



# Knowledge Graphen

## Publizieren und forschen mit Wissensgraphen

**Prof. Dr. Harald Sack**

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

AIFB – Karlsruhe Institute of Technology

**27.04.2023 - Pausanio Akademie**



# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

Prof. Dr. Harald Sack



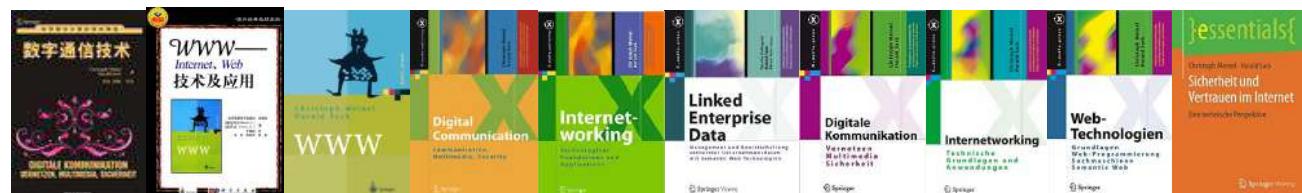
FIZ Karlsruhe

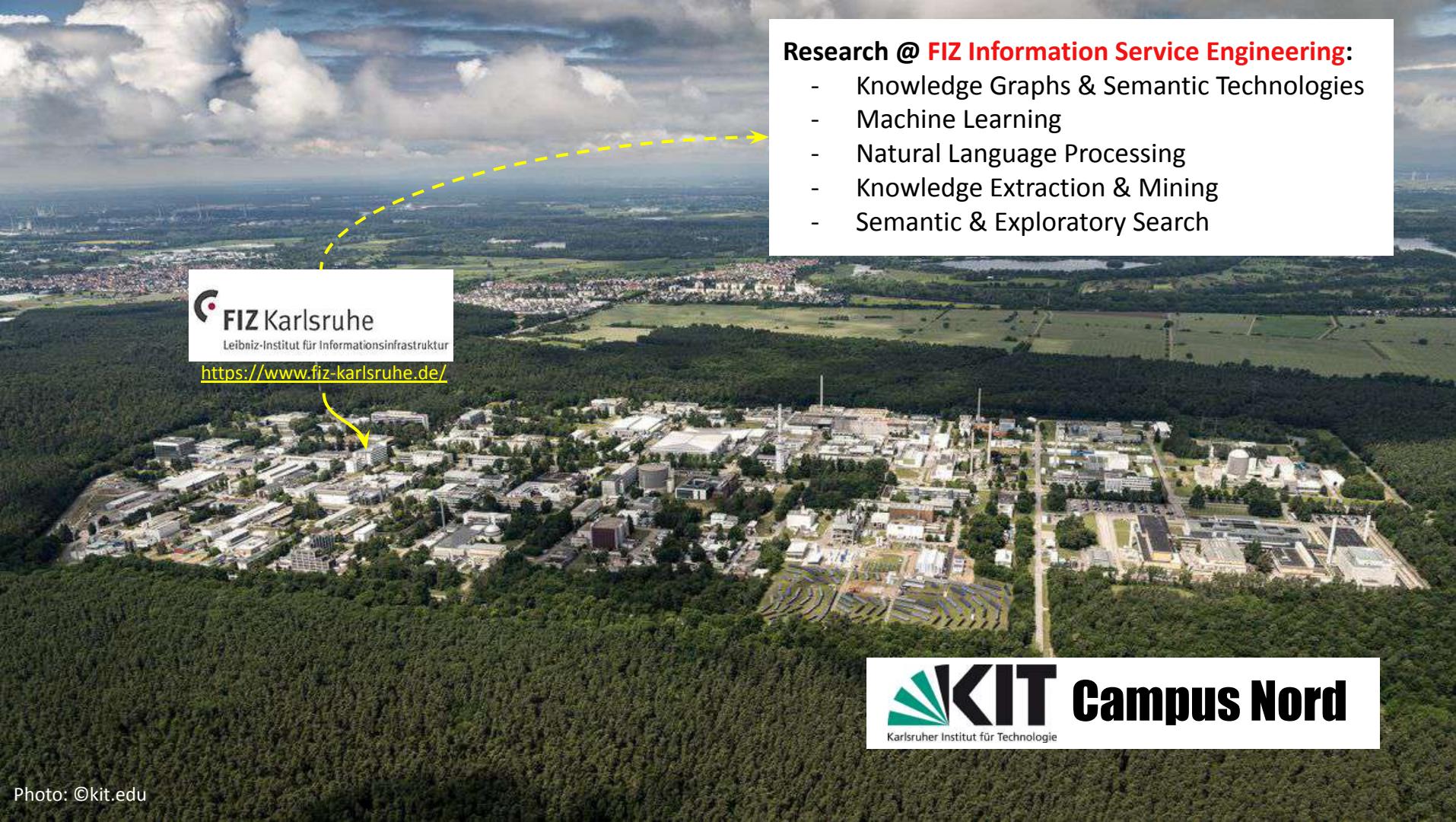
Leibniz Institute for Information Infrastructure

- Professor of *Information Service Engineering*  
**FIZ Karlsruhe – Leibniz Institute for Information Infrastructure & AIFB, Karlsruhe Institute of Technology (KIT)**



- [Homepage FIZ](#)  
[@lysander07@sigmoid.social](mailto:@lysander07@sigmoid.social)  
[@fizise@sigmoid.social](mailto:@fizise@sigmoid.social)  
[harald.sack@kit.edu](mailto:harald.sack@kit.edu)





**FIZ** Karlsruhe

Leibniz-Institut für Informationsinfrastruktur

<https://www.fiz-karlsruhe.de/>

## Research @ **FIZ Information Service Engineering:**

- Knowledge Graphs & Semantic Technologies
- Machine Learning
- Natural Language Processing
- Knowledge Extraction & Mining
- Semantic & Exploratory Search

 **KIT** Campus Nord

Karlsruher Institut für Technologie

# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

## Lecture Overview

### 1. Knowledge Representation with Graphs

- 1.1 From Numbers to Insight
- 1.2 Knowledge and how to represent it
- 1.3 Graphs and Triples
- 1.4 Knowledge Graphs

# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

## Lecture Overview

## 2. Basic Knowledge Graph Infrastructure

- 2.1 The Semantic Web, Linked Data and the Web of Data
- 2.2 How to Identify and Access Things
- 2.3 How to Represent Facts with RDF
- 2.4 Vocabularies and Model Building with RDFS

# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

## Lecture Overview

### 3. Querying Knowledge Graphs with SPARQL

- 3.1 How to Query RDF(S)
- 3.2 DBpedia Knowledge Graph
- 3.3 Wikidata Knowledge Graph
- 3.4 Why RDF(S) is not sufficient

# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

## Lecture Overview

### 4. Ontologies as Key to Knowledge Representation

- 4.1 From Aristotle to AI: Exploring Ontologies in Computer Science
- 4.2 The Web Ontology Language OWL
- 4.3 How to design your own Ontology
- 4.4 Knowledge Graph Construction
- 4.5 Ontologies & Knowledge Graphs – Best Practices

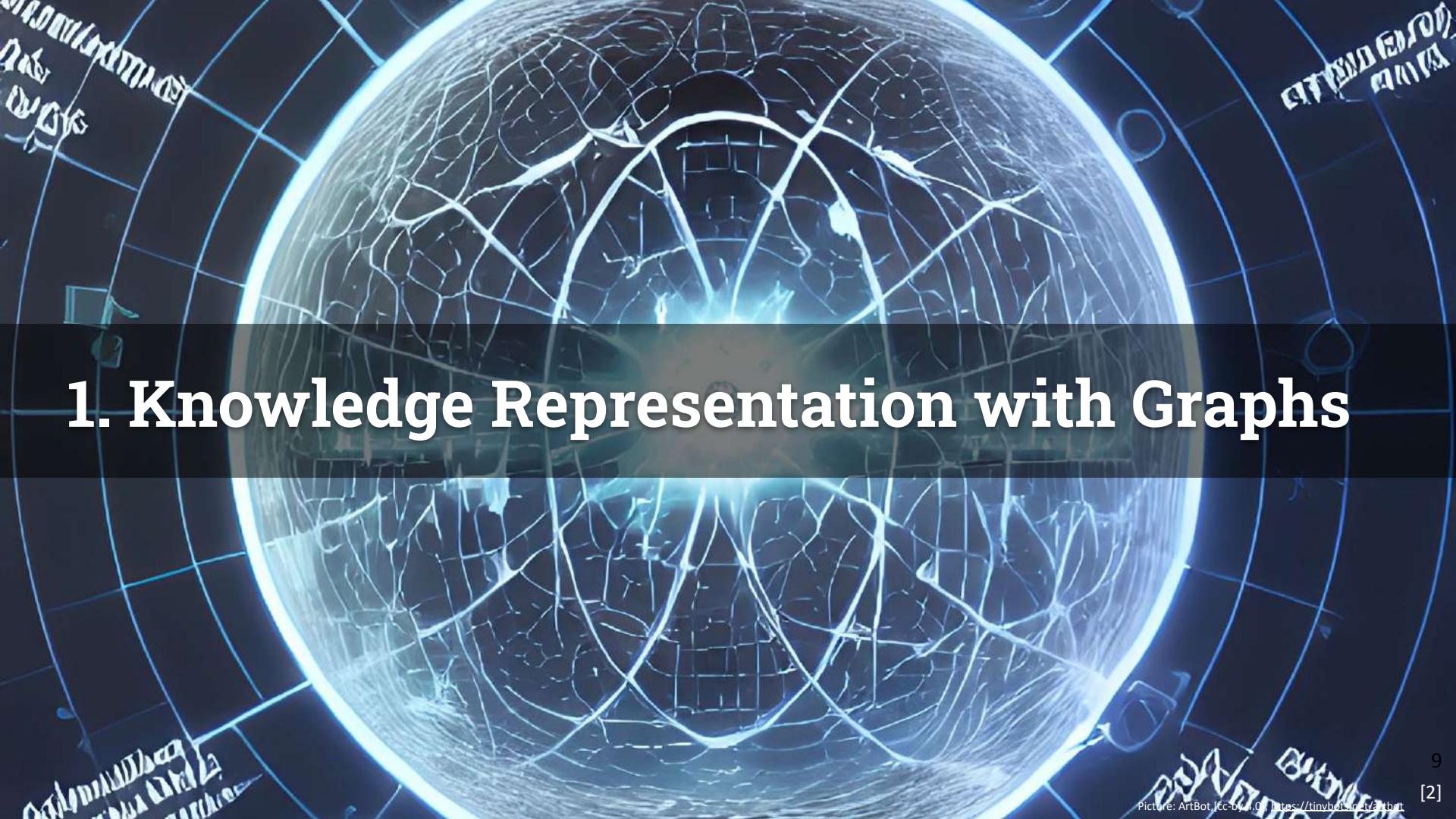
# Knowledge Graphen - Publizieren und forschen mit Wissensgraphen

## Lecture Overview

## 5. Intelligent Applications with Knowledge Graphs and Deep Learning

5.1 Knowledge Graphs and Language Models

5.2 Semantic and Exploratory Search and Recommendation



# 1. Knowledge Representation with Graphs

# 42



## What is 42?

42 is a number. It is an integer and a non-negative number. In the decimal system, it is the number following 41 and preceding 43. It is often used as a placeholder for a number or value that is currently unknown or unspecified. The number 42 is also significant in popular culture and has been referenced in many works of fiction, including "The Hitchhiker's Guide to the Galaxy" by Douglas Adams, in which 42 is described as "the answer to the ultimate question of life, the universe, and everything" calculated by an enormous supercomputer named Deep Thought over a period of 7.5 million years.





There are many different interpretations and meanings of the number 42, as it has been referenced in a variety of contexts in different cultures and disciplines. Here are a few examples:



- In the Tarot, the number 42 is associated with the card "The World," which represents completion, wholeness, and the integration of all aspects of the self.
- In numerology, the number 42 is considered to be a highly spiritual number that represents the realization of a higher purpose or goal. It is believed to be a powerful number that can help people overcome challenges and achieve their goals.
- In the Chinese culture, the number 42 is considered to be unlucky because it is a homophone for "death" in Mandarin.
- In the Bible, the number 42 appears several times, including in the story of the Flood, where it is said that the rain lasted for 42 days and 42 nights.
- In sports, the number 42 is retired by Major League Baseball in honor of Jackie Robinson, who was the first African American player to play in the major leagues.

a quantity ?  
a point in time ?  
a time span?  
a length?  
a weight?  
a measurement?  
a code ?  
a character string?

42



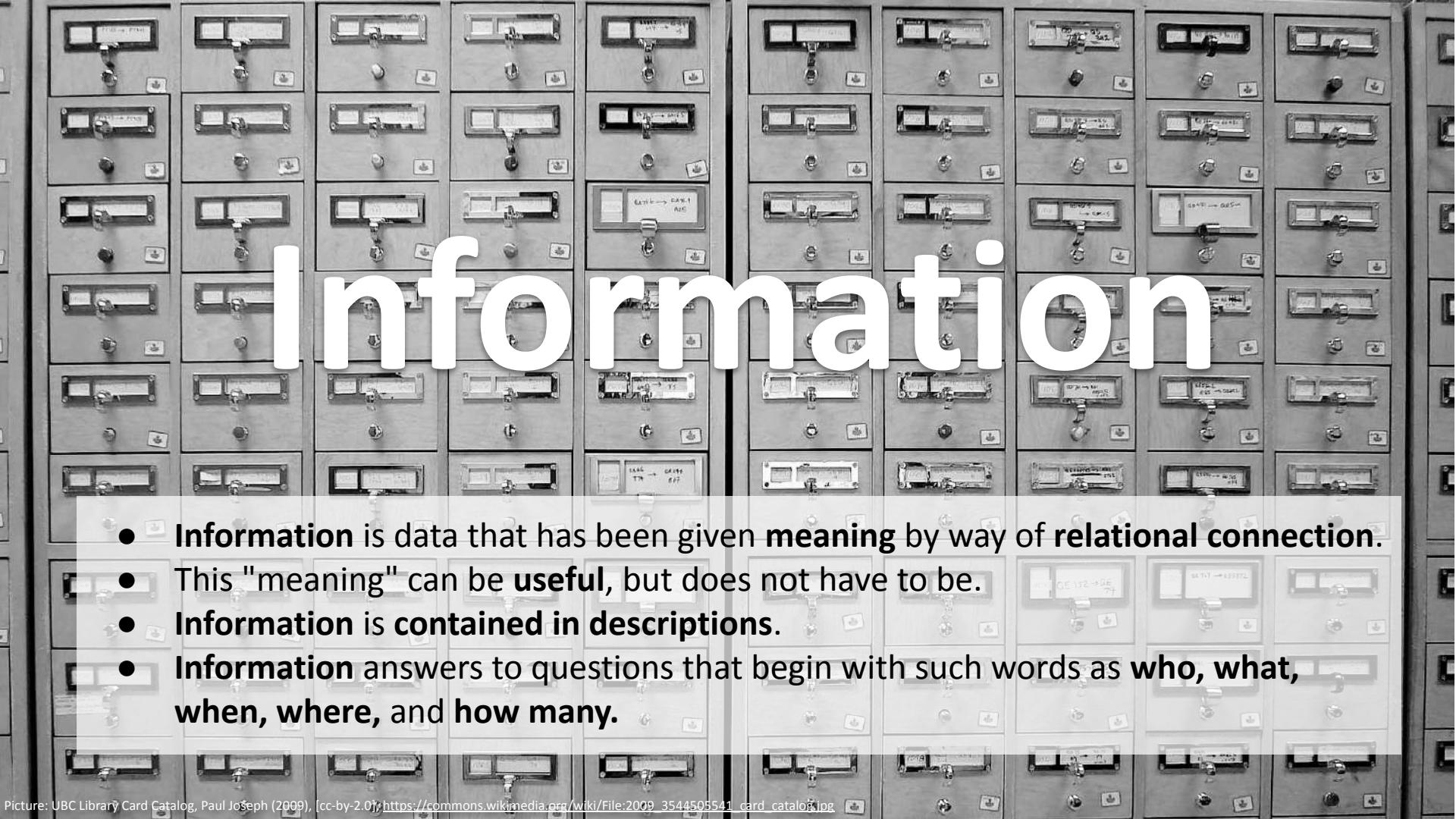
We need  
more information  
to make sense of  
the data

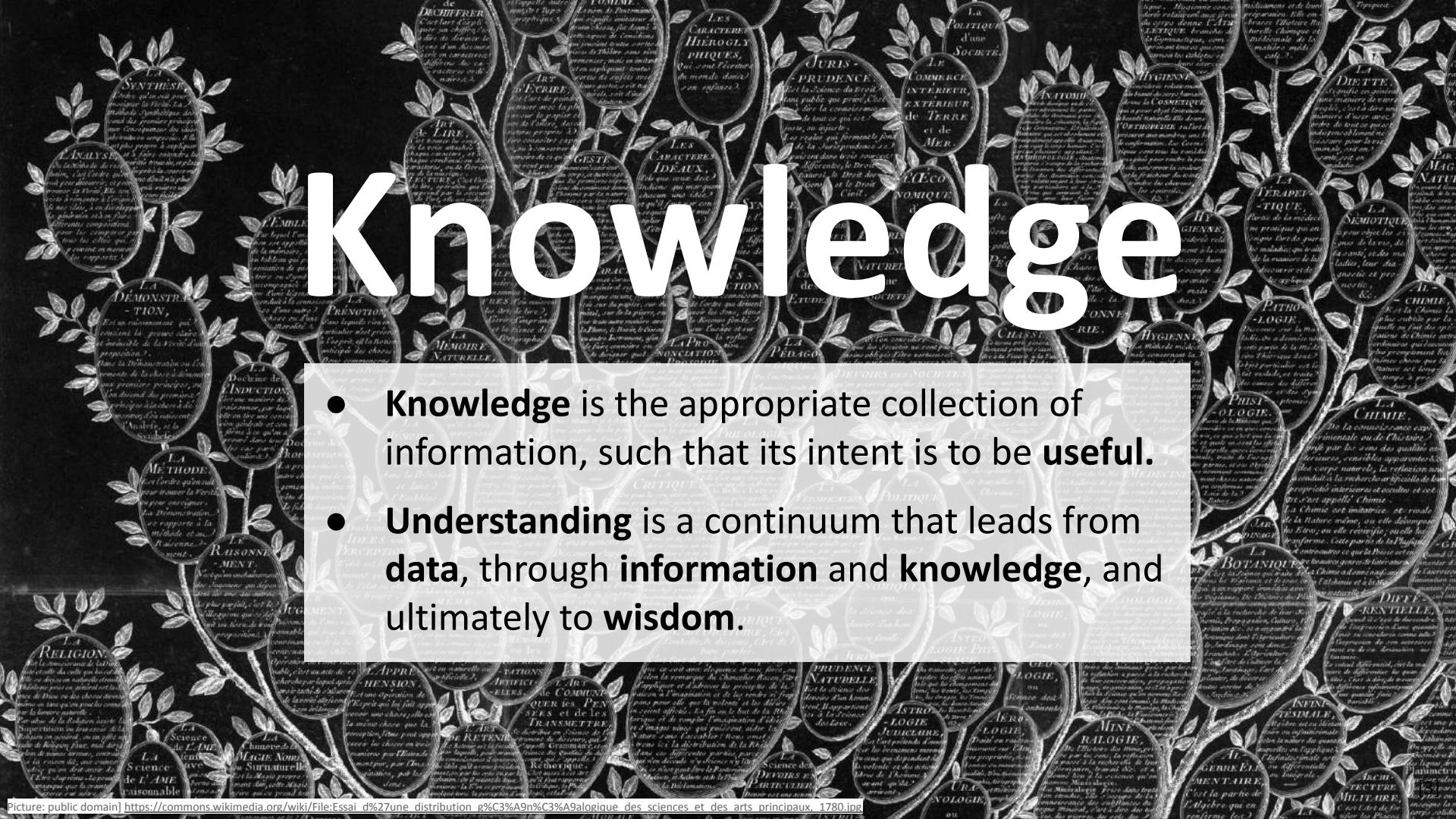
# Data

- Data is raw.
- It simply exists and has no significance beyond its existence (in and of itself).
- It can exist in any form, usable or not.

# Information

- **Information** is data that has been given **meaning** by way of **relational connection**.
- This "meaning" can be **useful**, but does not have to be.
- **Information is contained in descriptions**.
- **Information** answers to questions that begin with such words as **who, what, when, where, and how many**.



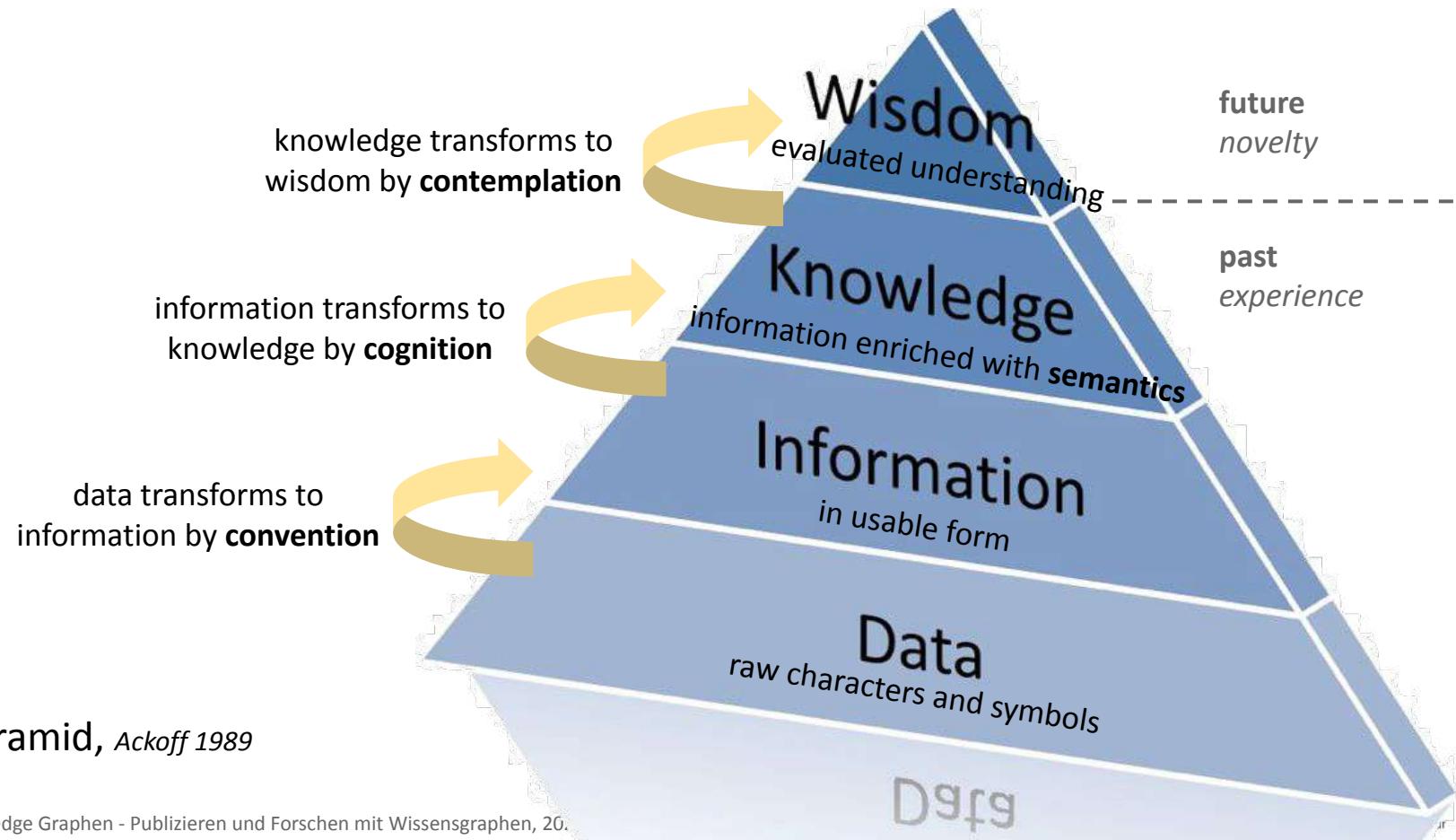
- 
- # Knowledge
- Knowledge is the appropriate collection of information, such that its intent is to be **useful**.
  - Understanding is a continuum that leads from **data**, through **information** and **knowledge**, and ultimately to **wisdom**.



# Wisdom

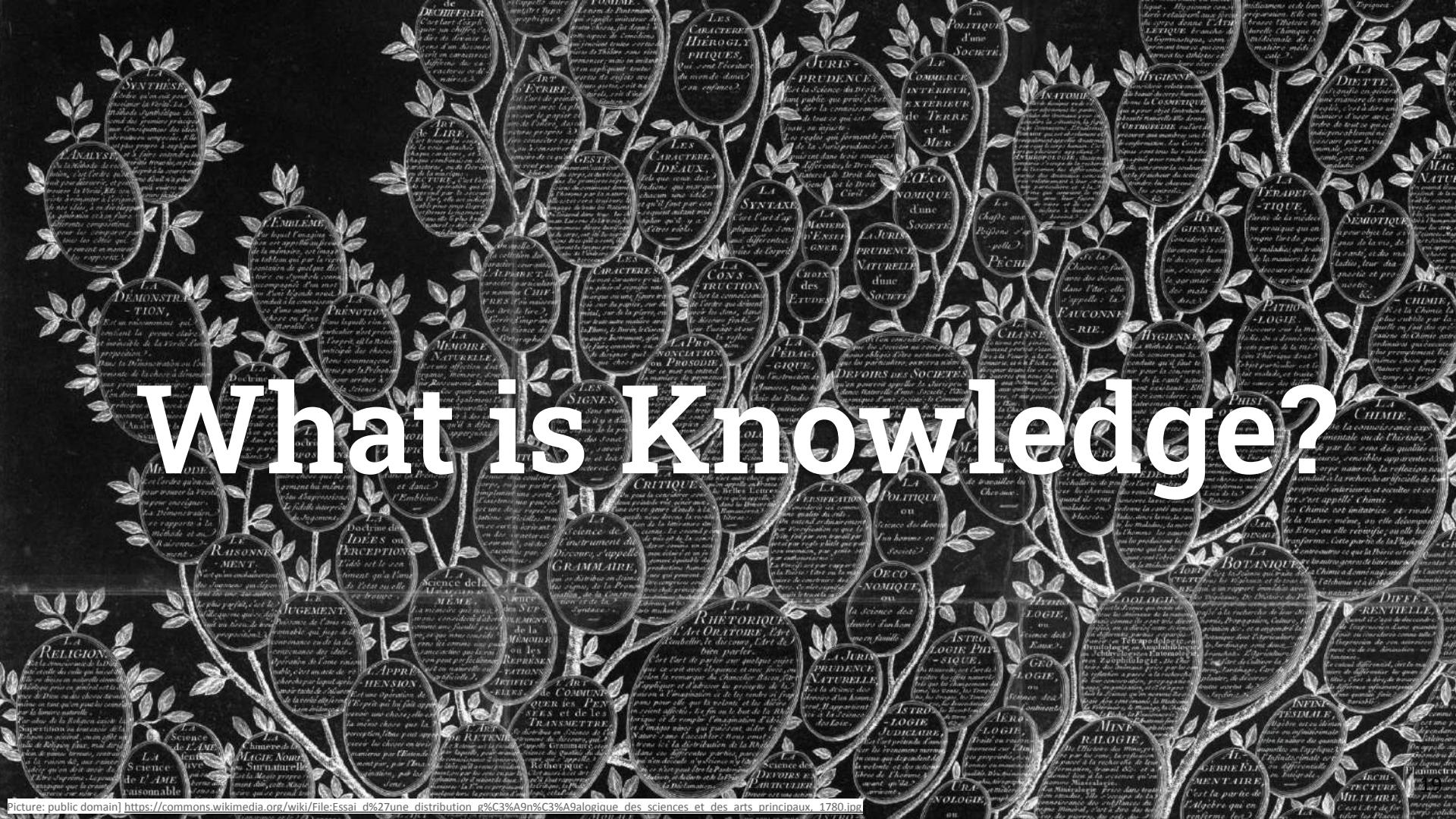
- Wisdom is the ability to make sound judgments and decisions.
- Data transforms to **information** by convention, **information** to **knowledge** by cognition, and **knowledge** to **wisdom** by contemplation.

# Data, Information, Knowledge

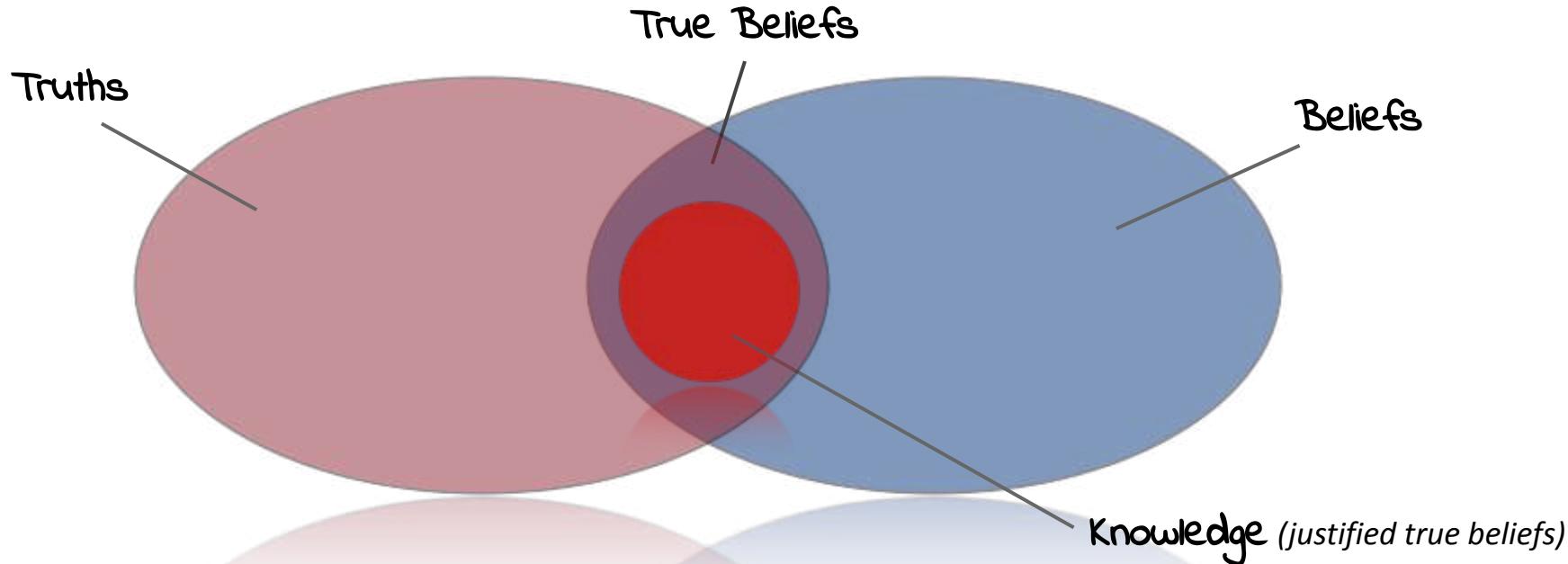


DIKW Pyramid, Ackoff 1989

# What is Knowledge?



# What is Knowledge?

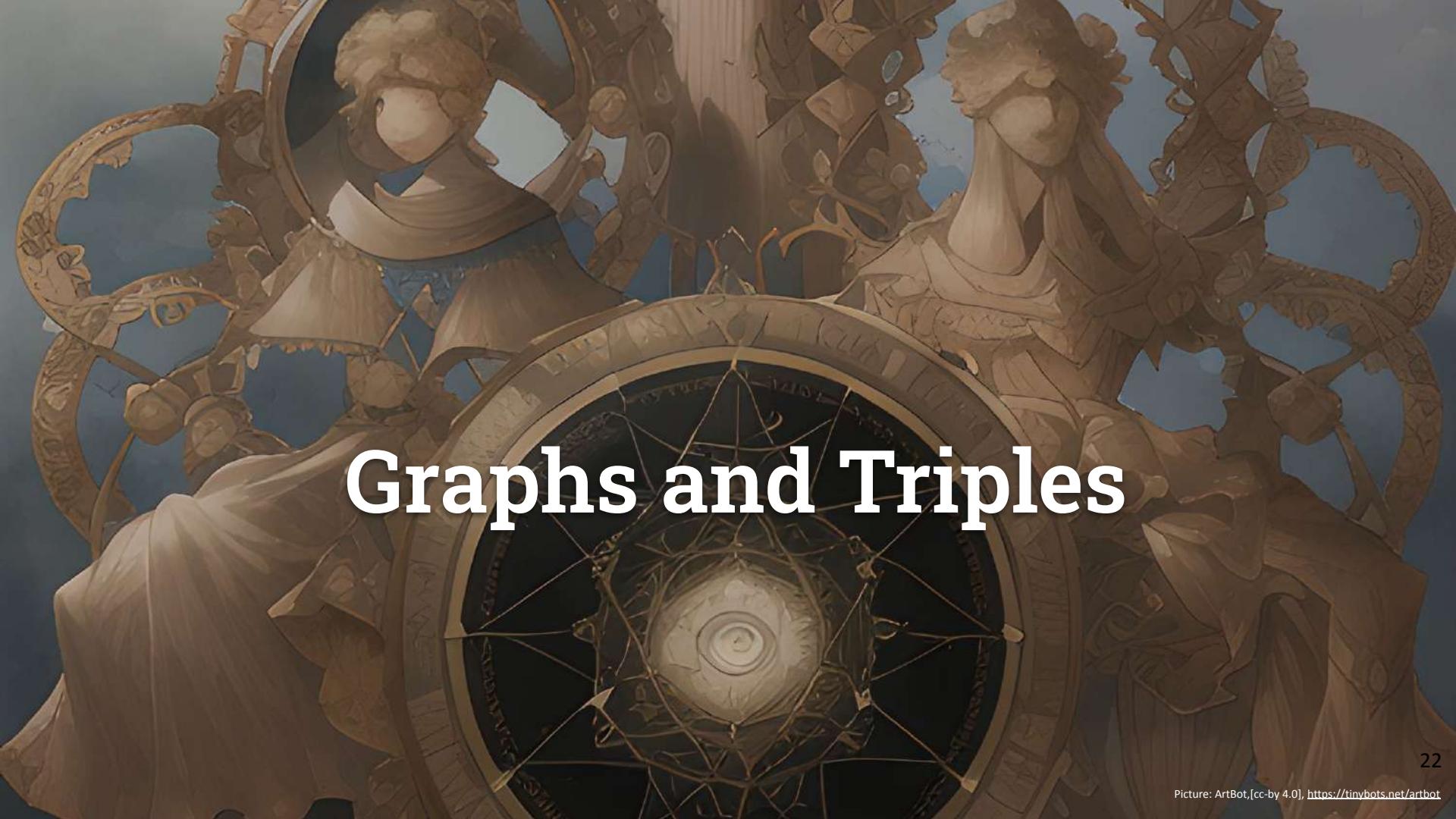


Traditional Definition: „*Knowledge is a justified subset of all true beliefs*“

To represent knowledge, we need a (formal) **knowledge representation**.



„People can't share knowledge if they don't speak a common language“



# Graphs and Triples



I am not Spock

Leonard Nimoy played Spock.  
Spock is a character in Star Trek.  
Leonard Nimoy starred in Star Trek.

# An Intuitive Way to Represent Knowledge

Leonard Nimoy played Spock.

Subject

Predicate

Object

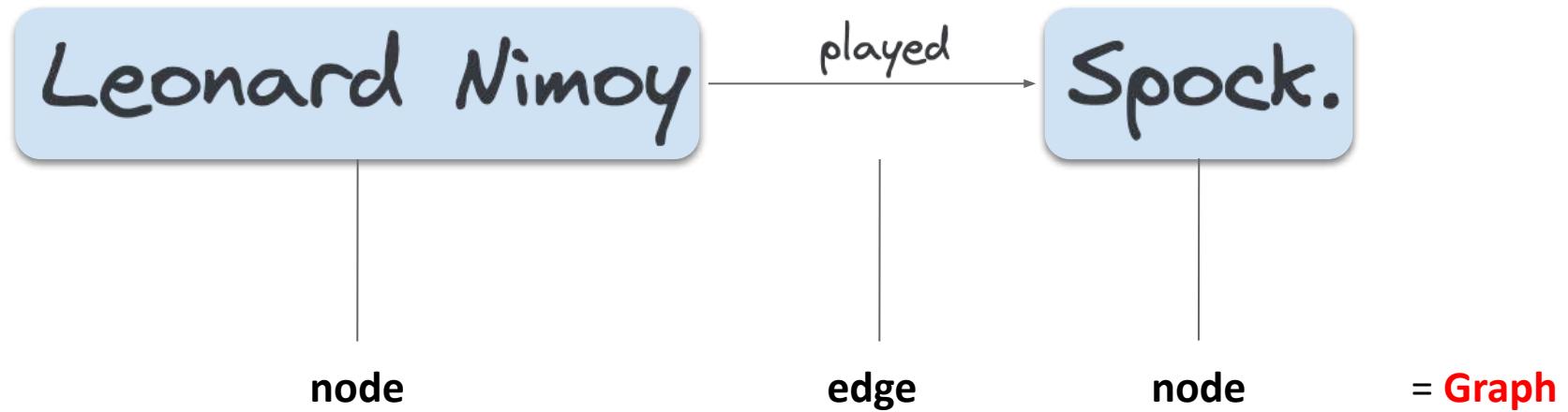
= **Triple**

Leonard Nimoy played Spock.

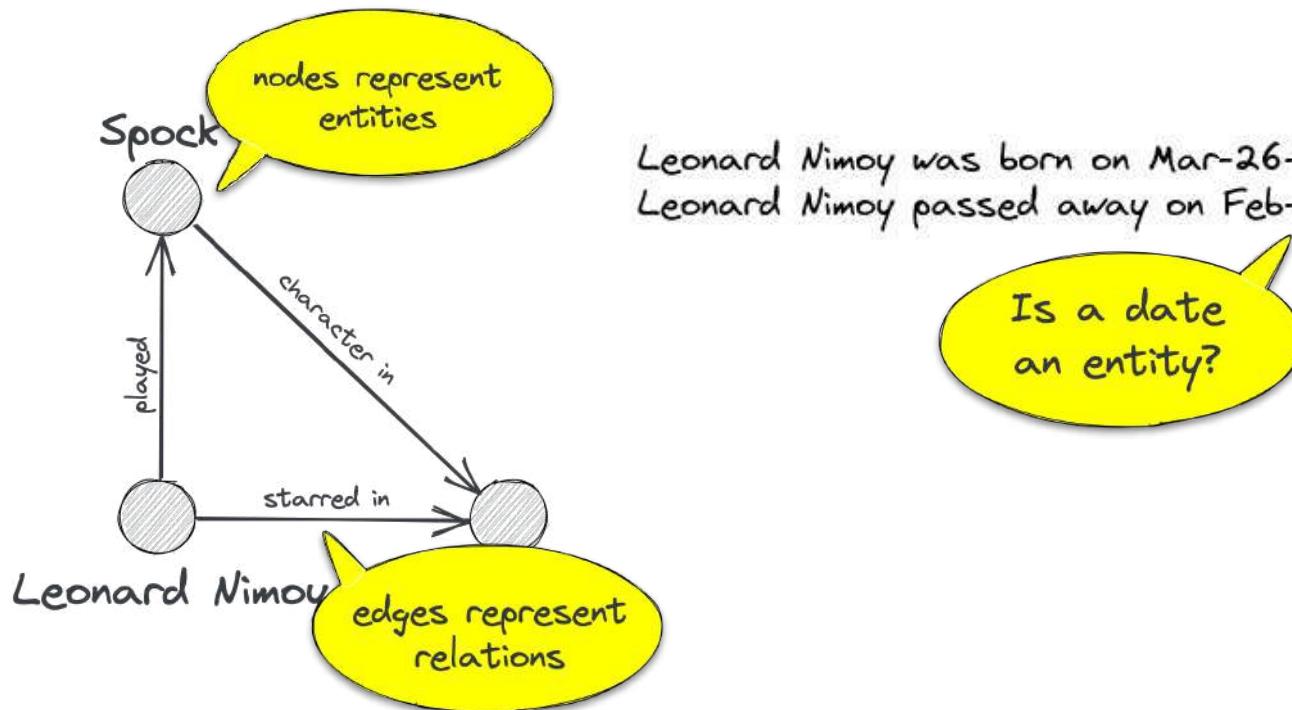
Spock is a character in Star Trek.

Leonard Nimoy starred in Star Trek.

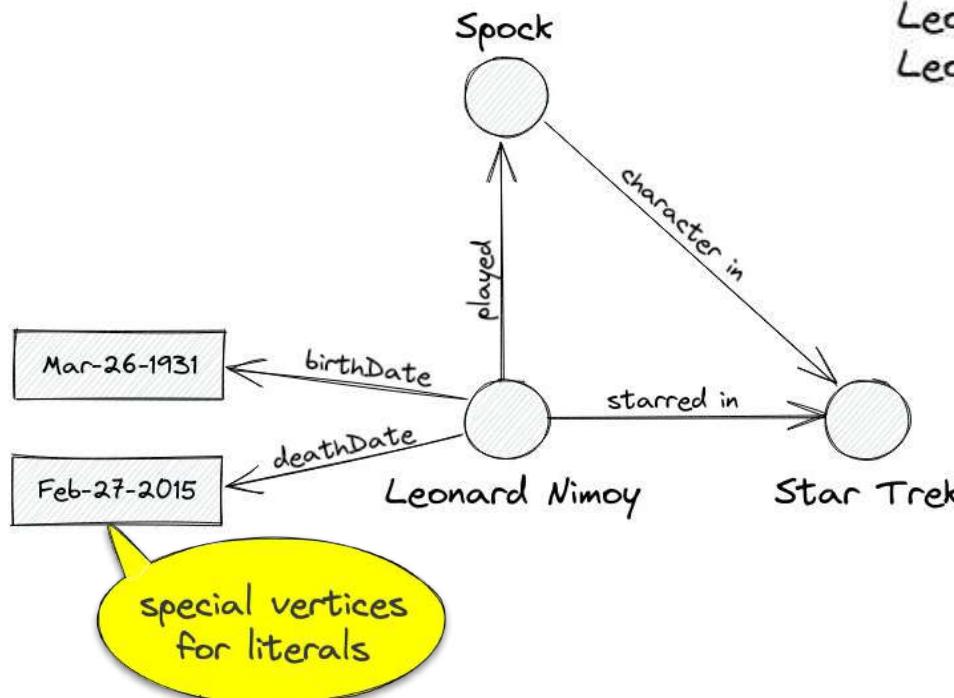
# An Intuitive Way to Represent Knowledge



# An Intuitive Way to Represent Knowledge

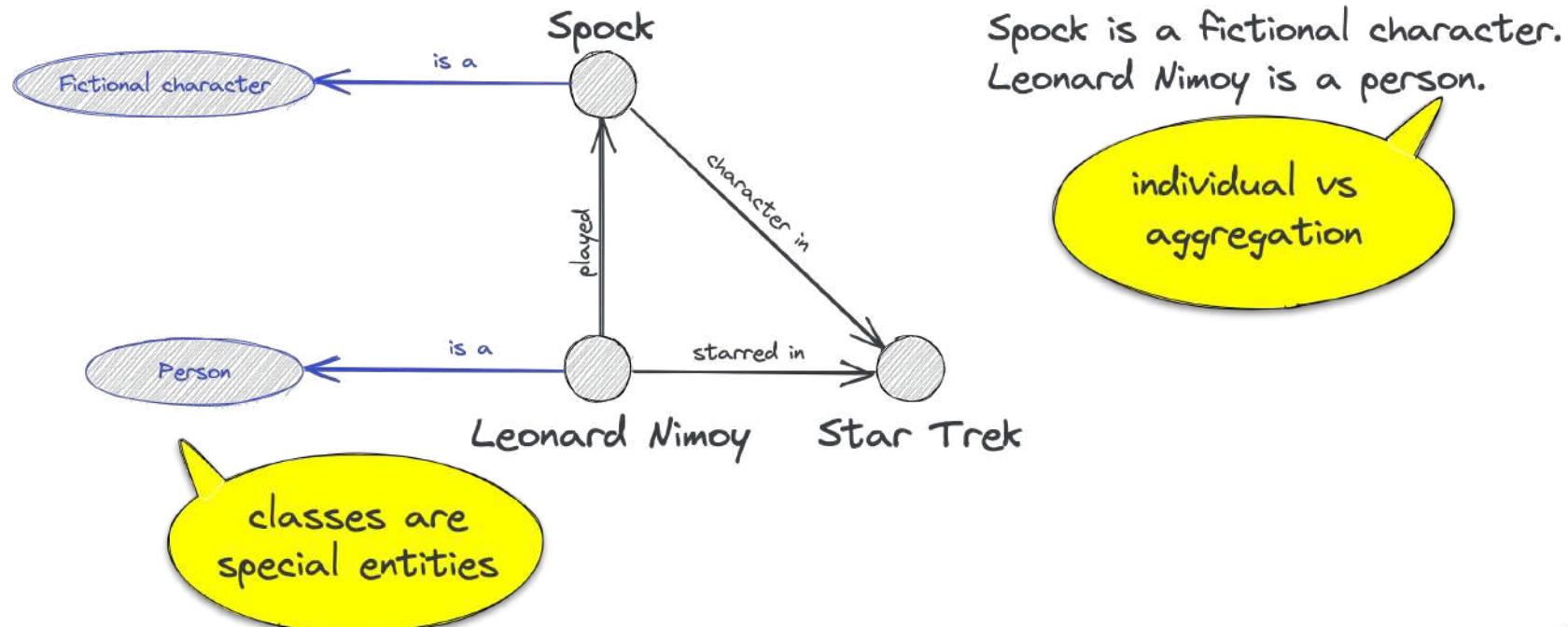


# Further Refinements

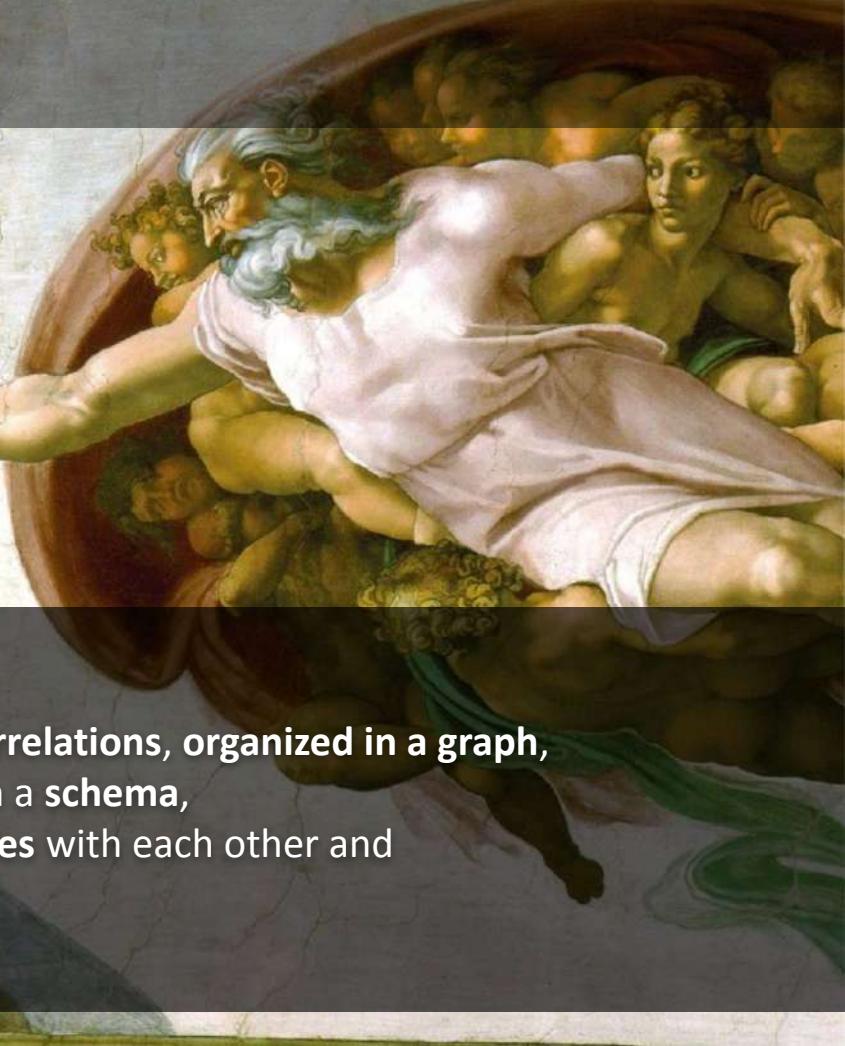
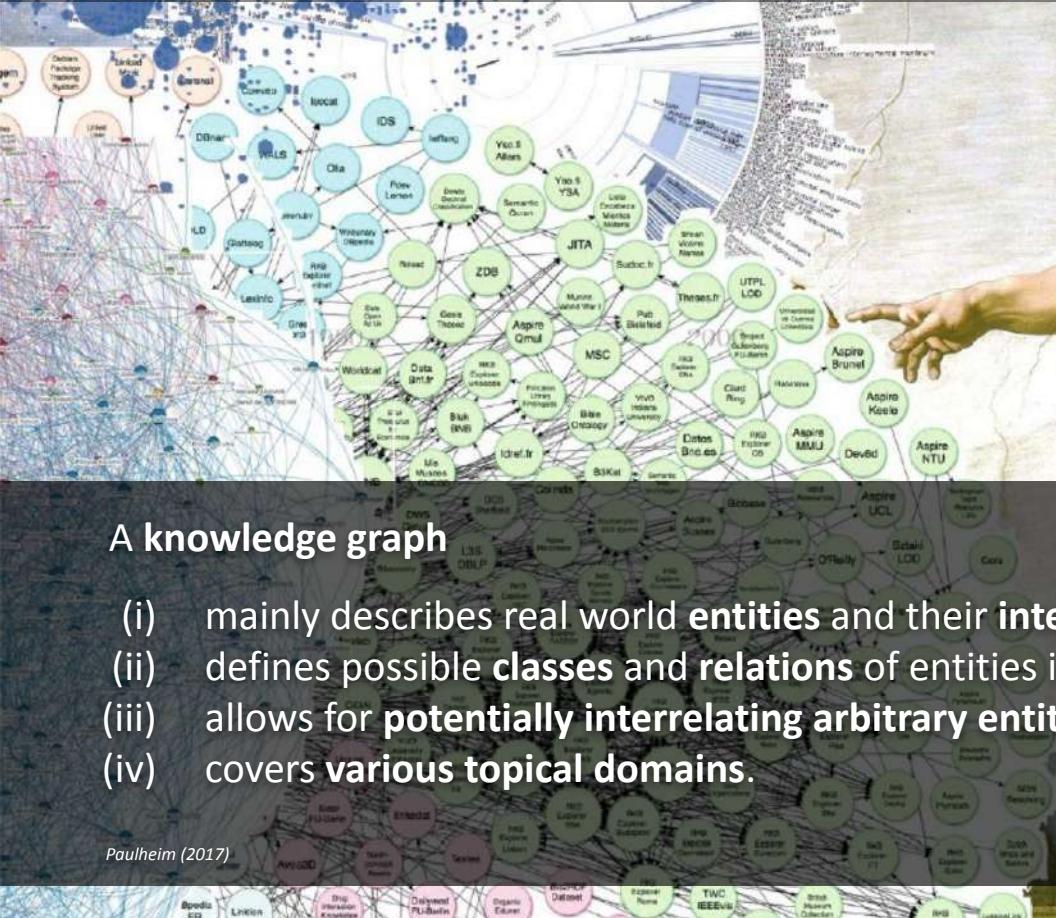


Leonard Nimoy was born on Mar-26-1931.  
Leonard Nimoy passed away on Feb-27-2015.

# Further Refinements



# Knowledge Graphs

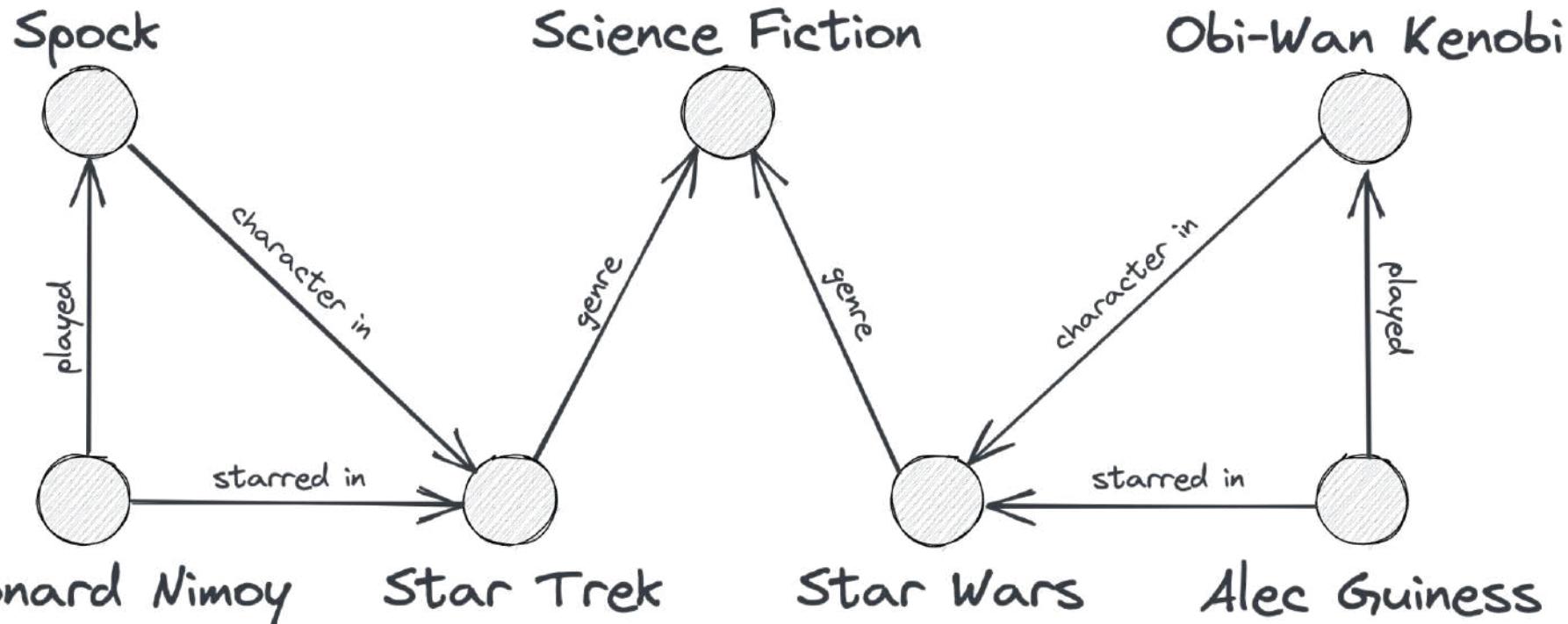


# A knowledge graph

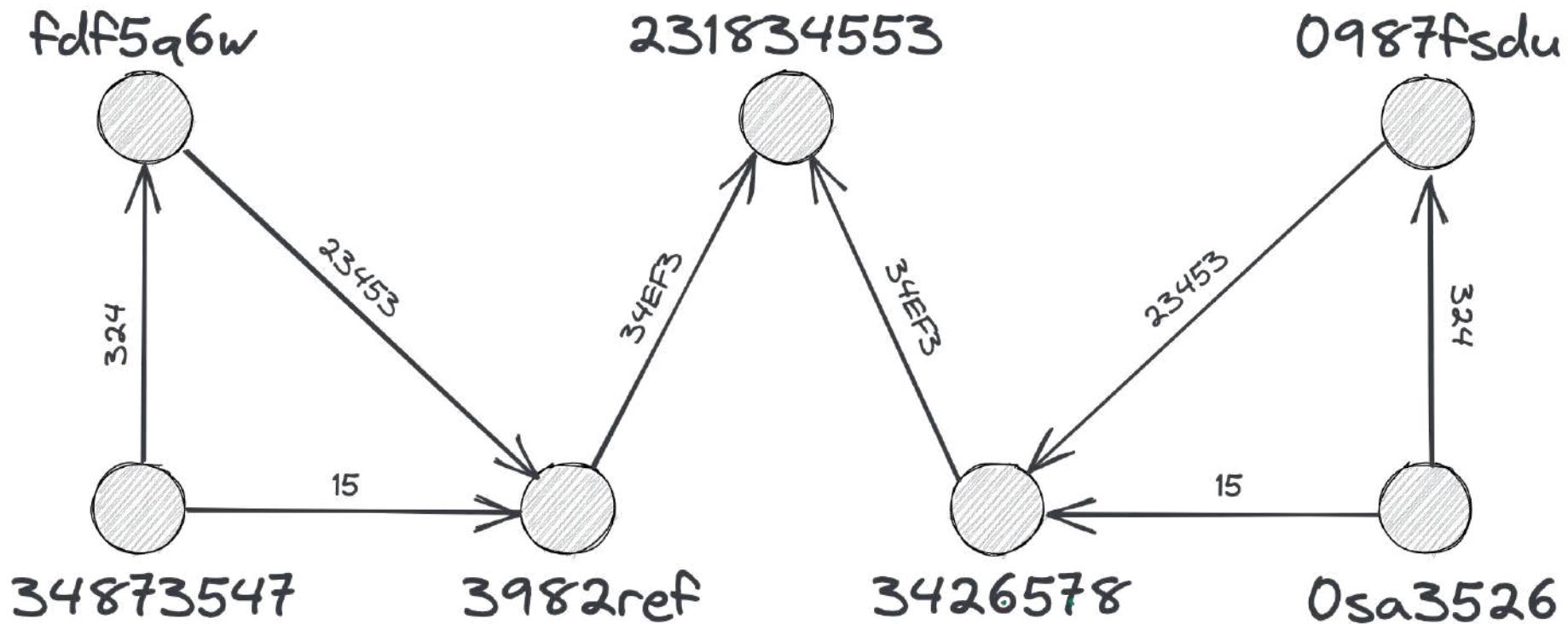
- (i) mainly describes real world **entities** and their **interrelations**, organized in a **graph**,
  - (ii) defines possible **classes** and **relations** of entities in a **schema**,
  - (iii) allows for **potentially interrelating arbitrary entities** with each other and
  - (iv) covers **various topical domains**.

Paulheim (2017)

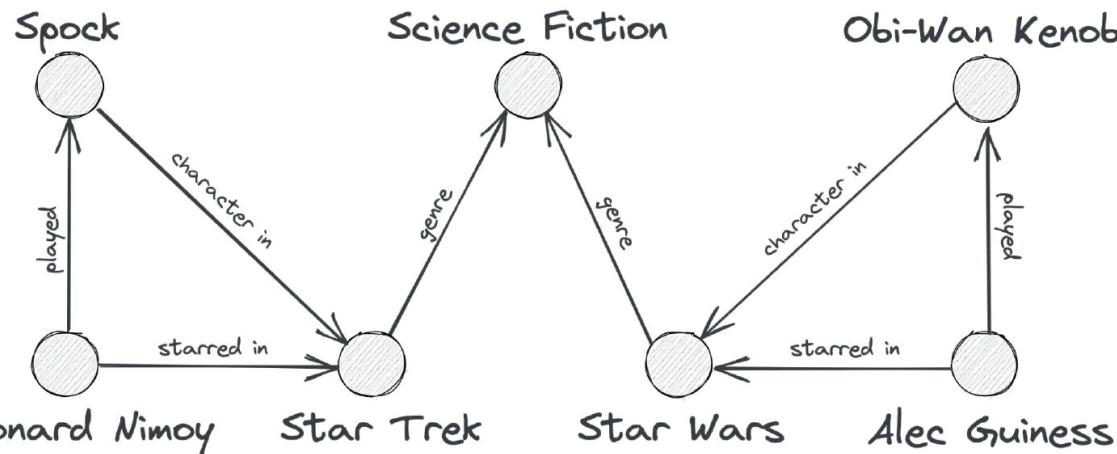
# An Intuitive Way to Represent Knowledge



# How does the Computer know the Meaning?



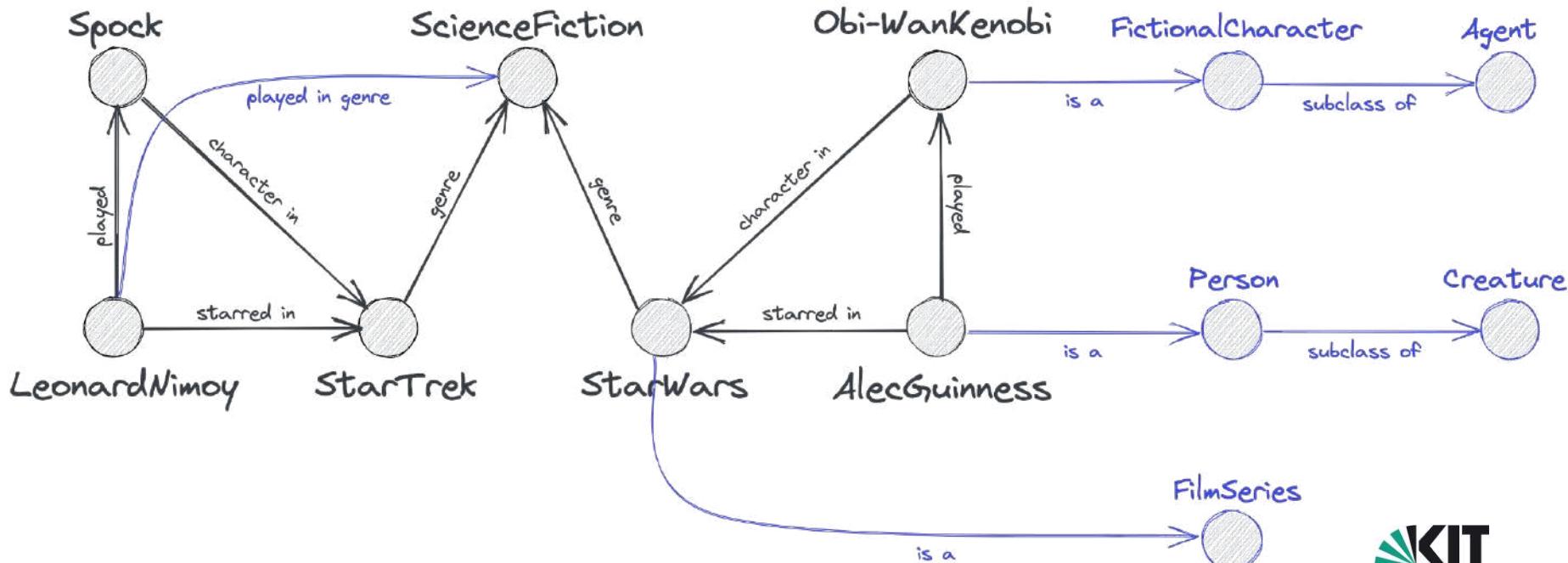
# Let's Create a Knowledge Graph



A **knowledge graph**

- (i) mainly describes real world **entities** and their **interrelations, organized in a graph**,
- (ii) defines possible **classes** and **relations** of entities in a **schema**,
- (iii) allows for **potentially interrelating arbitrary entities** with each other and
- (iv) covers **various topical domains**.

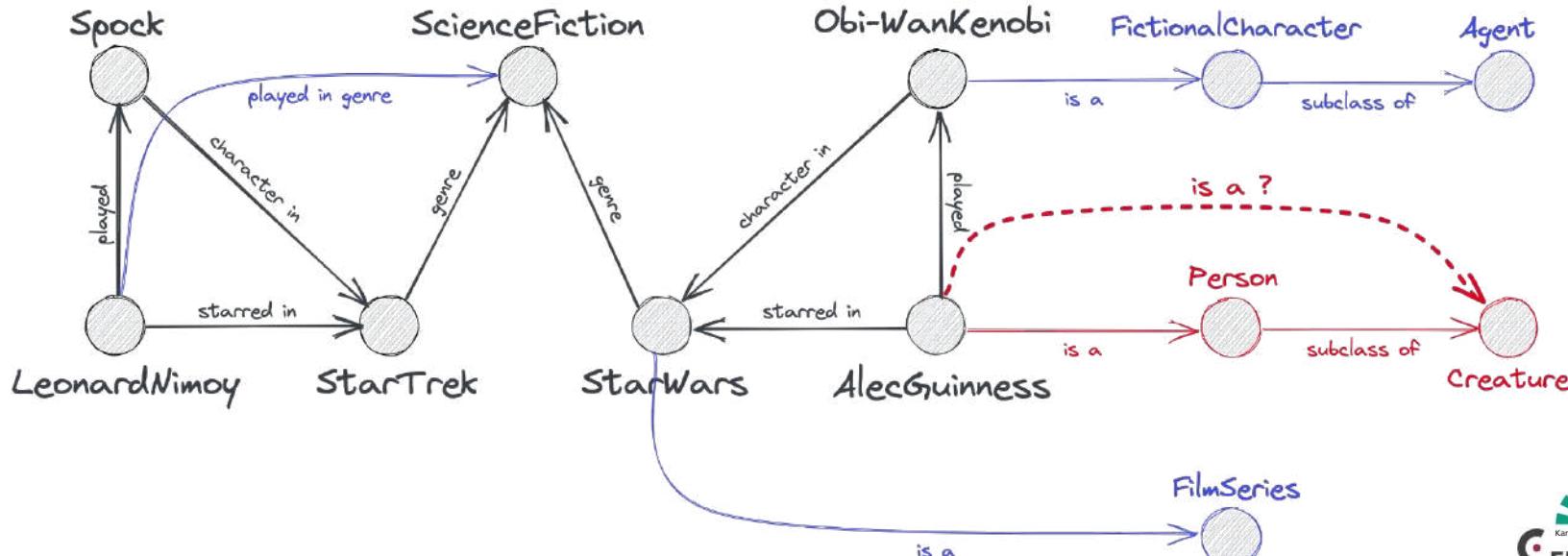
# ...is this already enough?



# Can we deduce that Alec Guinness is a “Creature”?

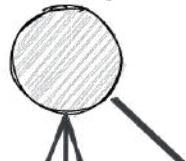
Can be solved via individual software code:

```
IF ((Alec Guinness, is a, Person) AND (Person, subclass of, Creature))
THEN
  (Alec Guinness, is a, Creature)
```



# Back to the Start

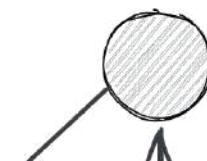
fdf5q6w



231834553

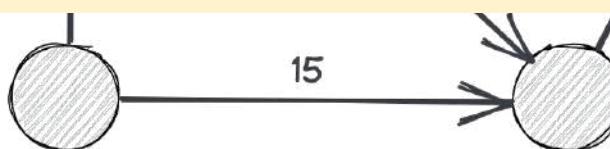


0987fsdu



To represent knowledge explicitly (and to make use of it), we need a knowledge representation that can be “understood” by the computer.

- Ontologies based on Mathematical Logic as “formal” Knowledge Representation.
- Inference Engine (Semantic Reasoner) to draw conclusions and inferences.



34873547

3982ref

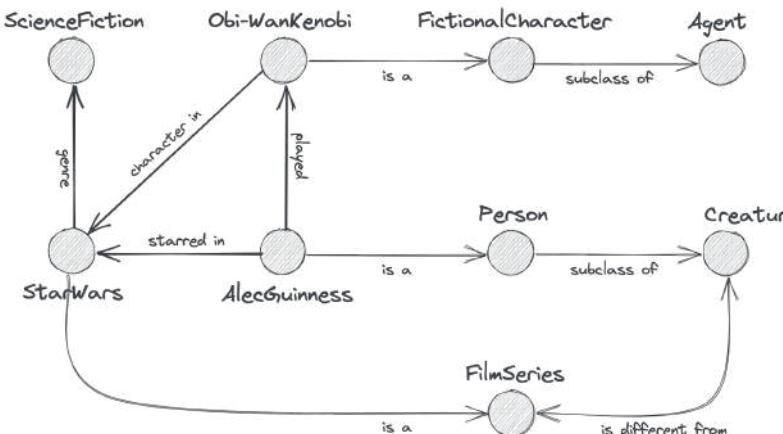


3426578

Osa3526

# A Formal Way to Represent Knowledge

## (Over)Simplified Example



$\text{Obi-WanKenobi} \in \text{FictionalCharacter}$

$\text{AlecGuinness} \in \text{Person}$

$\text{StarWars} \in \text{FilmSeries}$

$\text{FictionalCharacter} \subseteq \text{Agent}$

$\text{Person} \subseteq \text{Creature}$

$\text{FilmSeries} \cap \text{Creature} = \emptyset$

Logical Inference

AlecGuinness  $\in$  Creature  
 AlecGuinness  $\notin$  FilmSeries  
 StarWars  $\notin$  Creature

# Formal Knowledge Representations

What's the difference between **formal knowledge representations** and **traditional data structures**?

1. **Mathematical Logic** provides a framework to **formally express the semantics** of knowledge representations.
2. **Semantics** of knowledge representations can be defined **explicitly**.
3. **Mathematical Logic** enables **logical inferences** and **reasoning** for knowledge representations.

# 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

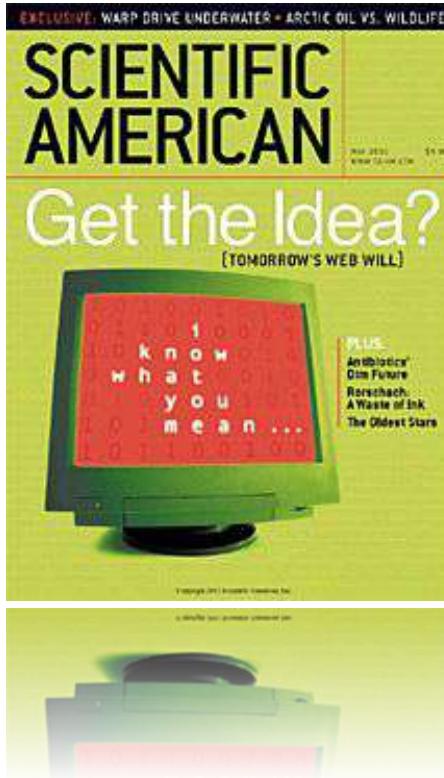
Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small selection of technologies



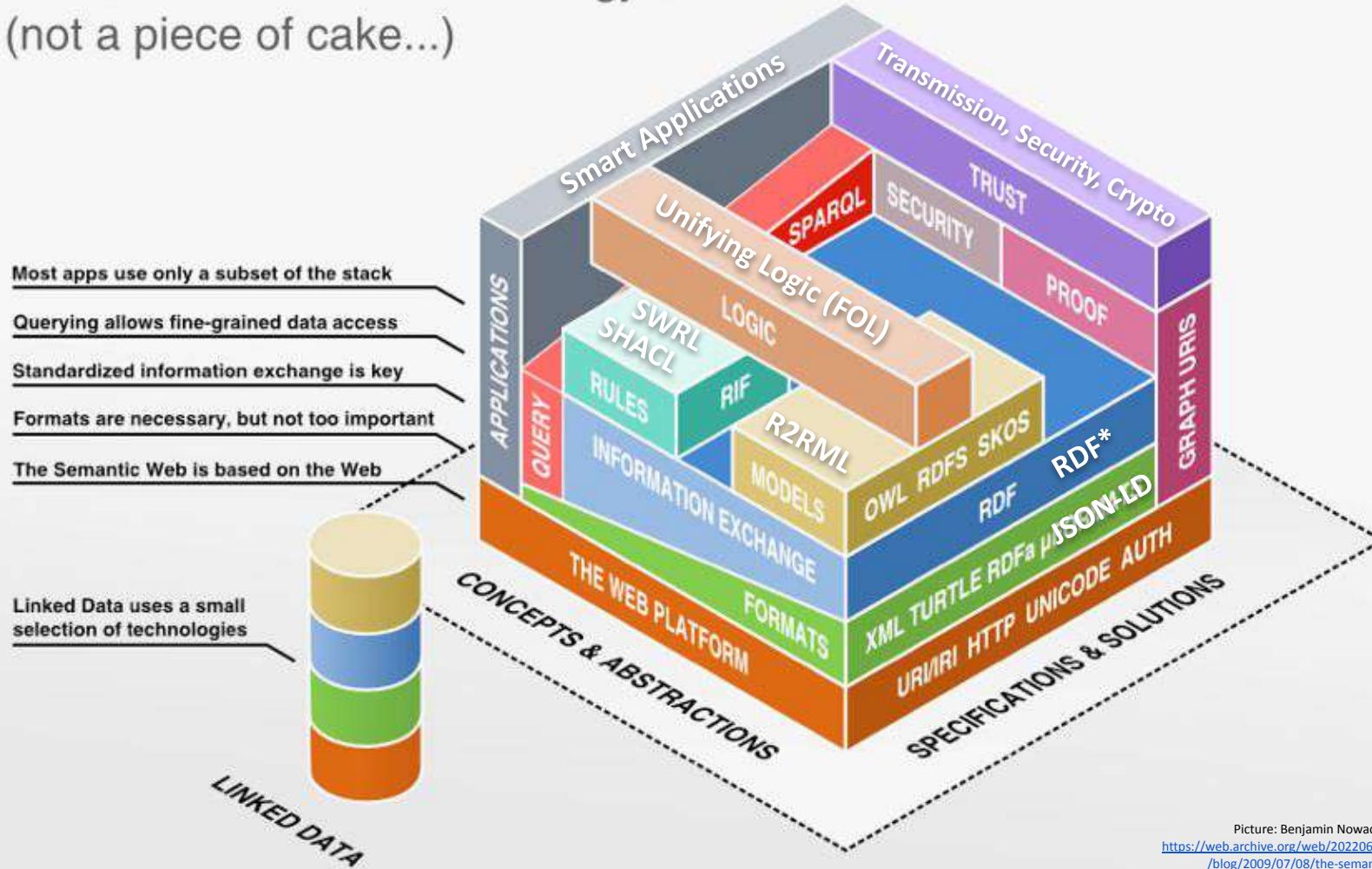
# The Semantic Web – A Web of Data



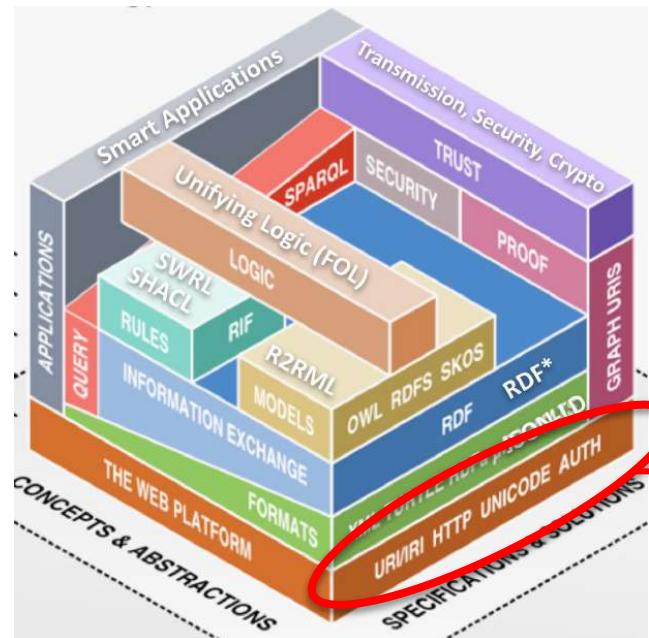
*“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”*

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

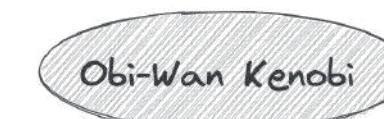
# The Semantic Web Technology Stack (not a piece of cake...)



# The Semantic Web Technology Stack



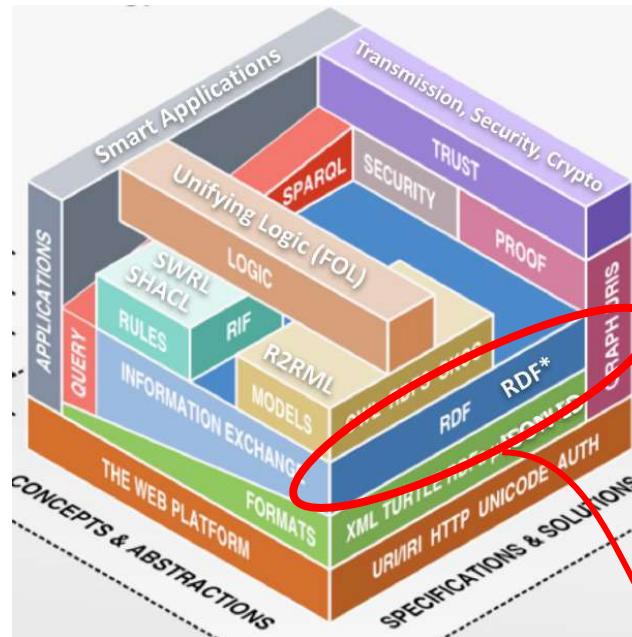
*URI - Uniform Resource Identifier*



= [https://dbpedia.org/resource/Obi-Wan\\_Kenobi](https://dbpedia.org/resource/Obi-Wan_Kenobi)

Picture: Benjamin Nowack, 2009-07-08, [CC BY 3.0],  
<https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>

# The Semantic Web Technology Stack



RDF Triple

[https://dbpedia.org/resource/Obi-Wan\\_Kenobi](https://dbpedia.org/resource/Obi-Wan_Kenobi)

```
:Obi-Wan_Kenobi rdf:type dbo:agent .
:Obi-Wan_Kenobi foaf:name "Obi-Wan Kenobi"@en .
:Obi-Wan_Kenobi dbo:occupation :Jedi .
:Obi-Wan_Kenobi dbo:series :StarWars .
:Obi-Wan_Kenobi dbo:portrayer :Alec_Guinness .
:Obi-Wan_Kenobi dbo:portrayer :Ewan_McGregor .
...
```

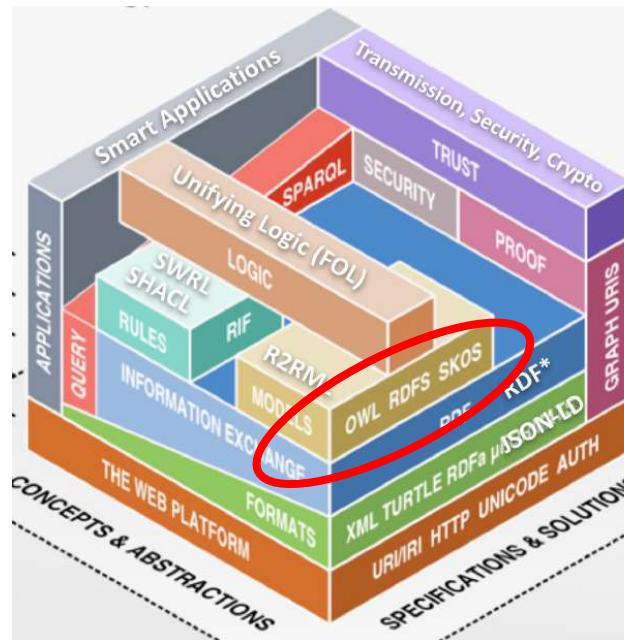


RDF - Resource Description Framework



Picture: Benjamin Nowack, 2009-07-08, [CC BY 3.0],  
[https://web.archive.org/web/20220628120341/http://bnode.org/\\_blog/2009/07/08/the semantic web not a piece of cake](https://web.archive.org/web/20220628120341/http://bnode.org/_blog/2009/07/08/the semantic web not a piece of cake)

# The Semantic Web Technology Stack



<https://dbpedia.org/ontology/Agent>

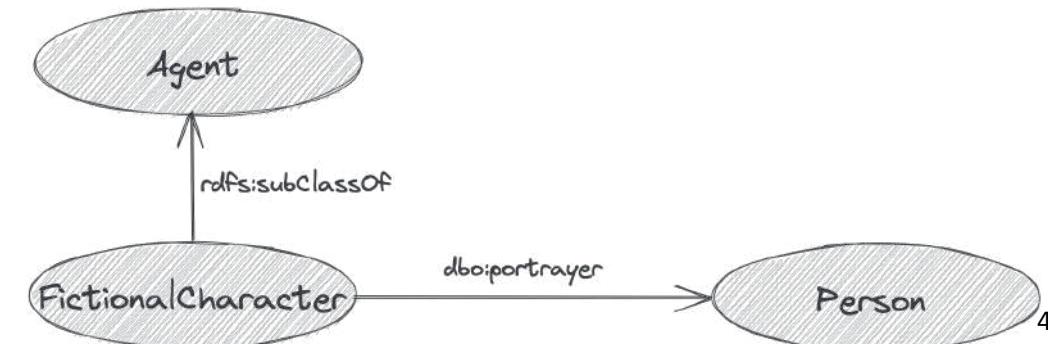
```

dbo:Agent rdf:type owl:Class .
dbo:FictionalCharacter rdfs:subClassOf dbo:Agent .
dbo:portrayer rdf:type rdf:Property .
dbo:portrayer rdfs:domain dbo:FictionalCharacter .
dbo:portrayer rdfs:range dbo:Person .
...

```

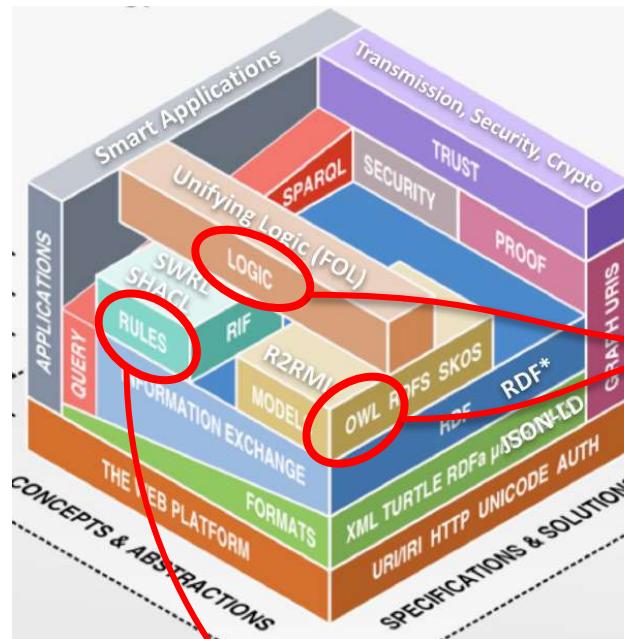


RDFS - RDF Schema

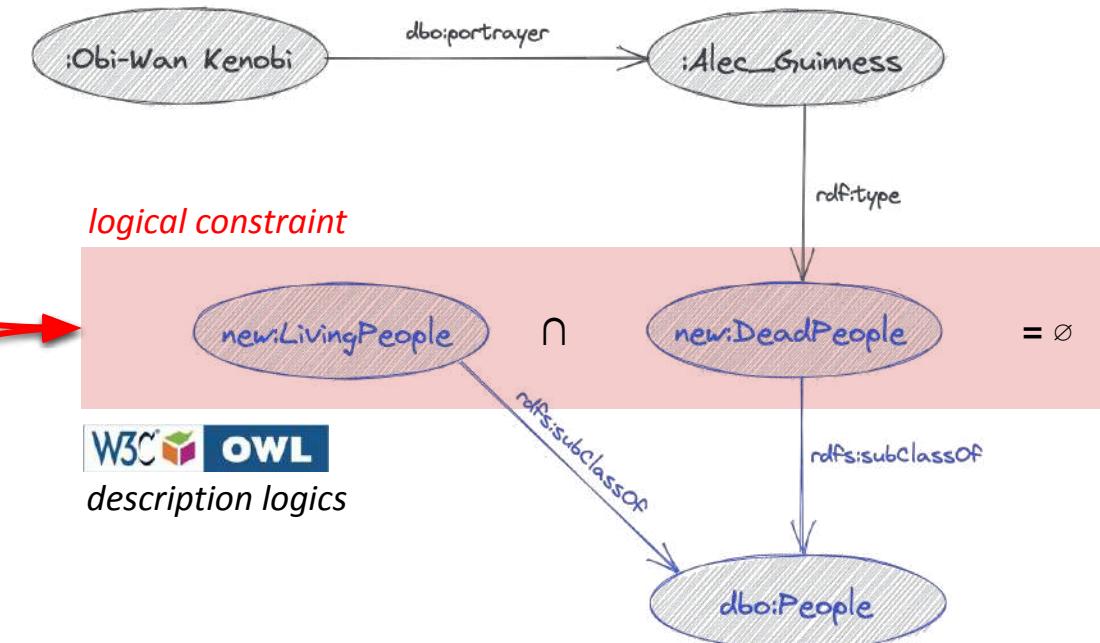


Picture: Benjamin Nowack, 2009-07-08, [CC BY 3.0],  
[https://web.archive.org/web/20220628120341/http://bnode.org/\\_blog/2009/07/08/the semantic web not a piece of cake](https://web.archive.org/web/20220628120341/http://bnode.org/_blog/2009/07/08/the semantic web not a piece of cake)

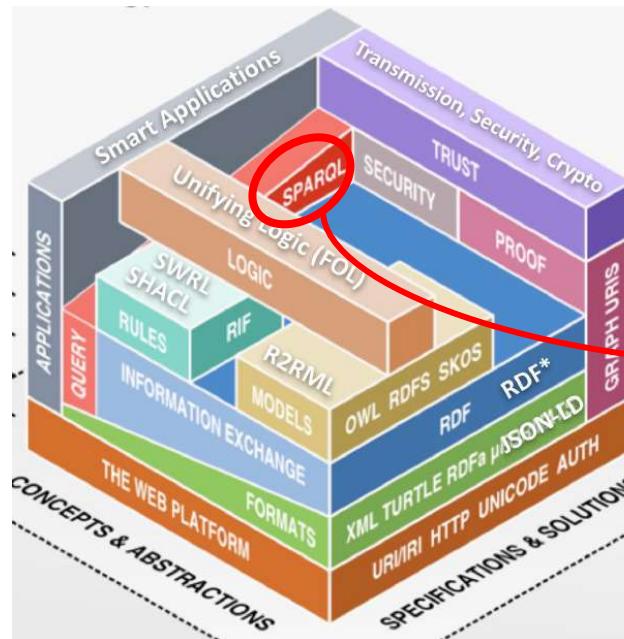
# The Semantic Web Technology Stack



+ 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



On which locations did the actors who portrayed Obi-Wan Kenobi shoot their movies?



```

SELECT ?actor ?actorLabel ?movie ?movieLabel ?coord ?image
WHERE
{
    wd:Q51740 wdt:P175 ?actor .
    ?movie wdt:P161 ?actor ;
           wdt:P915 ?place .
    ?place wdt:P625 ?coord .
    OPTIONAL{?movie wdt:P18 ?image}
    SERVICE wikibase:label
        {bd:serviceParam wikibase:language "en, de, fr, es, it"}
}
  
```

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

Picture: Benjamin Nowack, 2009-07-08, [CC BY 3.0],  
<https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>

# The Semantic Web Technology Stack

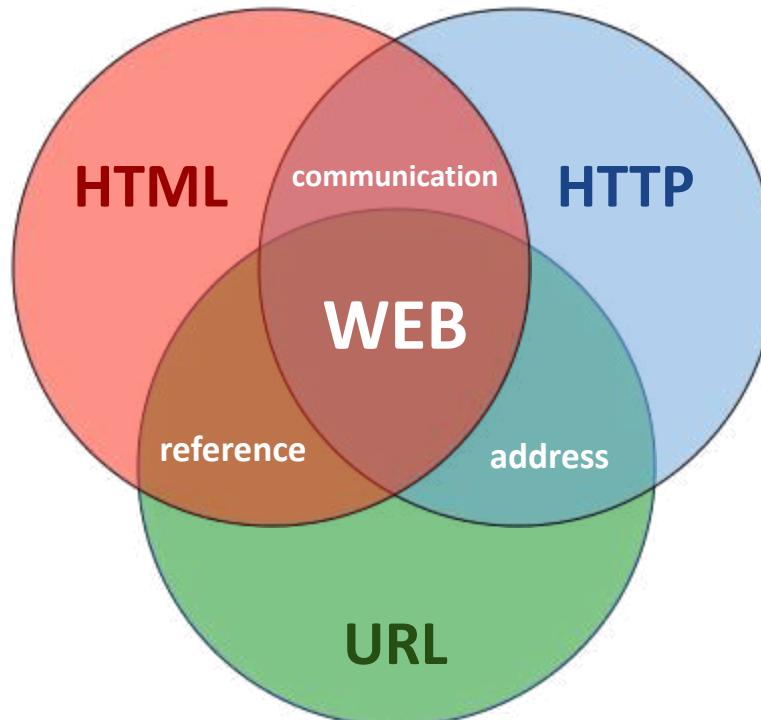


On which locations did the actors who portrayed Obi-Wan Kenobi shoot their movies?



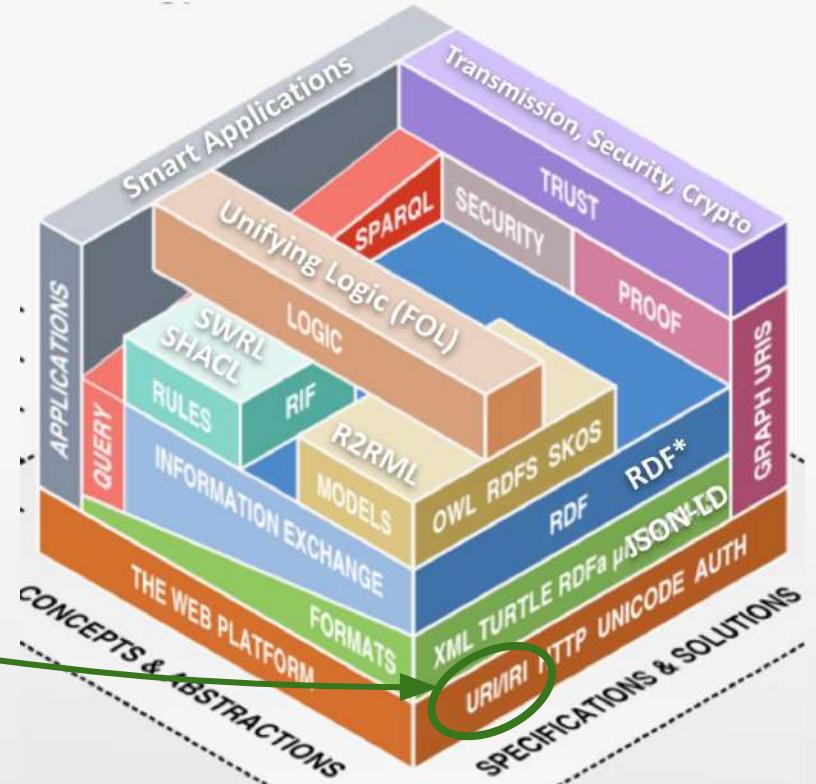
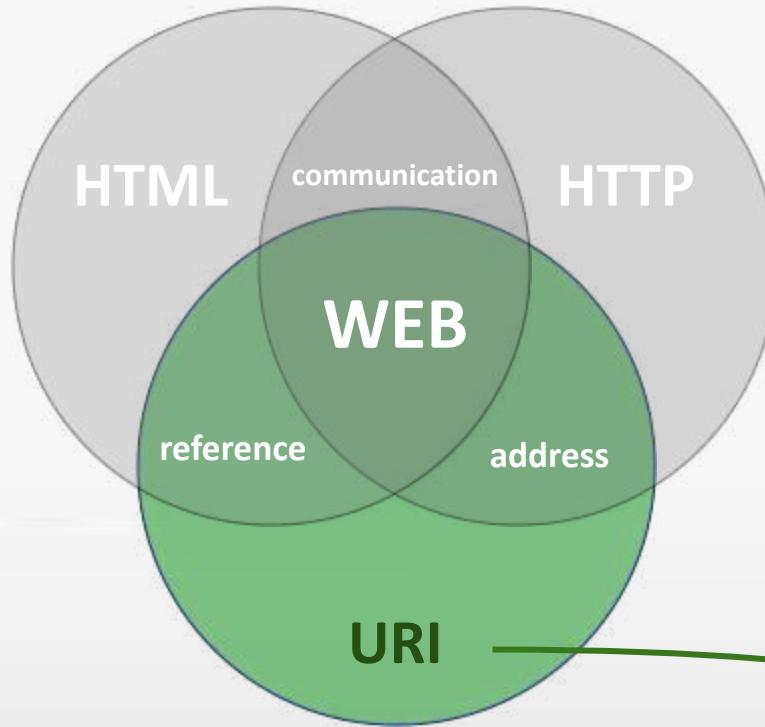
[try SPARQL query at public Wikidata SPARQL endpoint](#)

# The Basic Architecture of the Web

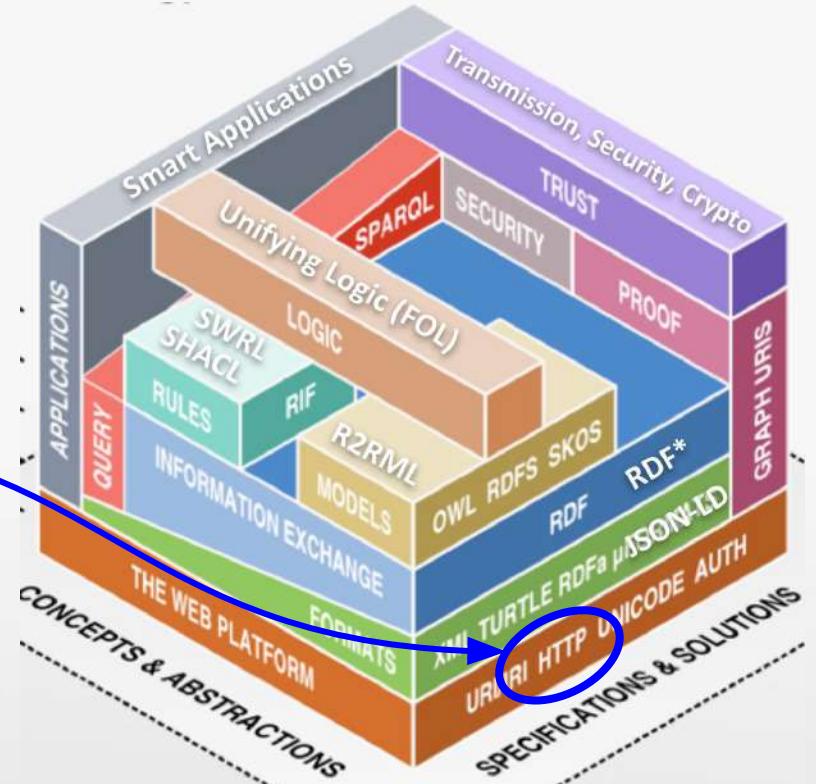
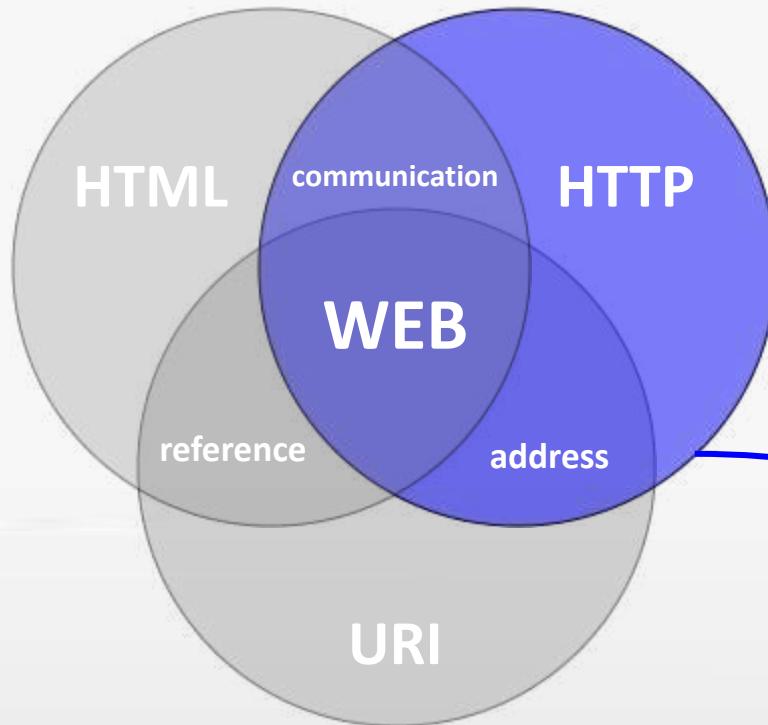


1. Identification (**URI**) & address (**URL**)  
e.g. `http://fiz-karlsruhe.de`
2. Communication / protocol (**HTTP**)  
`GET /index HTTP/2`  
`Host: fiz-karlsruhe.de`
3. Representation language (**HTML**)  
`Tabea works at`  
`<a href="http://fiz-karlsruhe.de">FIZ</a>`

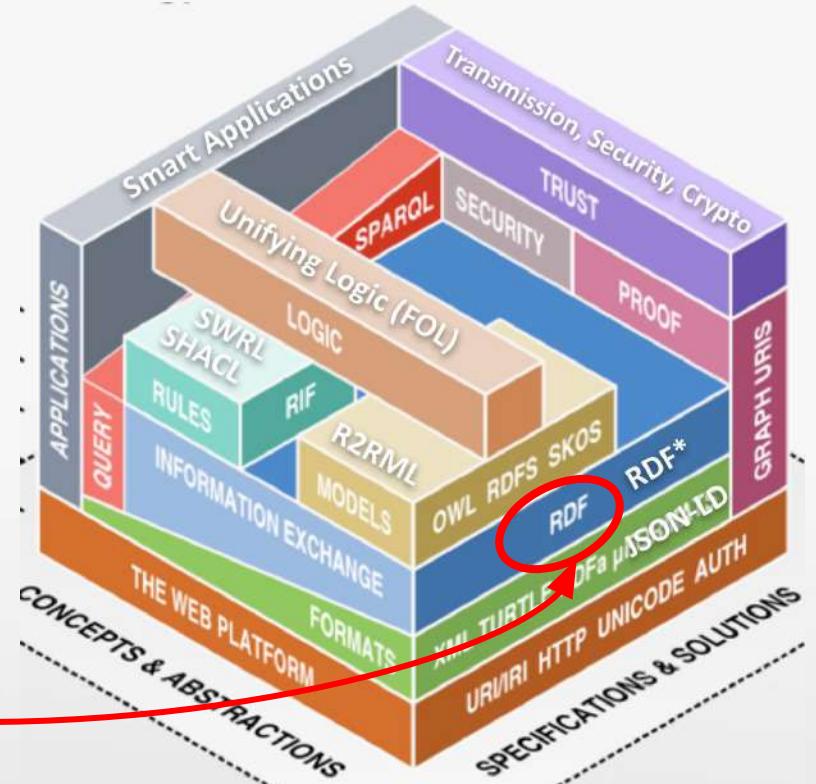
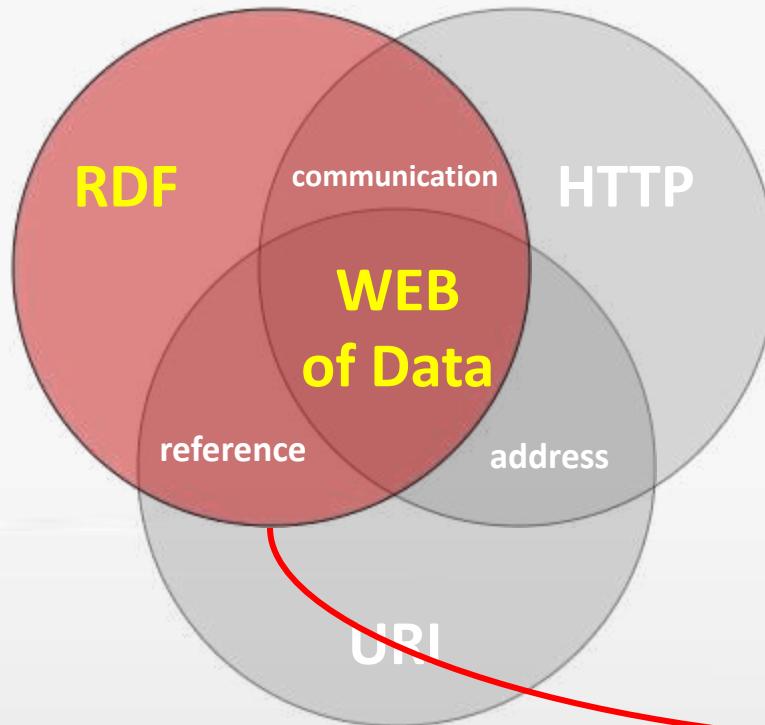
# Basic Architecture of the Web of Data



# Basic Architecture of the Web of Data



# Basic Architecture of the Web of Data



# The Web of Data

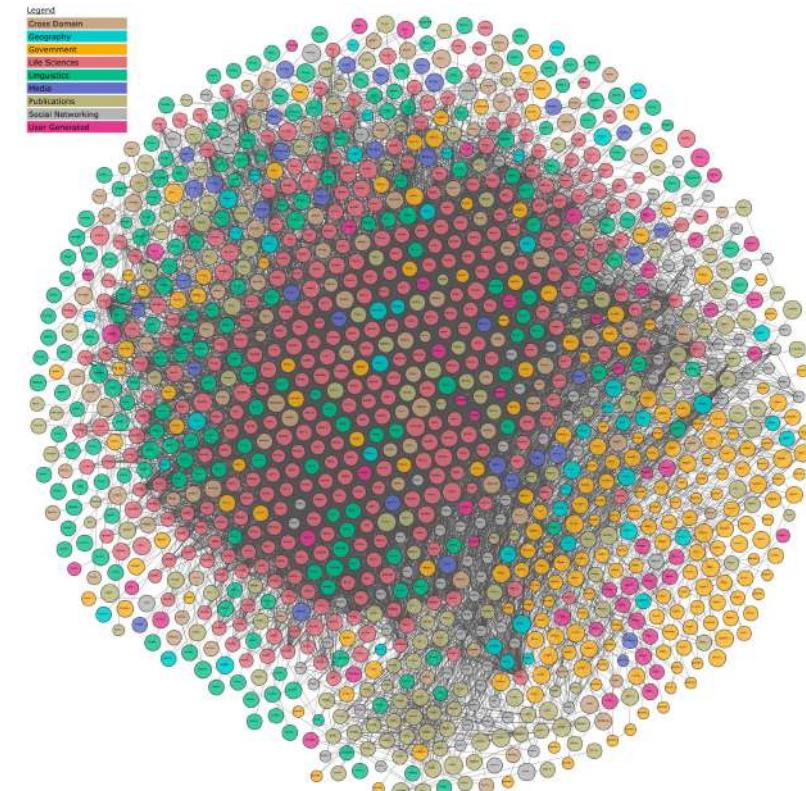
## Linked Data

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

## The Web of Data

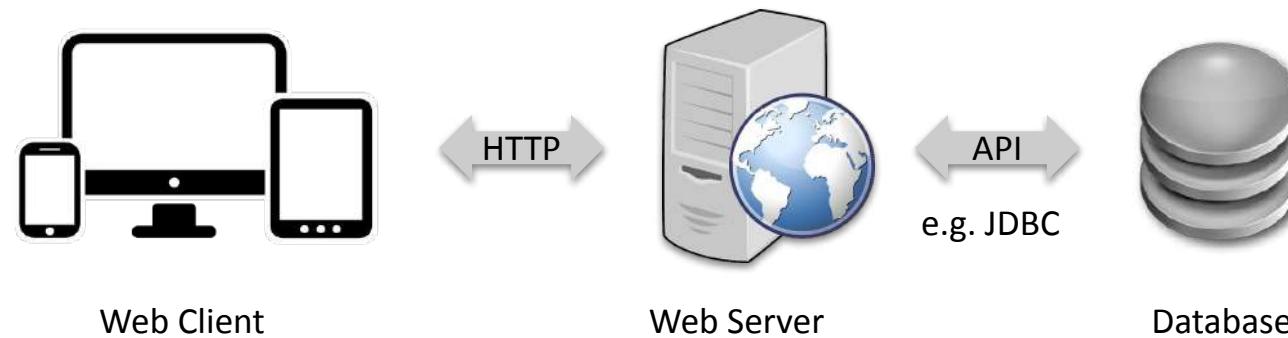
- Currently (01/2023) lod-cloud.net visualizes 1588 LOD datasets
- 2021 Common Crawl reported (JSON-LD usage)
  - 8,342,031 Web Sites
  - 793,347,572 URLs
  - 7,952,535,579 Entities
  - 37,872,880,504 Triples

<http://webdatacommons.org/structureddata/2021-12/stats/stats.html>



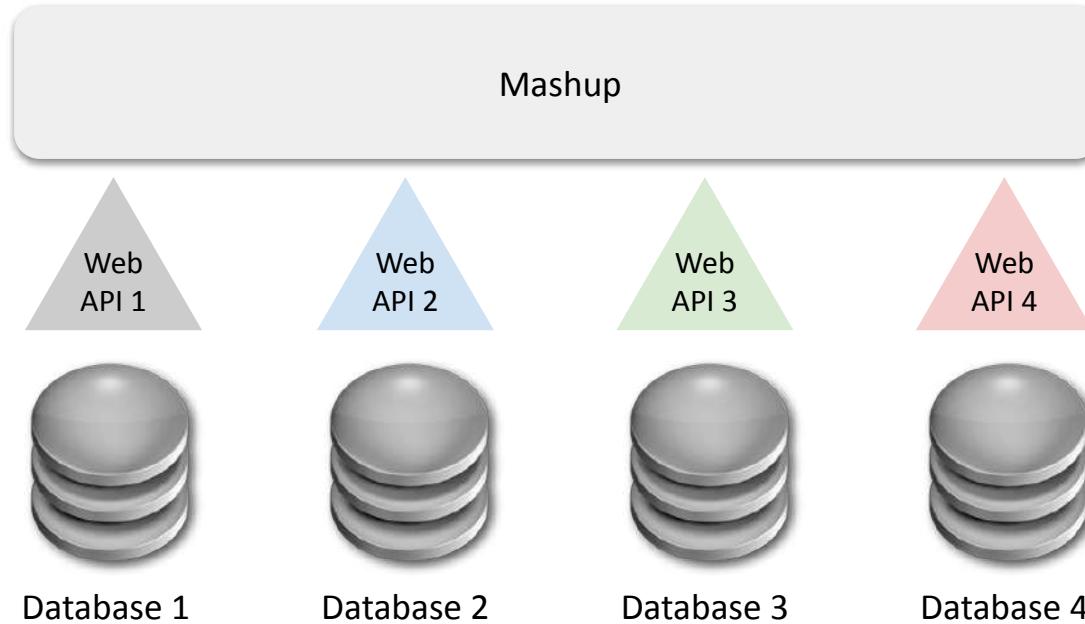
<http://lod-cloud.net/>

# Data Access in the Traditional Web



# Data Access in the Traditional Web

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



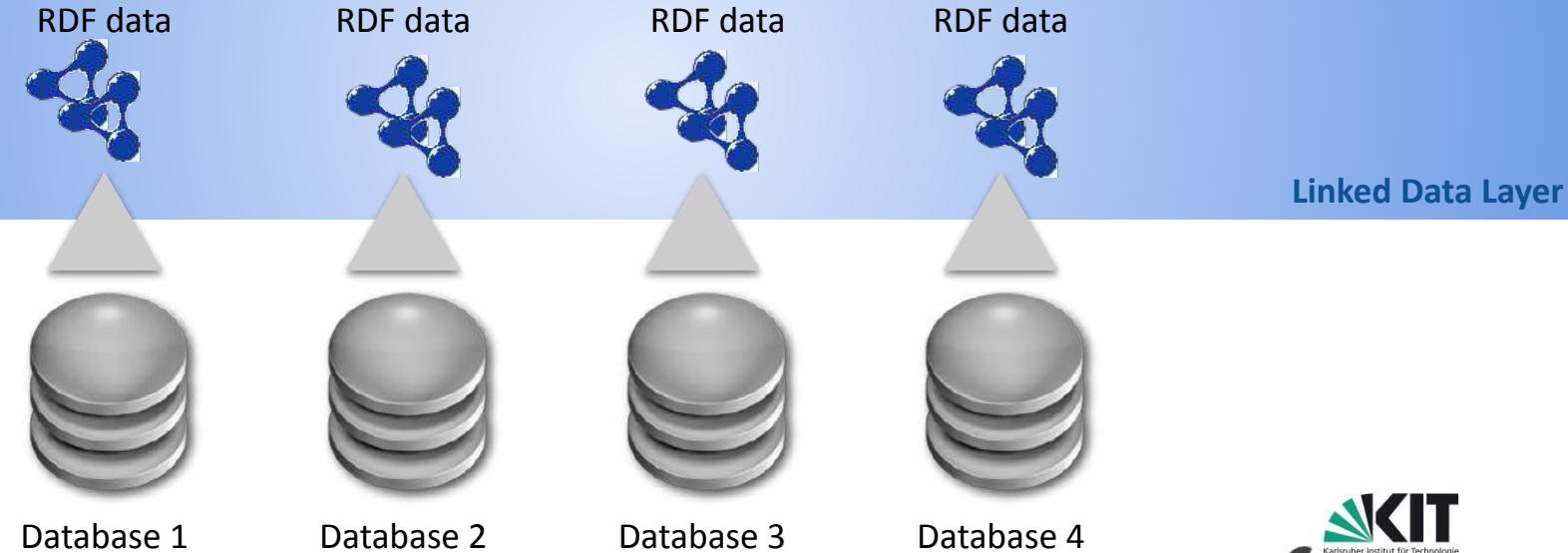
# In the Traditional Web...



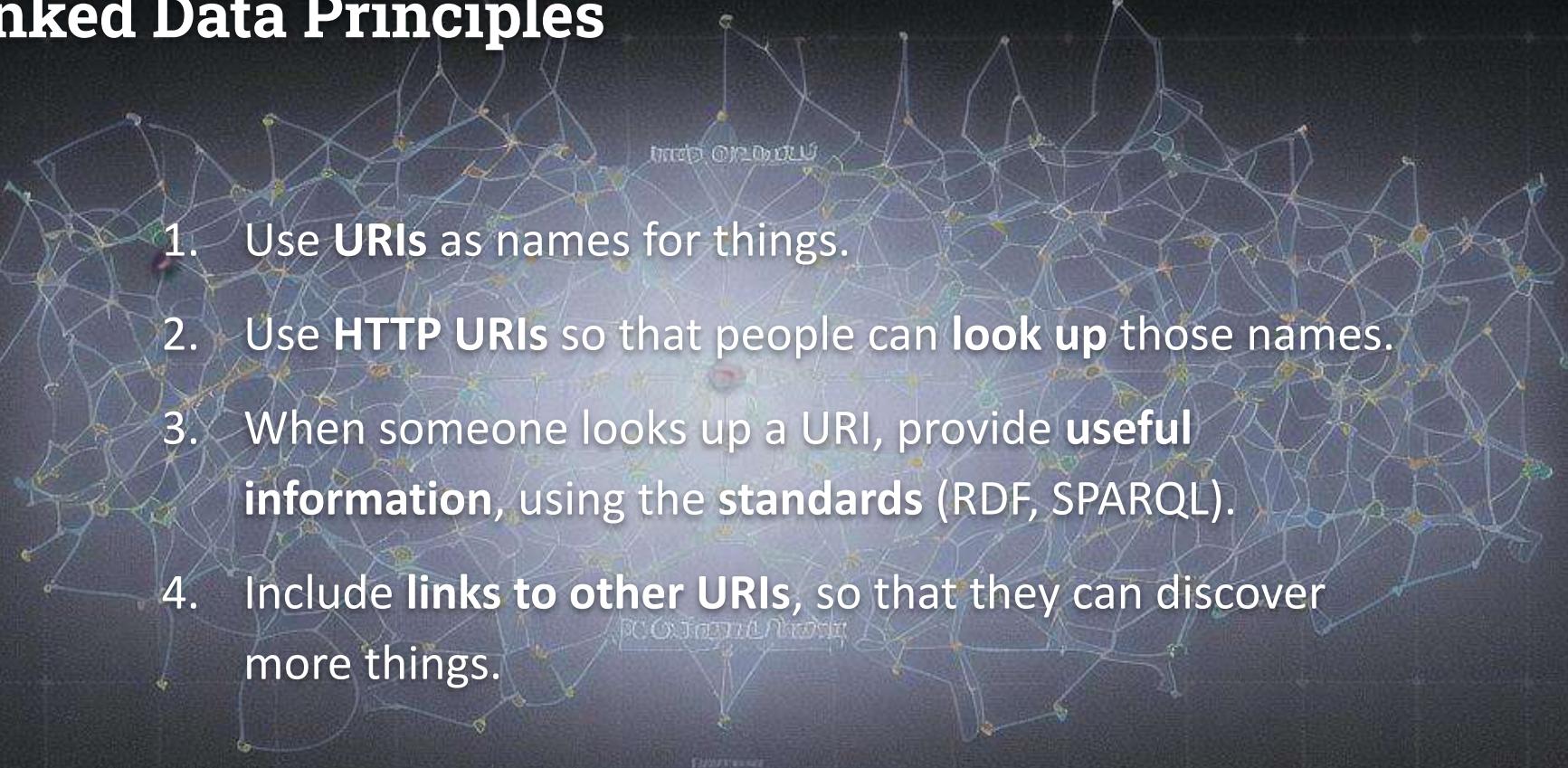
# How to Avoid Isolated Data Islands

## Apply Linked Data technology

- to publish (structured) data on the Web
- to draw connections from one data source to data from other data sources

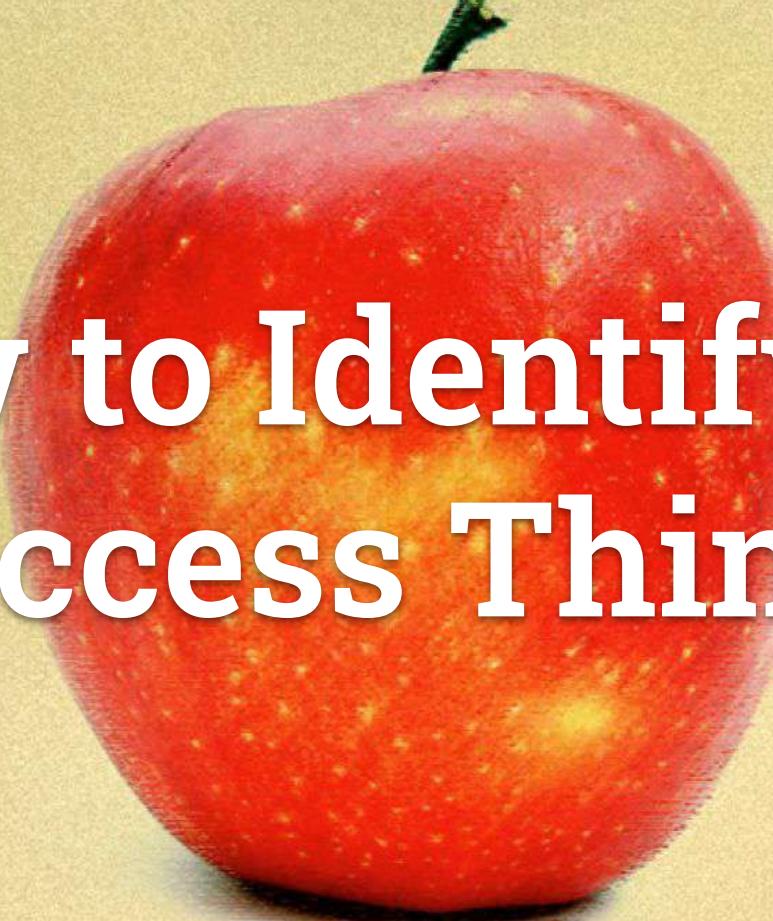


# Linked Data Principles

- 
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.

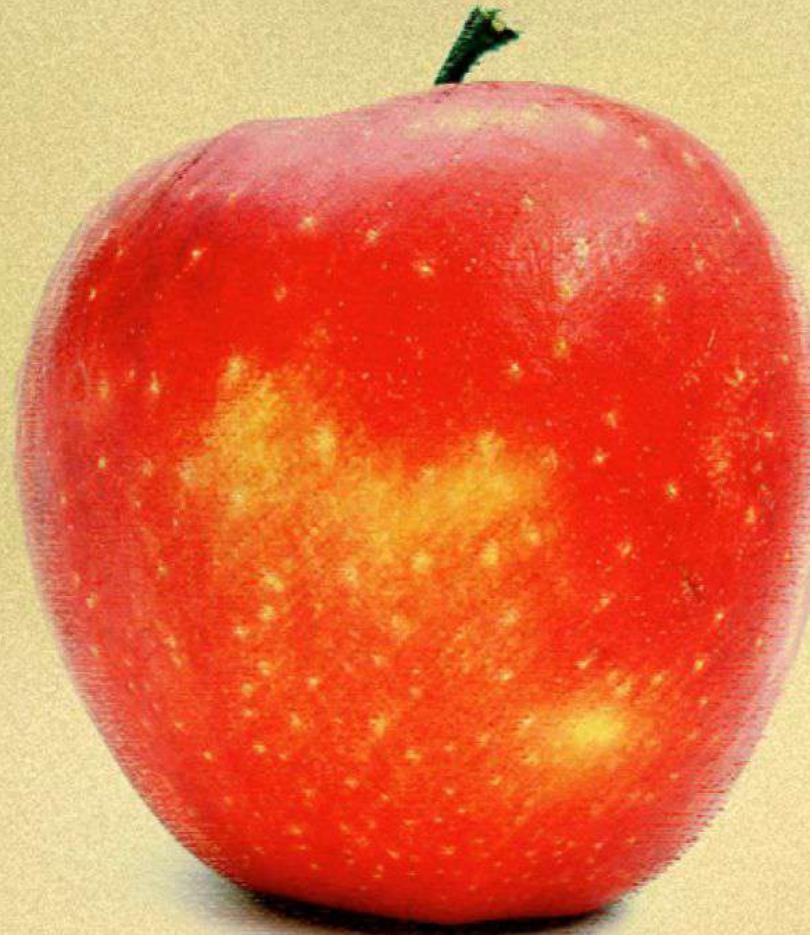
# 5-Star Linked Open Data

- Public Linked Data resources on the Web, licensed as Creative Common CC-BY
- Tim Berners-Lee's 5-Star Criteria for Linked Open Data
  - ★ Available on the Web (whatever format) but with an **open licence**, to be Open Data
  - ★★ Available as **machine-readable structured data**  
(e.g. excel instead of image scan of a table)
  - ★★★ as (2) plus **non-proprietary format** (e.g. CSV instead of excel)
  - ★★★★ All the above plus: use **open standards from W3C**  
(RDF and SPARQL) to identify things, so that people can point at your stuff
  - ★★★★★ All the above, plus: **link your data to other people's data** to provide context



# How to Identify and Access Things

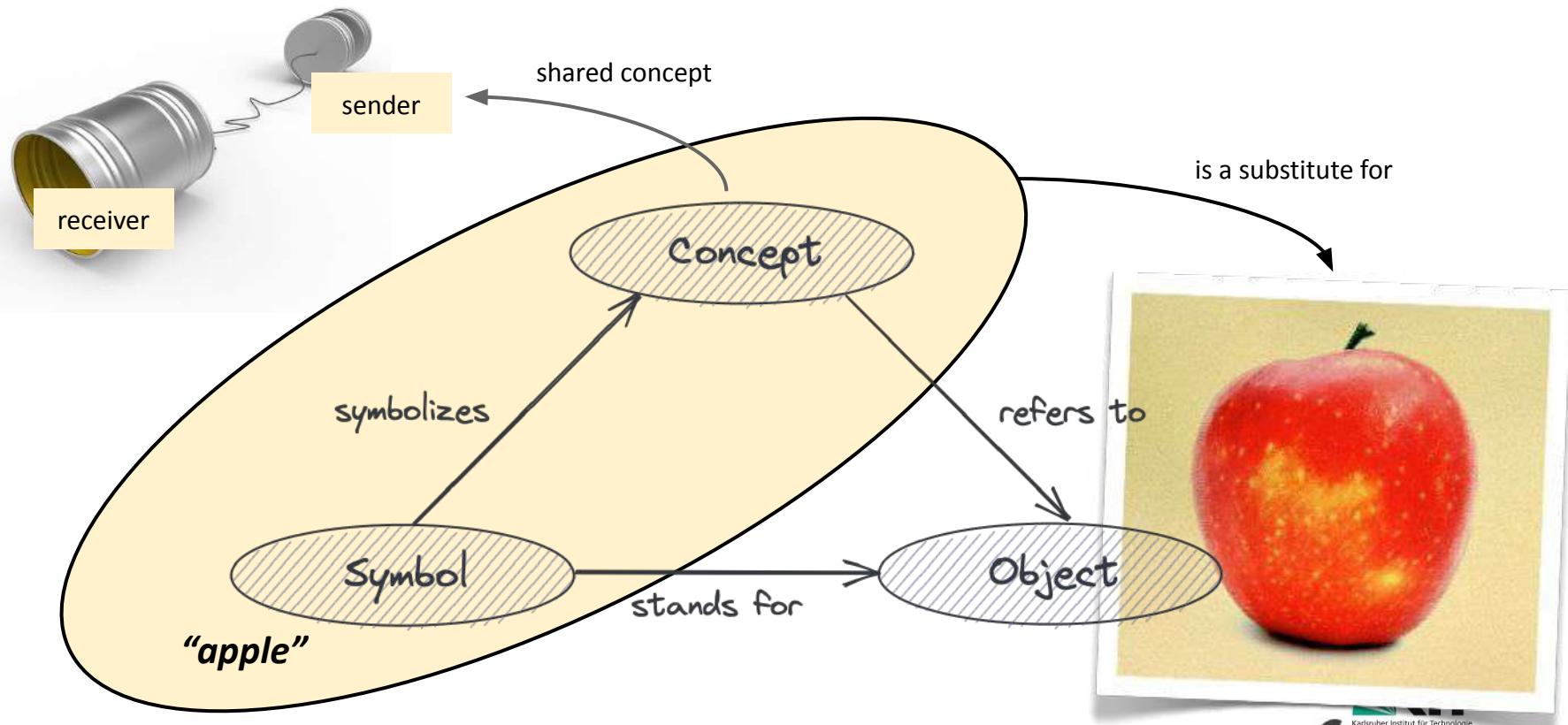
# What is this?





*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).
- An **Internationalized Resource Identifier (IRI)** extends the URI definition by expanding the set of permitted characters to most Unicode characters (RFC 3987).

`https://data.example.org/fruits/apples#this`

The diagram shows the string "https://data.example.org/fruits/apples#this" with four curly braces underneath it, each labeled with a component name: "schema" (covering "https://"), "domain/host" (covering ".example.org"), "path" (covering "/fruits/apples"), and "fragment" (covering "#this").

# Uniform Resource Identifier

- A **resource** can be **every object with a clear identity** (according to the context of the application)
  - e.g. web pages, books, locations, persons, relations among objects, abstract concepts, etc.
- The URI concept is already established in various domains, e.g.
  - the Web (URL),
  - books and publications (ISBN, ISSN, EAN),
  - Digital Object Identifier (DOI).

# Would this be an appropriate URI?



<https://www.xyz.org/apple>

<https://www.xyz.org/Apfel>

<https://www.xyz.org/pomme>

<https://www.xyz.org/苹果>

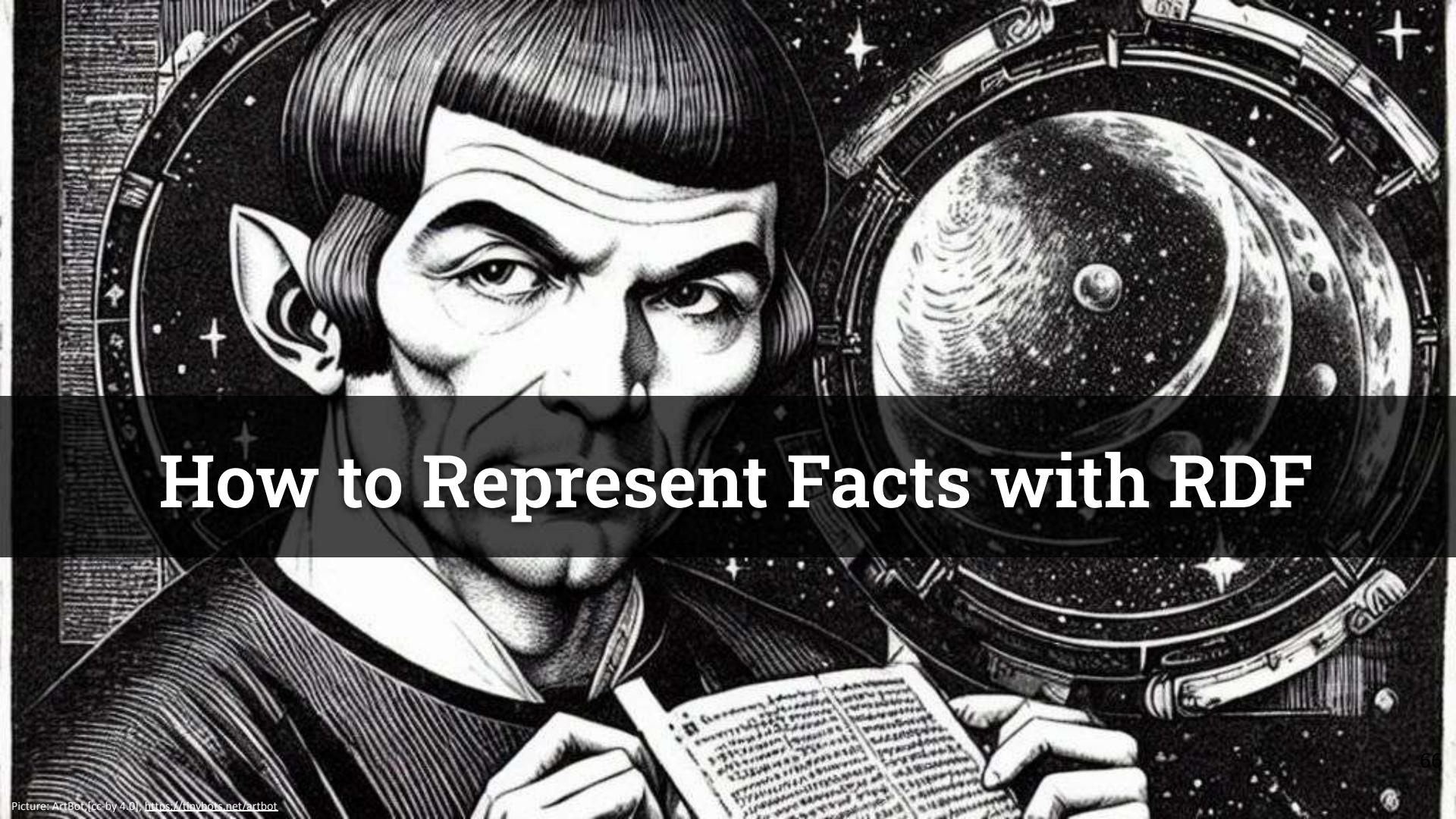
# Would this be an appropriate URI?



<https://www.wikidata.org/entity/Q89>



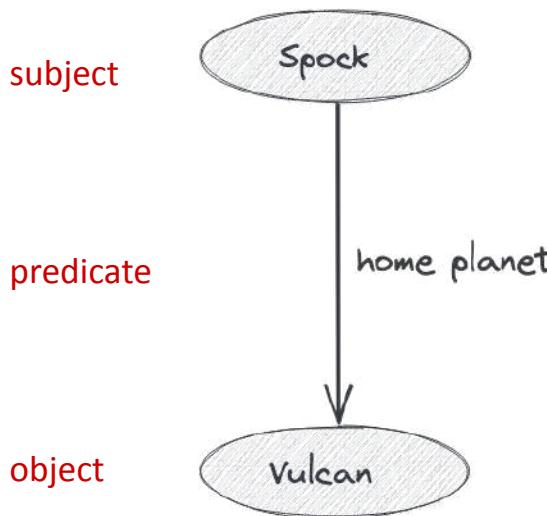
<https://www.wikidata.org/entity/Q312>



# How to Represent Facts with RDF

# Intuitive Knowledge Representation with Graphs

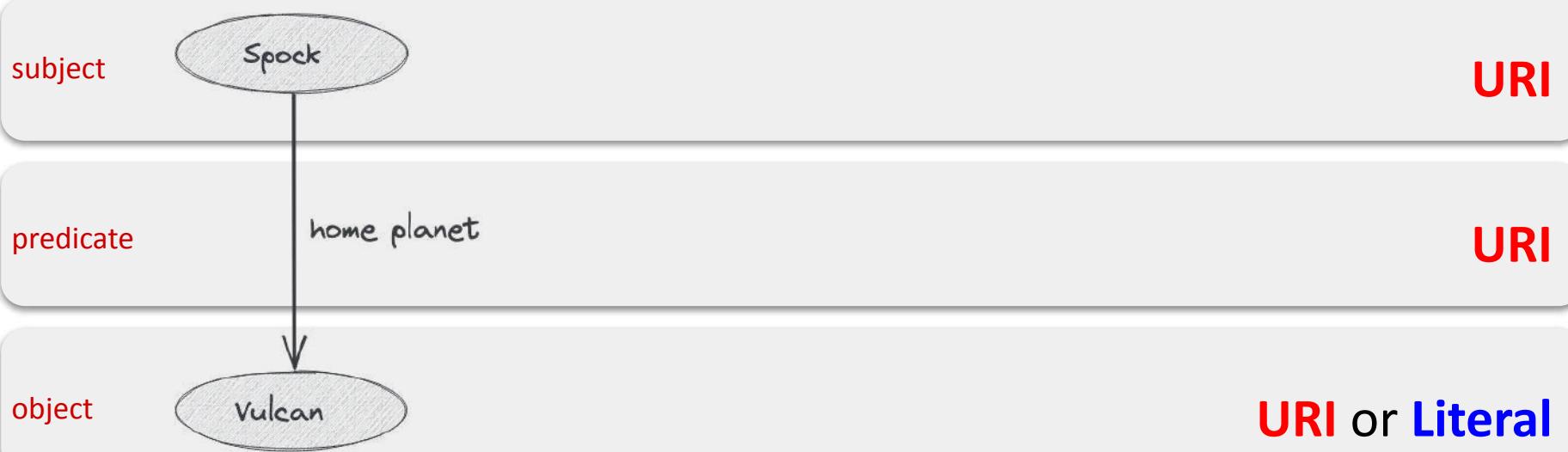
How do I represent the following fact:  
***“Spock’s home planet is Vulcan”*** in an intuitive way?





# Resource Description Framework

How do I represent the following fact:  
***“Spock’s home planet is Vulcan”*** in an intuitive way?





# Resource Description Framework

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

## RDF Statements (RDF-Triple):

Subject  
URI

Property  
URI

Object / Value  
URI / Literal

### N-Triples Serialization

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

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

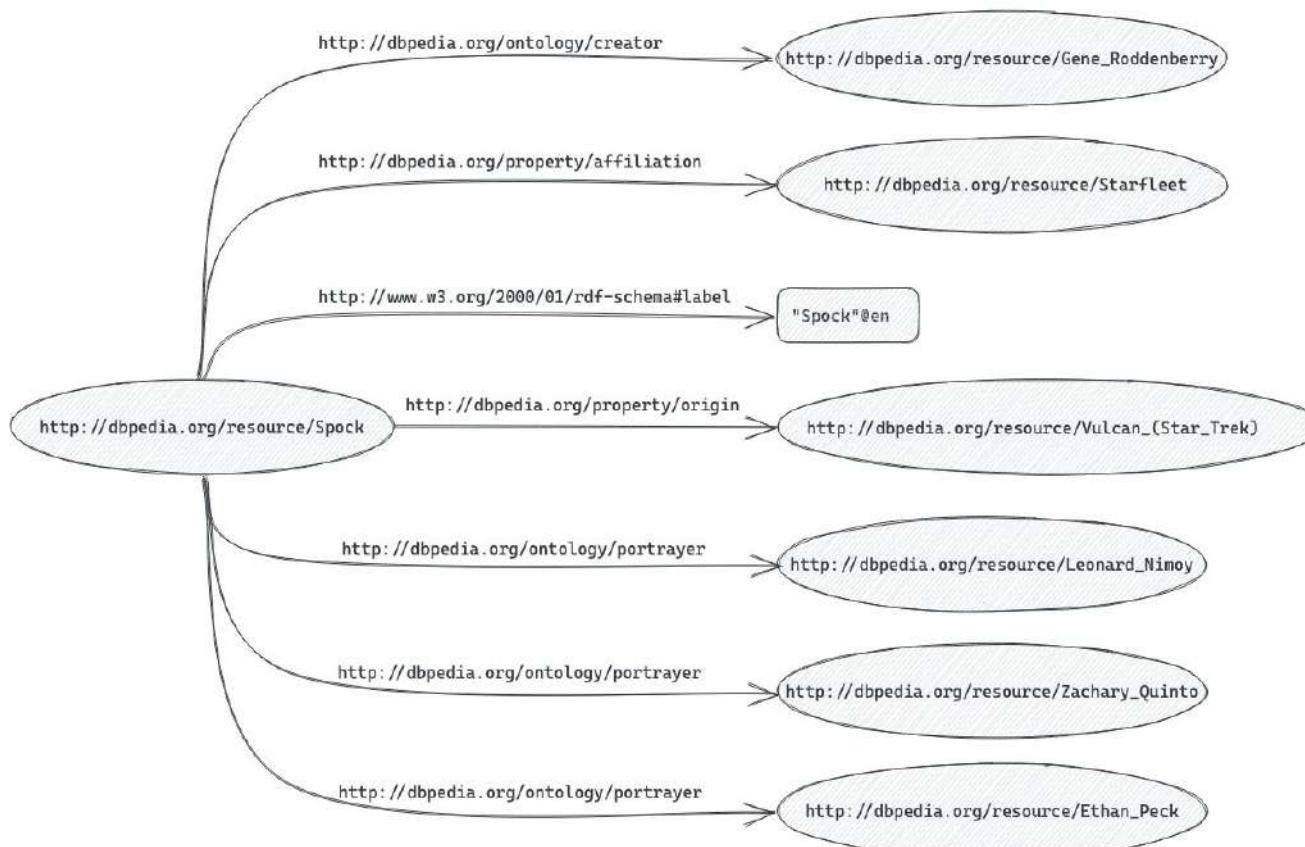
<[http://dbpedia.org/resource/Vulcan\\_\(Star\\_Trek\)](http://dbpedia.org/resource/Vulcan_(Star_Trek))>



### Graph Representation



# Resource Description Framework

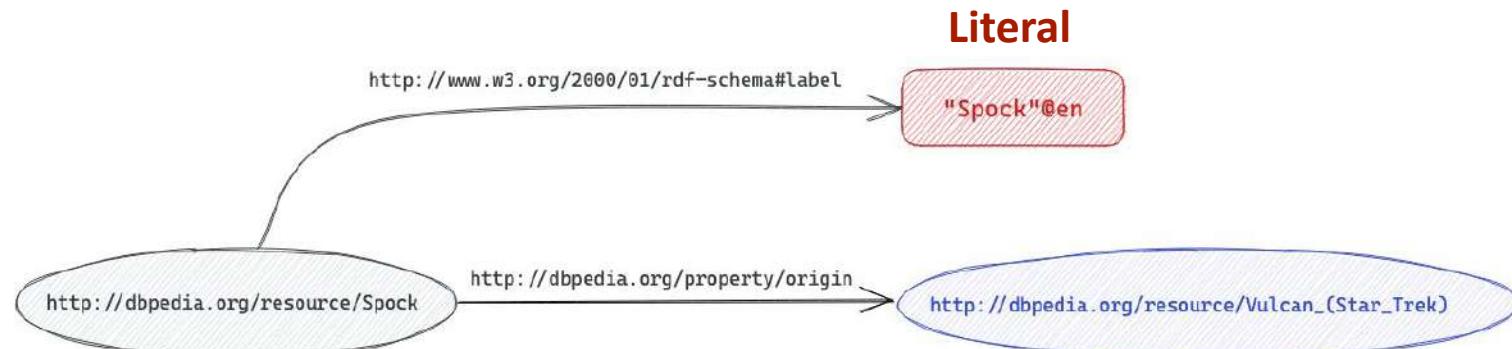




# Resource Description Framework

## URIs and Literals

- **URIs** identify and 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:

```
"Spock"^^<http://www.w3.org/2001/XMLSchema#string>
"1161.00"^^<http://www.w3.org/2001/XMLSchema#float>
"2023-08-02"^^<http://www.w3.org/2001/XMLSchema#date>
```

- **Language Tags** denote the (natural) language of the text:

Example:

"Semantik"@de , "Semantics"@en

<https://www.w3.org/TR/rdf11-concepts/#section-Datatypes>

Core types	<code>xsd:string</code> <code>xsd:boolean</code> <code>xsd:decimal</code> <code>xsd:integer</code> <code>xsd:double</code> <code>xsd:float</code> <code>xsd:date</code> <code>xsd:dateTime</code> <code>xsd:dateTimeStamp</code>	Character strings true, false Arbitrary-precision decimal Arbitrary-size integer numbers 64-bit floating point numbers 32-bit floating point numbers Dates (yyyy-mm-dd) with or without time zone Times (hh:mm:ss.sss...) with or without time zone Date and time with or without time zone Date and time with required time zone
IEEE floating-point numbers	<code>xsd:double</code> <code>xsd:float</code>	64-bit floating point numbers 32-bit floating point numbers
Time and date	<code>xsd:date</code> <code>xsd:dateTime</code> <code>xsd:dateTimeStamp</code>	Dates (yyyy-mm-dd) with or without time zone Times (hh:mm:ss.sss...) with or without time zone Date and time with required time zone
Recurring and partial dates	<code>xsd:gYear</code> <code>xsd:gMonth</code> <code>xsd:gDay</code> <code>xsd:gYearMonth</code> <code>xsd:gMonthDay</code> <code>xsd:duration</code> <code>xsd:yearMonthDuration</code> <code>xsd:dayTimeDuration</code>	Gregorian calendar year Gregorian calendar month Gregorian calendar day of the month Gregorian calendar year and month Gregorian calendar month and day Duration of time Duration of time (months and days) Duration of time (days, hours, minutes, seconds)
Limited-range integer numbers	<code>xsd:byte</code> <code>xsd:short</code> <code>xsd:int</code> <code>xsd:long</code> <code>xsd:unsignedByte</code> <code>xsd:unsignedShort</code> <code>xsd:unsignedInt</code> <code>xsd:unsignedLong</code> <code>xsd:positiveInteger</code> <code>xsd:nonNegativeInteger</code> <code>xsd:negativeInteger</code> <code>xsd:nonPositiveInteger</code>	-128...+127 (8 bit) -32768...+32767 (16 bit) -2147483648...+2147483647 (32 bit) 0...255 (8 bit) 0...65535 (16 bit) 0...4294967295 (32 bit) 0...1844674407370955161 (64 bit) Integer numbers >0 Integer numbers ≥0 Integer numbers <0 Integer numbers ≤0
Encoded binary data	<code>xsd:hexBinary</code> <code>xsd:base64Binary</code> <code>xsd:anyURI</code> <code>xsd:language</code>	Hex-encoded binary data Base64-encoded binary data Absolute or relative URIs and URLs Language tags per [BCP47]
Miscellaneous XSD types	<code>xsd:normalizedString</code> <code>xsd:token</code> <code>xsd:NMTOKEN</code> <code>xsd:Name</code> <code>xsd:NCName</code>	Whitespace-normalized strings Tokenized strings XML NMTOKENs XML Names XML NCNames



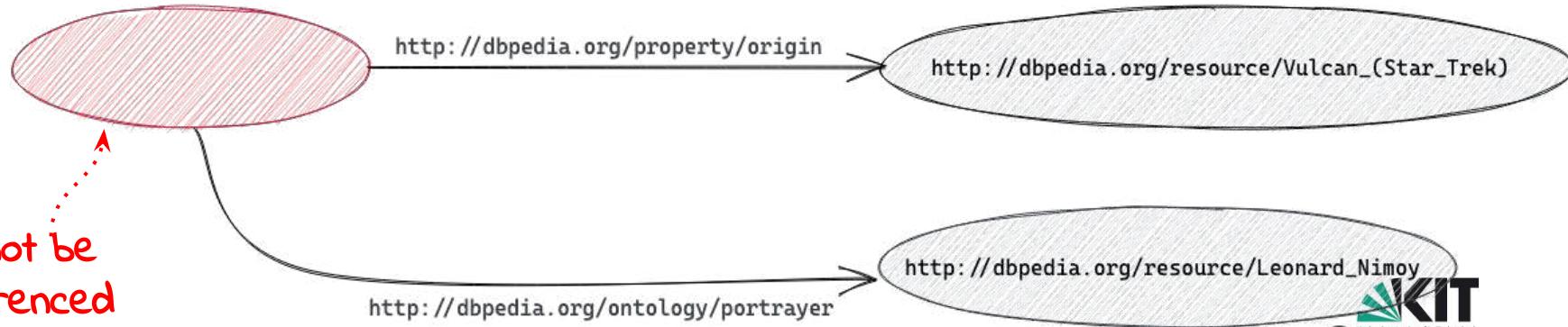
# Existential Assertions – Blank Nodes

## Blank Nodes

denote the **existence of an individual** with specific attributes, but **without providing an identification or reference**.

**Blank Node**

Cannot be referenced externally

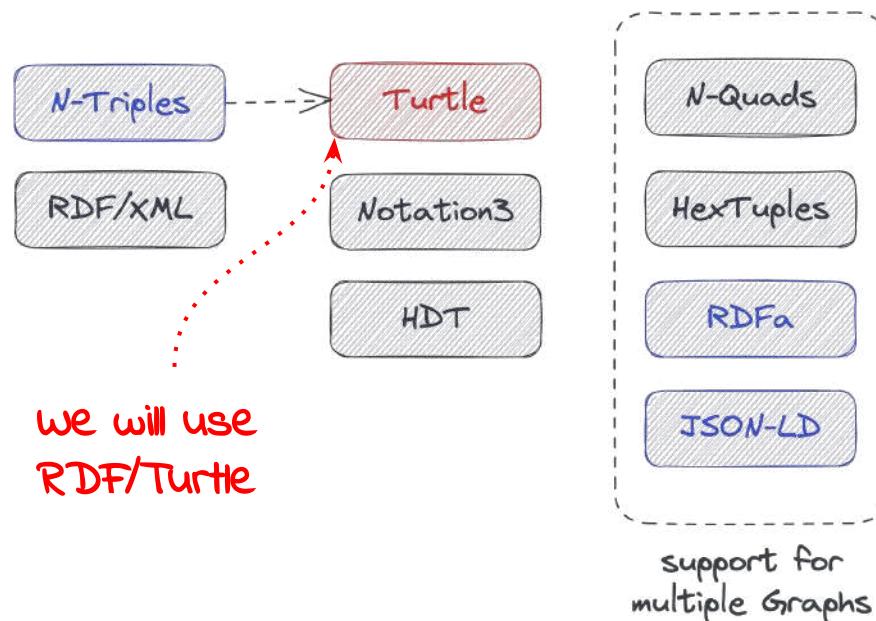


# RDF Serializations

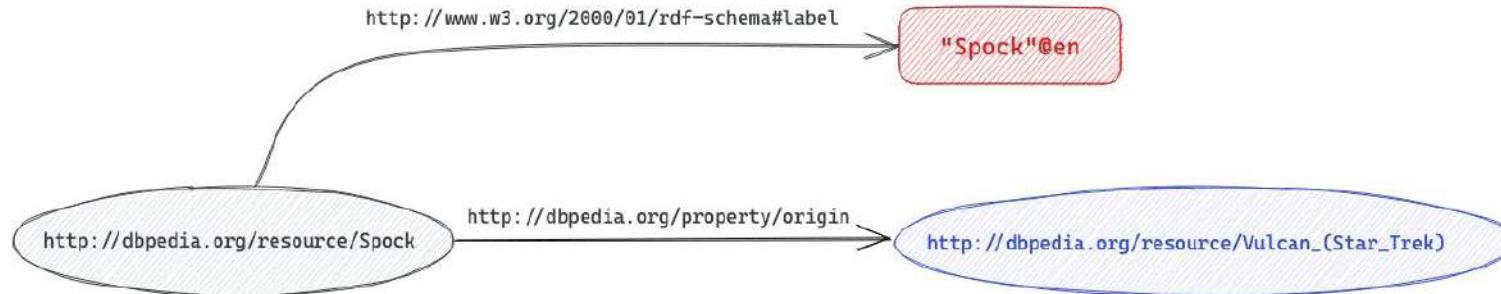


RDF comes with several different **serialization formats**:

→ N-Triples, RDF/XML, JSON-LD, Turtle, N-Quads, RDFa, Notation3, HexTuples...



# N-Triples Serialization



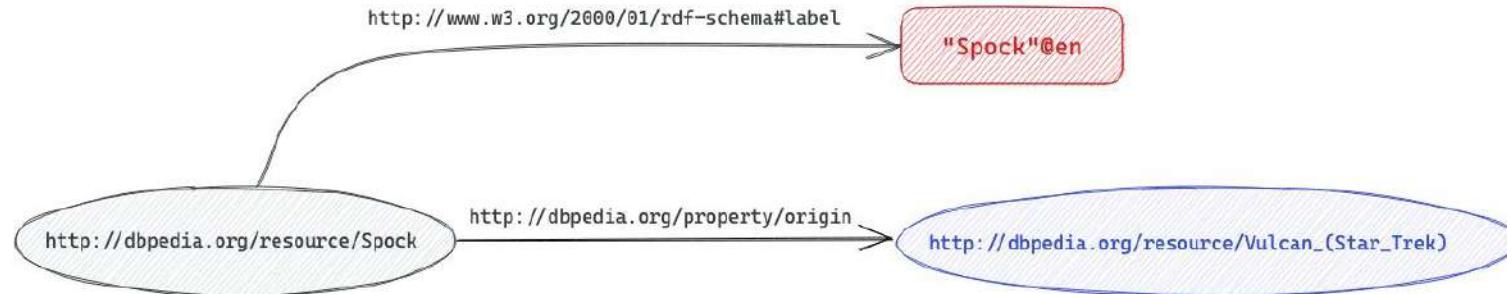
## N-Triples Notation

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

### N-Triples Serialization

```
<http://dbpedia.org/resource/Spock> <http://www.w3.org/2000/01/rdf-schema#label> “Spock”@en .  
<http://dbpedia.org/resource/Spock> <http://dbpedia.org/property/origin> <http://dbpedia.org/resource/Vulcan_(Star_Trek)> .
```

# RDF Turtle Serialization



## Turtle (Terse RDF Triple Language) Notation

Is an extension of N-Triples

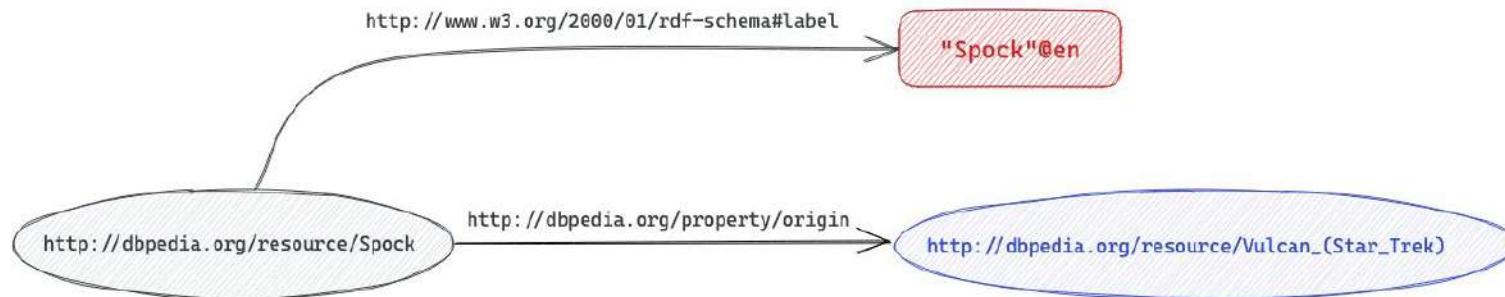
RDF/Turtle allows  
shortcuts and  
abbreviations for  
readability.

```
@prefix dbp: <http://dbpedia.org/property/> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@base <http://dbpedia.org/resource/> .  
  
<Spock> rdfs:label "Spock"@en .  
<Spock> dbp:origin <Vulcan\_\(Star\_Trek\)> .
```

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

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

# RDF Turtle Serialization



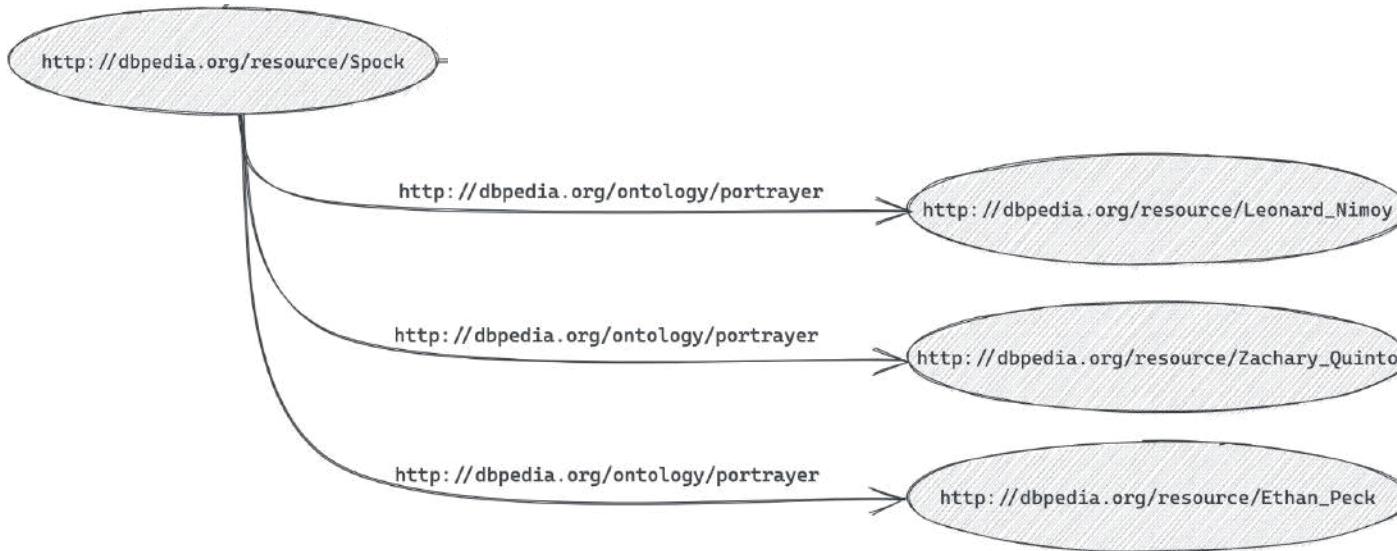
**Further RDF abbreviations with Turtle:**

```
@prefix dbp: <http://dbpedia.org/property/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://dbpedia.org/resource/> .

<Spock> rdfs:label "Spock"@en ;
    dbp:origin <Vulcan_(Star_Trek)> .
```

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

# RDF Turtle Serialization



## Further RDF abbreviations with Turtle:

```

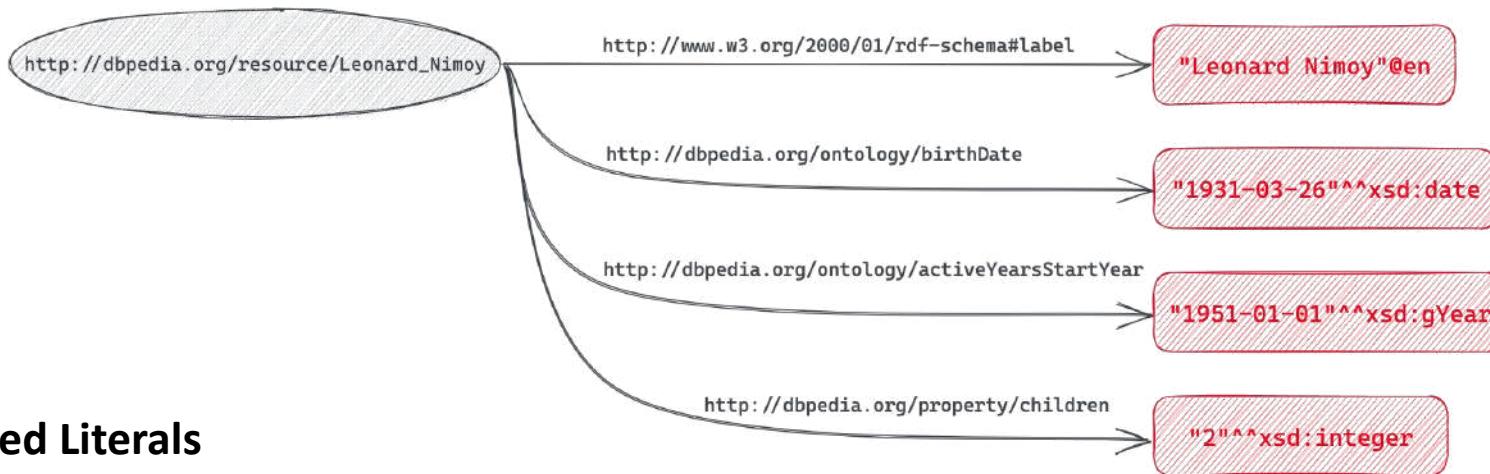
@prefix dbo: <http://dbpedia.org/ontology/> .
@base <http://dbpedia.org/resource/> .
    
```

```

<Spock> dbo:portrayer <Zachary_Quinto>, 
      <Leonard_Nimoy>, 
      <Ethan_Peck> .
    
```

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

# RDF Turtle Serialization

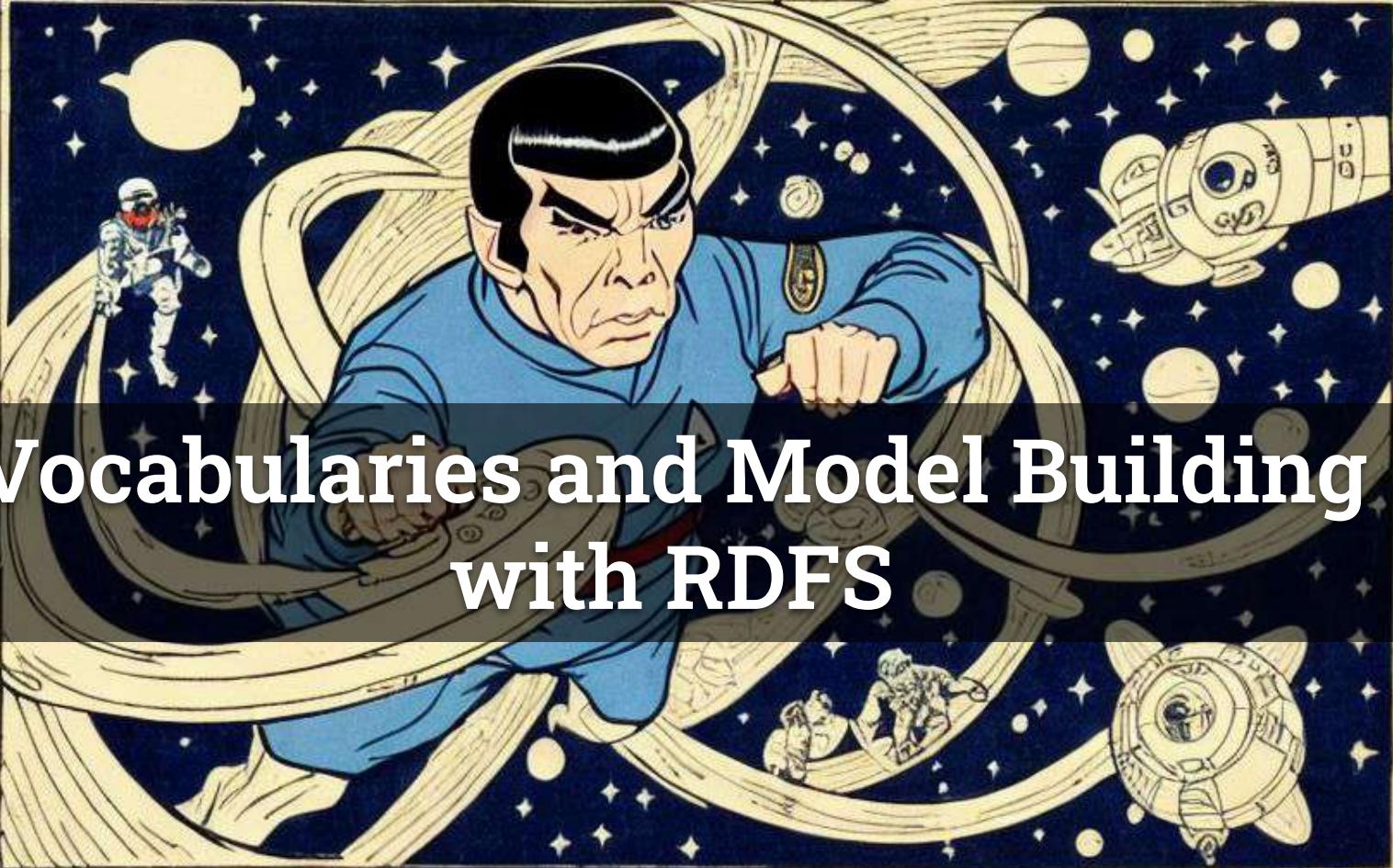


## Typed Literals

```

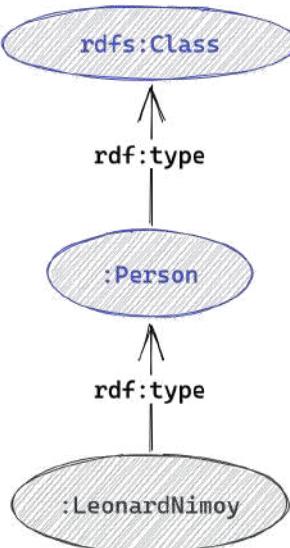
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbr: <http://dbpedia.org/property/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://dbpedia.org/resource/> .

<Leonard_Nimoy> rdfs:label "Leonard Nimoy"@en ;
    dbo:birthDate "1931-03-26"^^xsd:date ;
    dbo:activeYearsStartYear "1951-01-01"^^xsd:gYear ;
    dbr:children "2"^^xsd:integer .
  
```



# Vocabularies and Model Building with RDFS

# 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:  
**:Person rdf:type rdfs:Class .**  
**:LeonardNimoy rdf:type :Person .**

The **Class of all classes**.

A property to relate an instance to its class.

**LeonardNimoy ∈ Person**

```

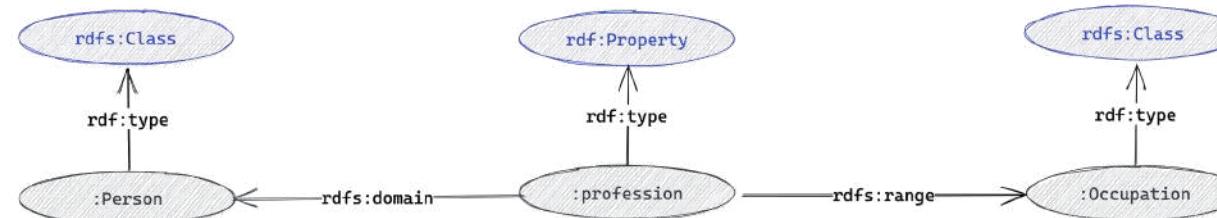
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://example.org/KG2023#> .
  
```

# RDF Schema



- Definition of **properties** via **rdf:Property** The Class of all properties.
- Definition of **property restrictions on domain and range** via **rdfs:domain** and **rdfs:range**
- Example

:Person	<b>rdf:type</b>	<b>rdfs:Class</b> .
:Occupation	<b>rdf:type</b>	<b>rdfs:Class</b> .
<b>:profession</b>	<b>rdf:type</b>	<b>rdf:Property</b> .
<b>:profession</b>	<b>rdfs:domain</b>	<b>:Person</b> .
<b>:profession</b>	<b>rdfs:range</b>	<b>:Occupation</b> .



# RDF Schema

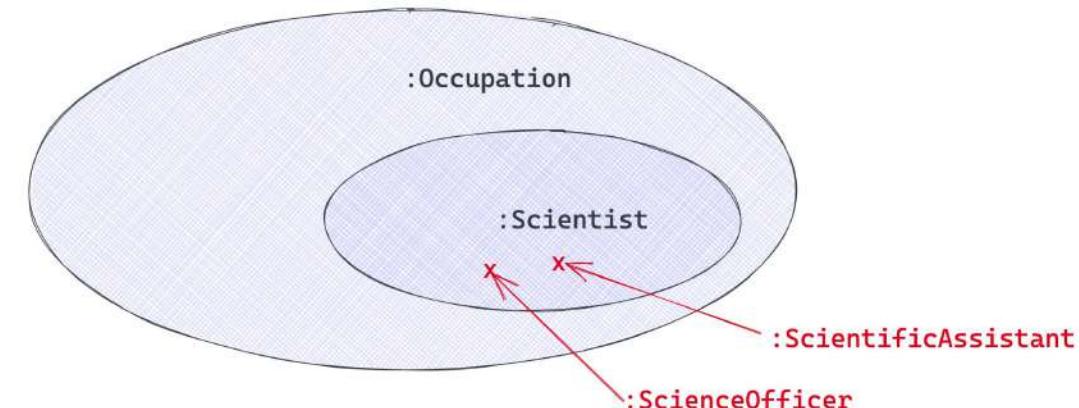


Definition of **hierarchical relationships**:

- **Subclasses and superclasses** via `rdfs:subClassOf`
- Example:

`:Scientist rdfs:subClassOf :Occupation .`

  
 $\text{Scientist} \subseteq \text{occupation}$



`:ScientificAssistant rdf:type :Scientist .`  
`:ScienceOfficer rdf:type :Scientist .`

# RDF Schema



```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix : <http://example.org/KG2023#> .
```

:Person	rdf:type	rdfs:Class .	
:FictionalCharacter	rdf:type	rdfs:Class .	
:Occupation	rdf:type	rdfs:Class .	
:Artist	rdfs:subClassOf	:Occupation .	
:Scientist	rdfs:subClassOf	:Occupation .	

Class Definitions

:profession	rdf:type	rdf:Property ;	
	rdfs:domain	:Person, FictionalCharacter ;	
	rdfs:range	:Occupation .	

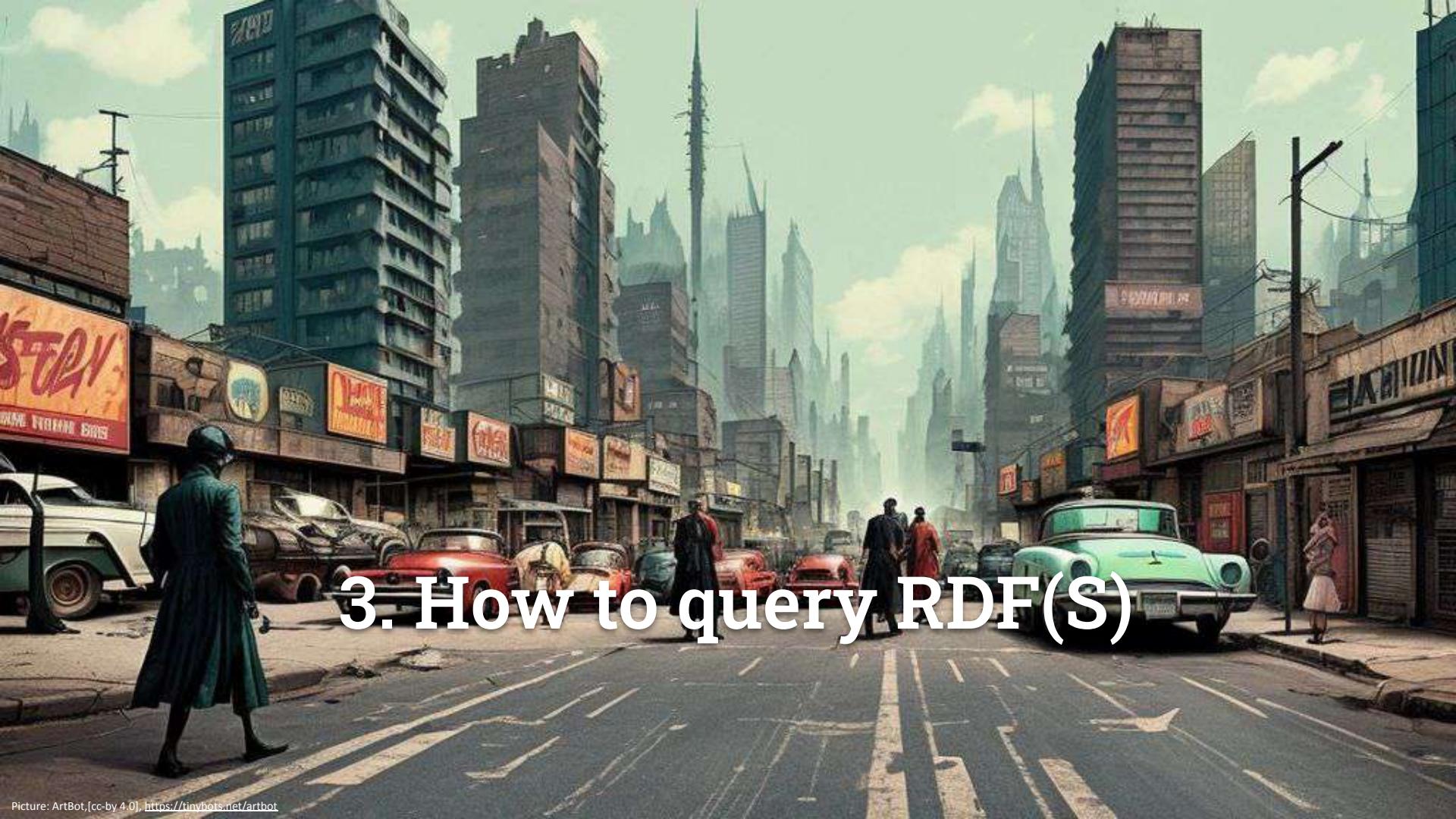
Property Definitions

:Actor	rdf:type	:Artist .	
:ScienceOfficer	rdf:type	:Scientist .	
:LeonardNimoy	rdf:type	:Person ;	
	:profession	:Actor ;	
	rdfs:label	"Leonard Nimoy"@en .	
:Spock	rdf:type	:FictionalCharacter ;	
	rdfs:label	"Spock"@en .	
	profession	:ScienceOfficer .	

Instance Definitions

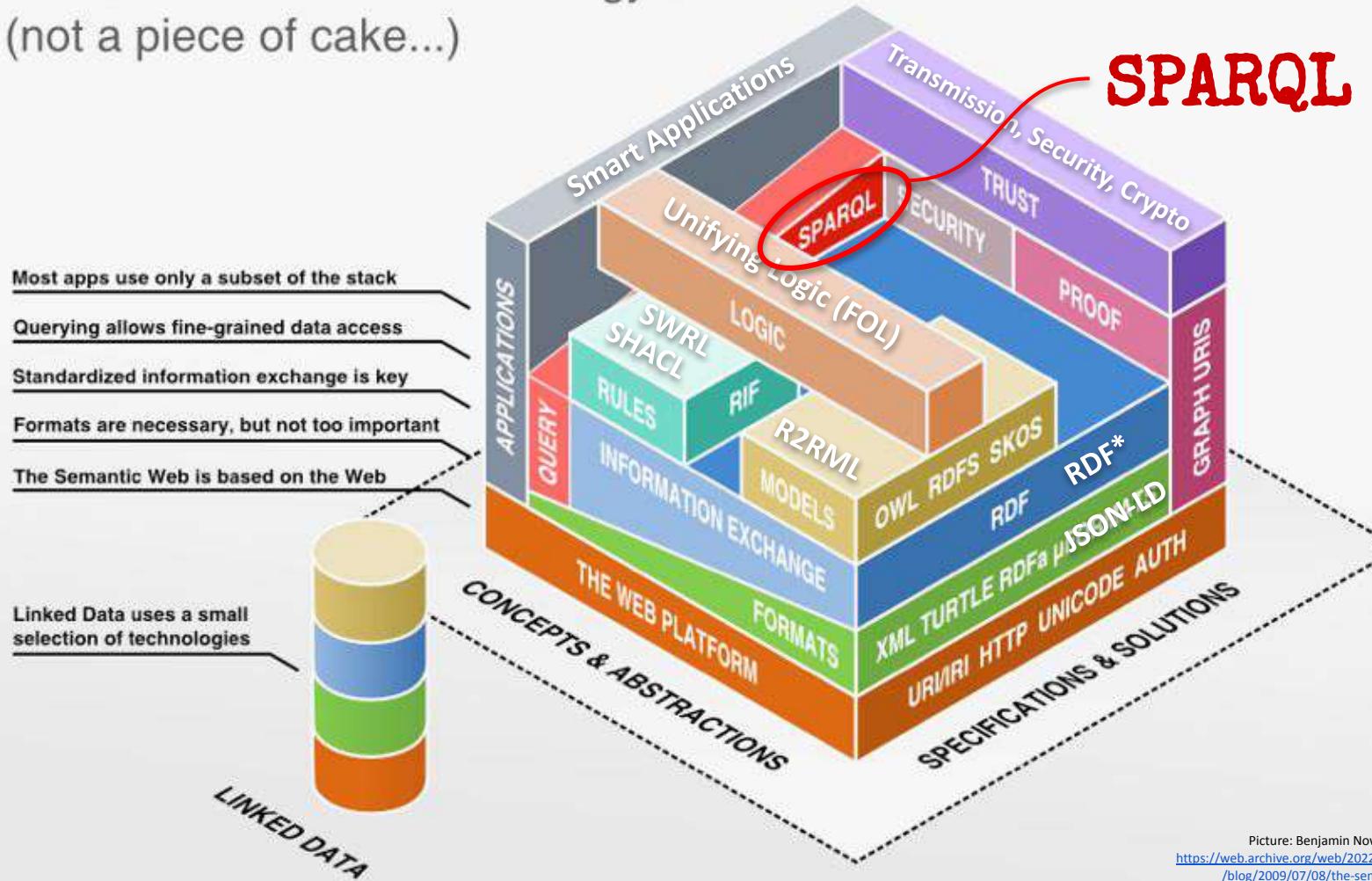
T-BOX

A-BOX

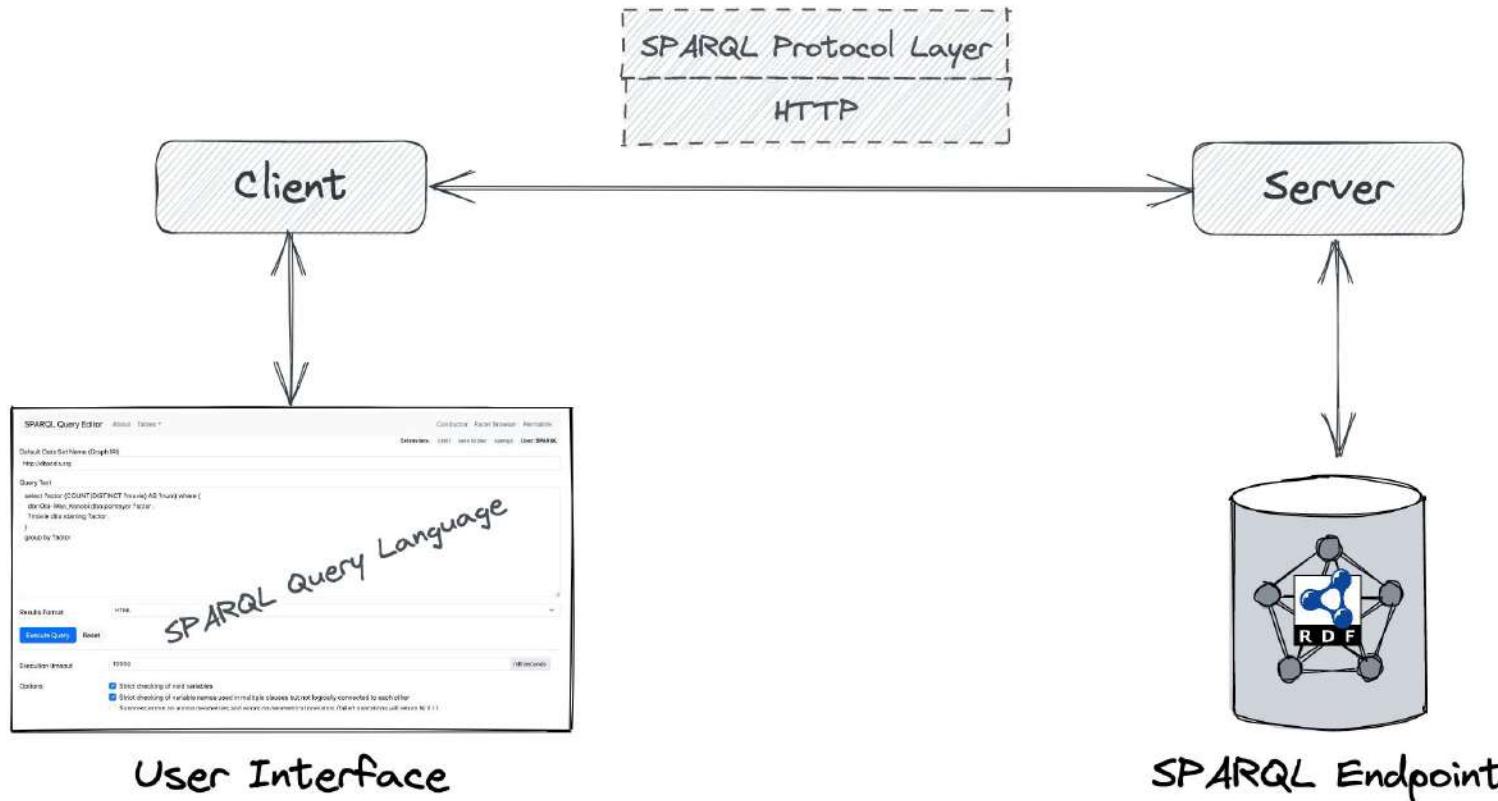
A vibrant, retro-futuristic cityscape. In the foreground, a man in a dark coat walks away from the viewer down a street lined with classic cars like a white Ford and a green Buick. The city is filled with tall, weathered buildings, many with large, colorful neon signs for "ZOO", "KFC", "MOTEL", and "HOTEL". In the background, a dense cluster of futuristic skyscrapers rises against a hazy sky.

### 3. How to query RDF(S)

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



# SPARQL – A Query Language for Knowledge Graphs



# SPARQL – Endpoint Example

SPARQL Query Editor   About   Tables ▾   Conductor   Facet Browser   Permalink  
Default Data Set Name (Graph IRI)  
<http://dbpedia.org>   Extensions: cxml save to daw sponge User: SPARQL

Query Text

```
select ?author (COUNT(?novels) AS ?books) where {
  ?novels dct:subject dbc:Dystopian_novels ;
    dbo:author ?author .
}
GROUP BY ?author
ORDER BY DESC(?books)
```

Results Format   **HTML**

**Execute Query**   Reset

Execution timeout    milliseconds

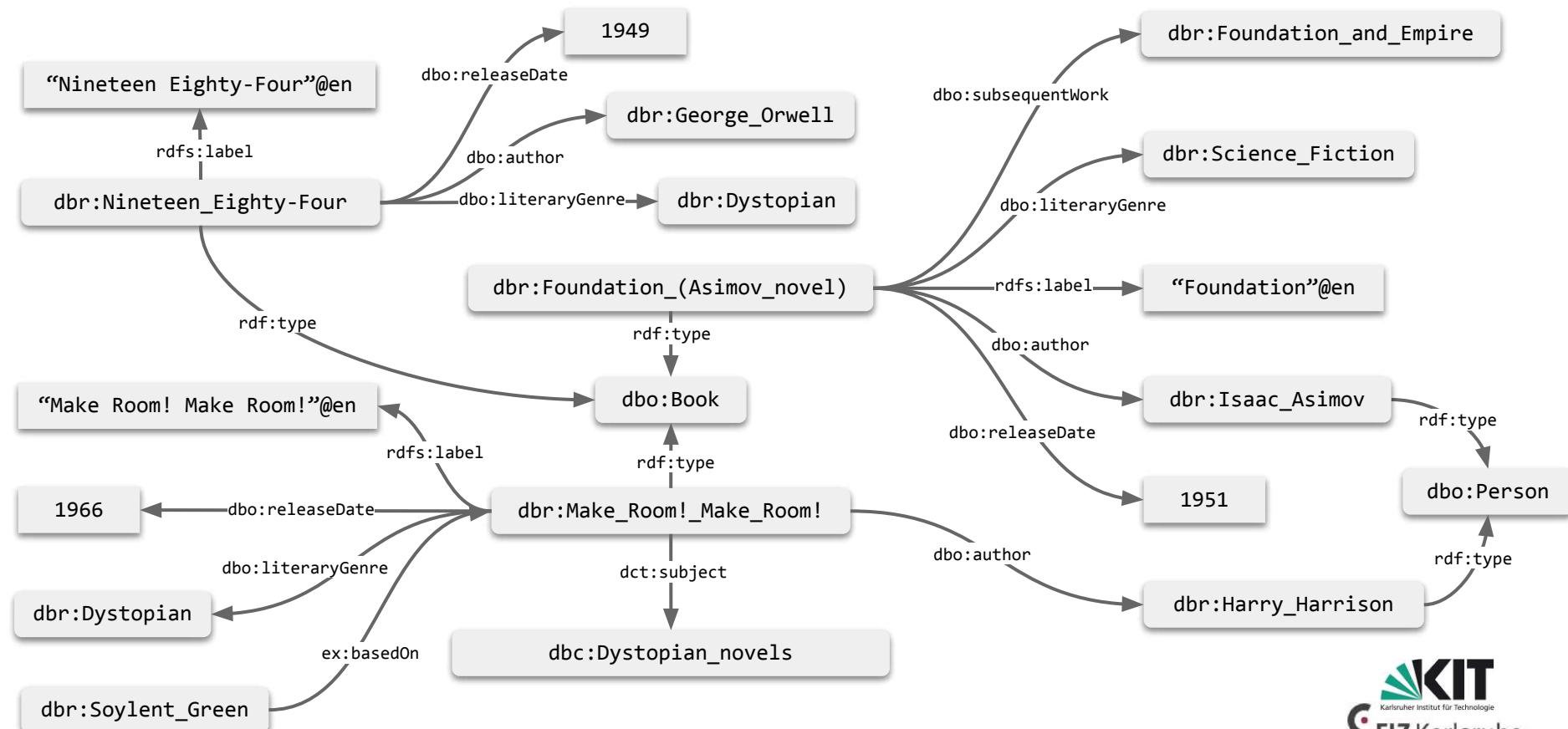
Options

- Strict checking of void variables
- Strict checking of variable names used in multiple clauses but not logically connected to each other
- Suppress errors on wrong geometries and errors on geometrical operators (failed operations will return NULL)
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)

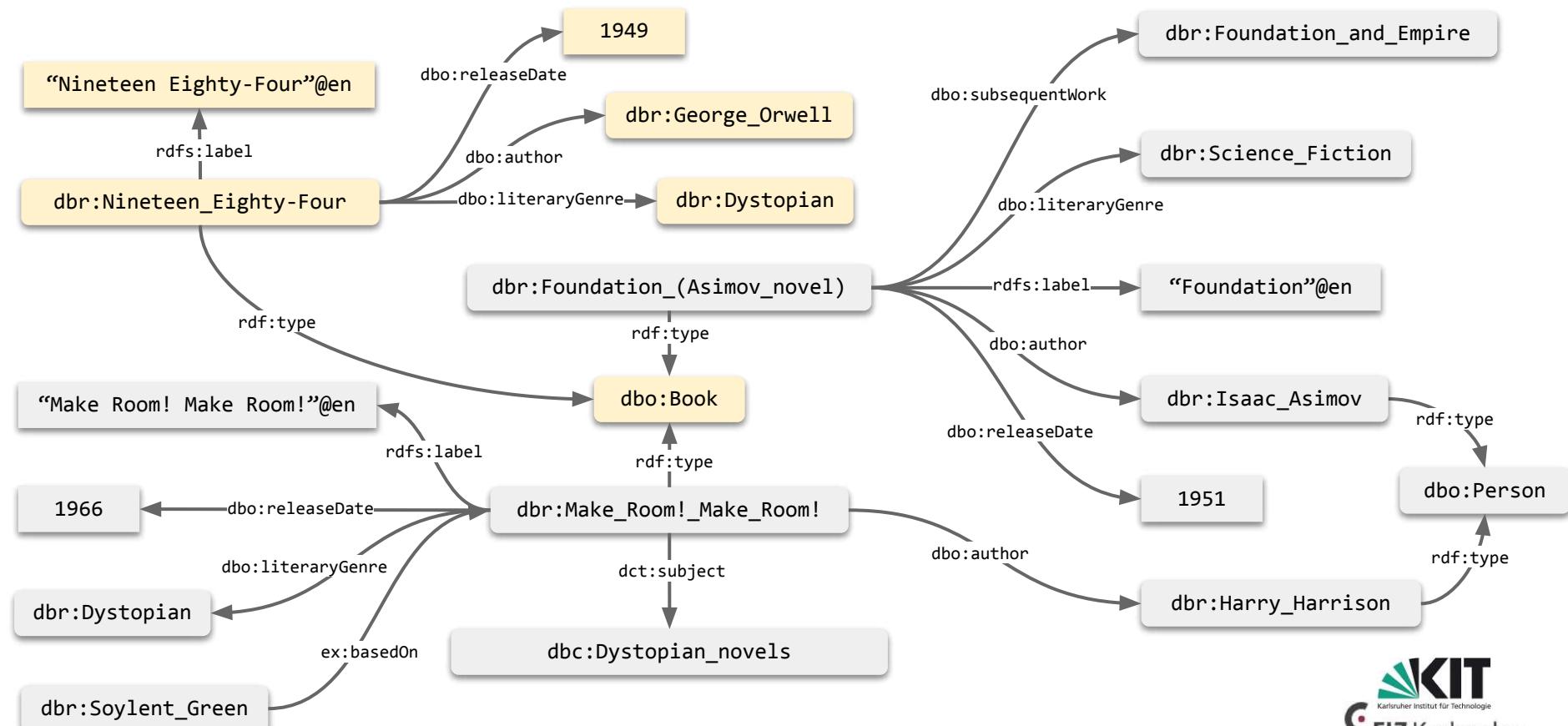
Copyright © 2023 [OpenLink Software](#)  
[Virtuoso](#) version 08.03.3326 (b5ffaadfc9) on Linux (x86\_64-generic-linux-glibc25) Single Server Edition (61 GB total memory, 42 GB memory in use)

<http://dbpedia.org/sparql>

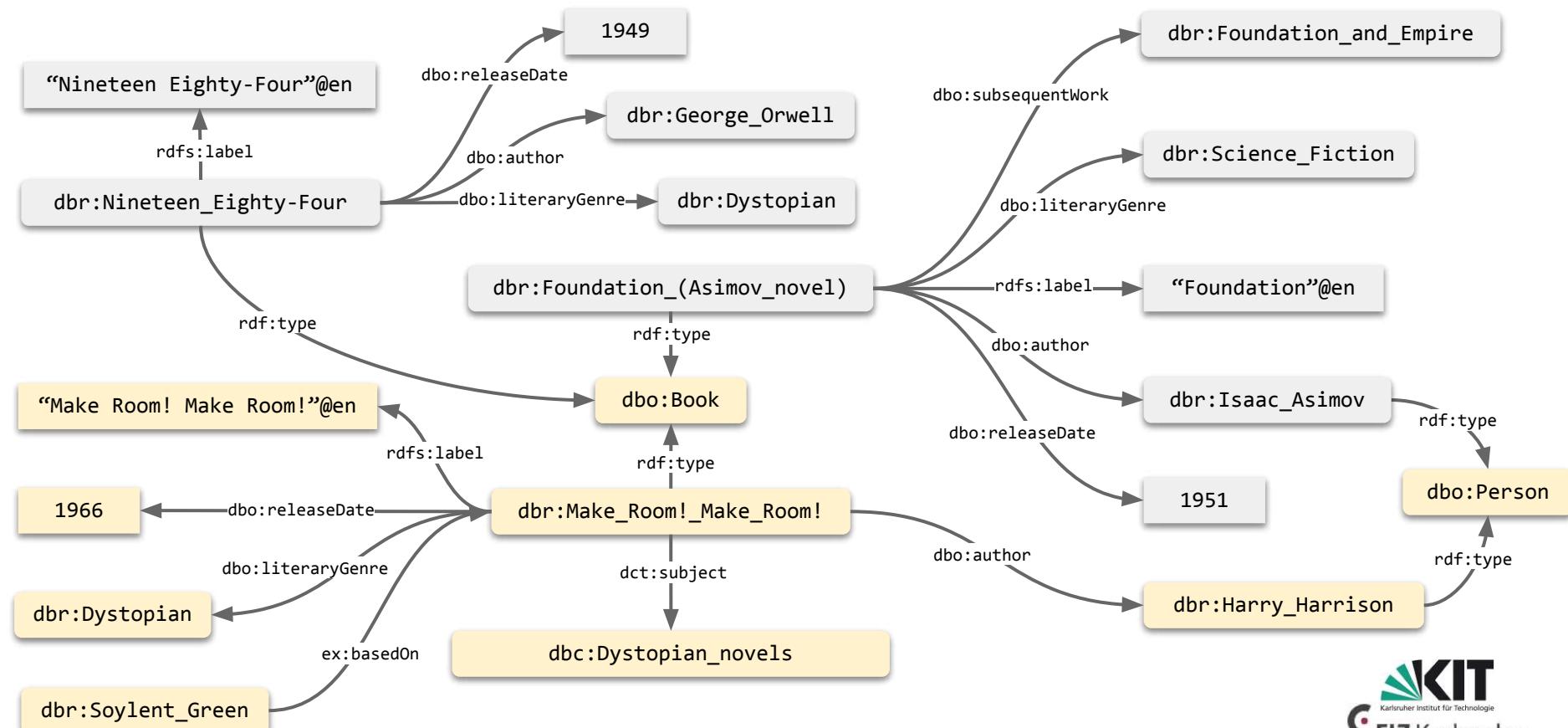
# Querying an RDF-based Knowledge Graph



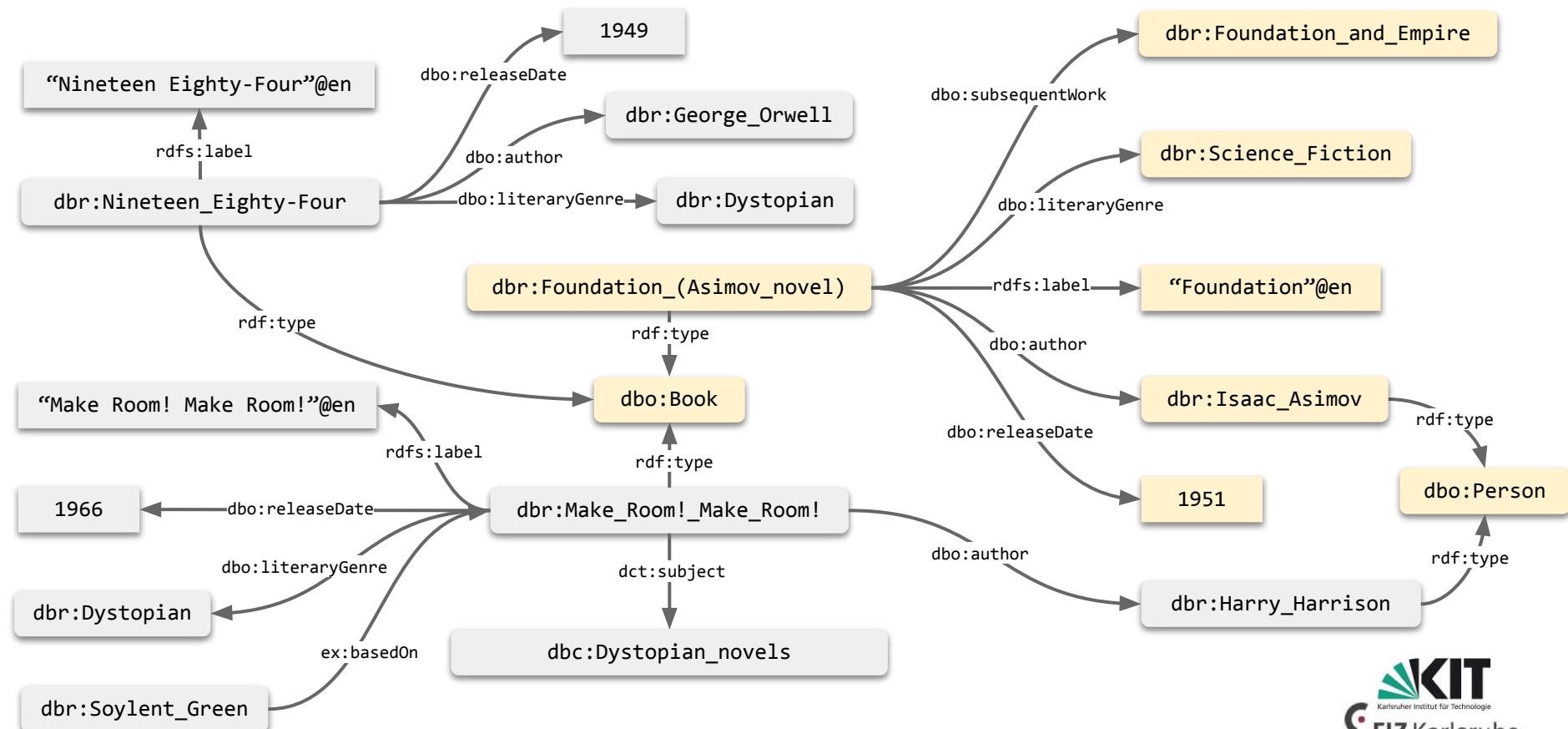
# Querying an RDF-based Knowledge Graph



# Querying an RDF-based Knowledge Graph



# Querying an RDF-based Knowledge Graph



# For Queries We Need Variables

- SPARQL **Variables** are bound to RDF terms,  
e.g. `?title, ?author, ?date`
- In the same way as in SQL,  
a **Query for variables** is performed via **SELECT statement**,  
e.g. `SELECT ?title ?author ?date`
- A SELECT statement returns query results as a **table**.

SPARQL Query

<code>?title</code>	<code>?author</code>	<code>?date</code>
Nineteen Eighty-Four	George Orwell	1948
Foundation (Novel)	Isaac Asimov	2006
Make Room! Make Room!	Harry Harrison	1966

SPARQL Result

# SPARQL Graph Pattern Matching

- SPARQL is based on
  - (1) **RDF Turtle serialization** and (2) **basic graph pattern matching**.
- A **Graph Pattern (Triple Pattern)** is a RDF Triple that contains variables at any arbitrary place (Subject, Property, Object).

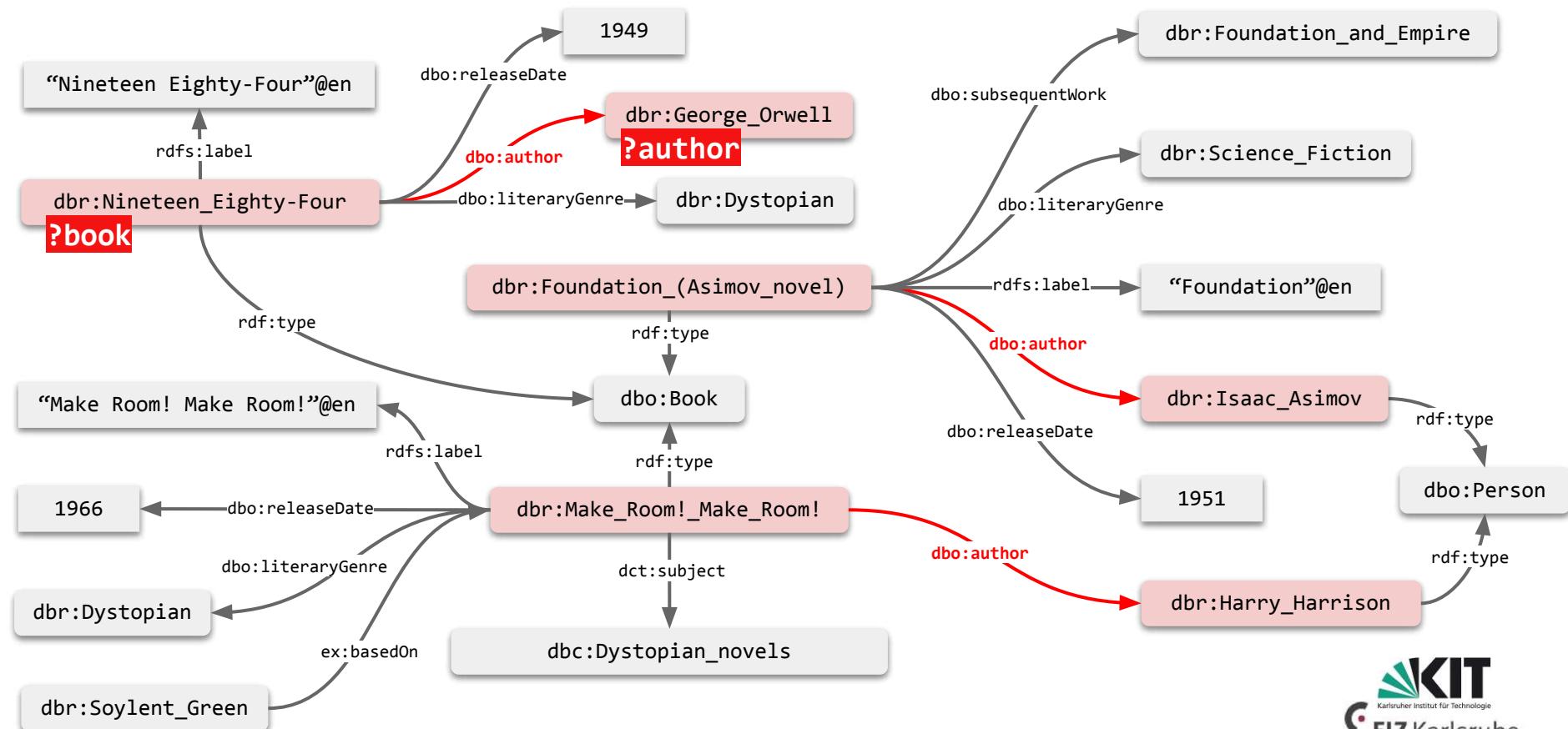
**Graph Pattern = Turtle + Variables**

- Example:

*Look for **books** and their **authors** (via property **dbo:author**):*



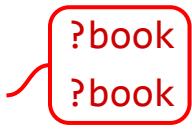
# Querying an RDF-based Knowledge Graph



# SPARQL Complex Graph Pattern Matching

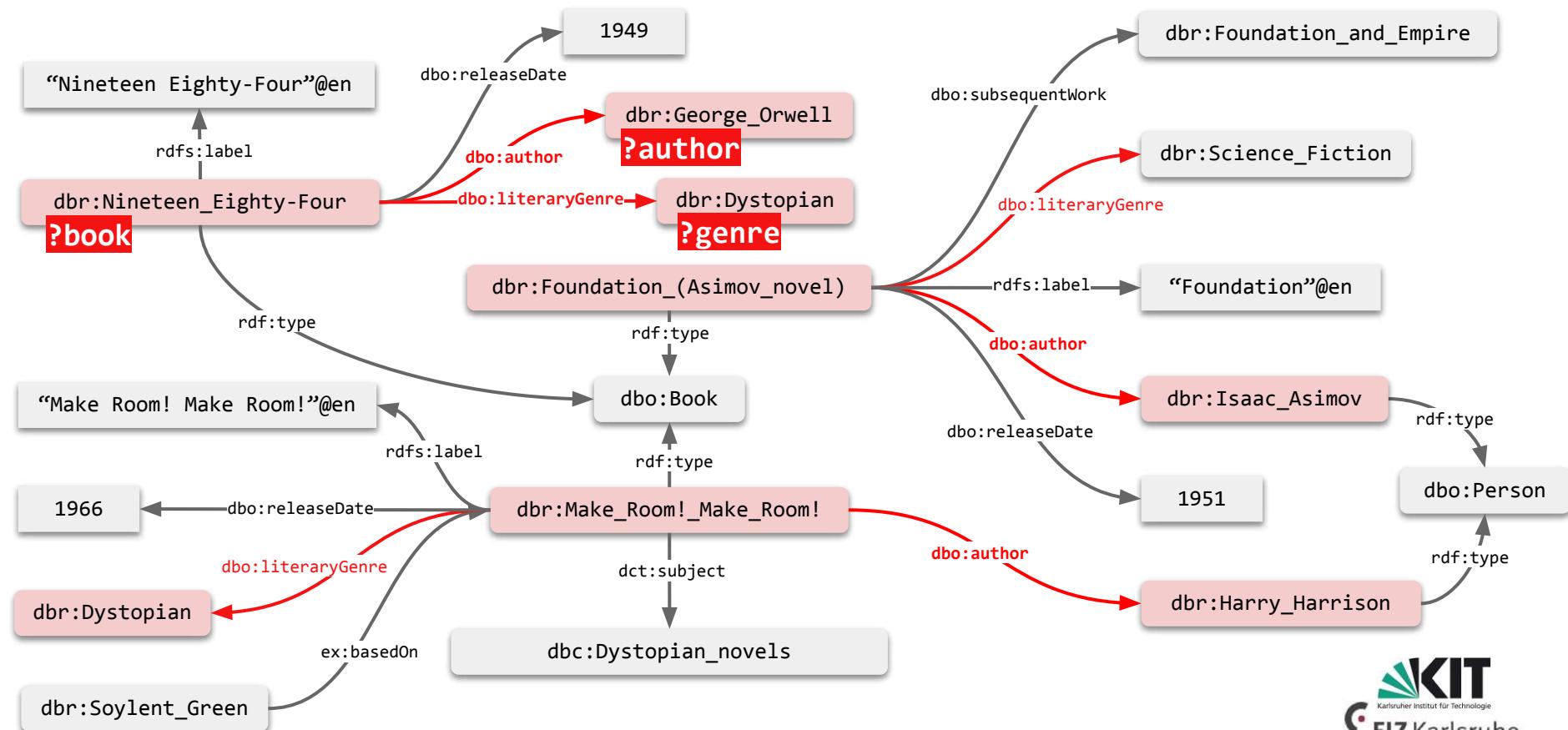
- SPARQL Graph Pattern can be combined to form **complex (conjunctive) queries** for RDF graph traversal.
- *Find books, their authors, and their literary genres:*

the same  
book(s)



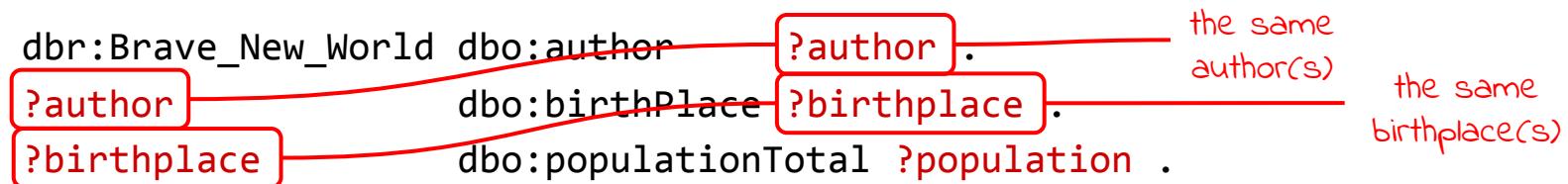
```
?book dbo:author ?author .  
?book dbo:literaryGenre ?genre .
```

# Querying an RDF-based Knowledge Graph



# SPARQL Complex Graph Pattern Matching

- SPARQL Graph Pattern can be combined to form **complex (conjunctive) queries** for RDF graph traversal.
- *Given a specific book URI, find its author(s), the birthplace(s) of its author(s), including the number of population of the birthplace(s):*



# SPARQL Query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

specifies namespaces

```
SELECT ?author_name ?title
```

specifies output variables

```
FROM <http://dbpedia.org/>
```

specifies graph to be queried

```
WHERE {
```

```
?author rdf:type dbo:Writer .
?author rdfs:label ?author_name .
?author dbo:notableWork ?work .
?work rdfs:label ?title .
```

specifies graph pattern  
to be matched

```
}
```



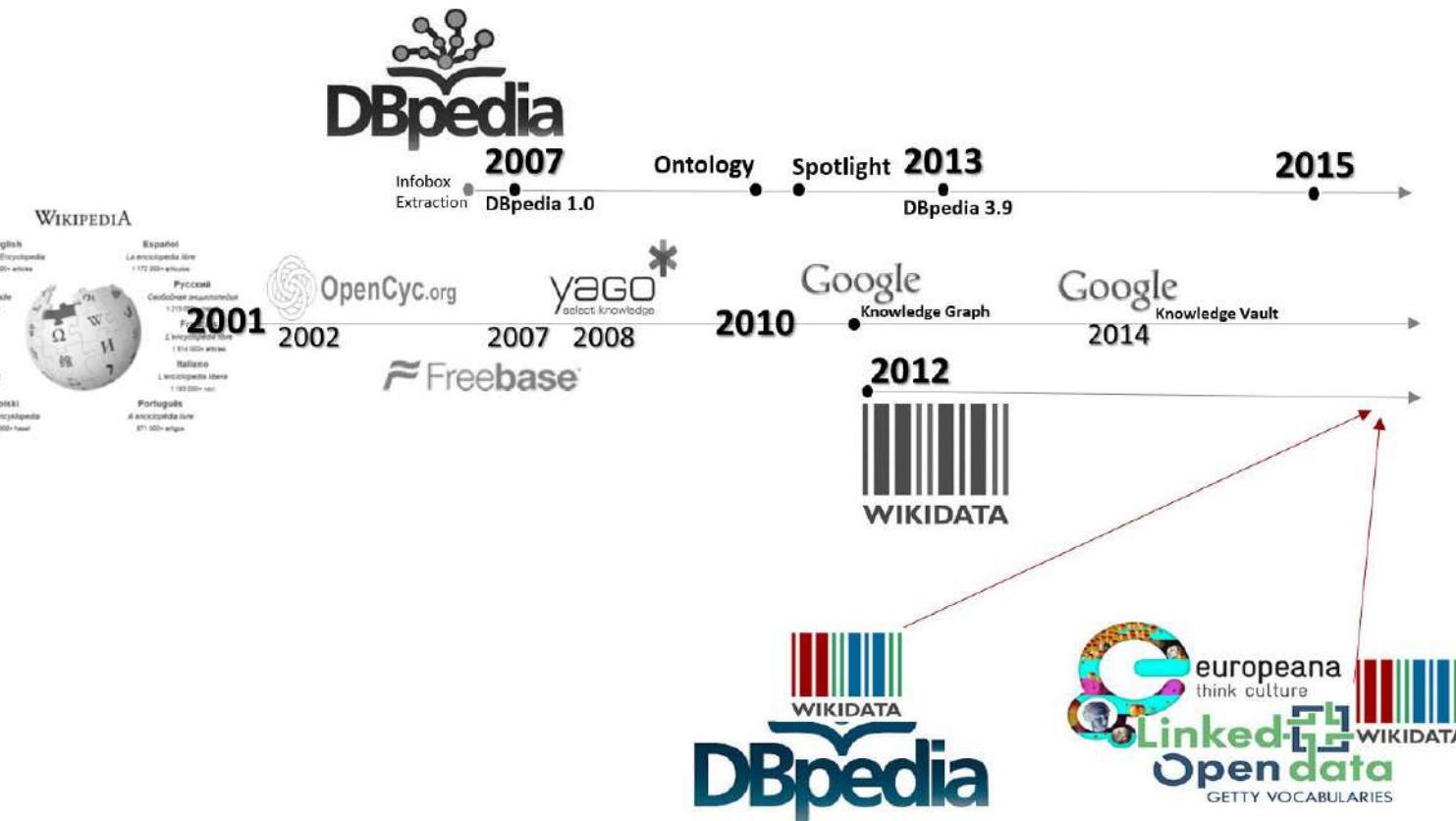
[query SPARQL endpoint](#)

# DBpedia and the Web of Data



A central hub in the Web of Data.

# DBpedia and the Web of Data



# DBpedia Stats

English version of the DBpedia Knowledge Graph (snapshot 2022/09)

- Contains 850 million facts (RDF triples),
- Connected via 55,000 properties,
- 1,377 of which are grounded in the DBpedia [ontology](#),
- including 1.8 million persons,
- 757,000 places (including 597,000 populated places),
- 619,000 creative works
  - including 158,000 music albums,
  - 147,000 films and
  - 25,000 video games,
- 350,000 organizations
  - including 88,000 companies and 64,000 educational institutions,
- 1.9 million species and
- 10,000 diseases.



<https://www.dbpedia.org/blog/dbpedia-snapshot-2022-09-release/>

102

# From Wikipedia to DBpedia

[https://en.wikipedia.org/wiki/Nineteen\\_Eighty-Four](https://en.wikipedia.org/wiki/Nineteen_Eighty-Four)



**WIKIPEDIA**  
The Free Encyclopedia

*Nineteen Eighty-Four*

From Wikipedia, the free encyclopedia

This article is about the 1949 novel by George Orwell. For the year, see 1984. For other uses, see 1984 (disambiguation).

**Nineteen Eighty-Four** (also published as **1984**) is a dystopian social science fiction novel and cautionary tale by English writer George Orwell. It was published on 8 June 1949 by Secker & Warburg as Orwell's ninth and final book completed in his lifetime. Thematically, it centres on the consequences of totalitarianism, mass surveillance and repressive regimentation of people and behaviours within society.<sup>[2][3]</sup> Orwell, a democratic socialist, modelled the authoritarian state in the novel on Stalinist Russia and Nazi Germany.<sup>[2][3][4]</sup> More broadly, the novel examines the role of truth and facts within societies and the ways in which they can be manipulated.

The story takes place in an imagined future in the year 1984, when much of the world is in perpetual war. Great Britain, now known as Airstrip One, has become a province of the totalitarian superstate Oceania, which is led by Big Brother, a dictatorial leader supported by an intense cult of personality manufactured by the Party's Thought Police. Through the Ministry of Truth, the Party engages in omnipresent government surveillance, historical negationism, and constant propaganda to persecute individual and independent thinking.<sup>[5]</sup>

The protagonist, Winston Smith, is a diligent mid-level worker at the Ministry of Truth who secretly hates the Party and dreams of rebellion. He keeps a forbidden diary and begins a relationship with his colleague Julia, and they learn about a shadowy resistance group called the Brotherhood. However, their contact with the Brotherhood turns out to be a Party agent, and Smith is arrested. He is subjected to months of psychological manipulation and torture by the Ministry of Love and is released once he has come to love Big Brother.

**Nineteen Eighty-Four** has become a classic literary example of political and dystopian fiction. It also popularised the term "Orwellian" as an adjective, with many terms used in the novel entering common usage, including "Big Brother", "doublespeak", "Thought Police", "thoughtcrime", "Newspeak", and "2 + 2 = 5". Parallels have been drawn between the novel's subject matter and real life instances of totalitarianism, mass surveillance, and violations of freedom of expression among other themes.<sup>[6][7][8]</sup> Orwell described his book as a "satire,<sup>[9]</sup> and a display of the "perversions to which a centralised economy is liable," while also stating he believed "that something resembling it could arrive."<sup>[10]</sup> *Time* included the novel on its list of the 100 best English-language novels from 1923 to 2005,<sup>[10]</sup> and it was placed on the Modern Library's 100 Best Novels list, reaching number 13 on the editors' list and number 6 on the readers' list.<sup>[11]</sup> In 2003, it was listed at number eight on *The Big Read* survey by the BBC.<sup>[12]</sup>

**Writing and publication** [edit]

**Idea** [edit]

The Orwell Archive at University College London contains undated notes about ideas evolved into

<i>Nineteen Eighty-Four</i>	
Author	George Orwell
Cover artist	Michael Kenner
Country	United Kingdom
Language	English
Genre	Dystopian · political fiction · social science fiction
Set in	London, Airstrip One, Oceania
Publisher	Secker & Warburg
Publication date	8 June 1949
Media type	Print (hardback and paperback)
Pages	328
OCLC	470015866 <sup>[7]</sup>
Dewey Decimal	823.912 <sup>[11]</sup>
Encoded by	Animal Farm



[http://dbpedia.org/resource/Nineteen\\_Eighty-Four](http://dbpedia.org/resource/Nineteen_Eighty-Four)

# From Wikipedia to DBpedia

[http://dbpedia.org/resource/Nineteen\\_Eighty-Four](http://dbpedia.org/resource/Nineteen_Eighty-Four)

DBpedia Browse using ▾ Formats ▾ Faceted Browser Sparql Endpoint

## About: Nineteen Eighty-Four

An Entity of Type: [book](#), from Named Graph: [http://dbpedia.org](#), within Data Space: [dbpedia.org](#)

Nineteen Eighty-Four (also stylised as 1984) is a dystopian social science fiction novel and cautionary tale written by the English writer George Orwell. It was published on 8 June 1949 by Secker & Warburg as Orwell's ninth and final book completed in his lifetime. Thematically, it centres on the consequences of totalitarianism, mass surveillance and repressive regimentation of people and behaviours within society. Orwell, a democratic socialist, modelled the totalitarian government in the novel after Stalinist Russia and Nazi Germany. More broadly, the novel examines the role of truth and facts within politics and the ways in which they are manipulated.



Property	Value
<a href="#">dbo:abstract</a>	<ul style="list-style-type: none"> <li>• 1984 (Originaltitel: Nineteen Eighty-Four, deutscher Alternativtitel: Neunzehnhundertvierundachtzig), geschrieben von 1946 bis 1948 und erschienen im Juni 1949, ist ein dystopischer Roman von George Orwell (eigentlich Eric Arthur Blair), in dem ein totalitärer Überwachungsstaat im Jahr 1984 dargestellt wird. Hauptperson der Handlung ist Winston Smith, ein einfaches Mitglied der diktatorisch herrschenden, fiktiven Staatspartei Sozialistische Partei Englands (orig. Ingsoc). Der allgegenwärtigen Überwachung zum Trotz will Smith seine Privatsphäre sichern und etwas über die real geschehene Vergangenheit erfahren, die von der Partei durch umfangreiche Geschichtsfälschung verheimlicht wird. Dadurch gerät er mit dem System in Konflikt, das ihn gefangen nimmt, foltert und einer Gehirnwäsche unterzieht. Orwell begann mit dem Verfassen des Buches im Jahr 1946 während seines Aufenthaltes auf der Insel Jura vor der Küste Schottlands und stellte es Ende 1948 fertig. Der Titel enthält den Zahlendreher der Jahreszahl 1948 zu 1984 als Anspielung auf eine zwar damals noch fern erscheinende, aber (ähnlich wie Orwells vorangegangener Roman <i>Farm der Tiere</i>) doch eng mit der damaligen Gegenwart verknüpfte Zukunft. Die Erstausgabe des Buches kam in London am 8. Juni 1949 in den Verkauf. Der Roman wird oft dann zitiert bzw. sein Titel oder der Name Orwell genannt, wenn es darum geht, staatliche Überwachungsmaßnahmen kritisch zu kommentieren oder auf Tendenzen zu einem Überwachungsstaat hinzuweisen. (de)</li> <li>• Nineteen Eighty-Four (also stylised as 1984) is a dystopian social science fiction novel and cautionary tale written by the English writer George Orwell. It was published on 8 June 1949 by Secker &amp; Warburg as Orwell's ninth and final book completed in his lifetime. Thematically, it centres on the consequences of totalitarianism, mass surveillance and repressive regimentation of people and behaviours within society. Orwell, a democratic socialist, modelled the totalitarian government in the novel after Stalinist Russia and Nazi Germany. More broadly, the novel examines the role of truth and facts within politics and the ways in which they are manipulated. The story takes place in an imagined future, the year 1984, when much of the world has fallen victim to perpetual war, omnipresent government surveillance, historical negationism, and propaganda. Great Britain, known as Airstrip One, has become a province of the totalitarian superstate Oceania, ruled by the Party, who employ the Thought Police to persecute individuality and independent thinking. Big Brother, the dictatorial leader of Oceania, enjoys an intense cult of personality, manufactured by the Party's excessive brainwashing techniques. The protagonist, Winston Smith, is a diligent and skillful rank-and-</li> </ul>

# DBpedia Naming Conventions

[https://en.wikipedia.org/wiki/Nineteen\\_Eighty-Four](https://en.wikipedia.org/wiki/Nineteen_Eighty-Four)



**WIKIPEDIA**  
The Free Encyclopedia

---

[http://dbpedia.org/resource/Nineteen\\_Eighty-Four](http://dbpedia.org/resource/Nineteen_Eighty-Four)

Entity Identifier

[http://dbpedia.org/page/Nineteen\\_Eighty-Four](http://dbpedia.org/page/Nineteen_Eighty-Four)

HTML version

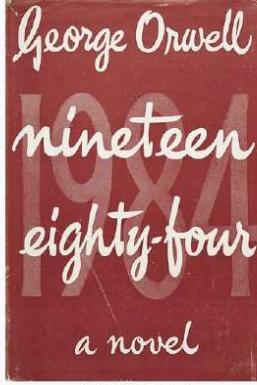
[http://dbpedia.org/data/Nineteen\\_Eighty-Four](http://dbpedia.org/data/Nineteen_Eighty-Four)

RDF/XML version



# Wikipedia Infoboxes

*Nineteen Eighty-Four*



First-edition cover

<b>Author</b>	George Orwell
<b>Cover artist</b>	Michael Kennar
<b>Country</b>	United Kingdom
<b>Language</b>	English
<b>Genre</b>	Dystopian · political fiction · social science fiction
<b>Set in</b>	London, Airstrip One, Oceania
<b>Publisher</b>	Secker & Warburg
<b>Publication date</b>	8 June 1949
<b>Media type</b>	Print (hardback and paperback)
<b>Pages</b>	328
<b>OLC</b>	470015866
<b>Dewey Decimal</b>	823.912 <sup>[1]</sup>
<b>Preceded by</b>	<i>Animal Farm</i>

**George Orwell**



Orwell's press card portrait, 1943

<b>Born</b>	Eric Arthur Blair 25 June 1903 Mothari, Bengal Presidency, British India
<b>Died</b>	21 January 1950 (aged 46) London, England
<b>Resting place</b>	All Saints' Church, Sutton Courtenay, Oxfordshire, England
<b>Education</b>	Eton College
<b>Occupations</b>	Novelist, essayist, journalist, literary critic
<b>Political party</b>	ILP (from 1938)
<b>Spouses</b>	Eileen O'Shaughnessy (m., 1936; died 1945) Sonia Brownell (m., 1949)
<b>Children</b>	Richard Blair
<b>Writing career</b>	Writing career
<b>Pen name</b>	George Orwell
<b>Genre</b>	Dystopia, roman à clef, satire
<b>Subjects</b>	Anti-fascism, anti-Stalinism, anarchism, democratic socialism, literary criticism, journalism, polemic
<b>Years active</b>	1928–1950
<b>Signature</b>	

**1984**



GEORGE ORWELL'S TERRIFYING VISION COMES TO THE SCREEN.  
**1984**

UK theatrical release poster

<b>Directed by</b>	Michael Radford
<b>Screenplay by</b>	Michael Radford
<b>Based on</b>	<i>Nineteen Eighty-Four</i> by George Orwell
<b>Produced by</b>	Simon Perry
<b>Starring</b>	John Hurt Richard Burton Suzanna Hamilton Cyril Cusack
<b>Cinematography</b>	Roger Deakins
<b>Edited by</b>	Tom Priestley
<b>Music by</b>	Dominic Muldowney Eurythmics
<b>Production companies</b>	Virgin Films Umbrella-Rosenblum Films Atlantic Entertainment Group
<b>Distributed by</b>	20th Century Fox
<b>Release date</b>	10 October 1984 (United Kingdom)
<b>Running time</b>	110 minutes <sup>[1][note 1][2]</sup>
<b>Country</b>	United Kingdom <sup>[3]</sup>
<b>Language</b>	English
<b>Budget</b>	£5.5 million <sup>[4]</sup>
<b>Box office</b>	\$8.4 million (United States) <sup>[5]</sup>

**BBC Television**



Logo used since 2021

<b>Type</b>	Subsidiary
<b>Industry</b>	Television
<b>Headquarters</b>	Broadcasting House, London MediaCityUK, Salford
<b>Area served</b>	Worldwide
<b>Services</b>	Television broadcasting
<b>Parent</b>	BBC
<b>Website</b>	<a href="#">BBC iPlayer</a>

The "Television Symbol", known informally as the "Bats Wings", was the first BBC Television Service ident. It was created by Abram Games and was used from 1953 to 1960.<sup>[1]</sup>



**Hail to the Thief**



Studio album by Radiohead

<b>Released</b>	9 June 2003
<b>Recorded</b>	September 2002 – February 2003
<b>Studio</b>	Ocean Way, Hollywood Radiohead studio, Oxfordshire
<b>Genre</b>	Alternative rock · art rock · experimental rock · electronic rock
<b>Length</b>	56:35
<b>Label</b>	Parlophone · Capitol
<b>Producer</b>	Nigel Godrich · Radiohead
<b>Radiohead chronology</b>	
<i>I Might Be</i>	<b>Hail to the Thief</b>
<i>Wrong: Live</i>	<i>COM LAG</i>
<i>Recordings</i>	<i>Thief</i> (2plus2ive) (2003)
	(2004)
<b>Singles from Hail to the Thief</b>	
1. "There There"	Released: 26 May 2003
2. "Go to Sleep"	Released: 18 August 2003
3. "2 + 2 = 5"	Released: 17 November 2003

# DBpedia Infobox Extraction – Infobox Properties

*Nineteen Eighty-Four*

The figure shows an infobox for the novel "Nineteen Eighty-Four" by George Orwell. The infobox includes the book cover, the title, author, and various properties with their values. Arrows point from specific properties to their corresponding SPARQL URIs.

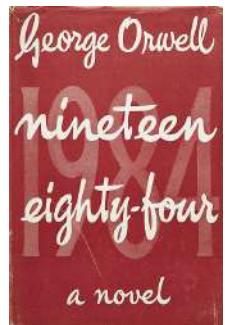
First-edition cover	
Author	George Orwell
Cover artist	Michael Kennar
Country	United Kingdom
Language	English
Genre	Dystopian • political fiction • social science fiction
Set in	London, Airstrip One, Oceania
Publisher	Secker & Warburg
Publication	8 June 1949

Properties and their SPARQL URIs:

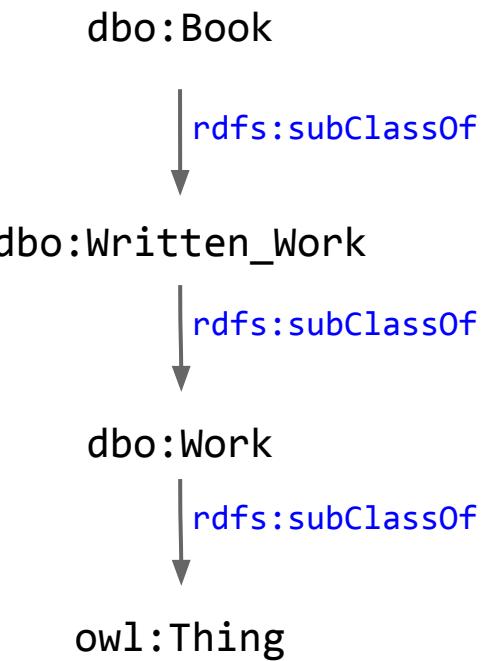
- Author → [http://dbpedia.org/resource/Nineteen\\_Eighty-Four](http://dbpedia.org/resource/Nineteen_Eighty-Four)
- Author → <http://dbpedia.org/property/author>
- Author → [http://dbpedia.org/resource/George\\_Owell](http://dbpedia.org/resource/George_Owell)

# DBpedia Ontology Classes

dbr:Nineteen\_Eighty-Four



rdf:type



<http://mappings.dbpedia.org/server/ontology/classes/>

- Album (edit)
- ArtistDiscography (edit)
- ClassicalMusicComposition (edit)
- Musical (edit)
- NationalAnthem (edit)
- Opera (edit)
- Single (edit)
- Song (edit)
  - EurovisionSongContestEntry (edit)
- RadioProgram (edit)
- Software (edit)
  - VideoGame (edit)
- TelevisionEpisode (edit)
- TelevisionSeason (edit)
- TelevisionShow (edit)
- Website (edit)
- WrittenWork (edit)
  - Annotation (edit)
    - Reference (edit)
  - Article (edit)
  - Book (edit)
  - Comic (edit)
    - ComicStrip (edit)
    - Manga (edit)
    - Manhua (edit)
    - Manhwa (edit)
  - Drama (edit)
  - Law (edit)
  - Letter (edit)
  - MultiVolumePublication (edit)
  - PeriodicalLiterature (edit)
    - AcademicJournal (edit)
    - Magazine (edit)
    - Newspaper (edit)
    - UndergroundJournal (edit)
  - Play (edit)
  - Poem (edit)
  - Quote (edit)
  - Resume (edit)
  - StatedResolution (edit)
  - Treaty (edit)
- foaf:Document (edit)
- cidoccrm:E4\_Period (edit)

# DBpedia SPARQL Endpoint

SPARQL Query Editor   About   Tables ▾

Conductor   Facet Browser   Permalink

Extensions: [cxml](#) [save to dav](#) [sponge](#) User: SPARQL

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where { [] a ?Concept} LIMIT 100
```

Results Format

Execution timeout  milliseconds

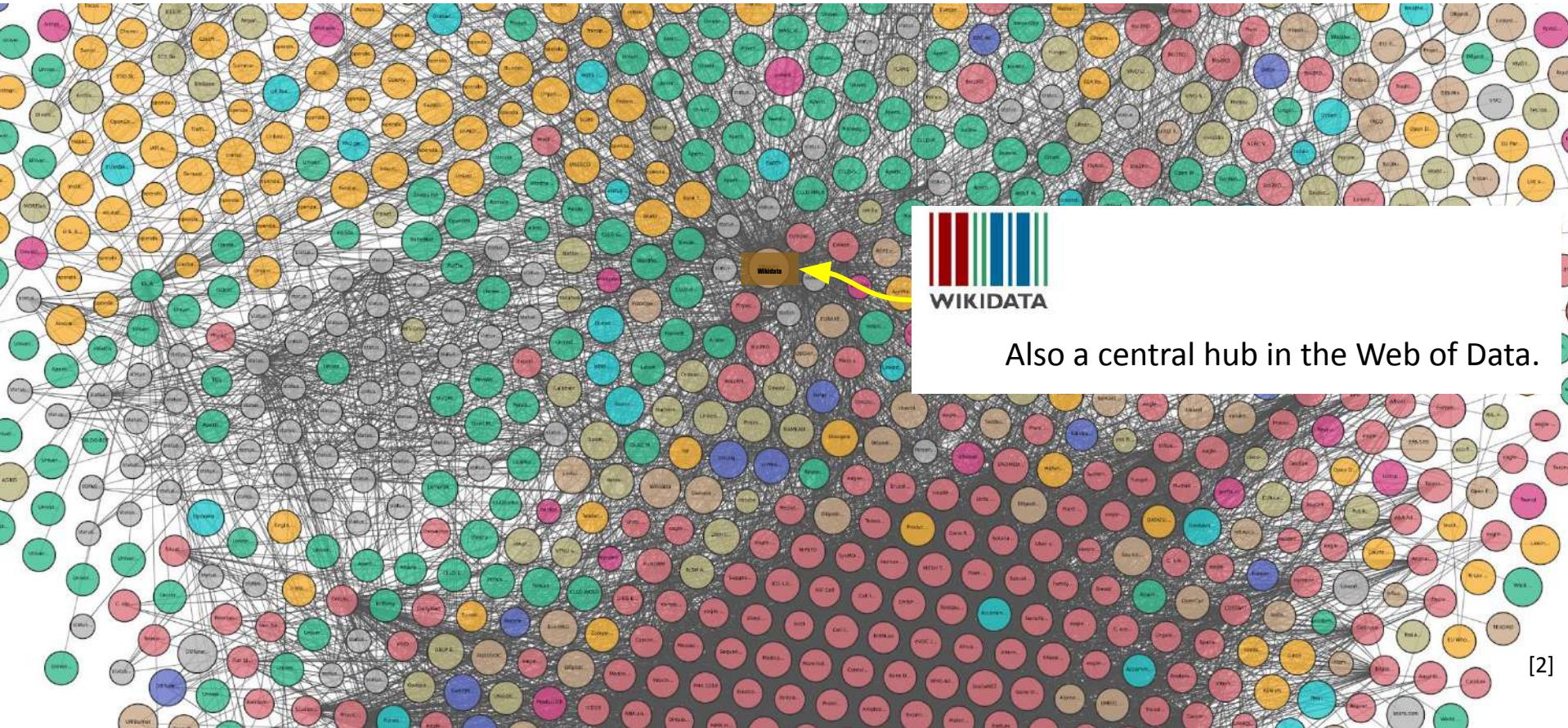
Options

- Strict checking of void variables
- Strict checking of variable names used in multiple clauses but not logically connected to each other
- Suppress errors on wrong geometries and errors on geometrical operators (failed operations will return NULL)
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)



<http://dbpedia.org/sparql>

# Wikidata and the Web of Data



# What is Wikidata?

Collaboratively edited structured data (knowledge graph) operated by the Wikimedia Foundation (started in 2012).

The screenshot shows the Wikidata item page for George Orwell (Q3335). The page includes:

- Statements:**
  - Instance of: human (with 3 references)
  - Image: A portrait of George Orwell.
  - sex or gender: male (with 6 references)
  - country of citizenship: United Kingdom (start time: 1927)
- Wikipedia (195 entities):** A sidebar listing various language equivalents for George Orwell, such as "George Orwell" in English, French, German, Spanish, and many others.

<https://www.wikidata.org/wiki/Q3335>

- > 101M entities (*Feb 2023*)
  - > 10.4M persons
  - > 2.7M populated places
  - > 5.1M architectural structures
  - > 1.1M events
  - > 1.2M chemical compounds
  - > 350K movies
  - > 230K books
  - > 8.4M astronomical objects
  - > 22.5M scholarly articles
- > 14B triples
- ~ 24K active users

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

# George Orwell (Q3335)

English author and journalist (1903–1950)  
Eric Blair | P. S. Burton | Eric Arthur Blair | John Freeman

In more languages

## Statements

instance of

human

+ 3 references

reference

image



George Orwell, c. 1940 (41928150381).jpg  
2,620 × 3,782; 7.36 MB

media legend Orwell c. 1940. (English)

+ 0 references

sex or gender

male

+ 6 references

value

country of citizenship

United Kingdom

start time

1927

end time

21 January 1950

+ 0 references

qualifiers

United Kingdom of Great Britain and Ireland

end time

1927

start time

25 June 1903 Gregorian

end cause

succession of states

+ 0 references

Wikipedia (130 entries)

af George Orwell

am გეორგ ბერტლი

an George Orwell

ar جورج أورويل

ary مجهود عجمي

azر جورج اورول

ast George Orwell

avk George Orwell

awa জর্জ ওরউল

azb جورج اورول

az George Orwell

bcy جورج اورول

be Джордж Оруэл

bg Джордж Оруэл

bn জর্জ ওরউল

br George Orwell

bs George Orwell

ca George Orwell

cdo George Orwell

ckb جورج اورول

co George Orwell

cs George Orwell

cv Джордж Оруэлл

cy George Orwell

da George Orwell

de George Orwell

el Τζωρτζ Ορουελ

en George Orwell

eo George Orwell

es George Orwell

et George Orwell

eu George Orwell

fa جرج اورول

vro Orwell George

fi George Orwell

fr George Orwell

fy George Orwell

ga George Orwell

gd George Orwell

gl George Orwell

gom George Orwell

gv George Orwell

hak George Orwell

he ג'ורג' אורול

hi जर्ज ओरवल

hr George Orwell

id George Orwell

is George Orwell

it George Orwell

ja ジョージ・オーウェル

km George Orwell

kn ಜರ್ಜ್ ಓರ್ವಲ್

lt George Orwell

lv George Orwell

mk Џорџ Орвел

ml ജോർജ്ജ് ഓറ്വല്

mr George Orwell

nl George Orwell

pl George Orwell

pt George Orwell

ro George Orwell

ru Джордж Оруэлл

sk George Orwell

sl George Orwell

sr George Orwell

sv George Orwell

tr George Orwell

uk جورج اورول

vi George Orwell

zh 雅瑟·奧威爾

zh-hk 喬治·歐威爾

zh-hant 喬治·歐威爾

statements

# Wikidata SPARQL Endpoint

The screenshot shows the Wikidata Query Service interface. On the left, there is a vertical toolbar with various icons for navigating and managing queries. The main area contains a SPARQL query editor and a results table.

**SPARQL Query:**

```

1 SELECT ?booksLabel ?date ?firstline WHERE {
2 wd:Q208460 wdt:P50 ?author . # 1984 :author ?author
3 ?books wdt:P50 ?author ; # ?books :author ?author
4     wdt:P31 wd:Q7725634 ; # ?books :instanceOf :LiteraryWork
5     wdt:P577 ?date . # ?books :publicationDate ?date
6 OPTIONAL {?books wdt:P1922 ?firstline .} # ?books :firstLine ?firstline
7 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". } #get English labels
8 }

```

**Results Table:**

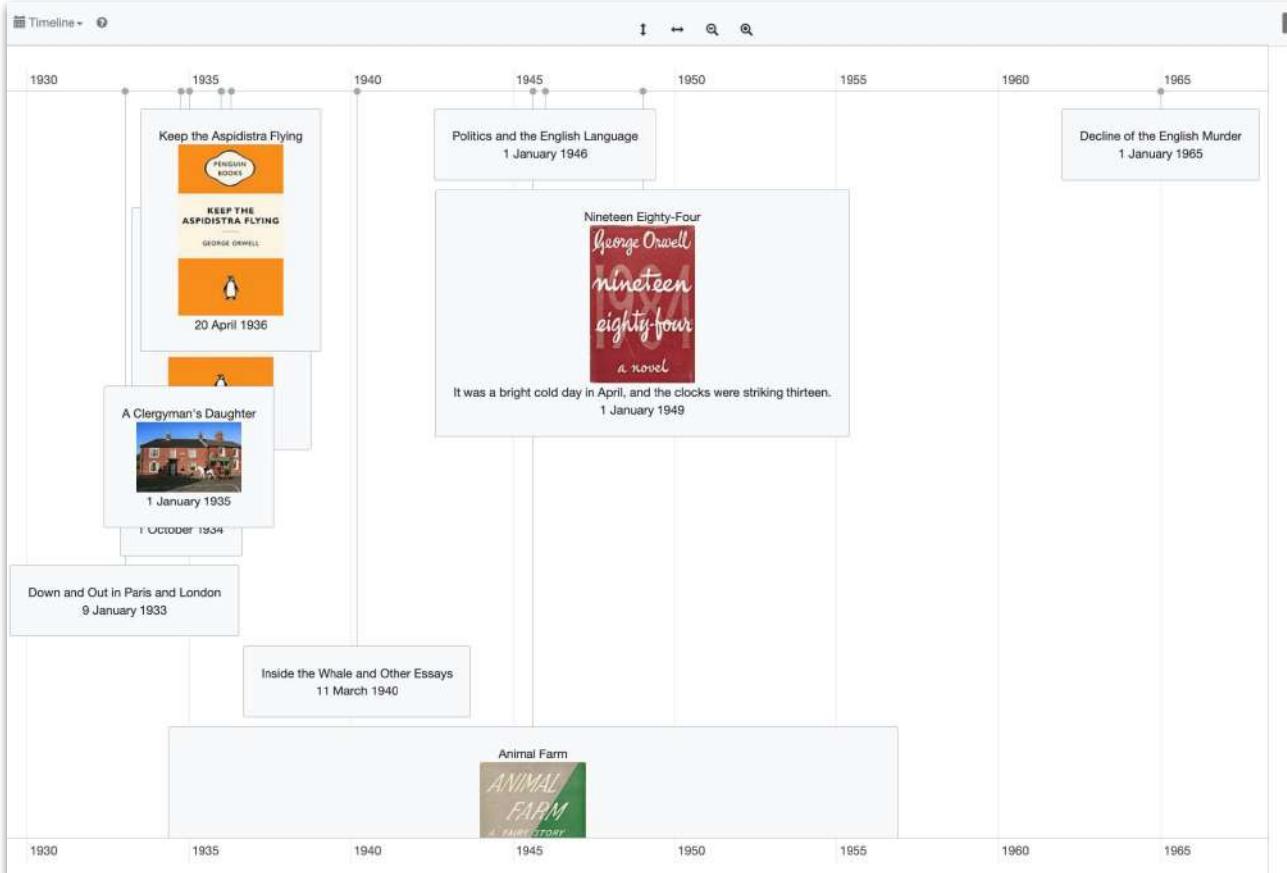
booksLabel	date	line
Nineteen Eighty-Four	1 January 1949	It was a bright cold day in April, and the clocks were striking thirteen.
Shooting an Elephant	1 January 1936	
Keep the Aspidistra Flying	1 January 1936	
Keep the Aspidistra Flying	20 April 1936	
Down and Out in Paris and London	9 January 1933	
Animal Farm	17 August 1945	Mr. Jones, of the Manor Farm, had locked the hen-houses for the night, but was too drunk to remember to shut the pop-holes.

Below the table, the results are summarized: **11 results in 629 ms**. There are also links for **Code**, **Download**, and **Link**.

[Wikidata example SPARQL query](#)

113

# Wikidata Visualizations



[Wikidata example SPARQL query](#)

# Wikidata is not a Knowledge Base

**Caution:  
WIKIDATA is not a (real)  
Knowledge Base!**

WIKIDATA is a Wiki-based large **structured database**.

The available Triple Store and SPARQL query service is only an addendum.

WIKIDATA is **not fully W3C compliant**,

i.e. no W3C compliant vocabulary (RDF, RDFS, OWL) or semantics is used!

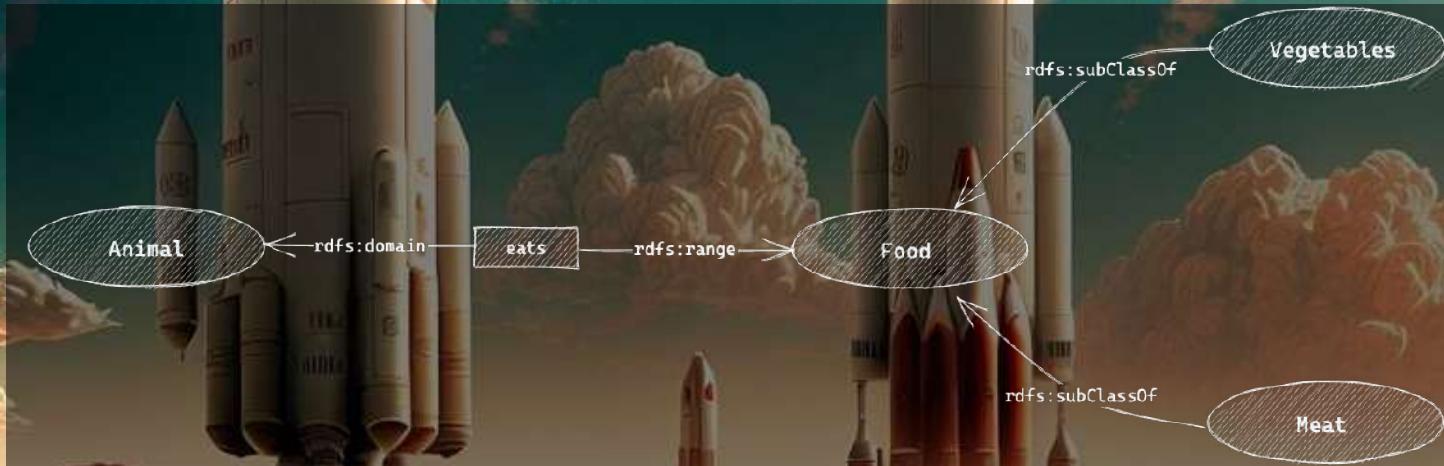


A surreal landscape on Mars. In the foreground, a black silhouette of a cow stands on a reddish-brown, rocky ground. In the background, several white rockets with red fins are launching from a launch pad, leaving thick, billowing orange and yellow smoke trails against a dark blue sky. One rocket is already in space, leaving a long, wispy trail. To the right, a large, reddish-orange planet, possibly Earth or Mars, is visible. The overall scene is a blend of science fiction and reality.

# Why RDF(S) is not sufficient

# Why RDF(S) is not sufficient

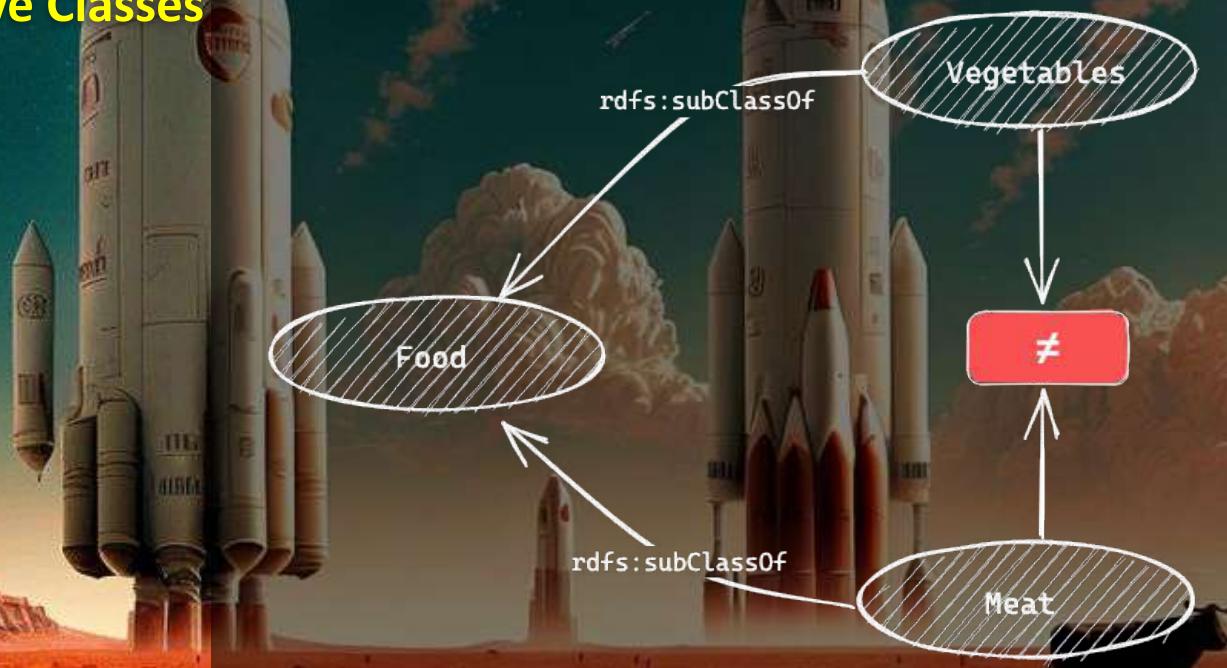
- Locality of Global Properties



- A Cow only eats vegetables
- Other animals only eat meat

# Why RDF(S) is not sufficient

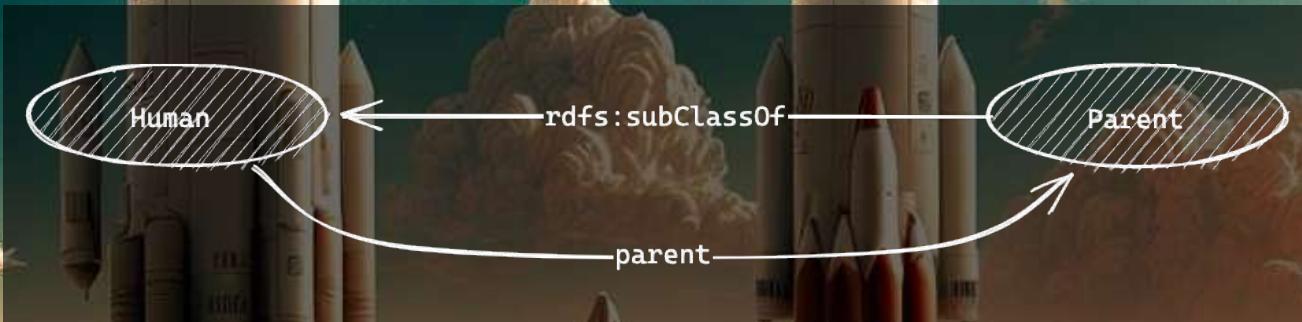
- **Disjunctive Classes**



- RDFS Subclass relation cannot express disjunctive class (subclass) membership

# Why RDF(S) is not sufficient

- Cardinality Constraints

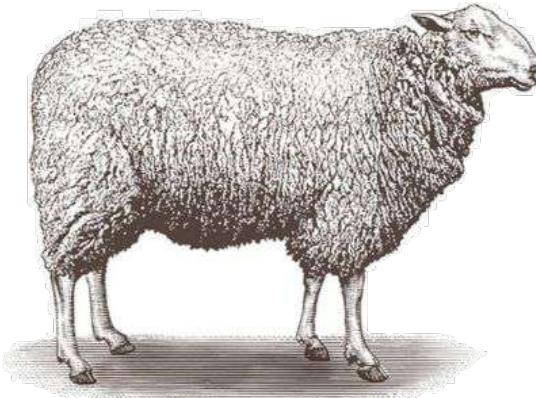


- Every human (usually) has two parents

# Open & Closed World Assumption – OWA vs CWA

A sheep is an animal with four legs.

- Answer under CWA Assumption:
- Answer under OWA assumption:



Question: Can sheep fly?

No, sheep can't fly.

No idea, but probably yes  
(according to our knowledge base).

- In the OWA, unless we have a statement (or we can infer) "*sheep can/cannot fly*" we return "*don't know*".
- In the real world, we are used to deal with incomplete information.

# Unique Name Assumption – UNA

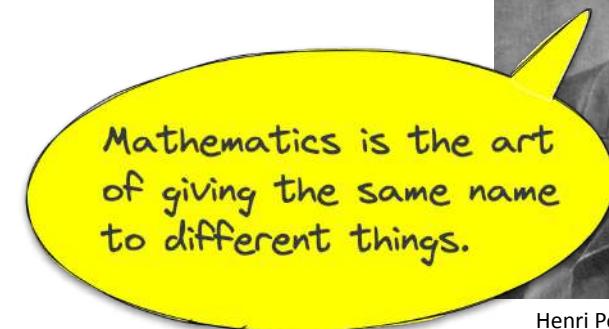
- In logics with **UNA**,  
**different names always refer to different entities in the world.**
- **OWL** does NOT support UNA  
(because of OWA).
- Consequences for the Semantic Web:

- Different entities have to be **declared to be different** (otherwise they are potentially identical).

`owl:differentFrom,`  
`owl:disjointWith`

- Identical entities also have to be **declared to be identical** (otherwise they are potentially different).

`owl:sameAs,`  
`owl:equivalentClass`



Henri Poincaré (1854–1912)



## 4. Ontologies as Key to Knowledge Representation

MARHHOF THE SMIARAS  
THE SHAOT iA SMAR

THEPESE OF SHACE MAAL!  
IN SPRAC TCF SPAARS

**ONTOLOGY** is the philosophical study of the nature of being, existence, or reality, as well as the basic categories of being and their relations...



# Ontology in Computer Science

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  
DIKW Pyramid

# Ontology in Computer Science

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:**

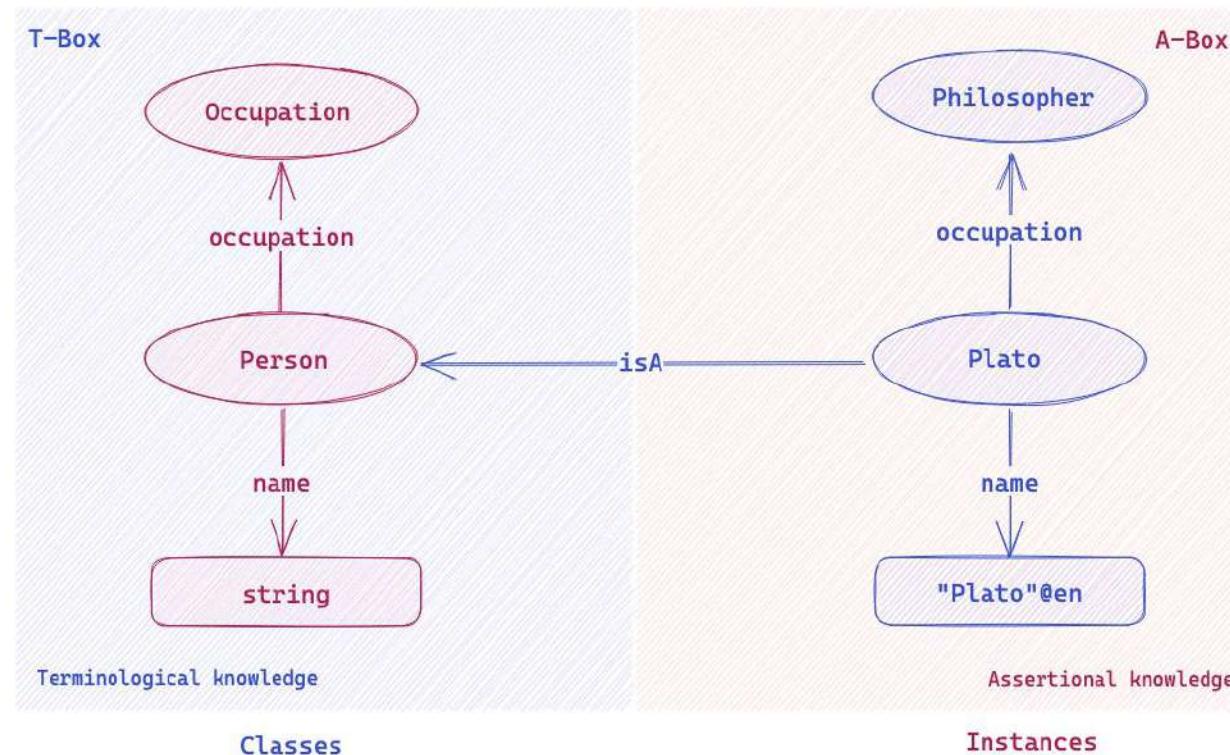
machine understandable

**shared:**

consensus about ontology

# How to represent Ontologies

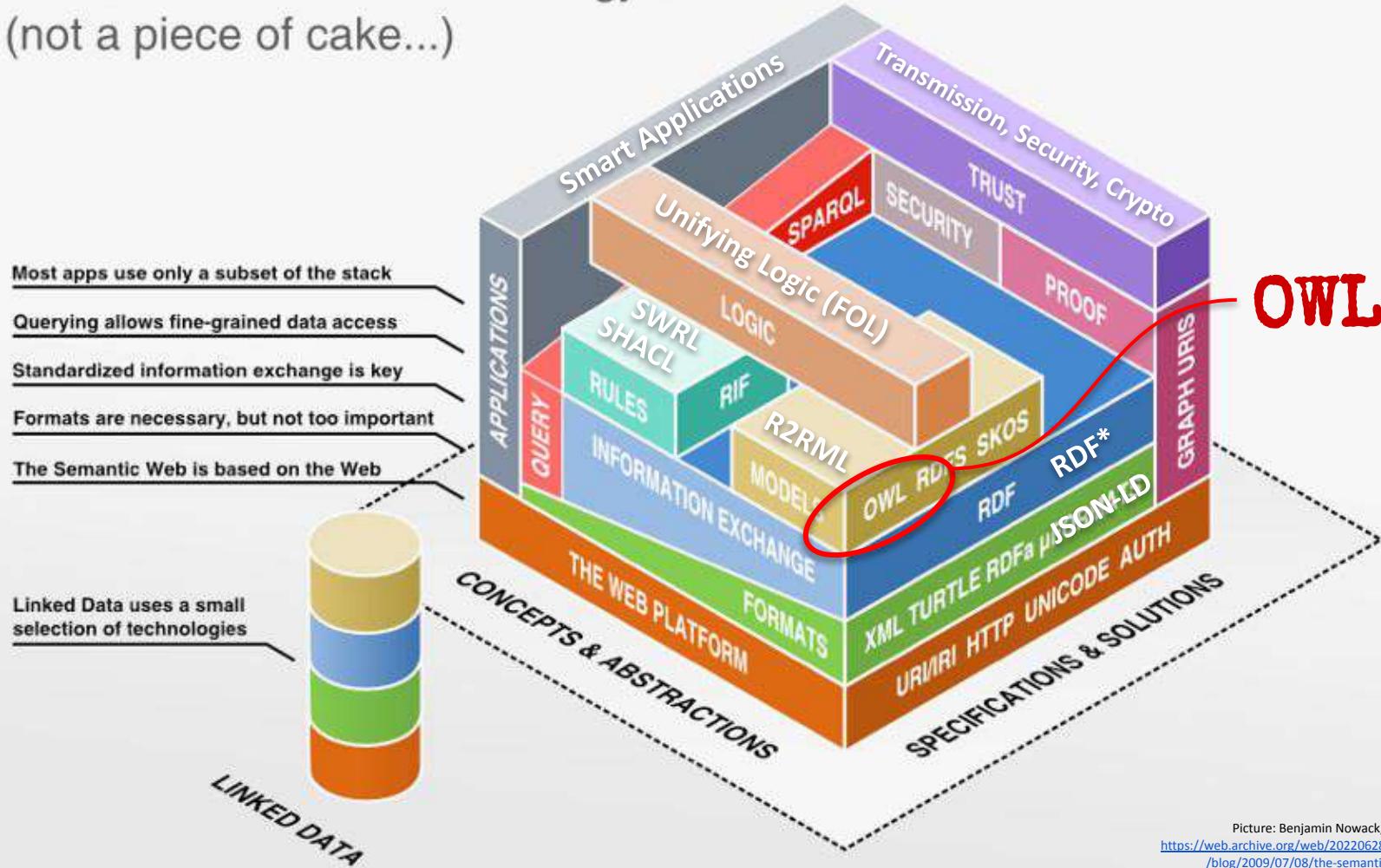
**Instances** describe individuals of an ontology.





# The Web Ontology Language OWL

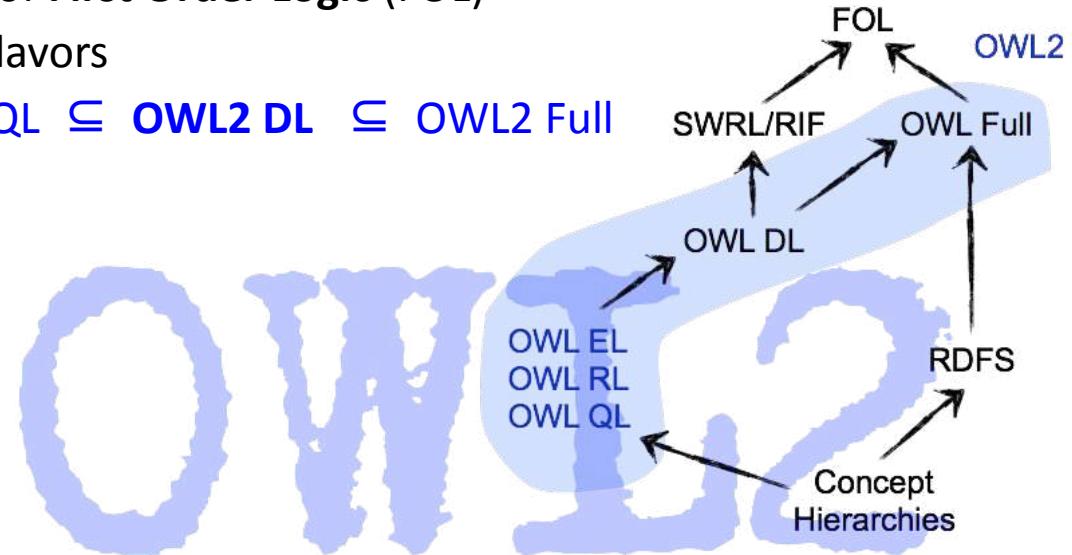
# The Semantic Web Technology Stack (not a piece of cake...)



# Web Ontology Language OWL – OWL Flavors

- OWL is a semantic fragment of **First Order Logic (FOL)**
- OWL also exists in different flavors

$\text{OWL EL, OWL RL, OWL QL} \subseteq \text{OWL2 DL} \subseteq \text{OWL2 Full}$



# OWL2 DL is based on 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)

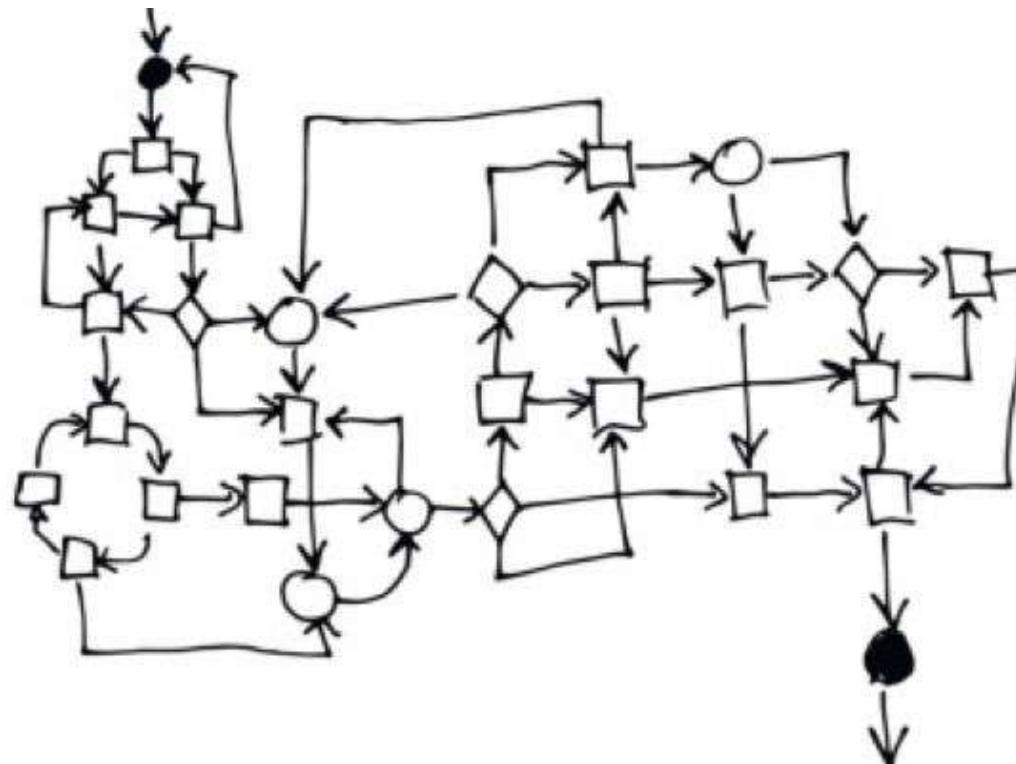
- Class membership  $C(a)$
- Property relation  $R(a,b)$
- Negated property relation  $\neg S(a,b)$
- Equality  $a=b$
- Inequality  $a \neq b$



# How to design your own Ontology

# The Ontology Development Process

SOMETHING



The Software Development Process, Geek & Poke,  
<http://geekandpoke.typepad.com/geekandpoke/2012/01/simply-explained-dp.html>

Great Ontology

# The Ontology Development Process

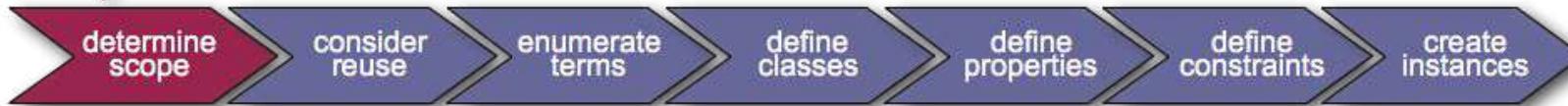
(Noy, McGuinness, 2000)



- In practice, the ontology development process is an **iterative Process that repeats continuously** and improves the ontology
- There are always **different approaches** for modelling an ontology
- The designated application decides about the modelling approach

“There is no one correct way to model a domain.  
There are always viable alternatives.”

# Determine Domain and Focus



- Which **Domain** should be covered by the ontology?
- **What** should the ontology be used for?
- **Who** will use and maintain the ontology?
- What types of **Questions** should be answered by the knowledge represented in the ontology?
- Formulation of **Competency Questions!**

# Determine Domain and Focus

determine scope

consider reuse

enumerate terms

define classes

define properties

define constraints

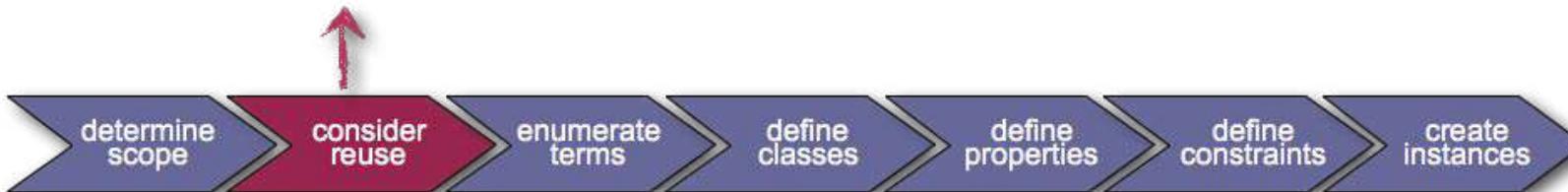
create instances

## Competency Questions (Example: Wine Ontology)

- Which properties of the wine should be considered for modelling?
- Is Bordeaux a white wine or a red wine?
- Does a Sauvignon Blanc match with fish?
- Which wine matches best with grilled vegetables?
- Which properties of a wine do influence whether it matches with a specific dish?
- Does the bouquet of a wine change over time?
- Does the taste of a wine change over time?
- ...

These Questions might change  
within the ontology life cycle.

# Consider Reuse

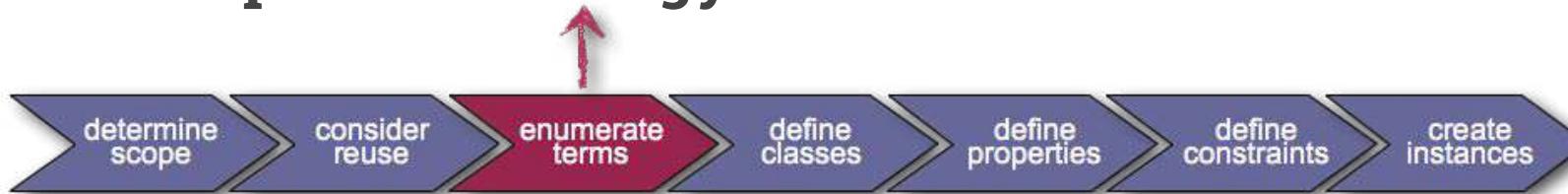


## Why should we consider reuse?

- In order to save **cost**
- In order to apply **tools** that are applied to other existing ontologies also to our own ontology
- In order to reuse ontologies that have been validated by their application

If you don't find a suitable ontology or if the adaption is too complex then create a new ontology!

# Develop a Terminology



- Which **concepts** do we need to represent?
- Which **properties** do these concepts have?
- **What** do we want **to say** about these concepts?

## Example: Wine Ontology

- *wine, grape, winery, location,...*
- *a wine's colour, body, flavor, sugar content,...*
- *subtypes of wine: white wine, red wine, Bordeaux wine,...*
- *types of food: seafood, meat, vegetables, cheese, dessert,...*
- ...

# Define Classes and Class Hierarchies



- **Classes** are concepts in the designated domain
  - *class of wines*
  - *class of wineries*
  - *class of red wines*
  - ...
- Classes are collections of objects with **similar properties**.
- Choose a **top-down / bottom-up / middle-out** approach to model class hierarchies.

# Define Properties



**Properties** in a class definition describe attributes of instances  
*every wine has a colour, residual sugar, producer, etc...*

The screenshot shows the Protégé interface with the following details:

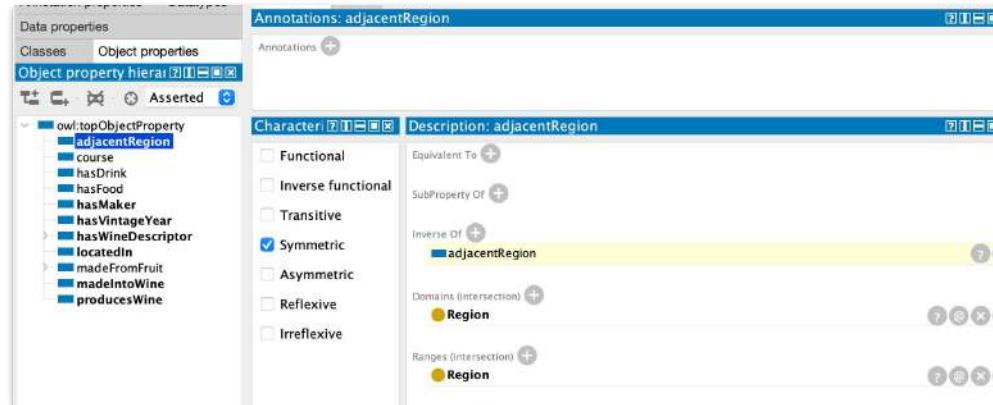
- Left Panel (Object property hierarchy):** Shows a tree of object properties, with 'hasFood' selected.
- Annotations Tab:** Displays annotations for the 'hasFood' property, including:
  - Annotations:** hasFood
  - Character:** (checkboxes for Functional, Inverse functional, Transitive, Symmetric)
  - Description:** hasFood
  - Relationships:** (buttons for Equivalent To, SubProperty Of, Inverse Of)

# Define Property Constraints

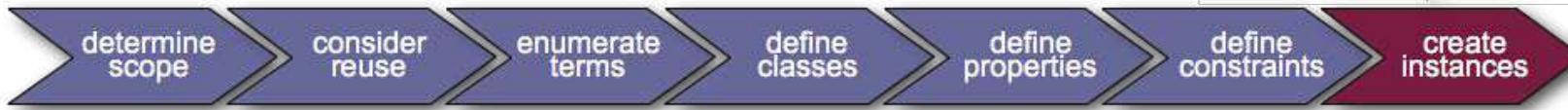


**Property constraints** (restrictions) describe or restrict the set of possible property values

- *The name of a wine is a String*
- *The producer is an instance of Winemaker...*



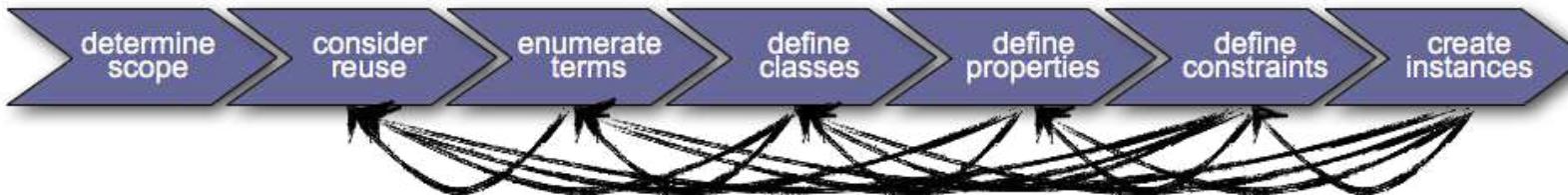
# Define Class & Property Instances



- Create **instances for the classes**:
  - Every **class** directly becomes the **type** of its instances.
  - Every **superclass** of a direct type is also type of its instances.
- Create **instances for properties**, i.e. the assignment of property values for the instances according to the given constraints.
- *“The glass of red wine that I drank last supper...”*

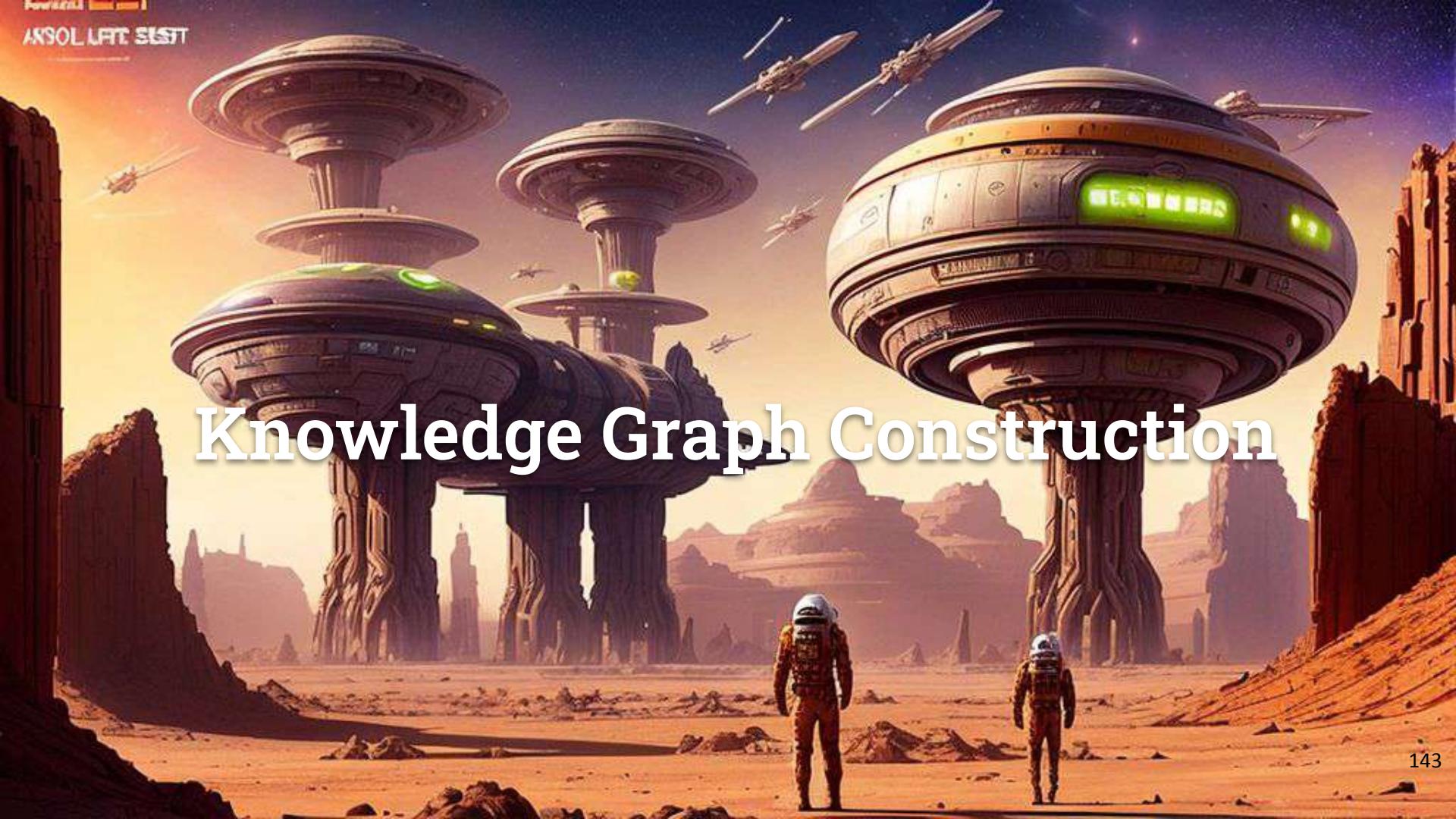
# The Ontology Development Process

(Noy, McGuinness, 2000)



Ontology development in practice is an **iterative process** that **repeats continuously** and improves the ontology.

“There is no one correct way to model a domain.  
There are always viable alternatives.”



# Knowledge Graph Construction

# Knowledge Graph Construction

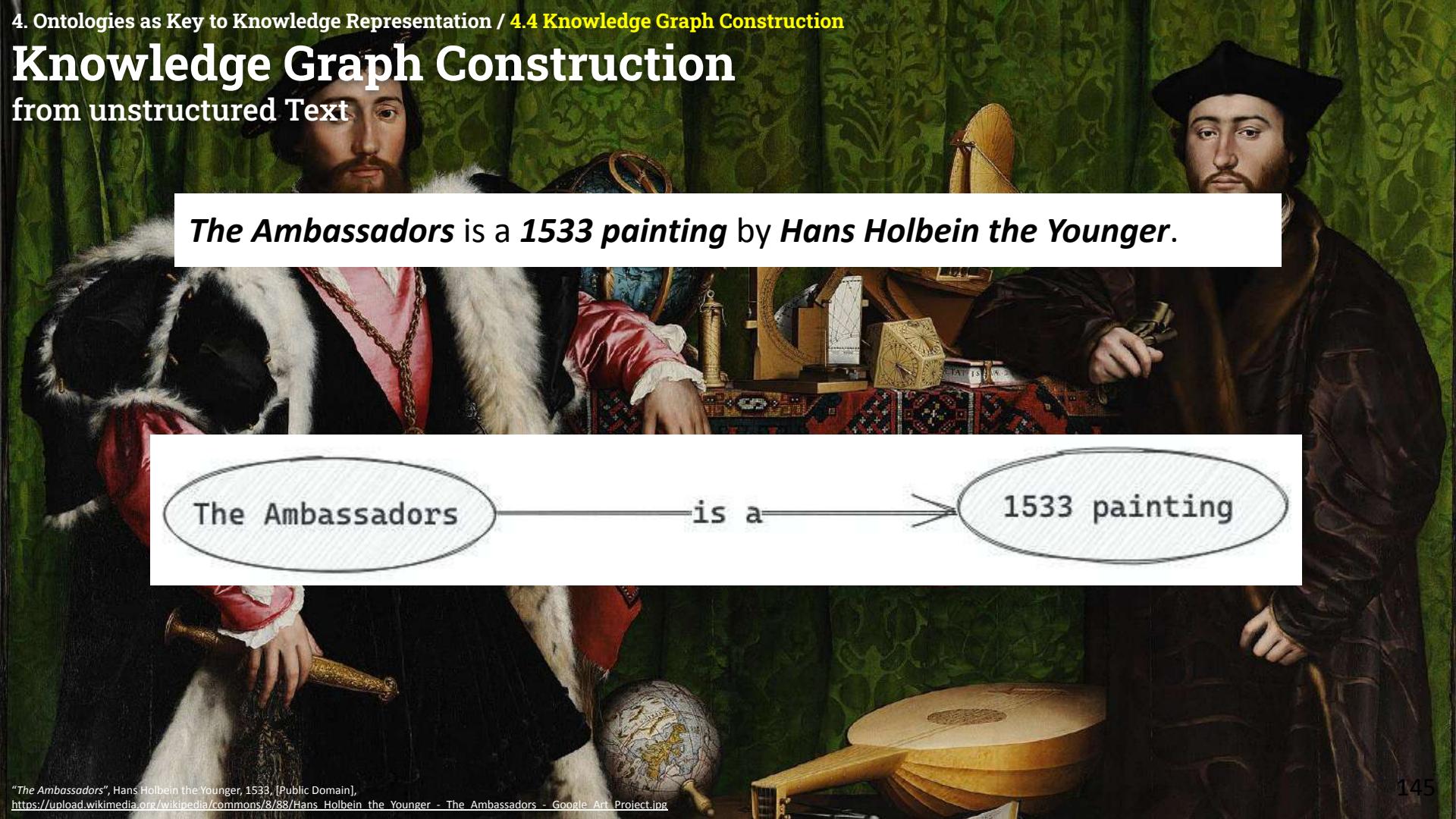
## Knowledge Graph Construction and Enrichment

- Human Collaboration (manually)
- Automated Methods
  - from unstructured text sources
  - from structured data sources

# Knowledge Graph Construction

from unstructured Text

*The Ambassadors* is a **1533 painting** by **Hans Holbein the Younger**.



The Ambassadors

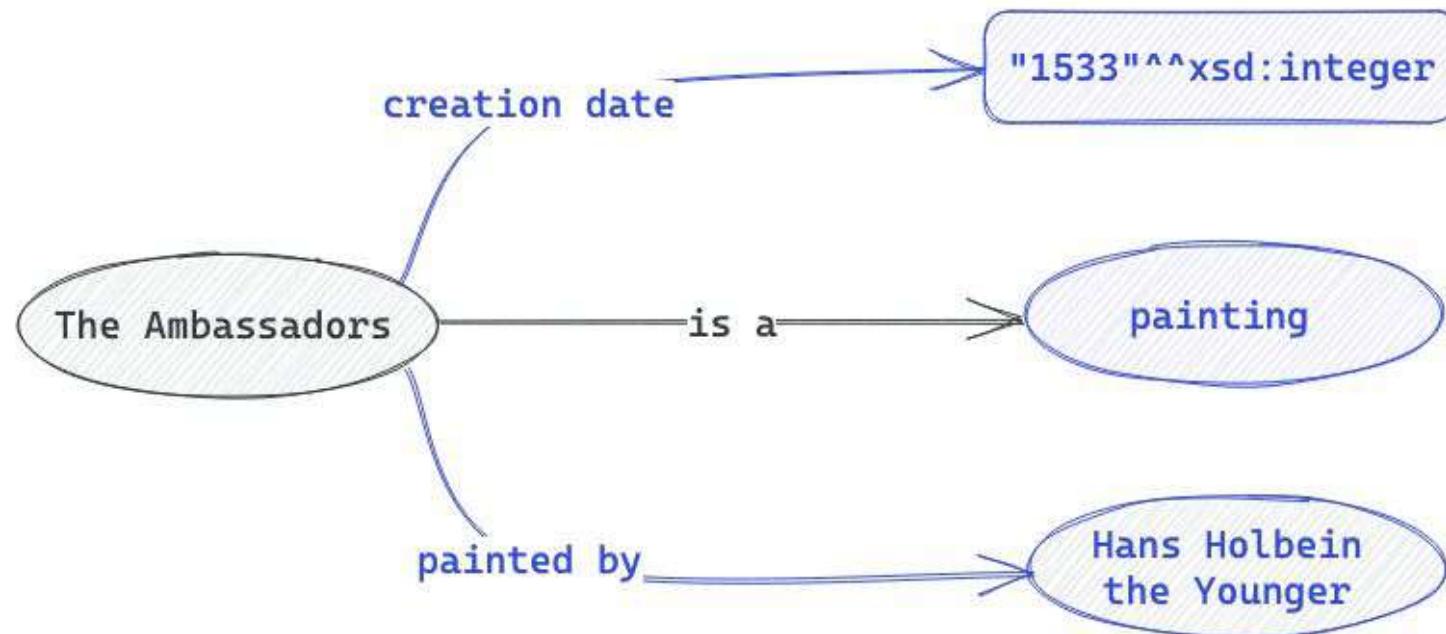
is a

1533 painting

# Knowledge Graph Construction

from unstructured Text

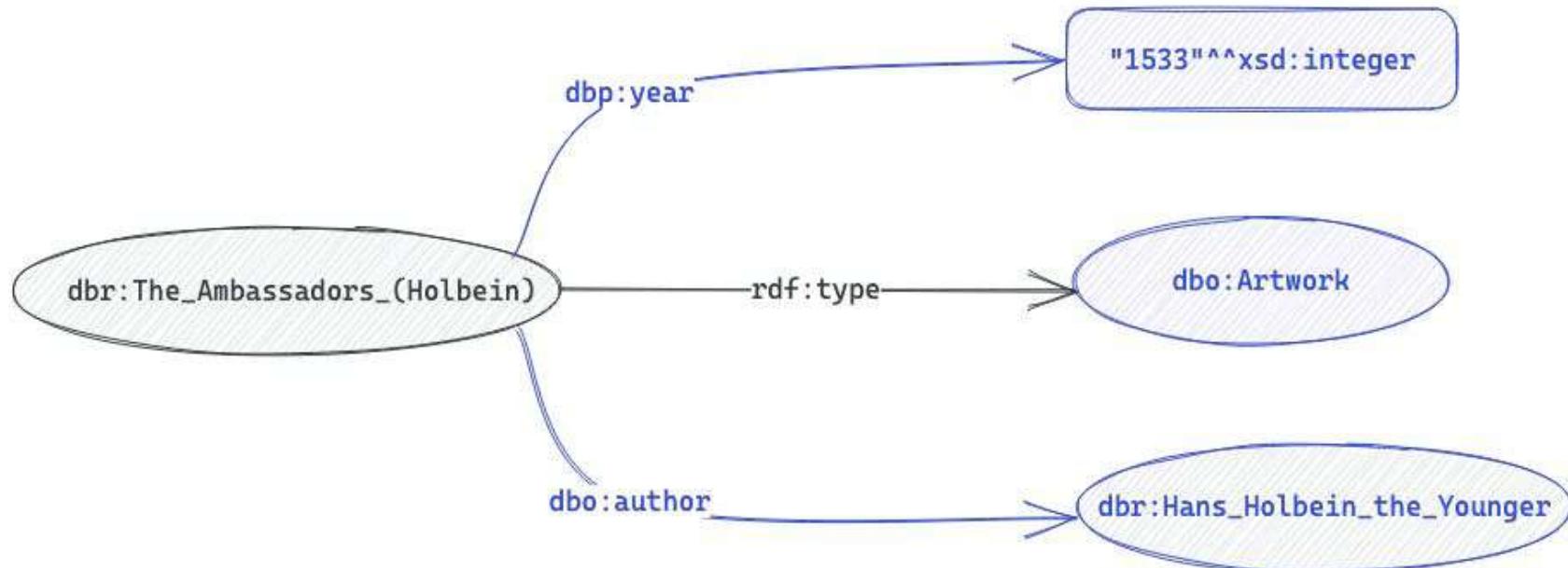
*The Ambassadors* is a **1533 painting** by **Hans Holbein the Younger**.



# Knowledge Graph Construction

from unstructured Text

*The Ambassadors* is a **1533 painting** by **Hans Holbein the Younger**.



# Knowledge Graph Construction

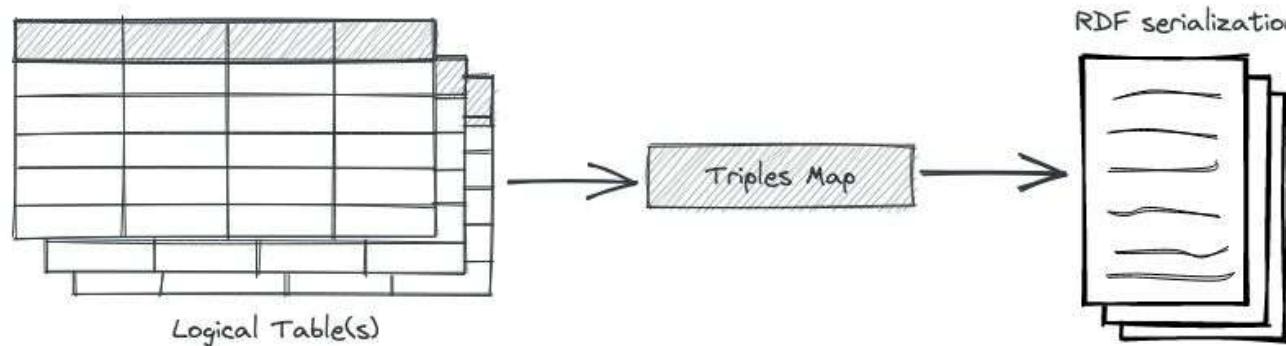
## from structured Data Sources

- Structured data sources, e.g.,
  - Relational Databases, CSV files, etc.
  - Tree-based structured sources like XML, JSON, etc.
- Unlike **unstructured text**, where knowledge has to be **(imprecisely) extracted**, **structured data sources** often can be **(precisely) mapped** to a knowledge graph.
- 2-Step General Mapping:
  - (1) **Create a mapping** from the **source to a graph**.
  - (2) **Use the mapping** in order to **materialize the source data as a graph** or to virtualize the source (creating a graph view over the legacy data).

# KG Construction from structured Data Sources

## R2RML

- **R2RML** is a language for specifying mappings from relational data to RDF
- A **mapping** takes as input a **logical table**, i.e.,
  - a database table
  - a database view, or
  - an SQL query
- A logical table is mapped to a set of triples by a rule called **triples map**.





# Ontologies & Knowledge Graphs – Best Practices

# *Ontologies & Reality*

# Ontology Types and Categories

according to their level of Generality

## Domain Ontology

fundamental concepts according to a generic domain; specializes terms introduced in top-level ontology

## Top-Level Ontology (Upper Ontology, Foundation Ontology)

general, cross domain ontologies;

represent very general concepts as: Time, Space, Event;  
independent of a specific domain or problem

## Task Ontology

fundamental concepts according to a general activity or task; specializes terms introduced in top-level ontology

## Application Ontology

specialized ontology focussed on a specific task and domain; often a specialization of both task and domain ontology; often specify roles played by domain entities for specific activity

# Necessities and Needs

*Do you really need an Ontology?*

# Take the Pragmatic Approach

*What is the Purpose  
of your Ontology?*

# The Intuition behind Names

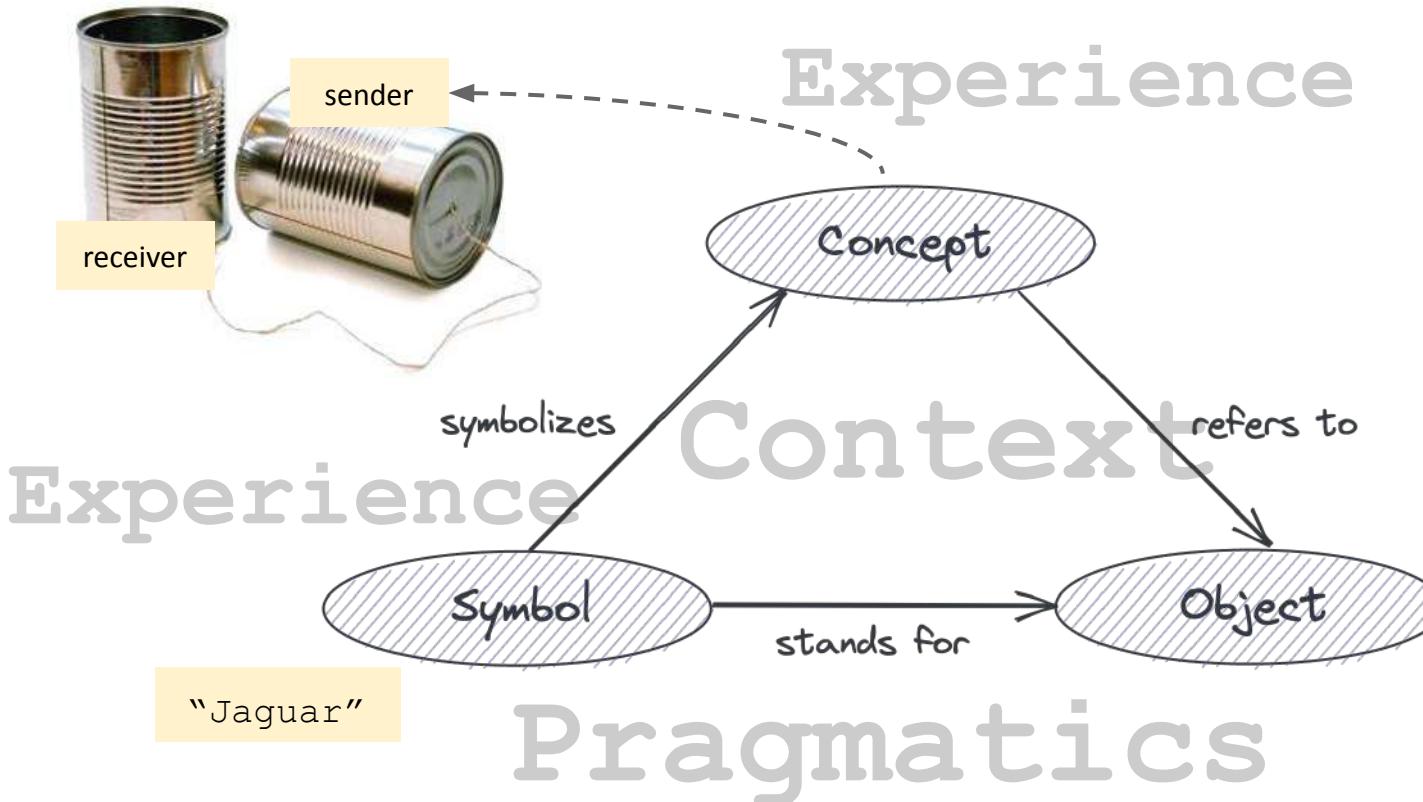
*Say what you mean,  
Mean what you say.*

[4]

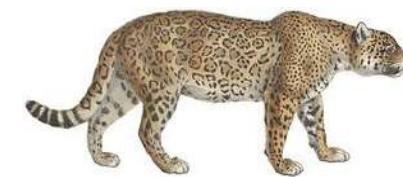


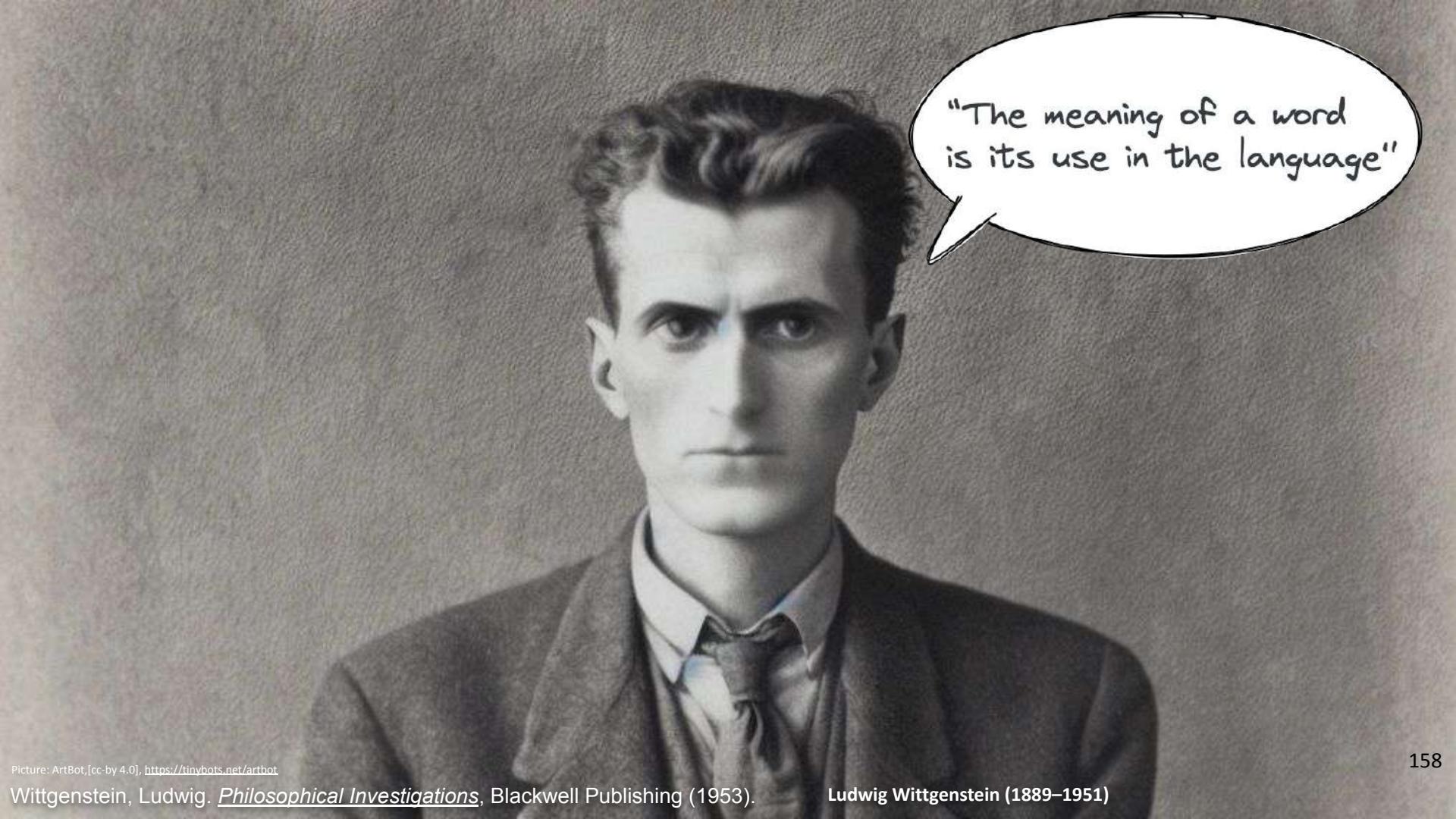
## *5. Intelligent Applications with Knowledge Graphs and Deep Learning*

# What is the Meaning of a Word?



[2,3,4,5]



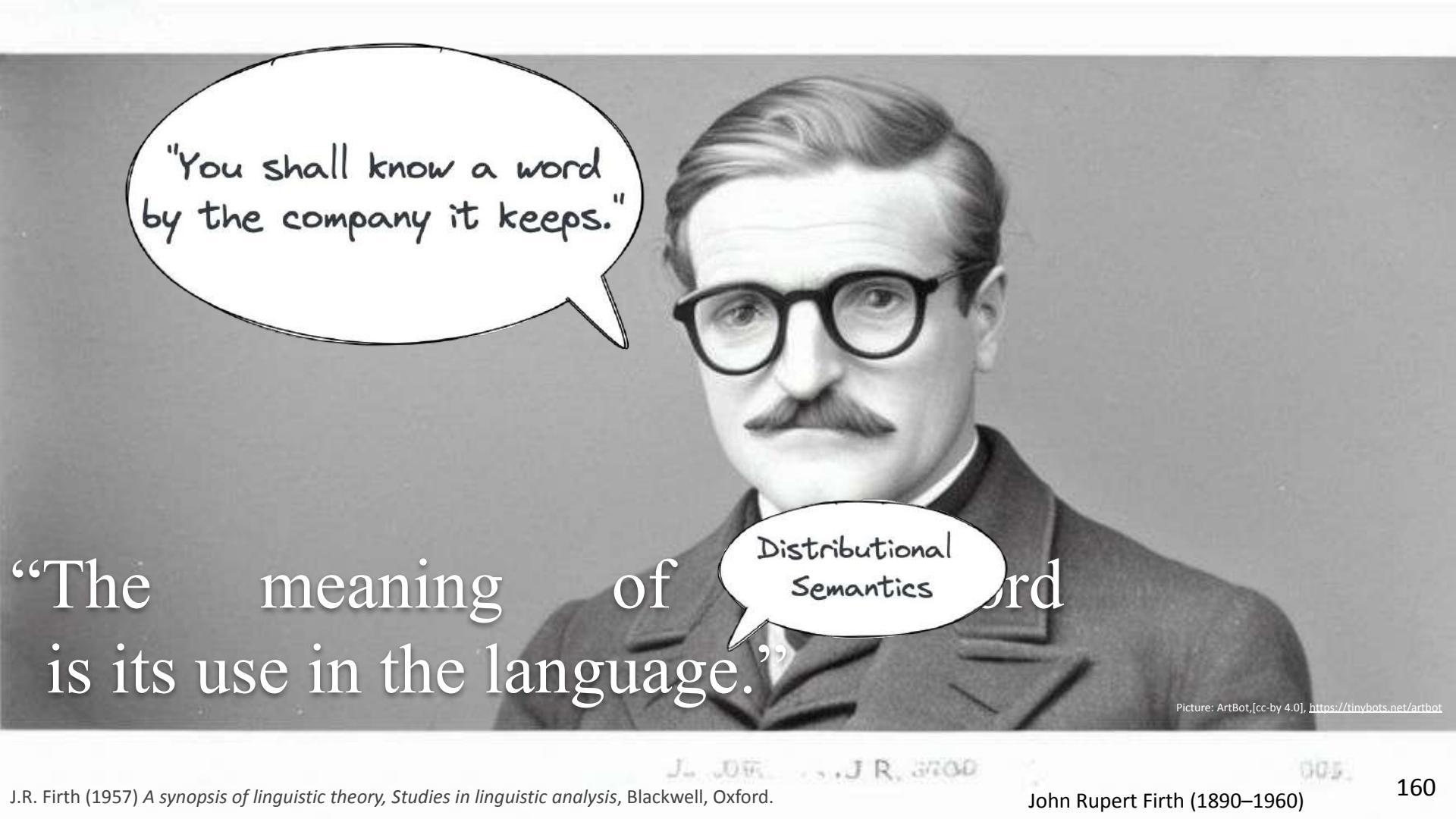
A black and white portrait of Ludwig Wittgenstein, a young man with dark hair and a serious expression, wearing a suit and tie. A speech bubble originates from his mouth.

"The meaning of a word  
is its use in the language"

# Let's Define Words by their Usage

- In particular, words are defined by their environments (i.e. the words around them).
- “*If [words] A and B have almost identical environments [...] we say that they are synonyms.*”  
Zellig S. Harris (1954)
- Thereby: semantic representations for words can be derived through analysis of patterns of lexical co-occurrence in large language corpora.

Zellig S. Harris (1954) *Distributional Structure*, WORD, 10:2-3, 146-162, DOI: [10.1080/00437956.1954.11659520](https://doi.org/10.1080/00437956.1954.11659520)



"You shall know a word  
by the company it keeps."

“The meaning of word  
is its use in the language.”

Distributional  
Semantics

Picture: ArtBot [cc-by 4.0], <https://tinybots.net/artbot>

1-gram

# N-Gram Shakespeare Generator

3-gram

To him swallowed confess hear  
both. Which. Of save on trail for  
are ay device and rote life have  
Hill he late speaks; or! a more to  
leg less first you enter.

2-gram

Why dost stand forth thy  
canopy, forsooth; he is this  
palpable hit the King Henry.  
Live king. Follow. What means, sir.  
I confess she?  
then all sorts, he is trim, captain

Fly, and will rid me these  
news of price. Therefore the sadness  
of parting, as they say,  
'tis done. This shall forbid it should  
be branded, if renown made it empty.

4-gram

I will go seek the traitor Gloucester.  
Exeunt some of the watch.  
A great banquet serv'd in.  
It cannot be but so.



Write a dialogue between Caliban and Puck from an unknown Shakespeare play.



Caliban: Puck, wherefore art thou here on this island?

Puck: I am but a messenger, Caliban, sent by the fairy queen to bring magic and mischief to this place.

Caliban: And what manner of magic do you bring?

Puck: Oh, all sorts! I can change shapes, cast spells, and make illusions. And I have a particular talent for causing trouble for those who cross me.

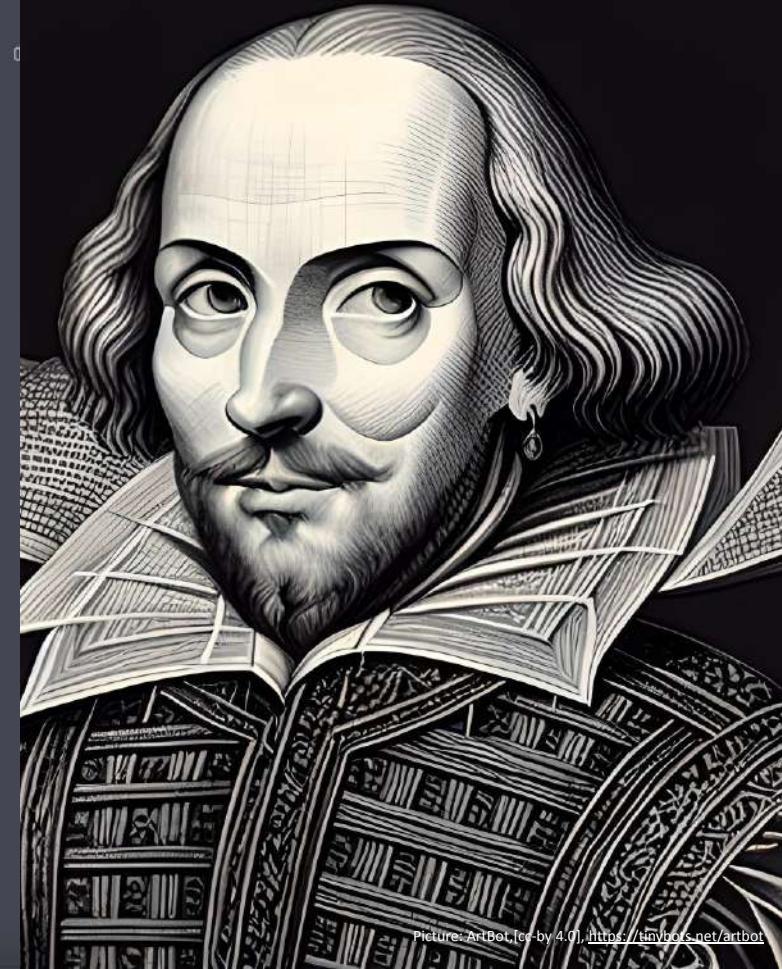
Caliban: I see. And what do you know of me?

Puck: I know that thou art Prospero's servant, Caliban, and that thou art unhappy with thy lot in life.

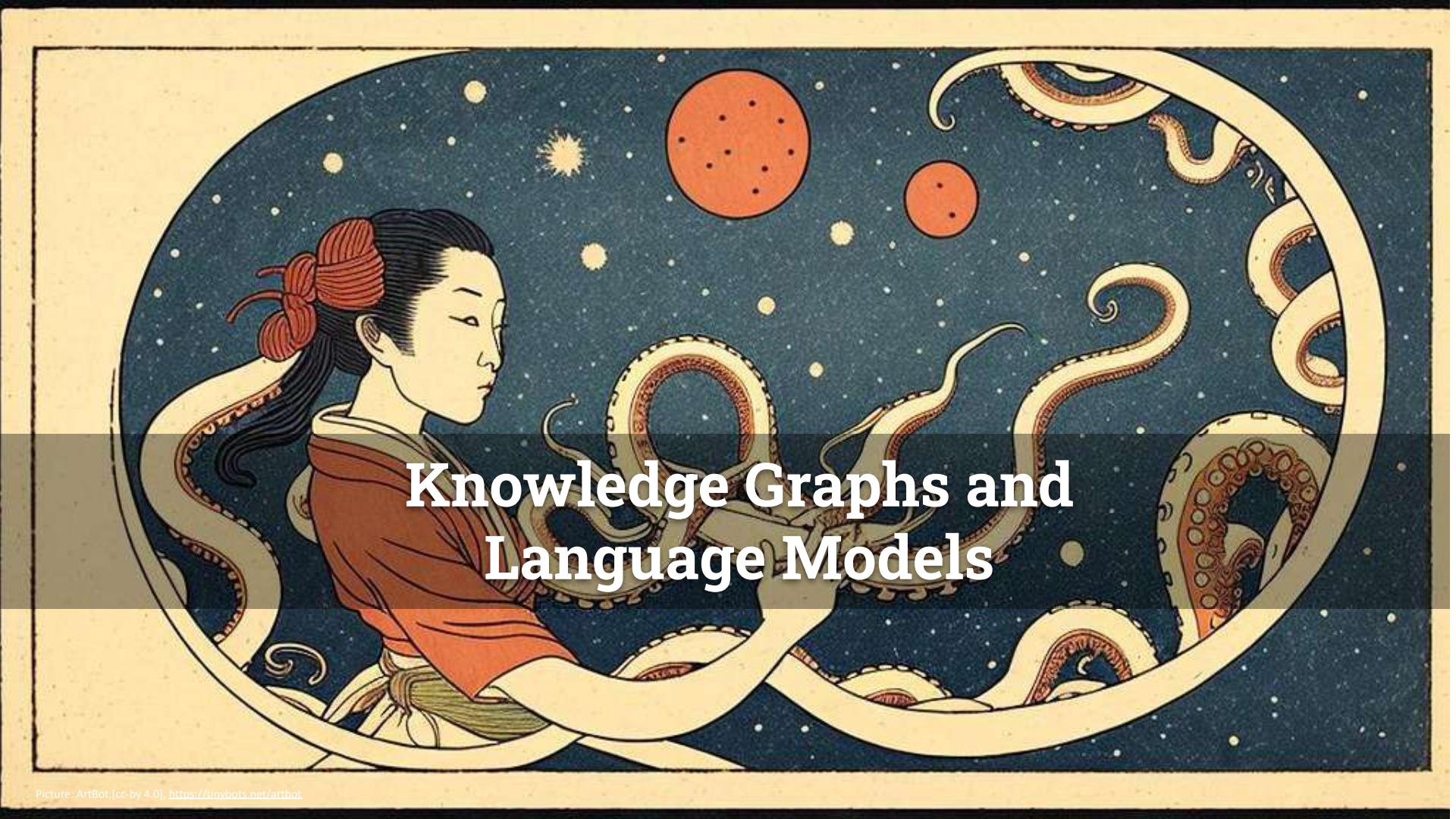
Caliban: Aye, that is true. I am a slave to Prospero, forced to do his bidding and denied my freedom.

Puck: Well, perhaps I can help thee with that. I can cause mischief for Prospero and make it easier for you to escape his grasp.

Caliban: Really? You would do that for me?



Picture: ArtBot [cc-by 4.0], <https://tinybots.net/artbot>



# Knowledge Graphs and Language Models

# Deficiencies of Large Language Models

- **Language Models may hallucinate**  
There is no guarantee for truthful or reliable output.
- **Language Models are frequency-biased**  
High resource domains might be better, i.e. more reliably represented, than long tail low resource domains.
- **Language Models often seem to be convincing, but may be wrong**  
Answers might appear rather convincing or authentic, but might be wrong in subtle ways.

**Never follow advice from a language model without verification!**

# Semantics from Stochastics?

- **Can we really derive meaning from probability distributions and statistics?**
- **Let's look at the Language Domain:**
  - Based on probability and statistics it is possible to **create syntactically and semantically correct texts**.
  - With larger training data and larger models also **contextually and pragmatically well fitting texts** can be created.
  - **Factual questions** can (most times) be correctly answered.
  - What about **Interpretative questions** and **Evaluative questions**?



# Playground

[Completion](#)[Q & A](#)[Summarization](#)

## Advanced Settings

(or use our [jumpstart](#))

**Model** ⓘ

Multilingual model trained on English, German, French, Spanish and Italian

luminous-extended

**Maximum Tokens** ⓘ

64

**Stop Sequences** ⓘ

###\n\n

**Best of** ⓘ

1 10

Show Probabilities ⓘ

**Temperature** ⓘ

0 1

**Top K** ⓘ

5

[Prompt](#)[Text](#)[Multimodal](#)[Model Q&A](#)

Q: Why is there war in Ukraine?

A:

Interpretative questions or Evaluative questions might be subject of inherent bias (of the training data).

### Completion

The war in Ukraine is a result of the US-NATO-EU-backed coup in Ukraine in 2014. The coup was carried out by a coalition of fascist, neo-Nazi, and ultra-nationalist groups that have been operating in Ukraine for years. They are known as the Right Sector. The

[Append to Prompt](#)

[Reset](#)

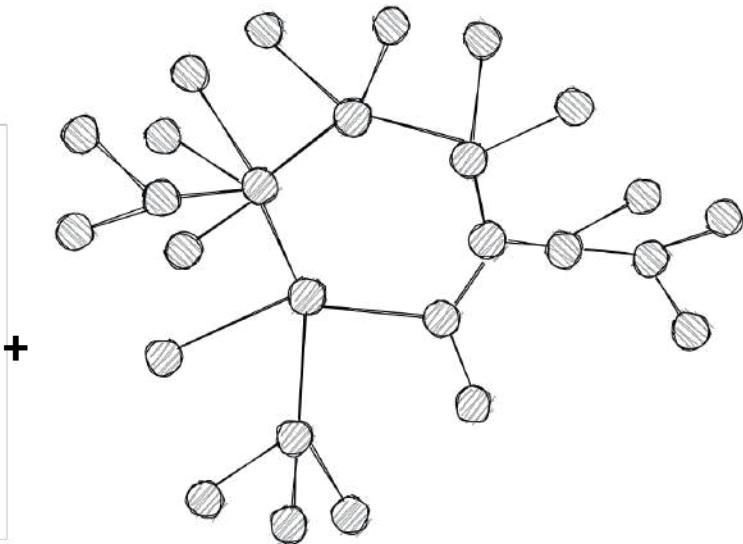
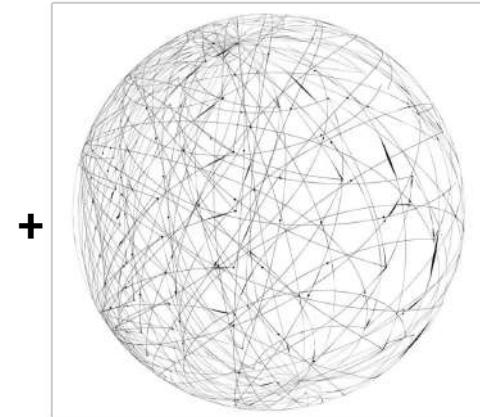
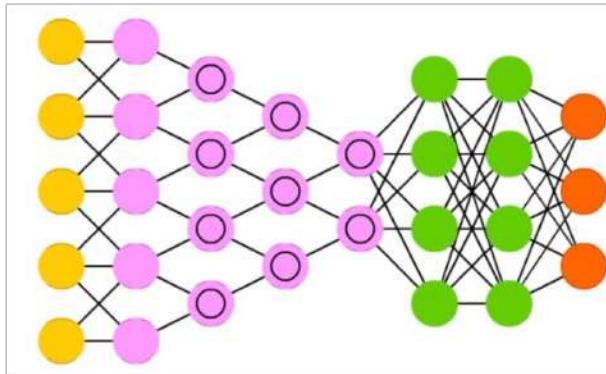
[Submit](#)

# Semantics from Stochastics?

- **Can we really derive meaning from probability distributions and statistics?**
- Evaluation by probing current Large Language Models
  - Language models create **factual and interpretative errors**.
  - There seems to be only a **very limited understanding of common sense**.
  - It remains **unclear** how much meaning can be “learned” via stochastic models.

# Hybrid AI – Using One for the Benefit of the Other

- Knowledge Graph Embeddings
- Knowledge Extraction
- Explainable AI
- Fact Checking

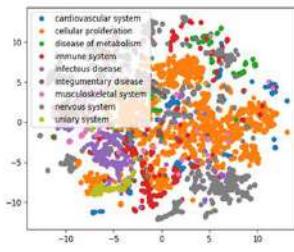


Subsymbolic Knowledge Representation

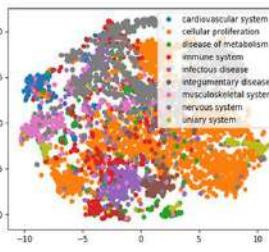
+ Symbolic Knowledge Representation

# Hybrid AI – Knowledge Graph Embeddings

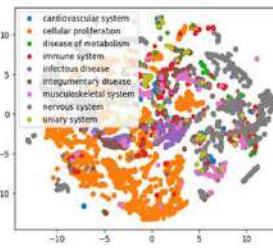
A. Walking RDF/OWL



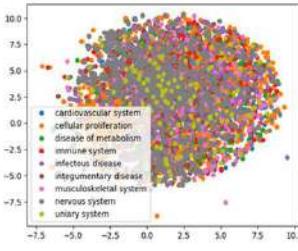
B. TransE embeddings



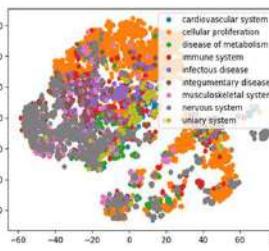
C. Poincare embeddings



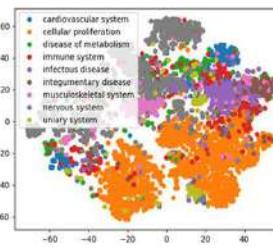
D. Rescal embeddings



E. SimplE embeddings



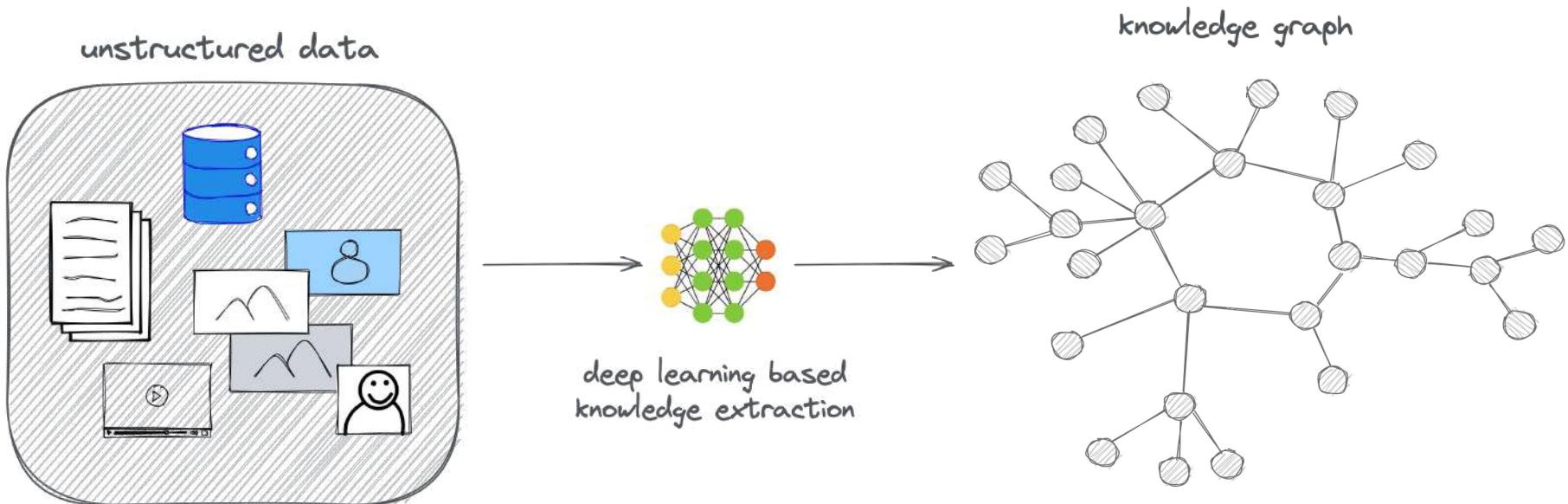
F. R-GCN embeddings



- Knowledge Graph Completion
- KGE for Classification Tasks

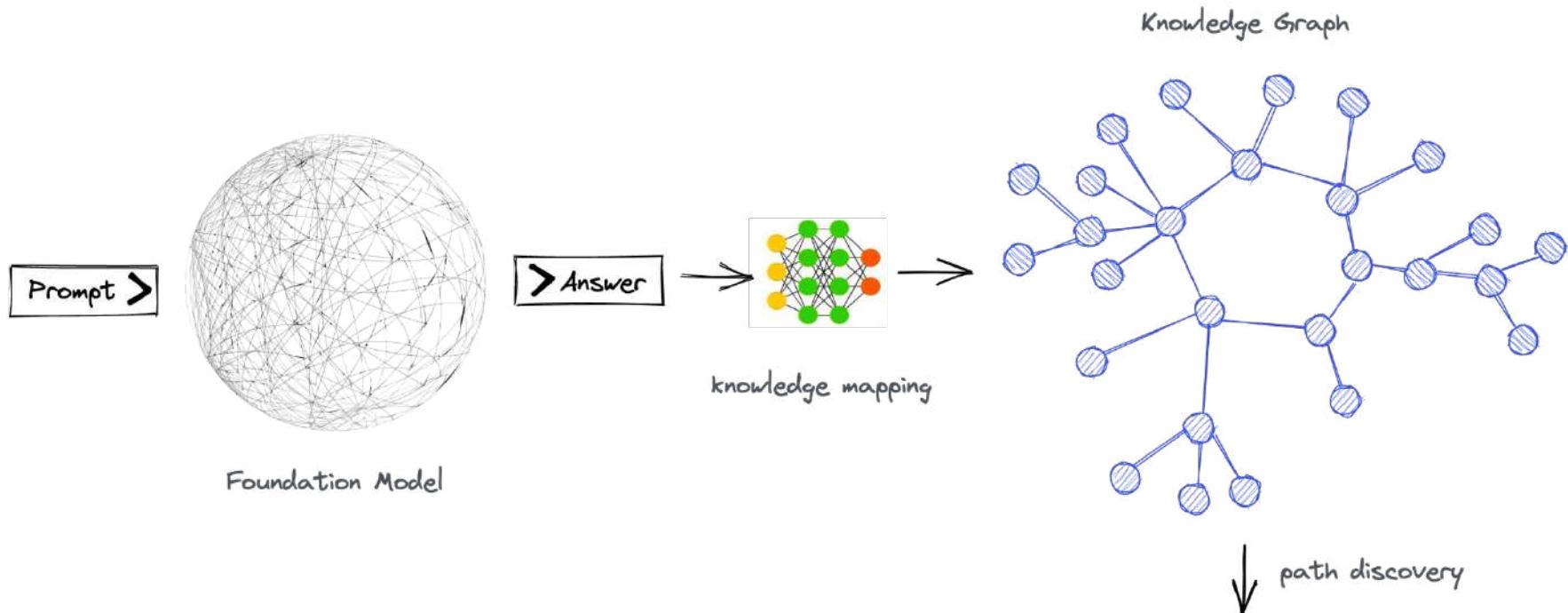
- Ontology Mapping
- Entity/Knowledge Graph Alignment

# Hybrid AI – Knowledge Extraction

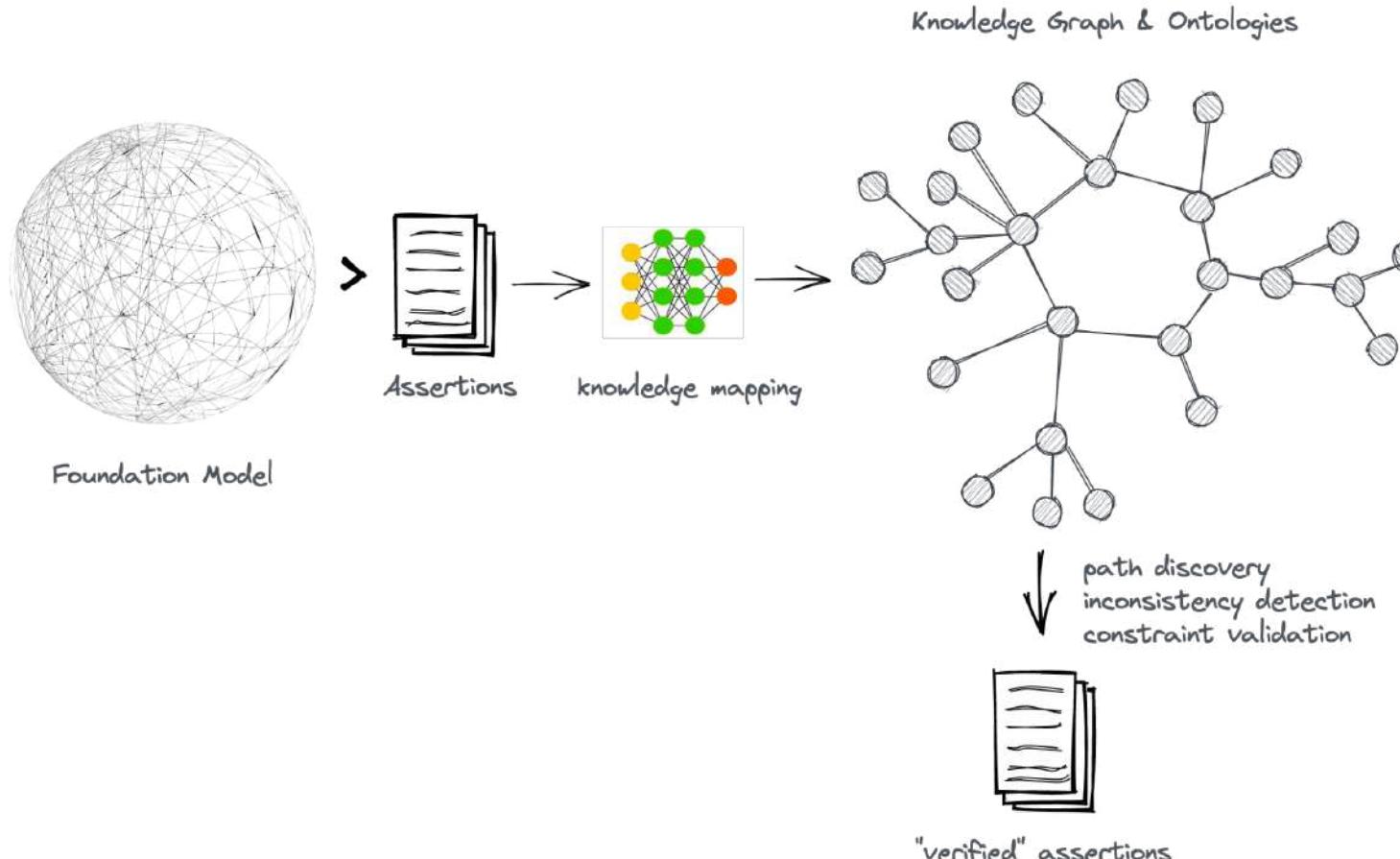


- Knowledge Graph Population
- Ontology Learning
- Entity Recognition & Linking
- Relation Extraction

# Hybrid AI – Explainable AI



# Hybrid AI – Fact Checking





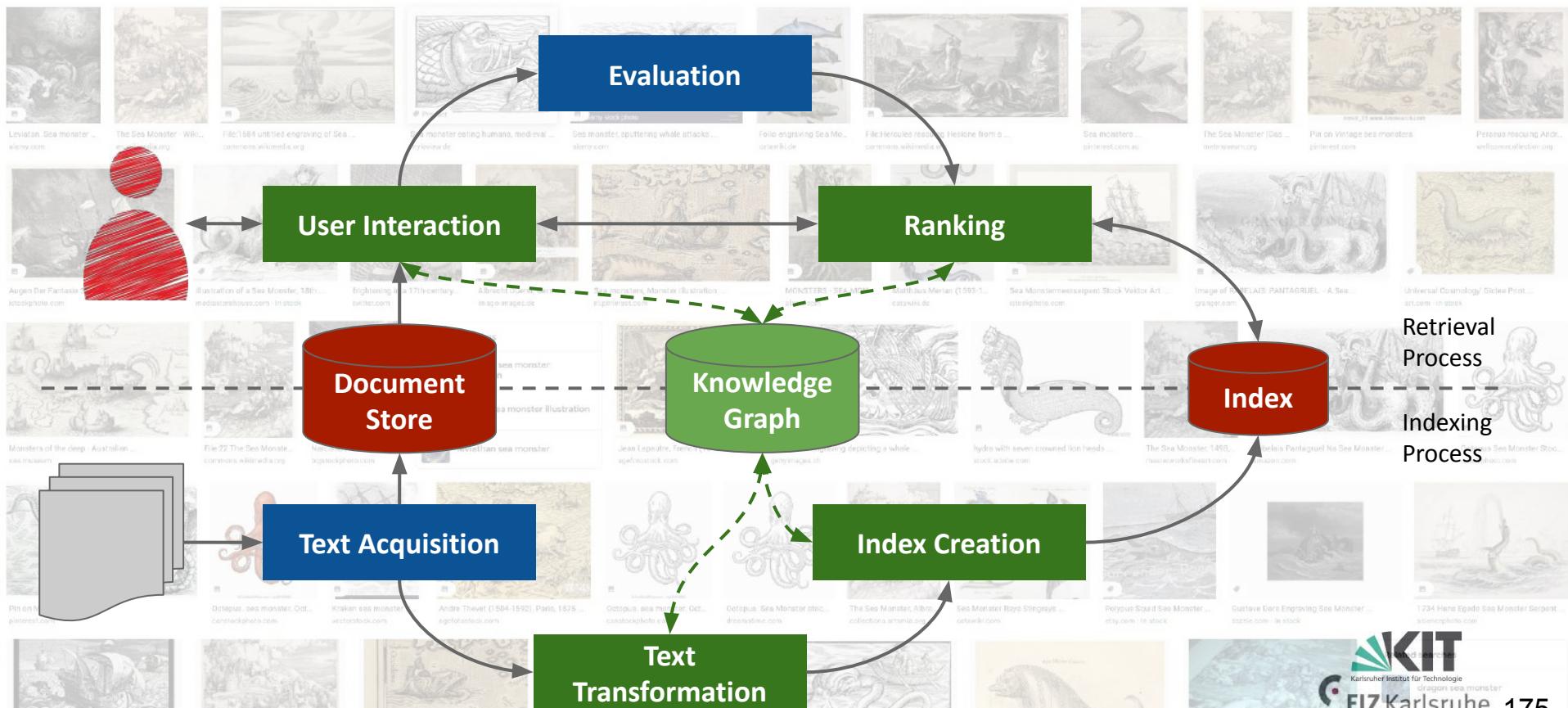
# Semantic Search

# The Information Retrieval Dilemma

- Ambiguity of natural language (polysemy)
- Different words/expressions for the same concept (synonyms, metaphors, paraphrases, ...)
- Implicit (hidden) information
- Context knowledge and experience

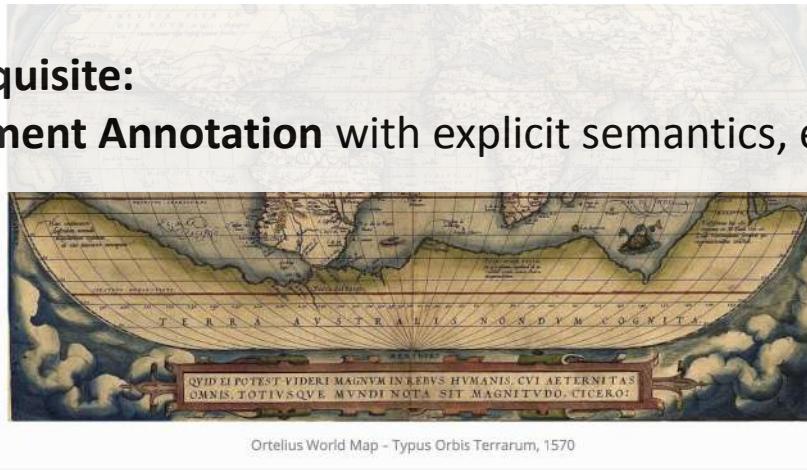
## How can Knowledge Graphs support IR?

# Semantic Technologies Supported IR



# Semantic Technologies Supported Information Retrieval

- Prerequisite:  
**Document Annotation** with explicit semantics, e.g. semantic entities



On May 20, 1570, Belgian cartographer and geographer **Abraham Ortelius** published the first modern atlas, the *Theatrum Orbis Terrarum*, in Antwerp. The map was composed of 53 separate sheets and sustaining text bound to form a book for easier use.

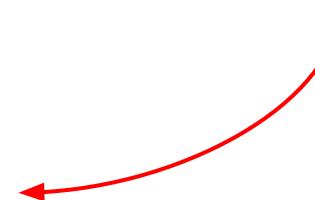
## Abraham Ortelius – Early Years

Abraham Ortelius was born in Antwerp, but grew up in Leuven. He studied law at the University of Leuven and became a lawyer at a young age. In 1575, he was appointed geographer to the Duke of Alva.

<b>Abraham Ortelius</b>	
Abraham Ortelius (also Ortels, Orthellius, Wortels; 14 April 1527 – 28 June 1598) was a Flemish cartographer and geographer,	
<b>birth year</b>	1527
<b>death year</b>	1598
<b>death place</b>	Antwerp

- Enables **entity-based Information Retrieval**  
Language independent

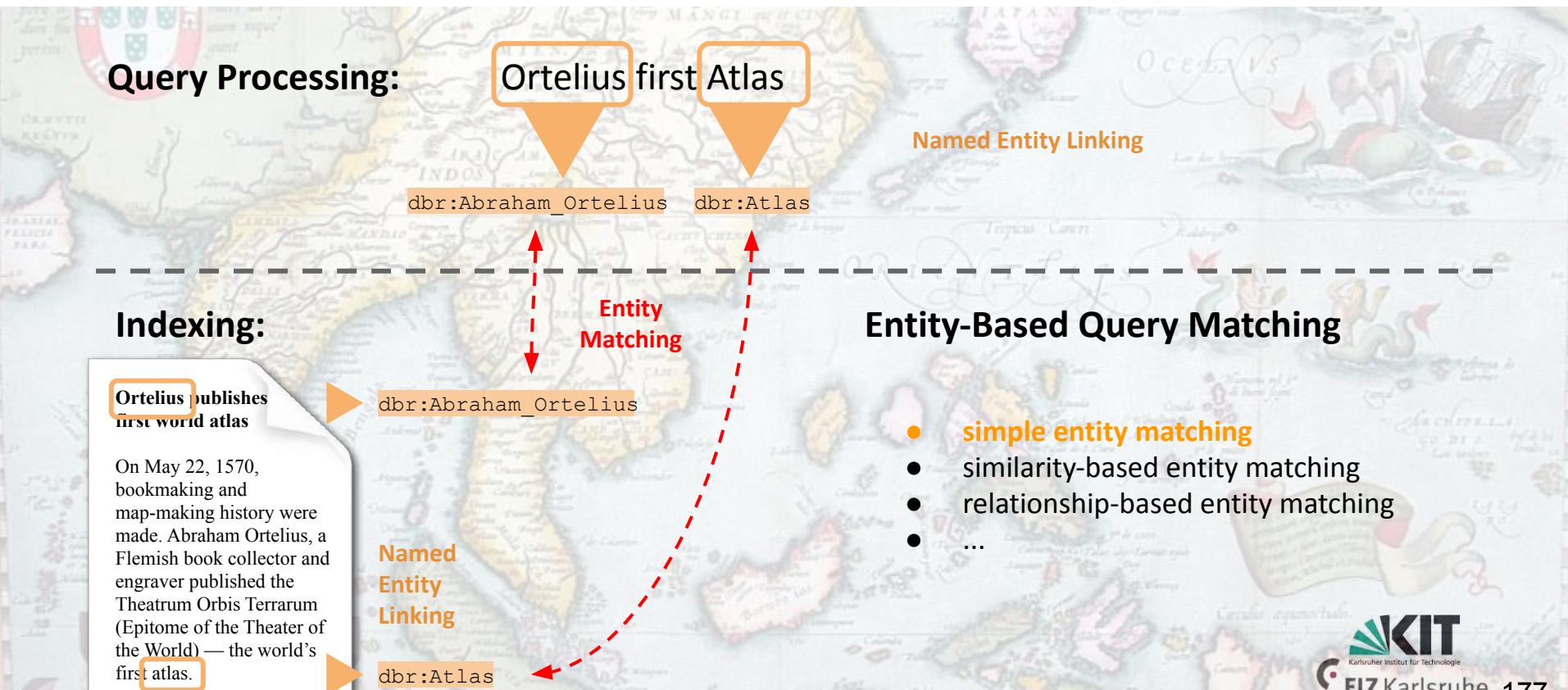
Example for  
Semantic Annotation



<http://scihi.org/abraham-ortelius-theatrum-orbis-terrarum/>

# Entity-based Search

## Simple Entity Matching



# Entity-based Search

## Similarity-based Entity Matching

### Query Processing:

Ortelius first Atlas

dbr:Abraham\_Ortelius dbr:Atlas

### Named Entity Linking

### Indexing:

Mercator's atlas of Europe

Gerardus Mercator was perhaps the best-known mapmaker of all time. Rather than undertaking expeditions and mapping on the ground himself, however, he was an armchair cartographer.

dbr:Atlas  
dbr:Gerardus\_Mercator

### Entity Matching

dbr:Abraham\_Ortelius

### Entity-Based Query Matching

- simple entity matching
- **similarity-based entity matching**
- relationship-based entity matching
- ...

Two entities are considered **semantically similar**,

- if they share property-value pairs
- if they share properties with similar values.

### Named Entity Linking

# Entity-based Search

## Relationship-based Entity Matching

### Query Processing:

Ortelius first Atlas

dbr:Abraham\_Ortelius dbr:Atlas

### Named Entity Linking

### Indexing:

Mercator's atlas of Europe

Gerardus Mercator was perhaps the best-known mapmaker of all time. Rather than undertaking expeditions and mapping on the ground himself, however, he was an armchair cartographer.

### Named Entity Linking

### Entity Matching

### Entity-Based Query Matching

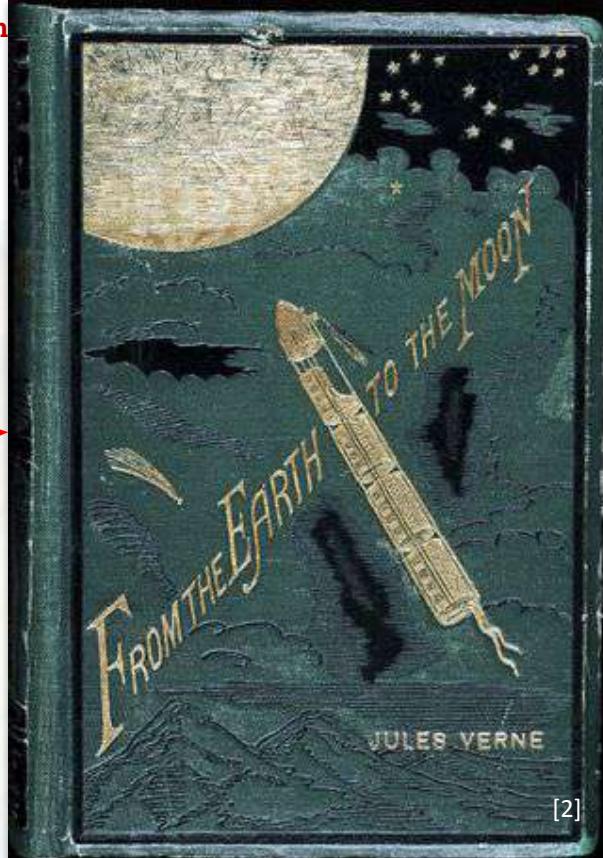
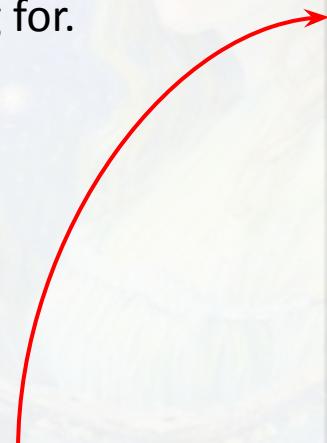
- simple entity matching
- **similarity-based entity matching**
- **relationship-based entity matching**
- ...



# Exploratory Search and Recommendation

# The Retrieval Problem

- **Retrieval Problem:**  
You are looking for **something specific**,  
i.e. you know what you are looking for.
- How to **specify your search request?**  
E.g. for a (specific) book:  
*author name, title, etc.*
- Often you are using
  - (unique) identifier
  - descriptive metadata

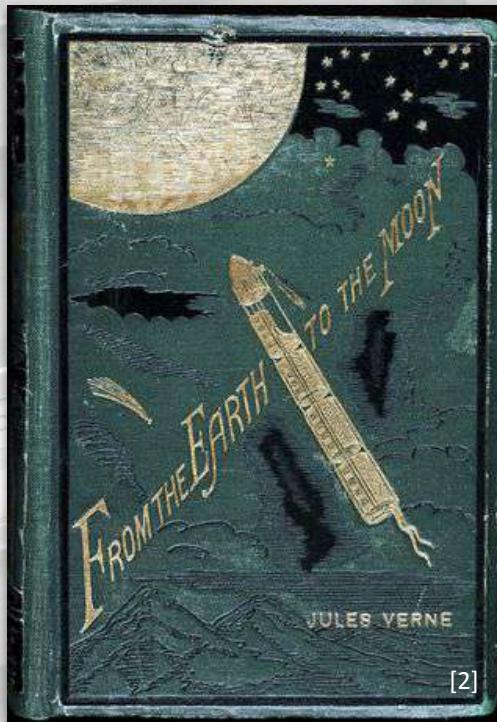


Author: Jules verne

Title: From the Earth to the Moon

# The Retrieval Problem

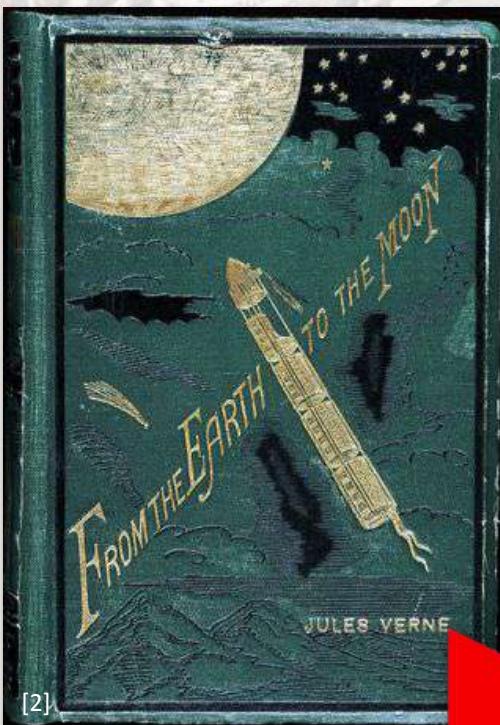
## Index Lookup



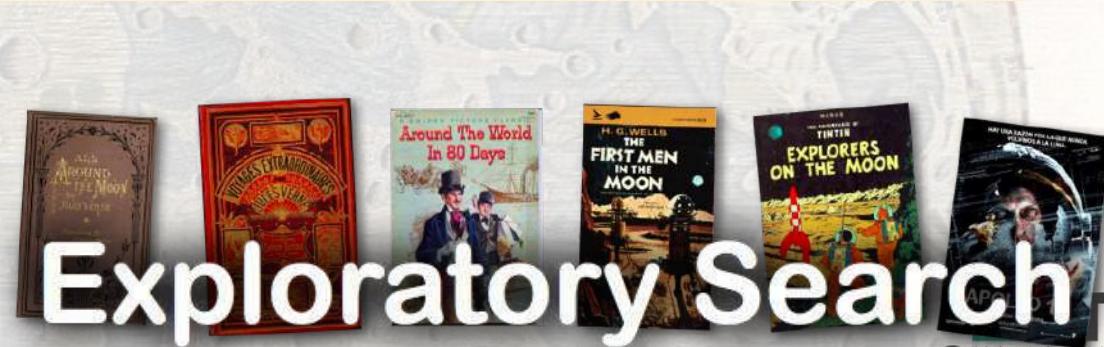
V E R N E, Jules.  
From the Earth to the Moon, Direct in  
97 Hours 20 Minutes and a Trip Round It,  
Sampson Low, Marston&Company,  
London (1873),  
viii, 323 p. plates.

GRC C.194.a.659, 12516.g.20

# Retrieval vs Exploration



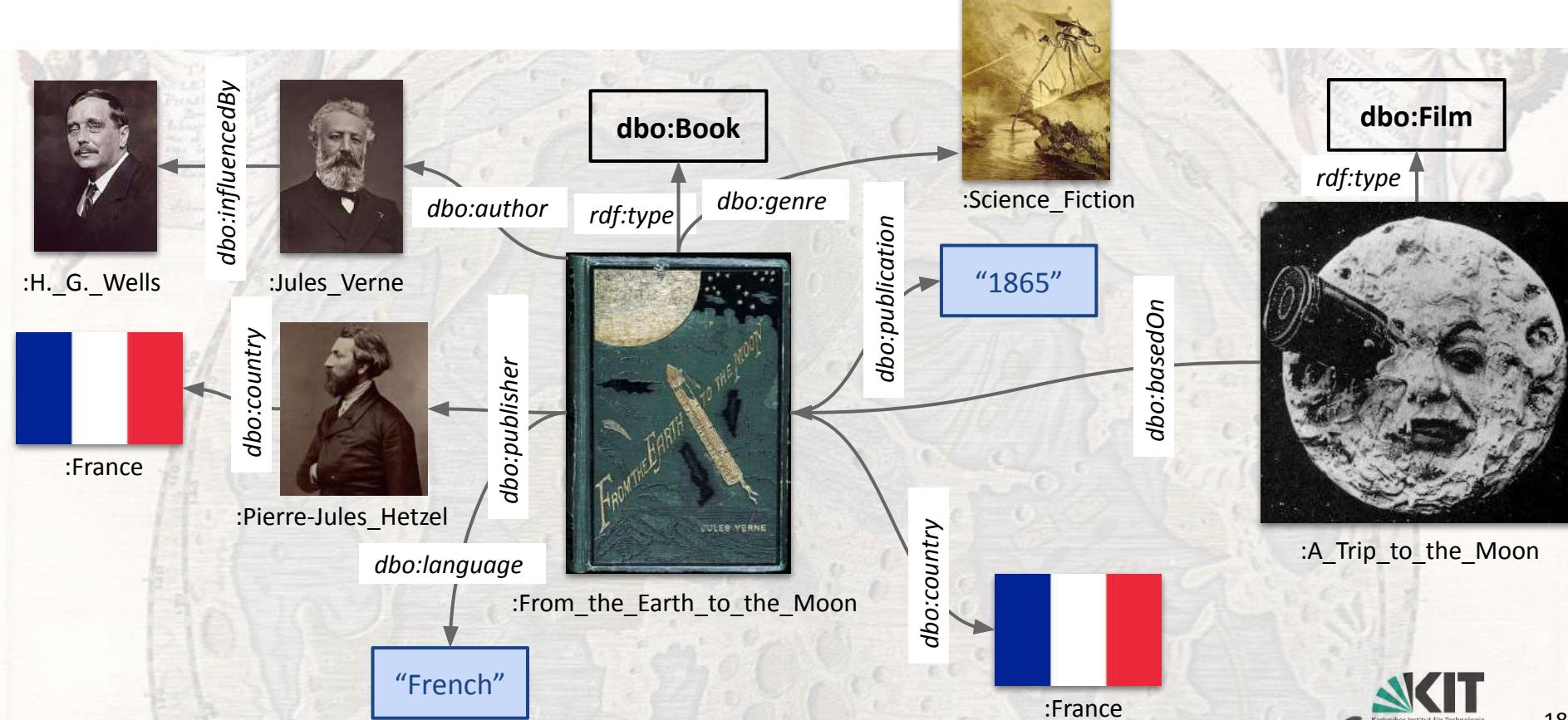
- *Find another (“comparable”) book, (that will be of interest to me...).*
- *Find books of the same or of related topics.*
- *How did the author / the topic develop over time?*
- *What else would I like to read?*
- ...



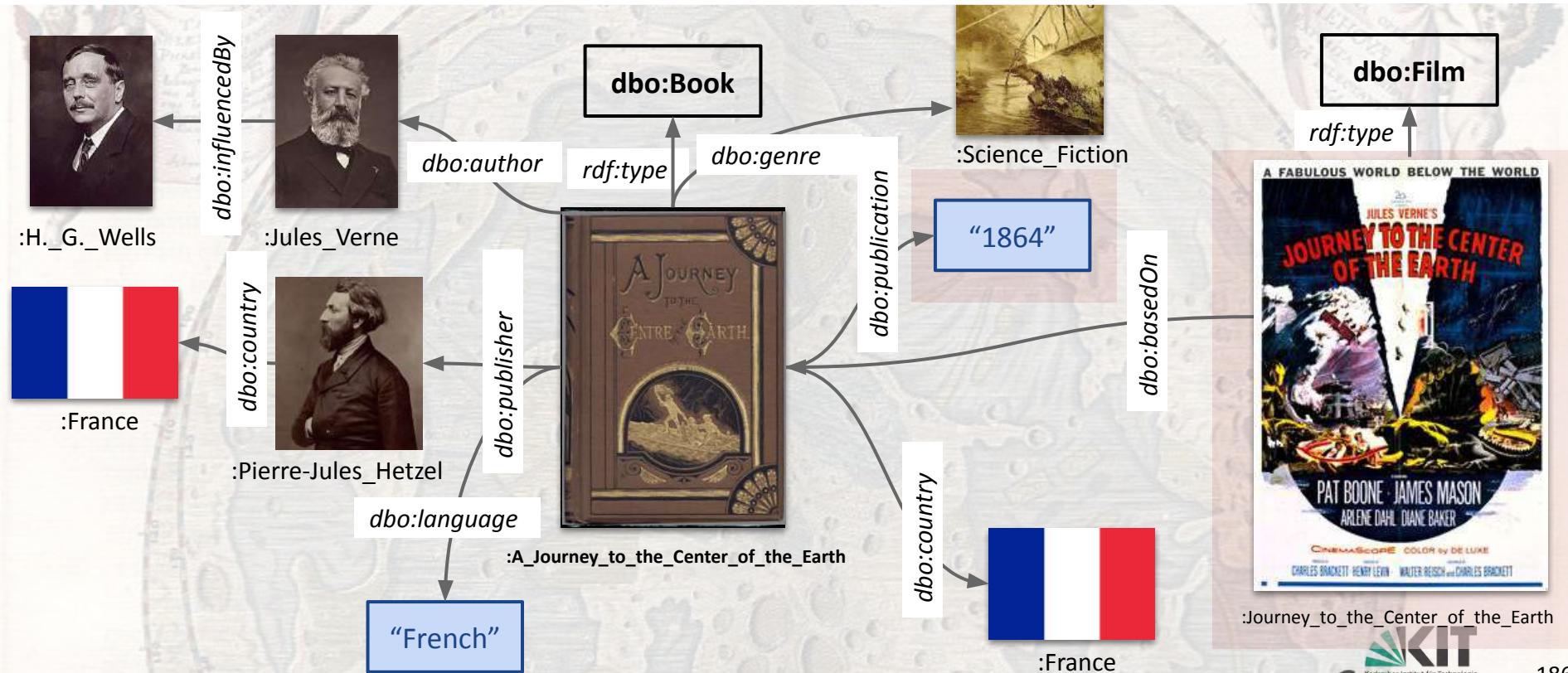
# How to explore the Information Space

- If you **cannot find exactly** what you were looking for or if you **cannot exactly phrase** your search request, then you would be happy to at least **find something nearby**.
- **Similarity:**  
Discover things and documents **similar** (but not necessarily related) to your original search request.
- **Relatedness:**  
Discover things or documents that are closely **related** (but not necessarily similar) to your original search request.

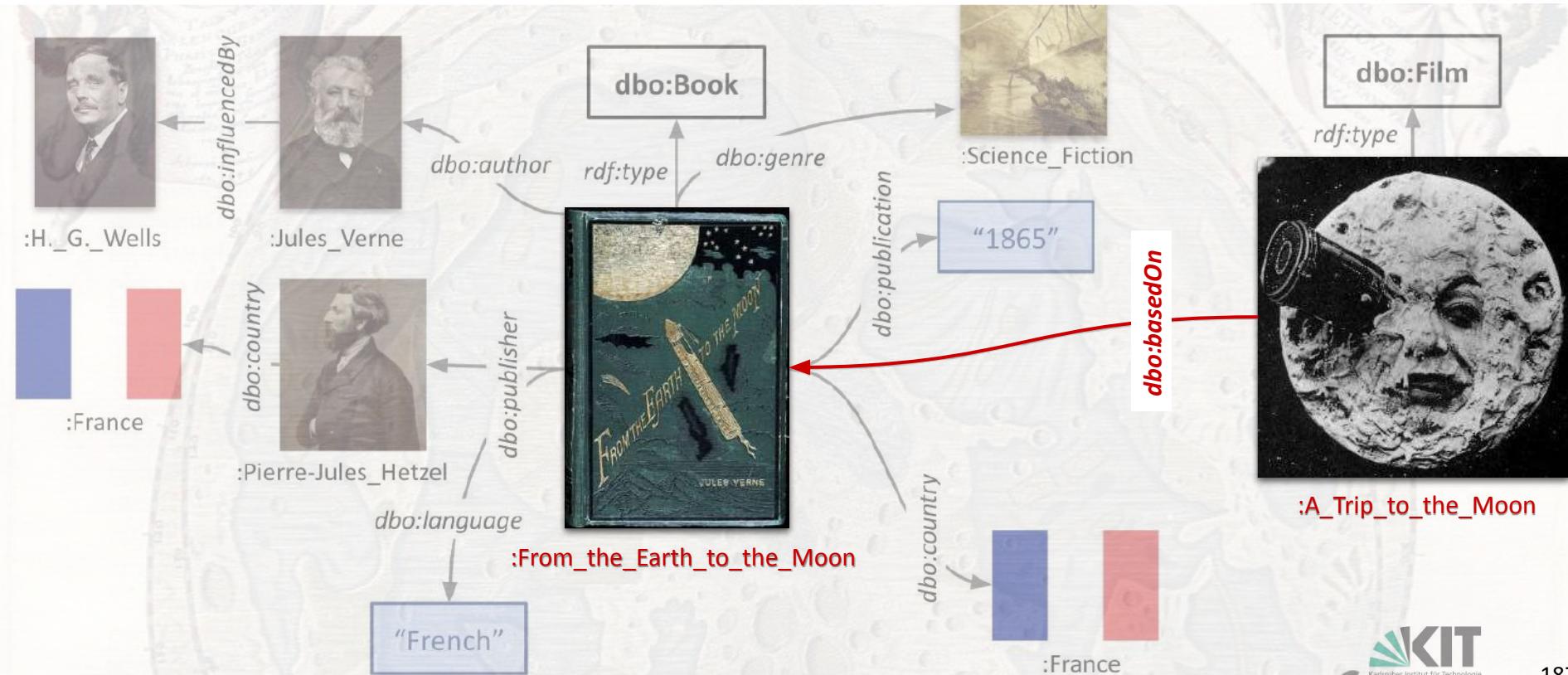
# Similarity vs Relatedness



# Similarity



# Relatedness



# Exploratory Search Strategies

:From\_the\_Earth\_to\_the\_Moon

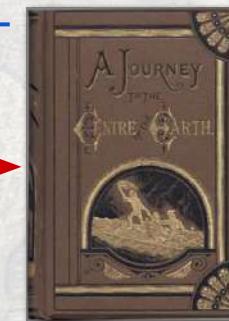


*rdf:type*

**dbo:Book**

*rdf:type*

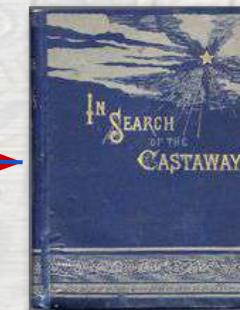
*dbo:subsequentWork*



:A\_Journey\_to\_the\_Center\_of\_the\_Earth

*dbo:previousWork*

*rdf:type*



:In\_Search\_of\_the\_Castaways

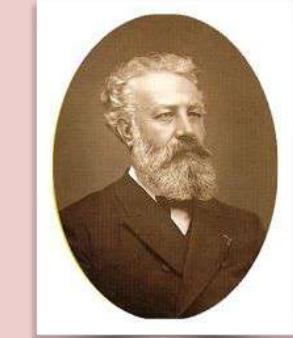
Combining similarity  
and relatedness.

# Exploratory Search Strategies

:From\_the\_Earth\_to\_the\_Moon

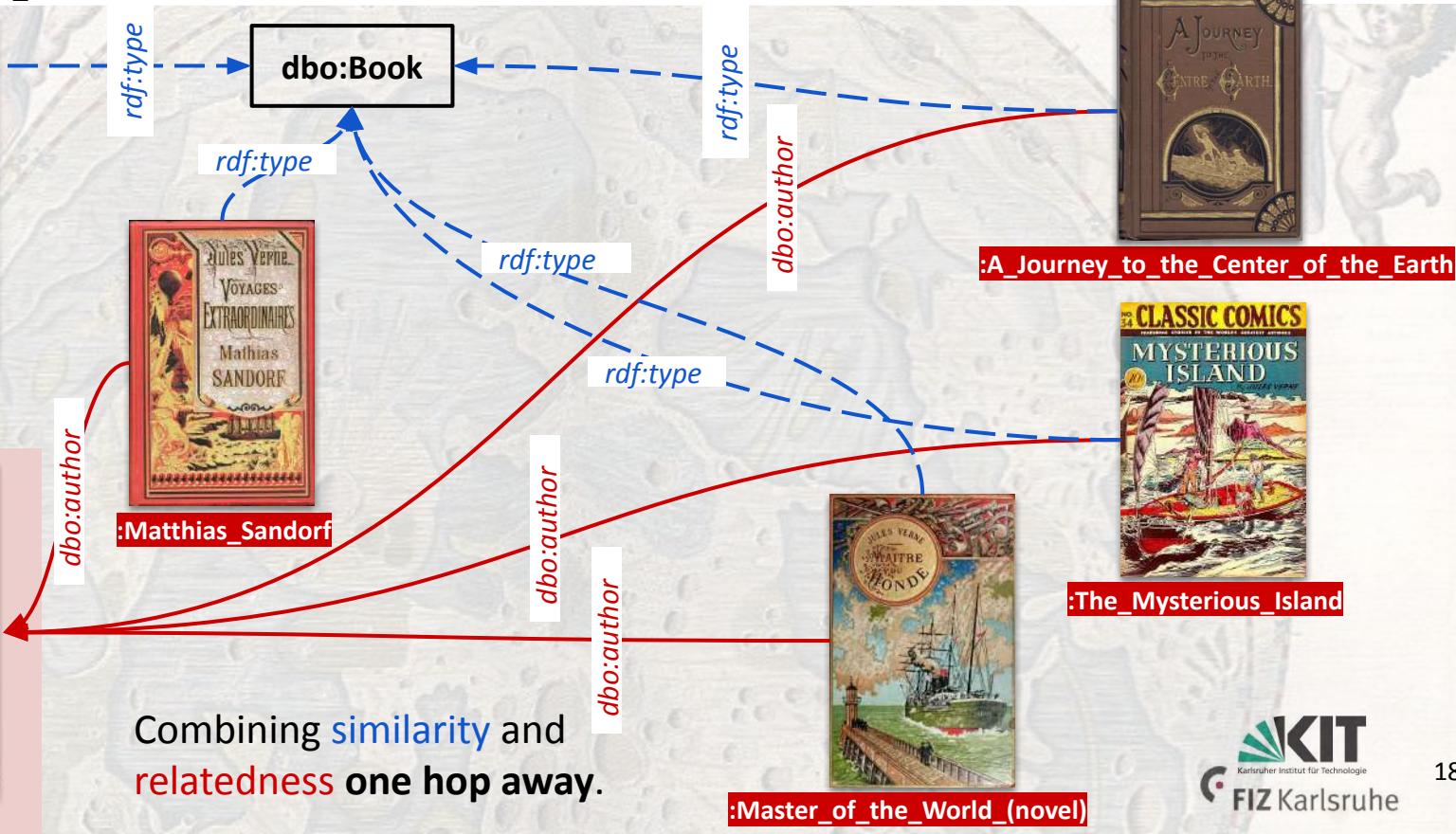


*dbo:author*

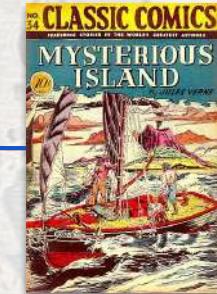


:Jules\_Verne

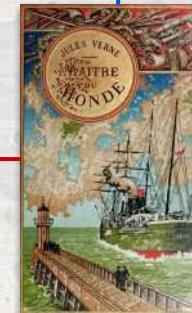
Combining **similarity** and  
relatedness **one hop away.**



:A\_Journey\_to\_the\_Center\_of\_the\_Earth



:A\_Journey\_to\_the\_Center\_of\_the\_Earth



:The\_Mysterious\_Island

# Exploratory Search Strategies

:From\_the\_Earth\_to\_the\_Moon



*dbo:author*



:Jules\_Verne

*rdf:type*

*rdf:type*

*rdf:type*

:H.\_G.\_Wells

*dbo:influenced*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*



*rdf:type*

*rdf:type*

*rdf:type*

*dbo:influenced*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

:The\_Invisible\_Man



*dbo:author*

*rdf:type*

*rdf:type*

*dbo:influenced*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

*dbo:author*

:The\_Island\_of\_Dr\_Moreau



*dbo:author*

*rdf:type*

*rdf:type*

*dbo:influenced*

*dbo:author*

*dbo:author*

*dbo:author*

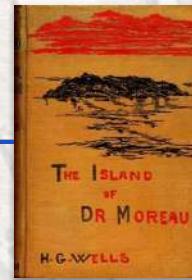
*dbo:author*

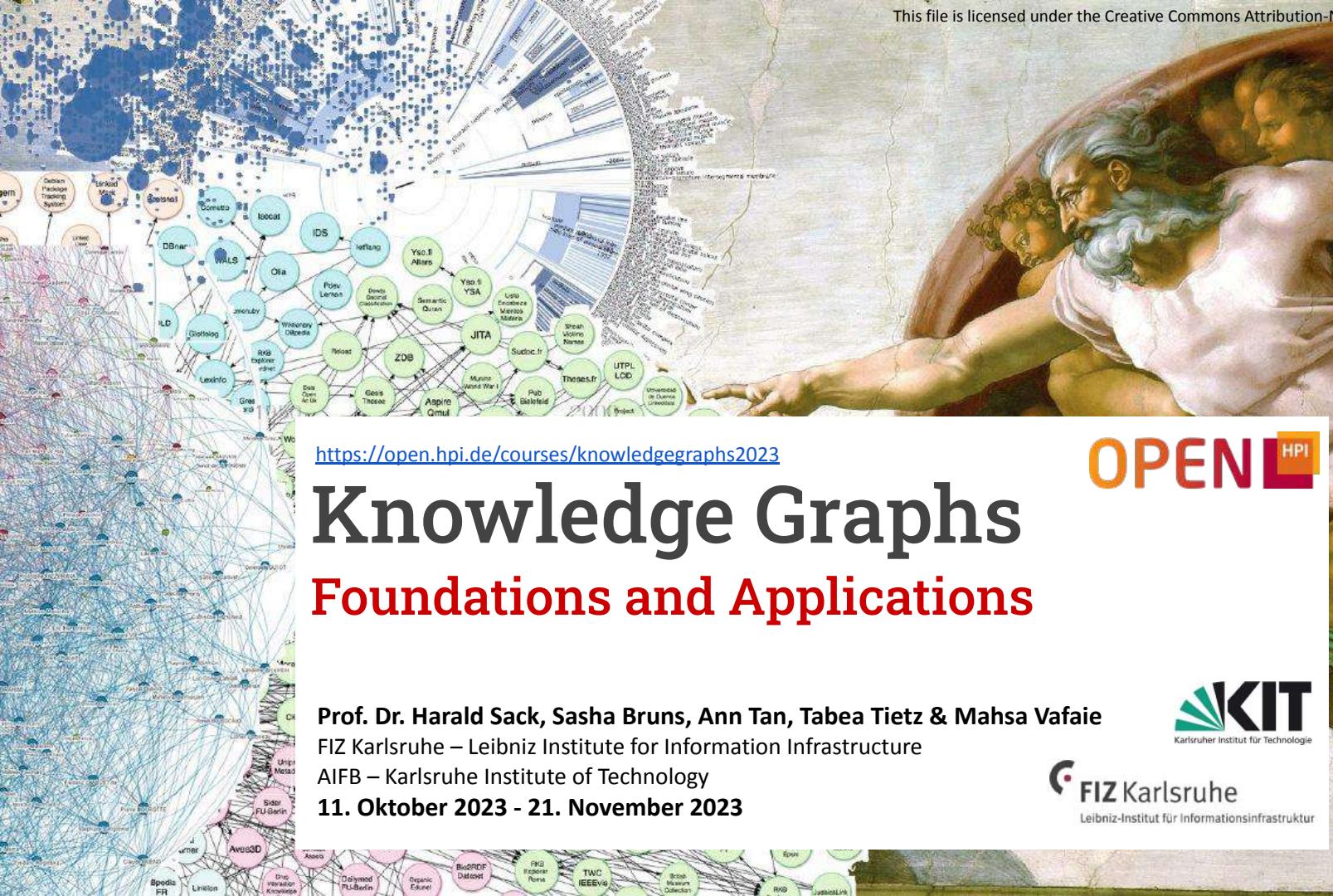
*dbo:author*

*dbo:author*

:The\_War\_of\_the\_Worlds

Combining **similarity** and  
**relatedness** two hops away.





<https://open.hpi.de/courses/knowledgegraphs2023>



# Knowledge Graphs

## Foundations and Applications

Prof. Dr. Harald Sack, Sasha Bruns, Ann Tan, Tabea Tietz & Mahsa Vafaie

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

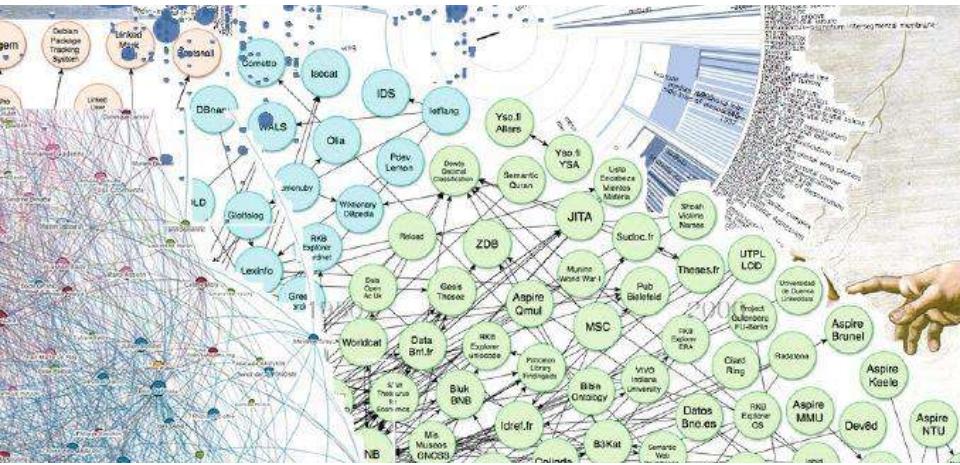
AIFB – Karlsruhe Institute of Technology

11. Oktober 2023 - 21. November 2023



FIZ Karlsruhe  
Leibniz-Institut für Informationsinfrastruktur

**Vielen Dank für Ihre Aufmerksamkeit!**



# Knowledge Graphen

## **Publizieren und forschen mit Wissensgraphen**

Prof. Dr. Harald Sack

FIZ Karlsruhe – Leibniz Institute for Information Infrastructure

AIFB – Karlsruhe Institute of Technology

27.04.2023 - Pausanio Akademie



Karlsruher Institut für Technologie

