

# Information Service Engineering

## Lecture 6: Knowledge Graphs - 1



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**

### 3.1 Knowledge Representation and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things - URIs

3.5 Resource Description Framework (RDF) as simple Data Model

3.6 Creating new Models with RDFS

3.7 Knowledge Graphs

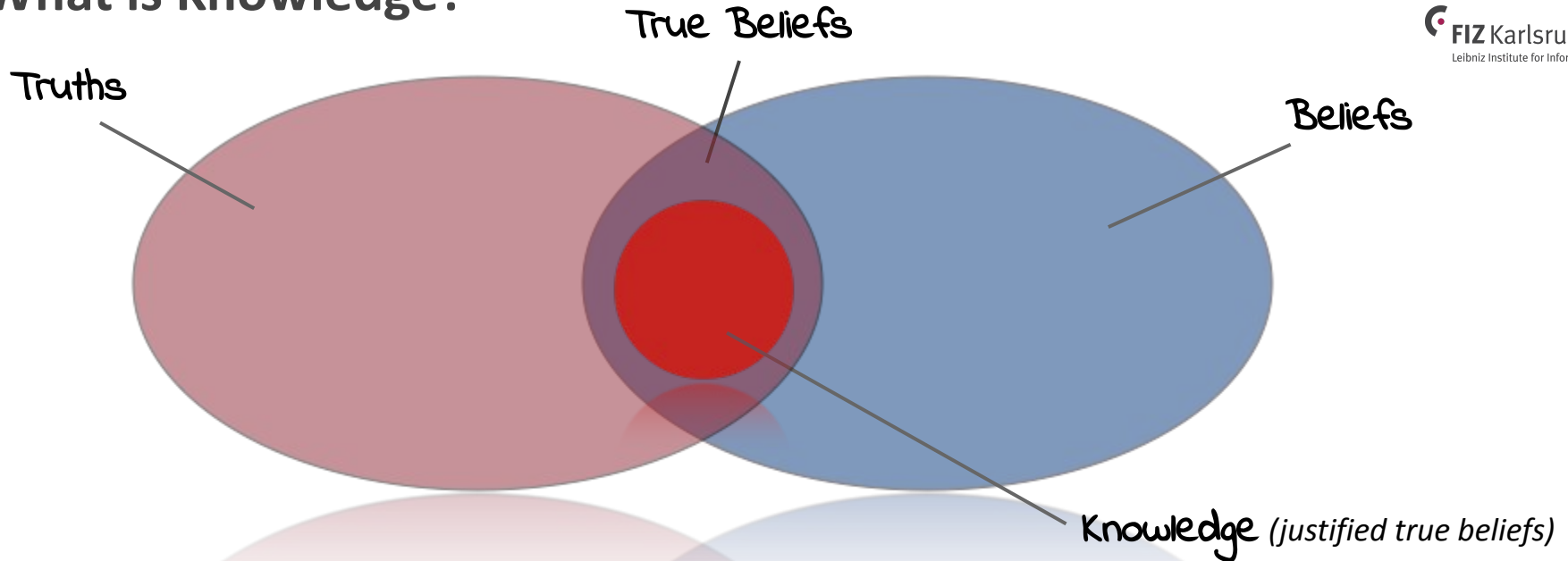
3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

3.10 Knowledge Graph Programming

What is Knowledge?

# What is Knowledge?

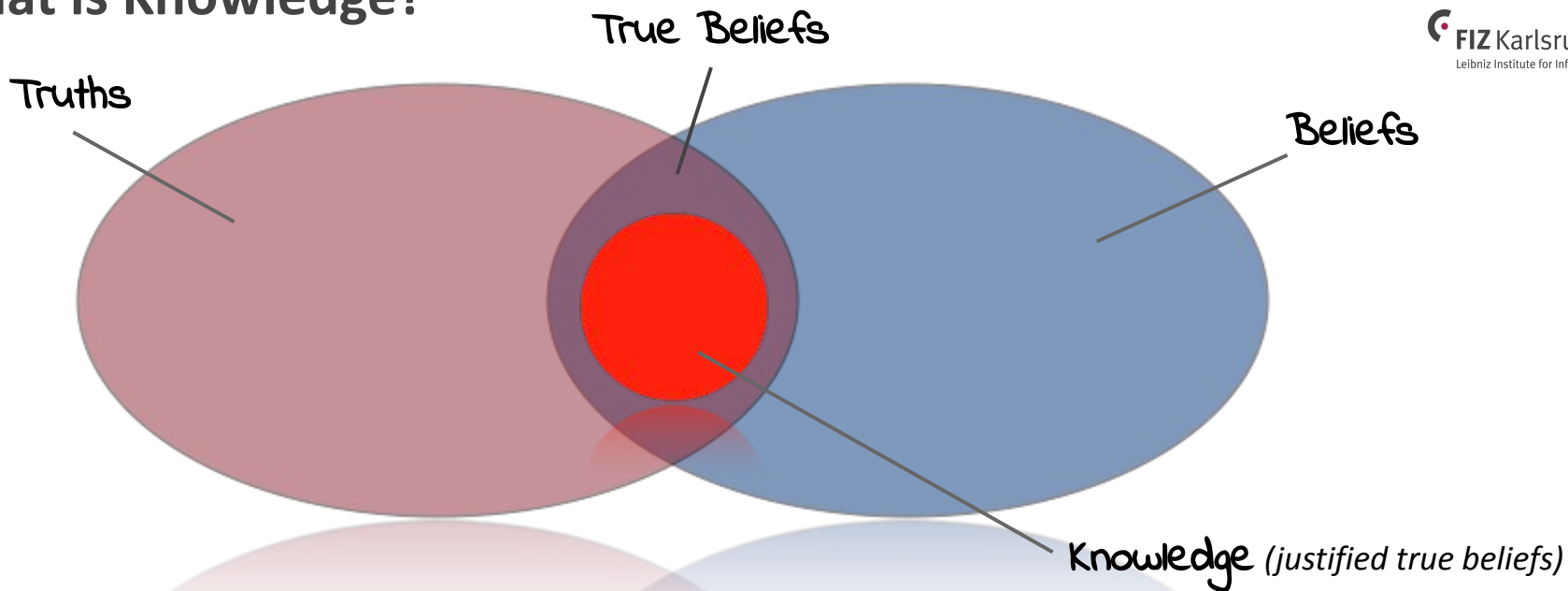


The Tripartite **Analysis of Knowledge**:  $S$  knows that  $p$  iff

- $p$  is true;
- $S$  believes that  $p$ ;
- $S$  is justified in believing that  $p$ .

[The Analysis of Knowledge](#), Stanford Encyclopedia of Philosophy, 2001.

# What is Knowledge?



Traditional Definition: „*Knowledge is a justified subset of all true beliefs*“.

To represent knowledge, we need a **formal knowledge representation = Ontologies**

Climate Change is  
the Everest of all  
problems...

# Understanding Information



symbol

Text: "Everest"

Entity Mapping  
Entity Disambiguation

Everest, Kansas

a small village

Everest, Gasfield

a gas field near Scotland

George Everest

a Surveyor General of India

Jack Everest

an Irish Football Player

...

Mount Everest

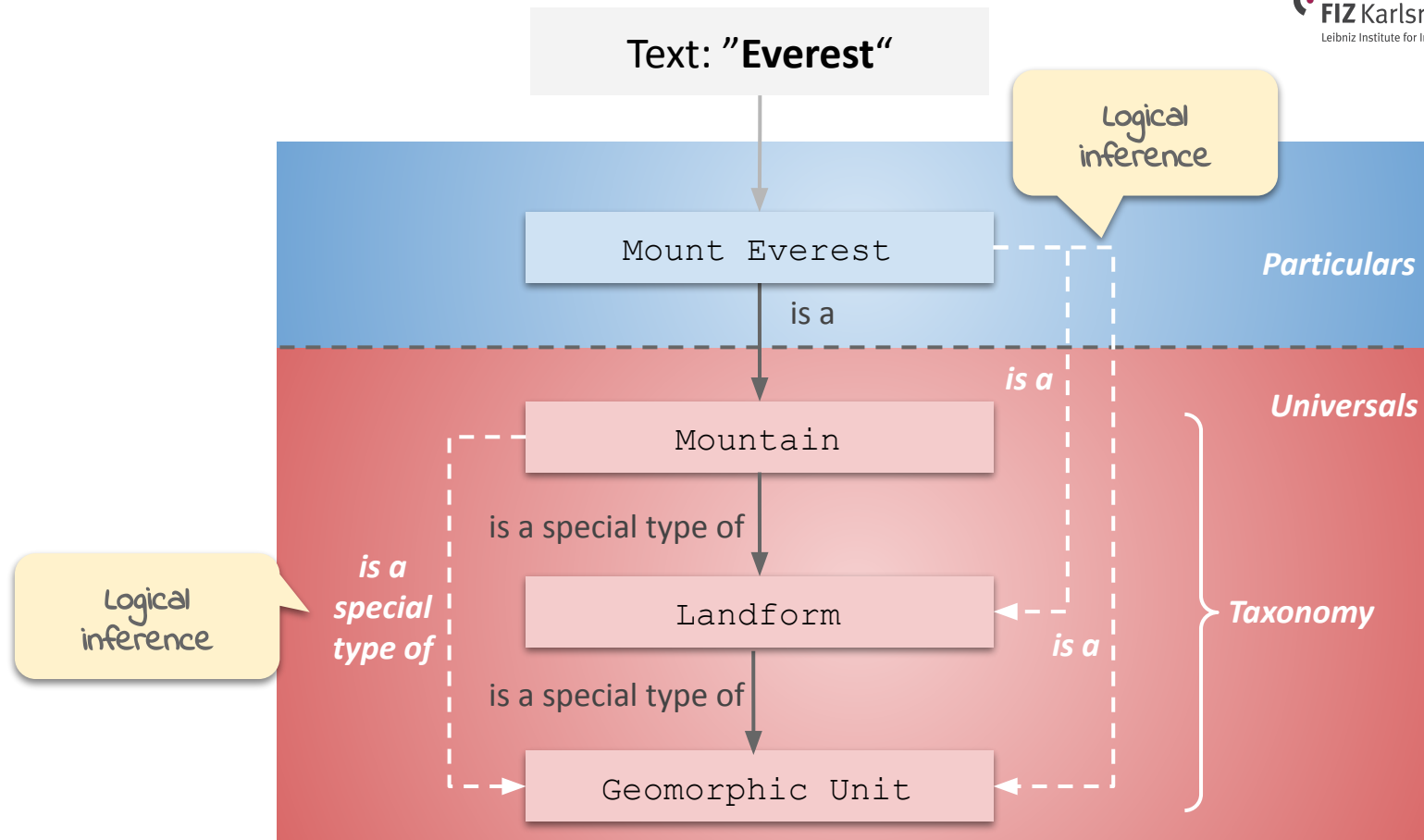
a mountain

concepts

## Disambiguation

= solution of ambiguities  
by explicitly mapping ambiguous  
language expressions to **unique  
knowledge representations**

# How to Represent Knowledge?





# Formal Knowledge Representation

- **Formal Knowledge Representation**
  - is a field of **artificial intelligence (AI)**,
  - which (unambiguously) captures the **semantics (meaning)** of **concepts, properties, relationships, and entities**
  - of specific **knowledge domains**, i.e., fields of interest or areas of concern,
  - as **structured data**.
- **Machines (computers)** must be able to **understand** formal knowledge representations.
- To “**understand**” a knowledge representation, the machine must be able to **interpret it correctly**.

# What is Ontology?

„A **theory of being**, which tries to **explain the being itself**, by developing a **system of universal categories** and their intrinsic **relationships**...“

Philosophy Definition

# What is an Ontology?

An ontology is an  
**explicit, formal specification of a shared conceptualization.**

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

Computer Science Definition

# What is an Ontology?

An ontology is an  
**explicit, formal specification of a shared conceptualization.**

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

- conceptualization:** abstract model  
(domain, identified relevant concepts, relations)
- explicit:** meaning of all concepts must be defined
- formal:** machine understandable
- shared:** consensus about ontology



# How to represent Ontologies?

- **Classes** are abstract groups, sets, or collections of objects and represent **ontology concepts**.
- Classes are characterized via **attributes**.
- **Attributes** are name-value pairs.

*„An **element** is a pure substance which cannot be broken down by chemical means, consisting of atoms which have identical numbers of protons in their atomic nuclei. The number of protons in the nucleus is the defining property of an element, and is referred to as the atomic number “*

informal description

## Element

- *name* <string>
- *chemical symbol* <string>
- *atomic number* <integer>
- *atomic weight* <float>
- *phase* <string>
- *melting point* <float>

...

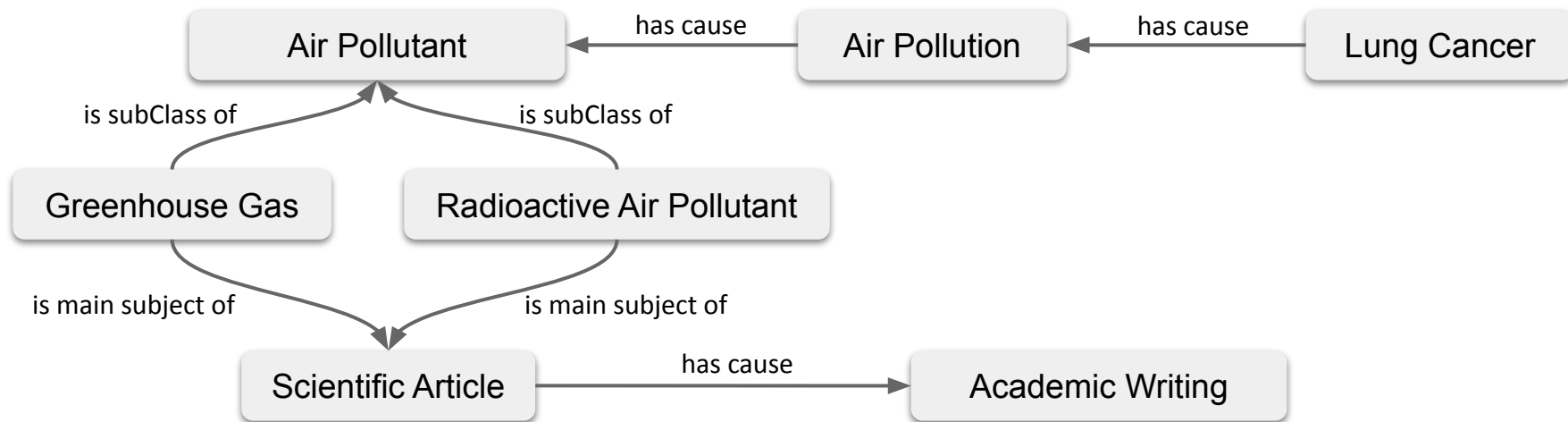
name <value type>

semi-formal description

attributes

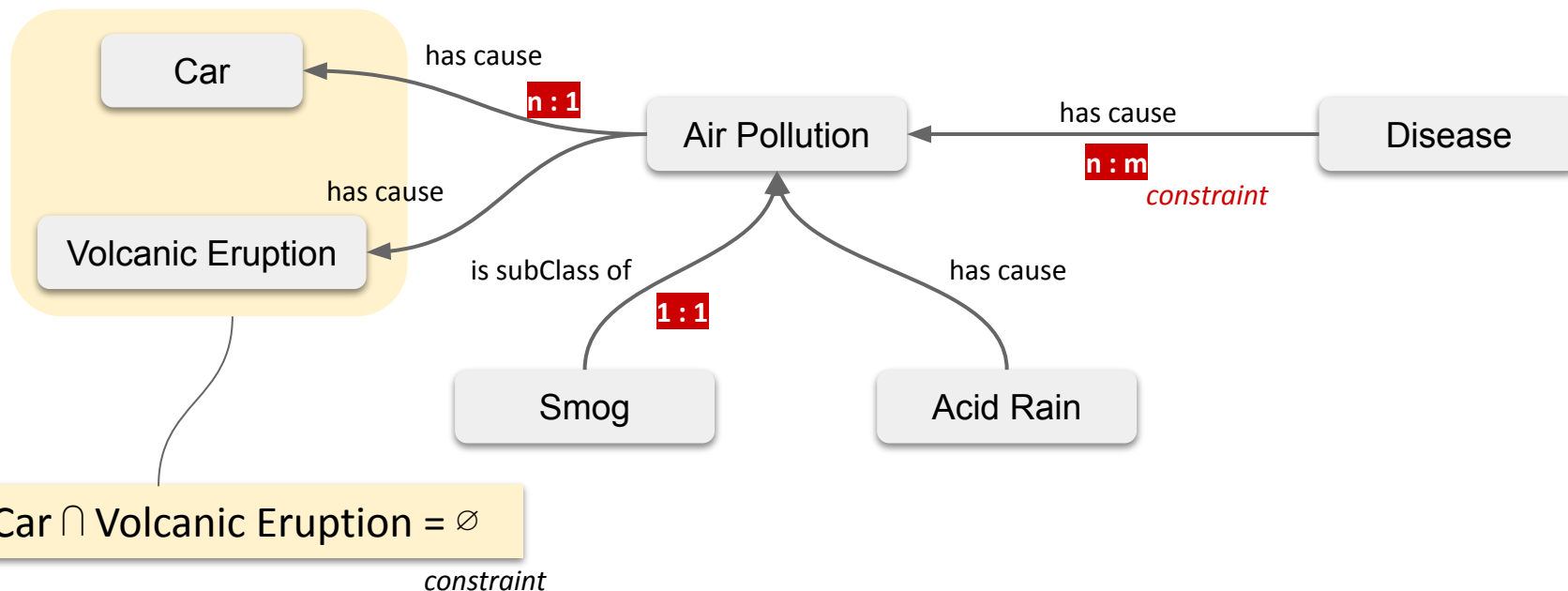
# How to represent Ontologies?

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



# How to represent Ontologies?

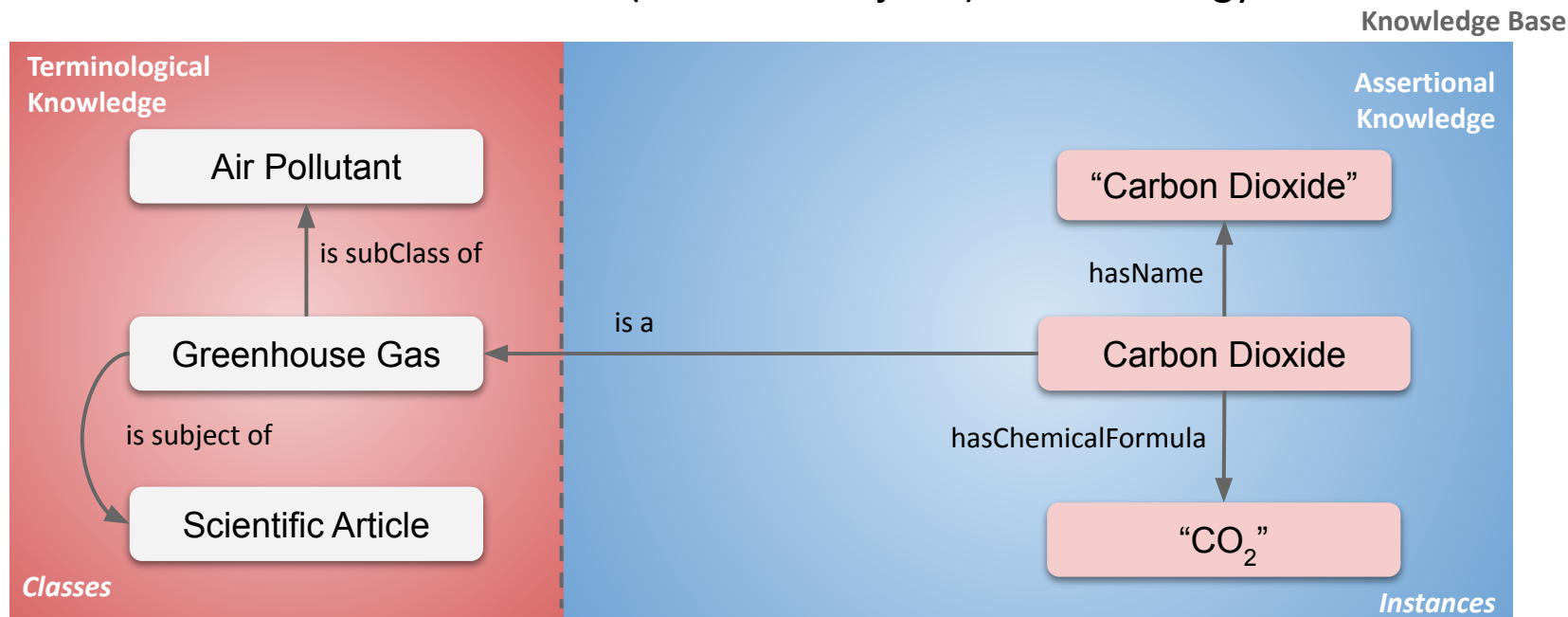
- For Relations and Attributes **Rules (Constraints)** can be defined that determine allowed/valid values.





# How to represent Ontologies?

- **Instances** describe individuals (individual objects) of an ontology.



- **Axioms** are assertions that together comprise the overall theory that the ontology describes in its domain of application.

3.1 Knowledge Representation and Ontologies

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

3.3 Linked Data Principles

3.4 How to identify and Access Things - URIs

3.5 Resource Description Framework (RDF) as simple Data Model

3.6 Creating new Models with RDFS

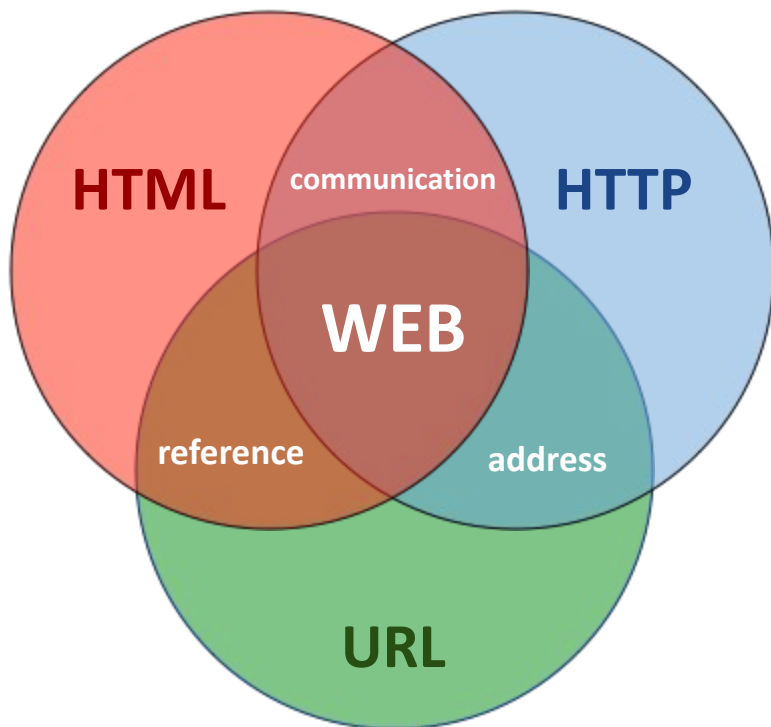
3.7 Knowledge Graphs

3.8 Querying Knowledge Graphs with SPARQL

3.9 More Expressivity with Web Ontology Language (OWL)

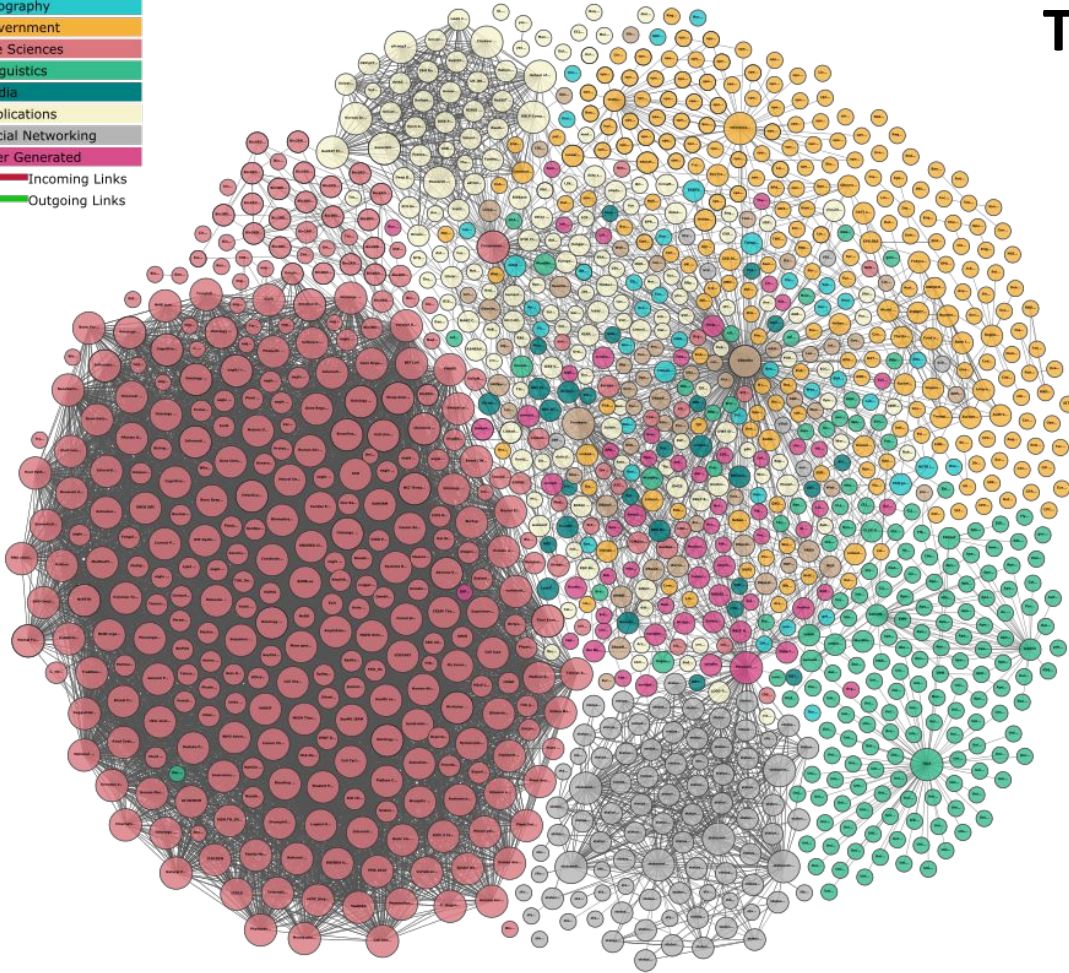
3.10 Knowledge Graph Programming

# The 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**)  
Mary works at  
`<a href="http://kit.edu">KIT</a>.`

# The Web of Data



# The Web of Data

## Data Centered Processing

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

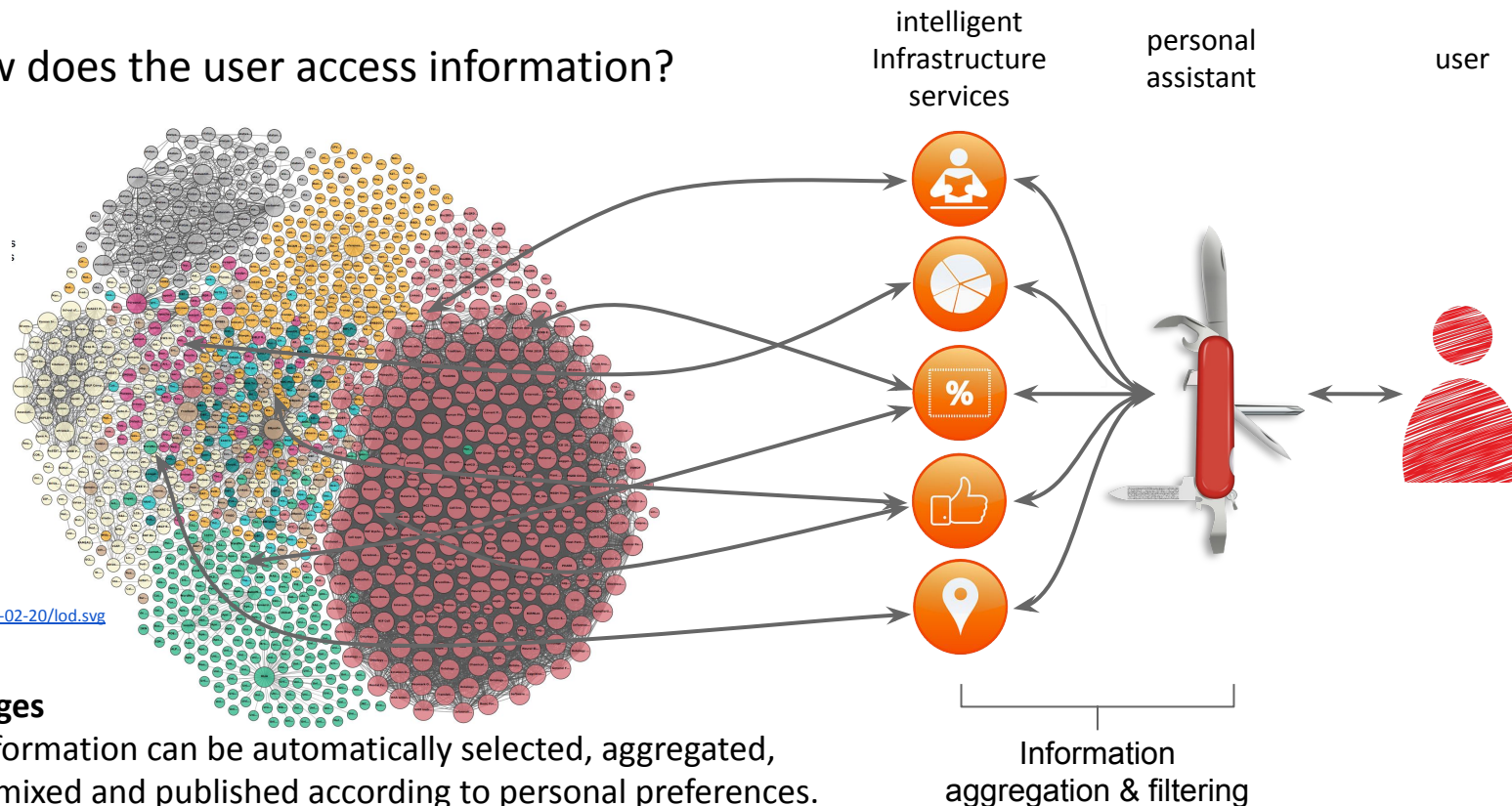
„The web of **human-readable documents** is being merged with a web of **machine understandable data**. The potential of the mixture of humans and machines working together and communication through the web could be immense.“

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

# The Web of Data

## Data Centered Processing

- How does the user access 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.

# The Semantic Web

- The Semantic Web is an **Extension of the current Web**.
- The meaning of information (Semantics) is made explicit by **formal (structured) and standardized knowledge representations (Ontologies)**.
- Thereby 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**.

*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

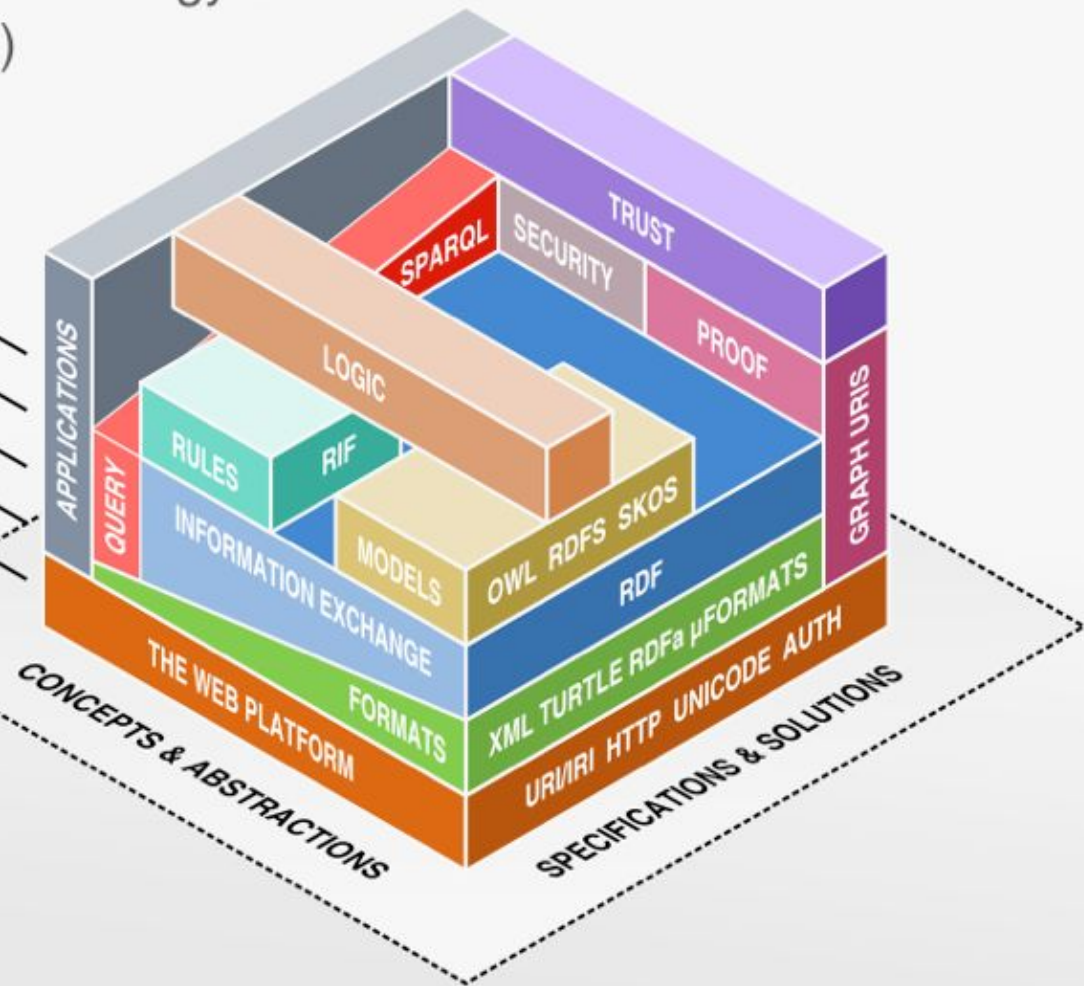
Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

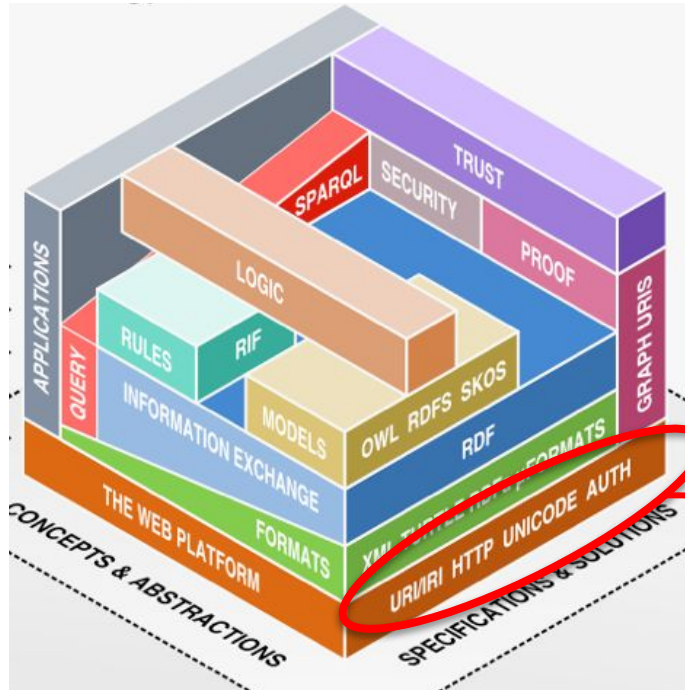
Linked Data uses a small selection of technologies

LINKED DATA





# The Semantic Web Technology Stack



Semantic Web Architecture, by Pastorcito@Wikimedia Commons, [CC-BY-SA]  
[https://commons.wikimedia.org/wiki/File:Arquitectura\\_Tecnologica\\_de\\_la\\_Web\\_Sem%C3%A1ntica.png](https://commons.wikimedia.org/wiki/File:Arquitectura_Tecnologica_de_la_Web_Sem%C3%A1ntica.png)

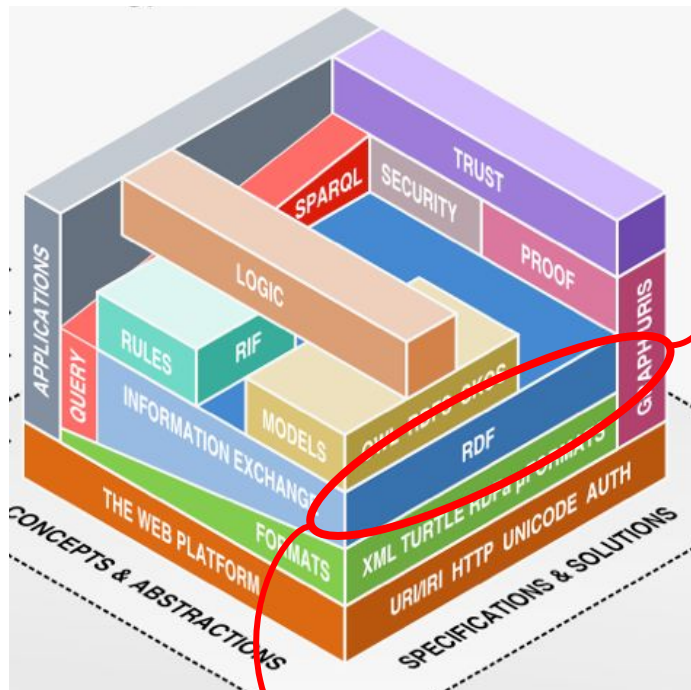
<https://www.wikidata.org/wiki/Q513>

URI - Uniform Resource Identifier

Mount Everest

[http://dbpedia.org/resource/Mount\\_Everest](http://dbpedia.org/resource/Mount_Everest)

# The Semantic Web Technology Stack



[http://dbpedia.org/resource/Mount\\_Everest](http://dbpedia.org/resource/Mount_Everest)

```
:Mount_Everest rdf:type dbo:Mountain .
:Mount_Everest foaf:name "Mount Everest"@en .
:Mount_Everest dbo:elevation 8848 .
:Mount_Everest dbo:namedAfter :George_Everest .
:George_Everest rdf:type dbo:Person .
:George_Everest dbo:birthdate "1790-07-04"^^xsd:date .
...
```



*RDF Resource Description Framework*



RDF Triple

:Mount\_Everest

*RDF Subject*

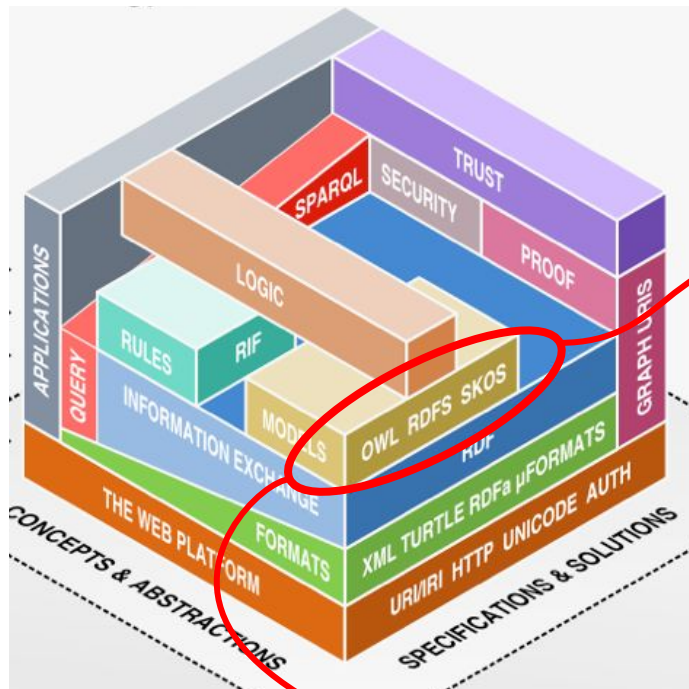
rdf:type

*RDF Property*

dbo:Mountain .

*RDF Object*

# The Semantic Web Technology Stack



Semantic Web Architecture, by Pastorcito@Wikimedia Commons, [CC-BY-SA]  
[https://commons.wikimedia.org/wiki/File:Architecture\\_Tecnolo%C3%83%89\\_de\\_la\\_Web\\_Sem%C3%A1ntica.png](https://commons.wikimedia.org/wiki/File:Architecture_Tecnolo%C3%83%89_de_la_Web_Sem%C3%A1ntica.png)

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

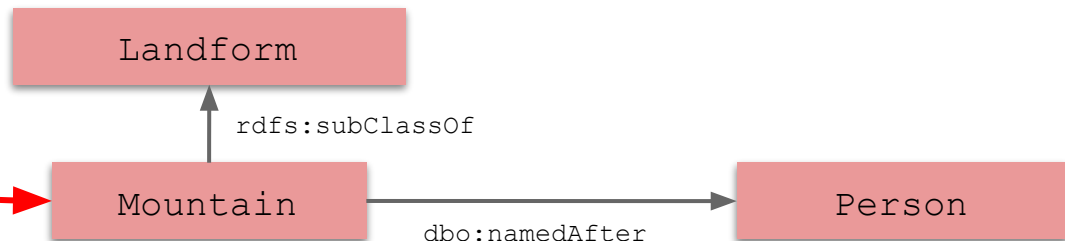
```

dbo:Mountain rdf:type owl:class .
dbo:Mountain rdfs:subClassOf dbo:Landform .
dbo:elevation rdf:type rdf:Property .
dbo:elevation rdfs:domain owl:Thing .
dbo:elevation rdfs:range xsd:integer .
dbo:namedAfter rdf:type rdf:Property .
dbo:namedAfter rdfs:domain owl:Thing .
dbo:namedAfter rdfs:range dbo:Person .
...

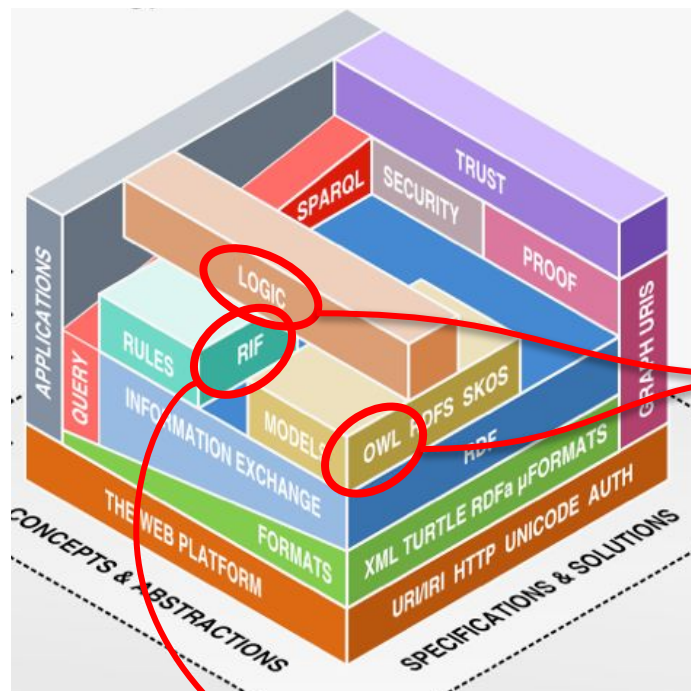
```

W3C RDFS

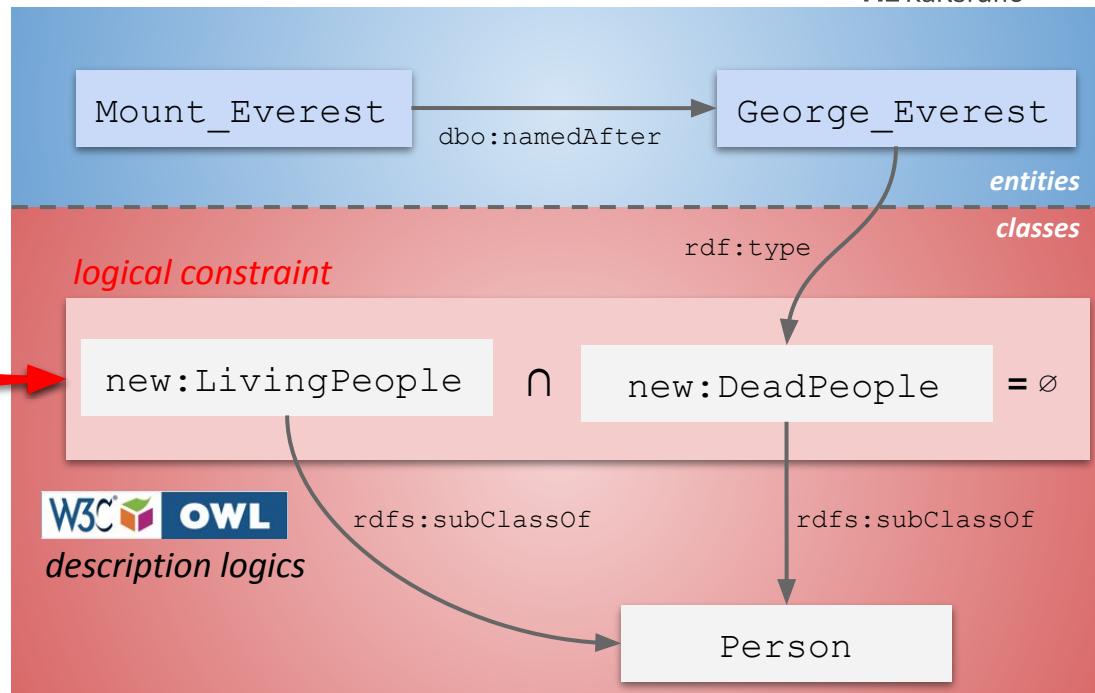
*RDF Schema*



# The Semantic Web Technology Stack



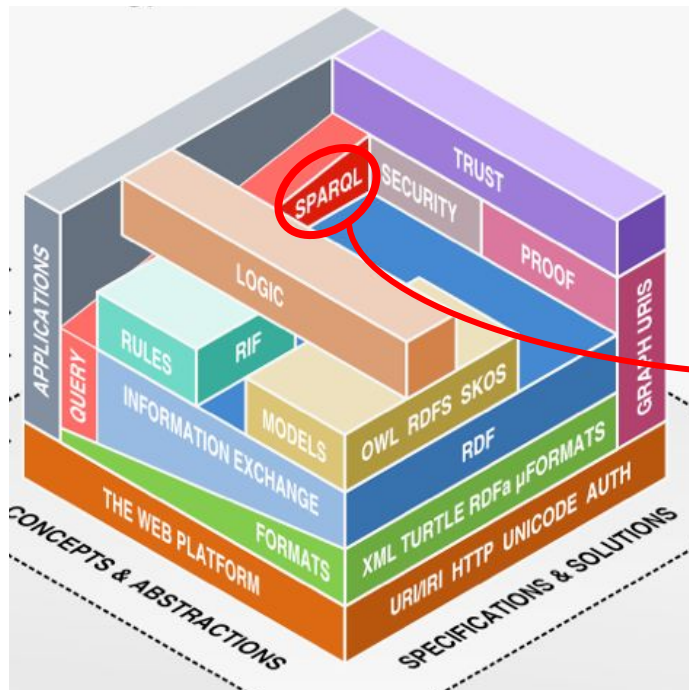
Semantic Web Architecture, by Pastorcito@Wikimedia Commons, [CC-BY-SA]  
[https://commons.wikimedia.org/wiki/File:Architecture\\_Tecnolo%C3%83ca\\_de\\_la\\_Web\\_Sem%C3%A1ntica.png](https://commons.wikimedia.org/wiki/File:Architecture_Tecnolo%C3%83ca_de_la_Web_Sem%C3%A1ntica.png)



+ *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



Semantic Web Architecture, by Pastorcito@Wikimedia Commons, [CC-BY-SA]  
[https://commons.wikimedia.org/wiki/File:Arquitectura\\_Tecnologica\\_de\\_la\\_Web\\_Sem%C3%A1ntica.png](https://commons.wikimedia.org/wiki/File:Arquitectura_Tecnologica_de_la_Web_Sem%C3%A1ntica.png)

List all **Mountains** and the **people who died there** ordered by the number of deaths...

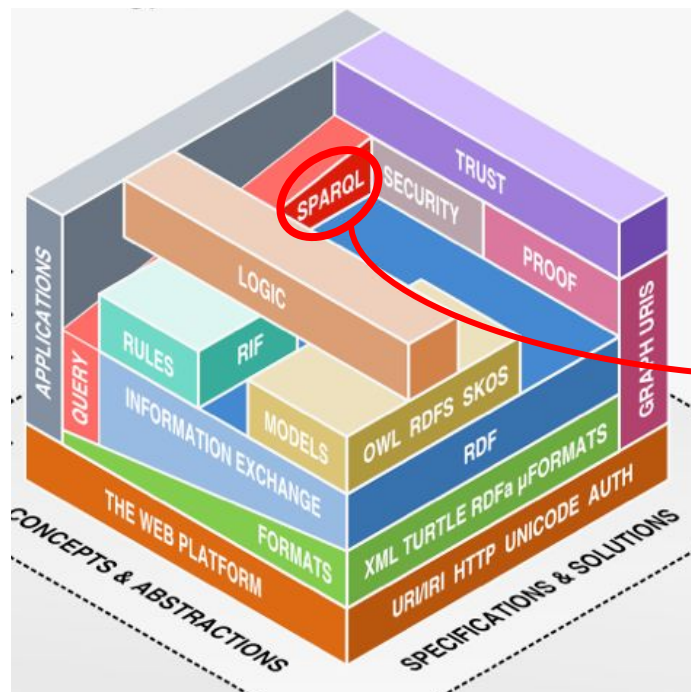


```
PREFIX rdf: <https://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT DISTINCT ?mountain (COUNT(?person) as ?cnt)
WHERE {
  ?mountain rdf:type dbo:Mountain .
  ?person dbo:deathPlace ?mountain .
}
GROUP BY ?mountain
ORDER BY DESC(?cnt)
```

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

# The Semantic Web Technology Stack



Semantic Web Architecture, by Pastorcito@Wikimedia Commons, [CC-BY-SA]  
[https://commons.wikimedia.org/wiki/File:Arquitectura\\_Tecnol%C3%B3gica\\_de\\_la\\_Web\\_Sem%C3%A1ntica.png](https://commons.wikimedia.org/wiki/File:Arquitectura_Tecnol%C3%B3gica_de_la_Web_Sem%C3%A1ntica.png)

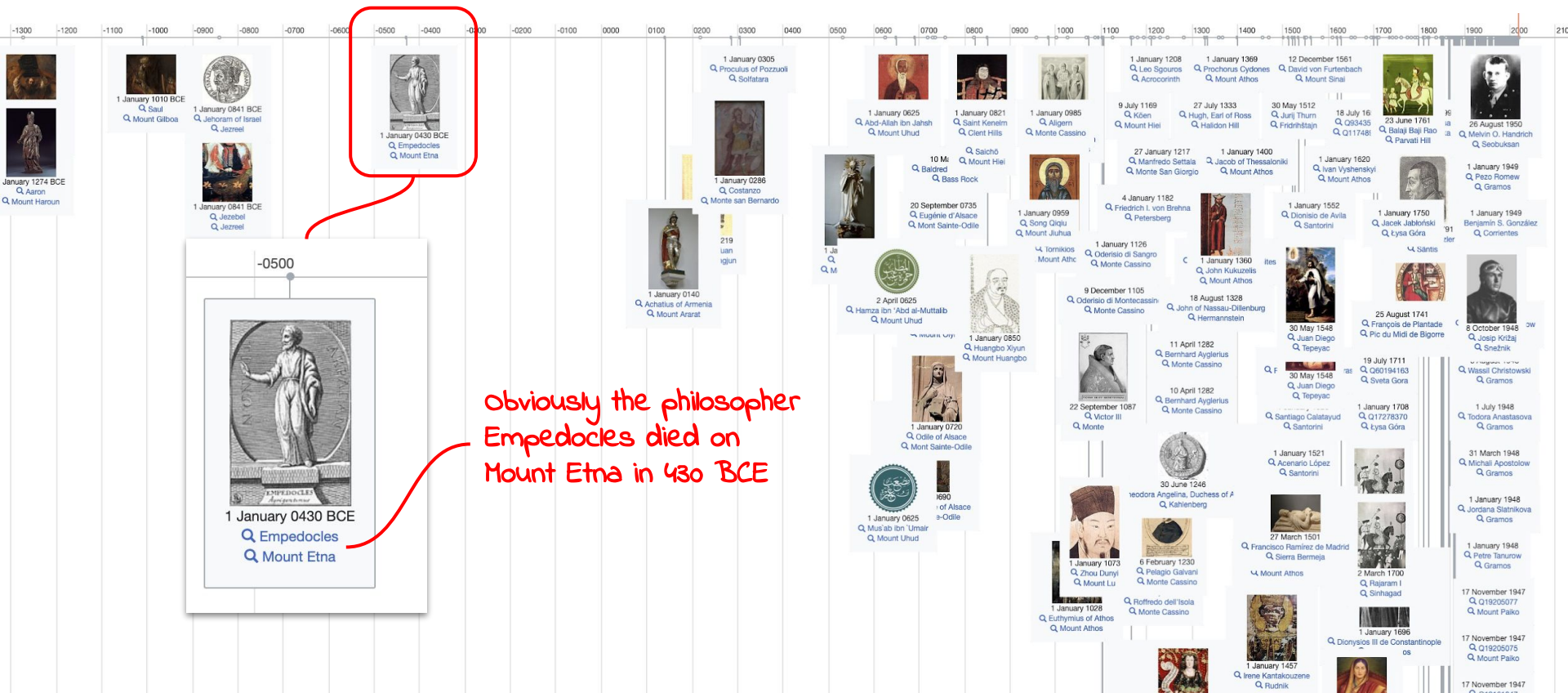
List all **Mountains** and the **people who died there** ordered by the number of deaths...



```
#defaultView:Timeline
SELECT DISTINCT ?mountain ?mountainLabel ?person
?personLabel
           ?date ?image
WHERE {
  ?mountain wdt:P31 wd:Q8502 .
  ?person wdt:P20 ?mountain .
  ?person wdt:P570 ?date .
  OPTIONAL {?person wdt:P18 ?image .}
  SERVICE wikibase:label
  {bd:serviceParam wikibase:language "en, de, fr"}
}
```

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

# The Semantic Web Technology Stack



3.1 Knowledge Representation and Ontologies

3.2 Semantic Web and the Web of Data

### **3.3 Linked Data Principles**

3.4 How to identify and Access Things - URIs

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 Web of Data

and how we make use of it



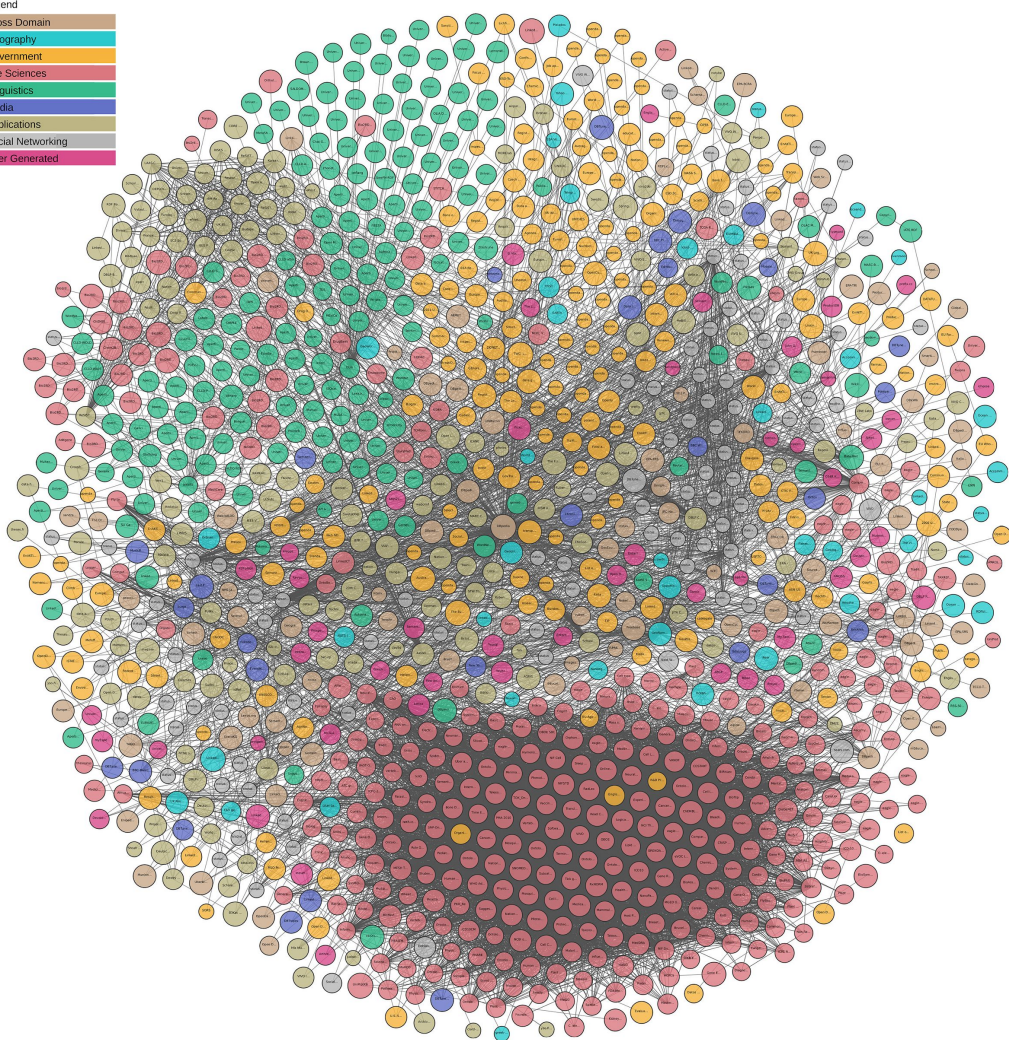
## Linked Data

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

## The Web of Data

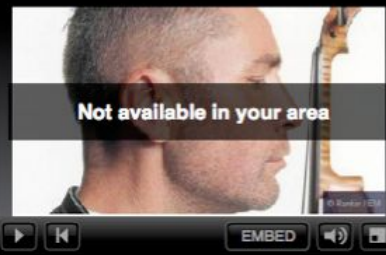
- 9,960 datasets
- >149 billion facts
- >800 million links  
(April 2017)

<http://lod-cloud.net/>



# Johann Sebastian Bach

Born 21 März 1685. Died 28 Juli 1750.



Not available in your area

**BBC PROMS**  
Nigel Kennedy And His Band Play Bach  
At the late night Prom, Nigel Kennedy and his band play arrangements of pieces by Bach.

JOHANN SEBASTIAN BACH 1 of 19 Featured on BBC MUSIC SHOWCASE

NIHEL KENNEDY AND HIS BAND PLAY...  
NIGEL KENNEDY PLAYS...  
SIEM ENO - THE BIRTH OF...  
J.S. BACH IN...

## Biography

Born in the German town of Eisenach, J. S. Bach was a chorister from violinist before taking his first organist post at Arnstadt while still a teenager. It was in Weimar, as court organist from 1708, that Bach began to produce mostly cantatas, and wrote many of his great organ works, as well as organ transcriptions of concertos, and violas.

In 1717 Prince Leopold offered him the position of Kapellmeister at Cöthen, where he wrote the Brandenburg Concertos, the four Orchestral Suites and the violin concertos, and married his second wife Anna Magdalena, who bore 13 children. Bach's fecksome duties in his final job, as Kantor of the Thomasschule in Leipzig from 1723 until his death, involved teaching Latin and Music, choir-training, and writing and directing music for the services.

Nevertheless he managed also to write the Mass in B minor, the six choral Motets, The Art of Fugue, The Musical Offering and Goldberg Variations during this time. His inventive contrapuntalism became unfashionable soon after his death until the early 19th century, since when his reputation has remained unquestioned.

Profile © Edward Ehesaria

## Links & Information

- LINKS**
- Biography at [jbsbach.org](http://jbsbach.org) and [pbs.org/nigelkennedyeducationbach.html](http://pbs.org/nigelkennedyeducationbach.html)
- Wikipedia article on Johann Sebastian Bach
- IMDb at [imdb.com/name/nm0001925](http://imdb.com/name/nm0001925)
- Last.fm page on Johann Sebastian Bach
- Discogs at [discogs.com/artist/johann-sebastian-bach](http://discogs.com/artist/johann-sebastian-bach)
- MusicBrainz entry on Johann Sebastian Bach

**PERSONAL RELATIONSHIPS**  
Parent of Johann Christoph Friedrich Bach, Johann Christian Bach, Wilhelm Friedemann Bach and Carl Philipp Emanuel Bach

Share This Page 1 so far  
Share Facebook Twitter

## Now On The BBC

Composers A-Z  
Explore more composers and their works at Radio 3

**Composers A-Z**

Discover the music of the great composers with Radio 3

## BBC Music Showcase

**BBC MUSIC SHOWCASE**  
Watch and listen to exclusive music clips

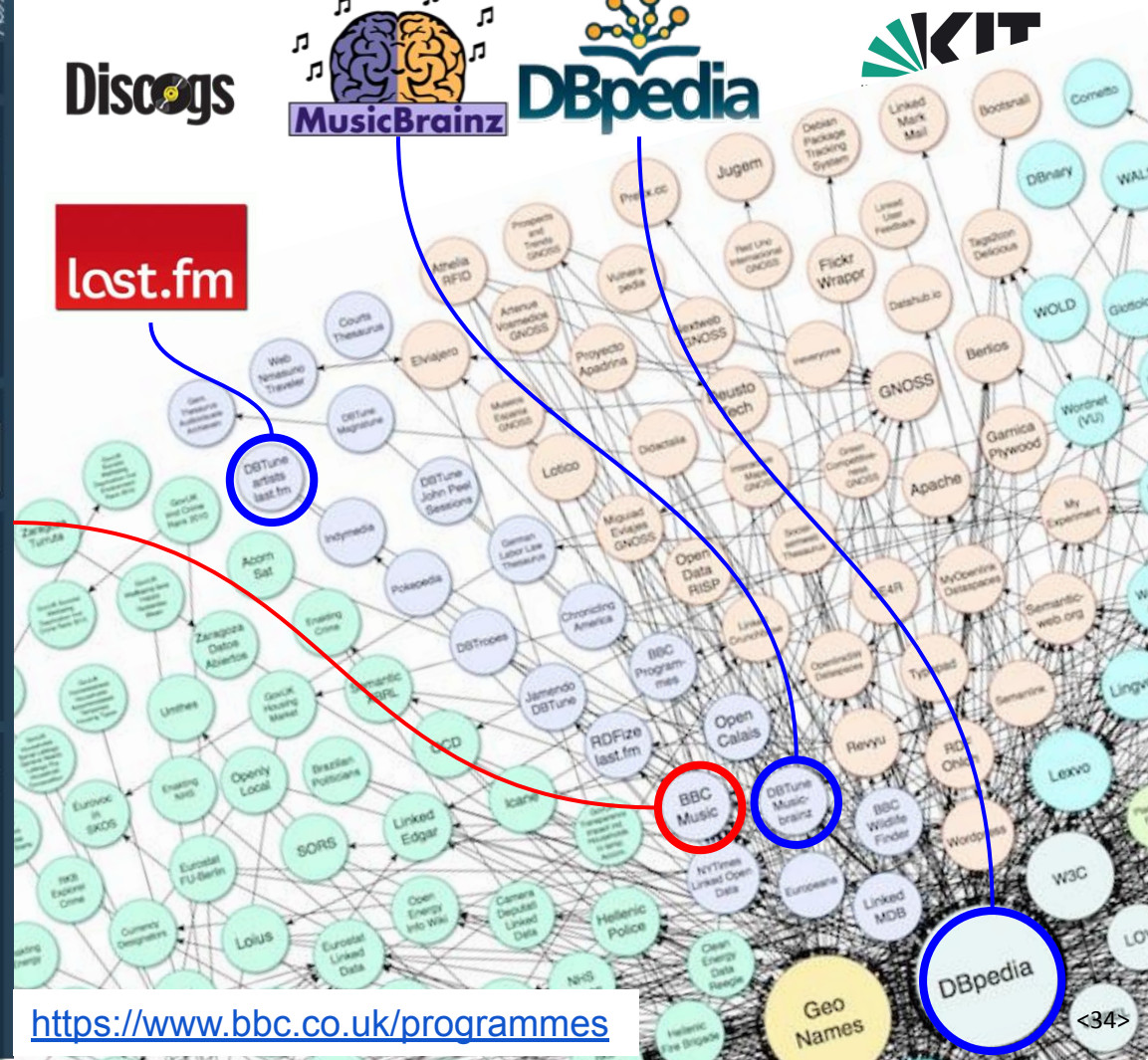
## Clips From Latest Album Review

**Keyboard Concertos (piano: Alexander Thamsud)**

- 1 Concerto for keyboard and orchestra BWV1054 in D minor - I Allegro
- 2 Concerto for keyboard and orchestra BWV1054 in D minor - II Allegro

## Played By

- Since December 2006
- Broadcast**  
BBC Radio 3
  - Desert Island Discs**  
BBC Radio 4
  - In Tune**  
BBC Radio 3
  - The Early Music Show**  
BBC Radio 3



<https://www.bbc.co.uk/programmes>

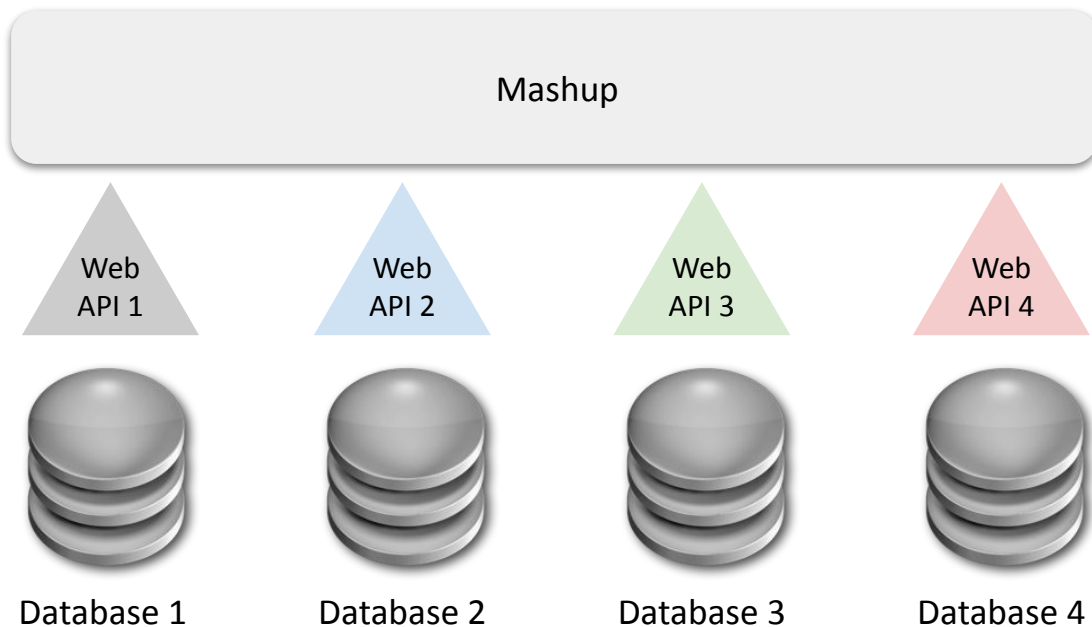
# Data Access in the traditional Web

- Data can only be found on the Web, if it is available at some website.



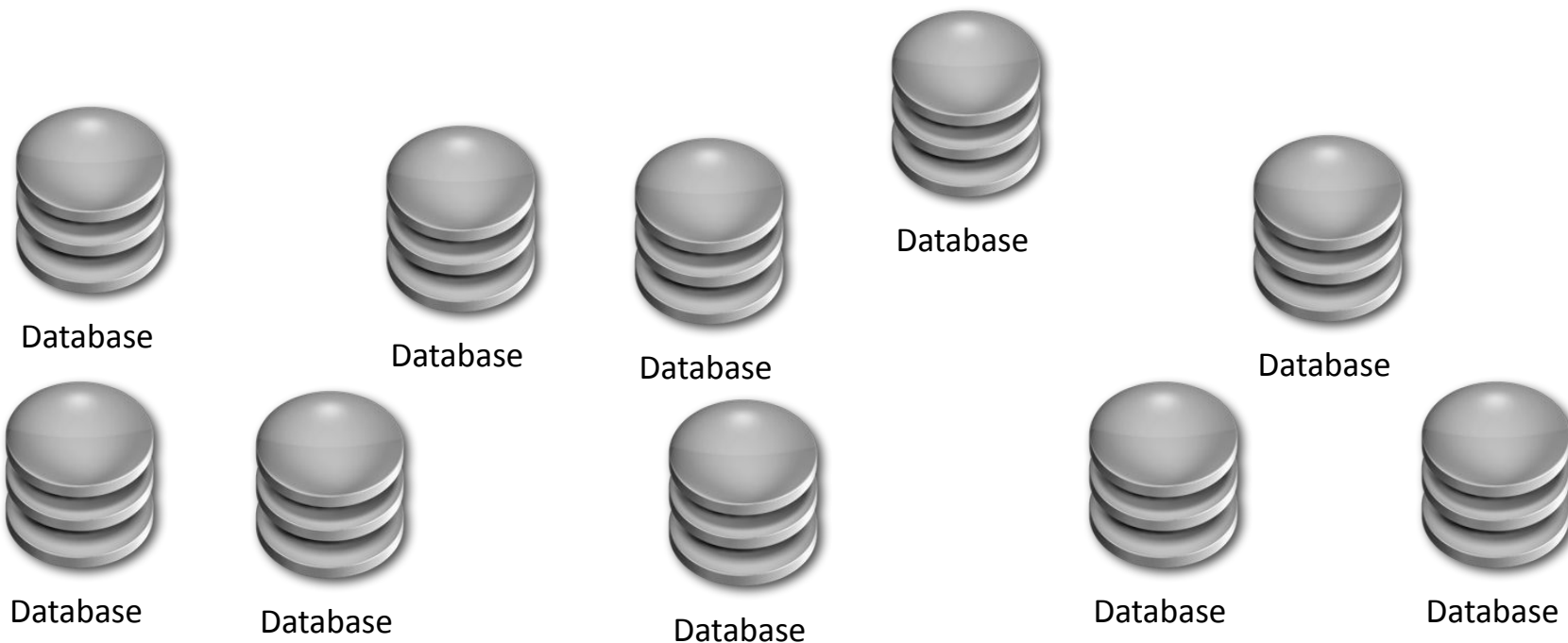
# Data Access in the traditional Web

- There is a number of different (proprietary) **Web APIs**, data exchange formats, and **Mashups** on top of that.



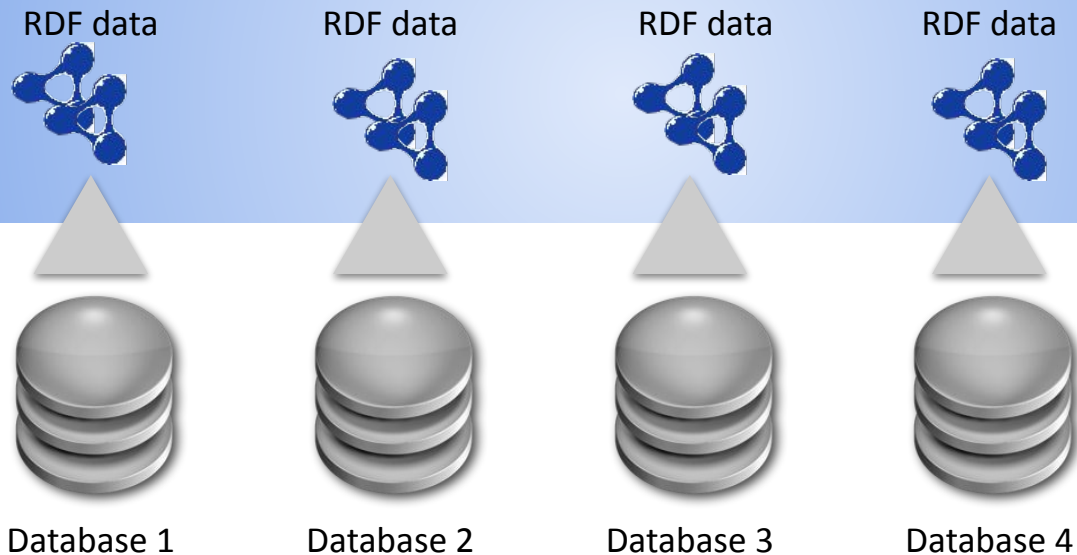
# In the traditional Web...

- Data is locked up in **small data islands**.
- Other applications usually cannot access this data...



# How to get rid of Closed Data Island?

- **Apply Linked Data technology**
  - to publish (structured) data on the web,
  - to draw connections from one data source to data from other data sources.



Linked Data Layer

# 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.

## The Benefit of using Linked Data at BBC Website

- **Information is dynamically aggregated** from external, publicly available data (Wikipedia, MusicBrainz, Last.FM, Discogs,...)
- **No Screen Scraping**
- **No specialized API**
- All data **available as Linked Open Data**
- Data access via **simple HTTP Request**
- Data is **always up-to-date without manual interaction**

Y. Raimond, T. Scott, P. Sinclair, L. Miller, S. Betts, and F. McNamara,  
Case Study: Use of Semantic Web Technologies on the BBC Web Sites, W3C, 2010.



3.1 Knowledge Representation and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

**3.4 How to identify and Access Things - URIs**

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

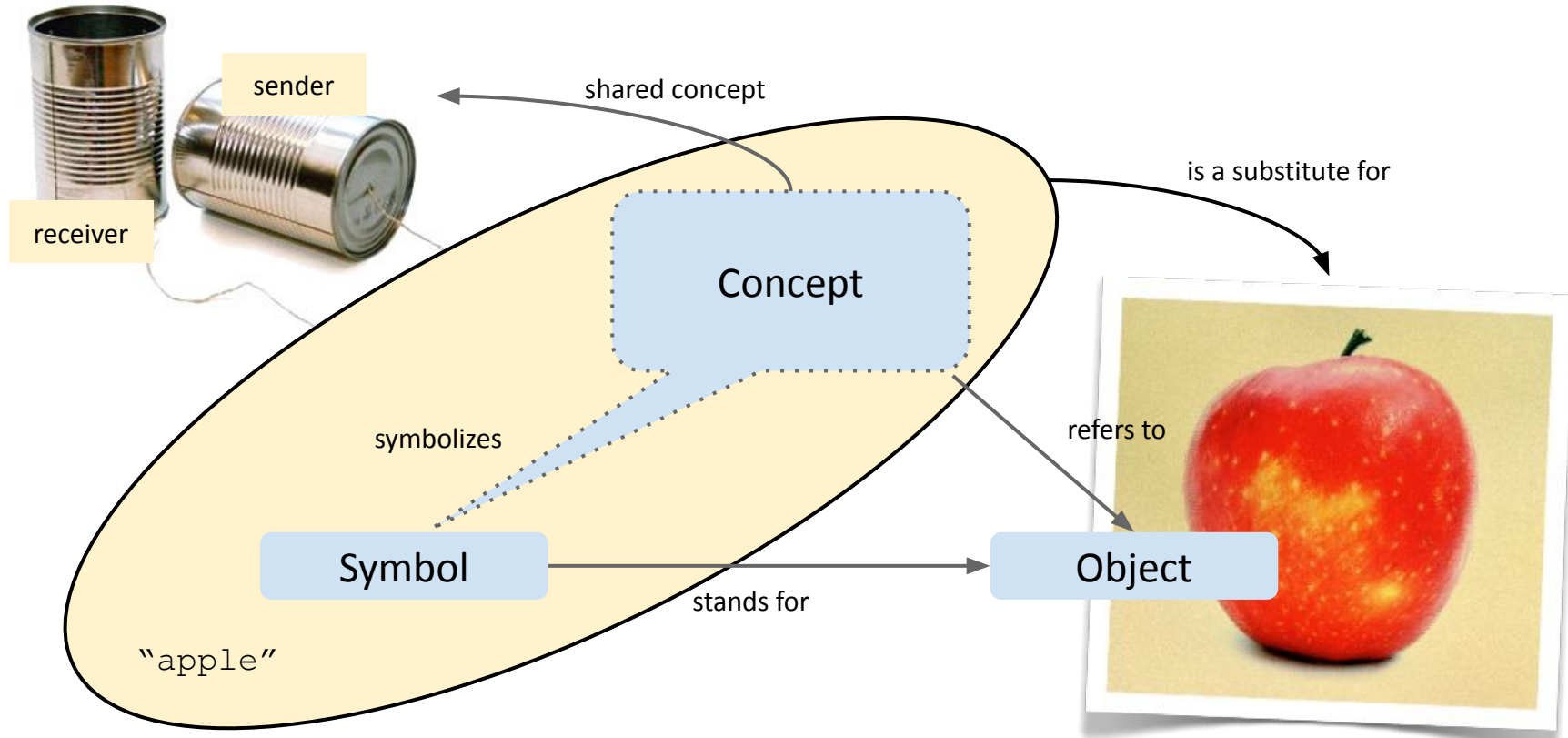
What is this?





*Ceci n'est pas une pomme.*

# Semiotic Triangle



# Uniform Resource Identifier

- A **Uniform Resource Identifier (URI)** defines a simple and extensible schema for worldwide unique identification of abstract or physical resources (RFC 3986).

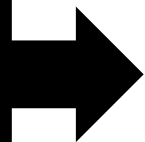
# Uniform Resource Identifier

- A **Uniform Resource Identifier (URI)** defines a simple and extensible schema for worldwide unique identification of abstract or physical resources (RFC 3986).
- A **Resource** can be every object with a clear identity (according to the context of the application),
  - as e.g., web pages, books, locations, persons, relations among objects, abstract concepts, etc.
- URI concept is already established in various domains, as e.g.,
  - the Web (URL),
  - Books and publications (ISBN, ISSN),
  - Digital Object Identifier (DOI).

# URL

identify  
what exists  
on the web

<http://www.mywebsite.org/>

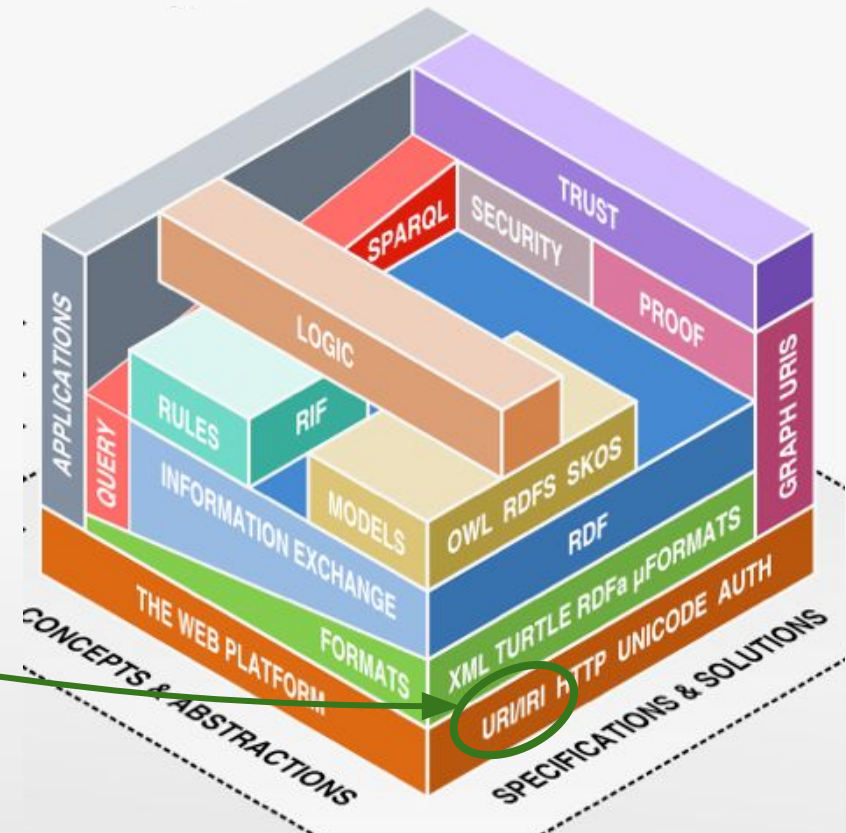
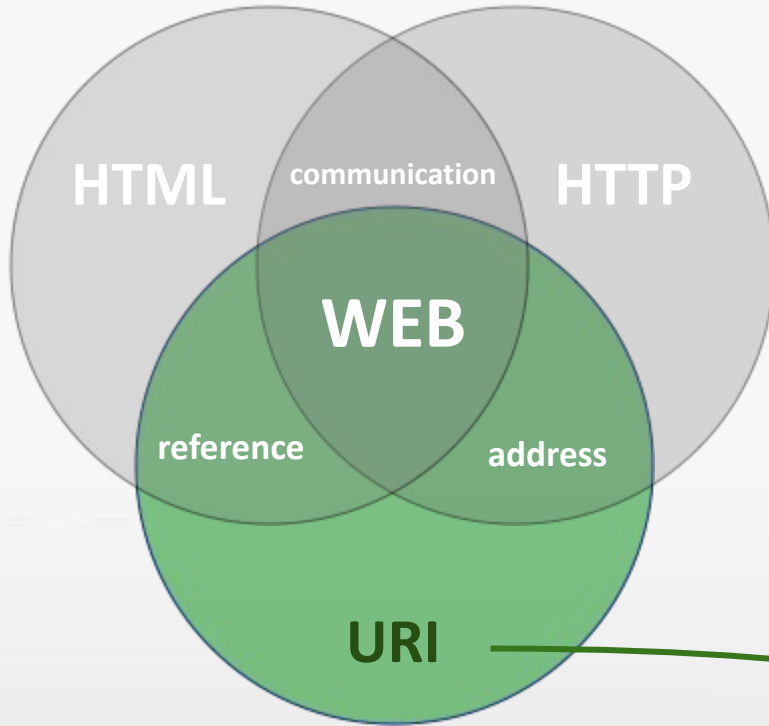


# URI

identify  
on the web  
what exists

[http://hpi.de/harald\\_foaf.rdf#me](http://hpi.de/harald_foaf.rdf#me)

# Linked Data and the Web





URI

<http://en.wikipedia.org/wiki/Earth/>

Resource

identifies

Designatum

describes

represents (stands for)

Representation

Designator

defines

Metadata:

Content-type: text/html

Data:

<!DOCTYPE html>

<html class="client-nojs" lang="en" dir="ltr">

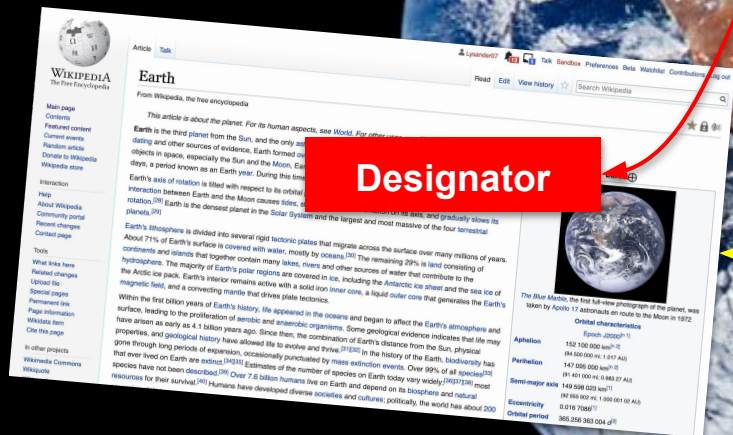
<head>

<meta charset="UTF-8"/>

<title>Earth - Wikipedia</title>

...

</html>



The Earth is described (designated) by a web page.

# What if a URI does not exist (yet)?

- Define a URI by yourself:
  - **Avoid overlaps** → use your own website as namespace
  - Enable **documentation at the same place** → HTTP Content Negotiation
- Use **separate URIs** for the **resource (*Designatum*)** and **its documentation (*Designator*)** via
  - HTTP Content Negotiation and/or
  - URI references (e.g. via "#" fragment identifier)

# HTTP Content Negotiation

Let's try an example:

- I want to have **information** about the Earth from DBpedia.

1 HTTP GET request  
Accept Header: text/html

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

URI represents *Designatum*

2 HTTP/2 303 See Other

<http://dbpedia.org/page/Earth>

3 URI represents *Designator*

HTTP GET request  
Accept Header: text/html



4 HTML Document

# HTTP Content Negotiation

Let's try another example:

- I want to have **machine readable information** about the Earth from DBpedia.

1 HTTP GET request  
Accept Header: application/rdf+xml `http://dbpedia.org/resource/Earth`  
URI represents *Designatum*

2 HTTP/2 303 See Other

3 `http://dbpedia.org/data/Earth`  
URI represents *Designator*



HTTP GET request  
Accept Header: application/rdf+xml

4 RDF/XML Document

# HTTP Content Negotiation

## Let's try it ourselves:

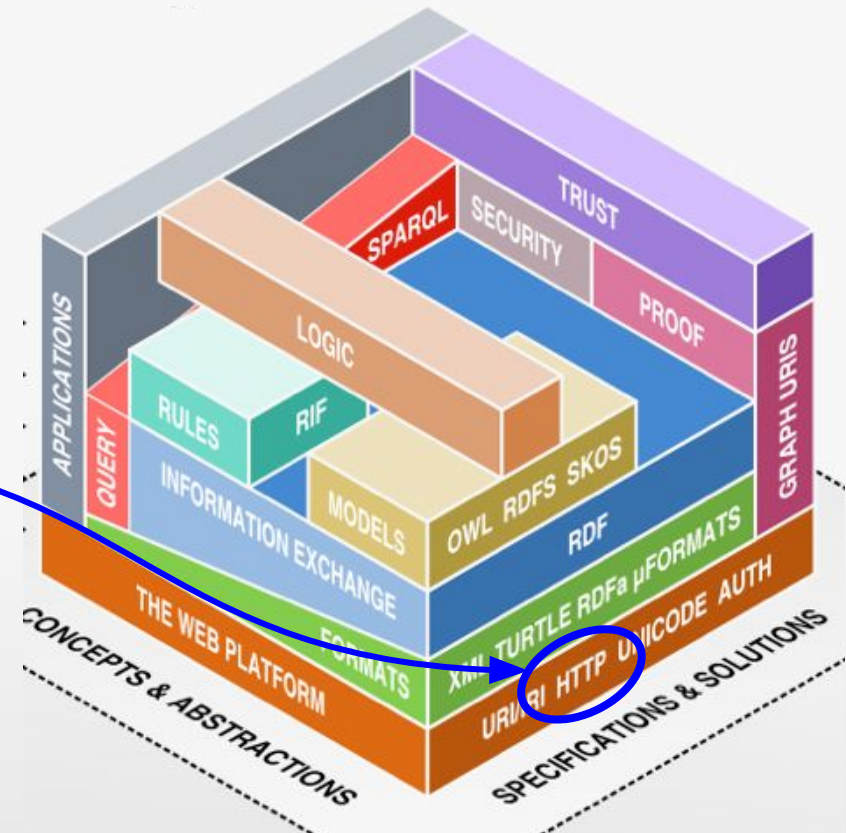
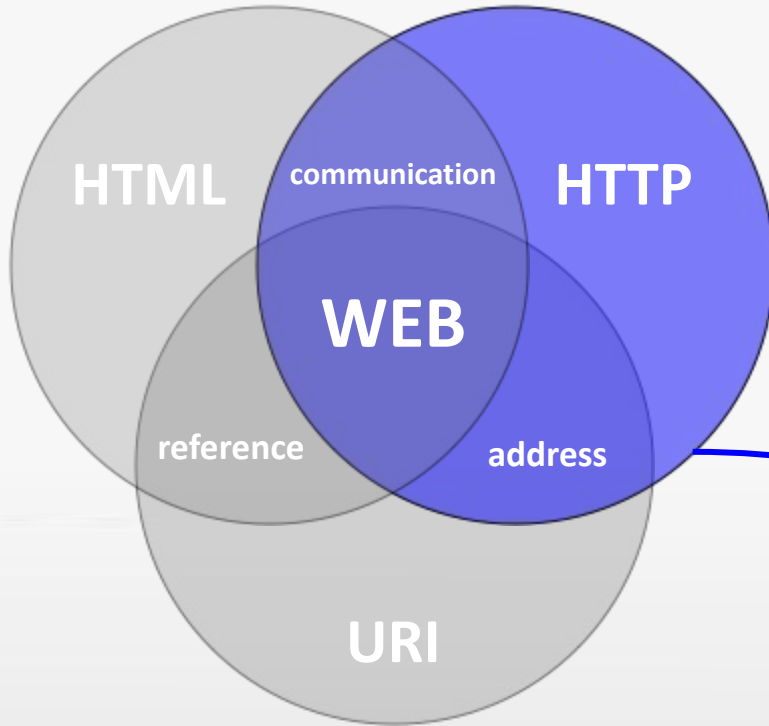
- I want to have **information** about the Earth from DBpedia:

```
curl -L -H "Accept: text/html" http://dbpedia.org/resource/Earth
```

- I want to have **machine readable information** about the Earth from DBpedia:

```
curl -L -H "Accept: application/rdf+xml" http://dbpedia.org/resource/Earth
```

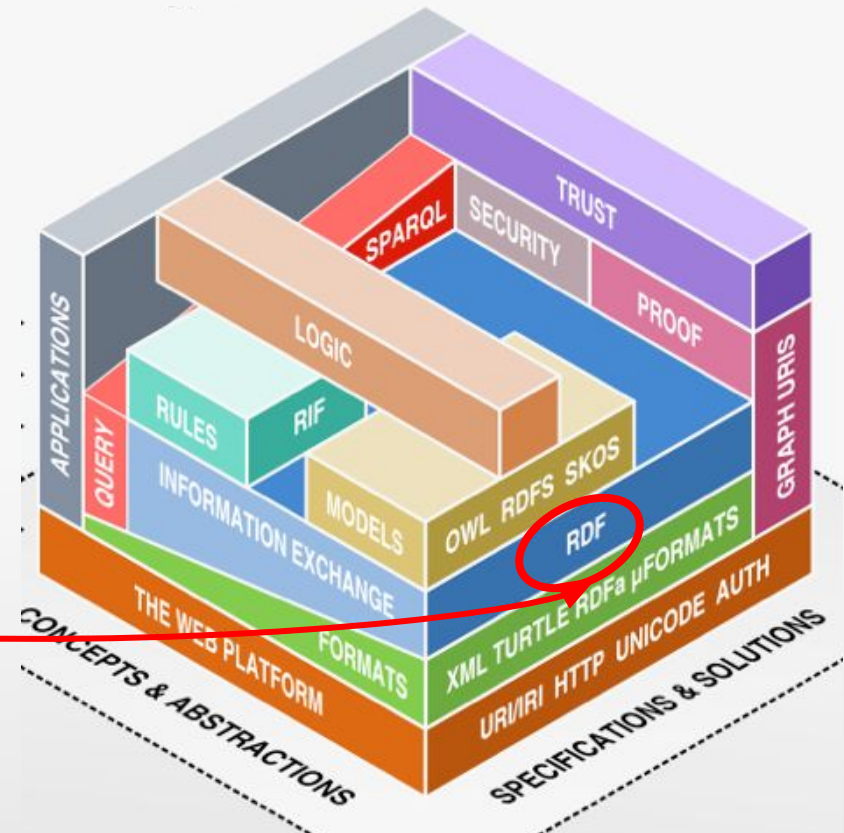
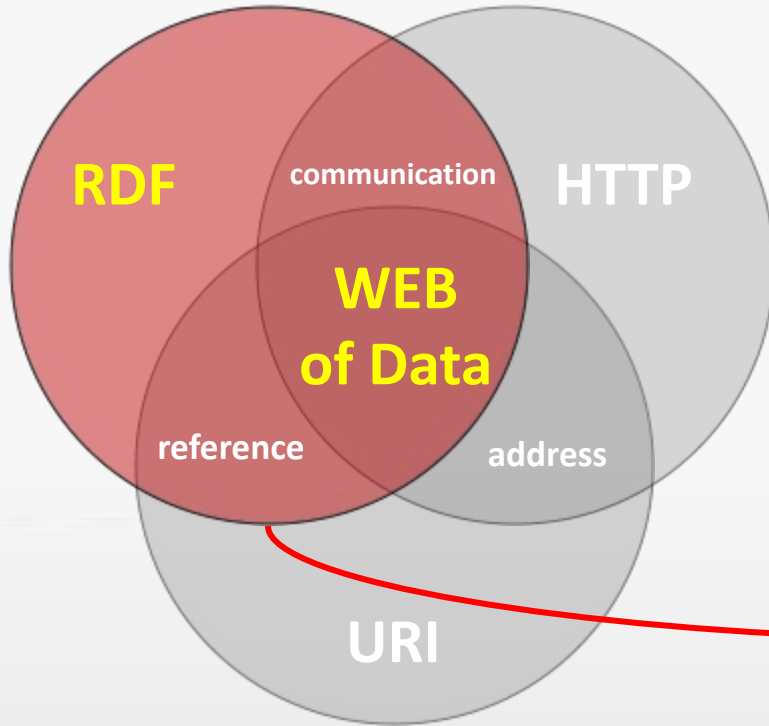
# Linked Data and the Web



# 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.

# Linked Data and the Web





3.1 Knowledge Representation and Ontologies

3.2 Semantic Web and the Web of Data

3.3 Linked Data Principles

3.4 How to identify and Access Things - URIs

**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 - 1

## Bibliography

- Amit Singhal, [Introducing the Knowledge Graph: things, not strings](#), Google Blog, May 16, 2012.
- J. J. Ichikawa, [The Analysis of Knowledge](#), Stanford Encyclopedia of Philosophy, 2001.
- Thomas R. Gruber: [A Translation Approach to Portable Ontology Specifications](#). Knowledge Acquisition, 5(2):199-220, 1993.
- Tim Berners-Lee, James Hendler, Ora Lassila: [The Semantic Web](#), Scientific American, 284(5), pp. 34-43(2001).
- S. Hitzler, S. Rudolph, Foundations of Semantic Web Technologies, Chapman / Hall, 2009.
- Y. Raimond, T. Scott, P. Sinclair, L. Miller, S. Betts, and F. McNamara, [Case Study: Use of Semantic Web Technologies on the BBC Web Sites](#), W3C, 2010.

# 3. Knowledge Graphs - 1

## Syllabus Questions

- What is **knowledge**?
- How does **knowledge representation** work?
- What are **formal knowledge representations**?
- What is an **ontology**? (Definition and Explanation)
- What are typical **constituents of an ontology**?
- What is the **Web of Data**?
- What is the **Semantic Web**?
- What is the difference between **URI** and **URL**?
- What's the difference between **identification** and **representation**?
- Explain the **Linked Data Principles**.
- What is the **benefit of using Linked Data Principles**?
- Explain **HTTP Content Negotiation** for human readable and machine readable resources.