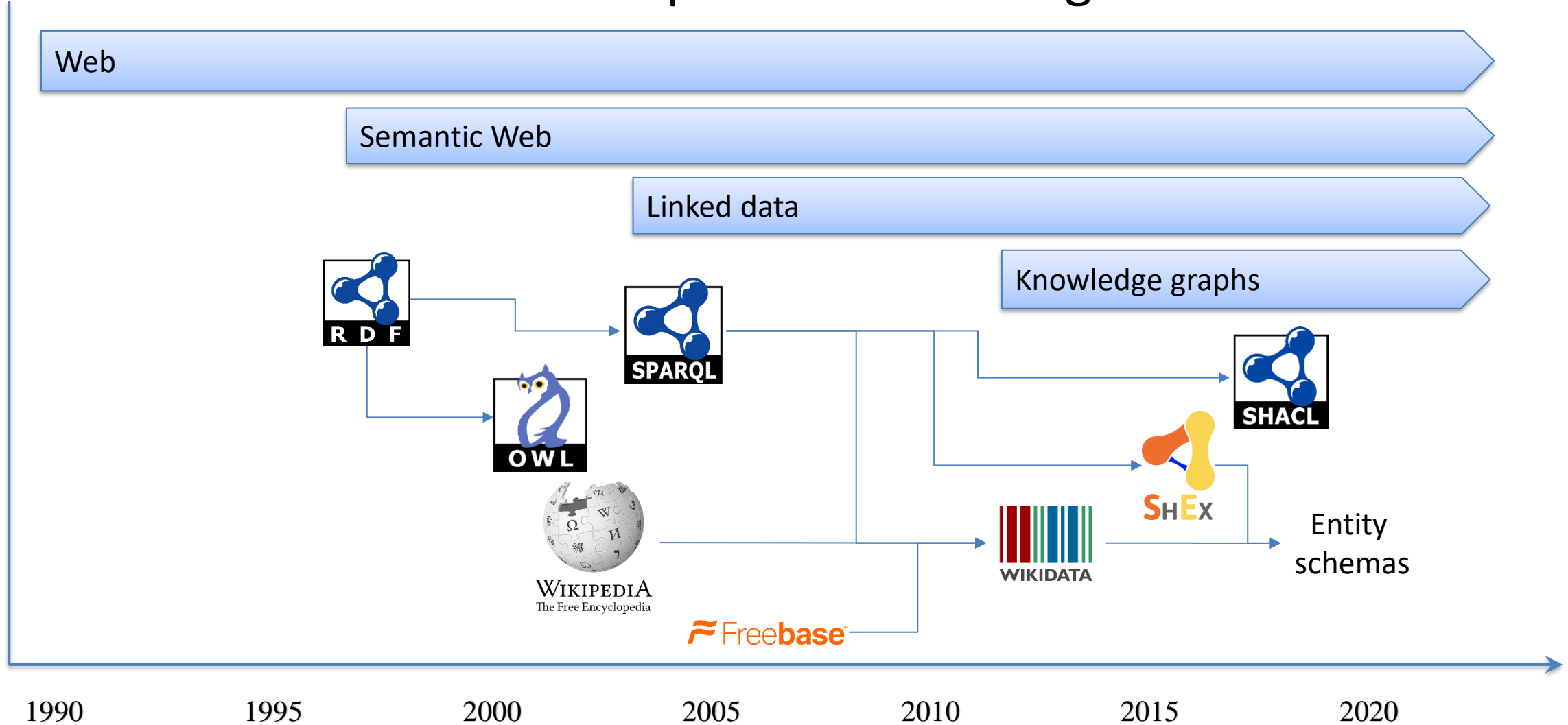
Applications:

Inferring shapes from data, Knowledge Graphs Subsets, etc.



Evolution

Timeline with some concepts and technologies...





RDF, the good parts...

RDF as an integration language

RDF as a *lingua franca* for semantic web and linked data

Basis for knowledge representation

RDF flexibility

Data can be adapted to multiple environments

Reusable data by default

RDF tools

RDF data stores & SPARQL

Several serializations: Turtle, JSON-LD, RDF/XML...

Can be embedded in HTML (Microdata/RDFa)





RDF, the other parts

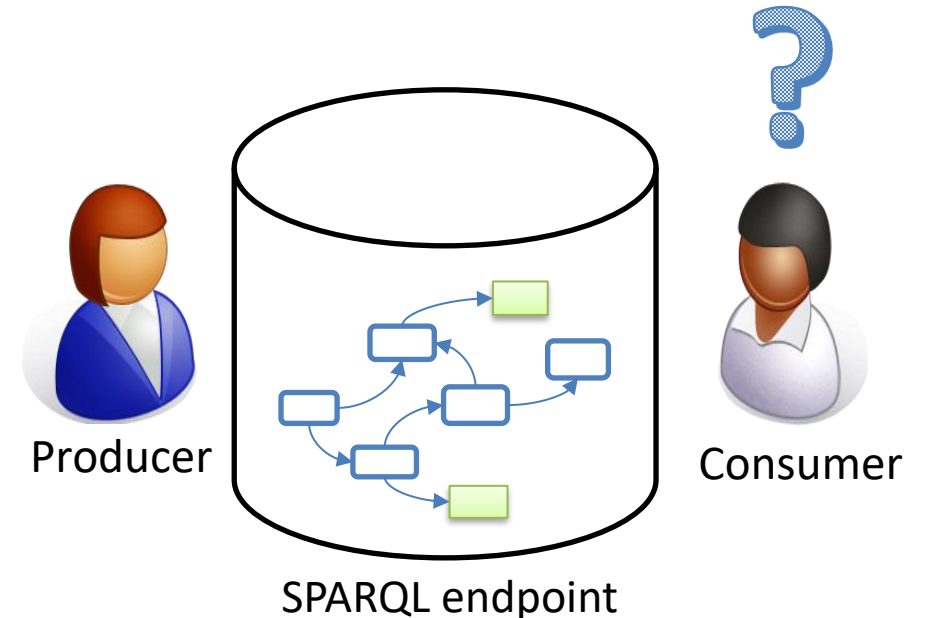
Consuming & producing RDF

Describing and validating RDF content

SPARQL endpoints are not well documented

Typical documentation = set of SPARQL queries

Difficult to know where to start doing queries





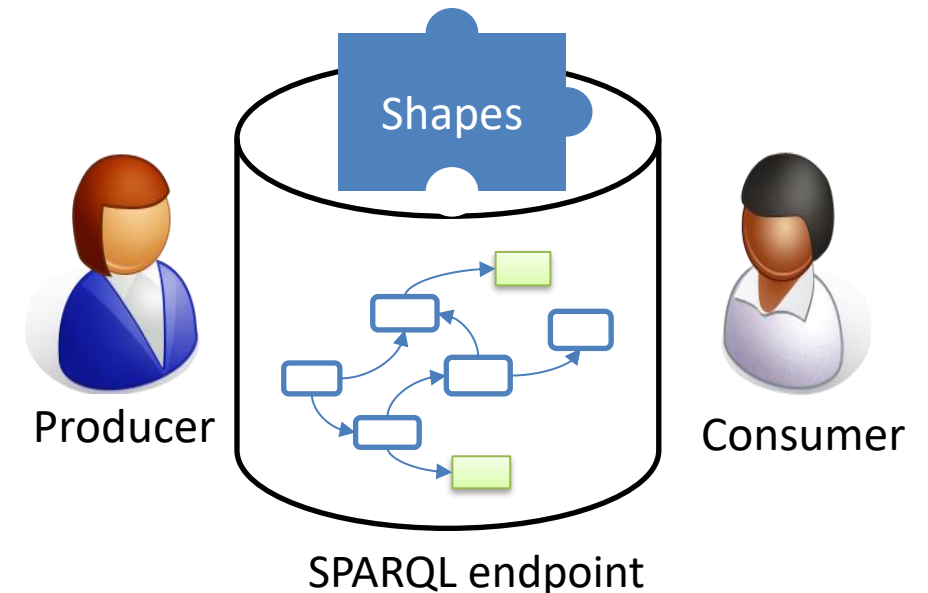
Why describe & validate RDF?

For producers

- Developers can understand the contents they are going to produce
- They can ensure they produce the expected structure
- Advertise and document the structure
- Generate interfaces

For consumers

- Understand the contents
- Verify the structure before processing it
- Query generation & optimization



In other technologies...

Technology	Schema
Relational Databases	DDL
XML	DTD, XML Schema, RelaxNG, Schematron
Json	Json Schema
RDF	?

Fill that gap



Schemas for RDF?

RDF flexibility doesn't want to impose a schema, but...

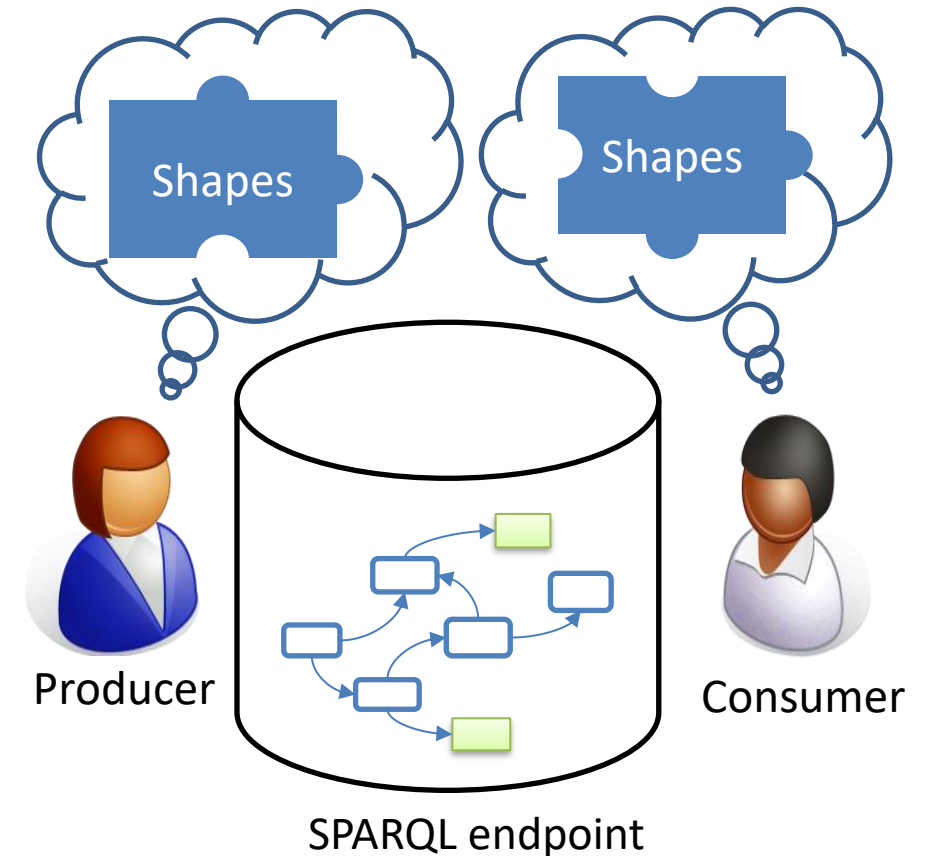
In practice, there are **implicit schemas**

Assumed by producers and consumers

Shapes can make schemas explicit

Handle malformed/incomplete data

Avoid defensive programming



Shapes for consensus building

Help domain experts define their own data models

Understandable by domain experts

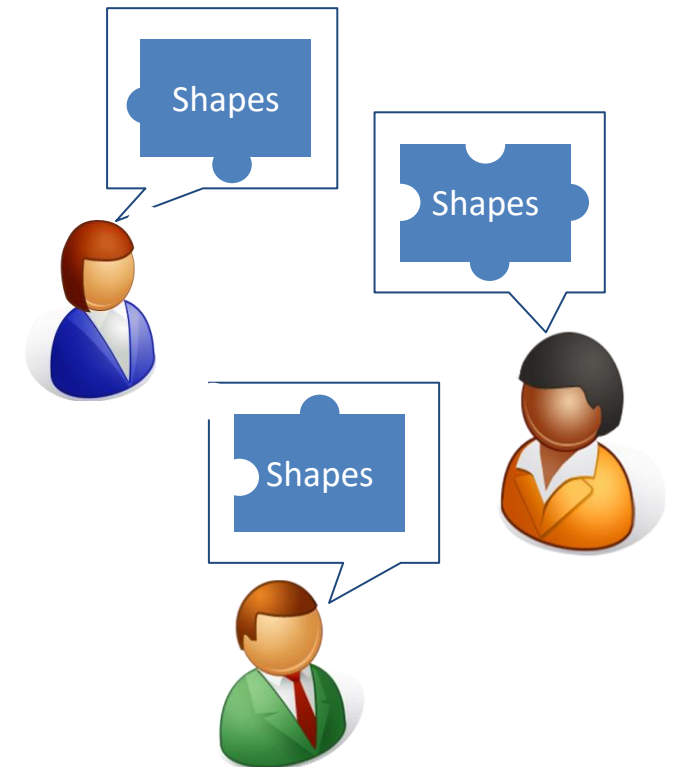
...and machine processable

Initial motivation: clinical data models ([FHIR](#))

Distributed data model

Different location, authorities,...

Extensible data models



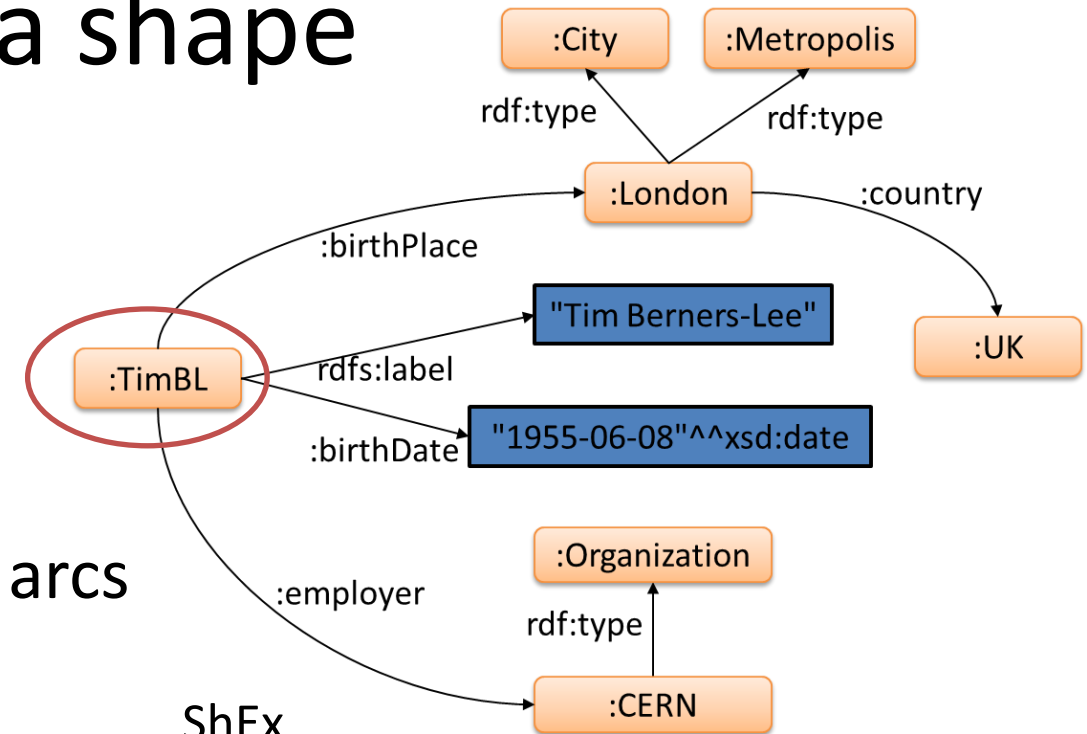
Example of a shape

A shape describes

The form of a node (node constraint)

Incoming/outgoing arcs from a node

Possible values associated with those arcs



RDF Node

```
:timbl rdfs:label "Tim Berners-Lee" ;
      :birthPlace :london ;
      :birthDate "1955-06-08"^^xsd:date ;
      :employer :CERN .
```

ShEx

```
<Researcher> {
  rdfs:label xsd:string ;
  :birthPlace @<Place> ? ;
  :birthDate xsd:date ? ;
  :employer @<Organization> * ;
}
```



ShEx & SHACL

2013 RDF Validation Workshop

Conclusions of the workshop:

There is a need of a higher level, concise language for RDF Validation

ShEx initially proposed (v 1.0)

2014 W3c Data Shapes WG chartered

2017 SHACL accepted as W3C recommendation

2017 ShEx 2.0 released as W3C Community group draft

2019 ShEx adopted by Wikidata

2024 IEEE ShEx (*work in progress*)