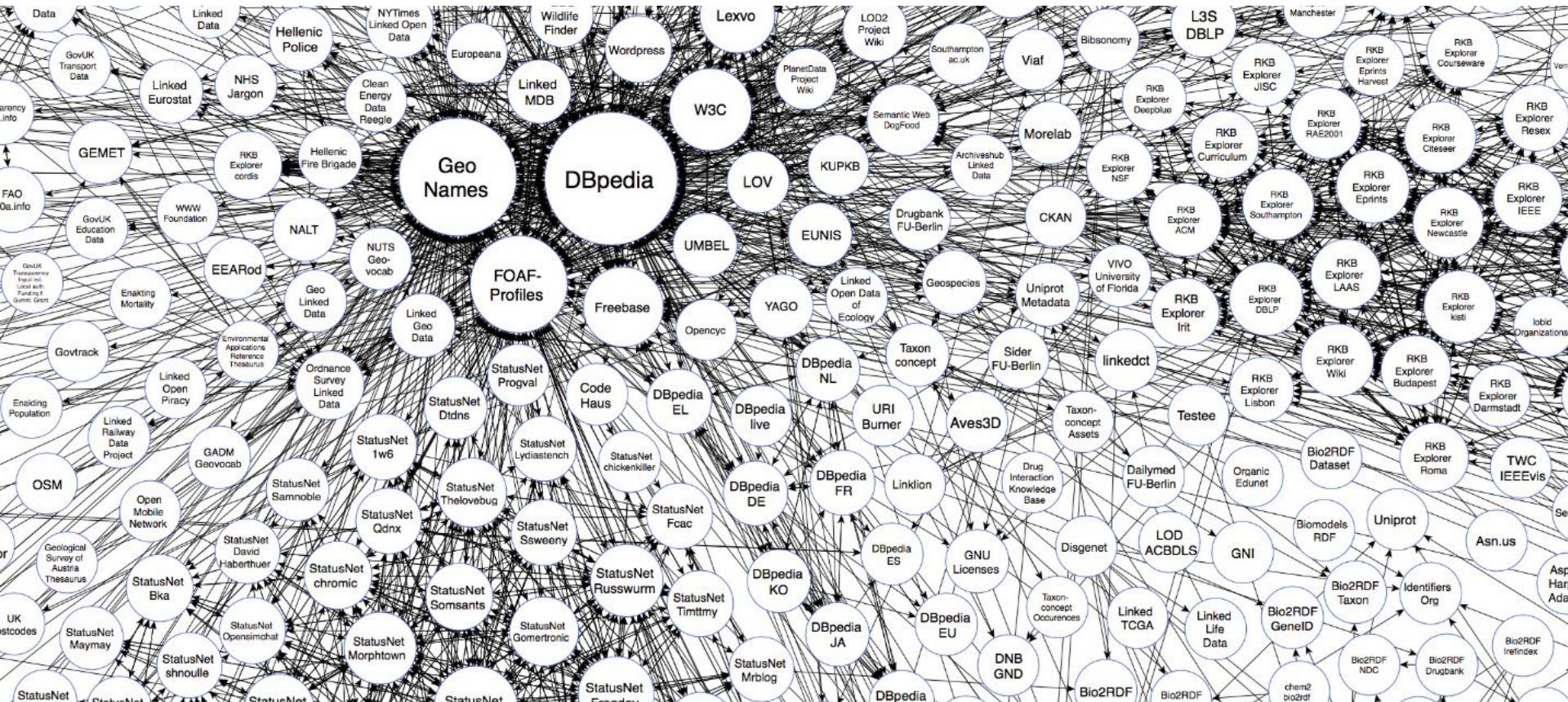


# Information Service Engineering

# 3. Linked Data Engineering



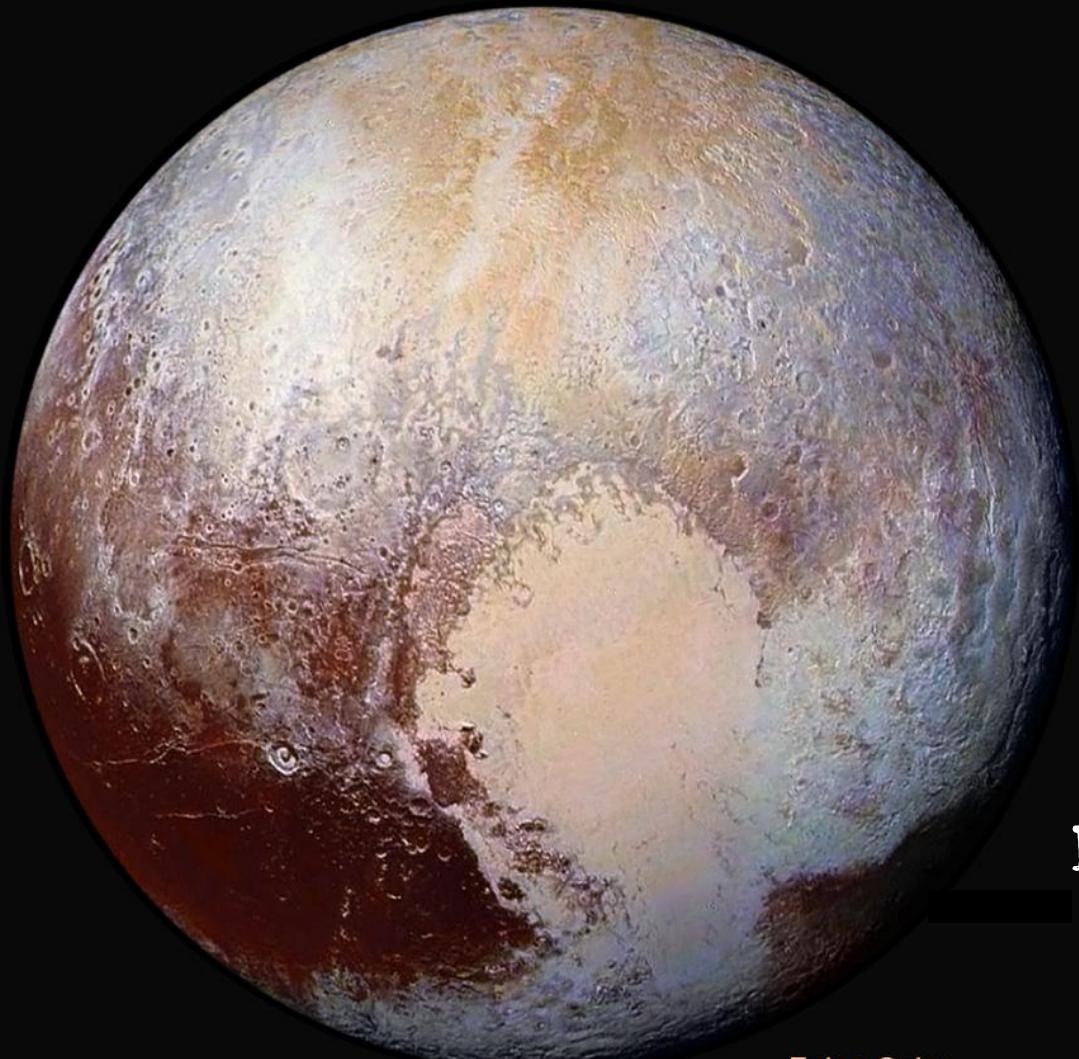
# Information Service Engineering

## Lecture 6: Linked Data Engineering - 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 name Things - URIs
- 3.5 Resource Description Framework (RDF) as simple Data Model
- 3.6 Creating new Models with RDFS
- 3.7 Querying RDF(S) with SPARQL
- 3.8 More Expressivity with Web Ontology Language (OWL)
- 3.9 Wikipedia, DBpedia, and Wikidata
- 3.10 Linked Data Programming

# Understanding Information

PLUTO  
Never Forget!  
(1930–2006)



**PLUTO**  
Never Forget!  
(1930–2006)

# Information Understanding



## Disambiguation

- solution of linguistic ambiguities

Text: "Pluto"

Pluto

Pluto

Pluto

HMS Pluto

...

Pluto

Entity Mapping  
Disambiguation

a Disney cartoon character

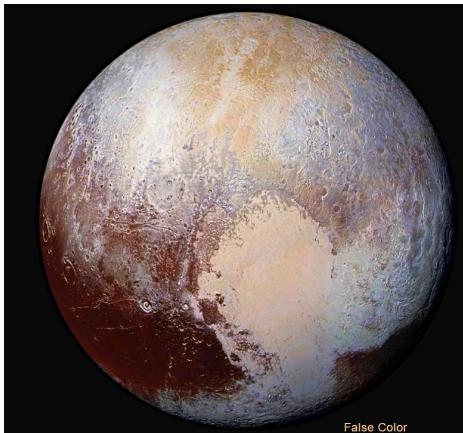
a Roman god

a song by Björk

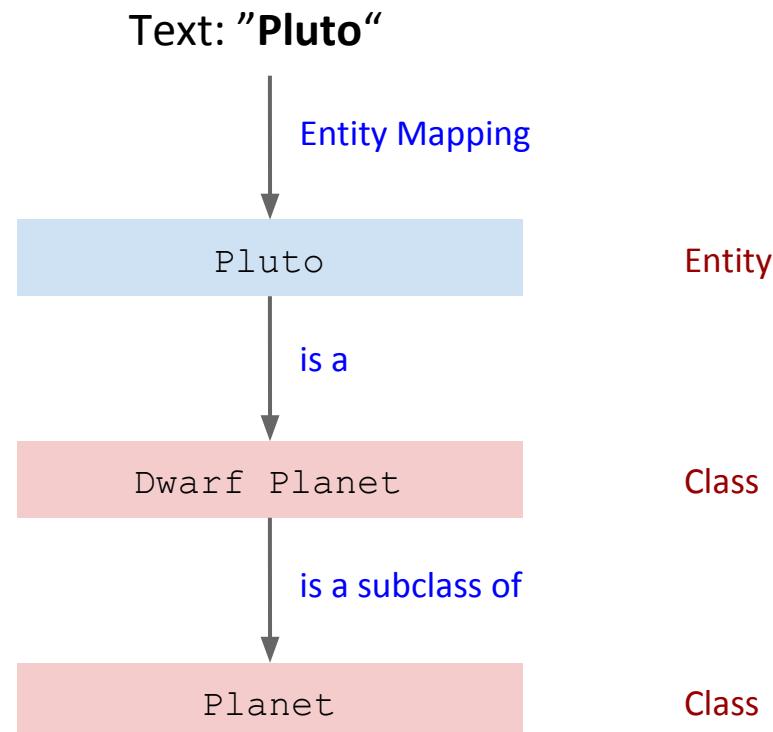
a ship

a dwarf planet

# Understanding Information



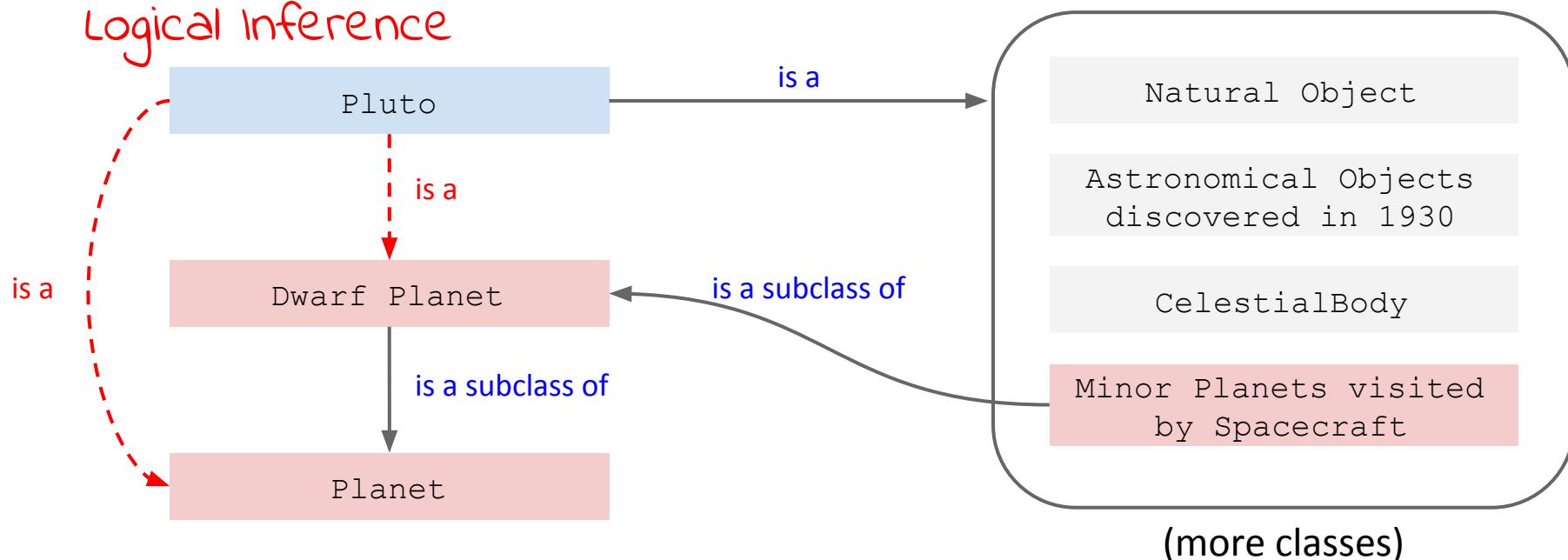
- The **Meaning (Semantics)** of entities and classes must be defined explicitly.



# Understanding Information

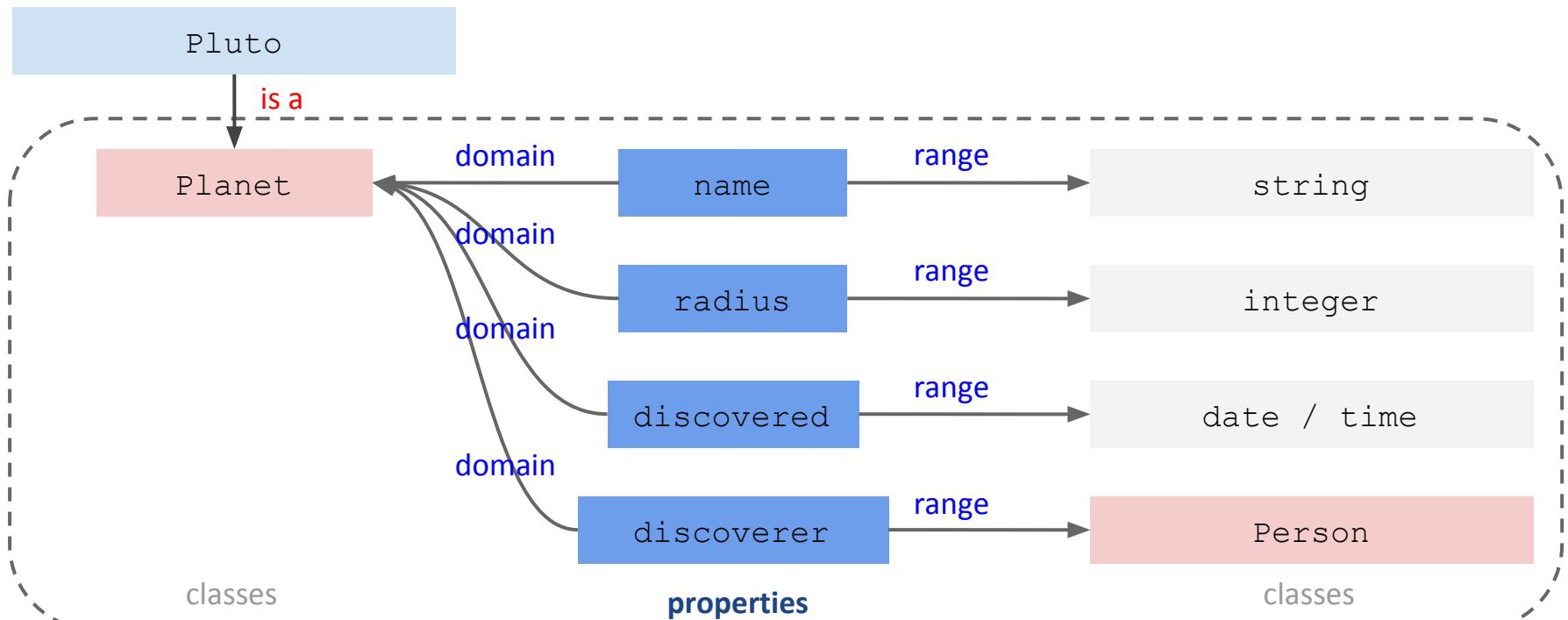
- The Meaning (Semantics) is expressed with the help of knowledge representations (**Ontologies**)

## Logical Inference



# Understanding Information

- The Meaning (Semantics) is expressed with the help of knowledge representations (**Ontologies**)



# Ontologies in Computer Science

"An ontology is an **explicit, formal specification of a shared conceptualization**. The term is borrowed from philosophy, where an Ontology is a systematic account of existence. For AI systems, what ‘exists’ is that which can be represented.“

*according to Thomas R. Gruber: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2):199-220, 1993.*

- |                           |   |
|---------------------------|---|
| <b>conceptualization:</b> | abstract model<br>(domain, identified relevant concepts, relations) |
| <b>explicit:</b>          | meaning of all concepts must be defined                             |
| <b>formal:</b>            | machine understandable  |
| <b>shared:</b>            | consensus about ontology  |

# How to represent Ontologies?

- Ontologies can be represented via **Classes, Relations and Instances**
- **Classes** are abstract groups, sets, or collections of objects and represent **ontology concepts**
- Classes are characterised via **attributes**
- **Attributes** are name-value pairs

*„The address contains the name, title, and place of residence of the person addressed“*

informal description

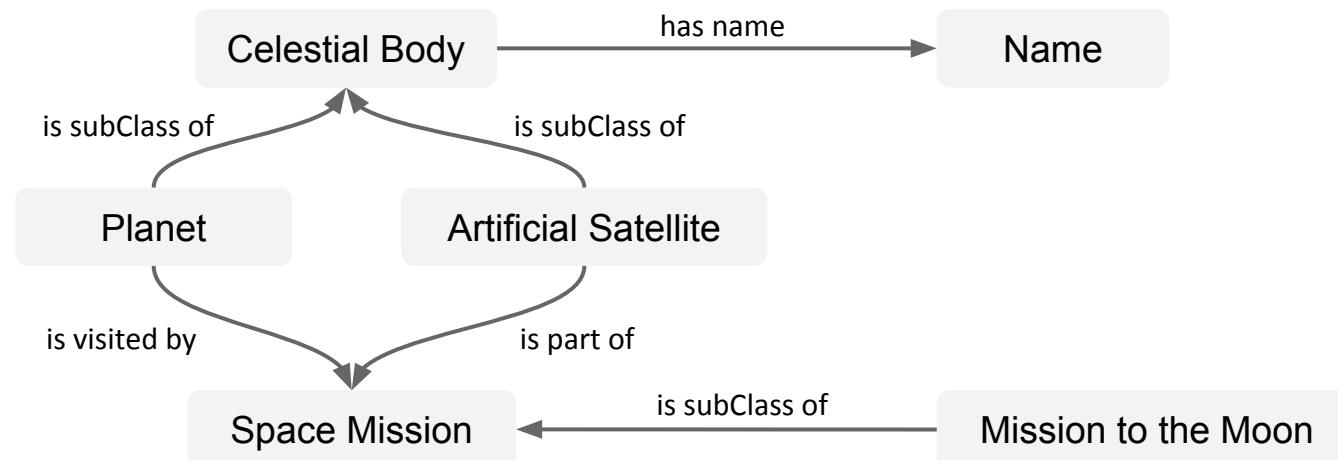
*Address*

- *given name <string>*
- *family name <string>*
- *street <string>*
- *ZIP code <int>*
- *city <string>*
- ...

semi-formal description

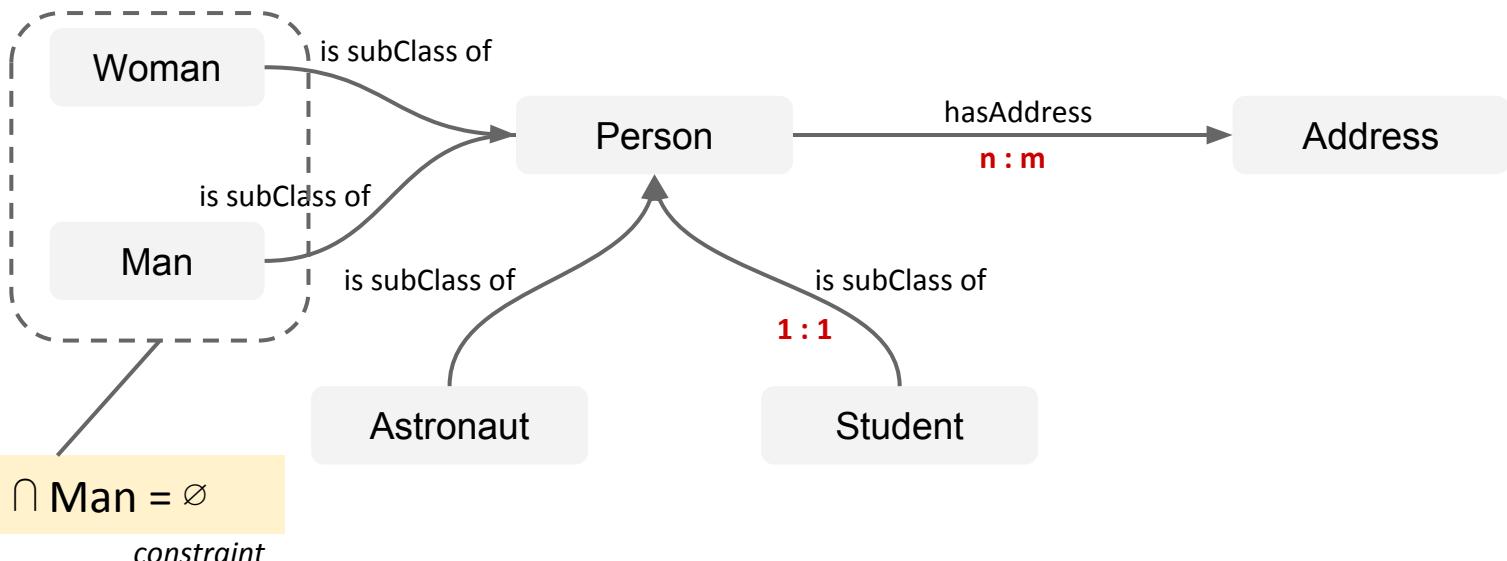
# How to represent Ontologies?

- Classes can be **related** to other classes
- **Relations** are special attributes, whose values are objects of (other) classes



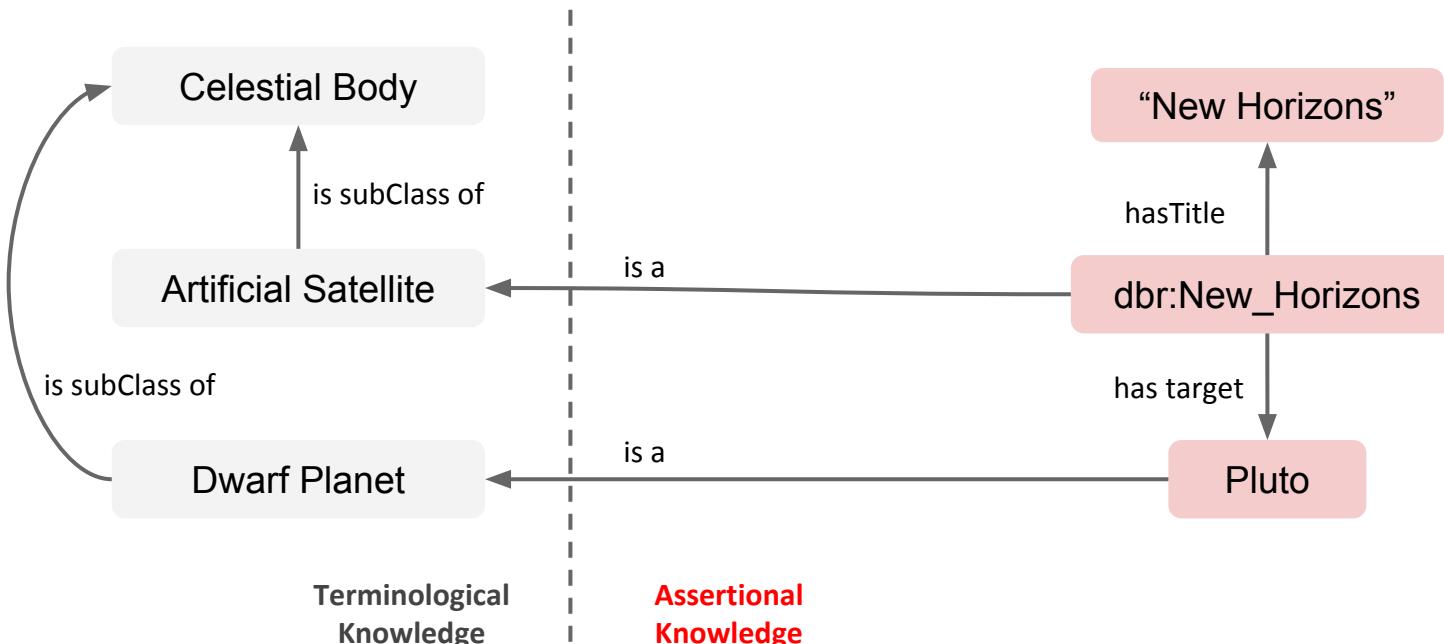
# How to represent Ontologies?

- For Relations and Attributes **Rules (Constraints)** can be defined that determine allowed/valid values



# How to represent Ontologies?

- Instances describe individuals of an ontology



# Information Service Engineering

## Lecture 6: Linked Data Engineering - 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 name Things - URIs
- 3.5 Resource Description Framework (RDF) as simple Data Model
- 3.6 Creating new Models with RDFS
- 3.7 Querying RDF(S) with SPARQL
- 3.8 More Expressivity with Web Ontology Language (OWL)
- 3.9 Wikipedia, DBpedia, and Wikidata
- 3.10 Linked Data Programming

# Third Generation: The Web of Data

Data Centered Processing

- The **Web of Data** is an upgrade of the Web of Documents
- It's the Web as a huge decentralised database (knowledge base) of **machine-accessible data**

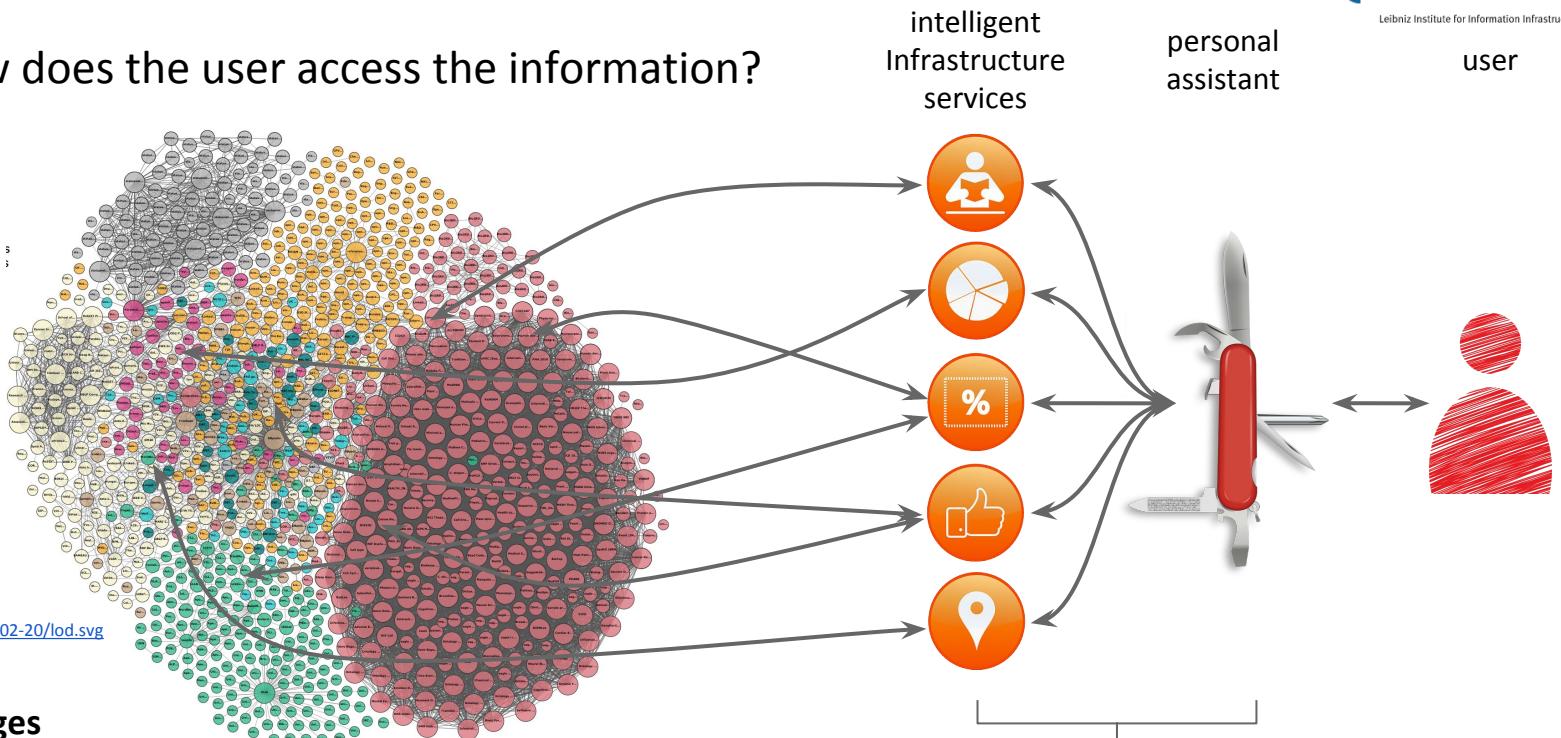
„The web of **human-readable document** is being merged with a web of **machine understandable data**. The potential of the mixture of humans and machines working together and communication through the web could be immense.“

*Tim Berners-Lee, [The World Wide Web: A very short personal history](#), May 1998*

# Third Generation: The Web of Data

## Data Centered Processing

- How does the user access the information?



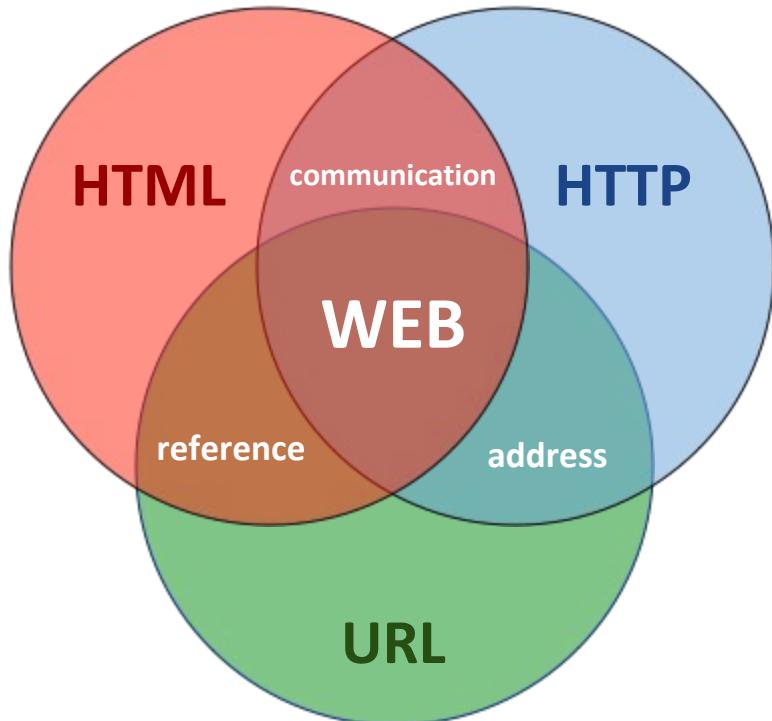
<http://lod-cloud.net/versions/2017-02-20/lod.svg>

### Advantages

- Information can be automatically selected, aggregated, remixed and published according to personal preferences

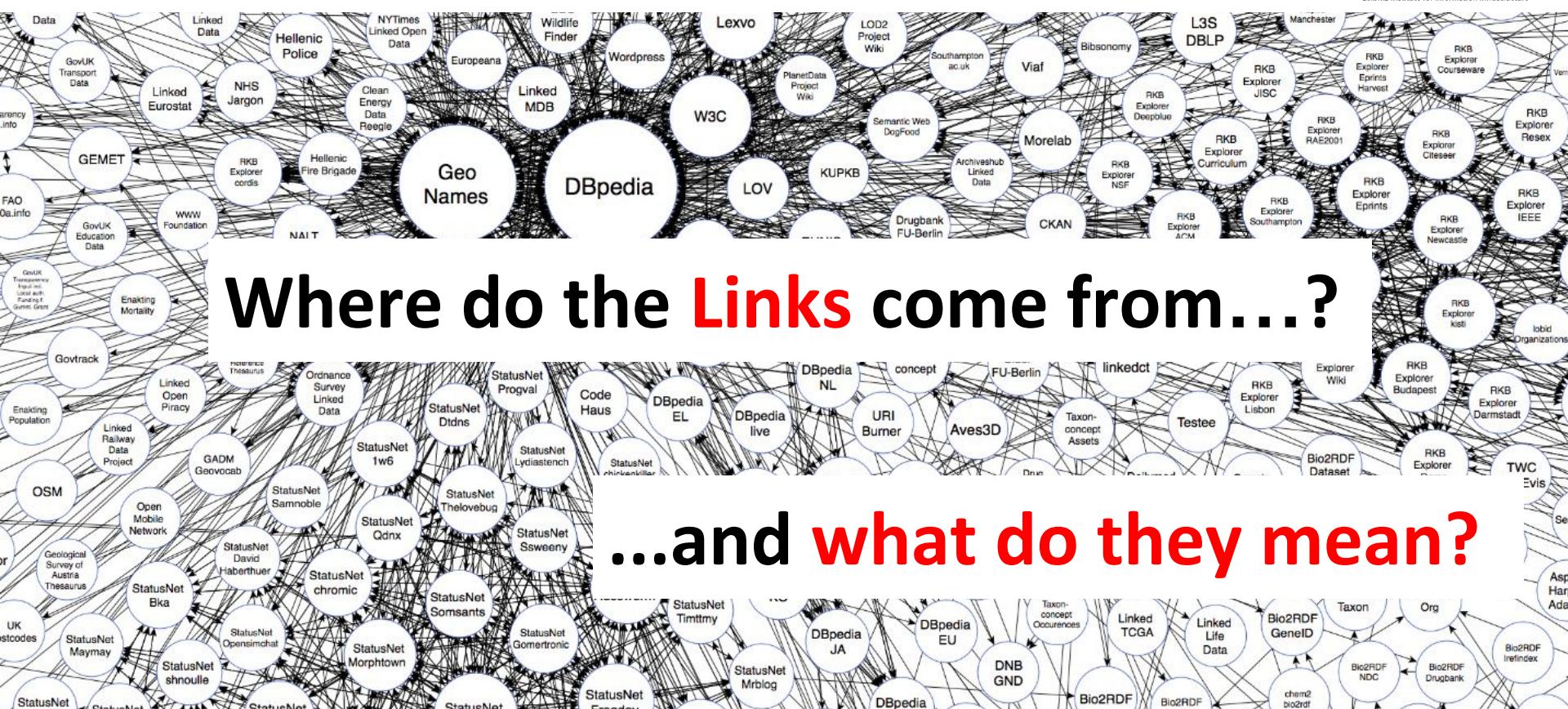
Information  
aggregation & filtering

# The Basic Architecture of the Web



1. Identification (**URI**) & address (**URL**)  
e.g. `http://kit.edu`
  
2. Communication / protocol (**HTTP**)  
`GET /index HTTP/2`  
`Host: kit.edu`
  
3. Representation language (**HTML**)  
`Rima studies at`  
`<a href="http://kit.edu">KIT</a>`

# Linked Data and the Web of Data



# The Semantic Web and the Web of Data

- The meaning of information (Semantics) is made explicit by **formal (structured) and standardized knowledge representations (Ontologies)**.
- Thus it will be possible,
  - to process the meaning of information automatically
  - to relate and integrate heterogeneous data
  - to deduce implicit (not evident) information from existing (evident) information in an automated way
- The Semantic Web is kind of a **global database** that contains a **universal network of semantic propositions**.

# The Semantic Web and the Web of Data



*„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“*

Tim Berners-Lee, James Hendler, Ora Lassila: [The Semantic Web](#), Scientific American, 284(5), pp. 34-43(2001)

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

Most apps use only a subset of the stack

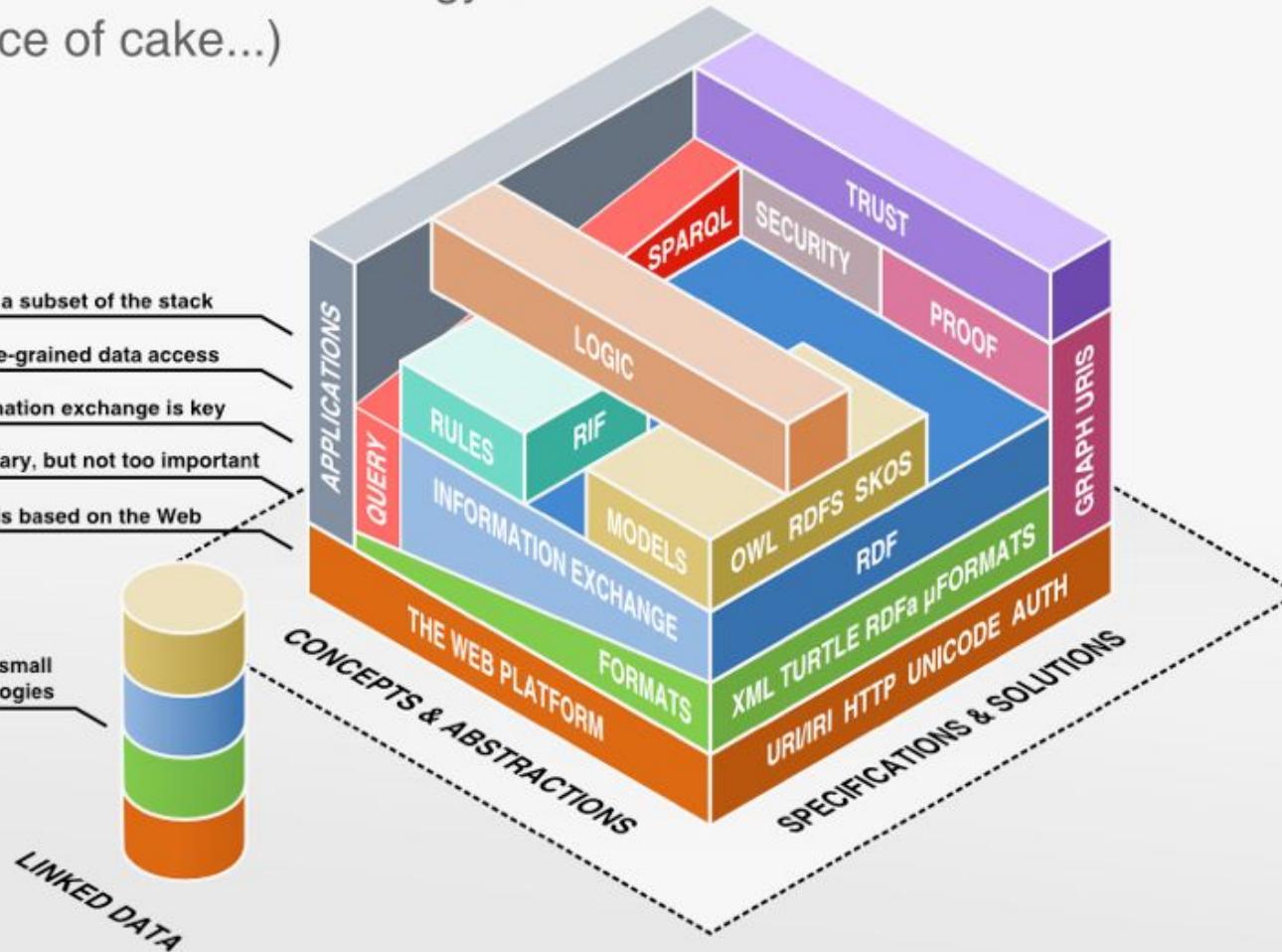
Querying allows fine-grained data access

Standardized information exchange is key

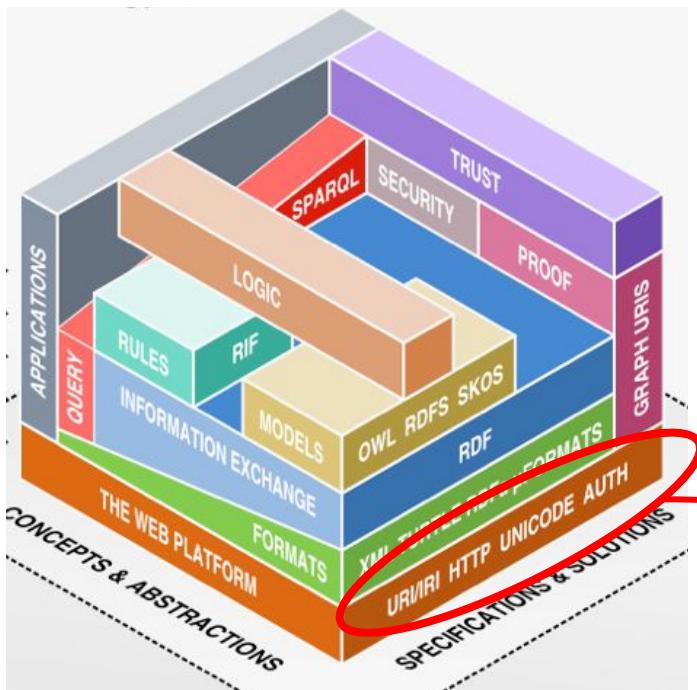
Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small selection of technologies



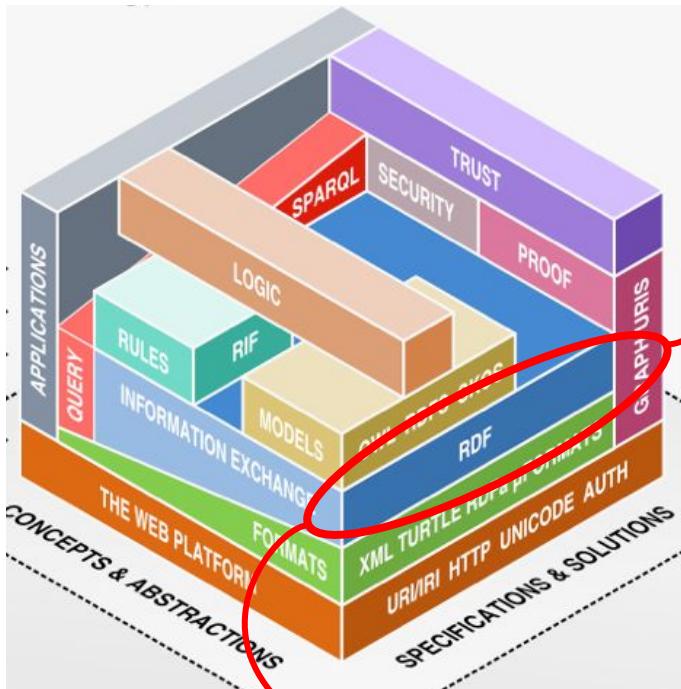
# The Semantic Web Technology Stack



Pluto

→ <http://dbpedia.org/resource/Pluto>

# The Semantic Web Technology Stack



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

```
:Pluto rdf:type dbo:Planet .
:Pluto foaf:name "Pluto"@en .
:Pluto dbo:discoverer :Clyde_Tombaugh .
:Pluto dbo:discovered "1930-02-18"^^xsd:date .
:Clyde_Tombaugh rdf:type dbo:Person .
:Clyde_Tombaugh dbo:birthdate "1906-02-04"^^xsd:date .
...
```

*RDF Resource Description Framework*



RDF Triple

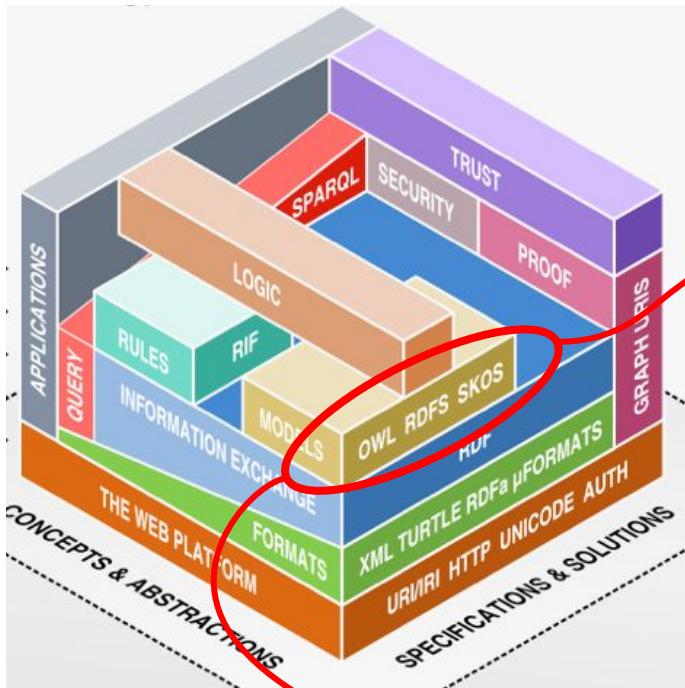
:Pluto  
RDF Subject

rdf:type  
RDF Property

dbo:Planet  
RDF Object



# The Semantic Web Technology Stack



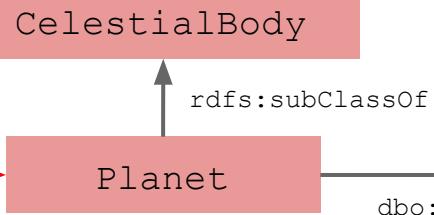
<http://dbpedia.org/ontology/Planet>

```

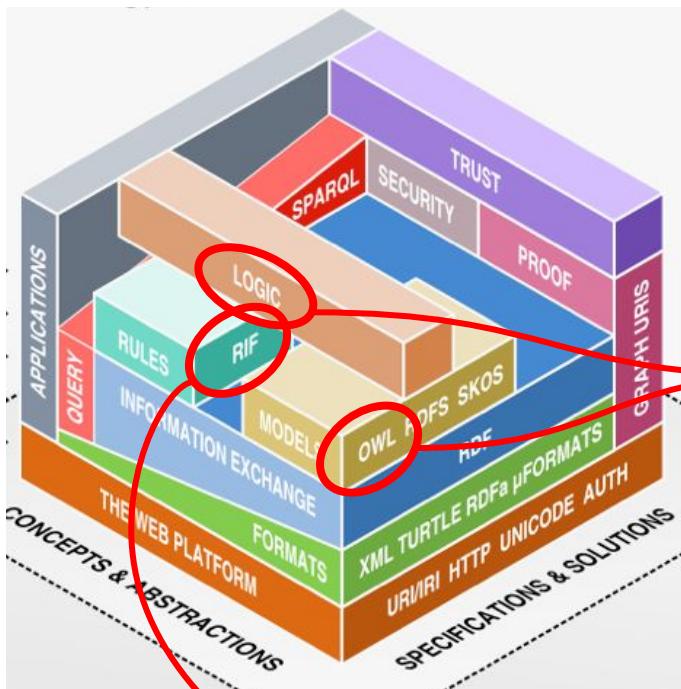
dbo:Planet rdf:type owl:Class .
dbo:Planet rdfs:subClassOf dbo:CelestialBody .
dbo:discovered rdf:type rdf:Property .
dbo:discovered rdfs:domain owl:Thing .
dbo:discovered rdfs:range xsd:date .
dbo:discoverer rdf:type rdf:Property .
dbo:discoverer rdfs:domain owl:Thing .
dbo:discoverer rdfs:range dbo:Person .
...
  
```



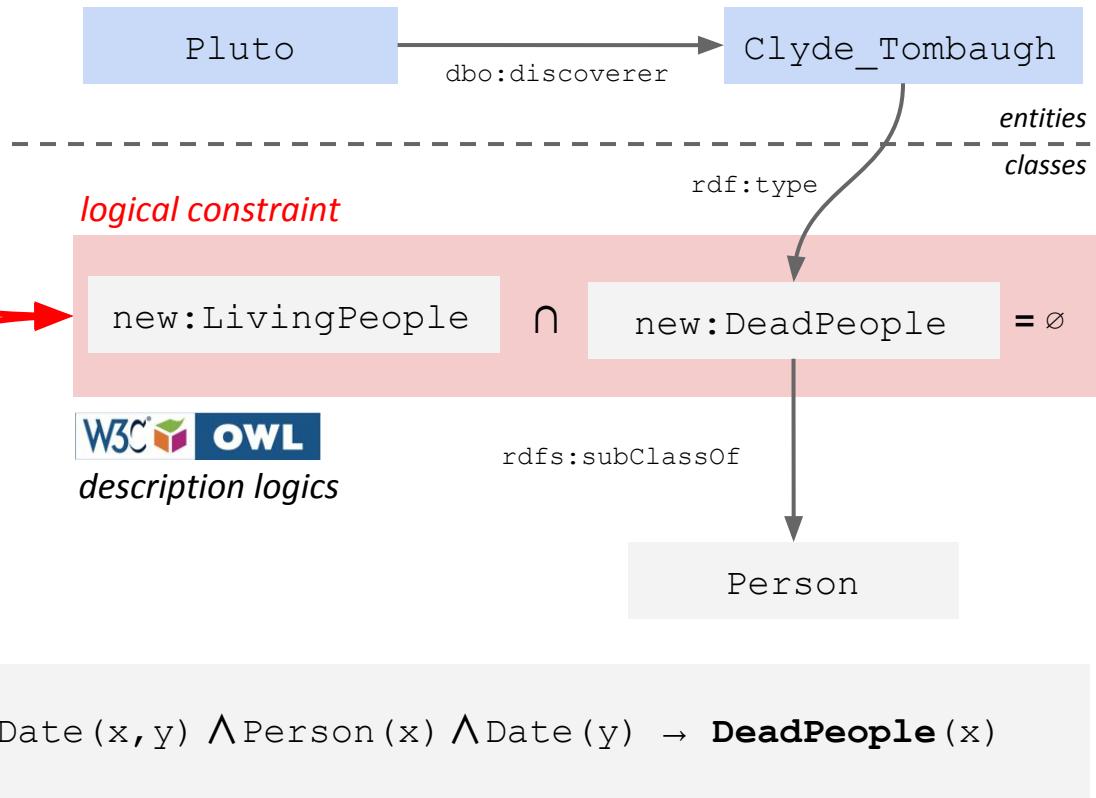
RDF Schema



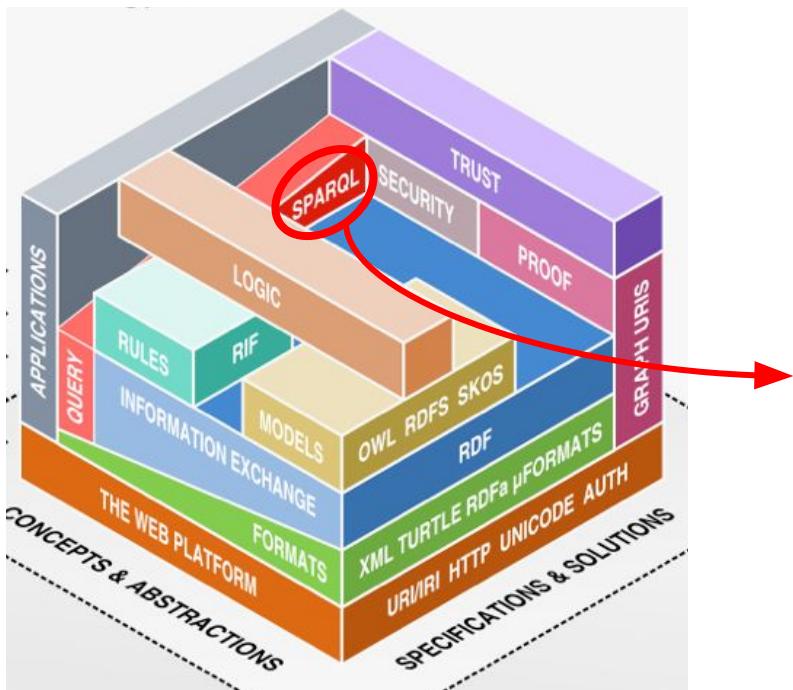
# The Semantic Web Technology Stack



+ logical rules       $\forall x. \exists y. \text{deathDate}(x, y) \wedge \text{Person}(x) \wedge \text{Date}(y) \rightarrow \text{DeadPeople}(x)$



# The Semantic Web Technology Stack



Look for all **Astronauts** who have been on a **Space Mission** which ended in a **Desaster**

```
PREFIX dcterm: <http://purl.org/dc/terms/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbc: <http://dbpedia.org/resource/Category:>

SELECT distinct ?astronaut ?mission
FROM <http://dbpedia.org/>
WHERE {
  ?astronaut rdf:type dbo:Astronaut;
              dbp:mission ?mission .
  ?mission    dcterm:subject dbc:Space_accidents_and_incidents .
}
```

[try SPARQL query at public DBpedia SPARQL endpoint](#)

# The Semantic Web Technology Stack

Look for all **Astronauts** who have been on a  
Space Mission which ended in a **Desaster**

astronaut	mission
<a href="http://dbpedia.org/resource/Neil_Armstrong">http://dbpedia.org/resource/Neil_Armstrong</a>	<a href="http://dbpedia.org/resource/Gemini_8">http://dbpedia.org/resource/Gemini_8</a>
<a href="http://dbpedia.org/resource/Edward_Higgins_White">http://dbpedia.org/resource/Edward_Higgins_White</a>	<a href="http://dbpedia.org/resource/Apollo_1">http://dbpedia.org/resource/Apollo_1</a>
<a href="http://dbpedia.org/resource/Gus_Grissom">http://dbpedia.org/resource/Gus_Grissom</a>	<a href="http://dbpedia.org/resource/Apollo_1">http://dbpedia.org/resource/Apollo_1</a>
<a href="http://dbpedia.org/resource/Roger_B._Chaffee">http://dbpedia.org/resource/Roger_B._Chaffee</a>	<a href="http://dbpedia.org/resource/Apollo_1">http://dbpedia.org/resource/Apollo_1</a>
<a href="http://dbpedia.org/resource/Gennadi_Strelkov">http://dbpedia.org/resource/Gennadi_Strelkov</a>	<a href="http://dbpedia.org/resource/Soyuz_7K-ST_No._16L">http://dbpedia.org/resource/Soyuz_7K-ST_No._16L</a>
<a href="http://dbpedia.org/resource/Kalpana_Chawla">http://dbpedia.org/resource/Kalpana_Chawla</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/William_C._McCool">http://dbpedia.org/resource/William_C._McCool</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Fred_Haise">http://dbpedia.org/resource/Fred_Haise</a>	<a href="http://dbpedia.org/resource/Apollo_13">http://dbpedia.org/resource/Apollo_13</a>
<a href="http://dbpedia.org/resource/Jim_Lovell">http://dbpedia.org/resource/Jim_Lovell</a>	<a href="http://dbpedia.org/resource/Apollo_13">http://dbpedia.org/resource/Apollo_13</a>
<a href="http://dbpedia.org/resource/Judith_Resnik">http://dbpedia.org/resource/Judith_Resnik</a>	<a href="http://dbpedia.org/resource/STS-51-L">http://dbpedia.org/resource/STS-51-L</a>
<a href="http://dbpedia.org/resource/David_Scott">http://dbpedia.org/resource/David_Scott</a>	<a href="http://dbpedia.org/resource/Gemini_8">http://dbpedia.org/resource/Gemini_8</a>
<a href="http://dbpedia.org/resource/David_M._Brown">http://dbpedia.org/resource/David_M._Brown</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Dick_Scobee">http://dbpedia.org/resource/Dick_Scobee</a>	<a href="http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster">http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster</a>
<a href="http://dbpedia.org/resource/Ellison_Onizuka">http://dbpedia.org/resource/Ellison_Onizuka</a>	<a href="http://dbpedia.org/resource/STS-51-L">http://dbpedia.org/resource/STS-51-L</a>
<a href="http://dbpedia.org/resource/Gregory_Jarvis">http://dbpedia.org/resource/Gregory_Jarvis</a>	<a href="http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster">http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster</a>
<a href="http://dbpedia.org/resource/Ilan_Ramon">http://dbpedia.org/resource/Ilan_Ramon</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Michael_J._Smith_(astronaut)">http://dbpedia.org/resource/Michael_J._Smith_(astronaut)</a>	<a href="http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster">http://dbpedia.org/resource/Space_Shuttle_Challenger_disaster</a>
<a href="http://dbpedia.org/resource/Michael_P._Anderson">http://dbpedia.org/resource/Michael_P._Anderson</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Rick_Husband">http://dbpedia.org/resource/Rick_Husband</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Christa_McAuliffe">http://dbpedia.org/resource/Christa_McAuliffe</a>	<a href="http://dbpedia.org/resource/STS-51-L">http://dbpedia.org/resource/STS-51-L</a>
<a href="http://dbpedia.org/resource/Laurel_Clark">http://dbpedia.org/resource/Laurel_Clark</a>	<a href="http://dbpedia.org/resource/STS-107">http://dbpedia.org/resource/STS-107</a>
<a href="http://dbpedia.org/resource/Viktor_Patsayev">http://dbpedia.org/resource/Viktor_Patsayev</a>	<a href="http://dbpedia.org/resource/Soyuz_11">http://dbpedia.org/resource/Soyuz_11</a>
<a href="http://dbpedia.org/resource/Vladislav_Volkov">http://dbpedia.org/resource/Vladislav_Volkov</a>	<a href="http://dbpedia.org/resource/Soyuz_11">http://dbpedia.org/resource/Soyuz_11</a>

[try SPARQL query at public DBpedia SPARQL endpoint](#)

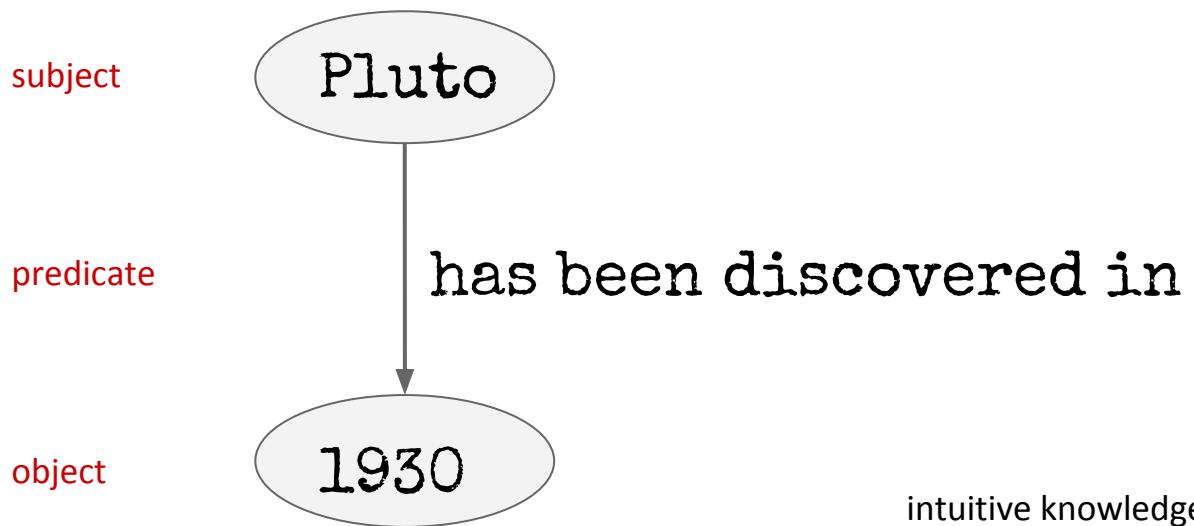
# Information Service Engineering

## Lecture 6: Linked Data Engineering - 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 name Things - URIs
- 3.5 Resource Description Framework (RDF) as simple Data Model**
- 3.6 Creating new Models with RDFS
- 3.7 Querying RDF(S) with SPARQL
- 3.8 More Expressivity with Web Ontology Language (OWL)
- 3.9 Wikipedia, DBpedia, and Wikidata
- 3.10 Linked Data Programming

# How to represent knowledge?

- How do I represent the following fact:  
*“Pluto has been discovered in 1930”* in an intuitive way?



## intuitive knowledge representation with a **directed graph**



# Resource Description Framework





# Resource Description Framework

- RDF Statements (RDF-Triple):

Subject <b>URI</b>	Property <b>URI</b>	Object / Value <b>URI / Literal</b>	RDF Building Blocks
-----------------------	------------------------	--	---------------------

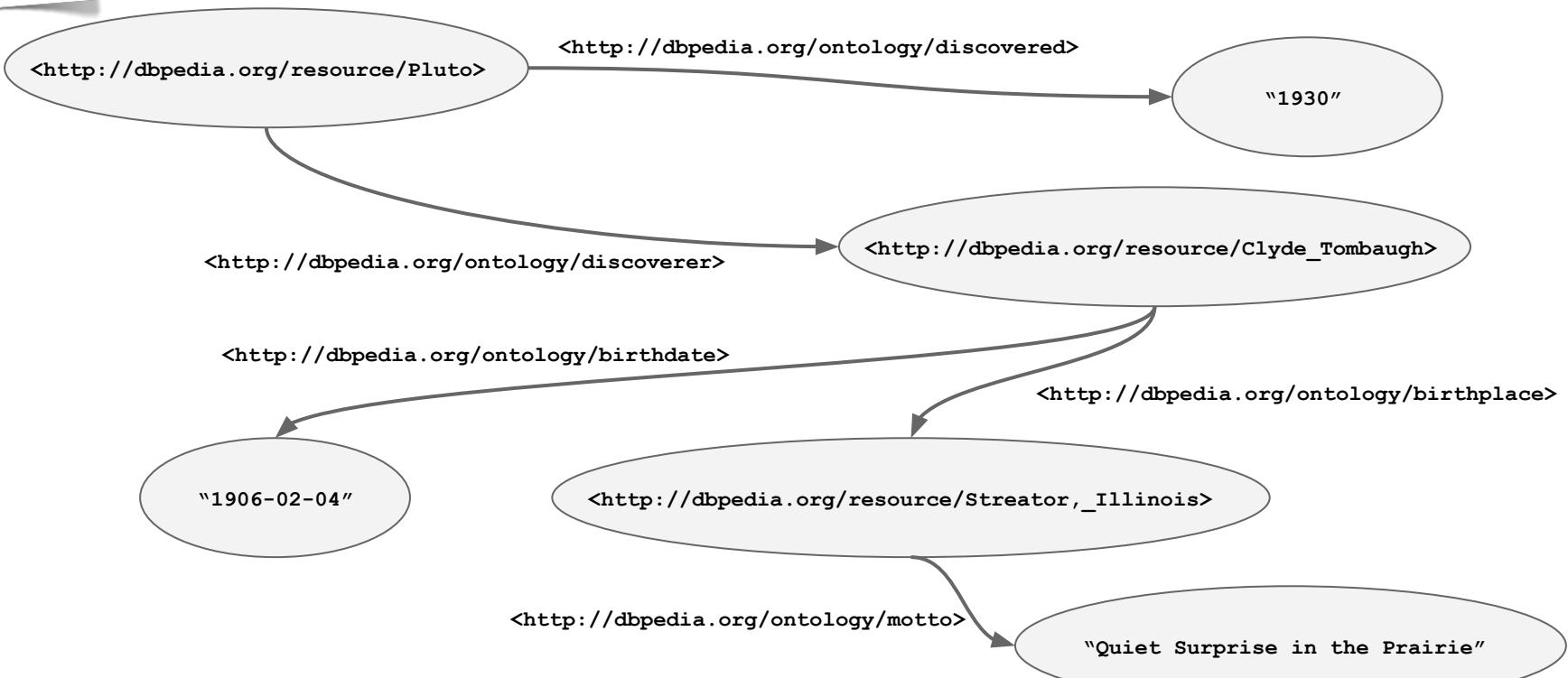
N-Triples Serialization

```
<http://dbpedia.org/resource/Pluto>    <http://dbpedia.org/ontology/discovered>    "1930" .
```





# Resource Description Framework

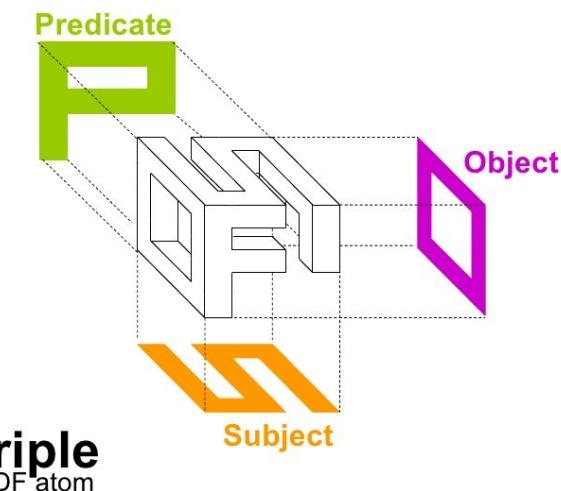




# Resource Description Framework

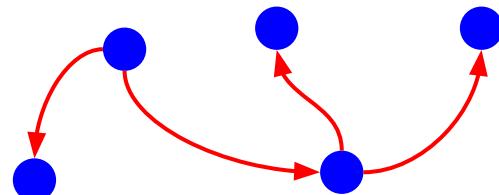
## RDF is a **Triple Model**

i.e. every piece of knowledge is broken down into  
( **subject** , **predicate** , **object** )



## RDF is a **Graph Model**

i.e. every piece of knowledge is interpreted as  
( **vertex** , **edge** , **vertex** )





# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .  
... ... ...  
  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .  
... ... ...  
  
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .  
... ... ...
```

Subject

Property

Object

RDF Triples



# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .  
...  
...  
...  
  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .  
...  
...  
...  
  
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .  
...  
...  
...
```

Individuals (Entities)



# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .  
... ... ...  
  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .  
... ... ...  
  
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .  
... ... ...
```

Classes



# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .  
... ... ...  
  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .  
... ... ...  
  
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .  
... ... ...
```

Literals



# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .
```

...

...

...

```
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .
```

...

...

...

```
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .
```

...

...

...

Properties



# Resource Description Framework

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer> <http://dbpedia.org/resource/Clyde_Tombaugh> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/CelestialBody> .  
<http://dbpedia.org/resource/Pluto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/place> .  
... ... ...  
  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthdate> "1906-02-04" .  
<http://dbpedia.org/resource/Clyde_Tombaugh> <http://dbpedia.org/ontology/birthplace> <http://dbpedia.org/resource/Streator,_Illinois> .  
... ... ...  
  
<http://dbpedia.org/resource/Streator,_Illinois> <http://dbpedia.org/ontology/motto> "Quiet Surprise in the Prairie" .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#lat> "41.120834"^^xsd:float .  
<http://dbpedia.org/resource/Streator,_Illinois> <http://www.w3.org/2003/01/geo/wgs84_pos#long> "-88.835281"^^xsd:float .  
... ... ...
```

Vocabularies / Ontologies

## Exercise 1: RDF Graph Format

“Fyodor Dostoevsky; 11 November 1821 – 9 February 1881, was a Russian novelist and short story writer.

His most acclaimed works include Crime and Punishment (1866) and The Idiot (1869).

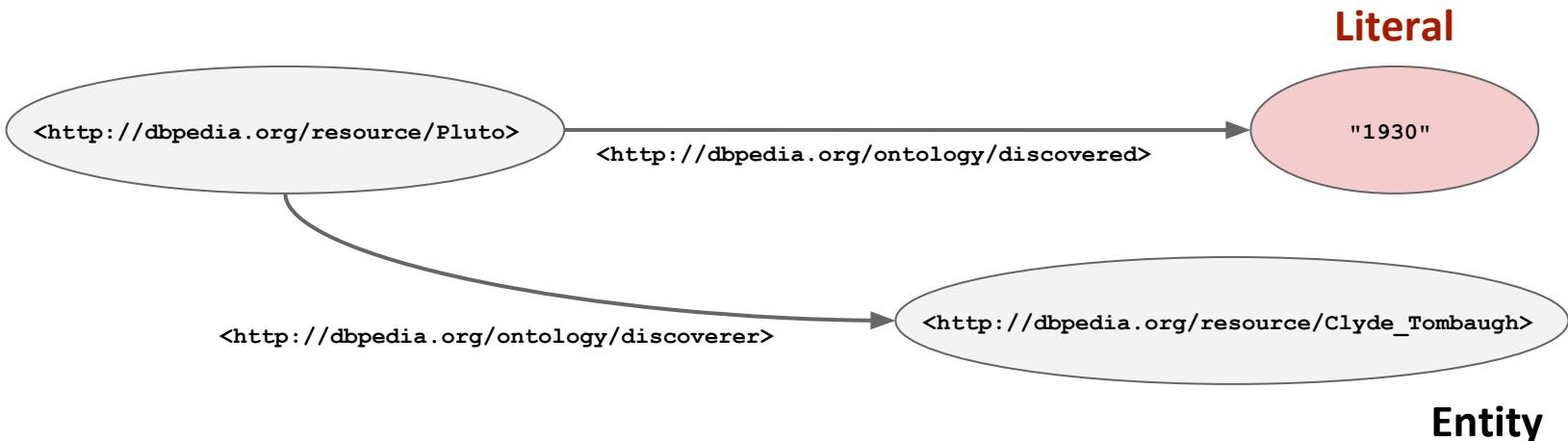
He was born in Moscow, Russia.”

**Exercise: Represent the above text in RDF in Graph format.**



# Resource Description Framework

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





# RDF Literals and Datatypes

- 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

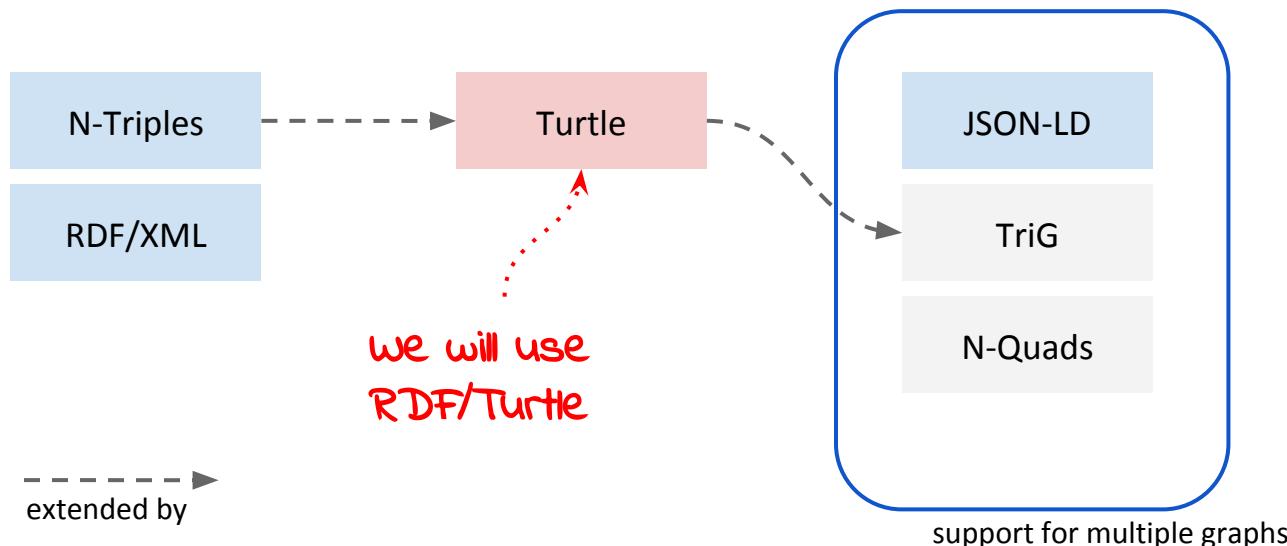
<http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/#xsd-datatypes>

<b>Core types</b>	xsd:string	Character strings
	xsd:boolean	true, false
<b>IEEE floating-point numbers</b>	xsd:decimal	Arbitrary-precision decimal numbers
	xsd:integer	Arbitrary-size integer numbers
<b>Time and date</b>	xsd:double	64-bit floating point numbers
	xsd:float	32-bit floating point numbers
<b>Recurring and partial dates</b>	xsd:date	Dates (yyyy-mm-dd) with or without time zone
	xsd:time	Times (hh:mm:ss.sss...) with or without time zone
<b>Limited-range integer numbers</b>	xsd:dateTime	Date and time with or without time zone
	xsd:dateTimeStamp	Date and time with required precision
<b>Miscellaneous XSD types</b>	xsd:gYear	Gregorian calendar year
	xsd:gMonth	Gregorian calendar month
<b>Encoded binary data</b>	xsd:gDay	Gregorian calendar day of the month
	xsd:gYearMonth	Gregorian calendar year and month
<b>Miscellaneous XSD types</b>	xsd:gMonthDay	Gregorian calendar month and day
	xsd:duration	Duration of time
<b>Miscellaneous XSD types</b>	xsd:yearMonthDuration	Duration of time (months and years)
	xsd:dayTimeDuration	Duration of time (days, hours, minutes, seconds)
<b>Miscellaneous XSD types</b>	xsd:byte	-128...+127 (8 bit)
	xsd:short	-32768...+32767 (16 bit)
<b>Miscellaneous XSD types</b>	xsd:int	-2147483648...+2147483647 (32 bit)
	xsd:long	-9223372036854775808...+9223372036854775807 (64 bit)
<b>Miscellaneous XSD types</b>	xsd:unsignedByte	0...255 (8 bit)
	xsd:unsignedShort	0...65535 (16 bit)
<b>Miscellaneous XSD types</b>	xsd:unsignedInt	0...4294967295 (32 bit)
	xsd:unsignedLong	0...18446744073709551615 (64 bit)
<b>Miscellaneous XSD types</b>	xsd:positiveInteger	Integer numbers >0
	xsd:nonNegativeInteger	Integer numbers ≥0
<b>Miscellaneous XSD types</b>	xsd:negativeInteger	Integer numbers <0
	xsd:nonPositiveInteger	Integer numbers ≤0
<b>Miscellaneous XSD types</b>	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary data
<b>Miscellaneous XSD types</b>	xsd:anyURI	Absolute or relative URLs and URIs
	xsd:language	Language tags per [BCP47]
<b>Miscellaneous XSD types</b>	xsd:normalizedString	Whitespace-normalized strings
	xsd:token	Tokenized strings
<b>Miscellaneous XSD types</b>	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
<b>Miscellaneous XSD types</b>	xsd:NCName	XML NCNames



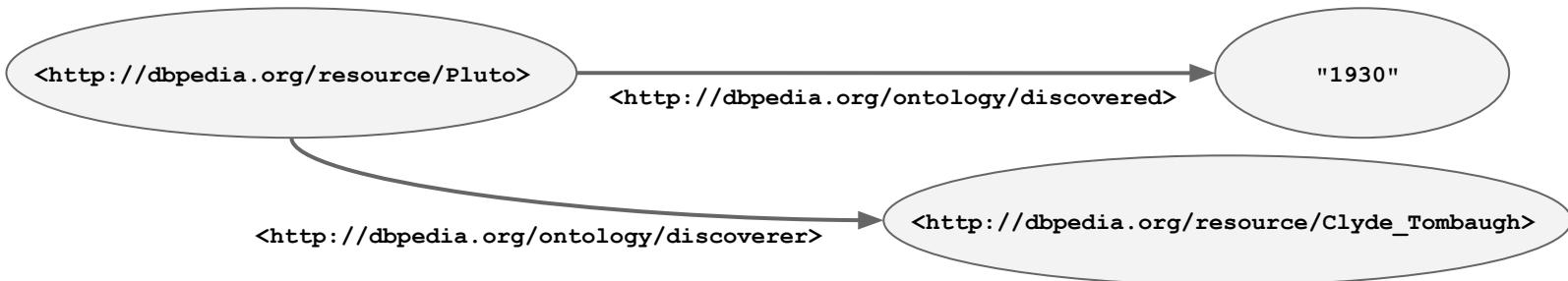
# RDF Serializations

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





# RDF Serializations

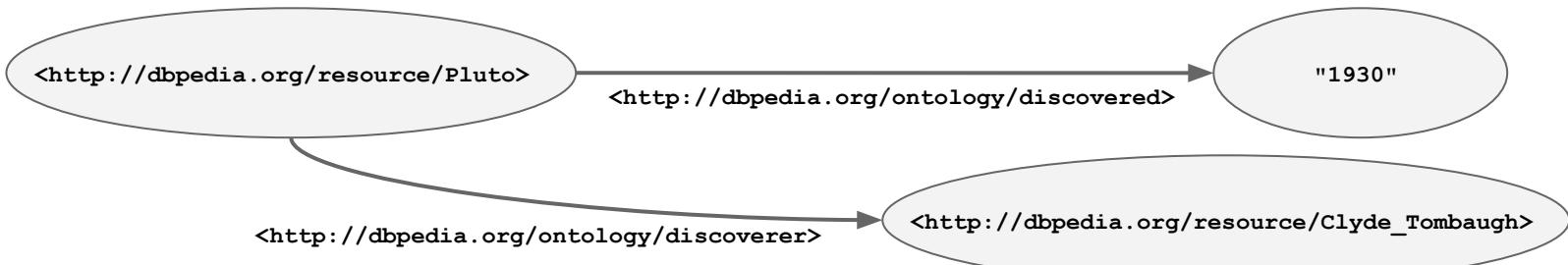


- **N-Triples Notation**

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

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontologydiscoverer>  
<http://dbpedia.org/resource/Clyde\_Tombaugh> .
```

# RDF Serializations



- **RDF/XML Notation**

S      P      O

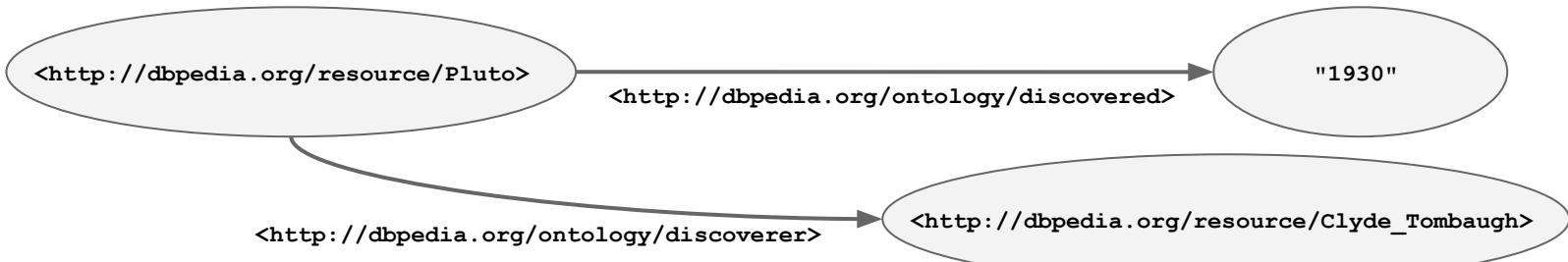
```

<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:ns0="http://dbpedia.org/ontology/">
  <rdf:Description rdf:about="http://dbpedia.org/resource/Pluto">
    <ns0:discovered>1930</ns0:discovered>
    <ns0:discoverer rdf:resource="http://dbpedia.org/resource/Clyde_Tombaugh"/>
  </rdf:Description>
</rdf:RDF>

```



# RDF Serializations



- JSON-LD Notation (RDF 1.1)

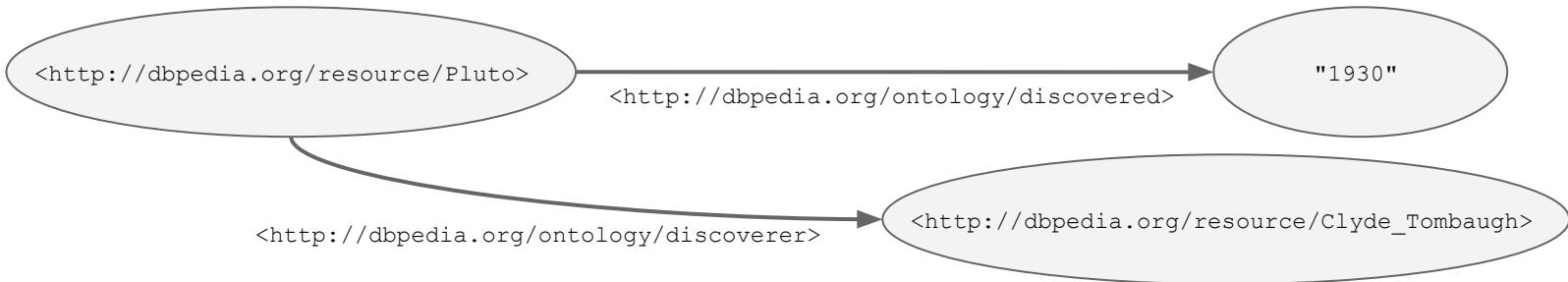
S      P      O

```
{ "@id" : "http://dbpedia.org/resource/Pluto" ,  
  "http://dbpedia.org/ontology/discovered" :  
    { "@value" : "1930" }  
  
,
```

```
  "http://dbpedia.org/ontologydiscoverer" :  
    { "@id" : "http://dbpedia.org/resource/Clyde_Tombaugh" }  
}
```

# RDF Serializations



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

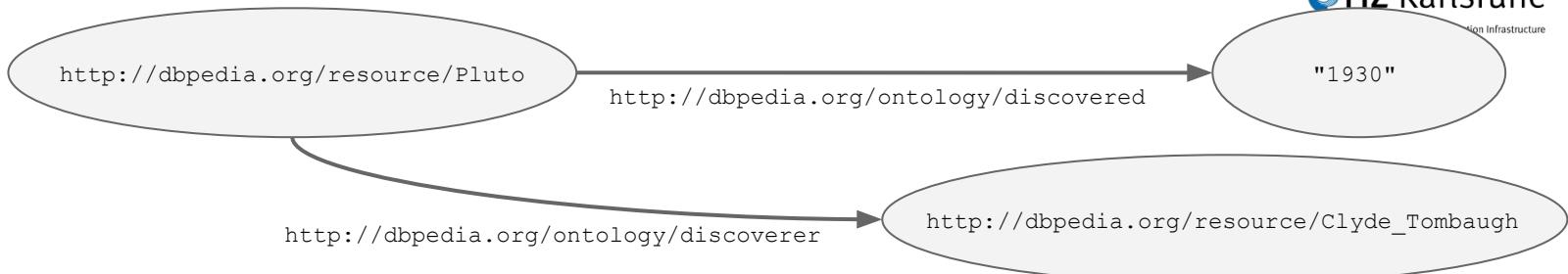
```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base  <http://dbpedia.org/resource/> .

<Pluto> dbo:discovered "1930" .

<Pluto> dbo:discoverer <Clyde_Tombaugh> .
```

RDF/Turtle allows  
shortcuts and  
abbreviations for  
readability

# RDF/Turtle



- starting with N-Triples

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .  

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

<http://dbpedia.org/resource/Clyde_Tombaugh> .
```

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

@base <http://dbpedia.org/resource/> .  

<Pluto> dbo:discovered "1930" .  

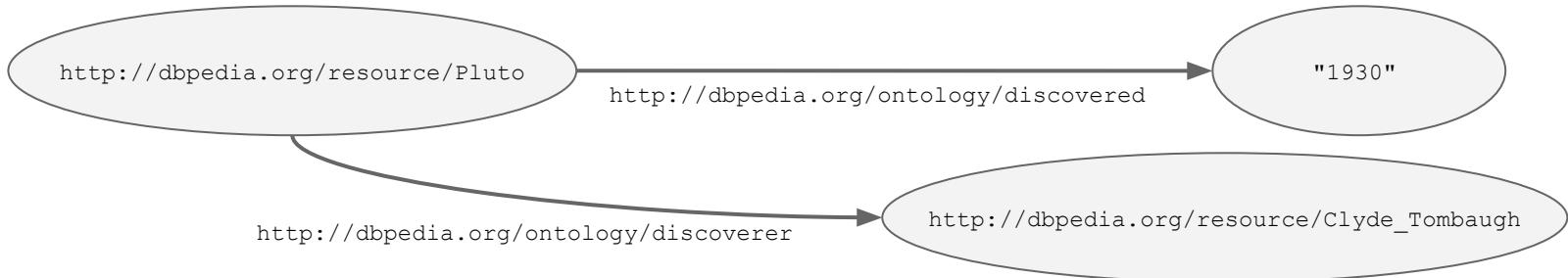
<Pluto> dbo:discoverer <Clyde_Tombaugh> .
```

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

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



# RDF/Turtle

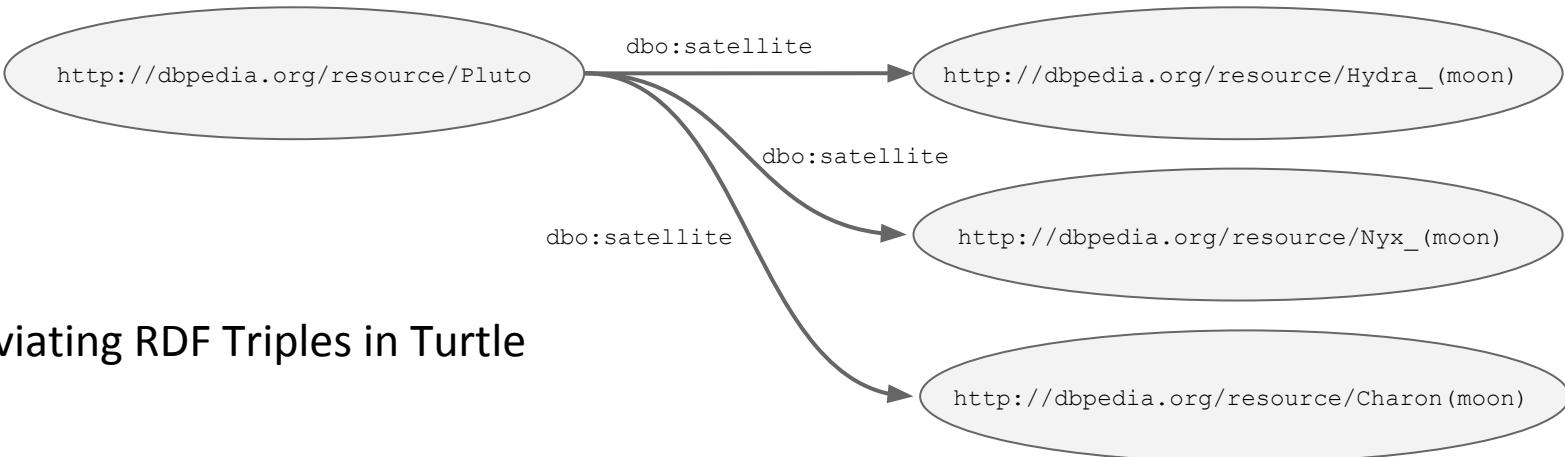


- Abbreviating RDF Triples in Turtle

```
@prefix dbo: <http://dbpedia.org/ontology/> .  
@base <http://dbpedia.org/resource/> .  
  
<Pluto> dbo:discovered "1930" ;  
      dbo:discoverer <Clyde_Tombaugh> .
```

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

# RDF/Turtle



- Abbreviating RDF Triples in Turtle

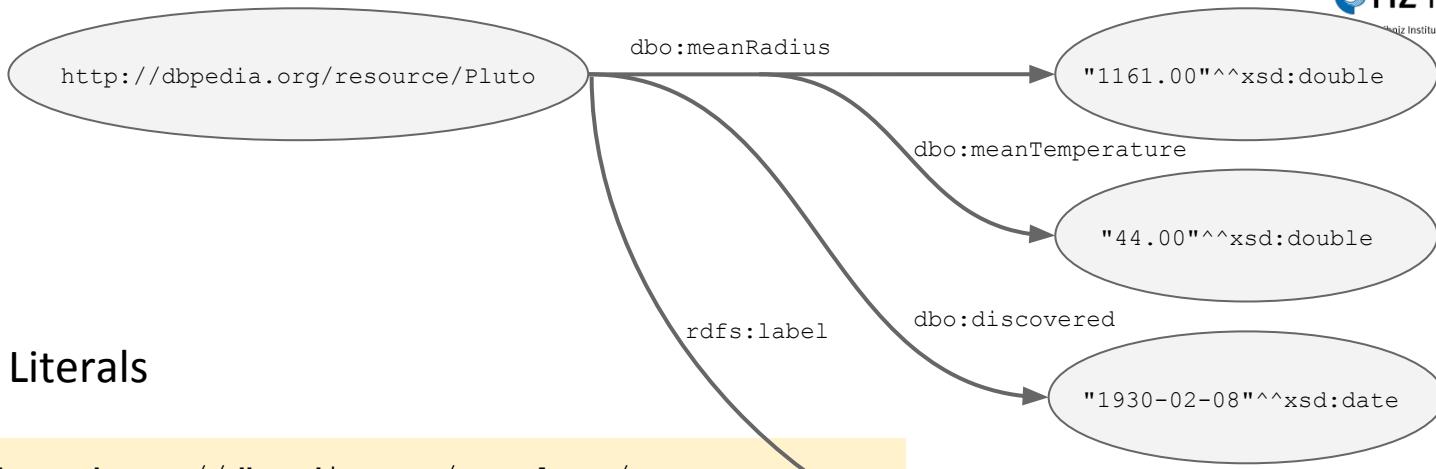
```

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

<Pluto> dbo:satellite <Hydra_(moon)> , 
          <Nyx_(moon)> , 
          <Charon_(moon)> .
  
```

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

# RDF/Turtle



- Typed Literals

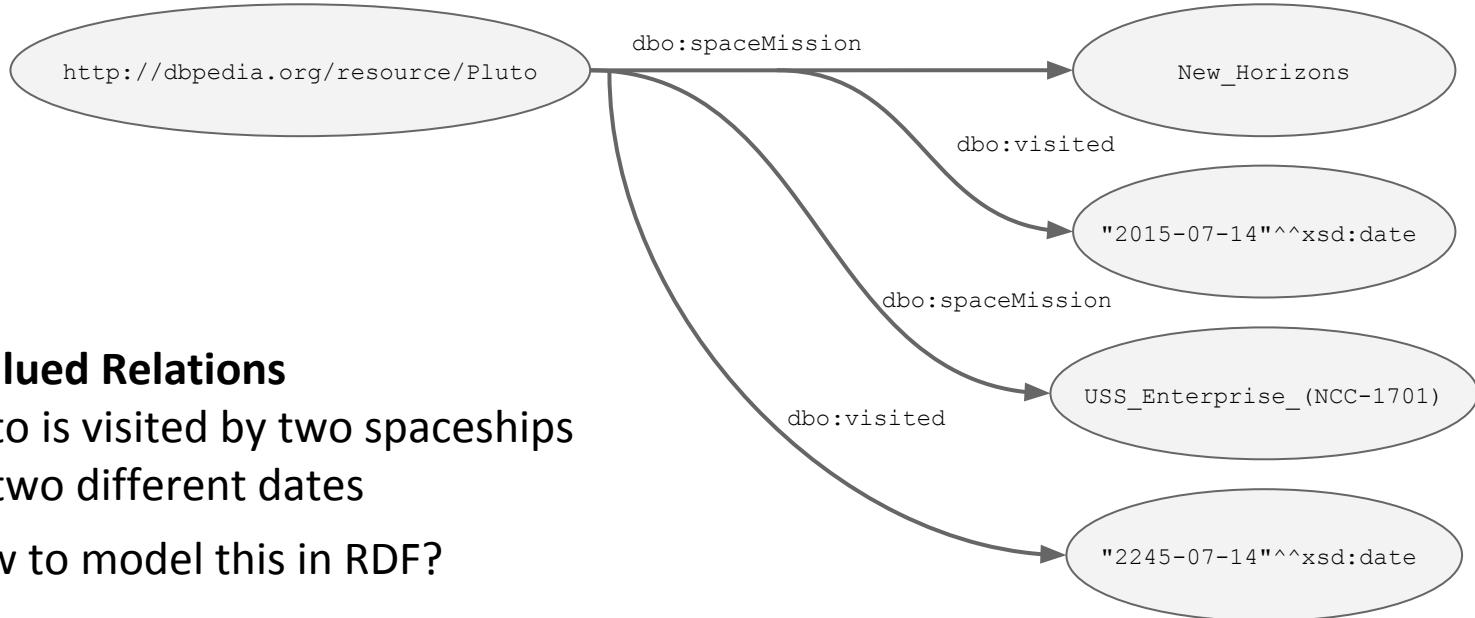
```

@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/> .

<Pluto> dbo:meanRadius "1161.00"^^xsd:double ;
      dbo:meanTemperature "44.00"^^xsd:double ;
      dbo:discovered "1930-02-18"^^xsd:date ;
      rdfs:label "Pluto"@en .
  
```



# RDF/Turtle

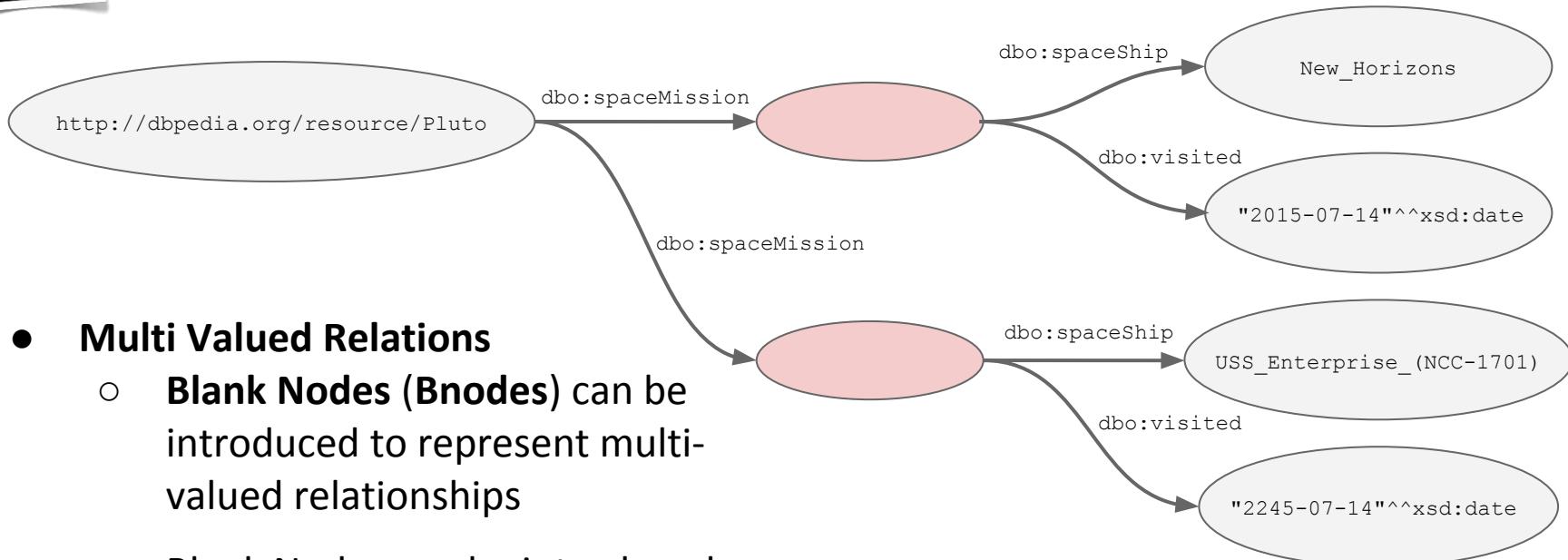


- **Multi Valued Relations**

- Pluto is visited by two spaceships on two different dates
- How to model this in RDF?
- **Problem:** unique association



# RDF/Turtle

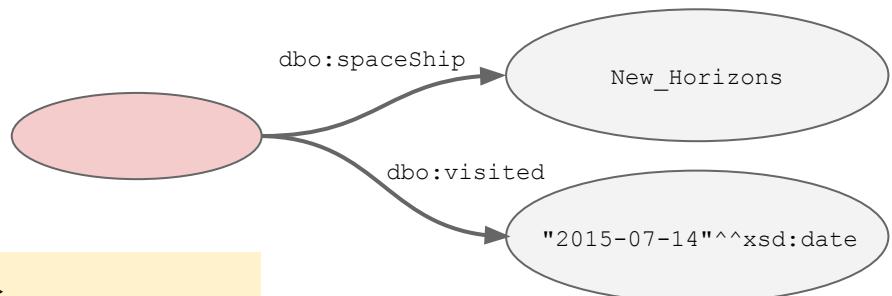


- **Multi Valued Relations**

- **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

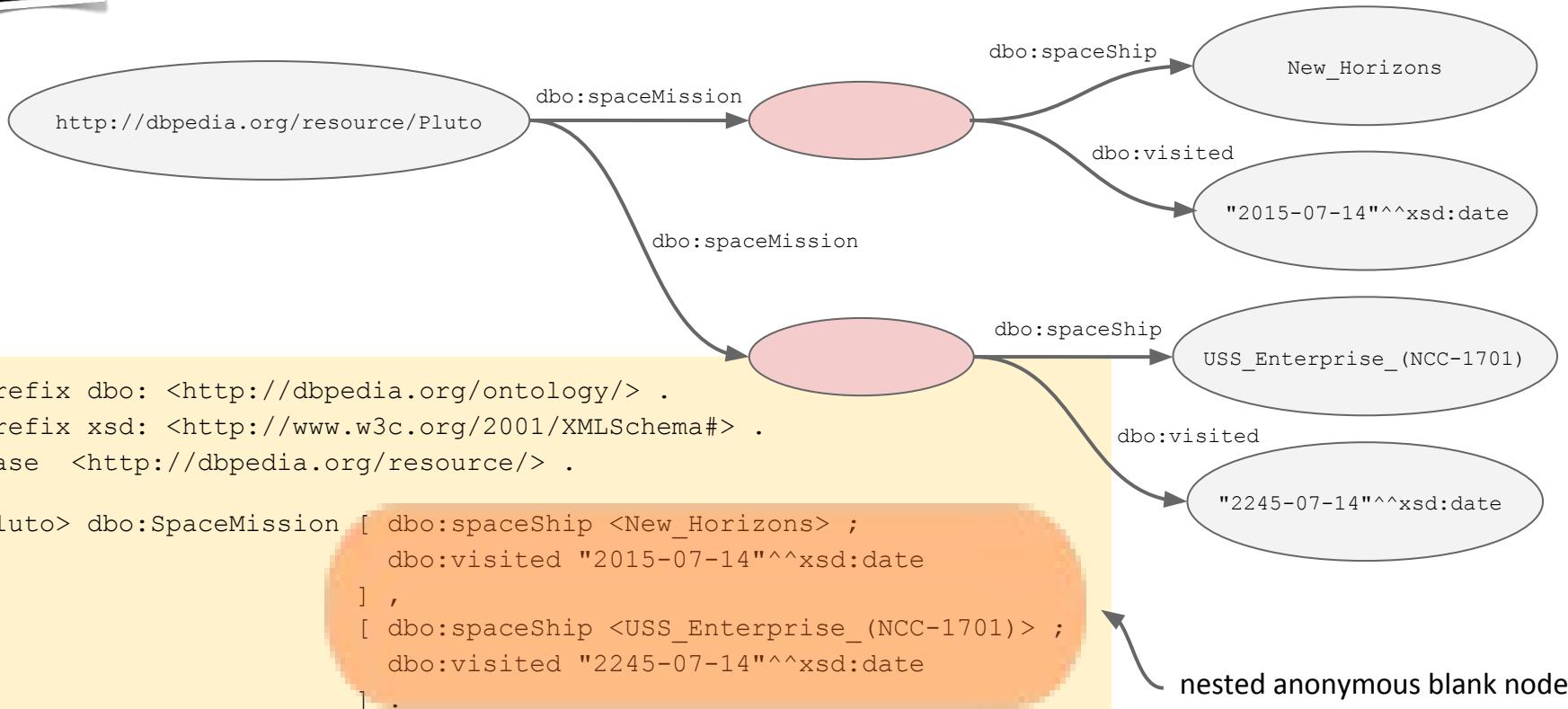


```
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .  
@base <http://dbpedia.org/resource/> .
```

```
[ ] dbo:spaceShip <New_Horizons> ;  
    dbo:visited "2015-07-14"^^xsd:date .
```

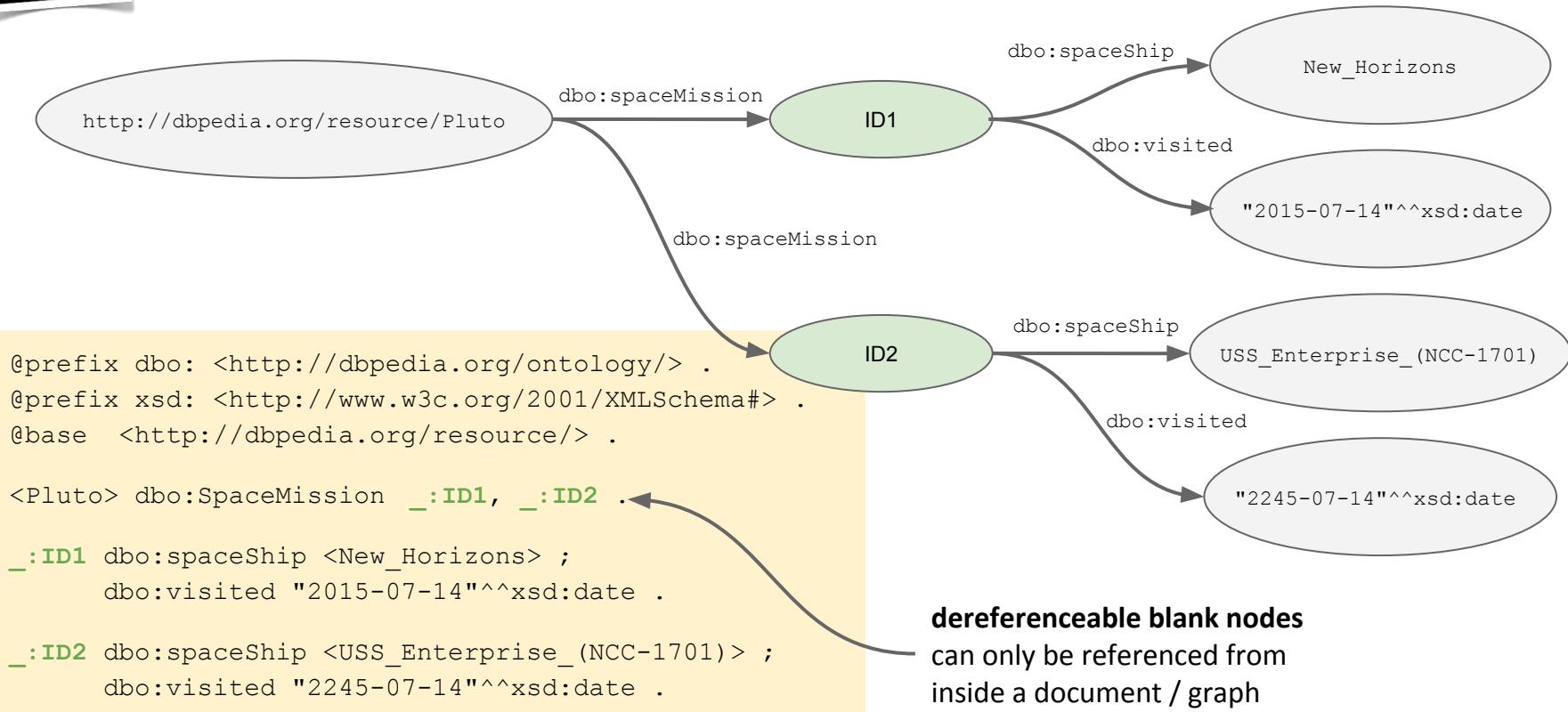
anonymous blank node as subject

# RDF/Turtle





# RDF/Turtle





# 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 Pluto has been discovered in 1930.”*

## Exercise 2: RDF Turtle Format

“Fyodor Dostoevsky; 11 November 1821 – 9 February 1881, was a Russian novelist and short story writer.

His most acclaimed works include Crime and Punishment (1866) and The Idiot (1869).

He was born in Moscow, Russia.”

**Exercise: Represent the above text in RDF in Turtle format.**

- 3.1 Knowledge Representations and Ontologies
- 3.2 Semantic Web and the Web of Data
- 3.3 Linked Data Principles
- 3.4 How to name Things - URIs
- 3.5 Resource Description Framework (RDF) as simple Data Model
- 3.6 Creating new Models with RDFS**
- 3.7 Querying RDF(S) with SPARQL
- 3.8 More Expressivity with Web Ontology Language (OWL)
- 3.9 Wikipedia, DBpedia, and Wikidata
- 3.10 Linked Data Programming

# What does it really mean?

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

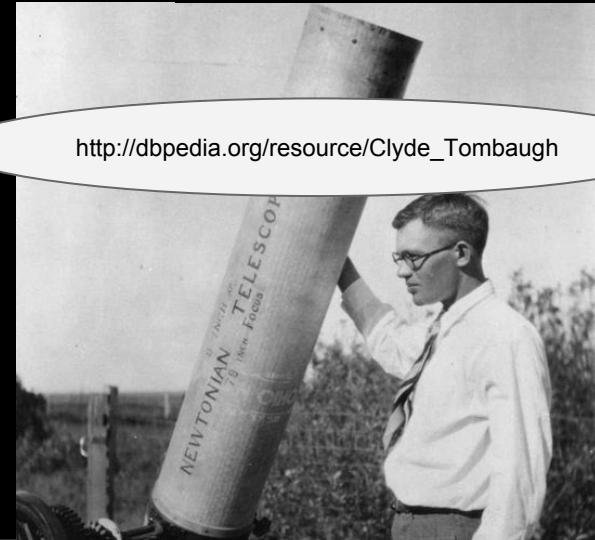
<http://dbpedia.org/property/satelliteOf>

[http://dbpedia.org/resource/Charon\\_\(moon\)](http://dbpedia.org/resource/Charon_(moon))

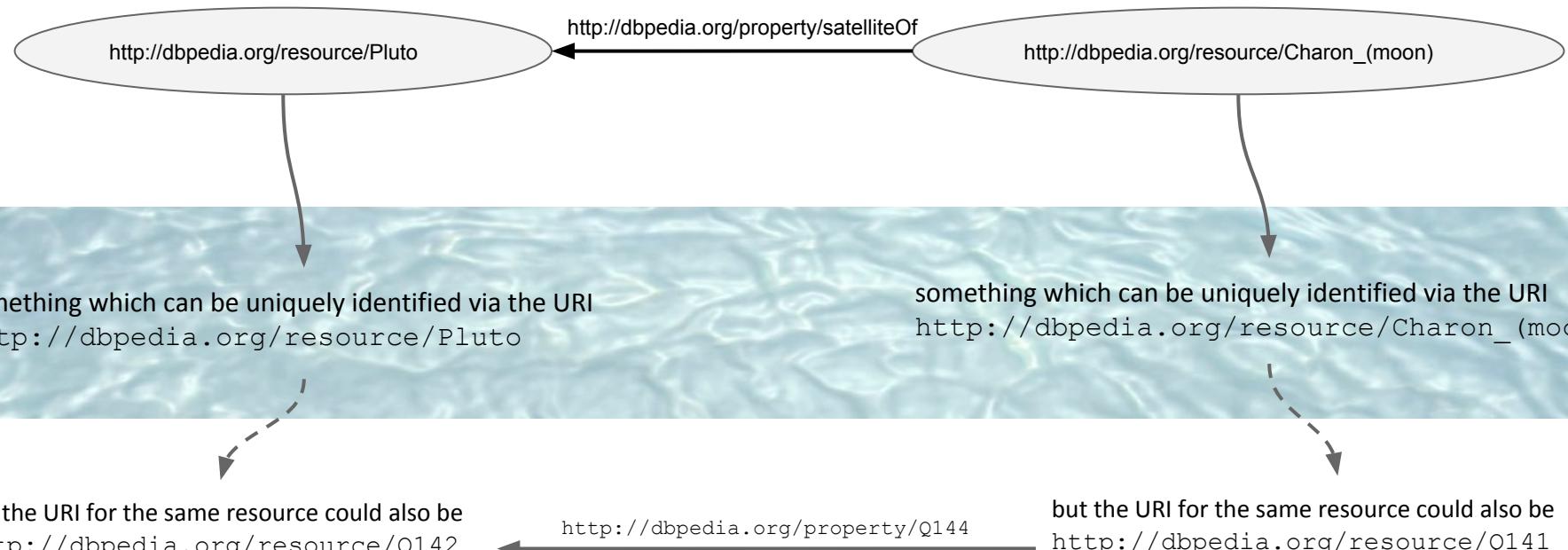
<http://dbpedia.org/ontology/discoverer>

[http://dbpedia.org/resource/Clyde\\_Tombaugh](http://dbpedia.org/resource/Clyde_Tombaugh)

Where does the intended meaning really come from?



# What does it really mean?



## We need more semantic expressivity...

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

**Most apps use only a subset of the stack**

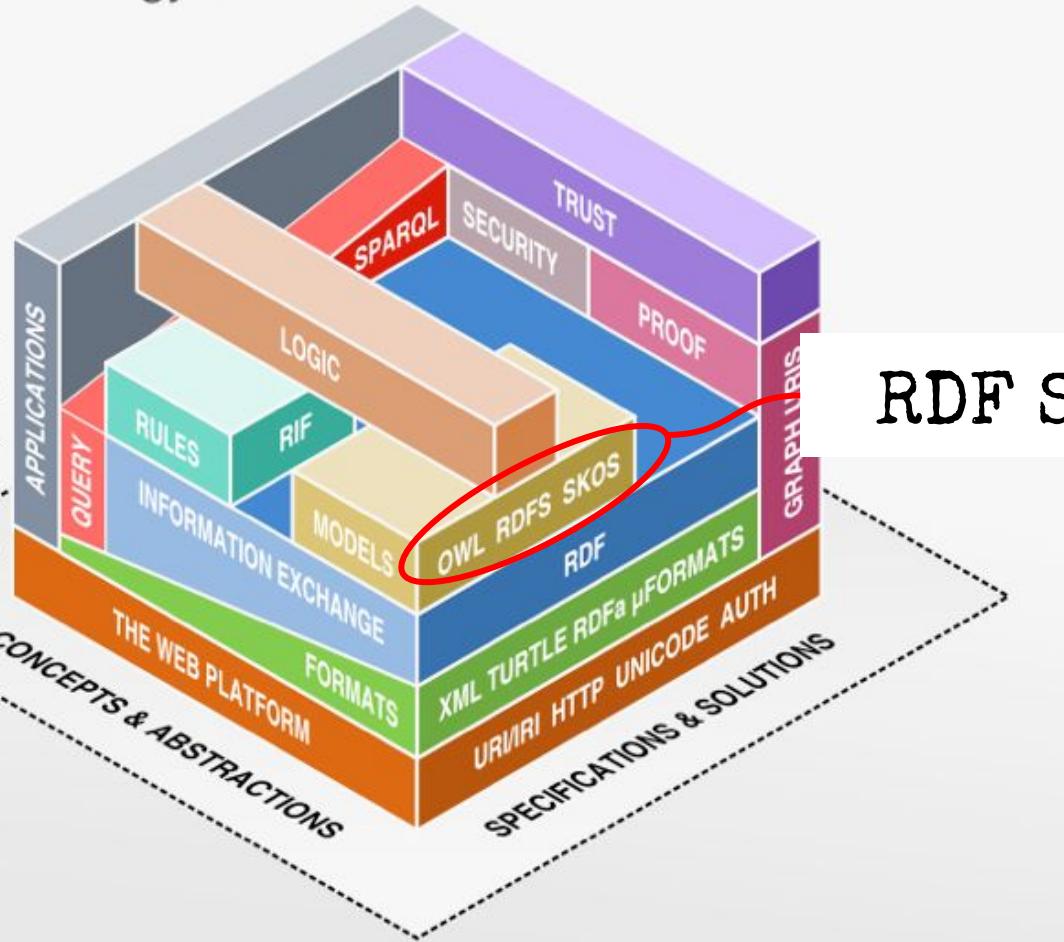
**Querying allows fine-grained data access**

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**



# RDF Schema



## 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:  
`:Planet rdf:type rdfs:Class .`  
`:Earth rdf:type :Planet .`

```
@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

```
:CelestialBody rdf:type rdfs:Class .  
:satelliteOf rdf:type rdf:Property .  
:satelliteOf rdfs:domain :CelestialBody .  
:satelliteOf rdfs:range :CelestialBody .
```



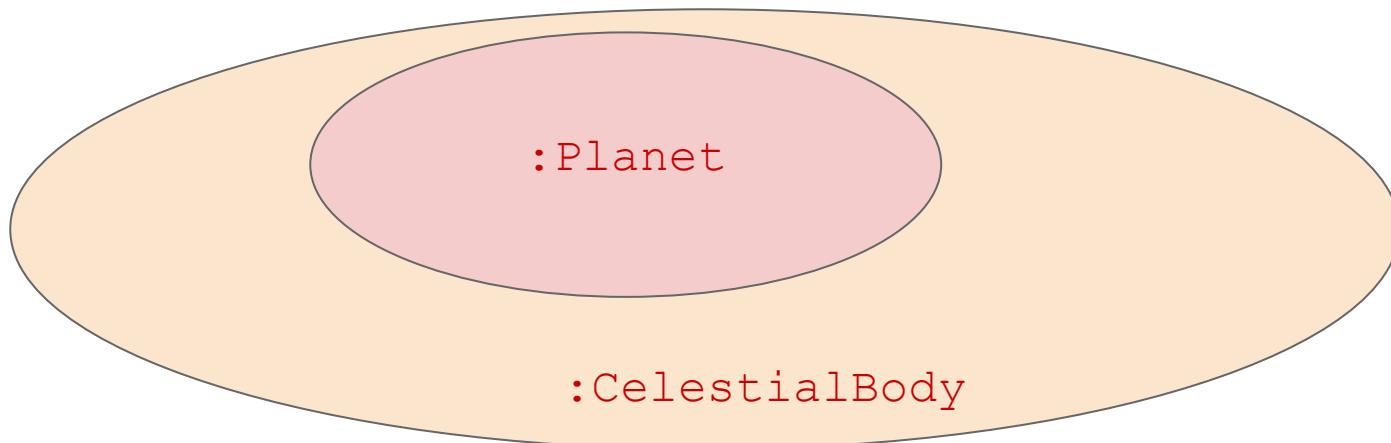
# RDF Schema

- 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:  
**:Planet rdfs:subClassOf :CelestialBody .**



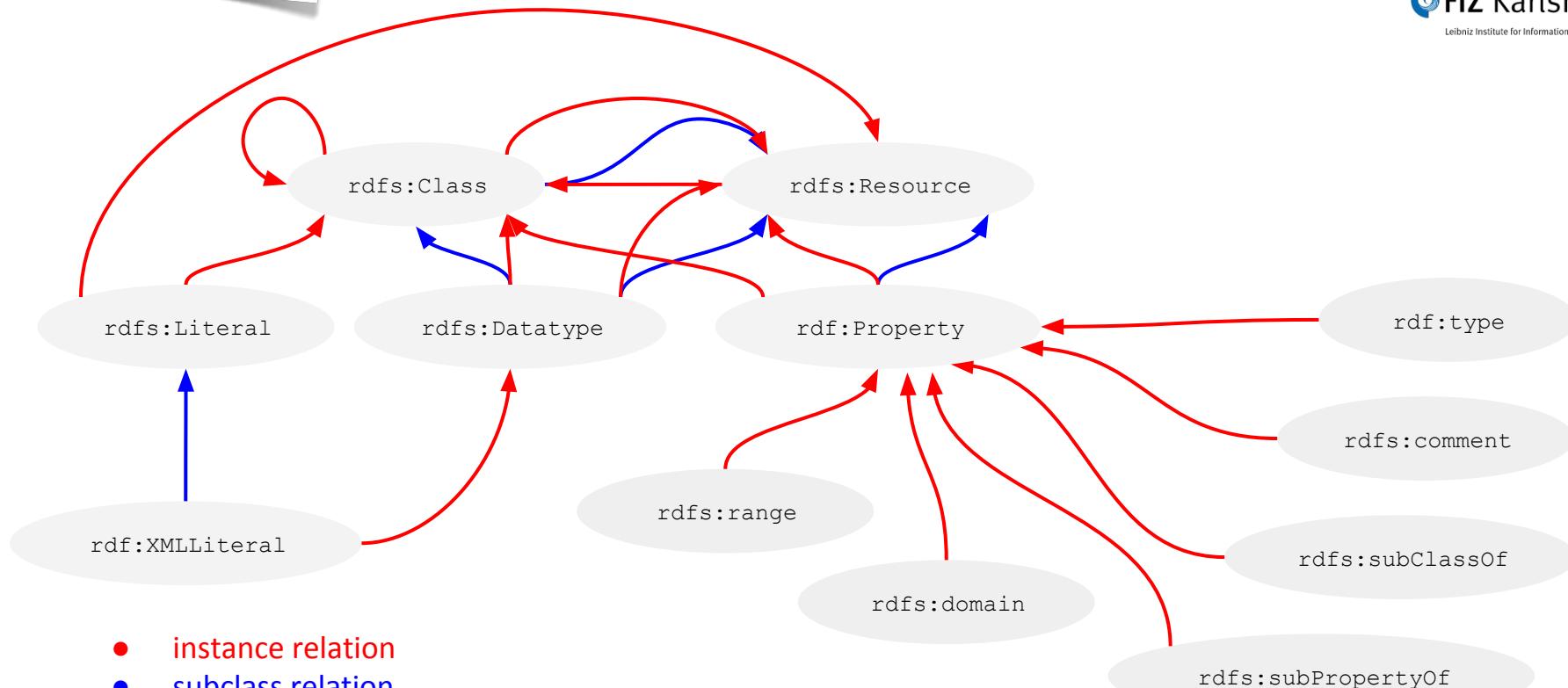


# RDF Schema

- Definition of **hierarchical relationships**:
  - **Subclasses** and **superclasses** via **rdfs:subClassOf**
  - Example:  
**:Planet rdfs:subClassOf :CelestialBody .**
  - **Subproperties** and **superproperties** via **subPropertyOf**
  - Example  
**:artificialSatelliteOf rdfs:subPropertyOf :satelliteOf .**



# RDFS Language Model



- **instance relation**
- **subclass relation**

[http://www.w3.org/TR/rdf-schema/#ch\\_classes](http://www.w3.org/TR/rdf-schema/#ch_classes)



## 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#> .
@prefix : <http://example.org/Space#> .
```

```
:Planet rdf:type rdfs:Class ;
        rdfs:subClassOf :CelestialBody .
:Satellite rdf:type rdfs:Class ;
        rdfs:subClassOf :CelestialBody .
:ArtificialSatellite rdf:type rdfs:Class ;
        rdfs:subClassOf :Satellite .
```

Class Definitions

```
:satelliteOf rdf:type rdf:Property ;
        rdfs:domain :CelestialBody .
        rdfs:range :CelestialBody .
```

Property Definitions

```
:Earth rdf:type :Planet .
:Moon rdf:type :Satellite ;
        :satelliteOf :Earth .
:Sputnik1 rdf:type :ArtificialSatellite ;
        :satelliteOf :Earth ;
        rdfs:label "Sputnik 1"@en ;
        rdfs:comment "the first artificial Earth satellite in 1957" .
```

Instance Definitions

## Exercise 3: RDF-Schema

Start from the Turtle triples of the previous example.

**Add the following classes:** Writer, Person, Location.

**Add the following properties:** locatedIn, birthDate, deathDate, authorOf, birthPlace

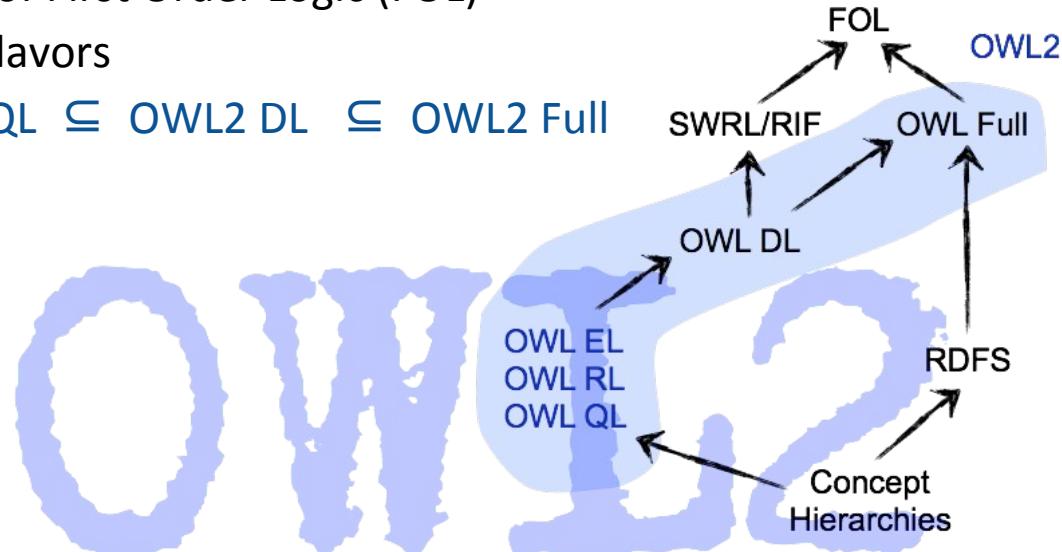
Add suitable **domain and range restrictions** for the properties.

Add some labels to the instances!

Finally: which new triples can a **reasoner** automatically deduce?

# Web Ontology Language OWL - OWL Flavors

- OWL is a semantic fragment of First Order Logic (FOL)
- OWL also exists in different flavors
  - $\text{OWL EL}, \text{OWL RL}, \text{OWL QL} \subseteq \text{OWL2 DL} \subseteq \text{OWL2 Full}$



# OWL2 is based on the Description Logic $SROIQ(\mathcal{D})$

## Class Expressions

- Class names A, B
- Conjunction  $C \sqcap D$
- Disjunction  $C \sqcup D$
- Negation  $\neg C$
- Exist. property restriction  $\exists R.C$
- Univ. property restriction  $\forall R.C$
- Self  $\exists S.Self$
- Greater-than  $\geq n S.C$
- Less-than  $\leq n S.C$
- Enumerated classes {a}

## Properties

- Property names R, S, T
- Simple properties S, T
- Inverse properties  $R^-$
- Universal property U

## Tbox (Class axioms)

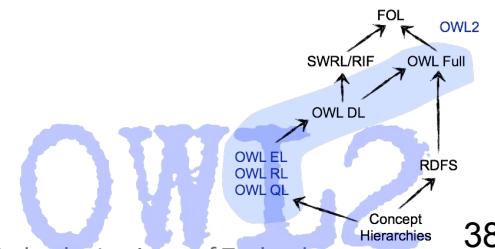
- Inclusion  $C \sqsubseteq D$
- Equivalence  $C \equiv D$

## Rbox (Property Axioms)

- Inclusion  $R_1 \sqsubseteq R_2$
- General Inclusion  $R^{(-)}_1 \circ R^{(-)}_2 \circ \dots \circ R^{(-)}_n \sqsubseteq R$
- Transitivity
- Symmetry
- Reflexivity
- Irreflexivity
- Disjunctiveness

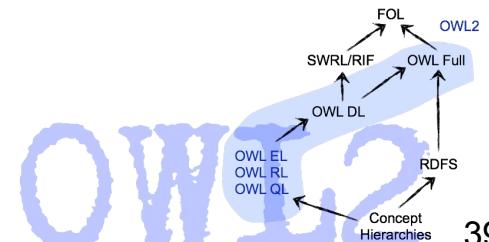
## Abox (Facts)

- Class membership  $C(a)$
- Property relation  $R(a, b)$
- Negated property relation  $\neg S(a, b)$
- Equality  $a = b$
- Inequality  $a \neq b$



# OWL Basic Building Blocks

- OWL namespace:  
`@prefix owl: <http://www.w3.org/2002/07/owl#>`
- There is a **Turtle Syntax** for OWL
- OWL axioms consist of the following three building blocks:
  - **Classes**
    - comparable with classes in RDFS
  - **Individuals**
    - comparable with class instances in RDFS
  - **Properties**
    - comparable with properties in RDFS

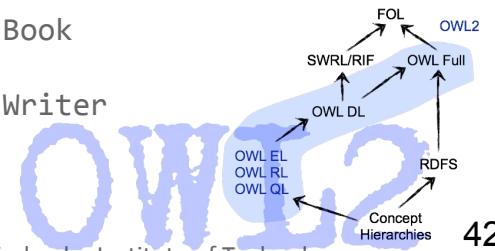


# OWL Object Properties

- there exist two property variants:
  - object properties
  - datatype properties
- **Object properties** have classes as range
 

```
:author a owl:ObjectProperty .
```
- Domain and Range of object properties
 

```
:author a owl:ObjectProperty ;
        rdfs:domain :Book ;
        rdfs:range :Writer .
```

 $\exists \text{author.} \top \sqsubseteq \text{Book}$ 
 $\top \sqsubseteq \forall \text{author.} \text{Writer}$ 


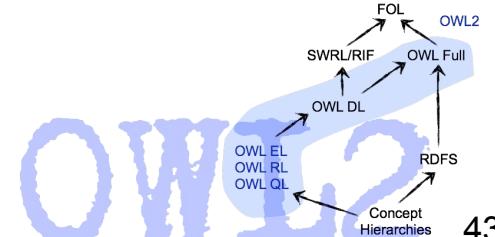
# OWL Datatype Properties

- **Datatype properties** have datatypes as range

```
:publicationYear a owl:DatatypeProperty .
```

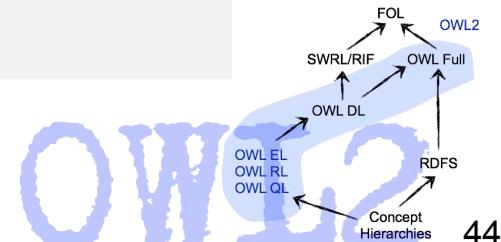
- Domain and Range of datatype properties

```
:publicationYear a owl:DatatypeProperty ;
    rdfs:domain :Book ;           ∃ publicationYear.⊤ ⊑ Book
    rdfs:range xsd:integer .     ⊤ ⊑ ∀ publicationYear.Integer
```



# OWL Properties and Individuals

```
:Book a owl:Class .  
:Writer a owl:Class .  
  
:author a owl:ObjectProperty ;  
    rdfs:domain :Book ;  
    rdfs:range :Writer .  
  
:publicationYear a owl:DatatypeProperty ;  
    rdfs:domain :Book ;  
    rdfs:range xsd:integer .  
  
:GeorgeOrwell a Writer .  
:NineteenEightyFour a :Book ;  
    :author :GeorgeOrwell ;  
    :publicationYear 1948 .
```

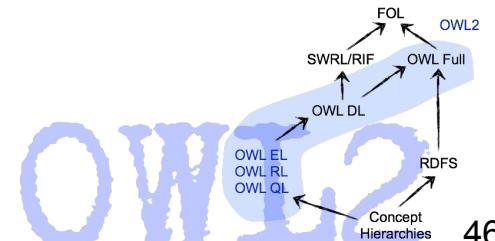


# OWL Class Hierarchies and Disjunctiveness

```
:Book a owl:Class .
:Writer a owl:Class .
:Novel a owl:Class ;
    rdfs:subClassOf :Book .
:Poet a owl:Class ;
    rdfs:subClassOf :Writer .
:Book owl:disjointWith :Writer .
```

Novel ⊑ Book  
 Poet ⊑ Writer  
 $\text{Book} \sqcap \text{Writer} \sqsubseteq \perp$

- via inference it can be entailed that :Novel and :Poet are also disjoint classes

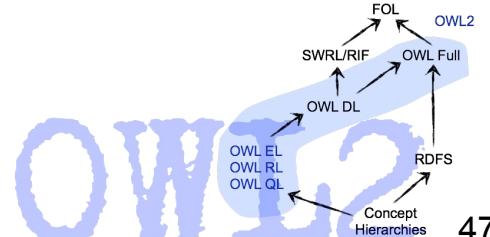


# OWL Class Hierarchies and Equivalence

```
:Writer a owl:Class .  
:Author a owl:Class :  
:Poet a owl:Class ;  
    rdfs:subClassOf :Writer .  
  
:Writer owl:equivalentClass :Author .
```

Poet  $\sqsubseteq$  Writer  
Writer  $\equiv$  Author

- via inference it can be entailed that :Poet is also an :Author



# OWL Individuals - Identity and Distinctiveness

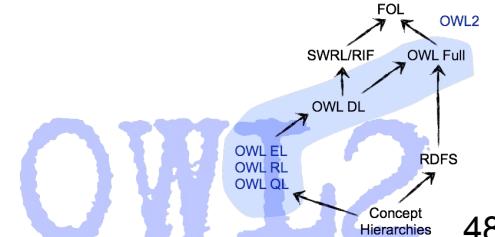
```
:NineteenEightyfour a :Novel ;
    :author :GeorgeOrwell ;
    :publicationYear 1948 ;
    owl:sameAs :ARX012345 .

:Novel a owl:Class ;
    rdfs:subClassOf :Book .

:Book a owl:Class.
```

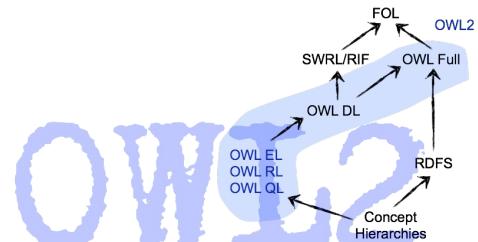
- via inference it can be entailed that :ARX012345 is a :Book
- difference of Individuals via **owl:differentFrom**

```
:ARX012345 a :Novel ;
    owl:differentFrom :ARX012346 .
```



# OWL Logical Class Constructors

- logical AND (conjunction): `owl:intersectionOf`  $\sqcap$
- logical OR (disjunction): `owl:unionOf`  $\sqcup$
- logical negation: `owl:complementOf`  $\neg$
- Logical constructors are applied to **create complex class descriptions** from atomic classes.

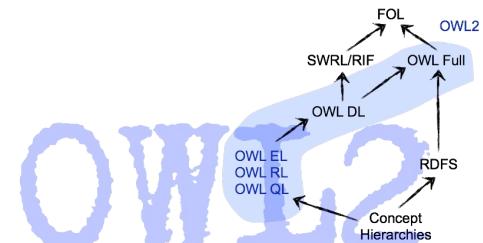


# OWL Logical Class Constructors

```
:Book a owl:Class .  
:  
:ThingsInStore a owl:Class .  
:  
:BooksInStore a owl:Class ;  
    owl:intersectionOf (:ThingsInStore :Book) .
```

BooksInStore ≡ ThingsInStore ⊓ Books

- The class `:BooksInStore` results from the intersection of all individuals of the classes `:ThingsInStore` and `:Book`

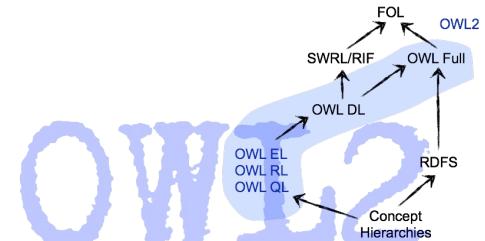


# OWL Logical Class Constructors

```
:Book a owl:Class ;  
    owl:equivalentClass [  
        owl:unionOf ( :Novel  
                      :Poetry  
                      :NonFiction )  
    ] .
```

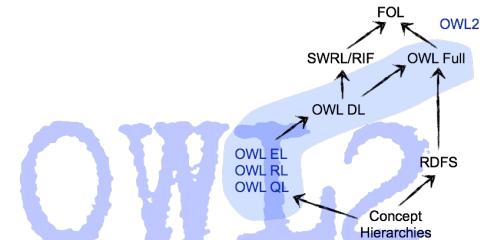
Book  $\equiv$  Novel  $\sqcup$  Poetry  $\sqcup$  NonFiction

- Novels, poetry, and non-fiction are also books



# OWL Property Restrictions

- OWL property restrictions are used to describe complex classes via properties
- restrictions on values:
  - `owl:hasValue`
  - `owl:allValuesFrom`
  - `owl:someValuesFrom`
- restrictions on cardinality:
  - `owl:cardinality`
  - `owl:minCardinality`
  - `owl:maxCardinality`

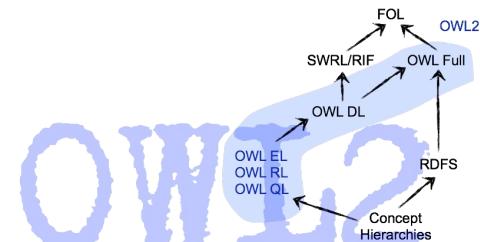


# OWL Property Restrictions with Constants

```
:OrwellsBooks a owl:Class ;
    rdfs:subClassOf
        [ a owl:Restriction ;
            owl:onProperty :author ;
            owl:hasValue :GeorgeOrwell ] .
```

$\text{OrwellsBooks} \sqsubseteq \text{author.}(\text{GeorgeOrwell})$

- Class `:OrwellsBooks` is described via fixed value assignment (=constant) of the individual `:GeorgeOrwell` to the property `:author`

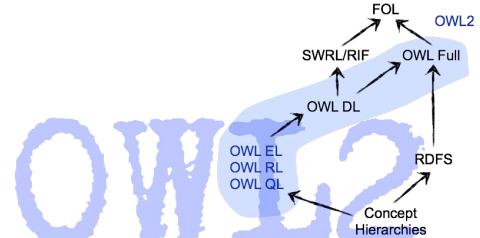


# OWL Properties Restriction with Strict Binding

```
:Poetry a owl:Class ;
    rdfs:subClassOf
        [ a owl:Restriction ;
          owl:onProperty :author ;
          owl:allValuesFrom :Poet ] .
```

$\text{Poetry} \sqsubseteq \forall \text{author.Poet}$

- **owl:allValuesFrom**  
fixes all instances of a specific class C  
as allowed range for a property p  
(strict binding)  $\forall p.C$



## Exercise 4: OWL (1)

**Transitive Property:**

XXX      rdf:type      owl:TransitiveProperty .

## Exercise 4: OWL

Start from the Turtle triples of the previous example.

### ✓ Add:

- Your URI of Dostoevsky is the **same as**:  
[http://dbpedia.org/page/Fyodor\\_Dostoyevsky](http://dbpedia.org/page/Fyodor_Dostoyevsky) and  
<https://www.wikidata.org/wiki/Q991>
- Moscow is the **same as**:  
<https://www.wikidata.org/wiki/Q649>
- State that your class "Writer" is **equivalent to**  
<http://dbpedia.org/ontology/writer>
- State that locatedIn is a **transitive property**