KNOWLEDGE GRAPHS 101

Krzysztof Kutt, PhD
Knowledge in AI Systems
WFAIS UJ

Graph Definition

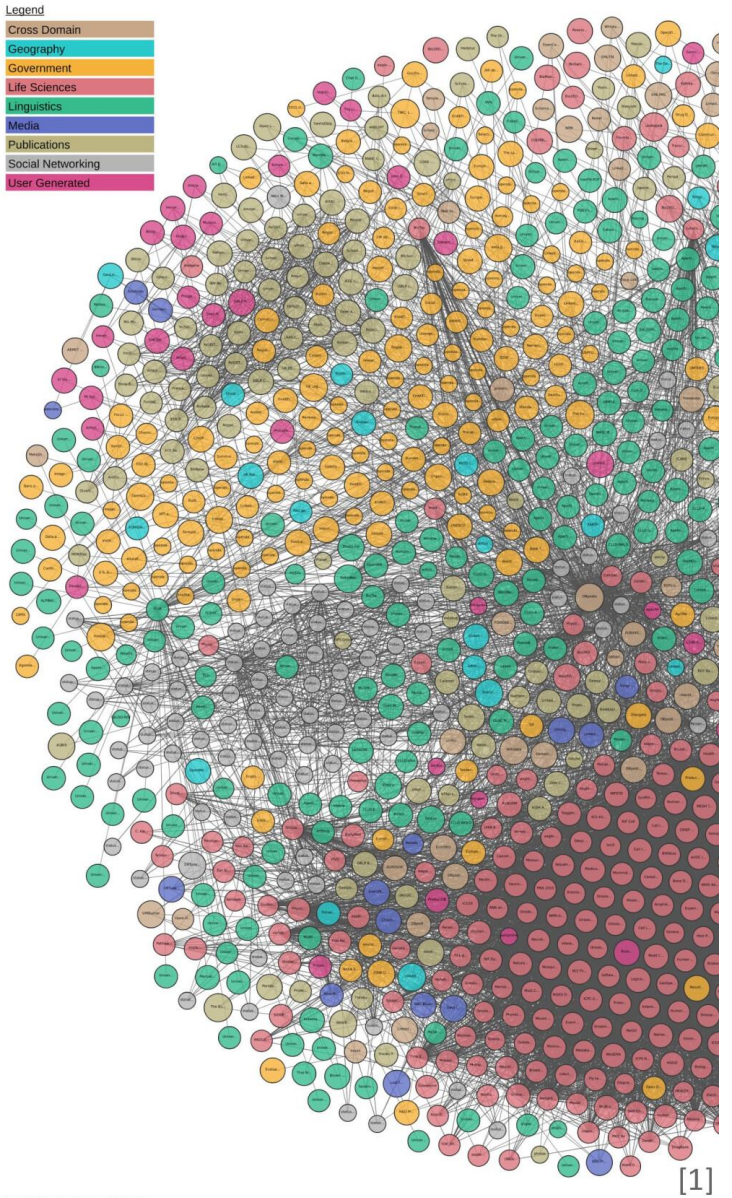
A Knowledge Graph is a Knowledge Base that is a **Graph**.

Definition 1.1

A **simple directed graph** **G=(V,E)** consists of a set **V** of **vertices**, |V|=n, and a set **E** of **directed edges**, $E \subseteq V \times V$, where each edge $e_i=(v_k, v_l)$, $e_i \in E$
is an ordered pair of two vertices (v_k, v_l) with $v_k, v_l \in V$.

Definition 1.2

- A **graph with self-loops** is a graph extended with the option of having edges that relate a vertex to itself.
- A **multi-graph** is a graph that may have multiple edges with the same vertices.
- An **edge-labelled graph** is a graph that has an additional **labelling function** $\lambda : E \rightarrow L$ that maps each edge in E to an element in a set of labels L (similarly for vertex-labelled graphs).



[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Knowledge Base Definition

A Knowledge Graph is a **Knowledge Base** that is a Graph.

A **knowledge base** (**KB**) is a technology used to store complex structured and unstructured information used by a computer system. The initial use of the term was in connection with expert systems which were the first knowledge-based systems.

Wikipedia

knowledge base

Free Online Dictionary of Computing

‹*artificial intelligence*›

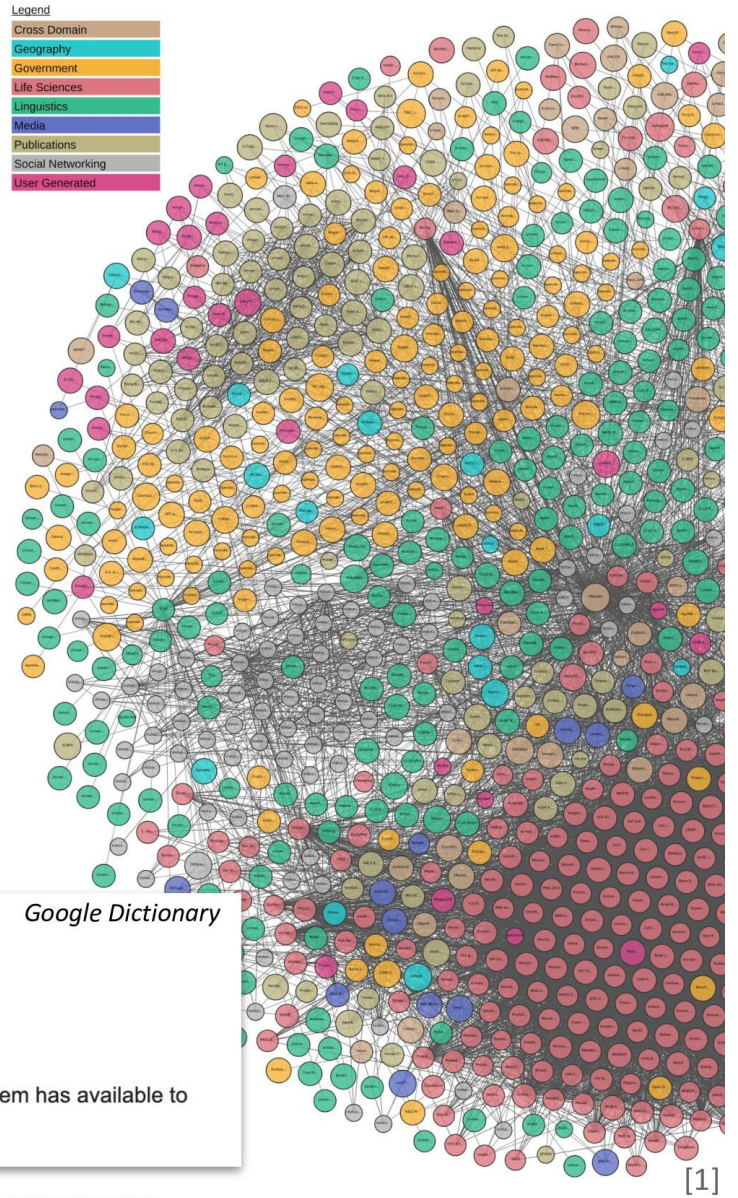
A collection of knowledge expressed using some formal knowledge representation language. A knowledge base forms part of a knowledge-based system (KBS).

knowledge base

Google Dictionary

noun

1. a store of information or data that is available to draw on.
2. the underlying set of facts, assumptions, and rules which a computer system has available to solve a problem.



[1]

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

KNOWLEDGE GRAPH

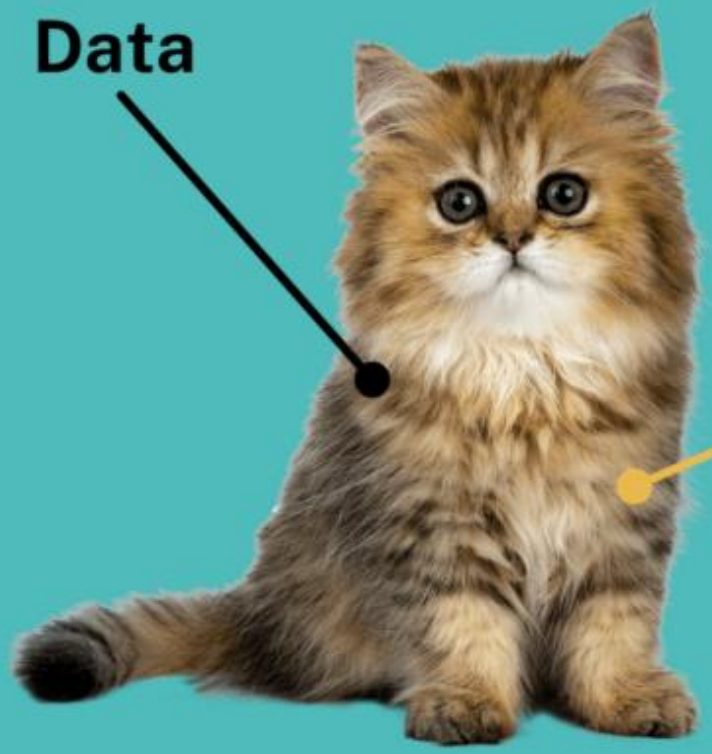
There is no single definition (they are even conflicting sometimes),
but to start with something...

Knowledge graph is a graph of data intended to accumulate and convey **knowledge of the real world**, whose **nodes represent entities** of interest and whose **edges represent** potentially different **relations** between these entities

Definition by: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

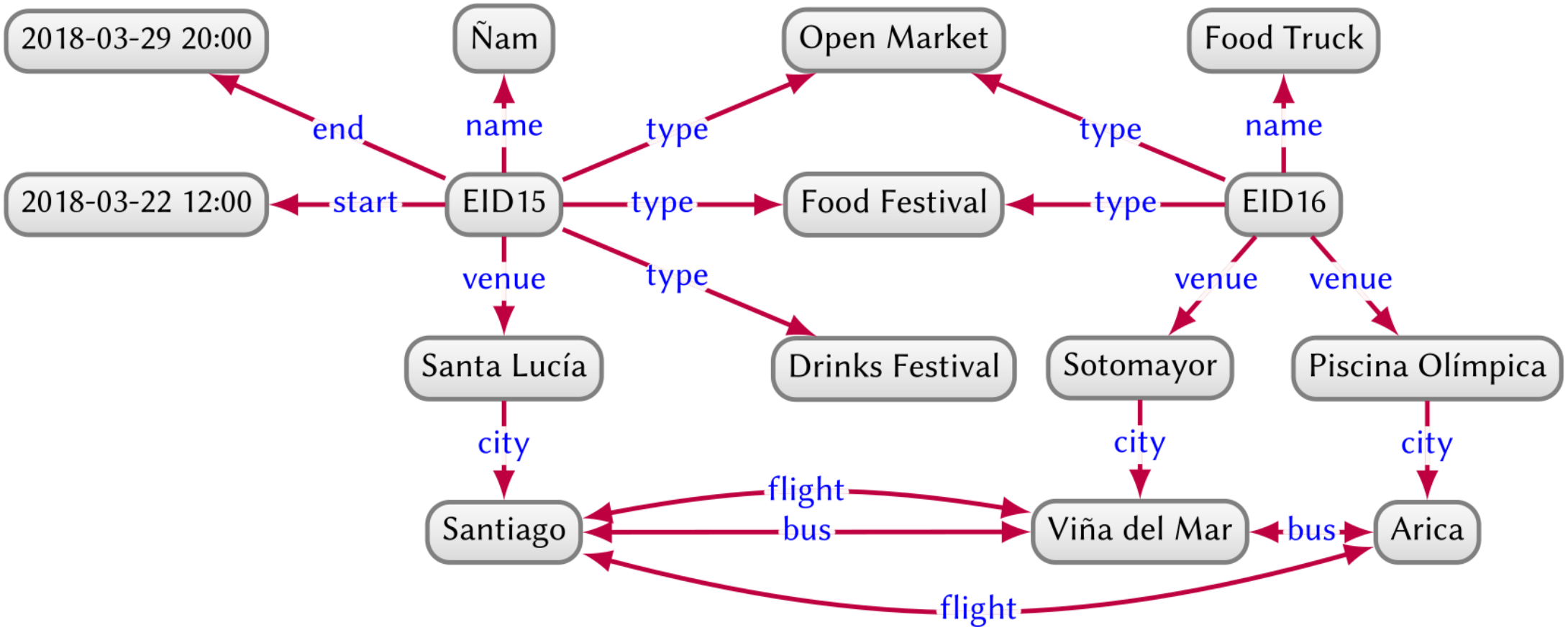
Metadata

Data



Filename:
Floofy
Author:
Hive Art Media
Date:
October 20, 2021

KNOWLEDGE GRAPH



WHY?

1. An **intuitive way** to represent knowledge
2. Flexible way to conceptualise, represent, and integrate **diverse and incomplete** data
3. Flexible integration of **different knowledge**, and concepts into one **single unified graph**
4. Facilitate **transfer learning** and **explainability** (different domains may have similar data structure)
5. Controlled **outputs** from ML models **with high consistency** (e.g., no hallucinations, as in LLMs)
6. …

MODELS

How do knowledge graphs look like?

DIRECTED EDGE-LABELLED GRAPHS

- Base for RDF
- Nodes (entities) and edges (binary relations)
- To extend the knowledge graph:
  add new edges (and nodes)
- Incomplete knowledge:
  simply omit a particular edge

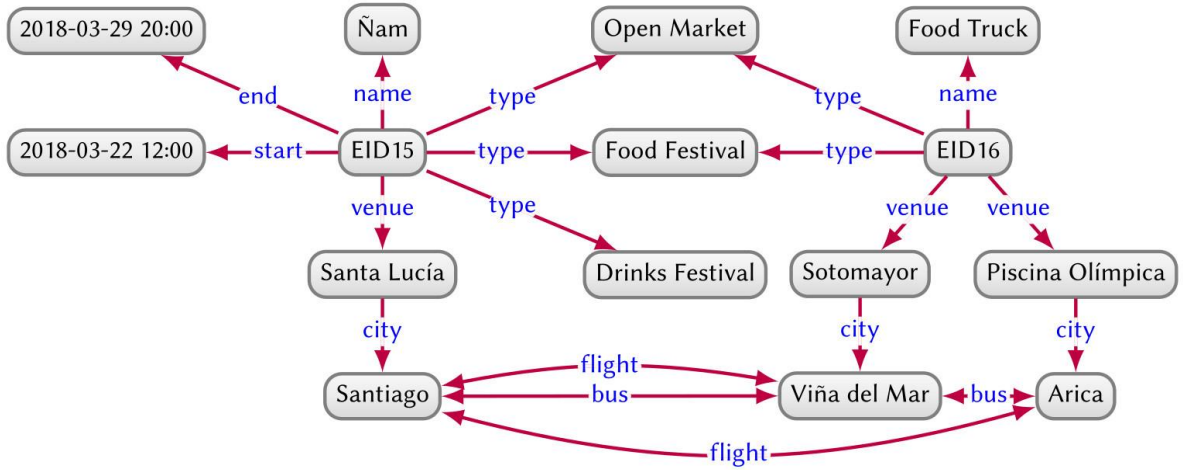


Fig. 1. Directed-edge labelled graph describing events and their venues.

Source: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

HETEROGENEOUS GRAPHS

- A.k.a. heterogeneous information networks
- Nodes and edges have types (as a part of the model itself; typically exactly one type)

(a) Del graph

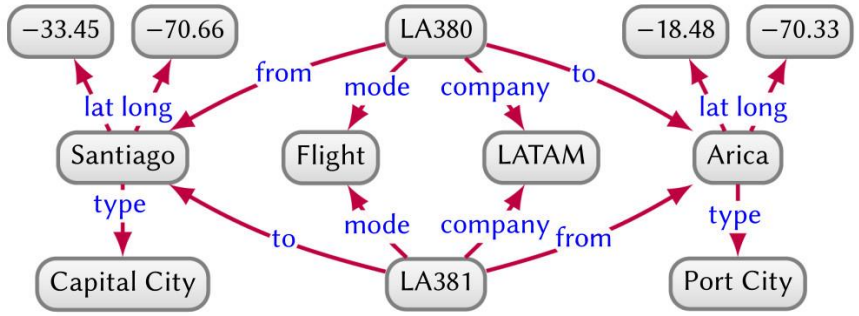
(b) Heterogeneous graph

Fig. 2. Data about capitals and countries in a del graph and a heterogeneous graph.

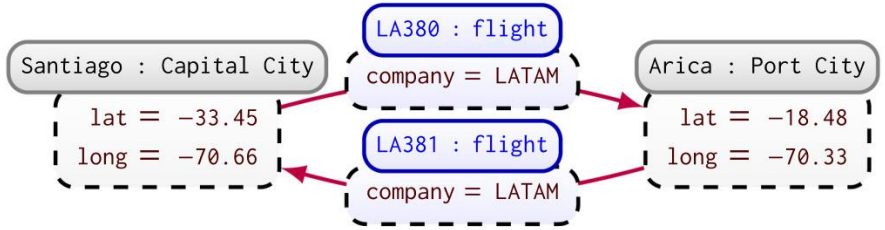
Source: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

PROPERTY GRAPHS

- Labels and property-value pairs can be associated with nodes and edges
- Not yet standardised
  (available in popular graph databases
  but particular implementations may differ)
- More intuitive representation, but requires more intricate query languages, formal semantics and inductive techniques



(a) Del graph



(b) Property graph

Source: A. Hogan et al. (2022), Knowledge Graphs, ACM Computing Surveys, 54(4), pp. 1-37.

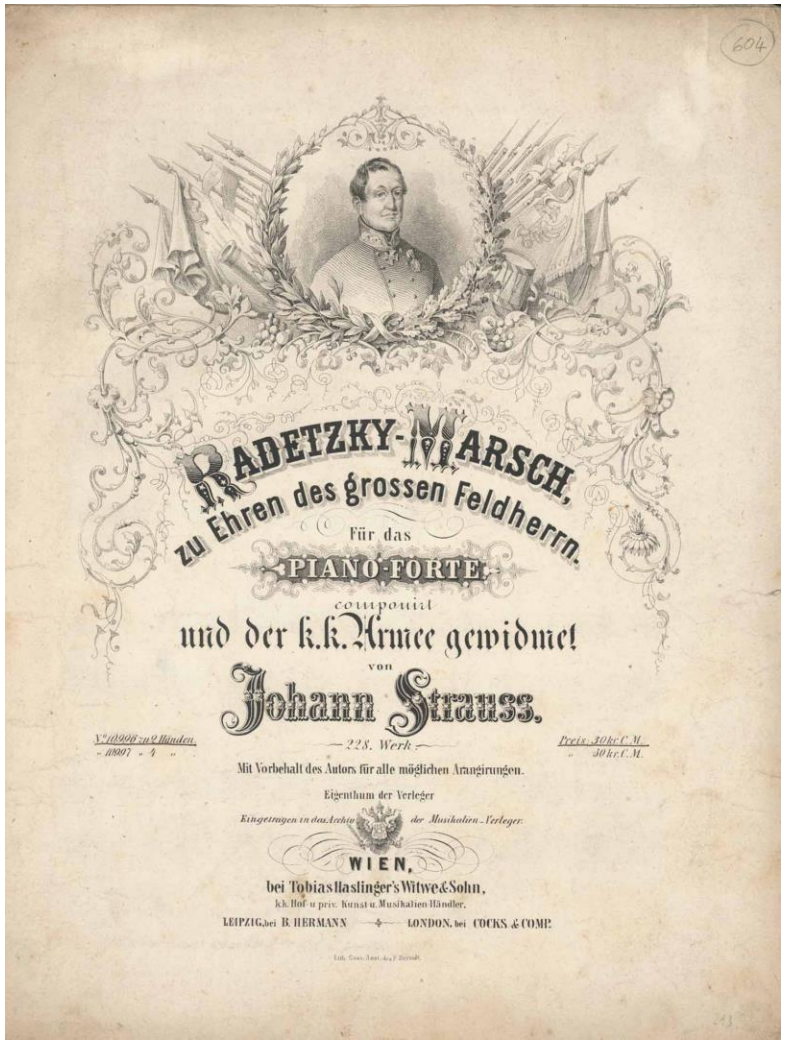
OTHER MODELS

- Models with hypernodes with nested graphs in nodes
- **Hypergraphs** allow edges that connect sets rather than pairs of nodes
- *And many others*, but data can be typically **converted from one model to another**

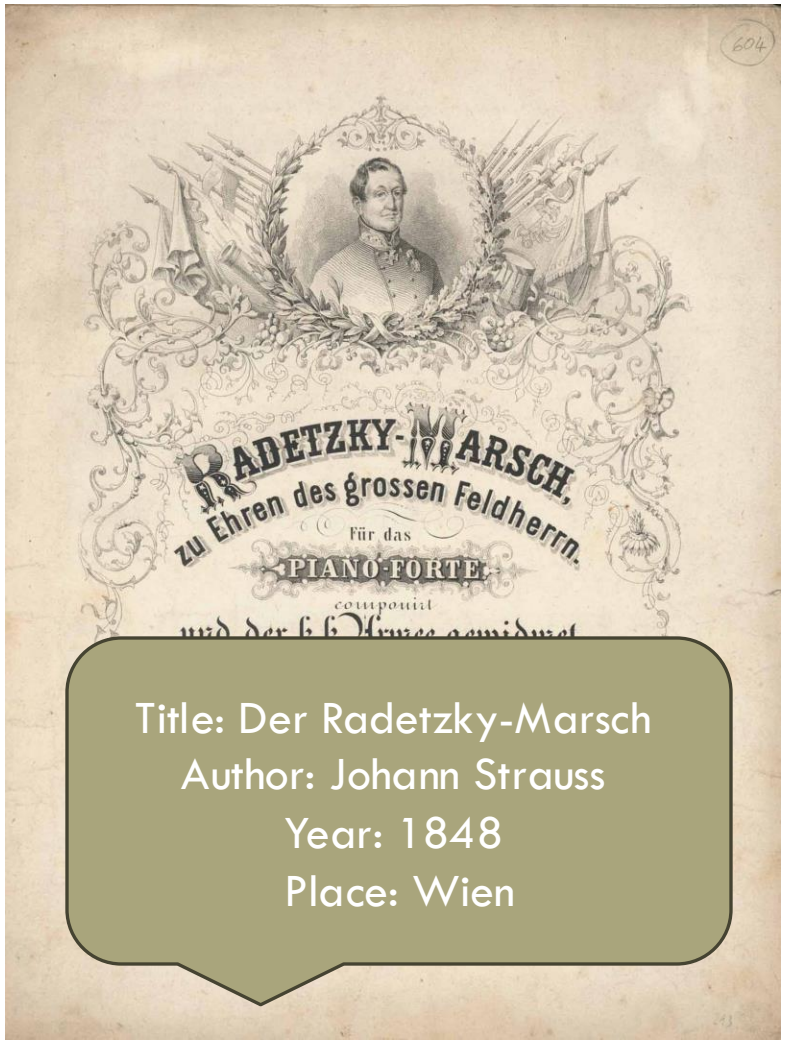
RDF MODEL

Simple graph in cultural heritage domain

LET'S START WITH SOME DOCUMENTS . . .



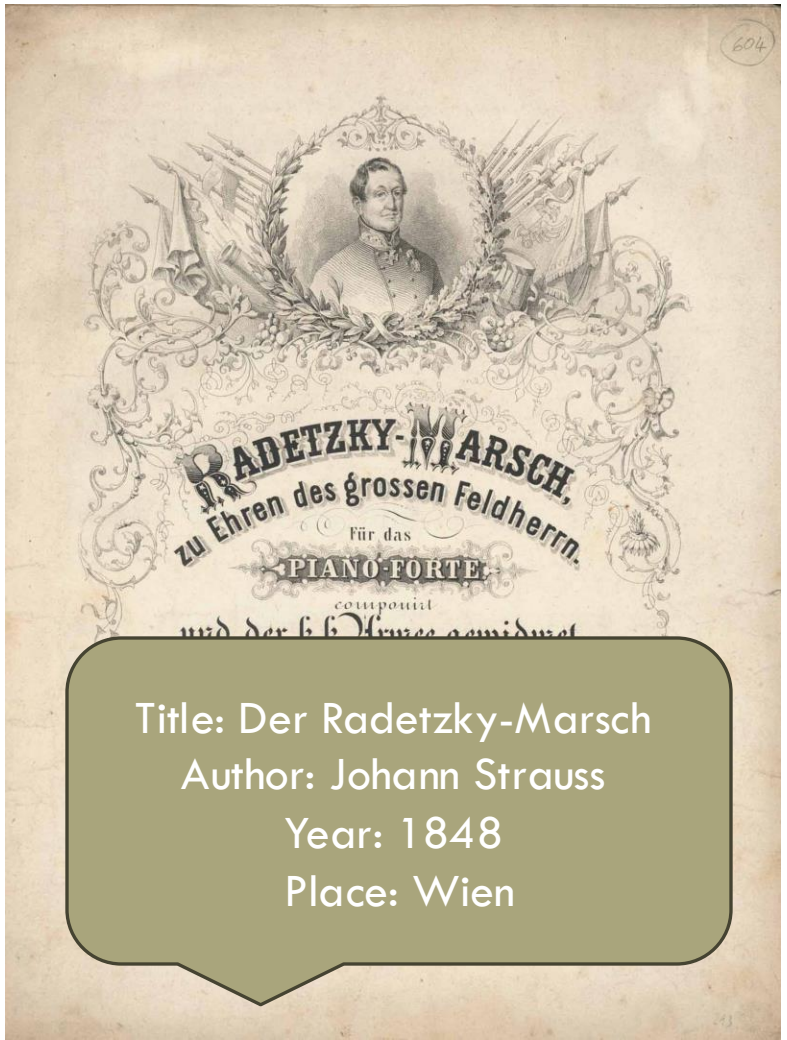
. . . AND THEIR METADATA



Title: Der Radetzky-Marsch
Author: Johann Strauss
Year: 1848
Place: Wien

Title: An der schönen blauen Donau
Author: Johann Strauss
Year: 1867
Place: Wien

STEP 1: SIMPLE SENTENCES



Title: Der Radetzky-Marsch
Author: Johann Strauss
Year: 1848
Place: Wien

The 1st document has title "Der Radetzky-Marsch".

Its author is Johann Strauss.

It was composed in 1848 in Wien.

STEP 1: SIMPLE SENTENCES

Title: An der schönen blauen Donau
Author: Johann Strauss
Year: 1867
Place: Wien

The 1st document has title "Der Radetzky-Marsch".

Its author is Johann Strauss.

It was composed in 1848 in Wien.

The 2nd document has title "An der schönen blauen Donau".

Its author is Johann Strauss.

It was composed in 1867 in Wien.

STEP 2: DIVIDE INTO TRIPLES WITH EXACTLY ONE INFO (SUBJECT, PREDICATE, OBJECT)

The 1st document has title "Der Radetzky-Marsch".

Its author is Johann Strauss.

It was composed in 1848 in Wien.

The 2nd document has title "An der schönen blauen Donau".

Its author is Johann Strauss.

It was composed in 1867 in Wien.

STEP 2: DIVIDE INTO TRIPLES WITH EXACTLY ONE INFO (SUBJECT, PREDICATE, OBJECT)

The 1st document has title "Der Radetzky-Marsch".

Its author is Johann Strauss.

It was composed in 1848 in Wien.

The 2nd document has title "An der schönen blauen Donau".

Its author is Johann Strauss.

It was composed in 1867 in Wien.

(1st document) (has title) (Der Radetzky-Marsch).

STEP 2: DIVIDE INTO TRIPLES WITH EXACTLY ONE INFO (SUBJECT, PREDICATE, OBJECT)

The 1st document has title "Der Radetzky-Marsch".

Its author is Johann Strauss.

It was composed in 1848 in Wien.

(1st document) (has title) (Der Radetzky-Marsch).

(1st document) (has author) (Johann Strauss).

(1st document) (composed in) (1848).

(1st document) (composed in) (Wien).

The 2nd document has title "An der schönen blauen Donau".

Its author is Johann Strauss.

It was composed in 1867 in Wien.

(2nd document) (has title) (An der schönen blauen Donau).

(2nd document) (has author) (Johann Strauss).

(2nd document) (composed in) (1867).

(2nd document) (composed in) (Wien).

STEP 3: CREATE NODES AND EDGES

(1st document) (has title) (Der Radetzky-Marsch).

(1st document) (composed in) (1848).

(1st document) (composed in) (Wien).

(1st document) (has author) (Johann Strauss).

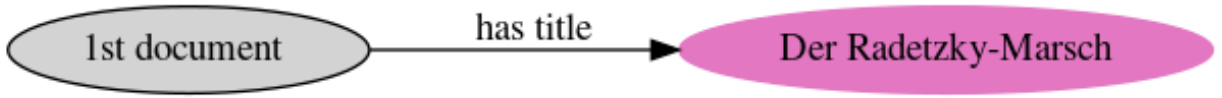
(2nd document) (has title) (An der schönen blauen Donau).

(2nd document) (composed in) (1867).

(2nd document) (composed in) (Wien).

(2nd document) (has author) (Johann Strauss).

STEP 3: CREATE NODES AND EDGES



(1st document) (has title) (Der Radetzky-Marsch).

(1st document) (composed in) (1848).

(1st document) (composed in) (Wien).

(1st document) (has author) (Johann Strauss).

(2nd document) (has title) (An der schönen blauen Donau).

(2nd document) (composed in) (1867).

(2nd document) (composed in) (Wien).

(2nd document) (has author) (Johann Strauss).

STEP 3: CREATE NODES AND EDGES

(1st document) (has title) (Der Radetzky-Marsch).

(1st document) (composed in) (1848).

(1st document) (composed in) (Wien).

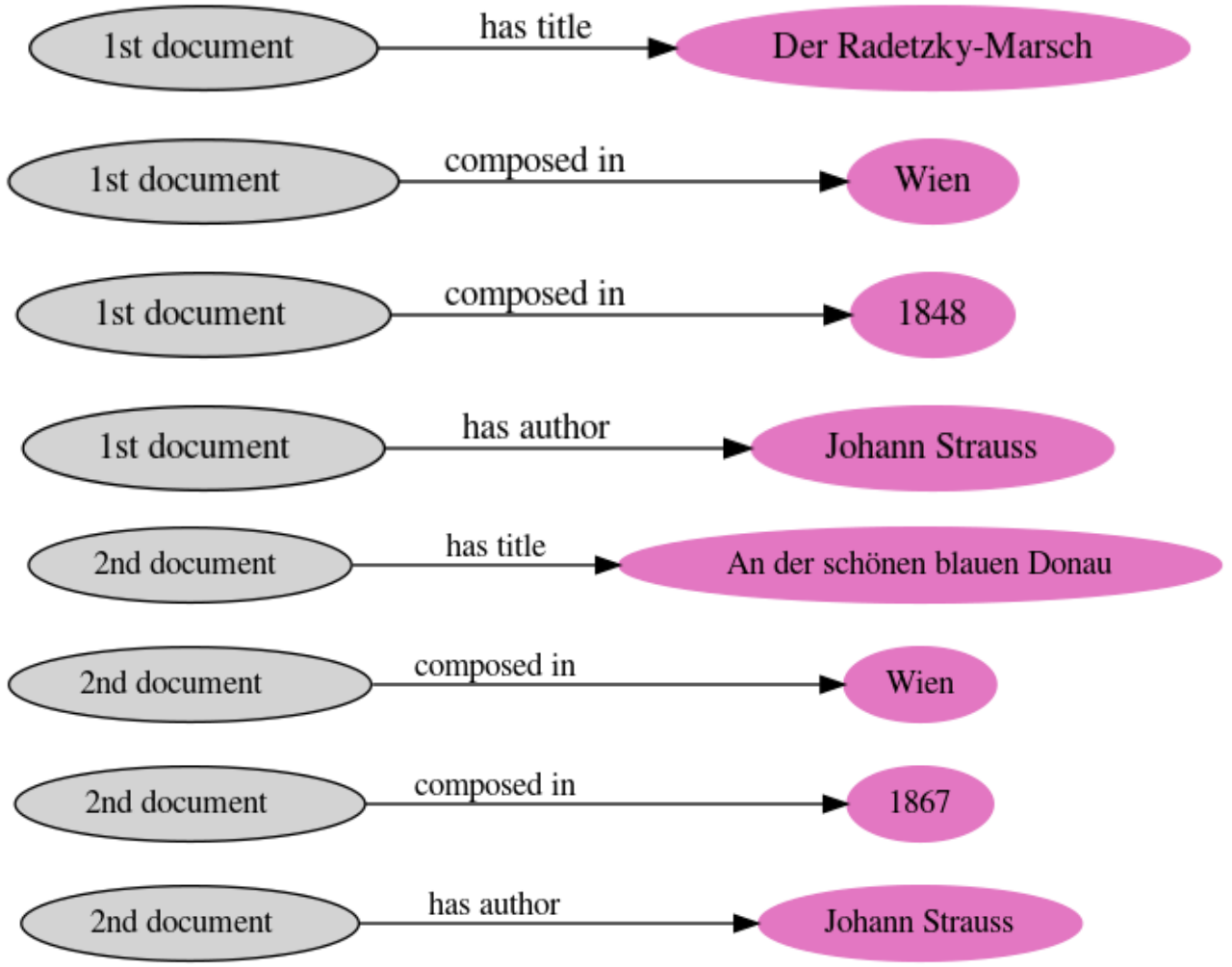
(1st document) (has author) (Johann Strauss).

(2nd document) (has title) (An der schönen blauen Donau).

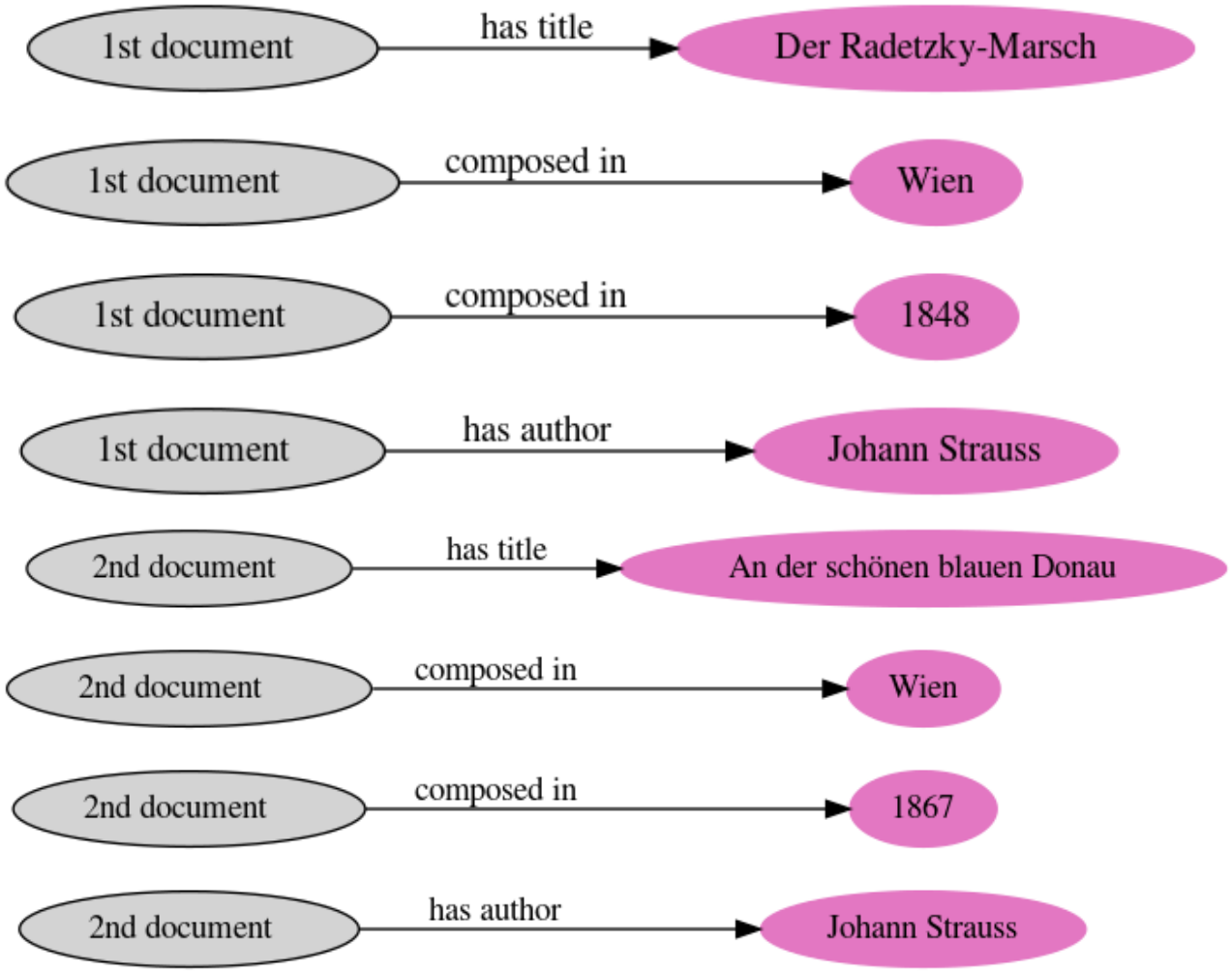
(2nd document) (composed in) (1867).

(2nd document) (composed in) (Wien).

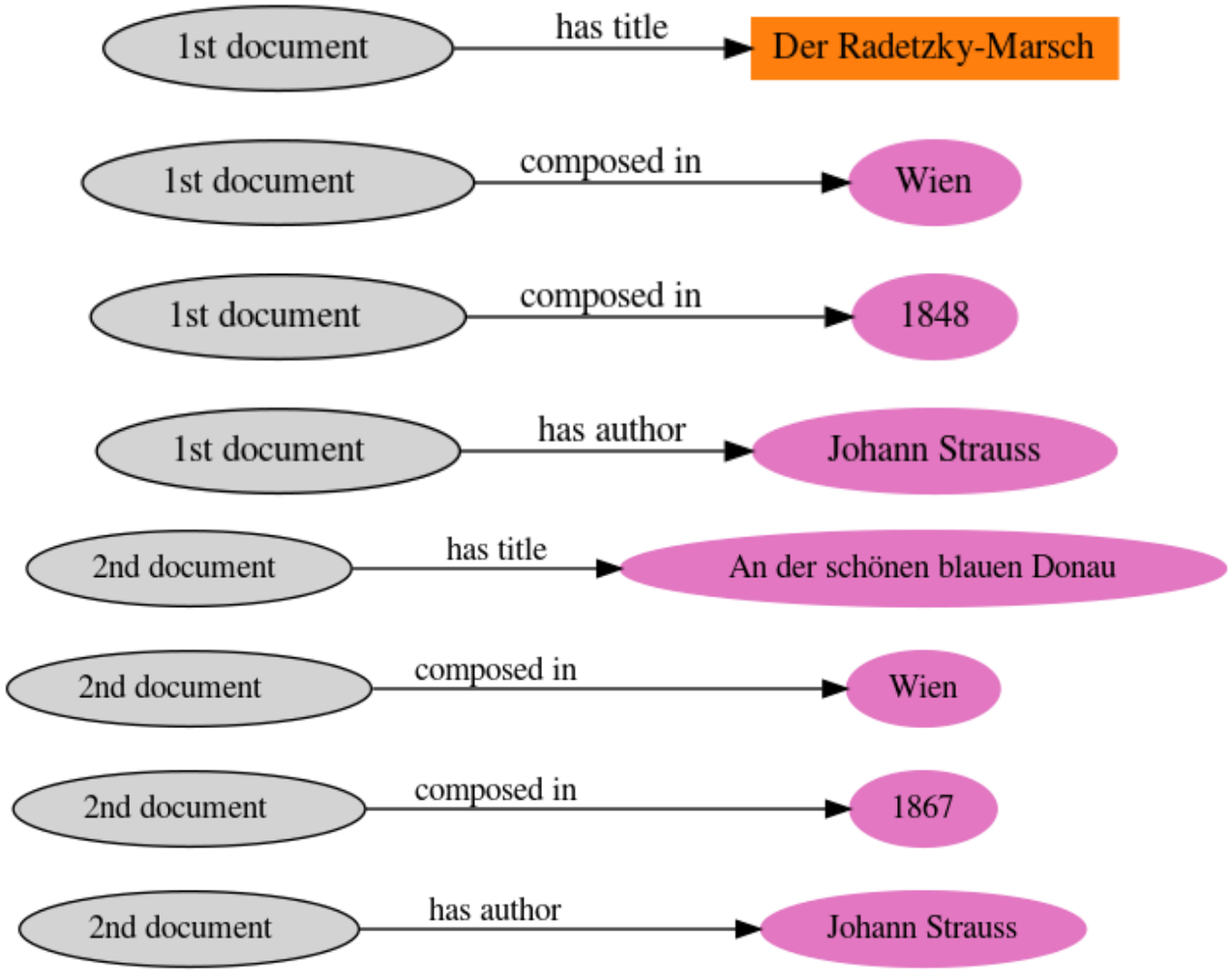
(2nd document) (has author) (Johann Strauss).



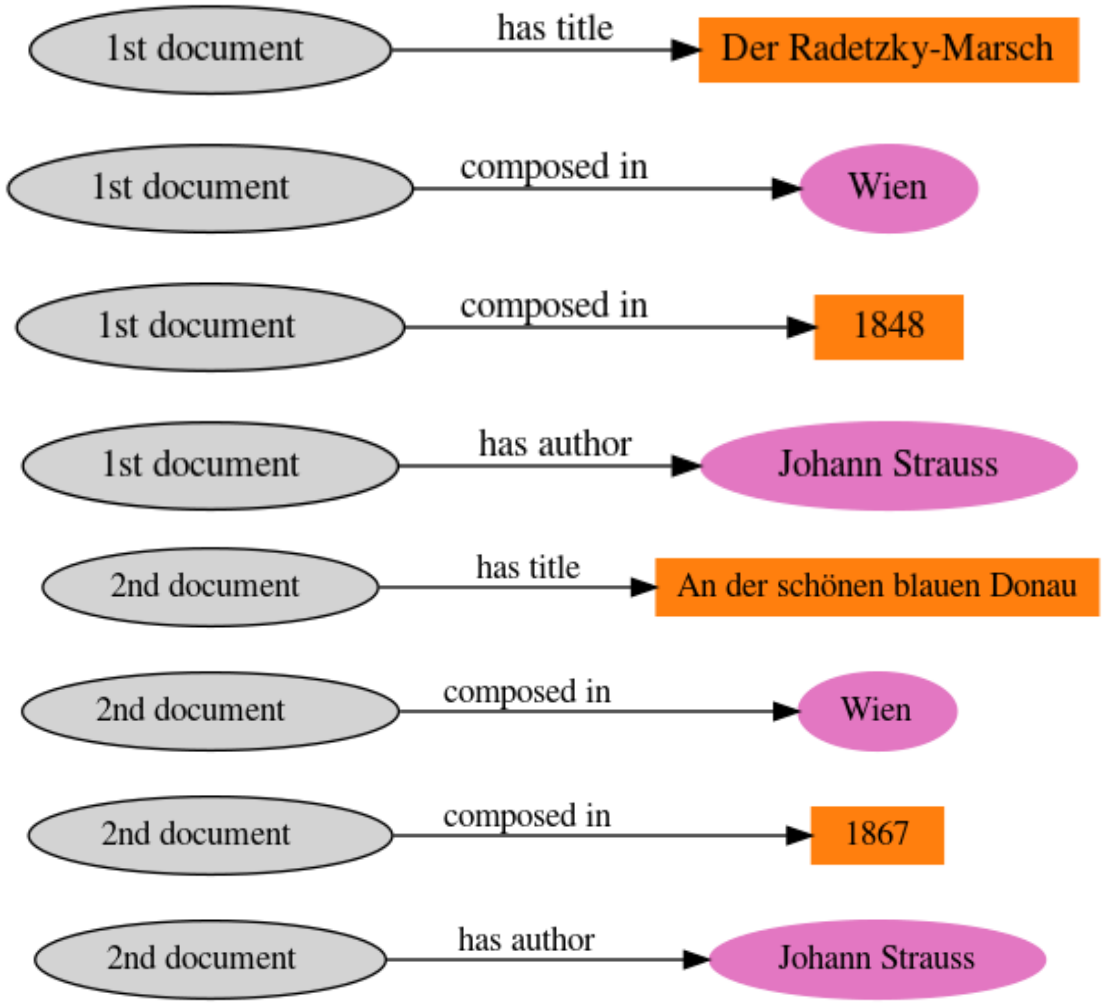
STEP 4: IDENTIFY THINGS AND LITERALS



STEP 4: IDENTIFY THINGS AND LITERALS



STEP 4: IDENTIFY THINGS AND LITERALS



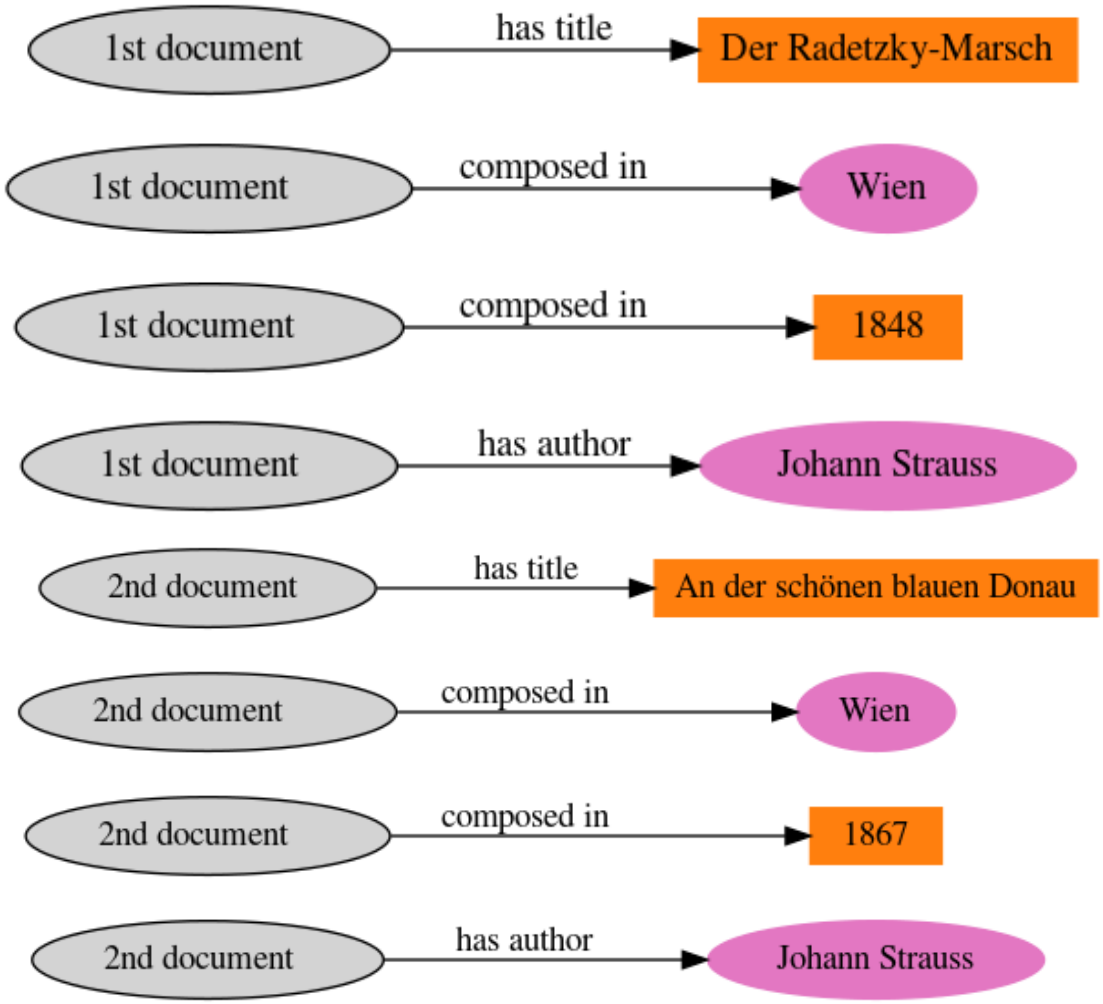
STEP 5: UNIQUE IDENTIFIERS

1st document = 1st document?

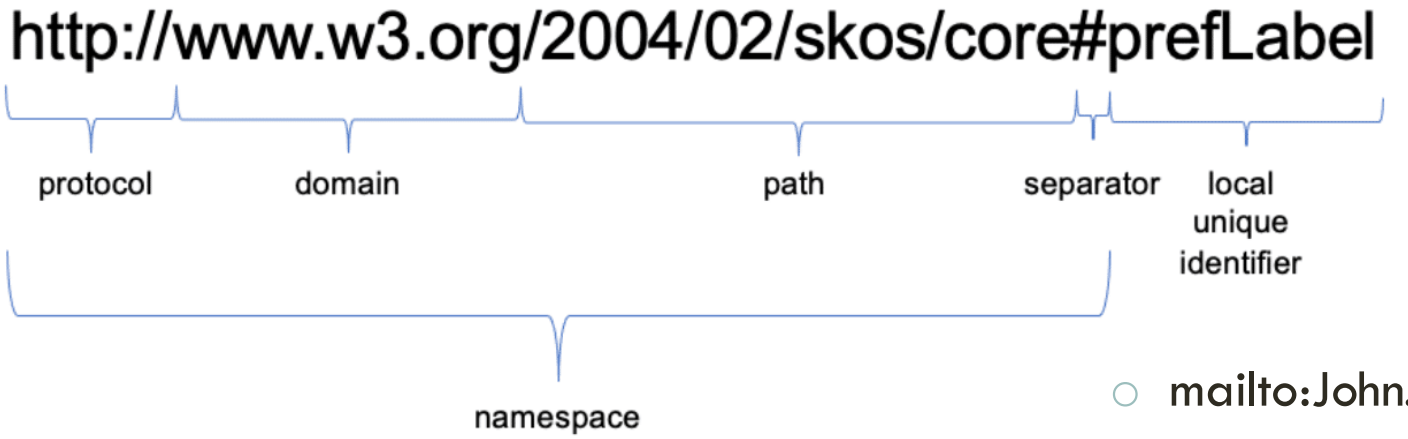
2nd document = 2nd document?

Wien = Wien?

Johann Strauss = Johann Strauss?



STEP 5: UNIQUE IDENTIFIERS



- mailto:John.Doe@example.com
- tel:+1-816-555-1212
- doi:10.1016/j.eswa.2023.119968
- urn:isbn:0451450523
- urn:ISSN:0167-6423
- urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66

Image from Resolving Uniform Resource Identifiers (URIs).

STEP 5A: UNIQUE IDENTIFIERS FOR THINGS

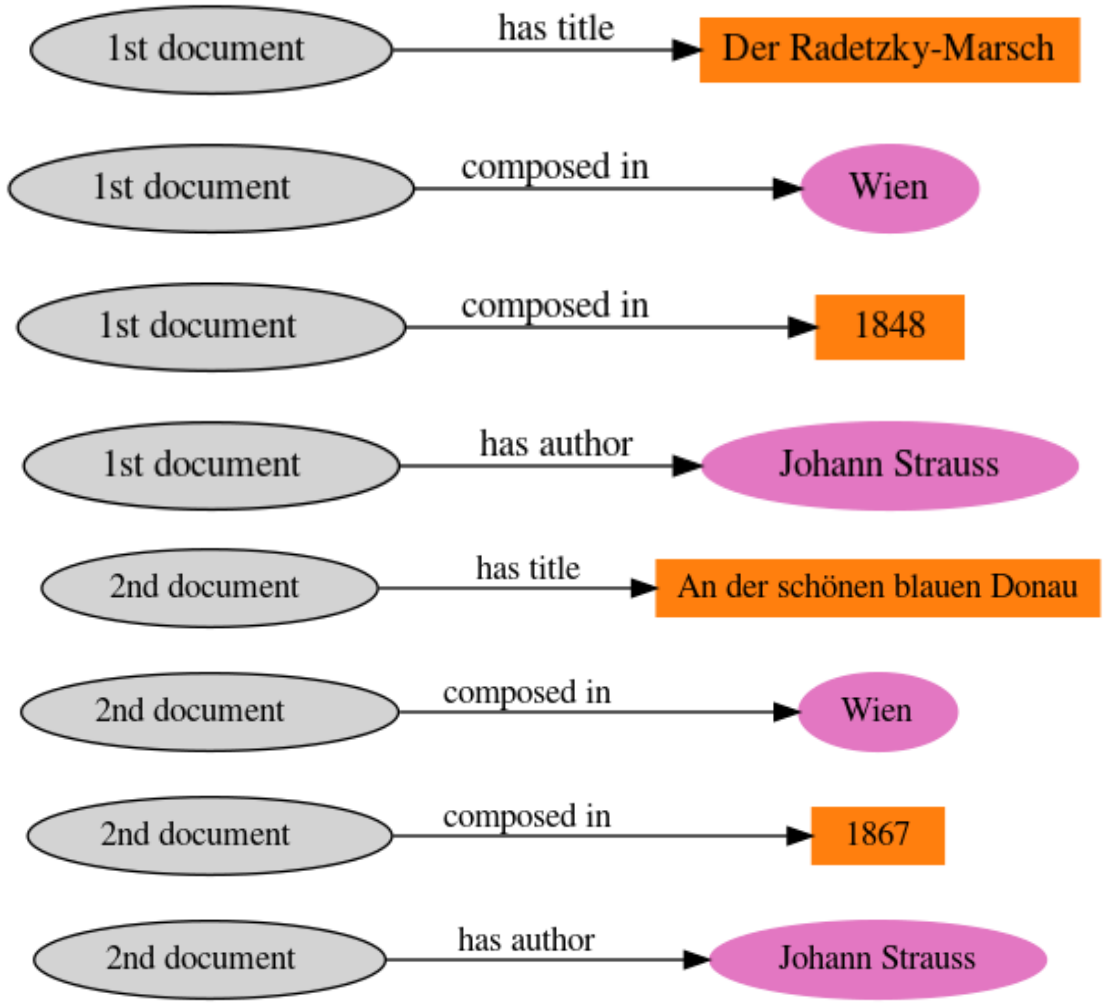
1st document = doi:strauss/march1

2nd document = doi:strauss/waltz1

Wien = https://www.geonames.org/2761369/

Johann Strauss = https://d-nb.info/gnd/118619098

Johann Strauss = https://d-nb.info/gnd/11861908X



STEP 5A: UNIQUE IDENTIFIERS FOR THINGS

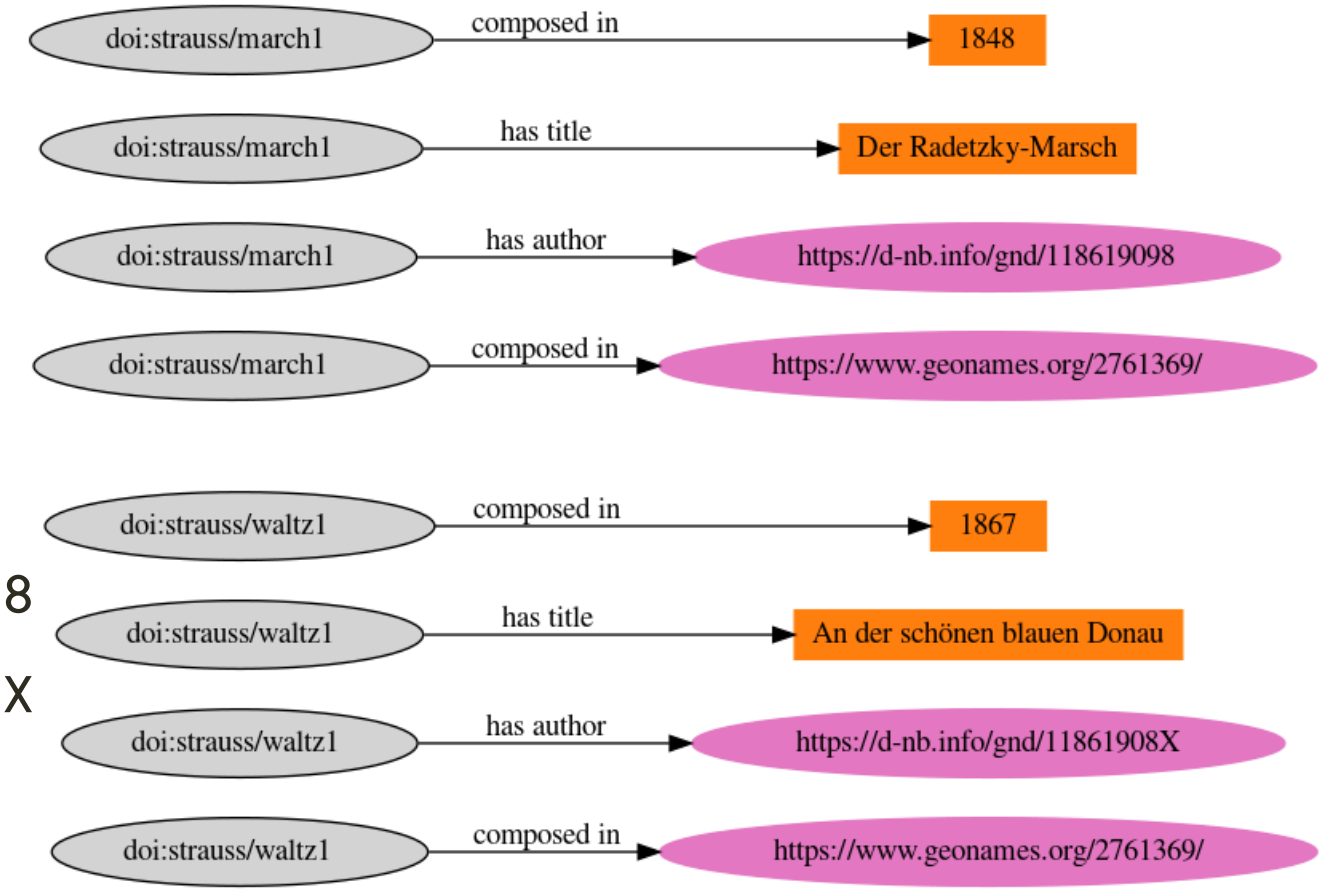
1st document = doi:strauss/march1

2nd document = doi:strauss/waltz1

Wien = https://www.geonames.org/2761369/

Johann Strauss = https://d-nb.info/gnd/118619098

Johann Strauss = https://d-nb.info/gnd/11861908X



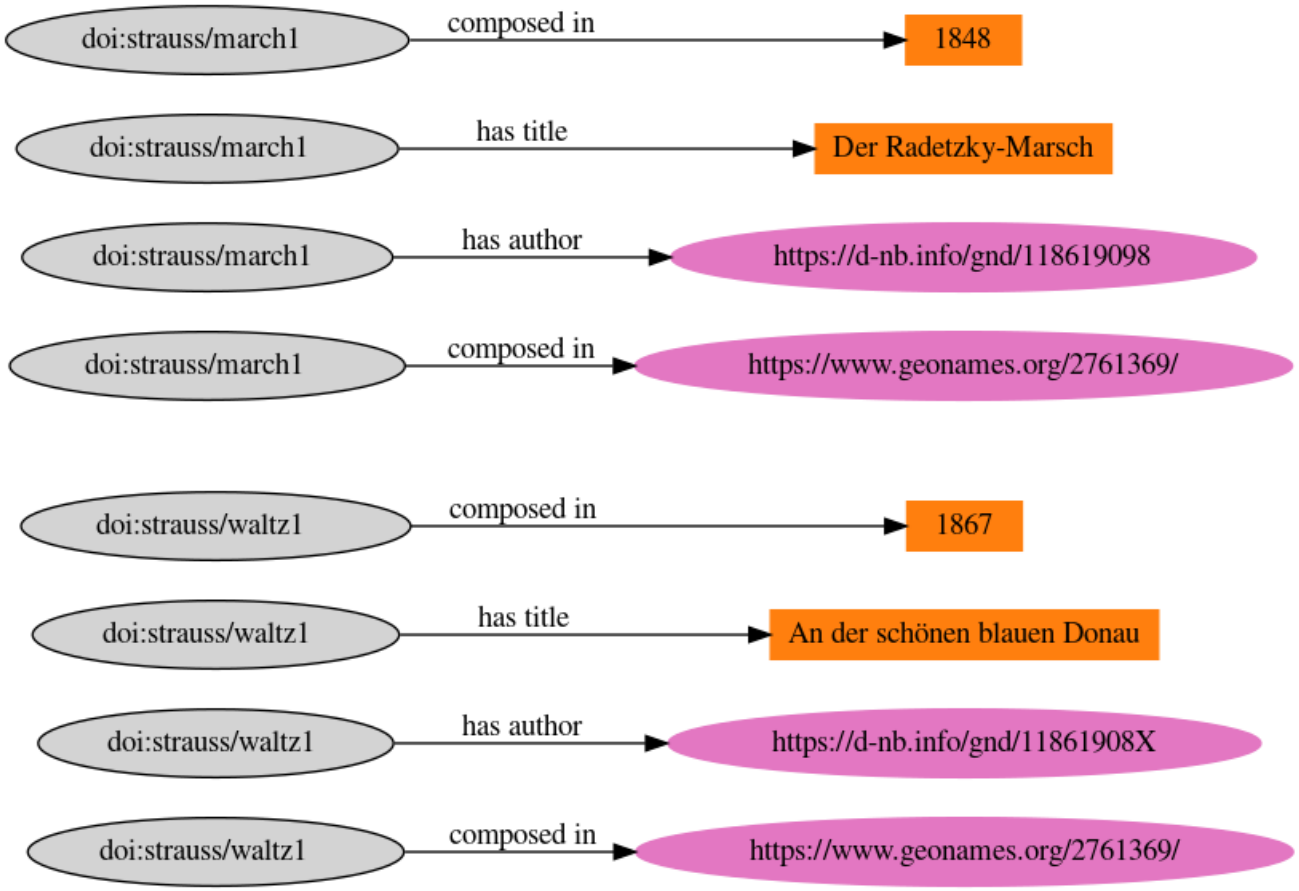
STEP 5B: UNIQUE IDENTIFIERS FOR PREDICATES

composed in =
https://uj.edu.pl/dict/composed_year

composed in =
https://uj.edu.pl/dict/composed_place

has title = https://uj.edu.pl/dict/has_title

has author = https://uj.edu.pl/dict/has_author



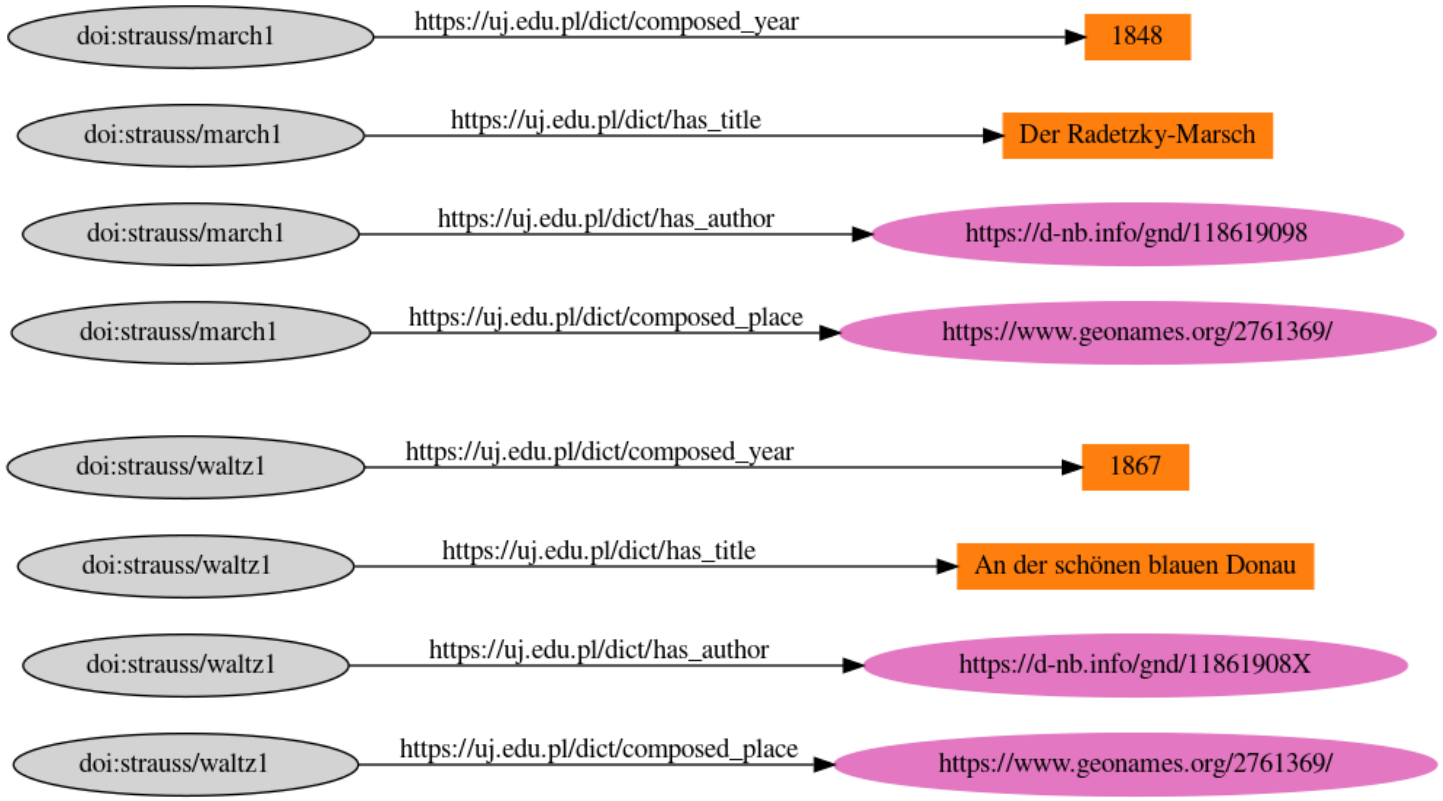
STEP 5B: UNIQUE IDENTIFIERS FOR PREDICATES

composed in = https://uj.edu.pl/dict/composed_year

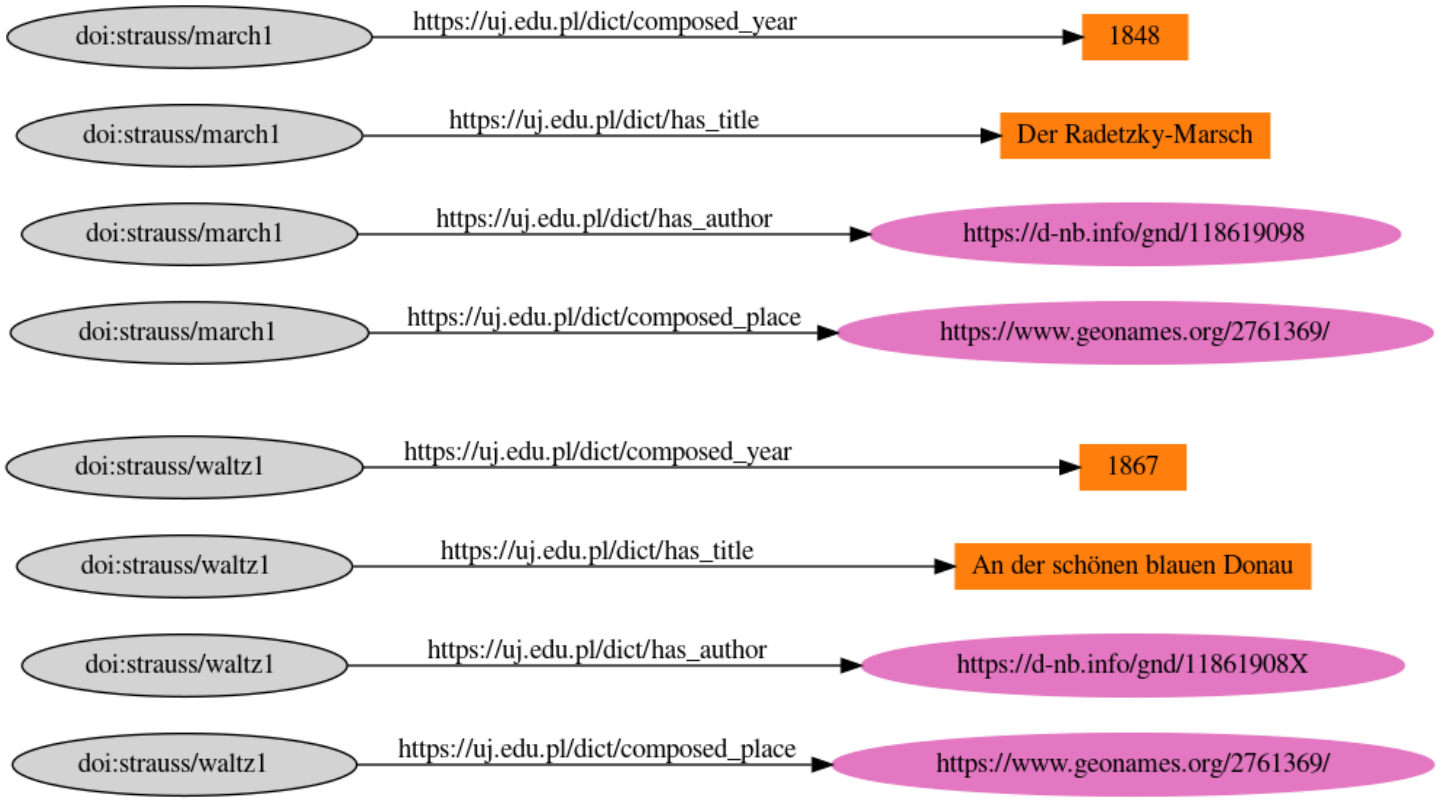
composed in = https://uj.edu.pl/dict/composed_place

has title = https://uj.edu.pl/dict/has_title

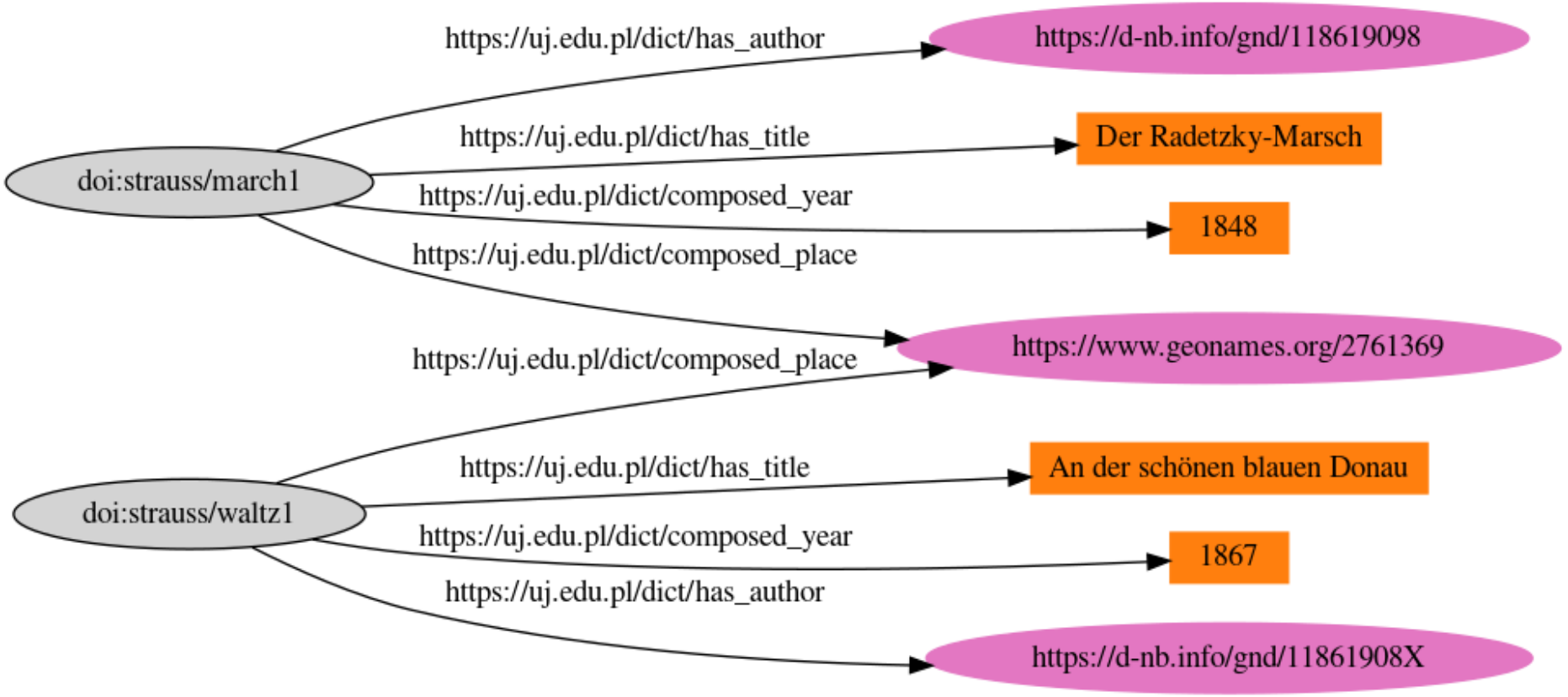
has author = https://uj.edu.pl/dict/has_author



FINAL STEP: MERGE THE SAME NODES INTO ONE



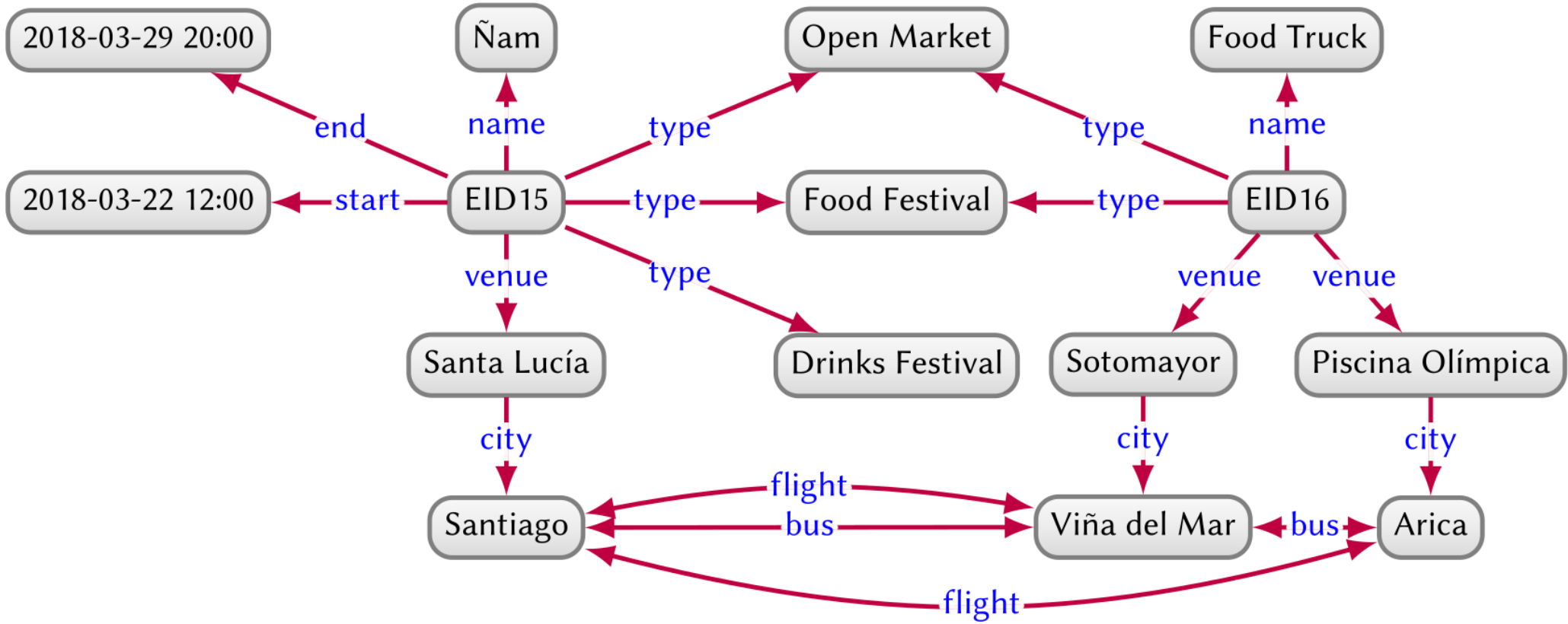
FINAL STEP: MERGE THE SAME NODES INTO ONE



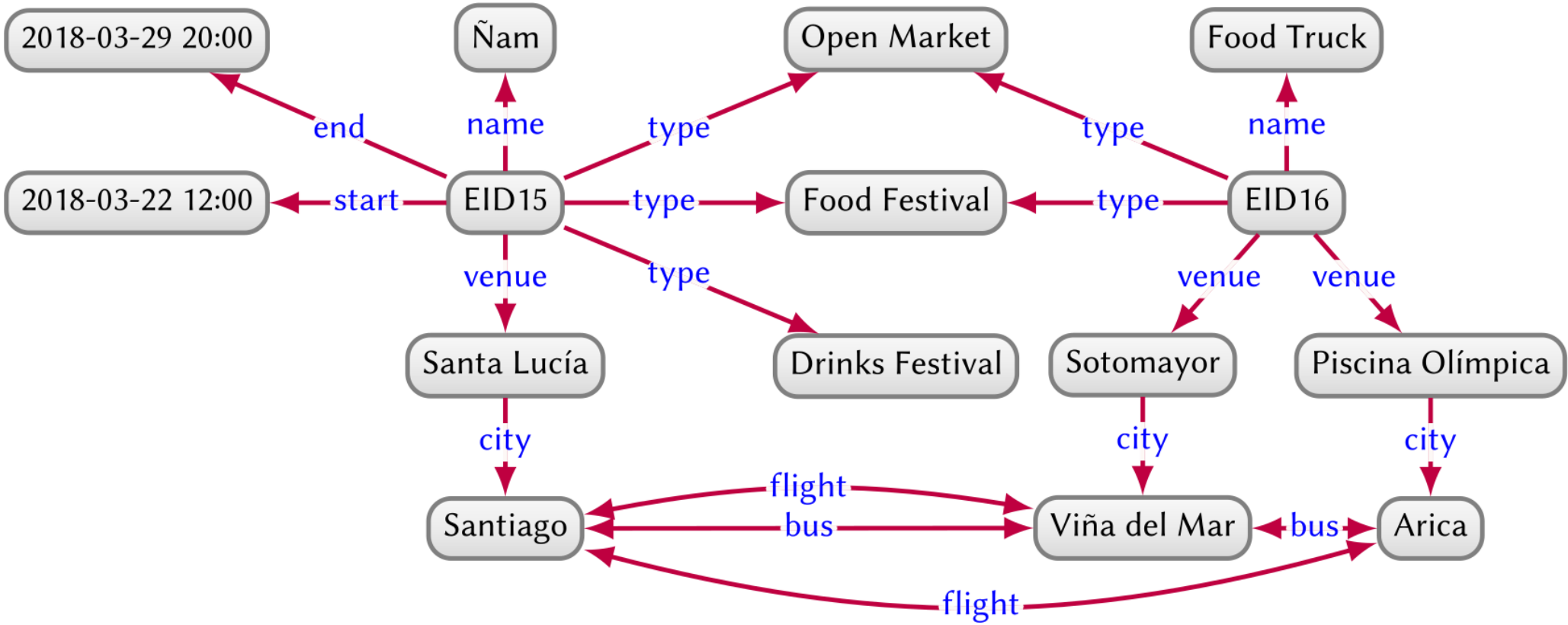
PATTERN MATCHING

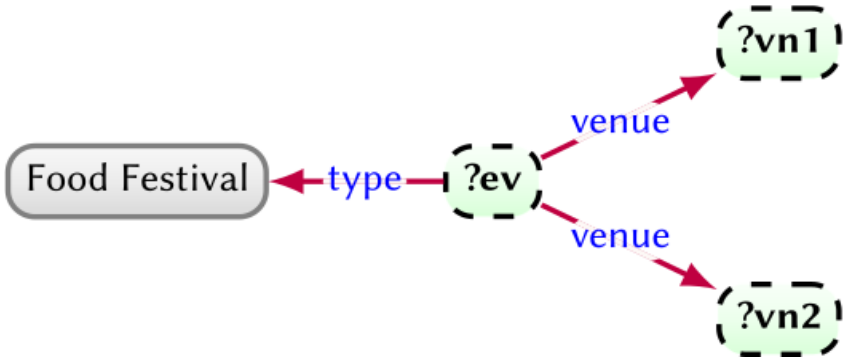
Patterns, patterns everywhere...

QUERY IS A PATTERN



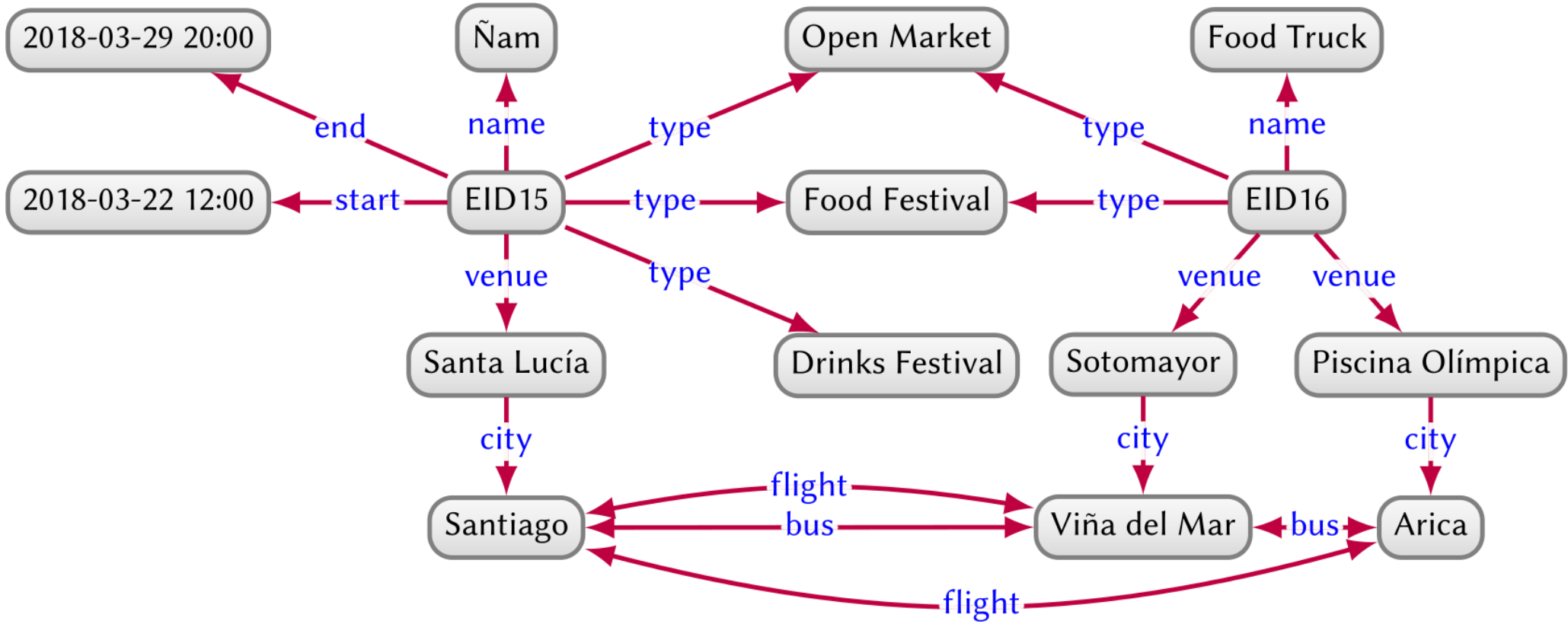
QUERY IS A PATTERN





?ev	?vn1	?vn2
EID16	Piscina Olímpica	Sotomayor
EID16	Sotomayor	Piscina Olímpica
EID16	Piscina Olímpica	Piscina Olímpica
EID16	Sotomayor	Sotomayor
EID15	Santa Lucía	Santa Lucía

QUERY IS A PATTERN



Q_1: ?event —type→ Food Festival

Q_2: ?event —type→ Drinks Festival

Q_3: ?event —venue→ ?ven —city→ Santiago

Q_4: ?event —start→ ?start

Q_5: ?event —name→ ?name

$$Q := ((((Q_1 \cup Q_2) \triangleright Q_3) \bowtie Q_4) \bowtie Q_5), \qquad Q(G) =$$

?event	?start	?name
EID16		Food Truck

SPARQL QUERY LANGUAGE

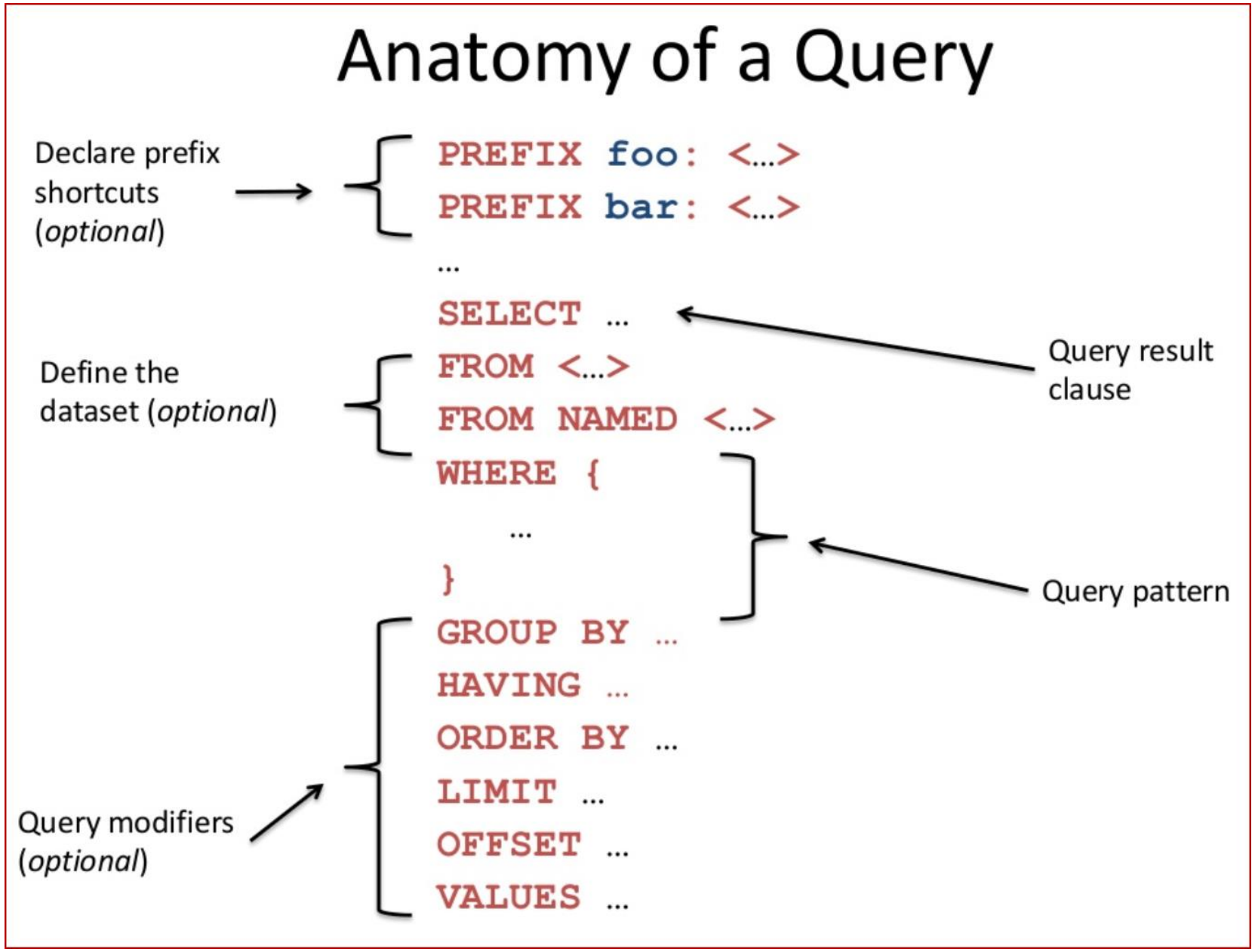


Image by Michael K. Bergman

QUERY FORMS

1. SELECT

Returns all, or a subset of, the variables bound in a query pattern match

2. CONSTRUCT

Returns an RDF graph specified by a set of triple templates

3. ASK

Returns a boolean indicating whether a query pattern matches or not

4. DESCRIBE

Returns an RDF graph that describes the resources found

5. UPDATE

Addition and removal of triples.
Graph management

VOCABULARIES AND ONTOLOGIES

How to structure the knowledge?

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

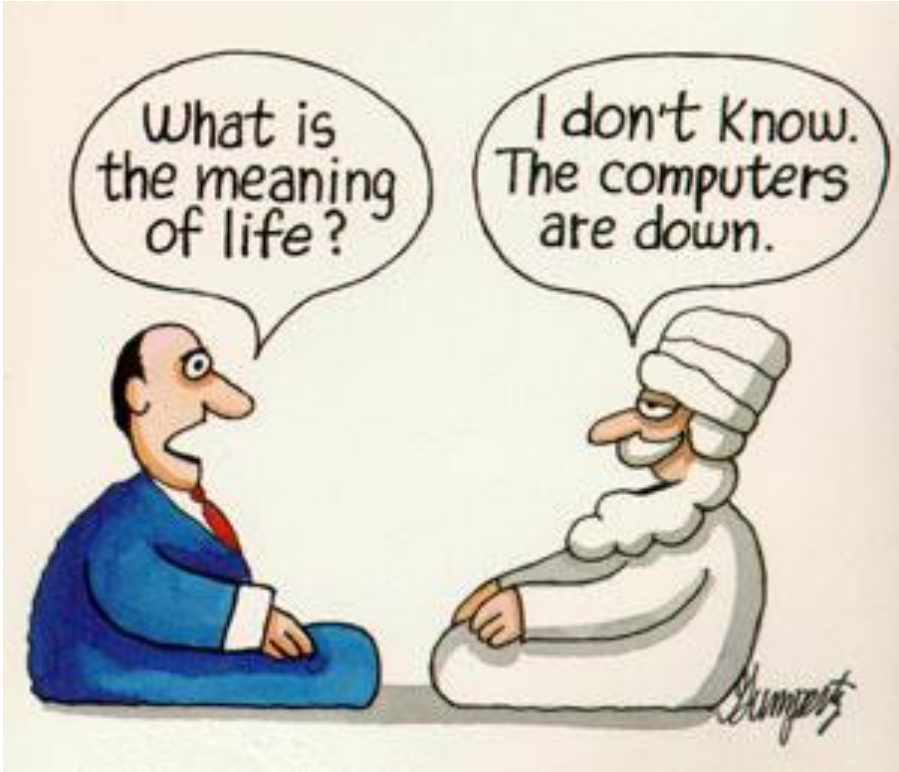
VOCABULARIES: SHARED MEANING OF WORDS (URIS)

- https://uj.edu.pl/dict/has_title
- https://uj.edu.pl/dict/Manuscript

vs

- https://schema.org/name
- https://schema.org/Manuscript

A vocabulary: a data model comprising classes, properties and relationships (URIs) which can be used for describing your data and metadata.



Manuscript

A Schema.org Type

This term is in the "new" area - implementation feedback and adoption from applications and websites can help improve our definitions.

Thing > CreativeWork > Manuscript

[more...]

A book, document, or piece of music written by hand rather than typed or printed.

Property	Expected Type	Description
Properties from CreativeWork		
about	Thing	The subject matter of the content. Inverse property: subjectOf
abstract	Text	An abstract is a short description that summarizes a CreativeWork.
accessMode	Text	The human sensory perceptual system or cognitive faculty through which a person may process or perceive information. Values should be drawn from the approved vocabulary.
accessModeSufficient	ItemList	A list of single or combined accessModes that are sufficient to understand all the intellectual content of a resource. Values should be drawn from the approved vocabulary.
accessibilityAPI	Text	Indicates that the resource is compatible with the referenced accessibility API. Values should be drawn from the approved vocabulary.
accessibilityControl	Text	Identifies input methods that are sufficient to fully control the described resource. Values should be drawn from the approved vocabulary.
accessibilityFeature	Text	Content features of the resource, such as accessible media, alternatives and supported enhancements for accessibility. Values should be drawn from the approved vocabulary.
accessibilityHazard	Text	A characteristic of the described resource that is physiologically dangerous to some users. Related to WCAG 2.0 guideline 2.3. Values should be drawn from the approved vocabulary.
accessibilitySummary	Text	A human-readable summary of specific accessibility features or deficiencies, consistent with the other accessibility metadata but expressing subtleties such as "short descriptions are present but long descriptions will be needed for non-visual users" or "short descriptions are present and no long descriptions are needed".
accountablePerson	Person	Specifies the Person that is legally accountable for the CreativeWork.
acquireLicensePage	CreativeWork or URL	Indicates a page documenting how licenses can be purchased or otherwise acquired, for the current item.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
alternativeHeadline	Text	A secondary title of the CreativeWork.

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

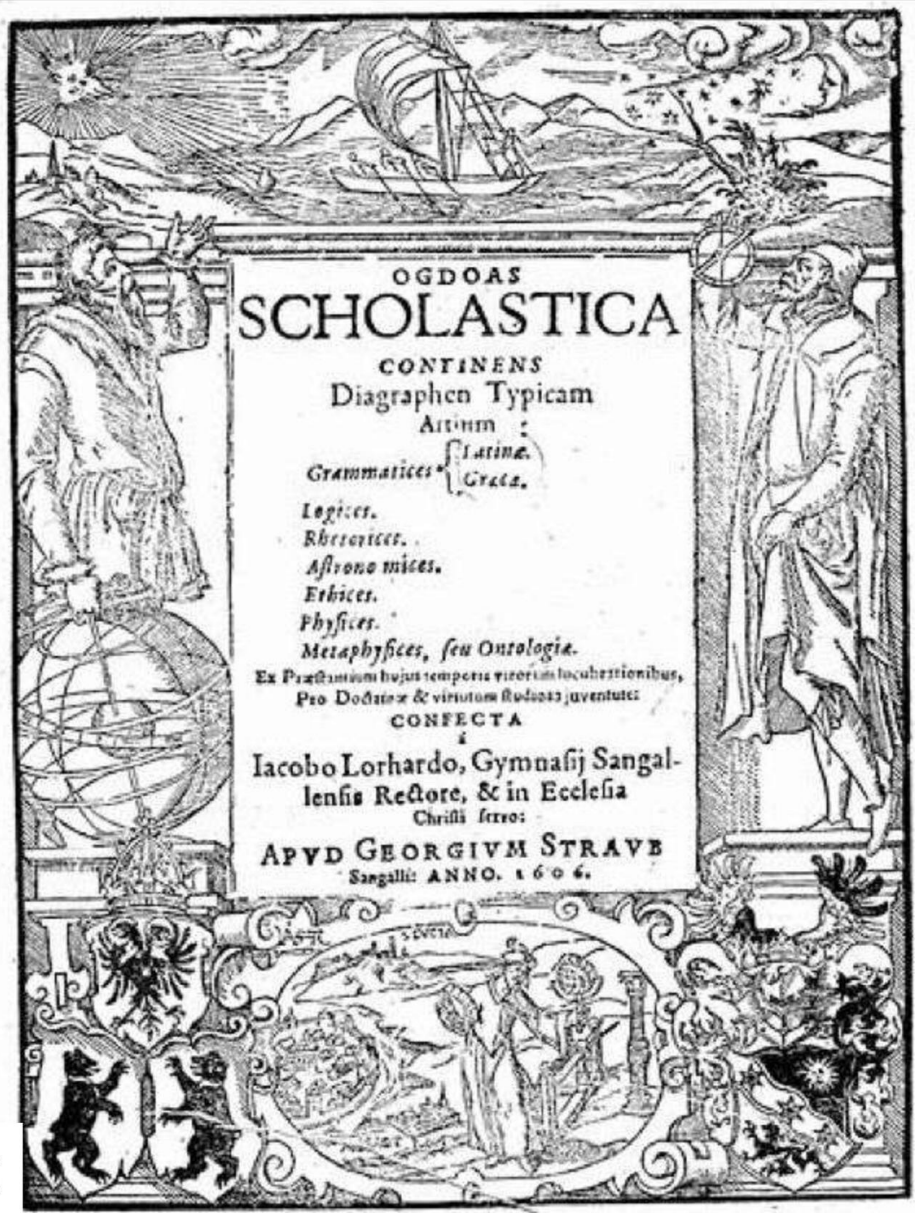
Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

What is Ontology?

- **Etymology:**
  - ον [greek] participle of “to be”
  - λογια [greek] science
- **(simplified) Definition:**

  “philosophical study of the nature of being, existence, or reality, as well as the basic categories of being and their relations....” (Wikipedia)
- **General Question:**

  “what does exist or can be said to exist?”
- The term ***ontology*** first turned up in 1606 in Jacob Lorhard’s *Ogdoas Scholastica*


OGDOAS
SCHOLASTICA
CONTINENS
Diagraphen Typicam
Artium :
Grammatices { Latinæ. Græcæ.
Logices.
Rhetoricæ.
Astronomices.
Ethices.
Physices.
Metaphysices, seu Ontologiæ.
Ex Præstantium hujus temporis virorum locubrationibus,
Pro Doctrinæ & virtutum studiosa juventute
CONFECTA
à
Iacobo Lorhardo, Gymnasij Sangallensis Rectore, & in Ecclesia Christi ferro:
APVD GEORGIVM STRAVB
Sangalli: ANNO. 1606.

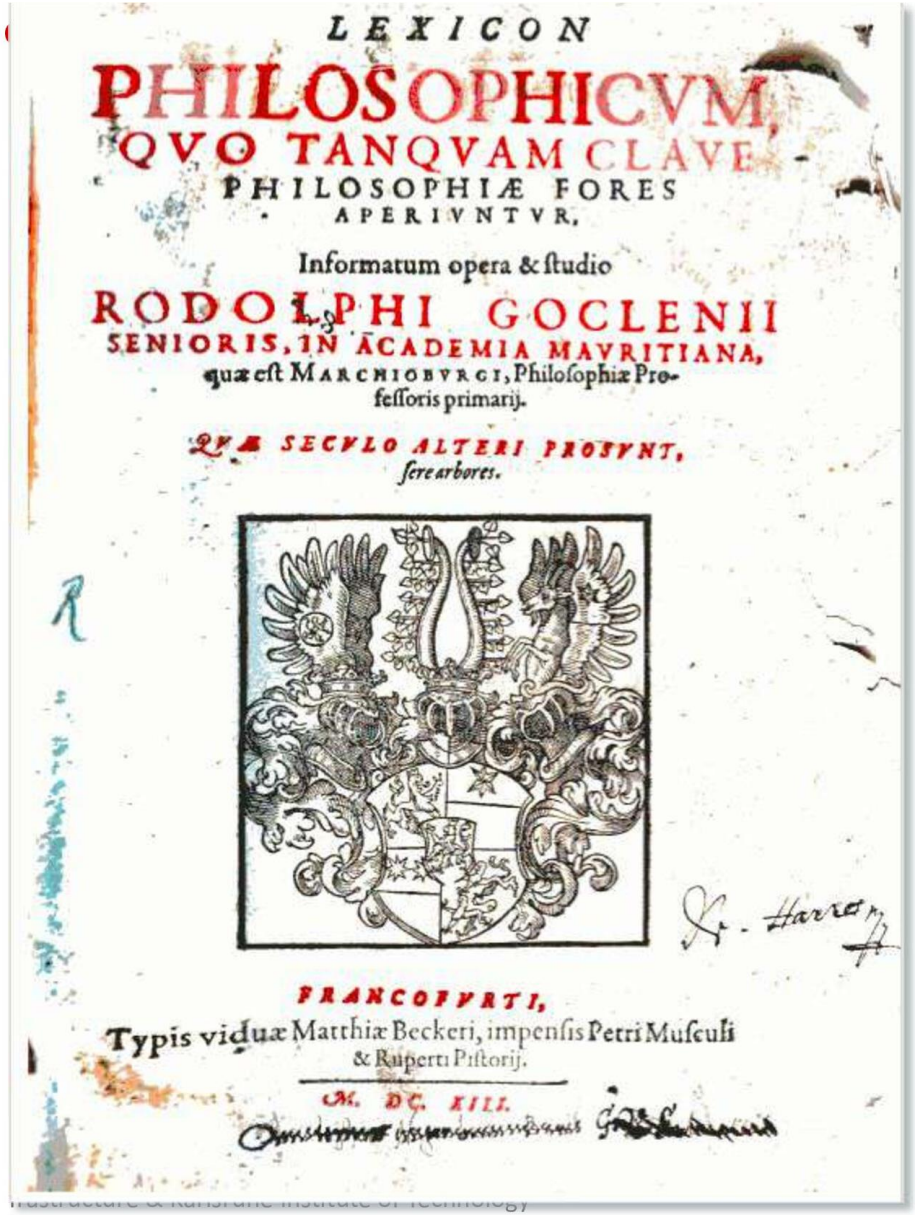
Jacob Lorhard: Ogdoas Scholastica, continens Diagraphen Typicam artium: Grammatices (Latinae, Graecae), Logices, Rhetorices, Astronomices, Ethices, Physices, Metaphysices, seu Ontologiae. Sangalli: Straub, 1606.

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

Fundamental Questions of Ontology

1. **What does it mean for a being to be?**
   - When are two things identical?
   - Is everything that exists also real?
   - Does something exist, if it is only possible?
   - Are there non-existing things?

2. **What categories of objects do exist?**
   - Do things exist that are only unique or only multiple (Universalia)?
   - Do things exist that are unilaterally dependent on others (Substances)?
   - Of which sort is this dependency (Causality)?
   - Do necessary properties exist (Essences)?
   - How do composed things relate to their components?


LEXICON
PHILOSOPHICVM,
QVO TANQVAM CLAVE
PHILOSOPHIÆ FORES
APERIVNTVR,
Informatum opera & ſtudio
RODOLPHI GOCLENII
SENIORIS, IN ACADEMIA MAVRITIANA,
quæ eſt MARCHIOBVRGI, Philoſophiæ Profeſſoris primarij.
QVÆ SECVLO ALTERI PROSVNT,
ſere arbores.
FRANCOFVRTI,
Typis viduæ Matthiæ Beckeri, impenſis Petri Muſculi
& Ruperti Piſtorij.
M. DC. XIII.

Rudolph Goclenius: Lexicon philosophicum, quo tantam clave philosophiae fores aperiuntur, 1613

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

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

[2]

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

How to represent Ontologies

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

*„A **philosopher** is a person who practices or investigates philosophy. The term philosopher comes from the Ancient Greek meaning 'lover of wisdom'. The coining of the term has been attributed to the Greek thinker Pythagoras (6th century BCE)“.*

informal description

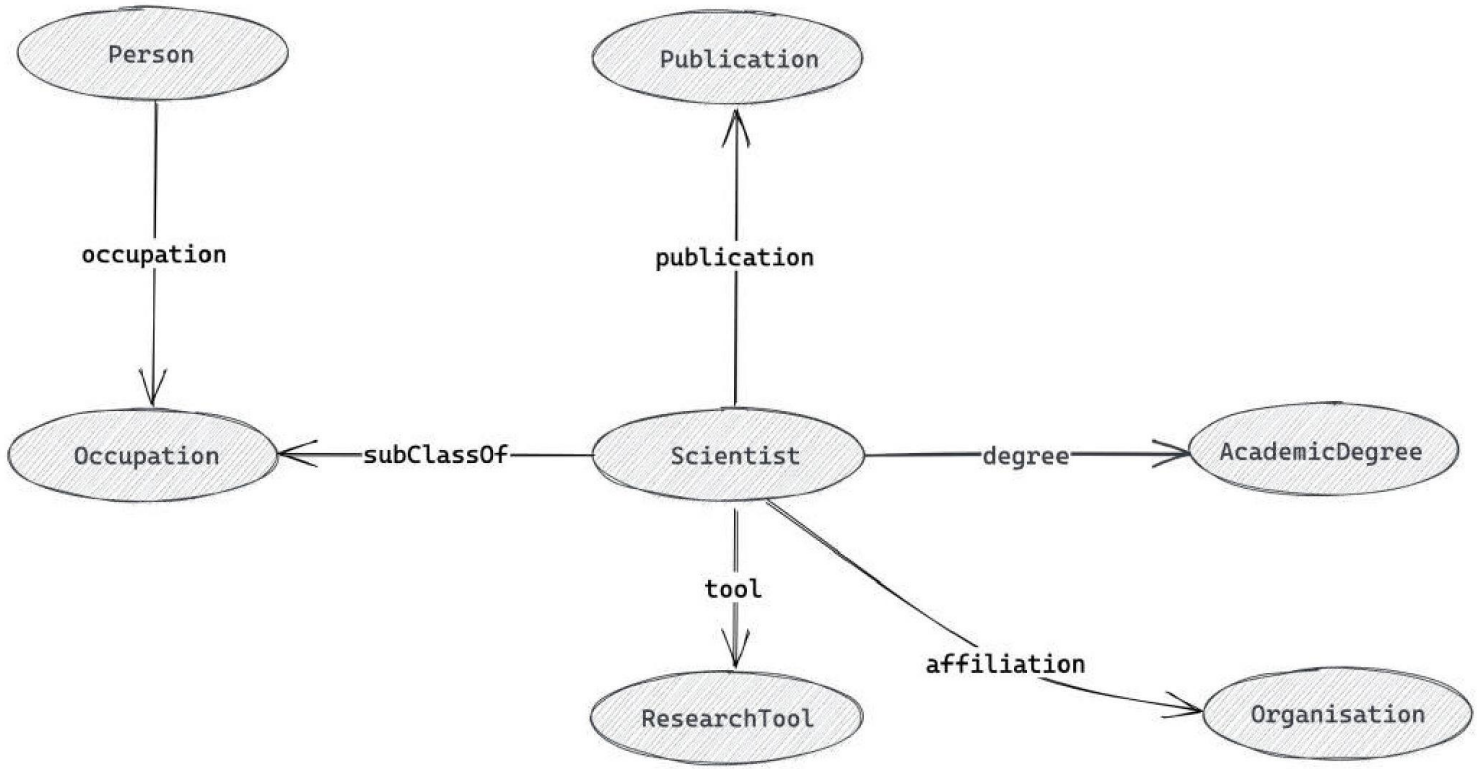
Philosopher

- *first name <string>*
- *surname <string>*
- *address <string>*
- *number of publications <string>*
- *impact factor <float>*
- *…*

semi-formal description

How to represent Ontologies

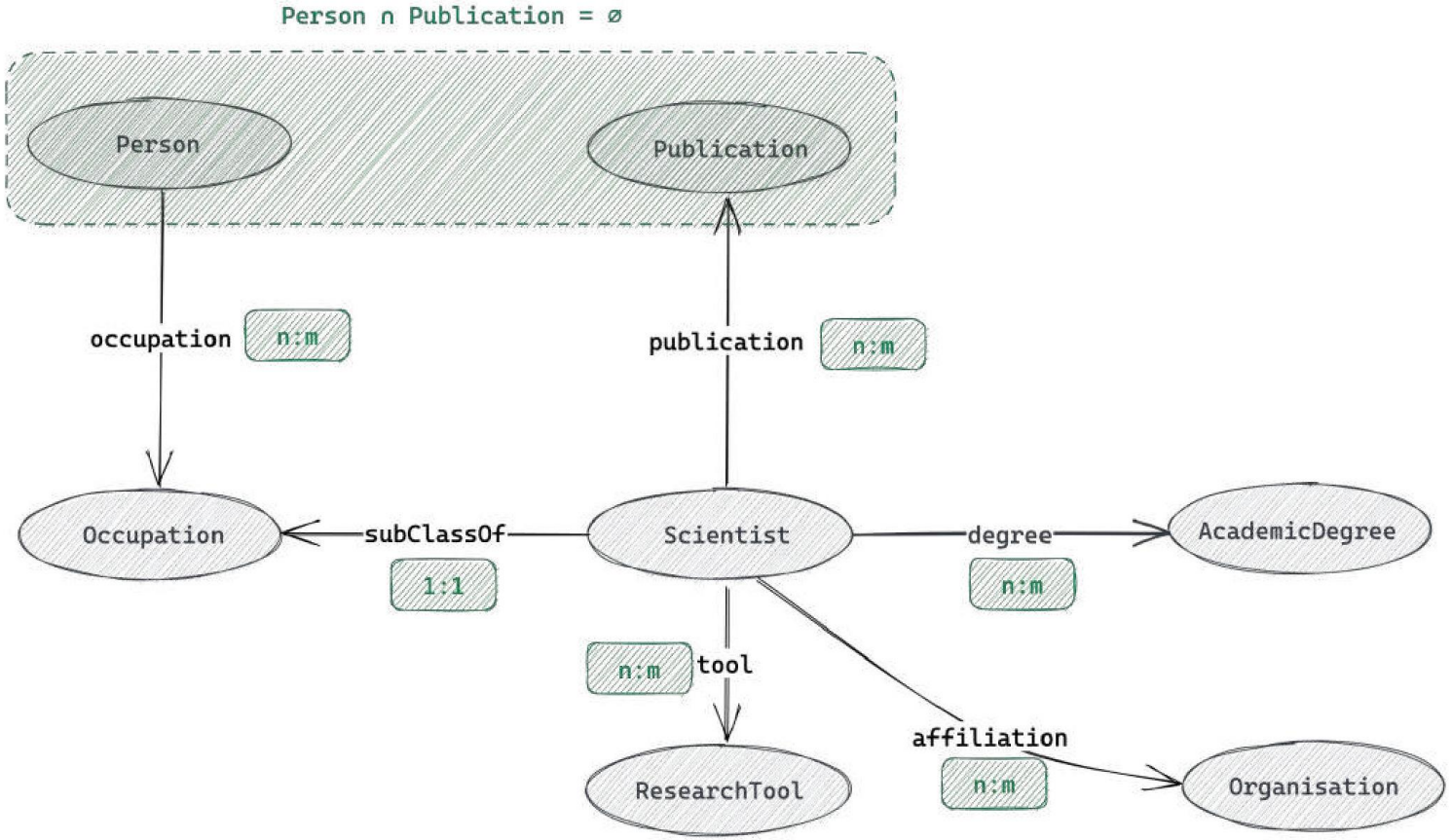
- Classes can be **related** to other classes
- **Relations** are special attributes, whose values are objects of (other) classes



Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

How to represent Ontologies

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



Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

How to represent Ontologies

- Classes, relations, and constraints can be combined to form (complex) **Statements** / **Assertions.**
- Special Case: **formal Axioms**

Example:

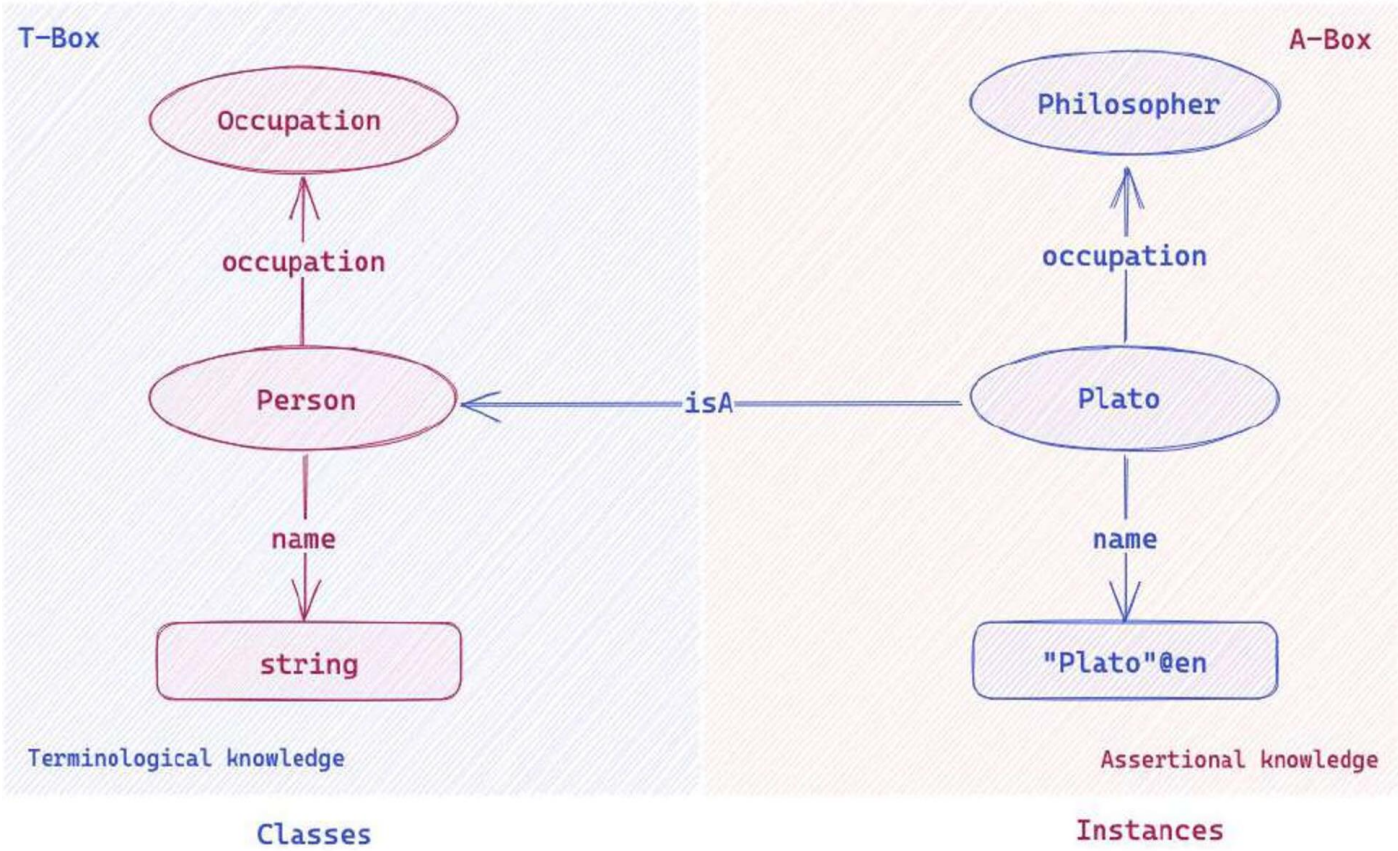
„A philosopher is somebody who knows himself.“

- **Axioms** describe knowledge that cannot be expressed simply with the help of other existing components.

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

How to represent Ontologies

Instances describe individuals of an ontology.



Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

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

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

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)

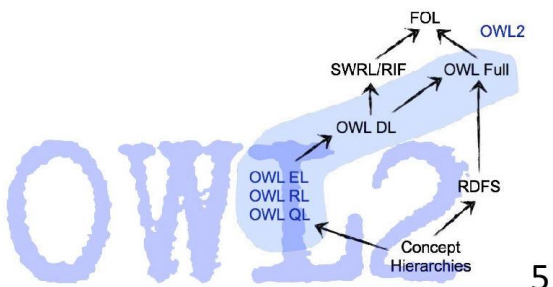
- Inclusion $C \sqsubseteq D$
- Equivalence $C \equiv D$

Rbox (Property Axioms)

- Inclusion $R_1 \sqsubseteq R_2$
- General Inclusion $R^{(-)}_1 \circ R^{(-)}_2 \circ \ldots \circ R^{(-)}_n \sqsubseteq R$
- 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 $a \neq b$



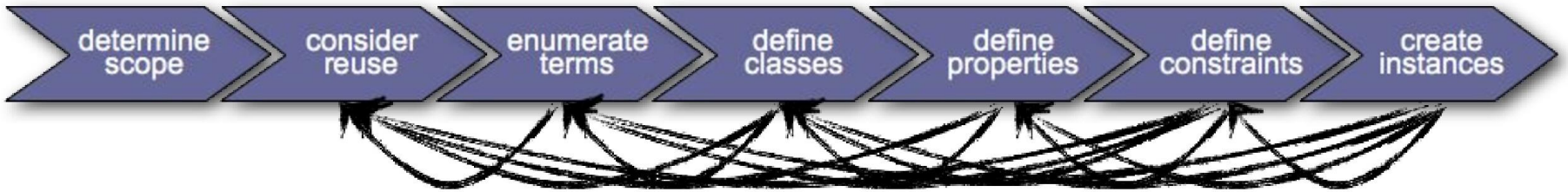
Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

ONTOLOGY 101

We want our own ontology!

Ontology Development 101

(Noy, McGuinness, 2000)

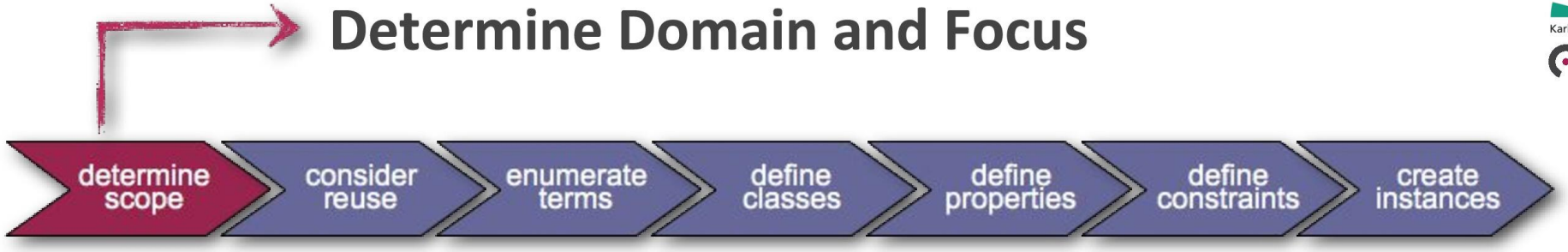


- In practice an **iterative Process** that **repeats continuously** and improves the ontology
- There are always **different approaches** for modelling an ontology
- In practice the designated application decides about the modelling approach

„There is no one correct way to model a domain.
There are always viable alternatives."

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Determine Domain and Focus



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

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Determine Domain and Focus

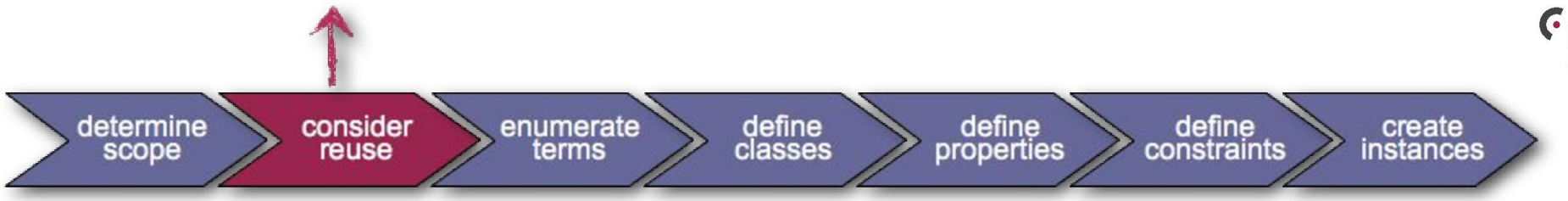
Competence Questions *(Example: Wine Ontology)*

- *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 for grilled vegetables?*
- *Which properties of a wine do influence whether it matches with a specific dish?*
- *Does the bouquet of a wine ch*
- *Does th*
- *...*

These Questions might change within the ontology life cycle

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Consider Reuse



- Why should we consider reuse?
  - In order to save **cost**
  - In order to apply **tools** that are applied for other existing ontologies also for 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!

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Develop a Terminology

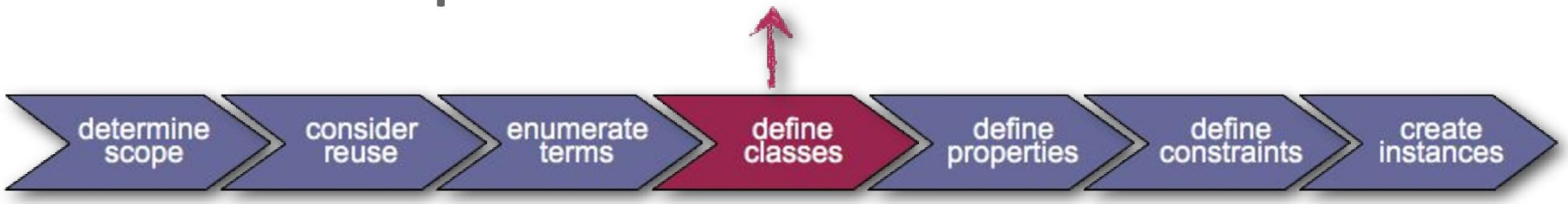
- About which **concepts** are we talking?
- Which **properties** have these concepts?
- **What** do we want **to say** about these concepts?

Example: ***Wine Ontology***

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

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

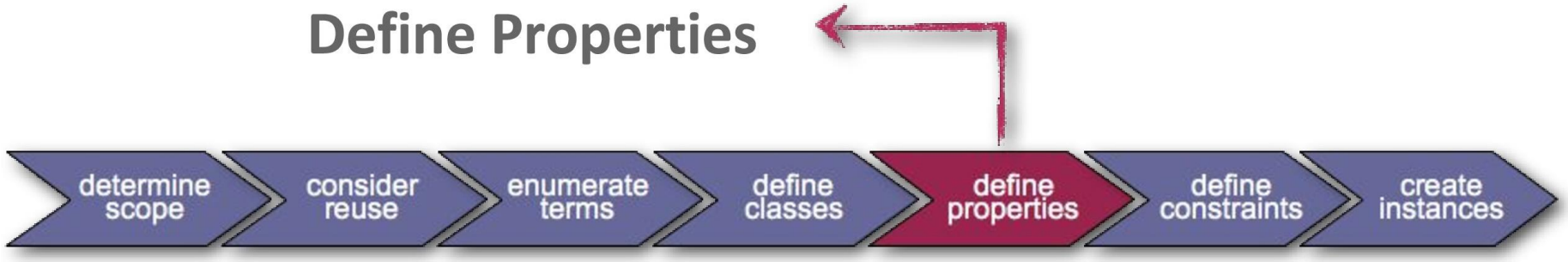
Develop Classes and Class Hierarchies



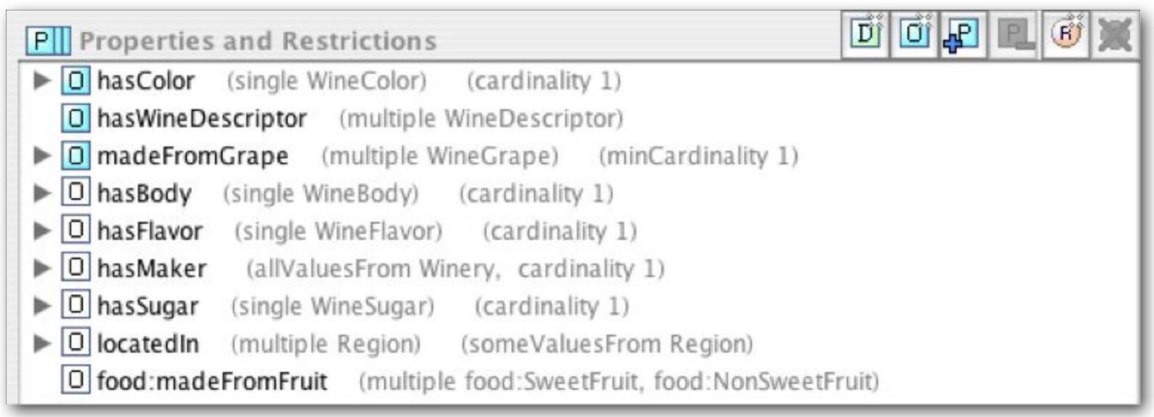
- **Classes** are concepts in the designated domain
  - *class of wines*
  - *class of wineries*
  - *class of red wines*
  - ...
- Classes are collections of objects with **similar properties.**
- Choose a **top-down / bottom-up / middle-out** approach to model class hierarchies.

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Define Properties



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



Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

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...*

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

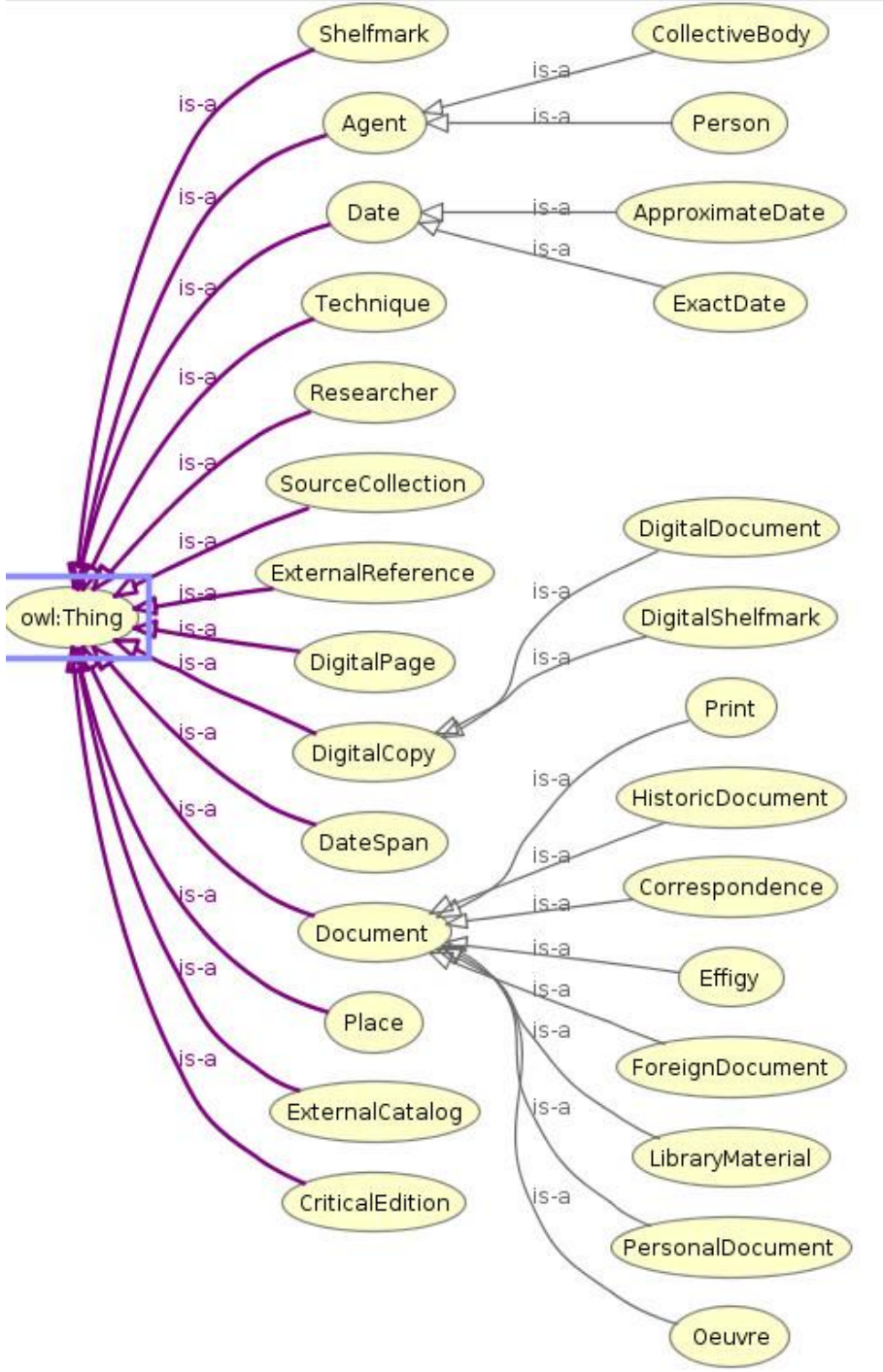
Definition of Instances

- 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. assignment of property values for the instances according to the given constraints

- *„The glass of red wine that I drank last supper...“*

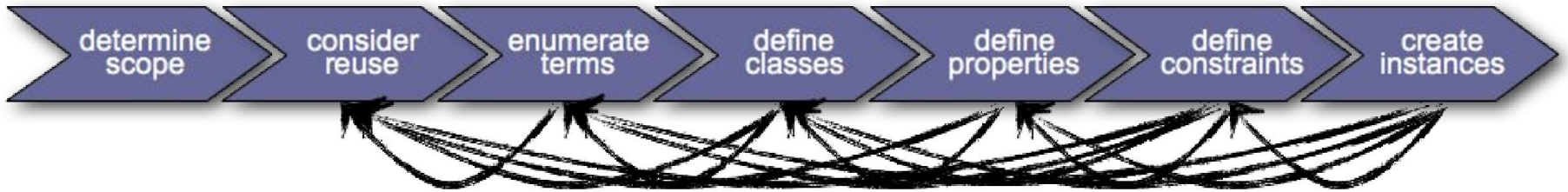
Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

THE GOOD, THE BAD, THE UGLY



Ontology Development 101

(Noy, McGuinness, 2000)



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

http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Slide from Knowledge Graphs course by prof. Harald Sack & Mehwish Alam (openHPI, 2020).

Table 13.1 Well-known ontologies

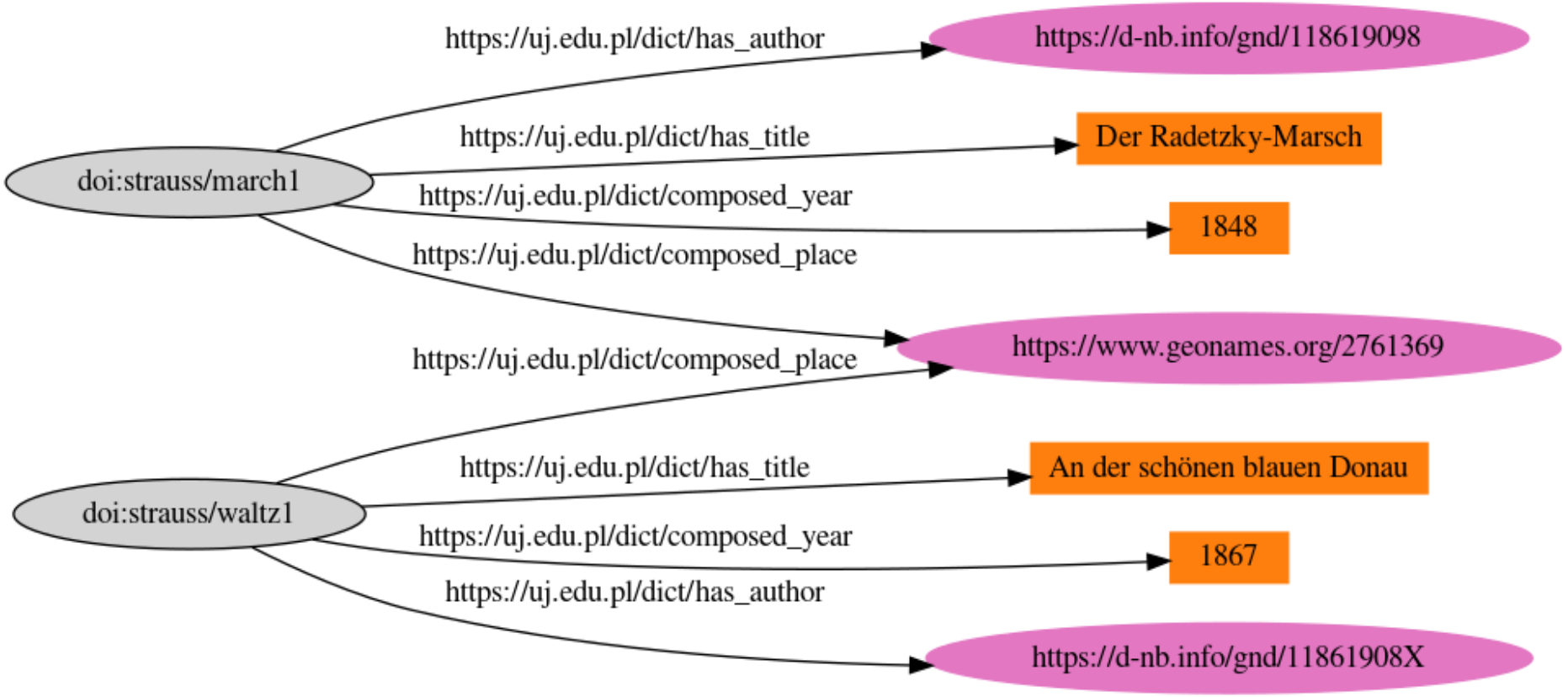
Ontology	Brief description
schema.org	Written in OWL, created by Google, Microsoft and Yahoo!. See Sect. 10.1
Dublin Core	Written in RDFS, offers terms for describing documents
FOAF	Written in OWL, offers terms for describing persons and their social network, and their relations to other related objects
Basic Geo	Written in RDFS, offers terms for representing latitude, longitude and altitude information about spatially-located objects
BIO	Written in RDFS, offers terms for describing biographical information about people, both living and dead. It is used often in FOAF documents
vCard RDF	Original vCard format's RDFS version
Creative Commons metadata	Semantically-Interlinked Online Communities Project, written in OWL, offers terms for interconnecting discussion methods such as blogs, forums and mailing lists
SIOC	Minimal means for managing RDF graph content directly via common HTTP operations
GoodRelations	Written in OWL, offers terms for specifying offerings and other relevant aspects of e-commerce on the Web
DOAP	Description of a Project, written in RDFS, offers terms for describing software projects similar to using FOAF to describe people
Music Ontology	Written in OWL, offers terms for describing music, such as the artists, albums, tracks, performances and arrangements
Programmes Ontology	Written in OWL, offers terms for describing brands, series (seasons), episodes, broadcast events and broadcast services, etc.

Table from A Developer's Guide to the Semantic Web (2014) by Liyang Yu.

LINKED OPEN DATA

You can combine knowledge from many data sources!

ARE JOHANN STRAUSS (118619098) AND JOHANN STRAUSS (11861908X) RELATED?



ARE JOHANN STRAUSS (118619098) AND JOHANN STRAUSS (1186190 8X) RELATED?

Johann Strauss II

80 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

"Johann Strauss" redirects here. For other uses, see Johann Strauss (disambiguation).

Johann Baptist Strauss II (German: ['joːhan bap'tɪst 'ʃtʁaʊs]; 25 October 1825 – 3 June 1899), also known as **Johann Strauss Jr.**, **the Younger** or **the Son** (German: *Johann Strauß Sohn*), was an Austrian composer of light music, particularly dance music and operettas as well as a violinist. He composed over 500 waltzes, polkas, quadrilles, and other types of dance music, as well as several operettas and a ballet. In his lifetime, he was known as "The Waltz King", and was largely responsible for the popularity of the waltz in Vienna during the 19th century. Some of Johann Strauss's most famous works include "The Blue Danube", "Kaiser-Walzer" (Emperor Waltz), "Tales from the Vienna Woods", "Frühlingsstimmen", and the "Tritsch-Tratsch-Polka". Among his operettas, *Die Fledermaus* and *Der Zigeunerbaron* are the best known.

Strauss was the son of Johann Strauss I and his first wife Maria Anna Streim. Two younger brothers, Josef and Eduard Strauss, also became composers of light music, although they were never as well known as their brother.

Johann Strauss II

ARE JOHANN STRAUSS (118619098) AND JOHANN STRAUSS (11861908X) RELATED?

The Web of Data

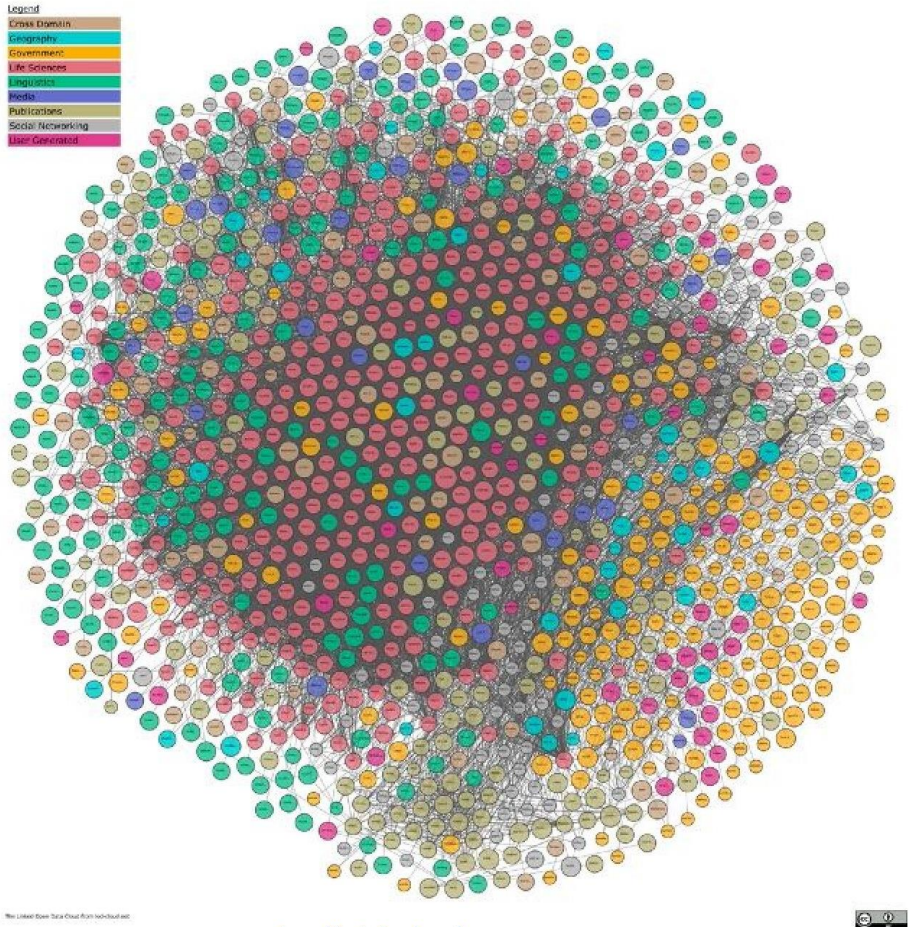
Linked Data

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

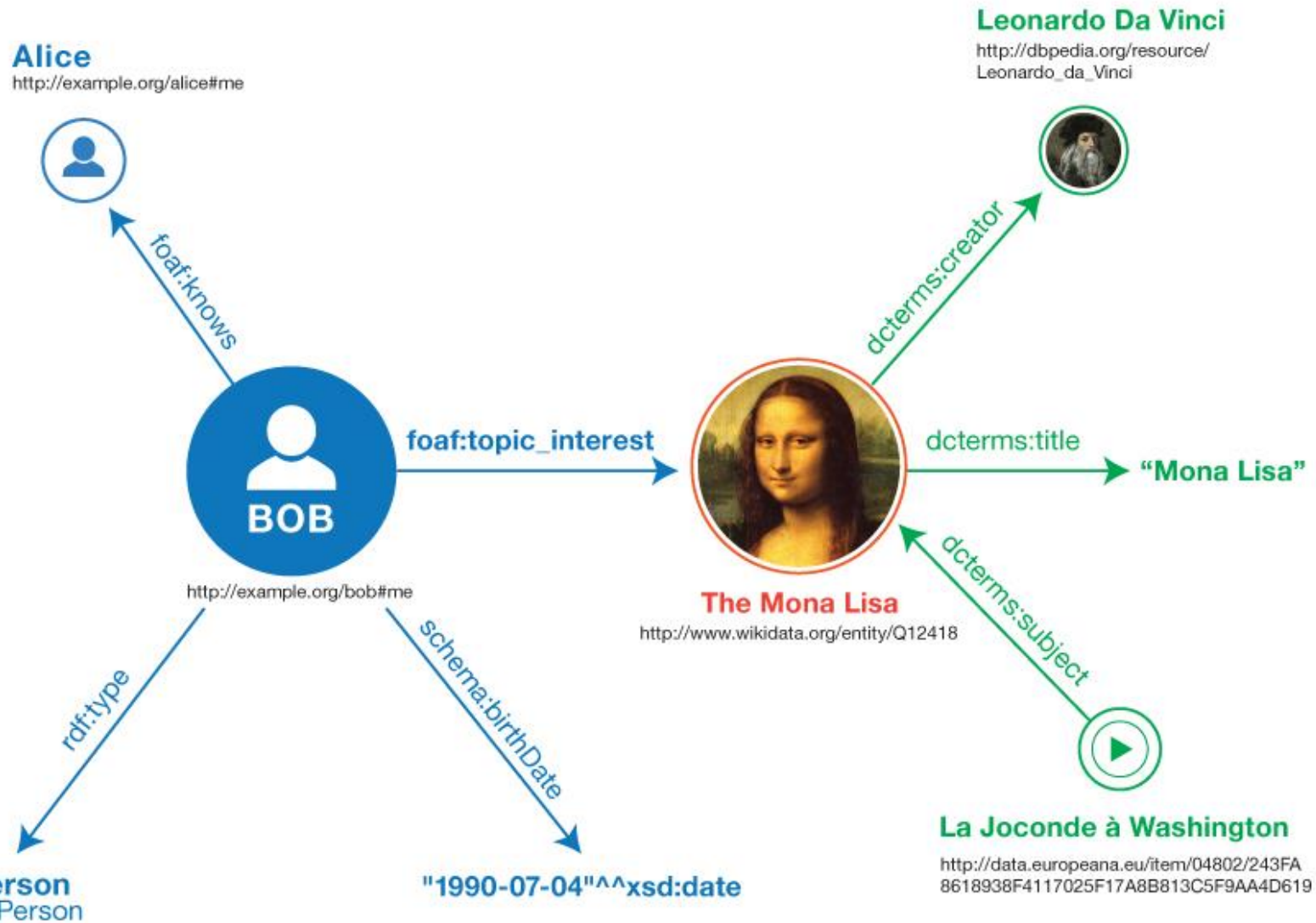


http://lod-cloud.net/

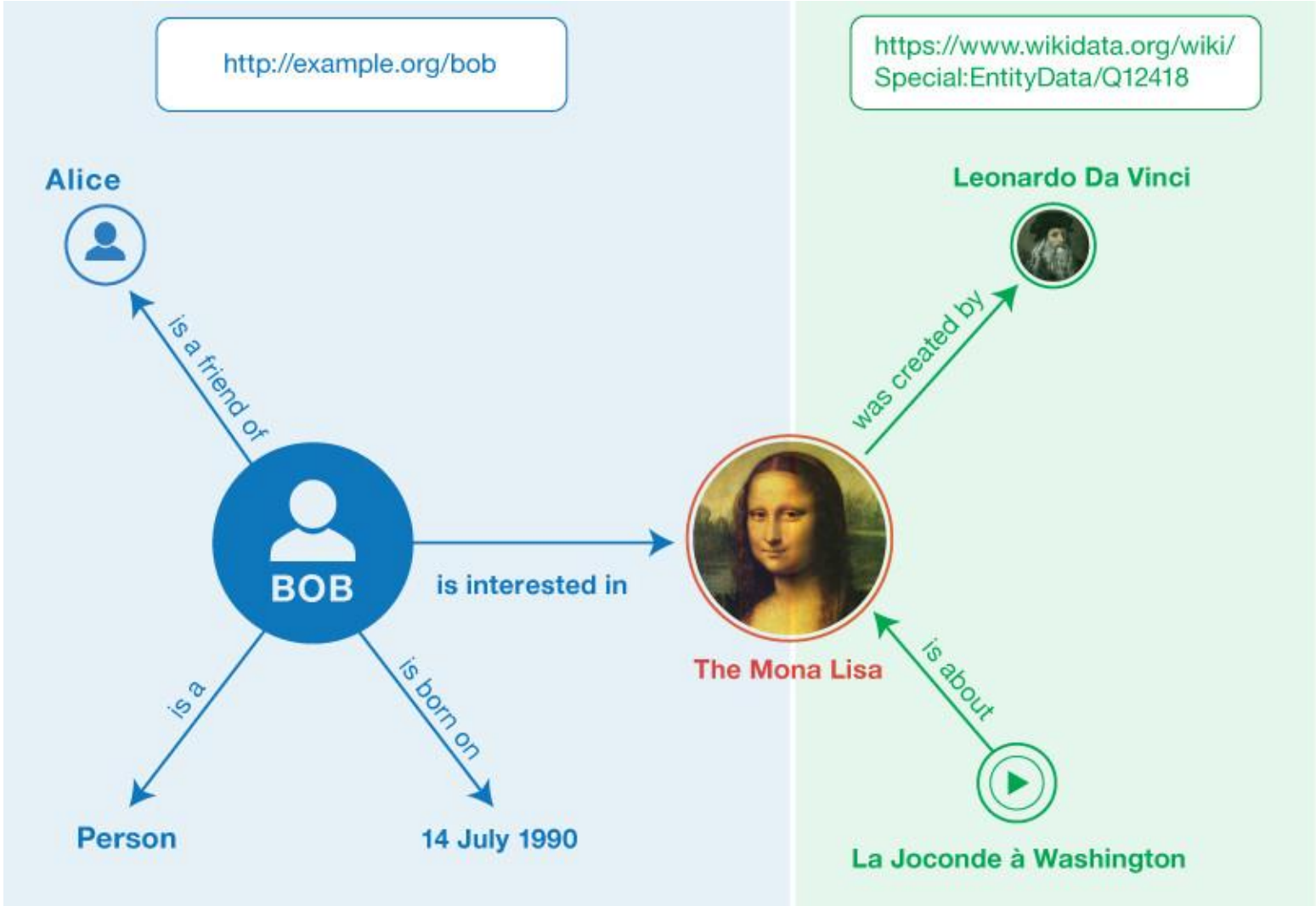
[2]

Slide from Knowledge Graphs by prof. Harald Sack (openHPI, 2023).

ONCE UPON A TIME, THERE WAS A GRAPH...



...STORED ON MANY SERVERS



KNOWLEDGE GRAPHS RECAP

- Formal knowledge representation
- Flexible metadata structure
- Pattern matching-based query engines
- Rich modeling capabilities with ontologies
- Linked Open Data network connecting various sources of knowledge into one big knowledge graph

STAY TUNED!

Knowledge Graphs 201:

- Real-life knowledge graphs
- Property graphs 101
- Graphs for ML (embeddings!)
- Semantic search
- Exploration over knowledge graphs
- GraphRAG

THANK YOU FOR YOUR ATTENTION!

GEIST Research Group: https://geist.re/

Krzysztof Kutt: https://krzysztof.kutt.pl/

KEEP CALM AND ASK QUESTIONS!

keep-calm.net