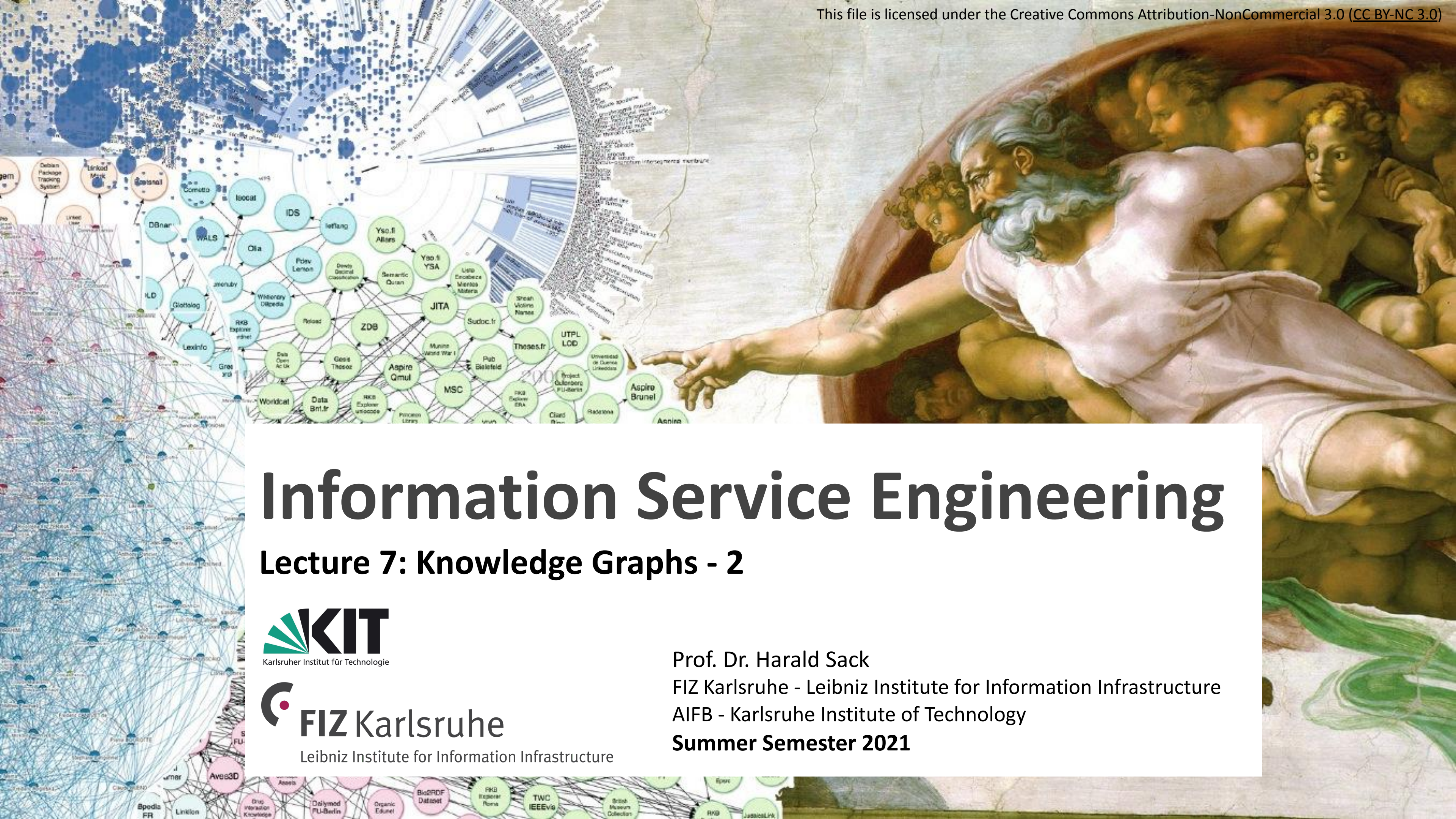


# Information Service Engineering

## Lecture 7: Knowledge Graphs - 2

KIT
Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

Prof. Dr. Harald Sack
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure
AIFB - Karlsruhe Institute of Technology

**Summer Semester 2021**

# Information Service Engineering

## Last Lecture: Knowledge Graphs - 1

**3.1 Knowledge Representations and Ontologies**

**3.2 Semantic Web and the Web of Data**

**3.3 Linked Data Principles**

**3.4 How to identify and Access Things**

3.5 Resource Description Framework (RDF) as simple Data Model

3.6 Creating new Models with RDFS

3.7 Knowledge Graphs

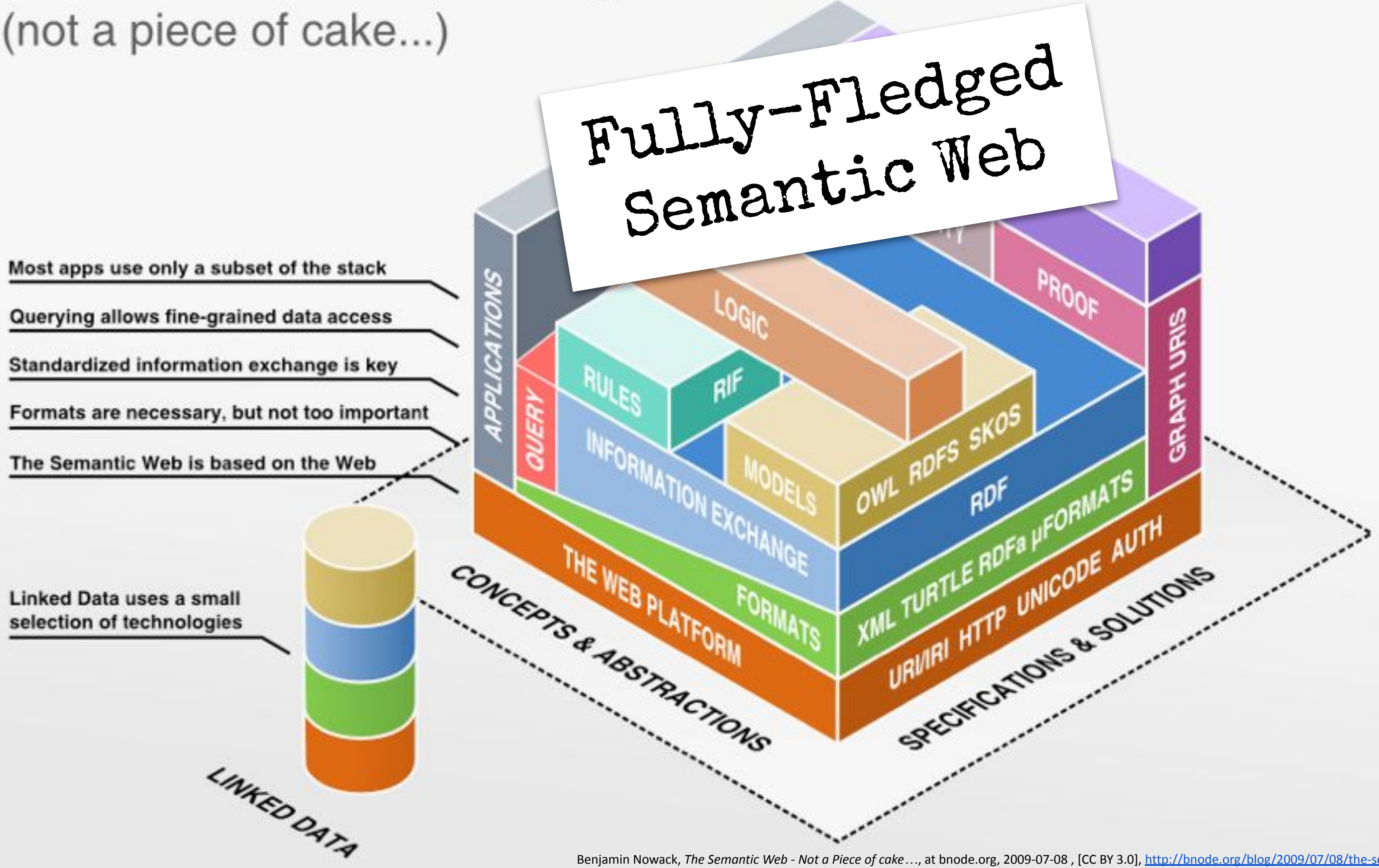
3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

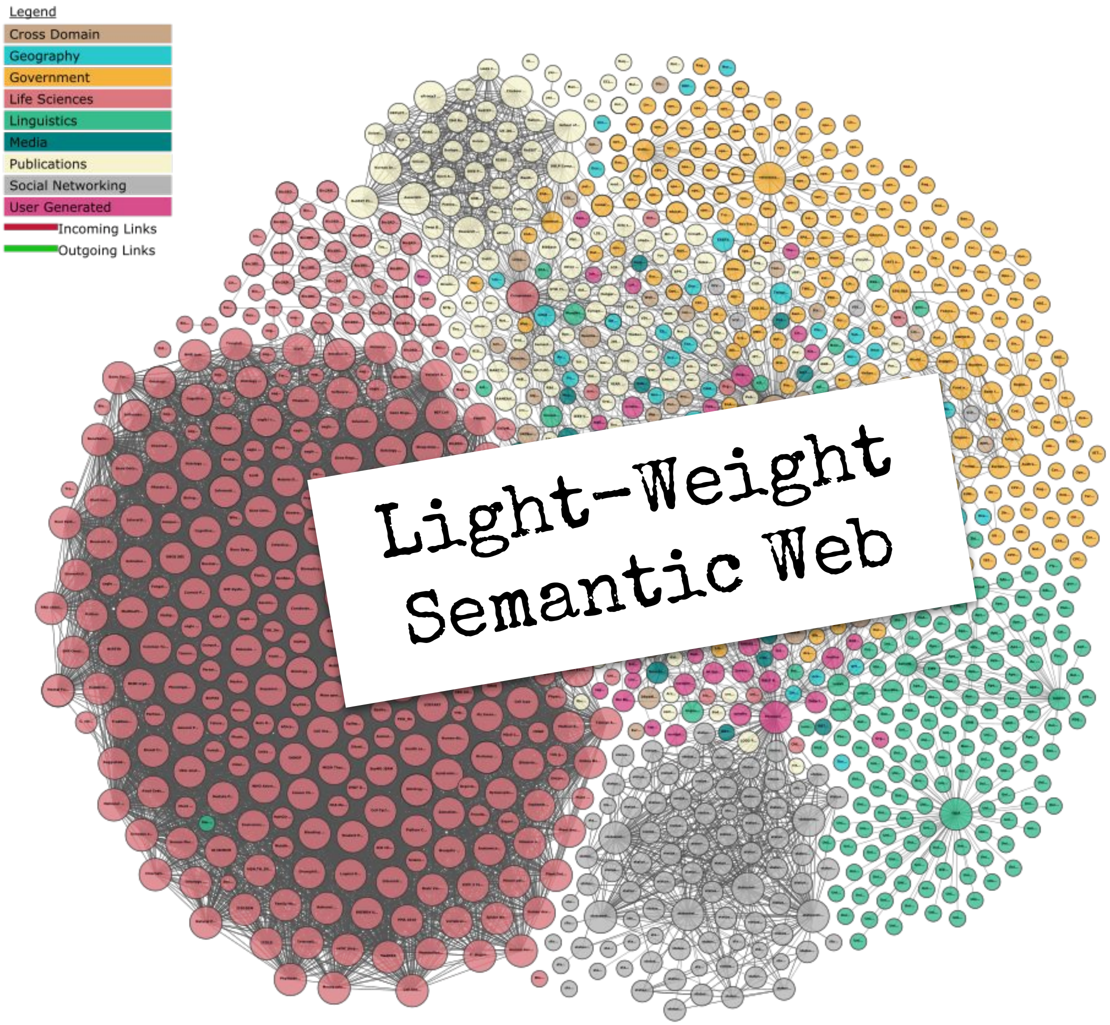
- Ontologies & Semantic Web
- Semantic Web Technology Stack
- Linked Data Principles
- URIs and URLs
- Designator and Designatum
- HTTP Content Negotiation

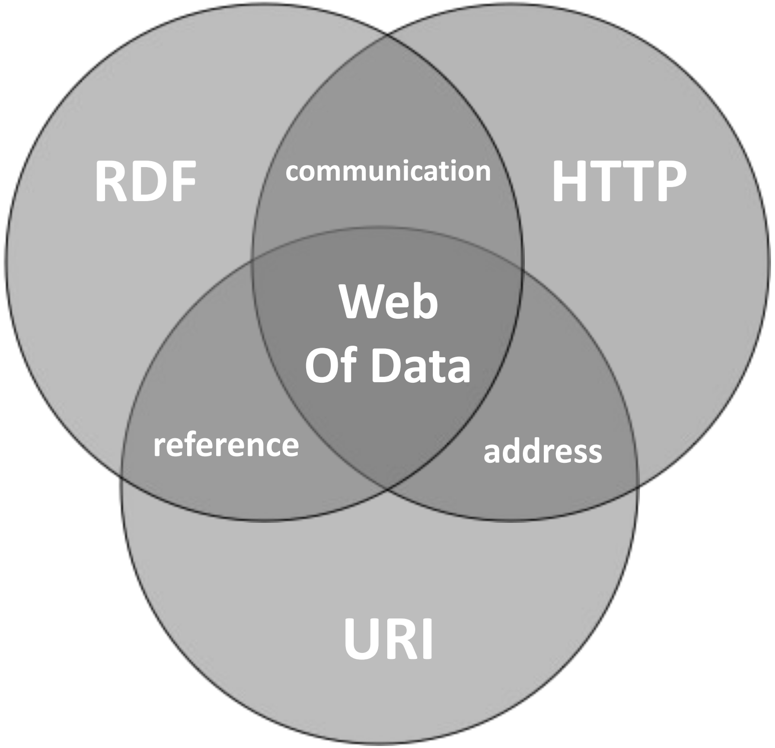
# The Semantic Web Technology Stack
(not a piece of cake...)


Fully-Fledged
Semantic Web
Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small
selection of technologies
LINKED DATA
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS
APPLICATIONS
QUERY
RULES
RIF
LOGIC
PROOF
GRAPH URIS
INFORMATION EXCHANGE
MODELS
OWL RDFS SKOS
RDF
FORMATS
XML TURTLE RDFa µFORMATS
THE WEB PLATFORM
URI/IRI HTTP UNICODE AUTH

Benjamin Nowack, *The Semantic Web - Not a Piece of cake*…, at bnode.org, 2009-07-08 , [CC BY 3.0], http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

# Light-Weight Semantic Web





# Information Service Engineering

## Lecture 7: Knowledge Graphs - 2

3.1 Knowledge Representations and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things

**3.5 Resource Description Framework (RDF) as simple Data Model**

3.6 Creating new Models with RDFS

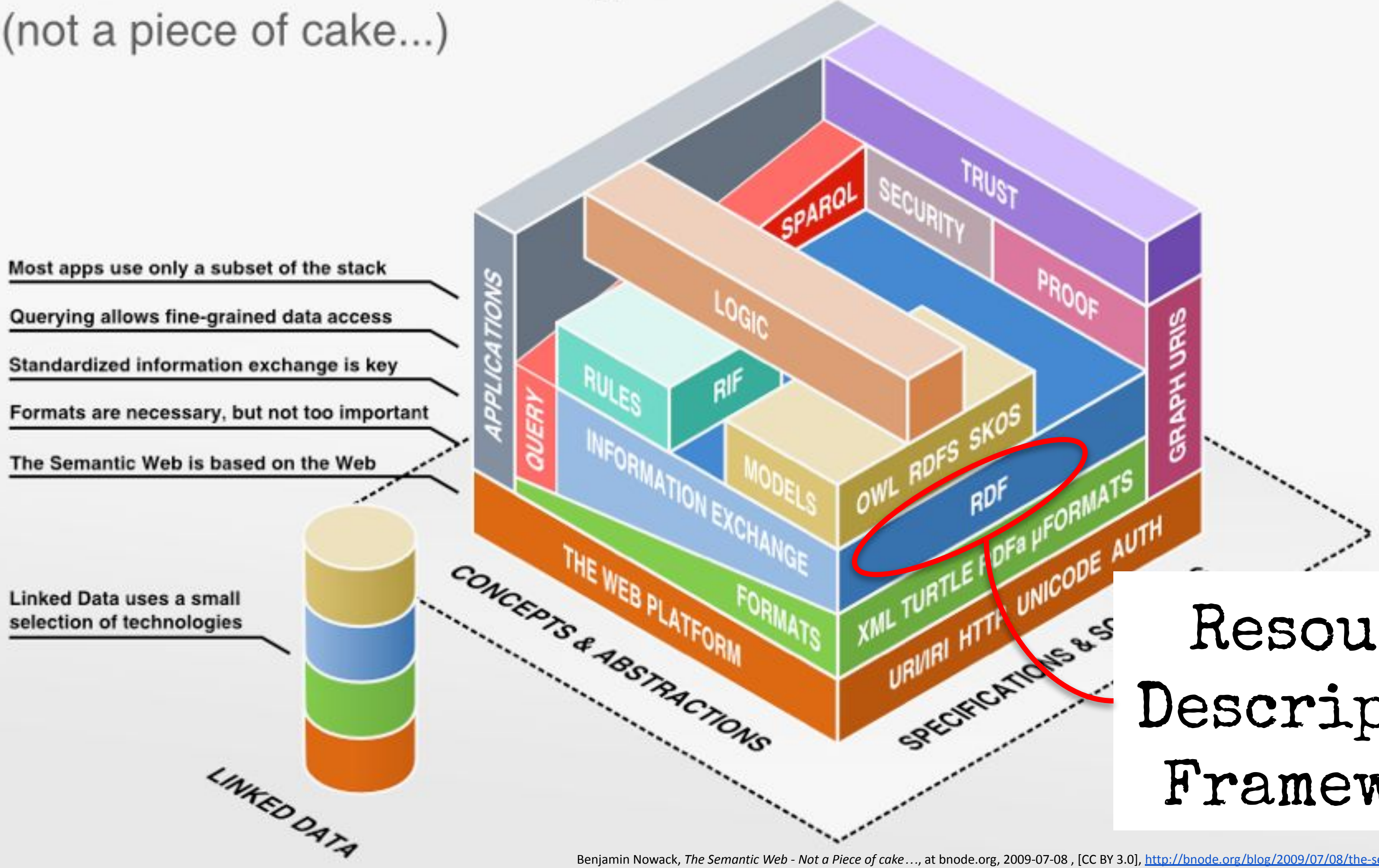
3.7 Knowledge Graphs

3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

# The Semantic Web Technology Stack
(not a piece of cake...)



Most apps use only a subset of the stack

Querying allows fine-grained data access

Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

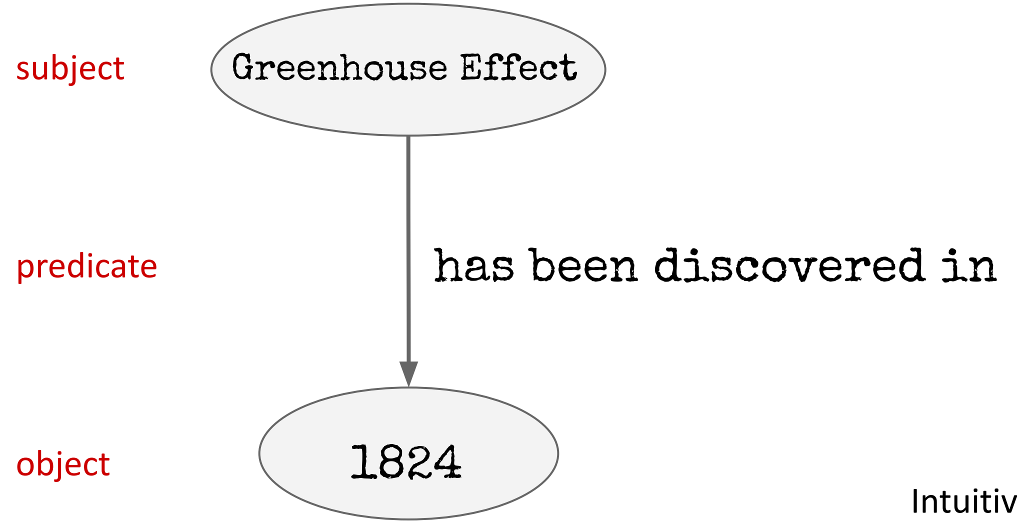
Linked Data uses a small selection of technologies

Resource Description Framework

Benjamin Nowack, *The Semantic Web - Not a Piece of cake*…, at bnode.org, 2009-07-08 , [CC BY 3.0], http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

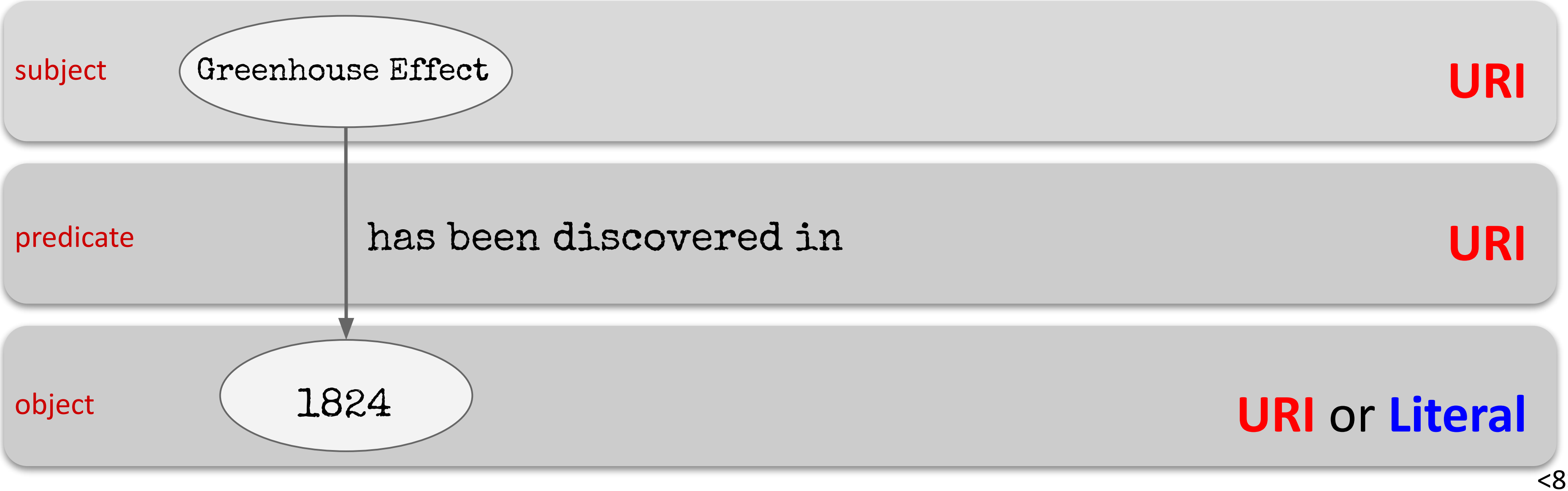
# How to represent Knowledge?

- How do I represent the following fact:
  “*The Greenhouse Effect has been discovered in 1824*” in an intuitive way?



Intuitive knowledge representation via a **directed graph**.

# Resource Description Framework



# Resource Description Framework

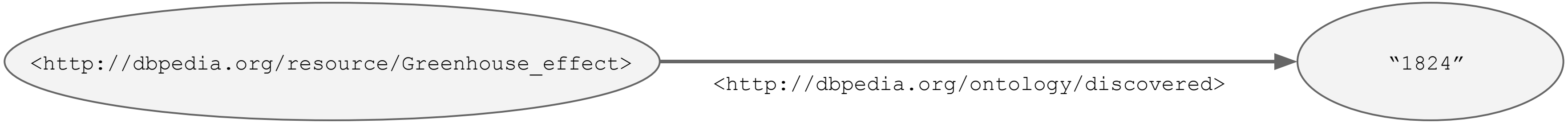
- **RDF Statements (RDF-Triple):**

In RDF the predicate of a statement is referred to as "Property"

| Subject | Property | Object / Value |
|---|---|---|
| **URI** | **URI** | **URI / Literal** |

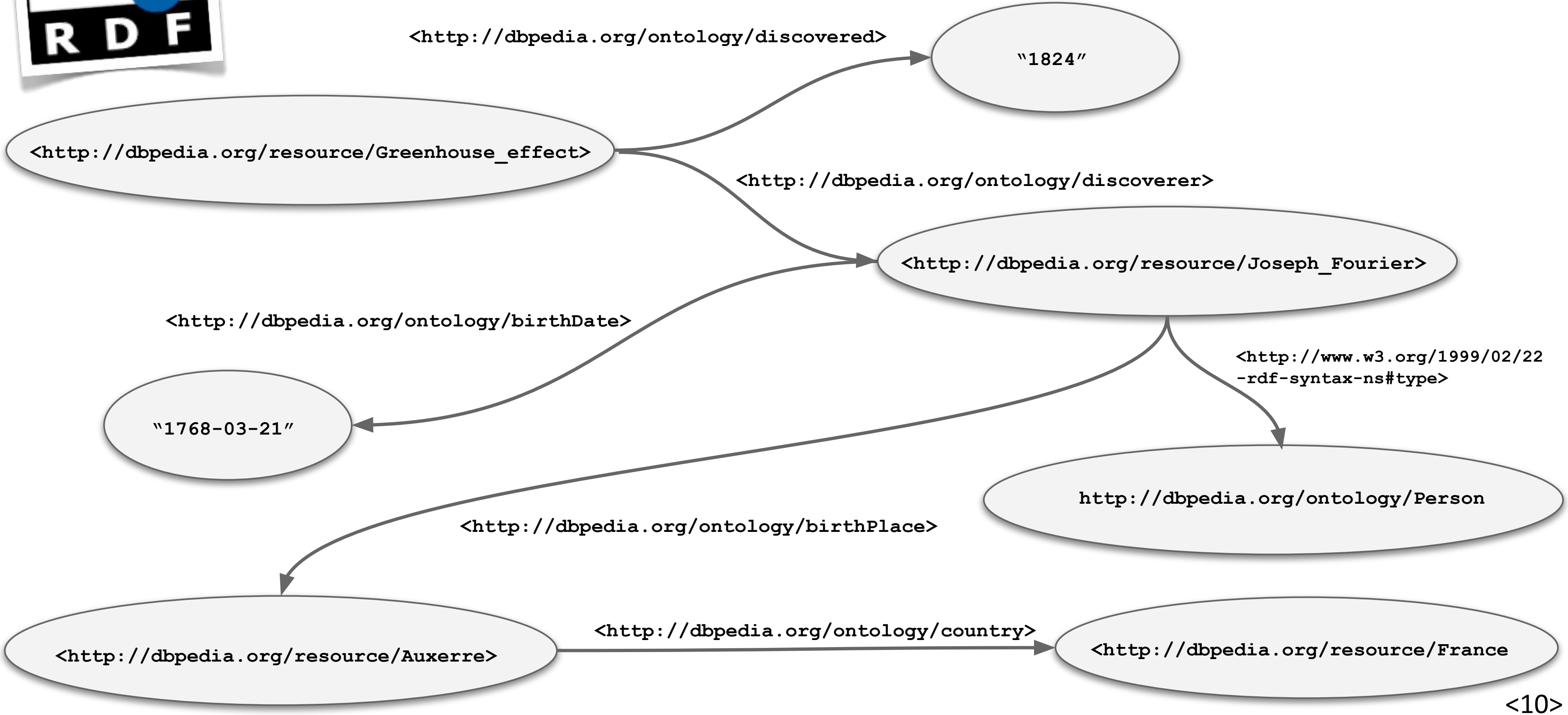
N-Triples Serialization

```
<http://dbpedia.org/resource/Greenhouse_effect>   <http://dbpedia.org/ontology/discovered>   "1824" .
```



Graph Representation

# Resource Description Framework



<http://dbpedia.org/resource/Greenhouse_effect>

<http://dbpedia.org/ontology/discovered>

"1824"

<http://dbpedia.org/ontology/discoverer>

<http://dbpedia.org/resource/Joseph_Fourier>

<http://dbpedia.org/ontology/birthDate>

"1768-03-21"

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

http://dbpedia.org/ontology/Person

<http://dbpedia.org/ontology/birthPlace>

<http://dbpedia.org/resource/Auxerre>

<http://dbpedia.org/ontology/country>

<http://dbpedia.org/resource/France

# Resource Description Framework

```
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discovered> “1824” .
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer> <http://dbpedia.org/resource/Joseph_Fourier> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Climate_change> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Atmosphere> .
                    ...                                       ...                                 ...

<http://dbpedia.org/resource/Joseph_Fourier> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Person> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthDate> “1768-03-21” .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Auxerre> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/field> <http://dbpedia.org/resource/Physicist> .
                    ...                                       ...                                   ...

<http://dbpedia.org/resource/Auxerre> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/France> .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> “47.798599”^^xsd:float .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#long> “3.567200”^^xsd:float .
                    ...                                       ...                                      ...
```

**Subject** **Property** **Object**

**RDF Triples**

# Resource Description Framework

**<http://dbpedia.org/resource/Greenhouse_effect>** <http://dbpedia.org/ontology/discovered> “1824” .
**<http://dbpedia.org/resource/Greenhouse_effect>** <http://dbpedia.org/ontology/discoverer> **<http://dbpedia.org/resource/Joseph_Fourier>** .
**<http://dbpedia.org/resource/Greenhouse_effect>** <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Climate_change> .
**<http://dbpedia.org/resource/Greenhouse_effect>** <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Atmosphere> .
... ... ...

**<http://dbpedia.org/resource/Joseph_Fourier>** <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Person> .
**<http://dbpedia.org/resource/Joseph_Fourier>** <http://dbpedia.org/ontology/birthDate> “1768-03-21” .
**<http://dbpedia.org/resource/Joseph_Fourier>** <http://dbpedia.org/ontology/birthPlace> **<http://dbpedia.org/resource/Auxerre>** .
**<http://dbpedia.org/resource/Joseph_Fourier>** <http://dbpedia.org/ontology/field> **<http://dbpedia.org/resource/Physicist>** .
... ... ...

**<http://dbpedia.org/resource/Auxerre>** <http://dbpedia.org/ontology/country> **<http://dbpedia.org/resource/France>** .
**<http://dbpedia.org/resource/Auxerre>** <http://www.w3.org/2003/01/geo/wgs84_pos#lat> “47.798599”^^xsd:float .
**<http://dbpedia.org/resource/Auxerre>** <http://www.w3.org/2003/01/geo/wgs84_pos#long> “3.567200”^^xsd:float .
... ... ...

**Individuals (Entities)**

# Resource Description Framework

<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discovered> “1824” .
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer> <http://dbpedia.org/resource/Joseph_Fourier> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> **<http://dbpedia.org/category/Climate_change>** .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> **<http://dbpedia.org/category/Atmosphere>** .
… … …

<http://dbpedia.org/resource/Joseph_Fourier> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> **<http://dbpedia.org/ontology/Person>** .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthDate> “1768-03-21” .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Auxerre> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/field> <http://dbpedia.org/resource/Physicist> .
… … …

<http://dbpedia.org/resource/Auxerre> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/France> .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> “47.798599”^^xsd:float .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#long> “3.567200”^^xsd:float .
… … …

**Classes / Concepts**

# Resource Description Framework

```
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discovered> "1824" .
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer> <http://dbpedia.org/resource/Joseph_Fourier> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Climate_change> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Atmosphere> .
                  ...                                       ...                                   ...

<http://dbpedia.org/resource/Joseph_Fourier> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Person> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthDate> "1768-03-21" .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Auxerre> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/field> <http://dbpedia.org/resource/Physicist> .
                  ...                                       ...                                     ...

<http://dbpedia.org/resource/Auxerre> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/France> .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "47.798599"^^xsd:float .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "3.567200"^^xsd:float .
                  ...                                       ...                                       ...
```

**Literals**

# Resource Description Framework

<http://dbpedia.org/resource/Greenhouse_effect> **<http://dbpedia.org/ontology/discovered>** “1824” .
<http://dbpedia.org/resource/Greenhouse_effect> **<http://dbpedia.org/ontology/discoverer>** <http://dbpedia.org/resource/Joseph_Fourier> .
<http://dbpedia.org/resource/Greenhouse_effect> **<http://purl.org/dc/terms/subject>** <http://dbpedia.org/category/Climate_change> .
<http://dbpedia.org/resource/Greenhouse_effect> **<http://purl.org/dc/terms/subject>** <http://dbpedia.org/category/Atmosphere> .
... ... ...

<http://dbpedia.org/resource/Joseph_Fourier> **<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>** <http://dbpedia.org/ontology/Person> .
<http://dbpedia.org/resource/Joseph_Fourier> **<http://dbpedia.org/ontology/birthDate>** “1768-03-21” .
<http://dbpedia.org/resource/Joseph_Fourier> **<http://dbpedia.org/ontology/birthPlace>** <http://dbpedia.org/resource/Auxerre> .
<http://dbpedia.org/resource/Joseph_Fourier> **<http://dbpedia.org/ontology/field>** <http://dbpedia.org/resource/Physicist> .
... ... ...

<http://dbpedia.org/resource/Auxerre> **<http://dbpedia.org/ontology/country>** <http://dbpedia.org/resource/France> .
<http://dbpedia.org/resource/Auxerre> **<http://www.w3.org/2003/01/geo/wgs84_pos#lat>** “47.798599”^^xsd:float .
<http://dbpedia.org/resource/Auxerre> **<http://www.w3.org/2003/01/geo/wgs84_pos#long>** “3.567200”^^xsd:float .
... ... ...

**Properties**

# Resource Description Framework

<**http://dbpedia.org/resource/**Greenhouse_effect> <**http://dbpedia.org/ontology**/discovered> “1824” .
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer> <http://dbpedia.org/resource/Joseph_Fourier> .
<http://dbpedia.org/resource/Greenhouse_effect> <**http://purl.org/dc/terms/**subject> <**http://dbpedia.org/category**/Climate_change> .
<http://dbpedia.org/resource/Greenhouse_effect> <http://purl.org/dc/terms/subject> <http://dbpedia.org/category/Atmosphere> .
… … …

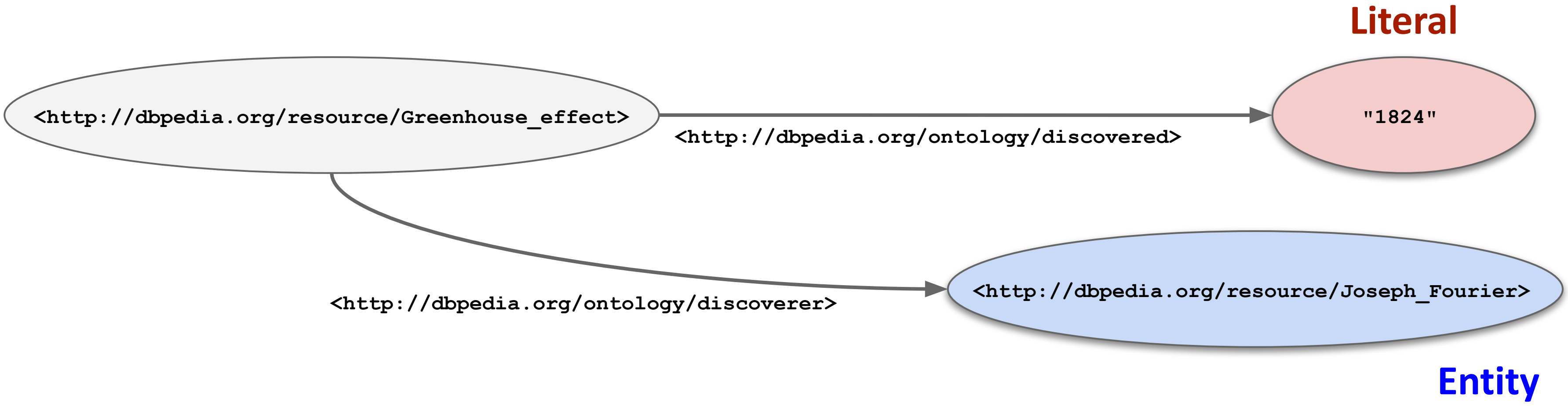
<http://dbpedia.org/resource/Joseph_Fourier> <**http://www.w3.org/1999/02/22-rdf-syntax-ns#**type> <http://dbpedia.org/ontology/Person> .
<http://dbpedia.org/resource/Joseph_Fourier> <**http://dbpedia.org/ontology**/birthDate> “1768-03-21” .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Auxerre> .
<http://dbpedia.org/resource/Joseph_Fourier> <http://dbpedia.org/ontology/field> <http://dbpedia.org/resource/Physicist> .
… … …

<http://dbpedia.org/resource/Auxerre> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/France> .
<http://dbpedia.org/resource/Auxerre> <**http://www.w3.org/2003/01/geo/wgs84_pos#**lat> “47.798599”^^xsd:float .
<http://dbpedia.org/resource/Auxerre> <http://www.w3.org/2003/01/geo/wgs84_pos#long> “3.567200”^^xsd:float .
… … …

**Vocabularies / Ontologies**

# Resource Description Framework

- **URIs and Literals**
  - **URIs** identify and reference resources uniquely.
  - **Literals** describe data values that don’t have a separate existence.



Literal

<http://dbpedia.org/resource/Greenhouse_effect>

<http://dbpedia.org/ontology/discovered>

"1824"

<http://dbpedia.org/ontology/discoverer>

<http://dbpedia.org/resource/Joseph_Fourier>

Entity

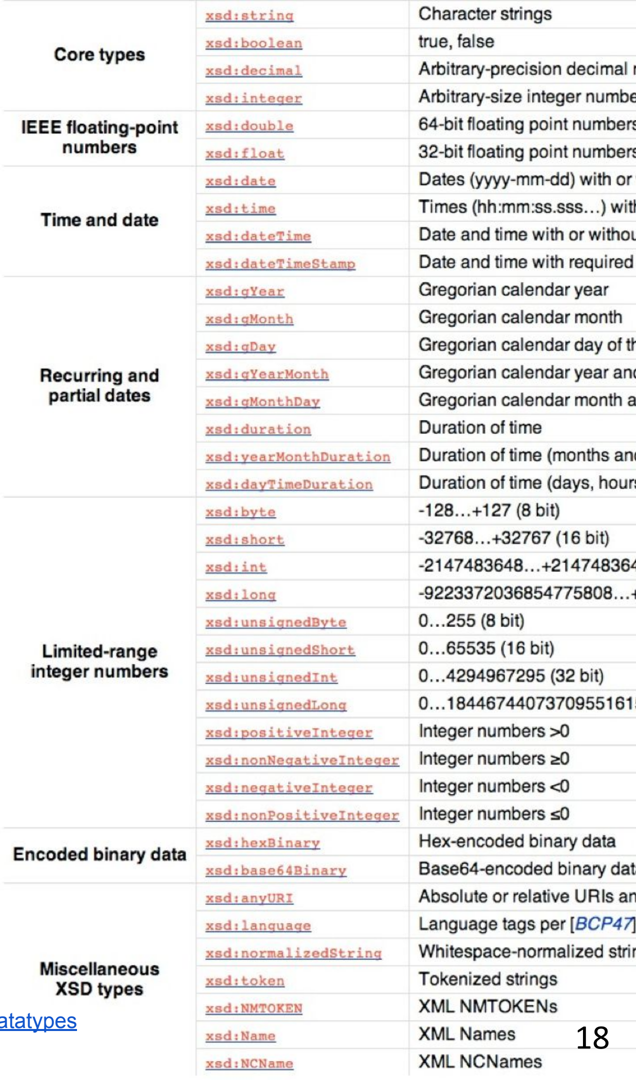
# RDF Literals and Data Types

- Typed literals can be expressed via **XML Schema datatypes**.
- Namespace for typed literals:
  `http://www.w3.org/2001/XMLSchema#`

- Examples:

```
"Semantics"^^<http://www.w3.org/2001/XMLSchema#string>
"1161.00"^^<http://www.w3.org/2001/XMLSchema#float>
"2015-08-02"^^<http://www.w3.org/2001/XMLSchema#date>
```

- **Language Tags** denote the (natural) language of the text.
  - Example:
    `"Semantik"@de , "Semantics"@en`

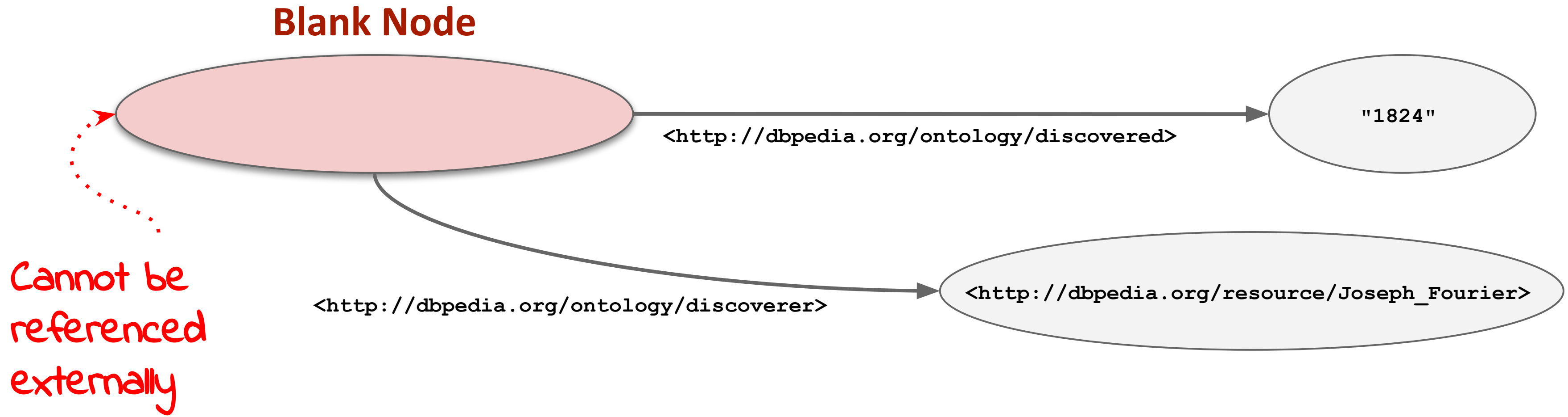


| | | |
|---|---|---|
| Core types | xsd:string | Character strings |
| | xsd:boolean | true, false |
| | xsd:decimal | Arbitrary-precision decimal r |
| | xsd:integer | Arbitrary-size integer numbe |
| IEEE floating-point numbers | xsd:double | 64-bit floating point numbers |
| | xsd:float | 32-bit floating point numbers |
| Time and date | xsd:date | Dates (yyyy-mm-dd) with or |
| | xsd:time | Times (hh:mm:ss.sss…) with |
| | xsd:dateTime | Date and time with or withou |
| | xsd:dateTimeStamp | Date and time with required |
| Recurring and partial dates | xsd:gYear | Gregorian calendar year |
| | xsd:gMonth | Gregorian calendar month |
| | xsd:gDay | Gregorian calendar day of th |
| | xsd:gYearMonth | Gregorian calendar year and |
| | xsd:gMonthDay | Gregorian calendar month a |
| | xsd:duration | Duration of time |
| | xsd:yearMonthDuration | Duration of time (months and |
| | xsd:dayTimeDuration | Duration of time (days, hours |
| Limited-range integer numbers | xsd:byte | -128…+127 (8 bit) |
| | xsd:short | -32768…+32767 (16 bit) |
| | xsd:int | -2147483648…+214748364 |
| | xsd:long | -9223372036854775808…+ |
| | xsd:unsignedByte | 0…255 (8 bit) |
| | xsd:unsignedShort | 0…65535 (16 bit) |
| | xsd:unsignedInt | 0…4294967295 (32 bit) |
| | xsd:unsignedLong | 0…1844674407370955161 |
| | 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 dat |
| Miscellaneous XSD types | xsd:anyURI | Absolute or relative URIs an |
| | xsd:language | Language tags per [*BCP47*] |
| | xsd:normalizedString | Whitespace-normalized strin |
| | xsd:token | Tokenized strings |
| | xsd:NMTOKEN | XML NMTOKENs |
| | xsd:Name | XML Names |
| | xsd:NCName | XML NCNames |

http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/#xsd-datatypes

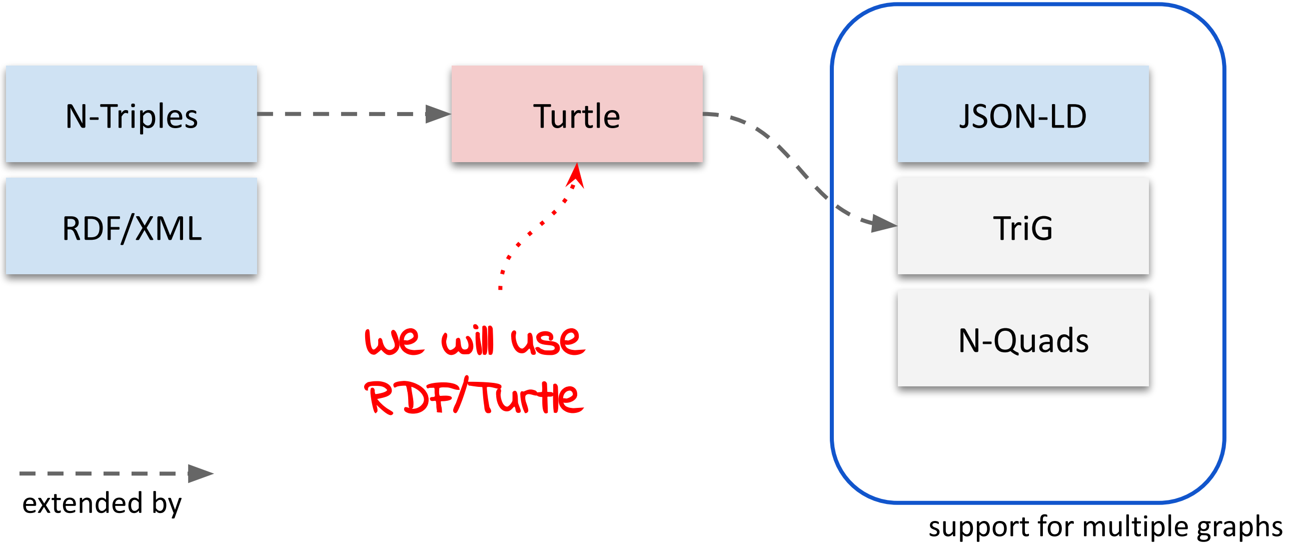
# RDF Blank Nodes

- **Blank Nodes**
  - Denote the **existence of an individual** with specific attributes, but **without providing an identification or reference**.


Blank Node
<http://dbpedia.org/ontology/discovered>
"1824"
<http://dbpedia.org/ontology/discoverer>
<http://dbpedia.org/resource/Joseph_Fourier>
Cannot be referenced externally

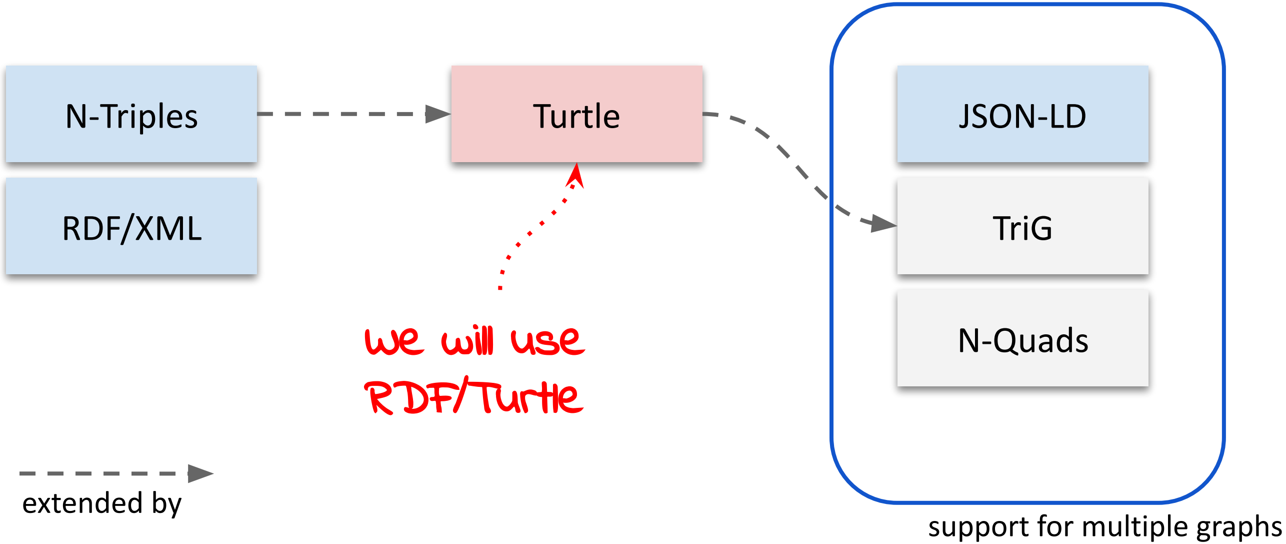
# RDF Serializations

- RDF comes with several different **serialization formats**:
  - N-Triples, RDF/XML, JSON, Turtle, TriG, N-Quads, RDFa, ...

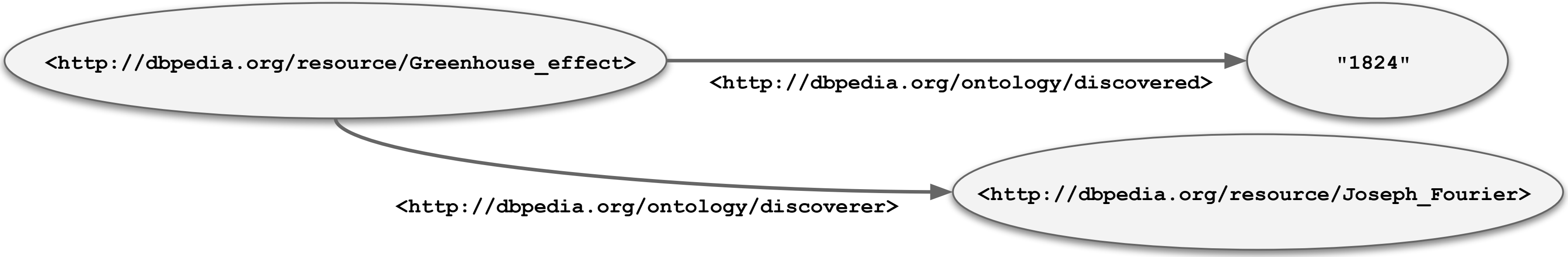


# RDF Serializations

- RDF comes with several different **serialization formats**:
  - N-Triples, RDF/XML, JSON, Turtle, TriG, N-Quads, RDFa, ...



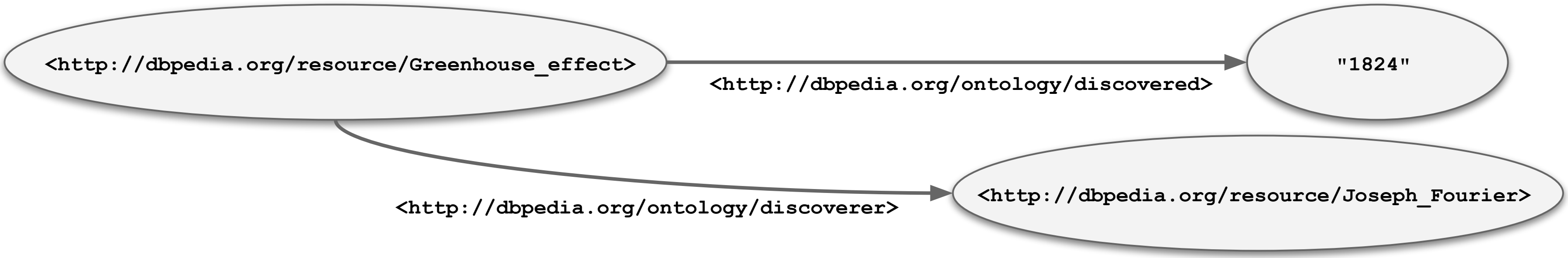
# RDF Serializations



- **N-Triples Notation**
  - **URIs/IRIs** in angle brackets
  - **Literals** in quotation marks
  - Triple ends with a **period**

```
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discovered> "1824" .

<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer>
<http://dbpedia.org/resource/Joseph_Fourier> .
```

# RDF Serializations

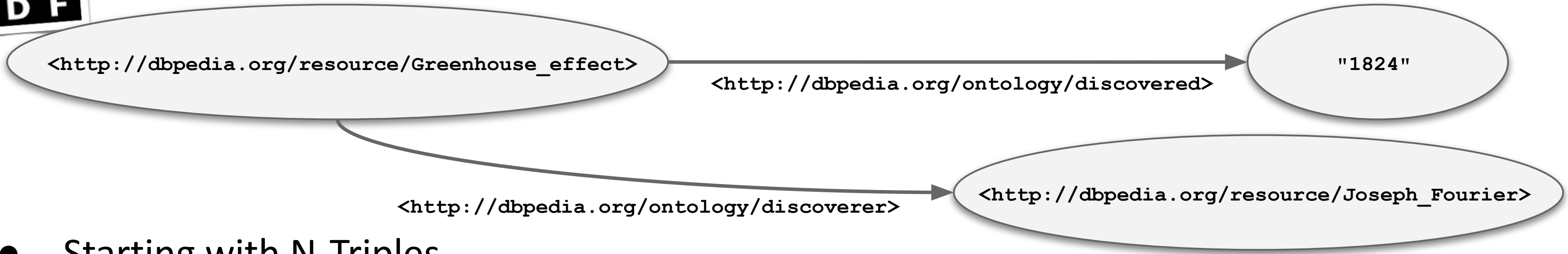


- **Turtle (Terse RDF Triple Language) Notation**
  - Extension of N-Triples

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base  <http://dbpedia.org/resource/> .

<Greenhouse_effect> dbo:discovered "1824" .

<Greenhouse_effect> dbo:discoverer <Joseph_Fourier> .
```

RDF/Turtle allows shortcuts and abbreviations for readability

# RDF/Turtle



- Starting with N-Triples

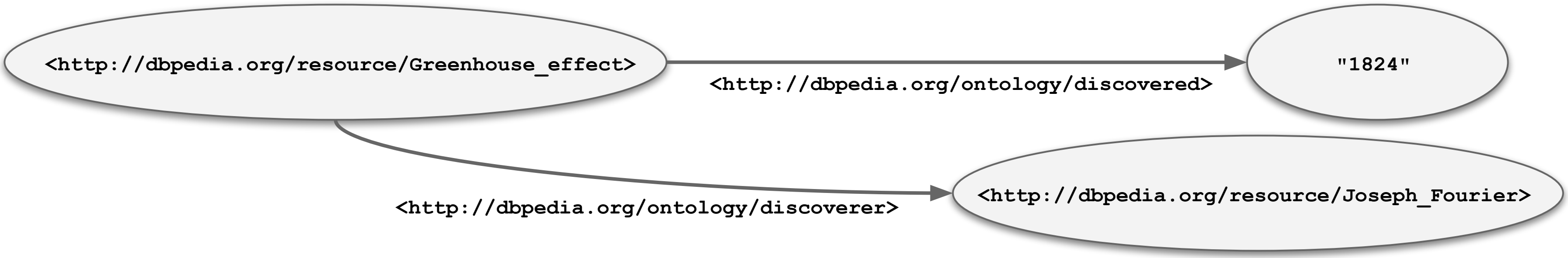
```
<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discovered> "1824" .

<http://dbpedia.org/resource/Greenhouse_effect> <http://dbpedia.org/ontology/discoverer>
<http://dbpedia.org/resource/Joseph_Fourier> .
```

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base   <http://dbpedia.org/resource/> .

<Greenhouse_effect> dbo:discovered "1824" .

<Greenhouse_effect> dbo:discoverer <Joseph_Fourier> .
```

**@prefix directive** associates prefix-label with URI

**@base directive** provides URI to complement all relative URIs

# RDF/Turtle

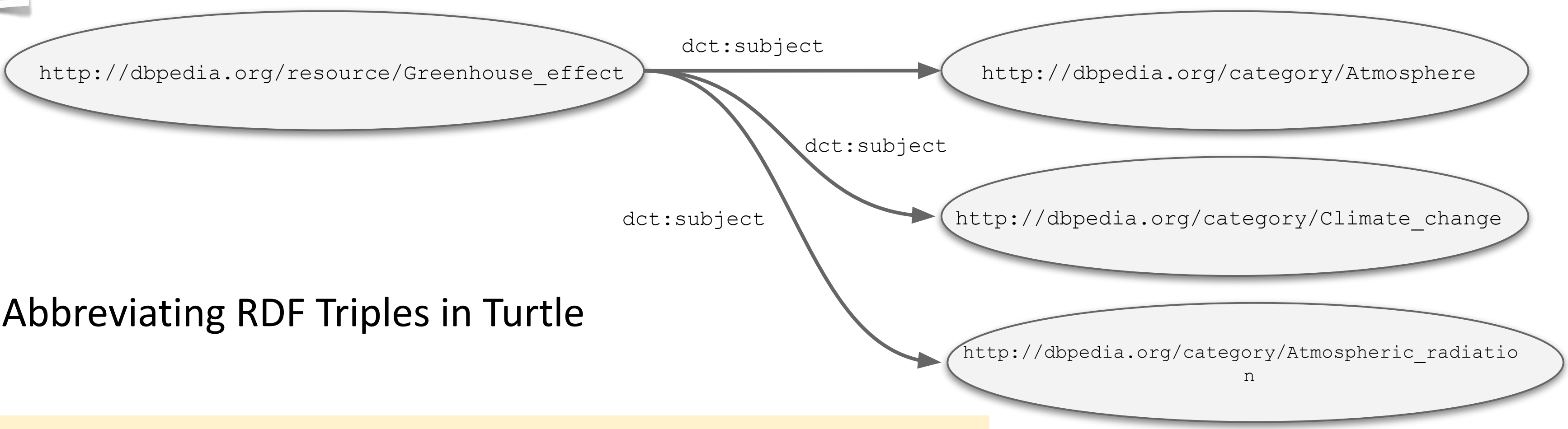


- Abbreviating RDF Triples in Turtle

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base  <http://dbpedia.org/resource/> .

<Greenhouse_effect> dbo:discovered "1824" ;
                    dbo:discoverer <Joseph_Fourier> .
```

**Semicolon** indicates that subsequent triples have the same subject **(predicate list)**

# RDF/Turtle

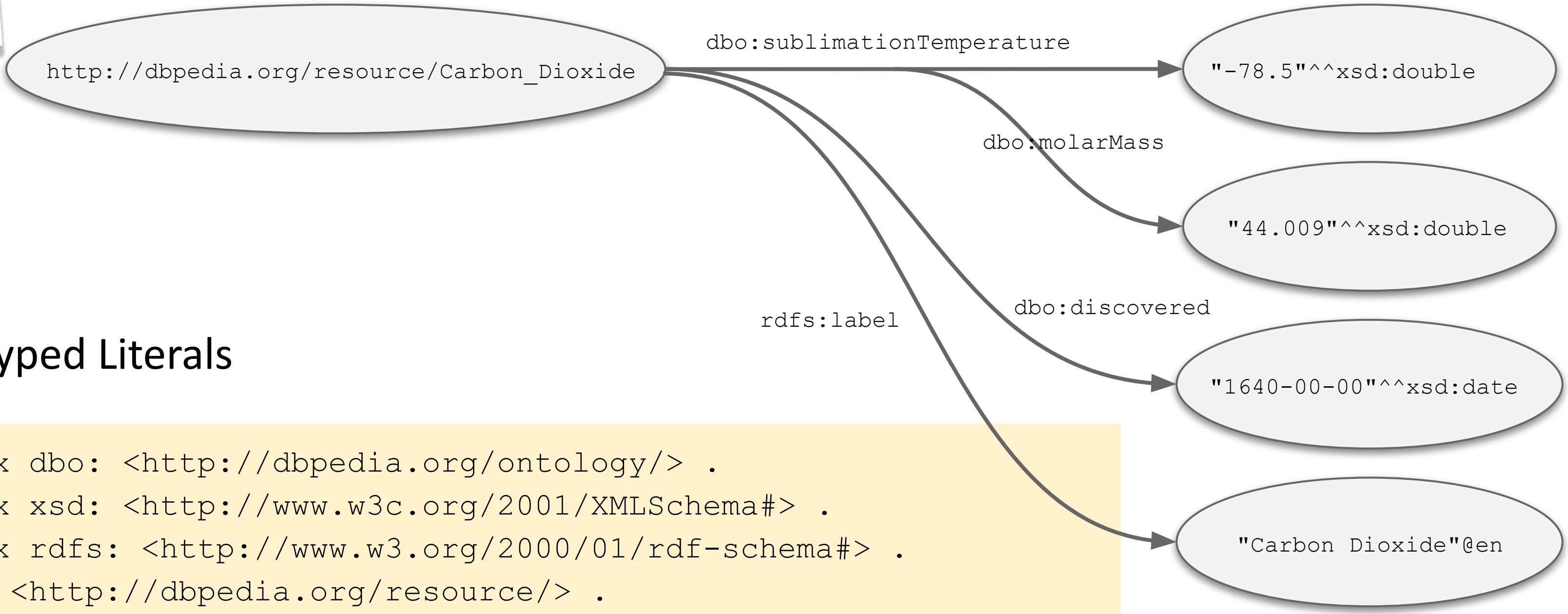

http://dbpedia.org/resource/Greenhouse_effect
dct:subject
http://dbpedia.org/category/Atmosphere
dct:subject
http://dbpedia.org/category/Climate_change
dct:subject
http://dbpedia.org/category/Atmospheric_radiatio
n

- Abbreviating RDF Triples in Turtle

```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix dbc: <http://dbpedia.org/category/> .
@base        <http://dbpedia.org/resource/> .

<Greenhouse_effect> dct:subject dbc:Atmosphere ,
                                dbc:Climate_change ,
                                dbc:Atmospheric_radiation .
```

**Comma** indicates that subsequent triples have same subject and property **(object list)**

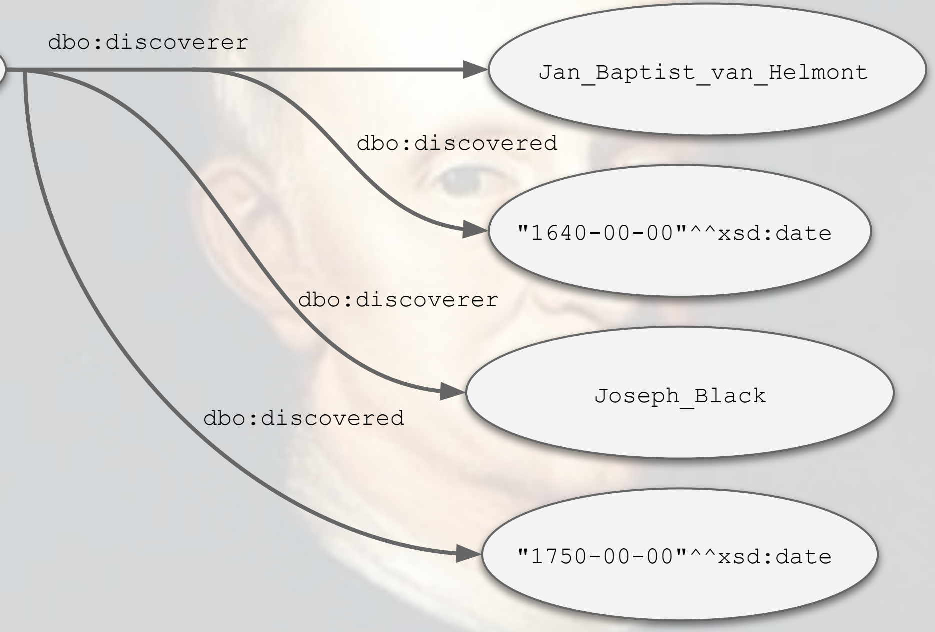
# RDF/Turtle



- Typed Literals

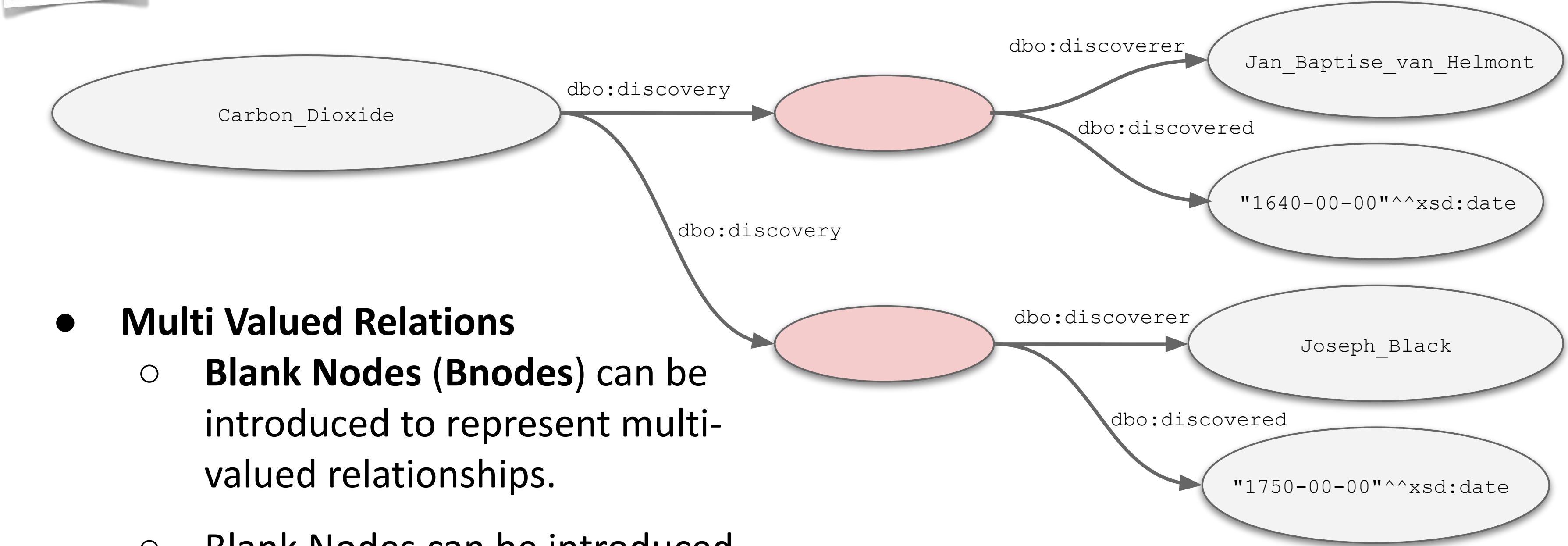
```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base  <http://dbpedia.org/resource/> .

<Carbon_Dioxide> dbo:sublimationTemperature "-78.5"^^xsd:double ;
                 dbo:molarMass "44.009"^^xsd:double ;
                 dbo:discovered "1640-00-00"^^xsd:date ;
                 rdfs:label "Carbon Dioxide"@en .
```

# RDF/Turtle



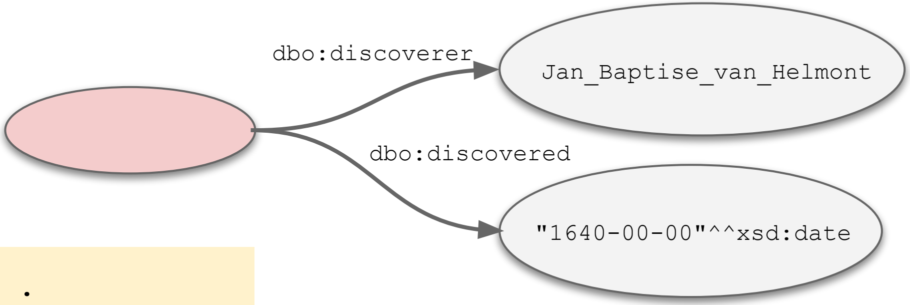
- **Multi Valued Relations**
  - *$CO_2$ was discovered / rediscovered by two scientists at different times.*
  - How to model this in RDF?
  - **Problem:** unique association

# RDF/Turtle



- **Multi Valued Relations**
  - ○ **Blank Nodes** (**Bnodes**) can be introduced to represent multi-valued relationships.
  - ○ Blank Nodes can be introduced for resources that don‘t need a name (auxiliary nodes).

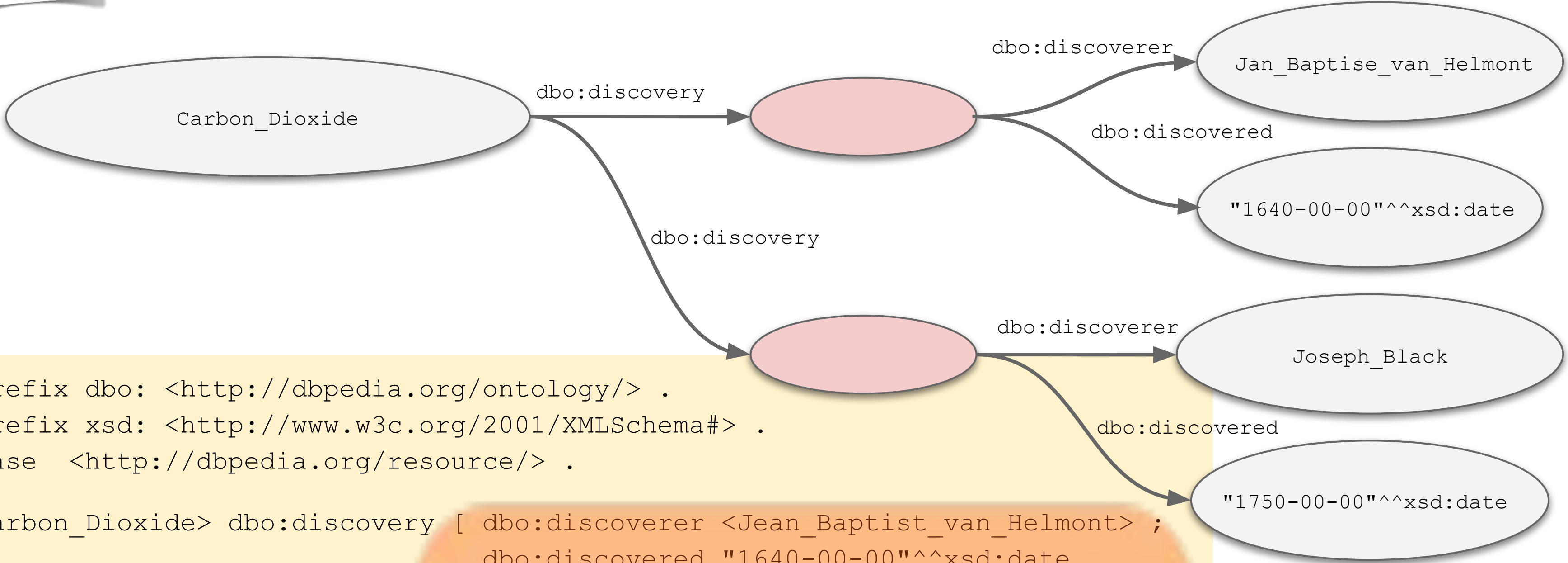
# RDF/Turtle



```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@base  <http://dbpedia.org/resource/> .

[] dbo:discoverer <Jan_Baptist_van_Helmont> ;
   dbo:discovered "1640-00-00"^^xsd:date .
```

**anonymous blank node** as subject

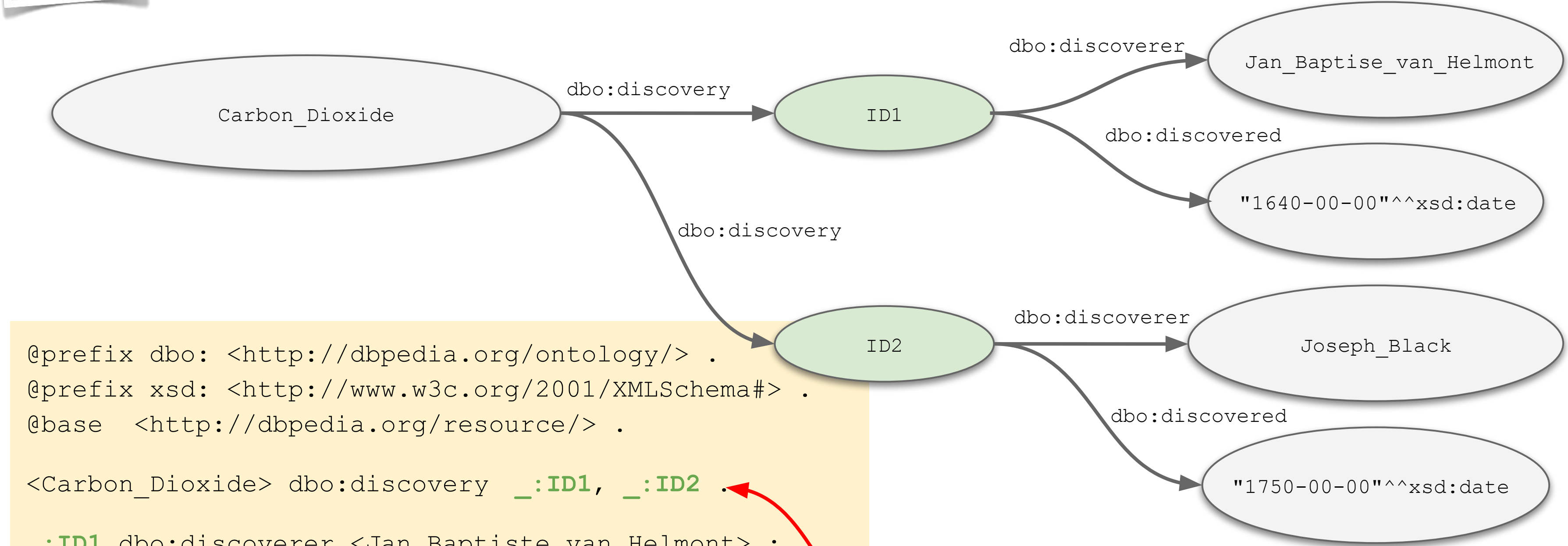
# RDF/Turtle



```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@base  <http://dbpedia.org/resource/> .

<Carbon_Dioxide> dbo:discovery [ dbo:discoverer <Jean_Baptist_van_Helmont> ;
                                 dbo:discovered "1640-00-00"^^xsd:date
                               ] ,
                               [ dbo:discoverer <Joseph_Black> ;
                                 dbo:discovered "1750-00-00"^^xsd:date
                               ] .
```

nested
anonymous blank
nodes

# RDF/Turtle



```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@base  <http://dbpedia.org/resource/> .

<Carbon_Dioxide> dbo:discovery  _:ID1, _:ID2 .

_:ID1 dbo:discoverer <Jan_Baptiste_van_Helmont> ;
      dbo:discovered "1640-00-00"^^xsd:date .

_:ID2 dbo:discoverer <John_Black> ;
      dbo:discovered "1750-00-00"^^xsd:date .
```

**dereferenceable blank nodes**
can only be referenced from
inside a document / graph

# More RDF Datastructures

- **RDF Containers** and **RDF Collections**
  - Data structures to enumerate any resources or literals
  - Open lists (RDF Containers) and closed lists (RDF Collections)
  - No new semantics, just “syntactic sugar”
- **RDF Reification**
  - Abstraction for making “statements about RDF triples”
  - Useful for modeling data provenance
    - e.g. to model
      *“DBpedia states that Carbon Dioxide has been discovered in 1824.”*

Further information: E. Prud'hommeaux, G. Carothers,
RDF 1.1 Turtle, Terse RDF Triple Language, W3C Recommendation 25 February 2014
https://www.w3.org/TR/turtle/

# Information Service Engineering

## Lecture 7: Knowledge Graphs - 2

3.1 Knowledge Representations and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things

3.5 Resource Description Framework (RDF) as simple Data Model

**3.6 Creating new Models with RDFS**

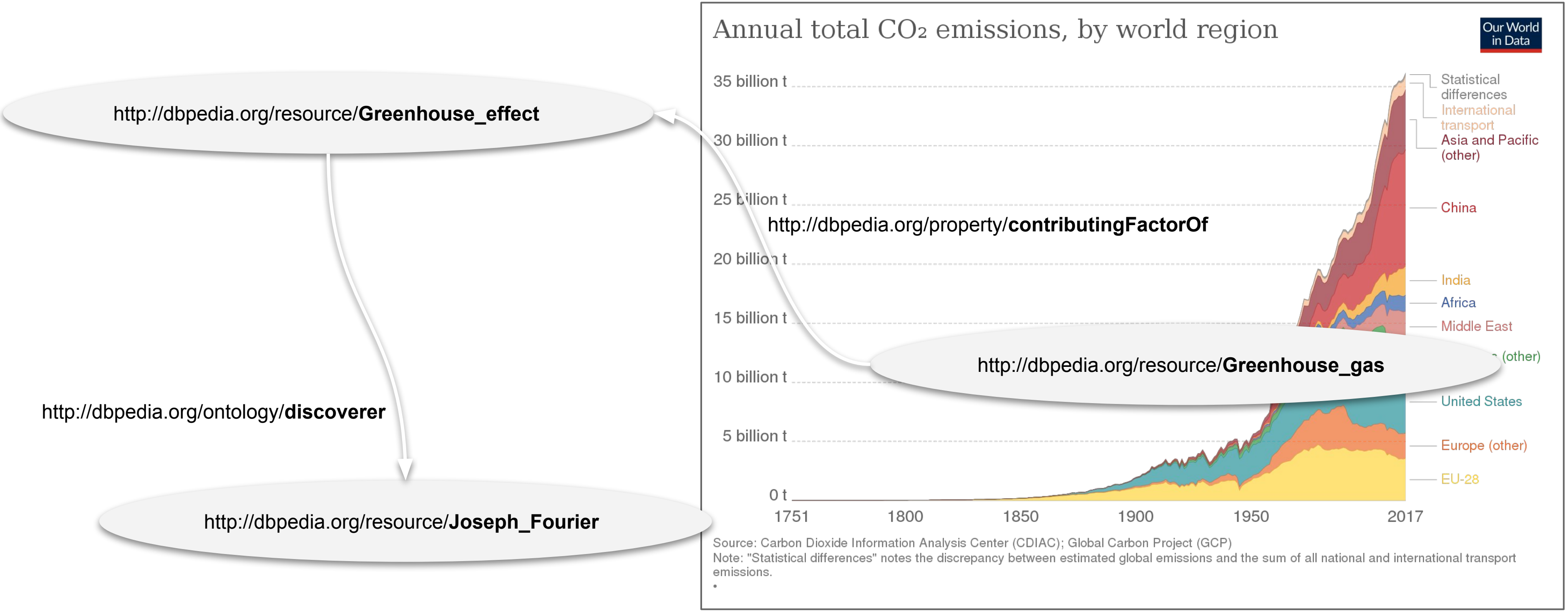
3.7 Knowledge Graphs

3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

# What does it really mean?



http://dbpedia.org/resource/**Greenhouse_effect**

http://dbpedia.org/property/**contributingFactorOf**

http://dbpedia.org/resource/**Greenhouse_gas**

http://dbpedia.org/ontology/**discoverer**

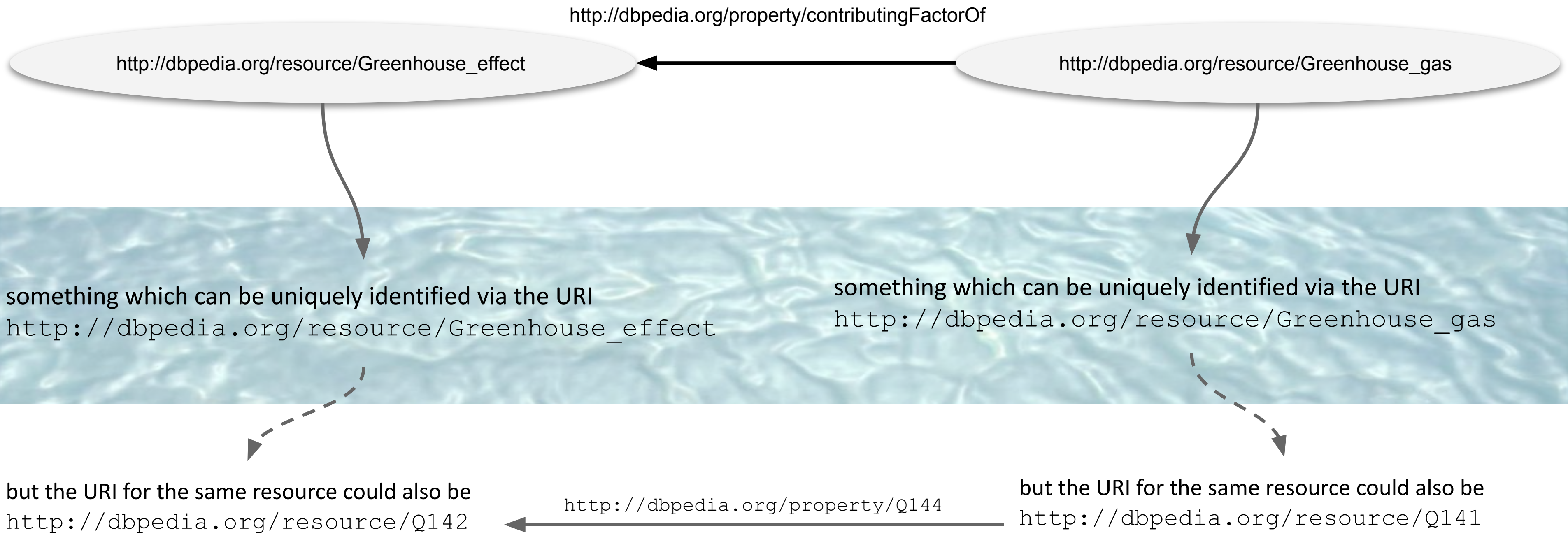
http://dbpedia.org/resource/**Joseph_Fourier**

Annual total CO₂ emissions, by world region

Source: Carbon Dioxide Information Analysis Center (CDIAC); Global Carbon Project (GCP)
Note: "Statistical differences" notes the discrepancy between estimated global emissions and the sum of all national and international transport emissions.

## Where does the intended meaning really come from?

# What does it really mean?



http://dbpedia.org/property/contributingFactorOf

http://dbpedia.org/resource/Greenhouse_effect

http://dbpedia.org/resource/Greenhouse_gas

something which can be uniquely identified via the URI
`http://dbpedia.org/resource/Greenhouse_effect`

something which can be uniquely identified via the URI
`http://dbpedia.org/resource/Greenhouse_gas`

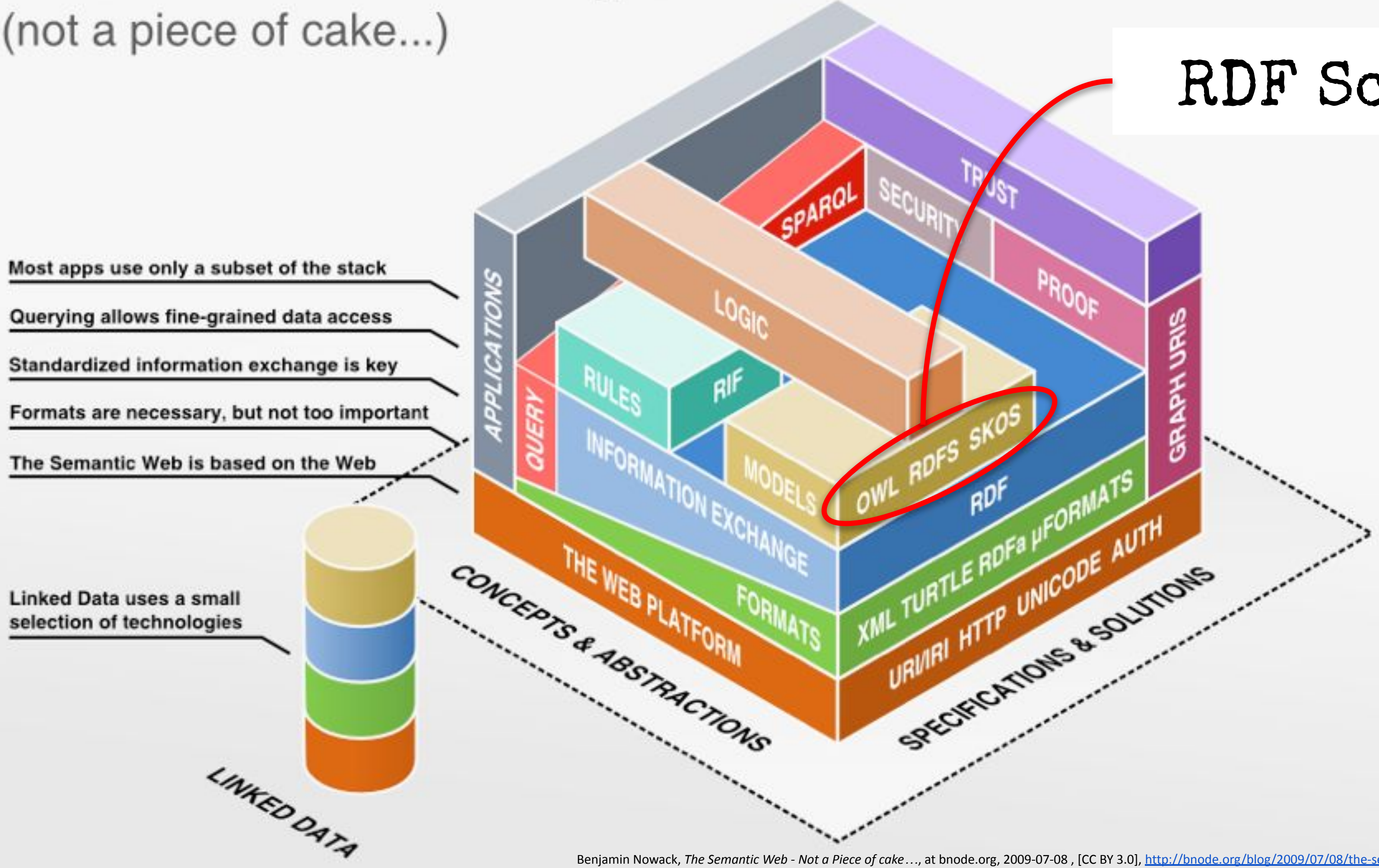
but the URI for the same resource could also be
`http://dbpedia.org/resource/Q142`

`http://dbpedia.org/property/Q144`

but the URI for the same resource could also be
`http://dbpedia.org/resource/Q141`

## We need more semantic expressivity...

# The Semantic Web Technology Stack
(not a piece of cake...)


Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small selection of technologies
LINKED DATA
APPLICATIONS
SPARQL
SECURITY
TRUST
PROOF
GRAPH URIS
LOGIC
RULES
RIF
QUERY
INFORMATION EXCHANGE
MODELS
OWL RDFS SKOS
RDF
FORMATS
XML TURTLE RDFa μFORMATS
THE WEB PLATFORM
URI/IRI HTTP UNICODE AUTH
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS

RDF Schema

Benjamin Nowack, *The Semantic Web - Not a Piece of cake*…, at bnode.org, 2009-07-08 , [CC BY 3.0], http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

# RDF Schema

- **RDF Schema**, officially called “**RDF Vocabulary Description Language**”

- RDF Schema allows:
  - Definition of **classes** via `rdfs:Class`
  - Class instantiation in RDF via `rdf:type`
  - Example:

```
:Greenhouse_gas   rdf:type rdfs:Class .
:Carbon_dioxide   rdf:type :Greenhouse_gas .
```

Carbon_dioxide ∈ Greenhouse_gas

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

# RDF Schema

- Definition of **properties** via `rdf:Property`
- Definition of **property restrictions on domain and range** via `rdfs:domain` and `rdfs:range`
- Example

```
:Person        rdf:type       rdfs:Class .
:discoverer    rdf:type       rdf:Property .
:discoverer    rdfs:domain    :Thing .
:discoverer    rdfs:range     :Person .
```

discoverer $\subseteq$ Thing × Person

# RDF Schema

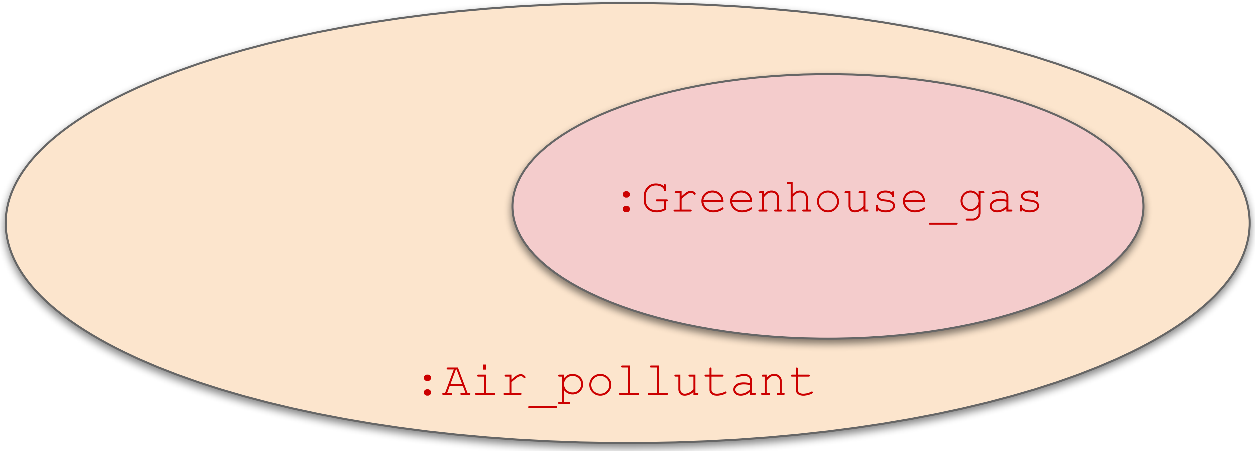
- Everything in the RDF model is a **resource.**
  - rdfs:Class rdf:type rdfs:Resource .
  - rdf:Property rdf:type rdfs:Resource .
  - rdfs:Literal rdf:type rdfs:Resource .
  - rdfs:XMLLiteral rdf:type rdfs:Resource .
  - rdfs:Datatype rdf:type rdfs:Resource .

# RDF Schema

- Definition of **hierarchical relationships**:
  - **Subclasses** and **superclasses** via `rdfs:subClassOf`
  - Example:

```
:Greenhouse_gas rdfs:subClassOf :Air_pollutant .
```

greenhouse_gas ⊆ Air_pollutant



# RDF Schema

- Definition of **hierarchical relationships**:
  - **Subclasses** and **superclasses** via `rdfs:subClassOf`
  - Example:

```
:Greenhouse_gas rdfs:subClassOf :Air_pollutant .
```

  - **Subproperties** and **superproperties** via `rdfs:subPropertyOf`
  - Example:

```
:sublimationTemperature rdfs:subPropertyOf
:temperature .
```

# RDF Schema

- Some more properties:
  - **`rdfs:seeAlso`**
    defines a relation of a resource to another, which explains it.
  - **`rdfs:isDefinedBy`**
    subproperty of `rdfs:seeAlso`, defines the relation of a resource to its definition.
  - **`rdfs:comment`**
    comment, usually as text.
  - **`rdfs:label`**
    „readable“ name of a resource (contrary to ID).

# RDFS Example

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@orefix owl:  <http://www.w3.org/2002/07/owl#>
@prefix :     <http://example.org/Climate#> .
```

```
:Greenhouse_gas      rdf:type        rdfs:Class ;
                     rdfs:subClassOf :Air_pollutant .
:Person              rdf:type        rdfs:Class .
:Scientist           rdfs:subClassOf  :Person .
:Physicist           rdfs:subClassOf  :Scientist .
:Chemist             rdfs:subClassOf  :Scientist .
```

Class Definitions

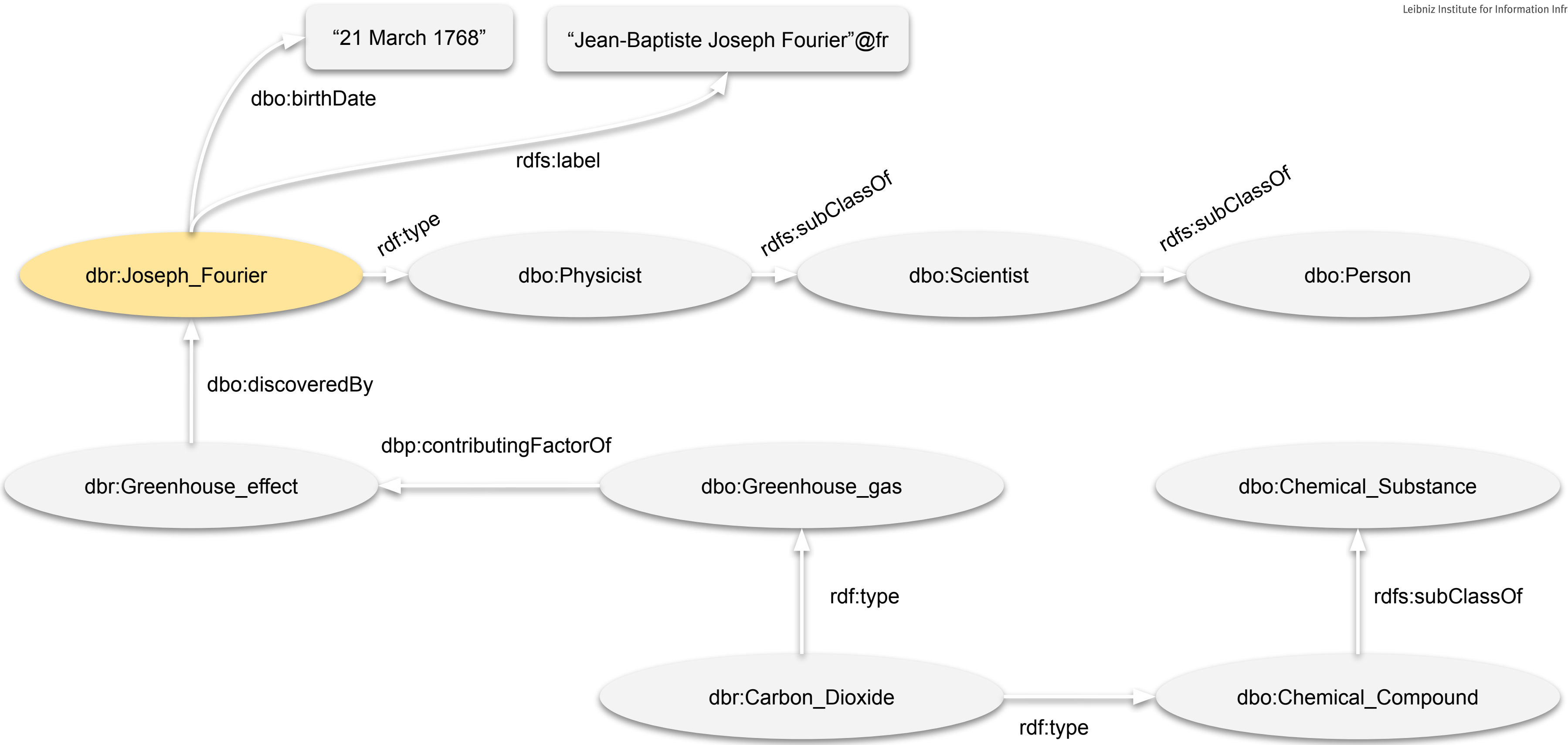
```
:discoverer      rdf:type     rdf:Property ;
                 rdfs:domain owl:Thing .
                 rdfs:range   :Person .
```

Property Definitions

```
:Carbon_dioxide              rdf:type      :Greenhouse_gas ;
                             :discoverer   :Jan_Baptist_van_Helmont ;
                             :discoverer   :Joseph_Black .
:Jan_Baptist_van_Helmont    rdf:type      :Physicist .
:Joseph_Black                rdf:type      :Chemist ;
                             rdfs:label    "Joseph Black"@en ;
                             rdfs:comment "co-discovered CO2" .
```

Instance Definitions

# How much knowledge (semantics) is there?



# RDF(S) Semantics

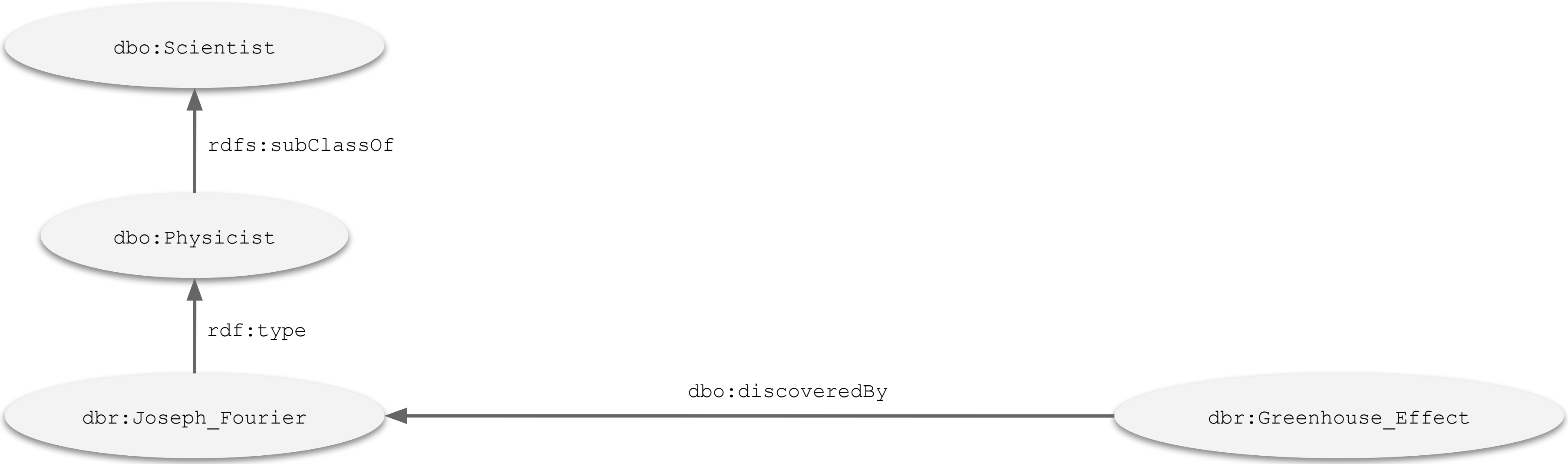
- In difference to other data definition languages RDF(S) is based on a **formal semantics**.
- Formal semantics enables RDF(S) to draw **valid** and **sound logical inferences**.

- Examples:
- dbr:Joseph_Fourier rdf:type dbo:Physicist .
- dbo:Physicist rdfs:subClassOf dbo:Scientist .
- dbo:discoveredBy rdfs:subPropertyOf dbo:perceivedBy

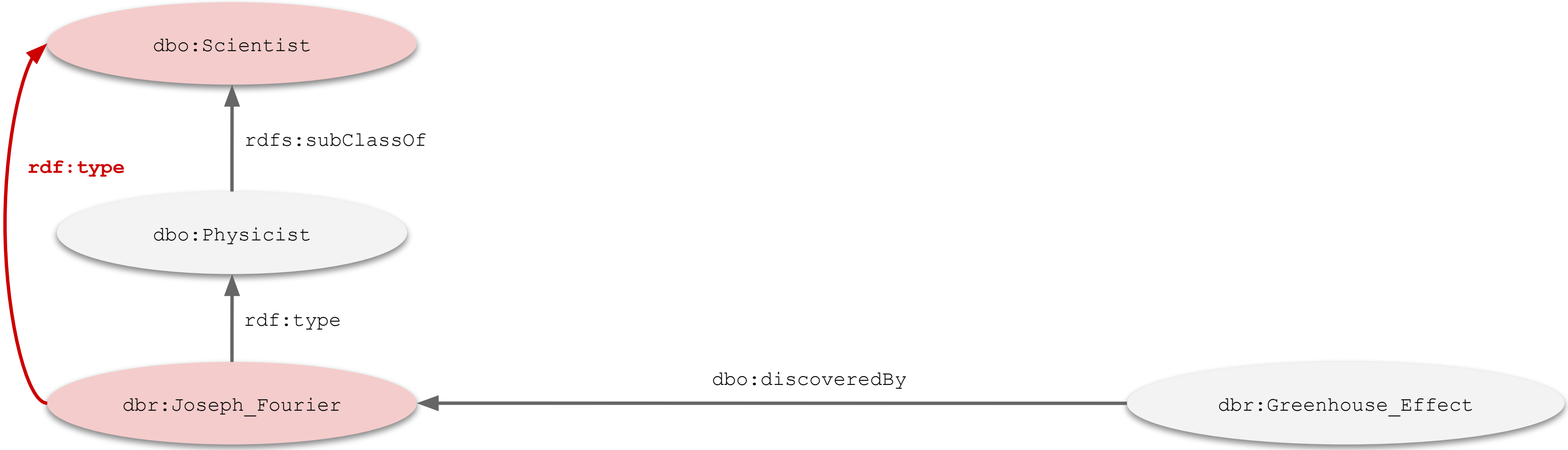
dbr:Joseph_Fourier ∈ dbo:Physicist
dbo:Physicist ⊆ dbo:Scientist
dbo:discoveredBy ⊆ dbo:perceivedBy

**Model-theoretic Semantics**

# Which Conclusions can we deduce with RDF(S) Semantics



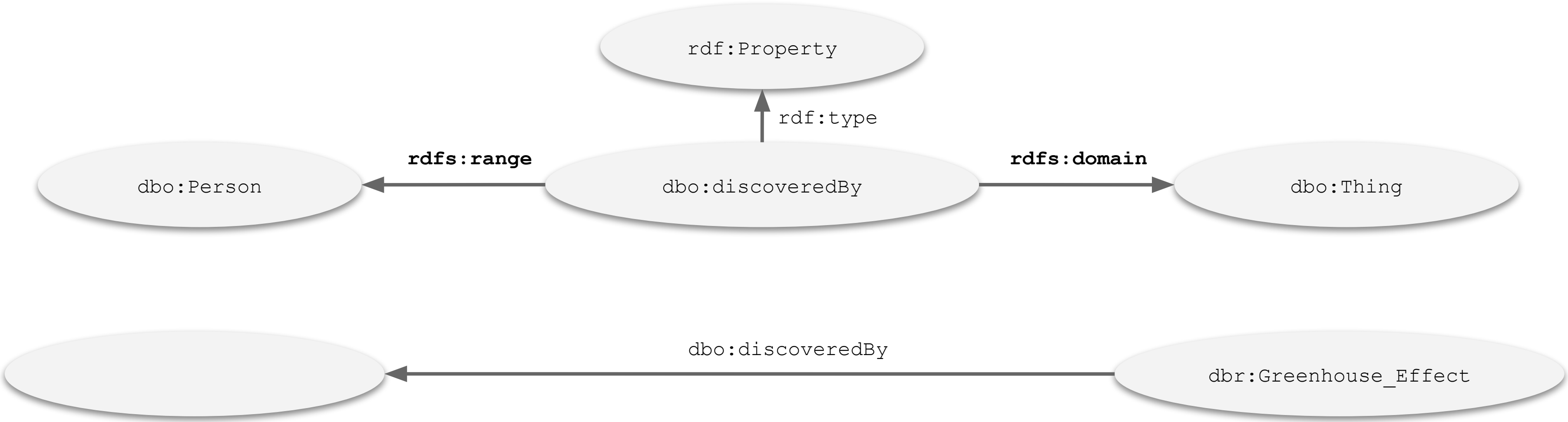
# Which Conclusions can we deduce with RDF(S) Semantics



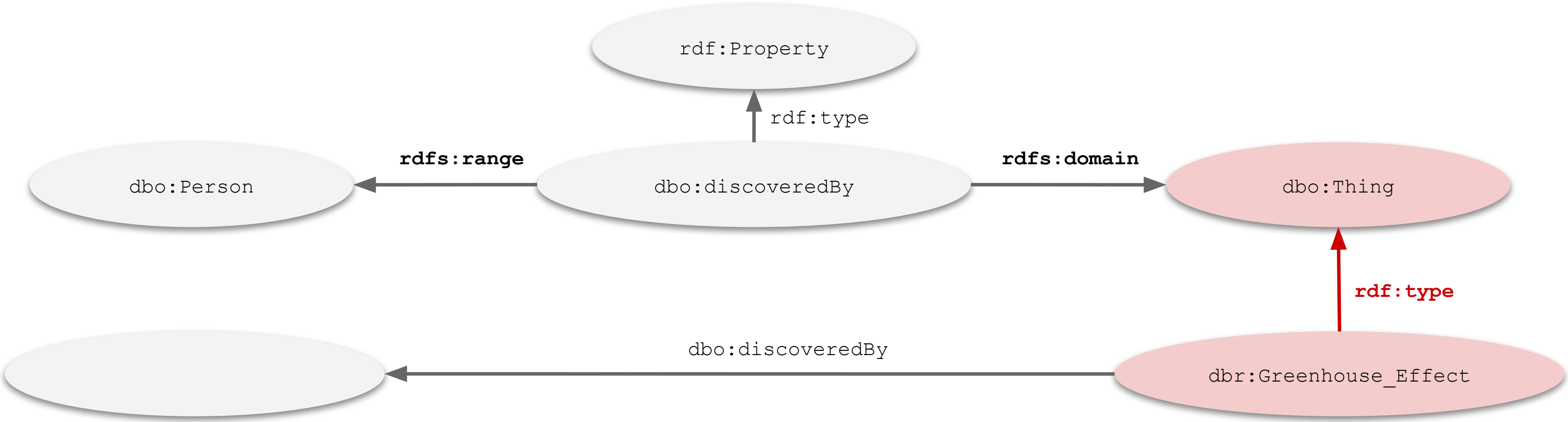
(1) Deduction of **new facts** from a **class hierarchy**.

$$\forall i, c_1, c_2: T(i, \text{rdf:type}, c_1) \wedge T(c_1, \text{rdfs:subClassOf}, c_2) \rightarrow T(i, \text{rdf:type}, c_2)$$

# Which Conclusions can we deduce with RDF(S) Semantics



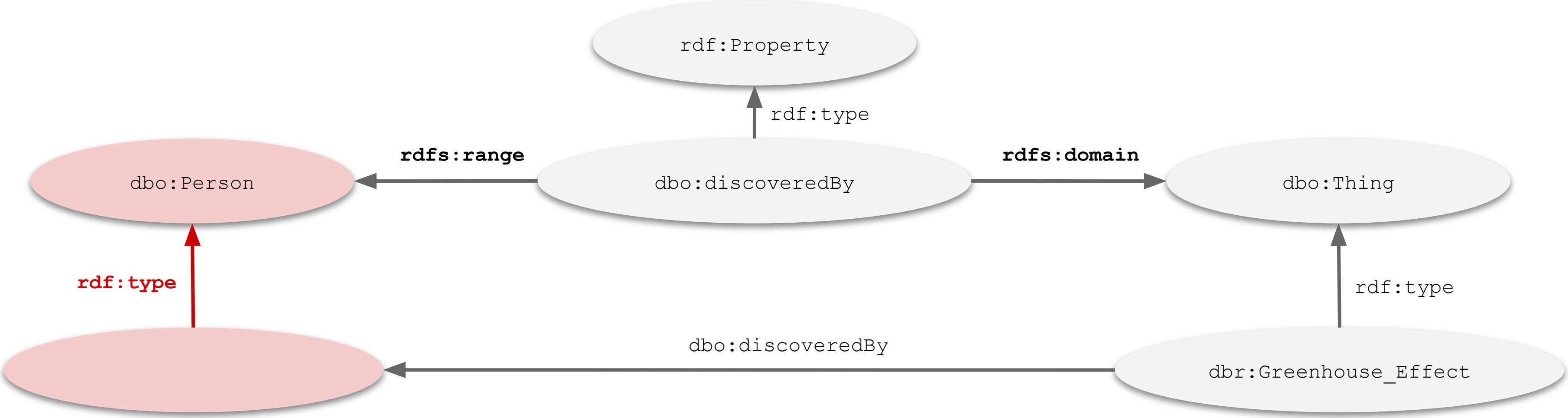
# Which Conclusions can we deduce with RDF(S) Semantics



(2) Deduction of entity **class membership** from the **domain** of one of its properties

$$\forall i_1, i_2, c_1, c_2, p: \quad T(i_1, p, i_2) \land T(p, \text{rdfs:domain}, c_1) \land T(p, \text{rdfs:range}, c_2) \rightarrow T(i_1, \text{rdf:type}, c_1) \land T(i_2, \text{rdf:type}, c_2)$$

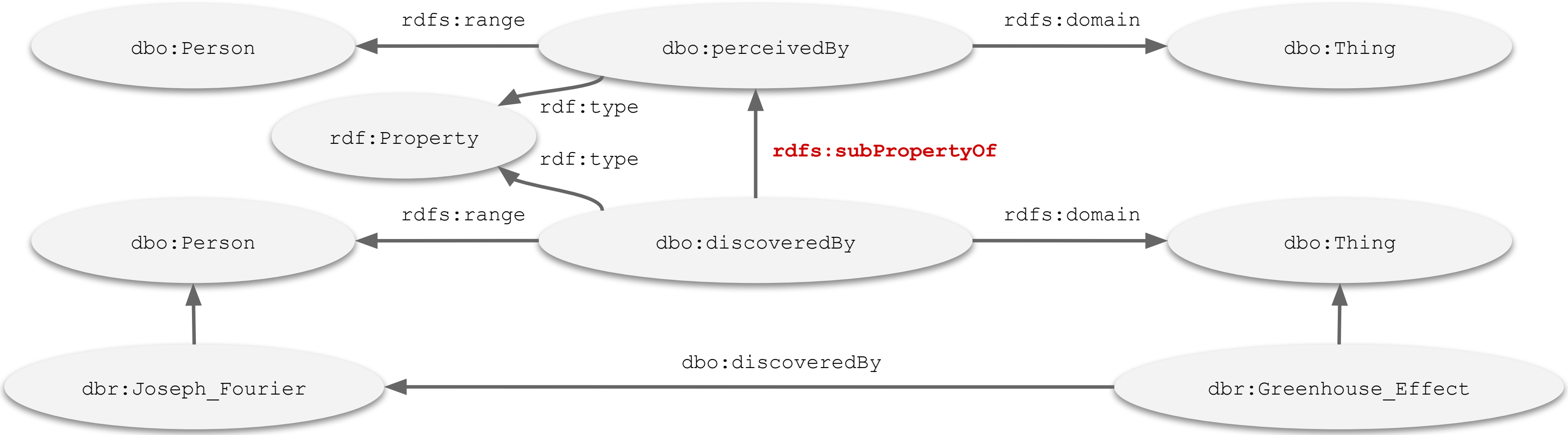
# Which Conclusions can we deduce with RDF(S) Semantics



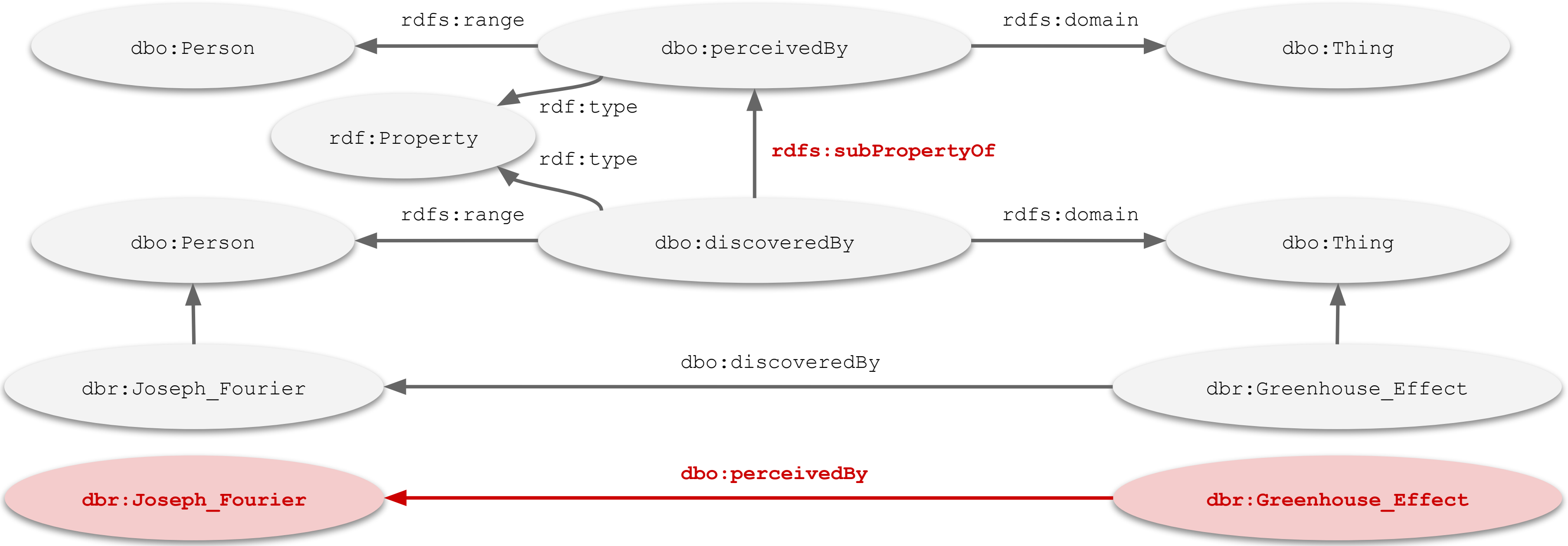
(3) Deduction of entity **class membership** from the **range** of one of its properties

$\forall i_1, i_2, c_1, c_2, p:\quad T(i_1, p, i_2) \wedge$
$T(p, \text{rdfs:domain}, c_1) \wedge T(p, \text{rdfs:range}, c_2)$
$\rightarrow T(i_1, \text{rdf:type}, c_1) \wedge T(i_2, \text{rdf:type}, c_2)$

# Which Conclusions can we deduce with RDF(S) Semantics



# Which Conclusions can we deduce with RDF(S) Semantics



(4) Deduction of **new facts** from **subproperty** relationships

$$\forall i_1, i_2, p_1, p_2: T(i_1, p_1, i_2) \land T(p_1, \text{rdfs:subPropertyOf}, p_2) \rightarrow T(i_1, p_2, i_2)$$

# Information Service Engineering

## Lecture 7: Knowledge Graphs - 2

3.1 Knowledge Representations and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things

3.5 Resource Description Framework (RDF) as simple Data Model

3.6 Creating new Models with RDFS

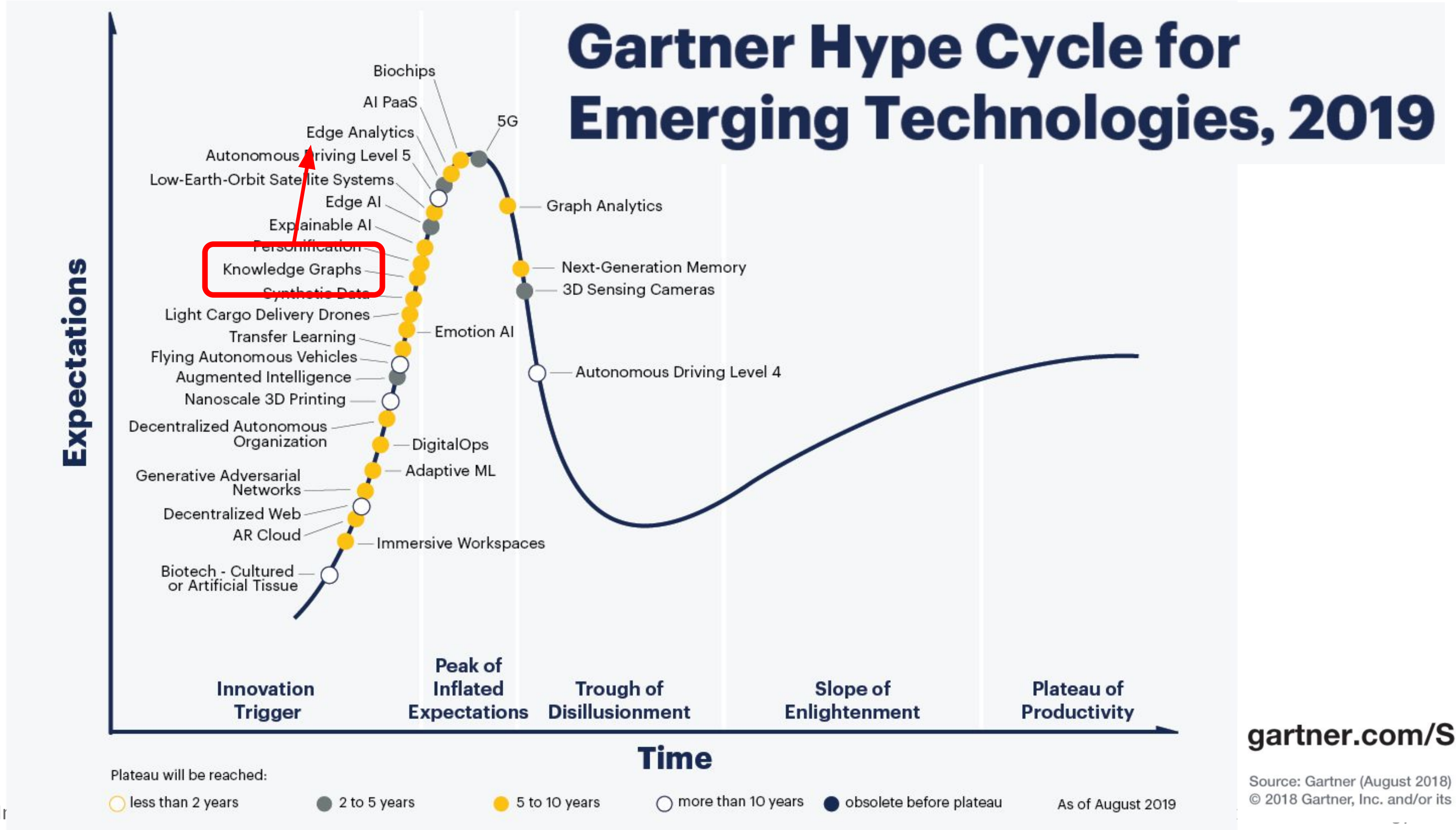
**3.7 Knowledge Graphs**

3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

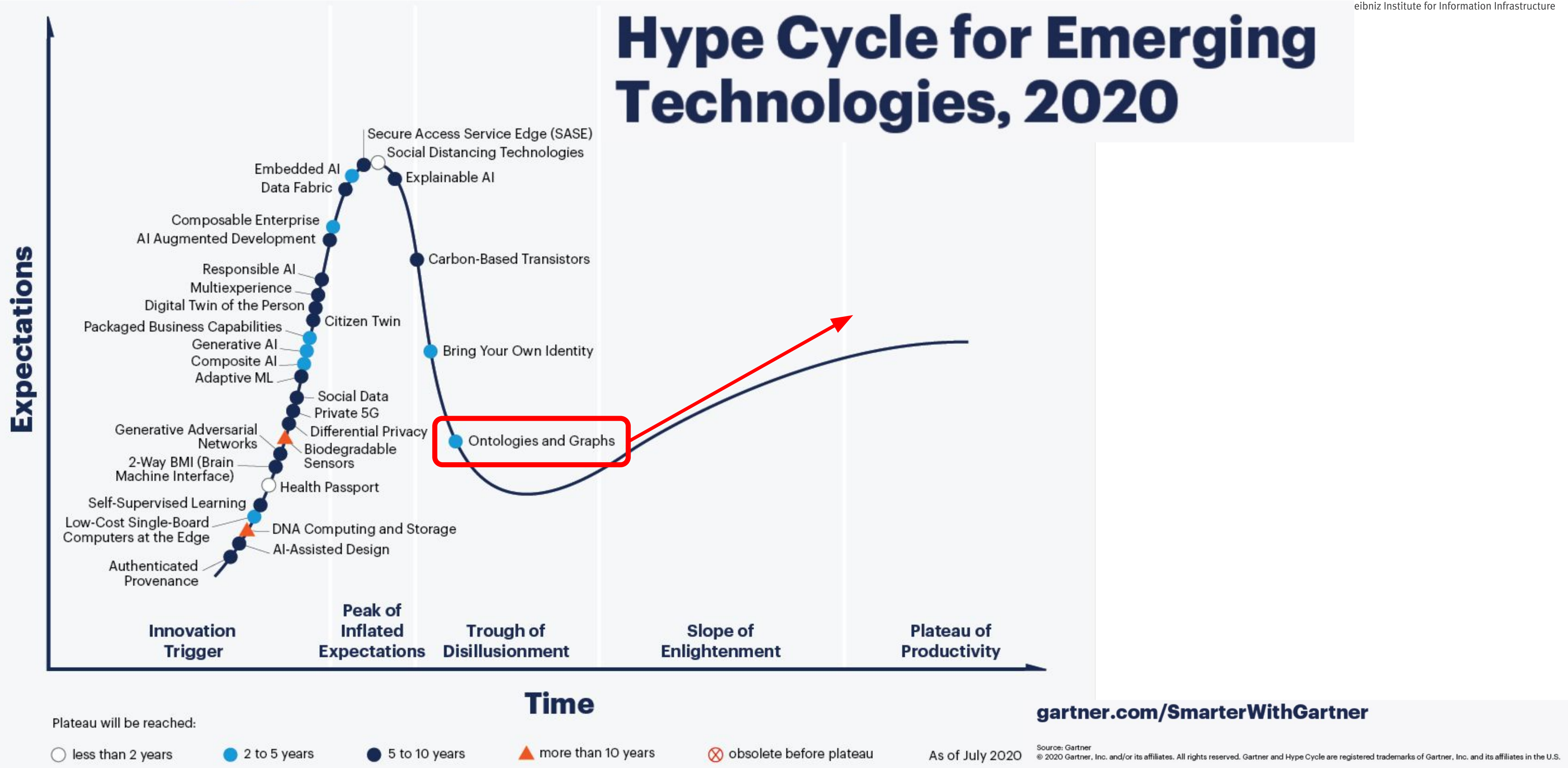
# Are Knowledge Graphs Relevant in Practice?



gartner.com/SmarterWithGartner

Source: Gartner (August 2018)

# Are Knowledge Graphs Relevant in Practice?

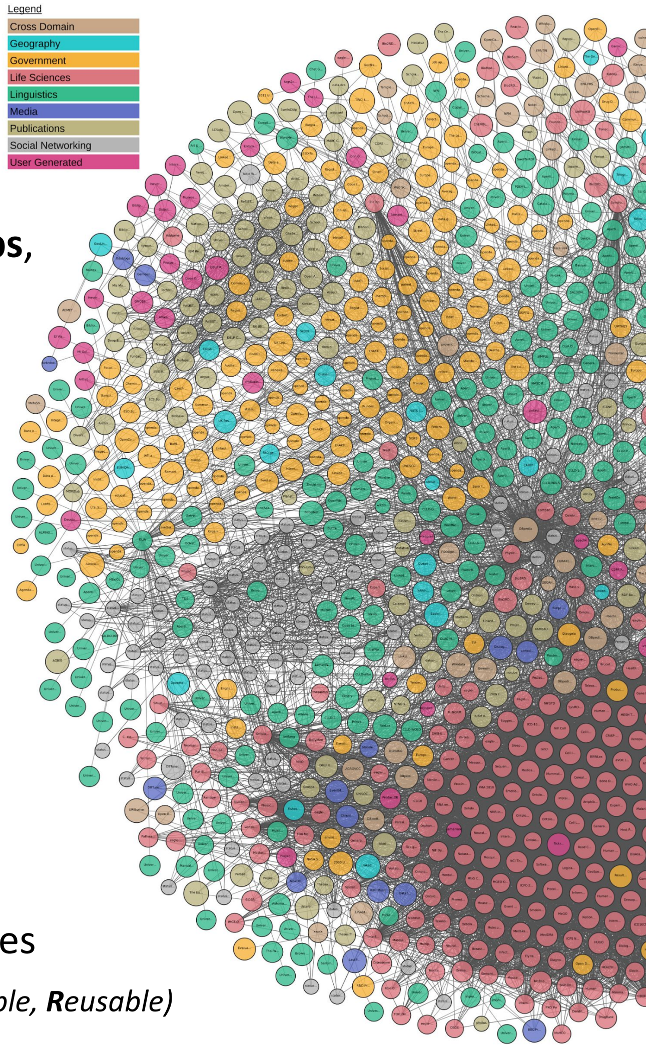


gartner.com/SmarterWithGartner

Source: Gartner
© 2020 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner and Hype Cycle are registered trademarks of Gartner, Inc. and its affiliates in the U.S.

# Knowledge Graphs - A Definition

- A **Graph** consisting of **concepts**, **classes**, **properties**, **relationships**, and **entity descriptions**
- Based on **formal knowledge representations** (RDF(S), OWL)
- Data can be **open** (e.g. DBpedia, WikiData), **private** (e.g. supply chain data), or **closed** (e.g. product models)
- Data can be **original**, **derived**, or **aggregated**
- We distinguish
  - **instance data** (ground truth),
  - **schema data** (vocabularies, ontologies)
  - **metadata** (e.g. provenance, versioning, licensing)
- **Taxonomies** are used to categorize entities
- **Links** exist between internal and external data
- Including **mappings** to data stored in other systems and databases
- *Fully compliant to **FAIR Data principles*** *(**F**indable, **A**ccessible, **I**nteroperable, **R**eusable)*

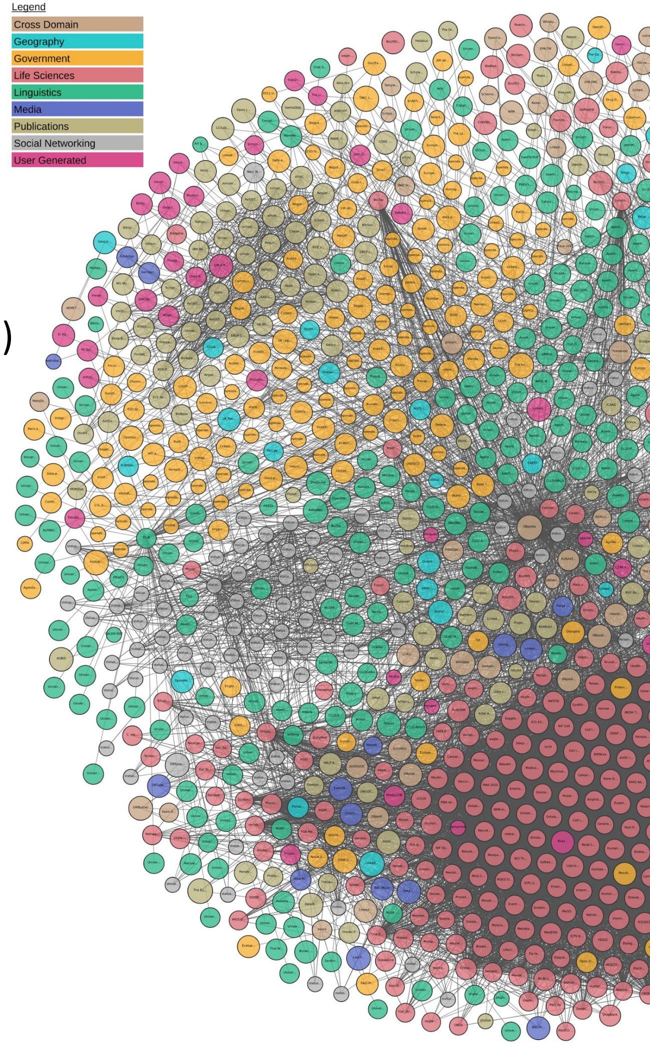


The Linked Open Data Cloud from lod-cloud.net

# Knowledge Graphs - A Definition (continued)

- A Knowledge Graph is a data set that is:
  - **structured** (in the form of a specific data structure)
  - **normalised** (consisting of small units, such as vertices and edges)
  - **connected** (defined by the - possibly distant - connections between objects)

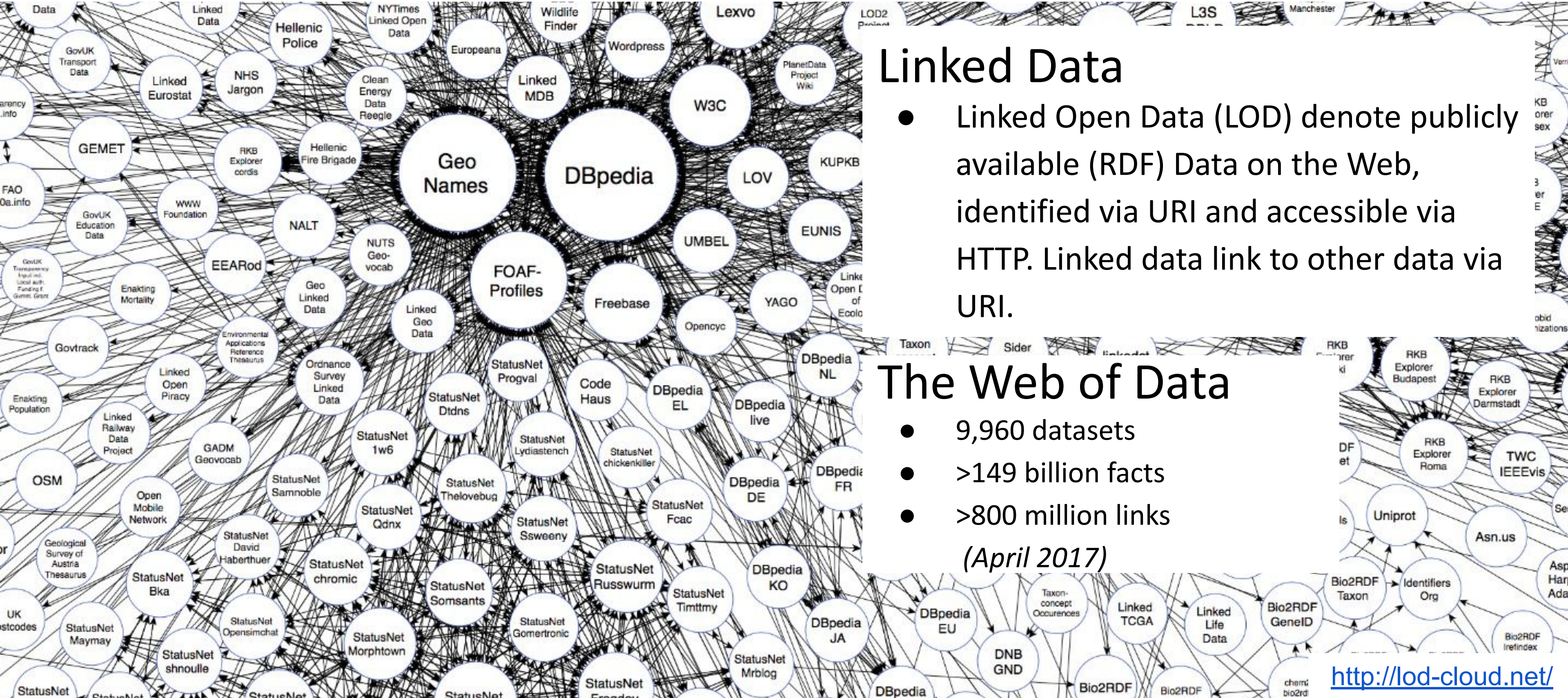
- Moreover, Knowledge Graphs are typically:
  - **explicit** (created purposefully with an intended meaning)
  - **declarative** (meaningful in itself, independent of a particular implementation or algorithm)
  - **annotated** (enriched with contextual information to record additional details and meta-data)
  - **non-hierarchical** (more than just a tree-structure)
  - **large** (millions rather than hundreds of elements)



The Linked Open Data Cloud from lod-cloud.net

https://iccl.inf.tu-dresden.de/w/images/f/ff/KG2018-Lecture-01-print.pdf

# The Web of Data - A Web of Knowledge Graphs



## Linked Data

- Linked Open Data (LOD) denote publicly available (RDF) Data on the Web, identified via URI and accessible via HTTP. Linked data link to other data via URI.

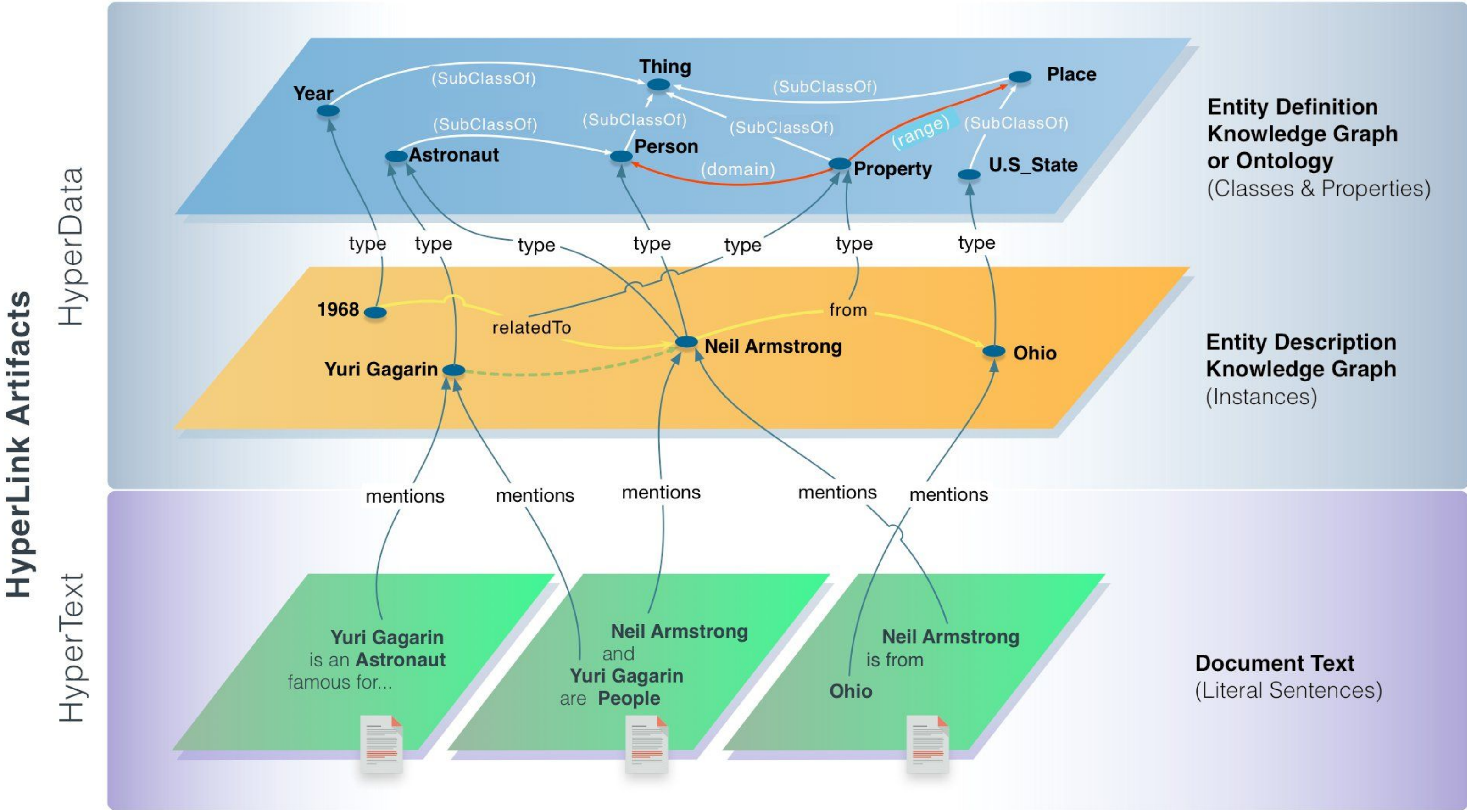
## The Web of Data

- 9,960 datasets
- >149 billion facts
- >800 million links

*(April 2017)*

http://lod-cloud.net/

# Knowledge Graph — for General Knowledge



**Entity Definition Knowledge Graph or Ontology** (Classes & Properties)

**Entity Description Knowledge Graph** (Instances)

**Document Text** (Literal Sentences)

Twitter @kidehen, 08.11.19,01:35

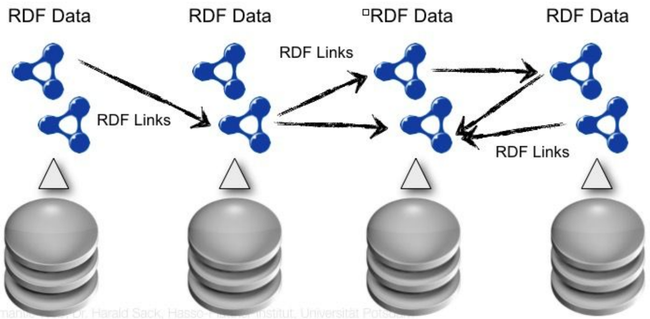
# Linked Data Principles

1. Use **URIs** as names for 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.

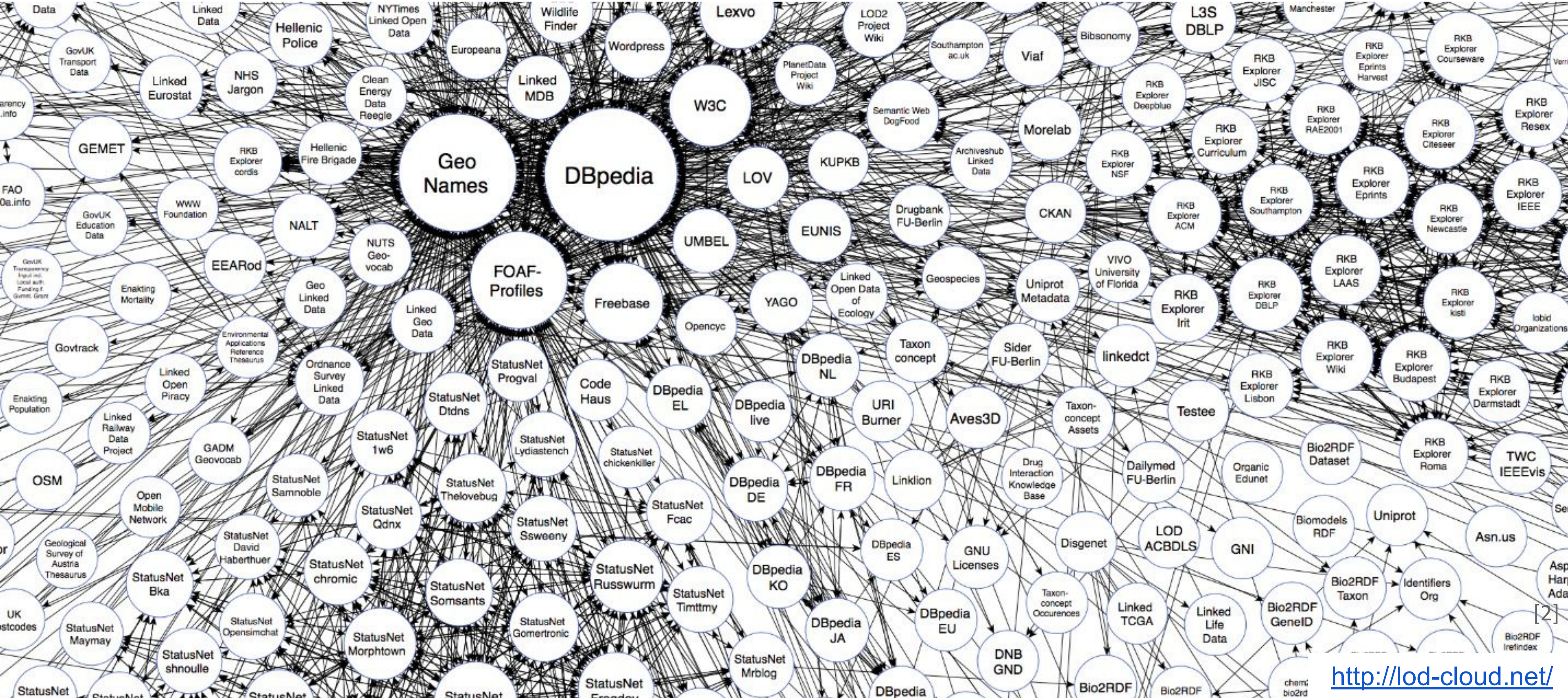
*(Tim Berners-Lee, Linked Data, 2006, [http://www.w3.org/DesignIssues/LinkedData.html](http://www.w3.org/DesignIssues/LinkedData.html))*

# Advantages of Linked Open Data vs. APIs

- **Simple** and **generic API** for various heterogeneous data sources enables **simple reuse** and **data sharing** among applications.
- **RDF Data model** guarantees (simple) **extensibility.**
- **Transport via http**, standard Port 80, prevents firewall adaption.
- **Ontologies** enable **meaningful connections** between data sources.
- **Reasoning** over Linked Data enables to **generate new knowledge**, i.e. inference from implicit to explicit knowledge.



# What keeps the Web of Linked Data together?



[2]

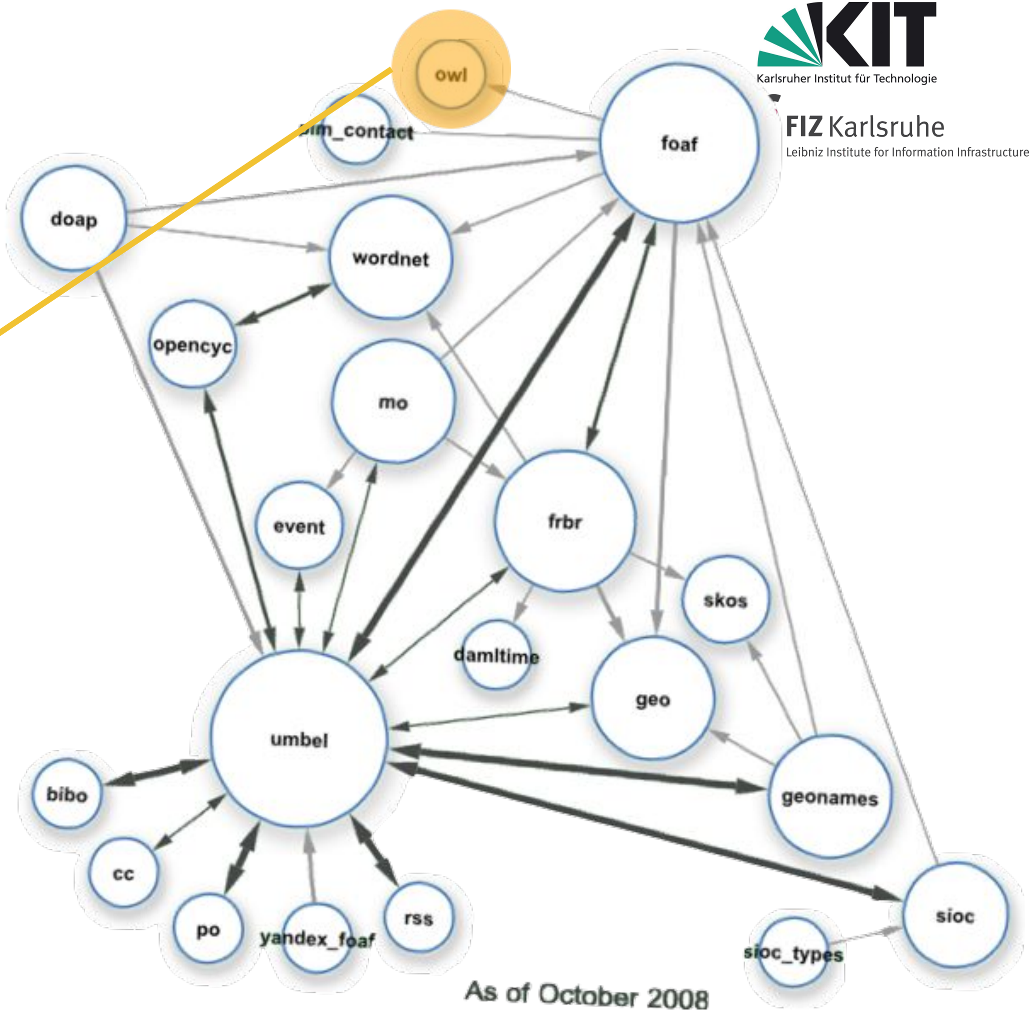
http://lod-cloud.net/

# Ontologies to Bind Them...

- Ontologies that keep the Web of Data together

- **OWL**
  `owl:sameAs` connects identical individuals
  `owl:equivalentClass` connects equivalent classes

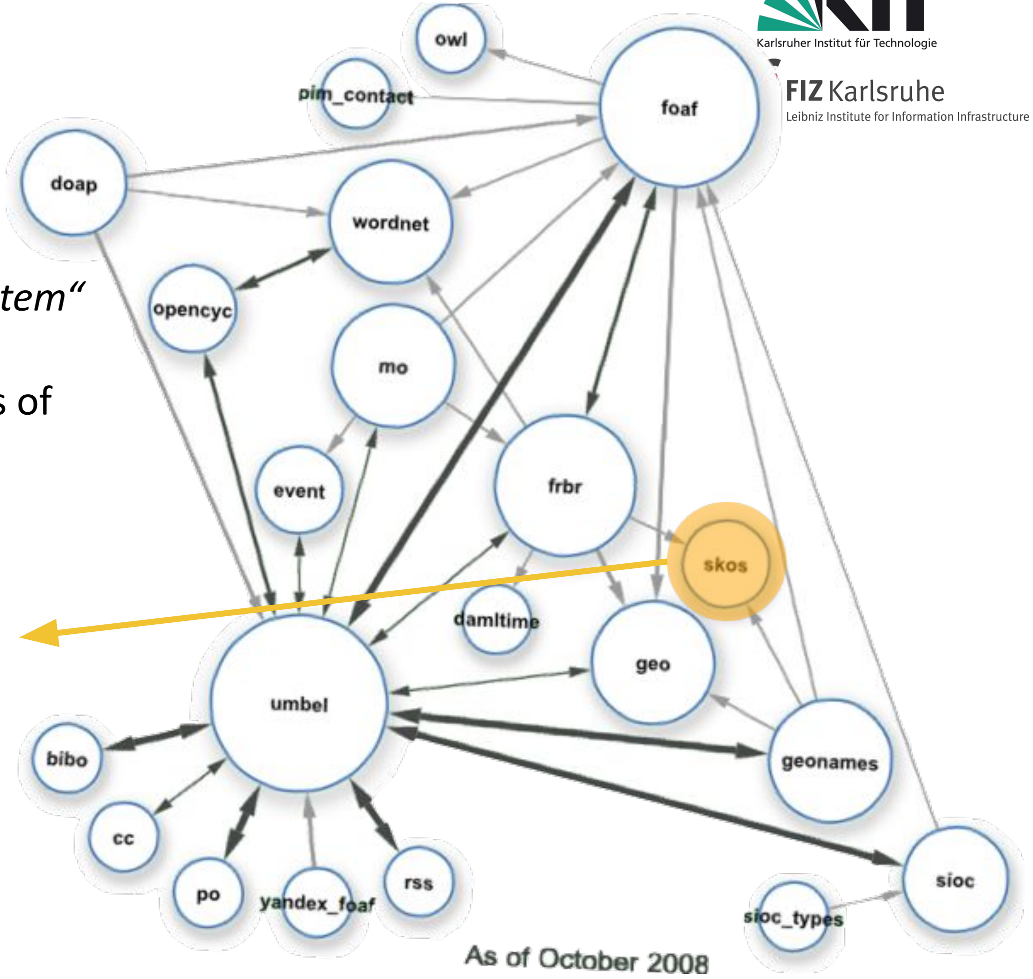
```
dbr:Joseph_Fourier owl:sameAs wd:Q8772 .
```



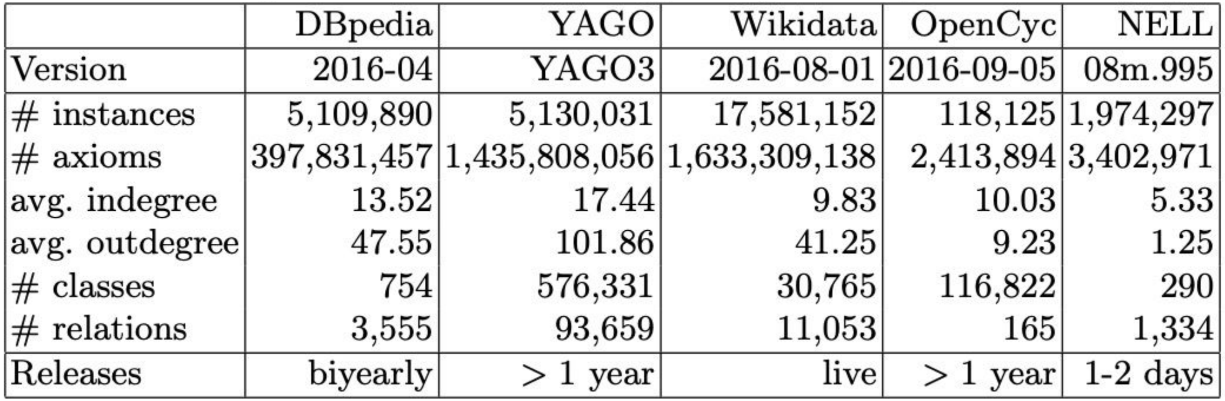
As of October 2008

# Ontologies to Bind Them...

- **SKOS**
  - *„Simple Knowledge Organization System“*
  - Based on RDF and RDFS
  - Applied for definitions and mappings of vocabularies and ontologies
    - `skos:Concept (=classes)`
    - `skos:narrower`
    - `skos:broader`
    - `skos:related`
    - **`skos:exactMatch, skos:narrowMatch, skos:broadMatch, skos:relatedMatch`**

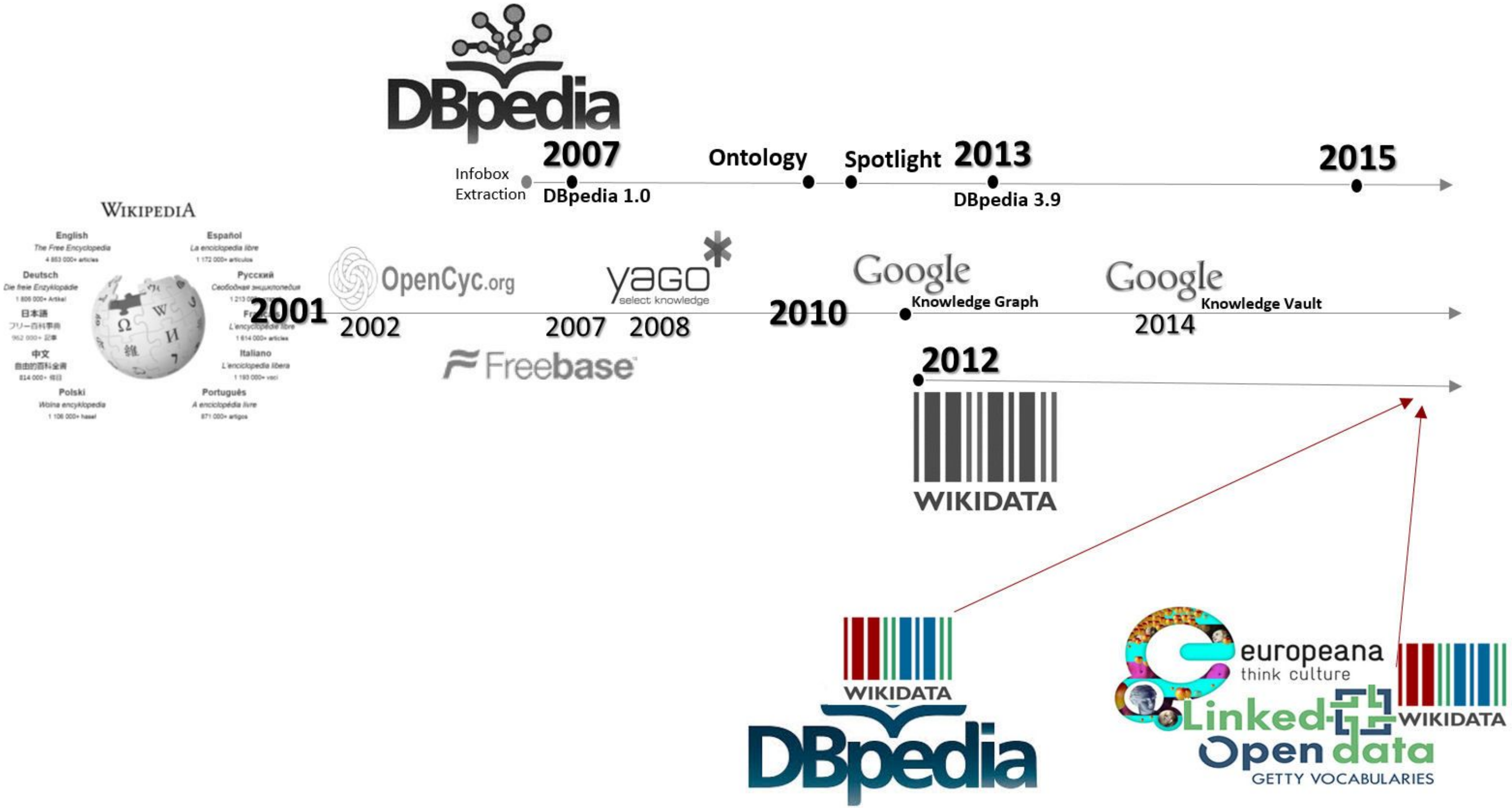


# Popular Publicly Available Knowledge Graphs



| | DBpedia | YAGO | Wikidata | OpenCyc | NELL |
|---|---:|---:|---:|---:|---:|
| Version | 2016-04 | YAGO3 | 2016-08-01 | 2016-09-05 | 08m.995 |
| # instances | 5,109,890 | 5,130,031 | 17,581,152 | 118,125 | 1,974,297 |
| # axioms | 397,831,457 | 1,435,808,056 | 1,633,309,138 | 2,413,894 | 3,402,971 |
| avg. indegree | 13.52 | 17.44 | 9.83 | 10.03 | 5.33 |
| avg. outdegree | 47.55 | 101.86 | 41.25 | 9.23 | 1.25 |
| # classes | 754 | 576,331 | 30,765 | 116,822 | 290 |
| # relations | 3,555 | 93,659 | 11,053 | 165 | 1,334 |
| Releases | biyearly | > 1 year | live | > 1 year | 1-2 days |

# Knowledge Bases Timeline



DBpedia

Infobox Extraction

2007

DBpedia 1.0

Ontology

Spotlight

2013

DBpedia 3.9

2015

WIKIPEDIA

2001

OpenCyc.org

2002

Freebase

yago select knowledge

2007

2008

2010

Google Knowledge Graph

Google Knowledge Vault

2014

2012

WIKIDATA

WIKIDATA DBpedia

europeana think culture

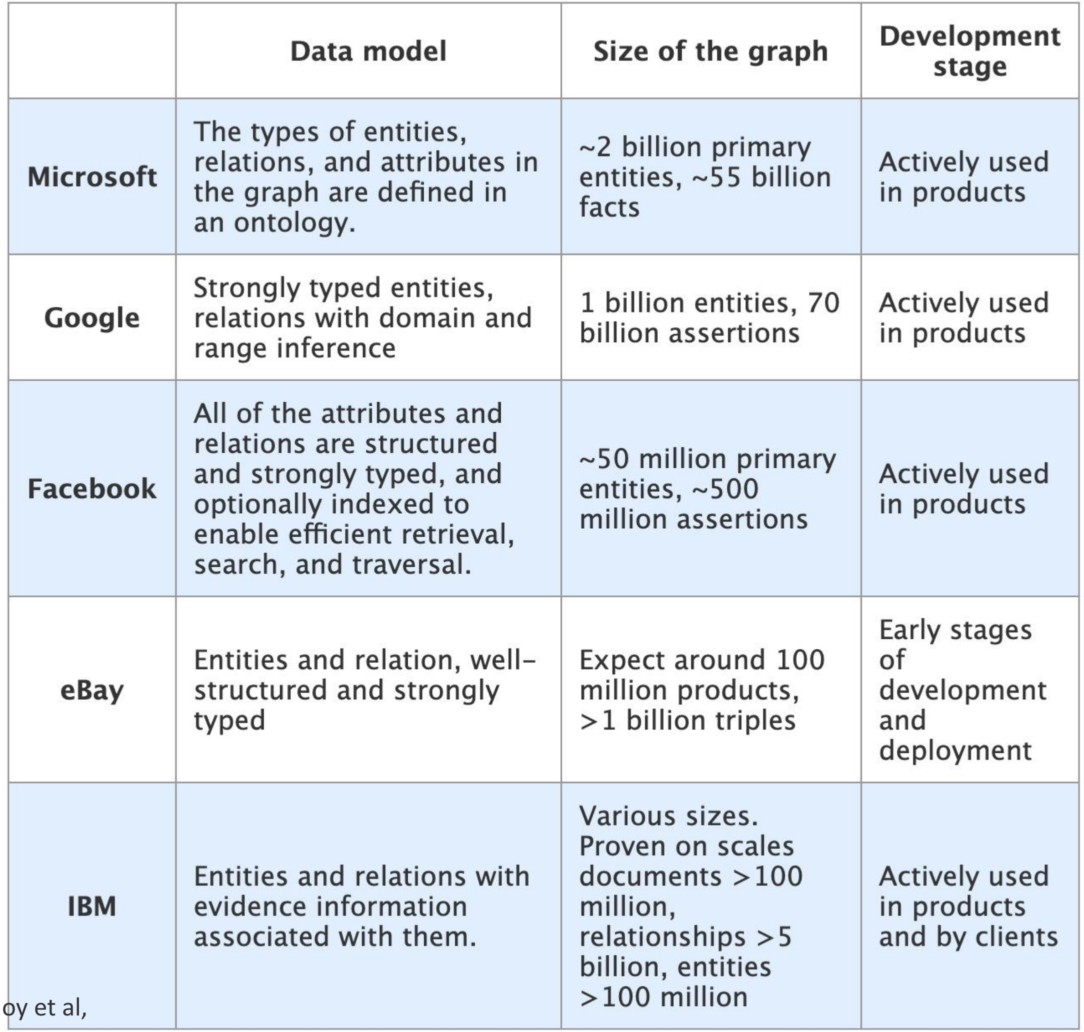
Linked Open data

GETTY VOCABULARIES

WIKIDATA

[3]

# Popular (Proprietary) Knowledge Graphs



| | Data model | Size of the graph | Development stage |
|---|---|---|---|
| **Microsoft** | The types of entities, relations, and attributes in the graph are defined in an ontology. | ~2 billion primary entities, ~55 billion facts | Actively used in products |
| **Google** | Strongly typed entities, relations with domain and range inference | 1 billion entities, 70 billion assertions | Actively used in products |
| **Facebook** | All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal. | ~50 million primary entities, ~500 million assertions | Actively used in products |
| **eBay** | Entities and relation, well-structured and strongly typed | Expect around 100 million products, >1 billion triples | Early stages of development and deployment |
| **IBM** | Entities and relations with evidence information associated with them. | Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million | Actively used in products and by clients |

Noy et al, *Industry-scale Knowledge Graphs: Lessons and Challenges* (2019)

# Information Service Engineering

## Next Lecture: Knowledge Graphs - 3

3.1 Knowledge Representations and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things

3.5 Resource Description Framework (RDF) as simple Data Model

3.6 Creating new Models with RDFS

3.7 Knowledge Graphs

**3.8 Querying Knowledge Graphs with SPARQL**

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

# 3. Knowledge Graphs - 2

## Bibliography

- E. Prud'hommeaux, G. Carothers,
  RDF 1.1 Turtle, Terse RDF Triple Language, W3C Recommendation 25 February 2014.
  https://www.w3.org/TR/turtle/

- S. Hitzler, S. Rudolph,
  Foundations of Semantic Web Technologies, Chapman / Hall, 2009.

- M. Krötzsch, Knowledge Graphs, TU Dresden, WS2018.

- *Interested in the Greenhouse effect?*
  P. Lynch, How Joseph Fourier discovered the greenhouse effect, Irish Times,
  Mar 21, 2019.

# 3. Knowledge Graphs - 2

## Syllabus Questions

- How does RDF represent information?
- What is the difference between entities and (typed) literals in the role of RDF objects?
- What can RDF blank nodes be used for?
- What is the difference between RDF blank nodes and regular RDF nodes?
- Why is RDF alone not sufficient for knowledge representation?
- What is RDFS used for?
- What are the limits of RDF?
- What is DBpedia?
- What is a Wikipedia Infobox?
- What is a Knowledge Graph?
- Explain the Linked Data Principles.
- What are the advantages of Linked Data vs. proprietary APIs?
- What kind of links glue the Web of Data (Linked Data Cloud) together?
- Why are Knowledge Graphs fully compliant to the FAIR principles?