



# metaphacts

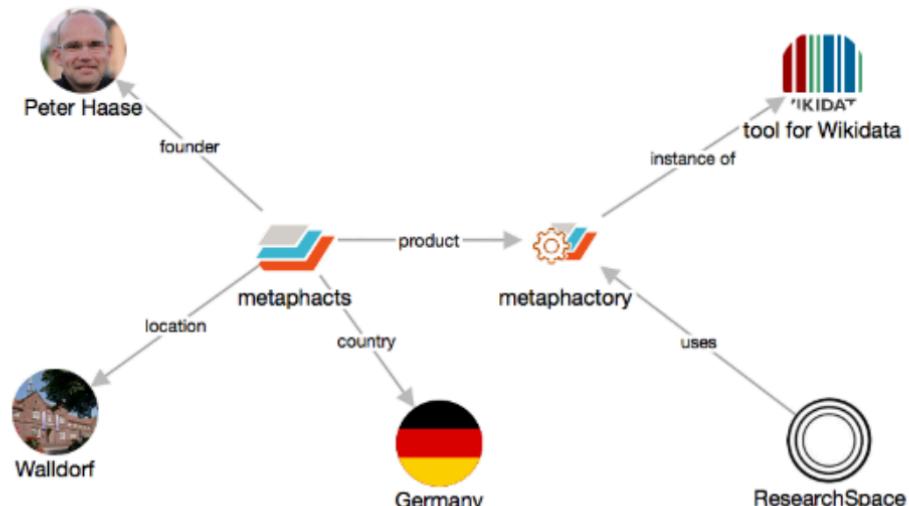
Smart Data. Smart Apps. Smart Decisions.

## Getting Started with Knowledge Graphs

Smart Data Conference  
January 30, 2017, San Francisco Bay  
Peter Haase

## Peter Haase

- Interest and experience in ontologies, semantic technologies and Linked Data
- PhD in KR and semantic technologies
- 15 years in academic research and software development
- Contributor to OWL 2 standard



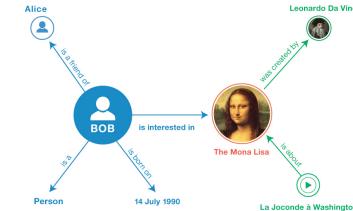
## metaphacts Company Facts

- Founded in Q4 2014
- Headquartered in Walldorf, Germany
- Currently ~10 people
- Platform for knowledge graph interaction & application development

## Introduction: What are Knowledge Graphs?

## Examples and Applications

- Wikidata
- Cultural Heritage
- Industrial Applications



## Standards and Principles

## metaphactory Knowledge Graph Platform

## Hands-on Exercises





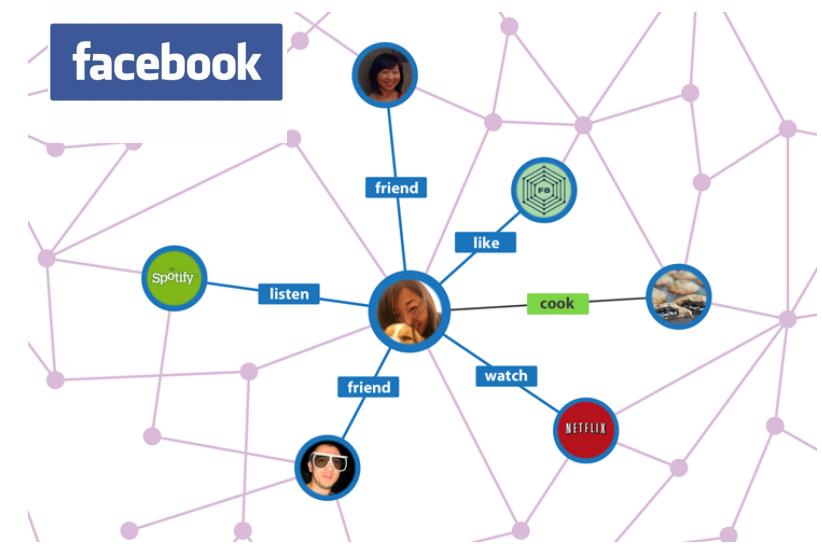
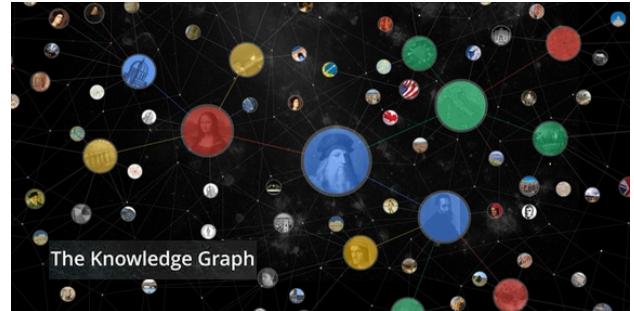
# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## Introduction

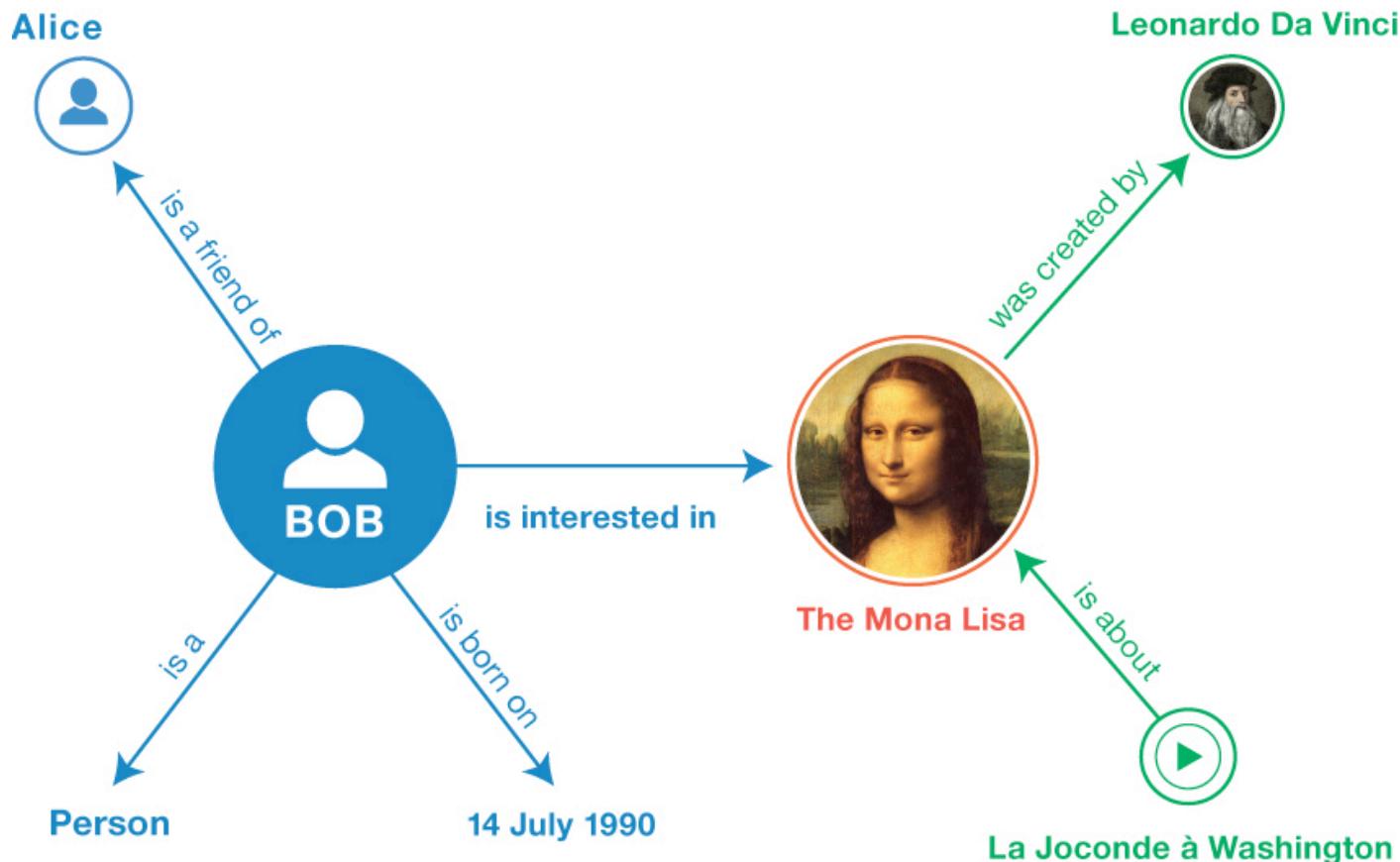
What are Knowledge Graphs?

# The Rise of Knowledge Graphs



- We need a structured and formal representation of knowledge
- We are surrounded by entities, which are connected by relations
- Graphs are a natural way to represent entities and their relationships
- Graphs can be managed efficiently

# A (very small) Knowledge Graph



<http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/example-graph.jpg>

- **Semantic descriptions** of **entities** and their **relationships**
- Uses a **knowledge representation formalism**  
(Focus here: RDF, RDF-Schema, OWL)
- **Entities**: real world objects (things, places, people) and abstract concepts (genres, religions, professions)
- **Relationships**: graph-based data model where relationships are first-class
- **Semantic descriptions**: types and properties with a well-defined meaning (e.g. through an ontology)
- Possibly axiomatic knowledge (e.g. rules) to support automated reasoning

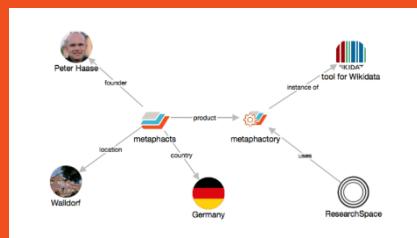
## Applications

- Semantic Search
- Question Answering
- Analytics
- Dashboards
- Knowledge Sharing
- Knowledge Management

## Algorithms

- Inferencing
- Machine Learning
- Entity Recognition
- Disambiguation
- Text Understanding
- Recommendations

## Knowledge Graph



- Entities
- Relationships
- Semantic Descriptions

## Data Sources

Data Transformation, Integration  
Natural Language Processing



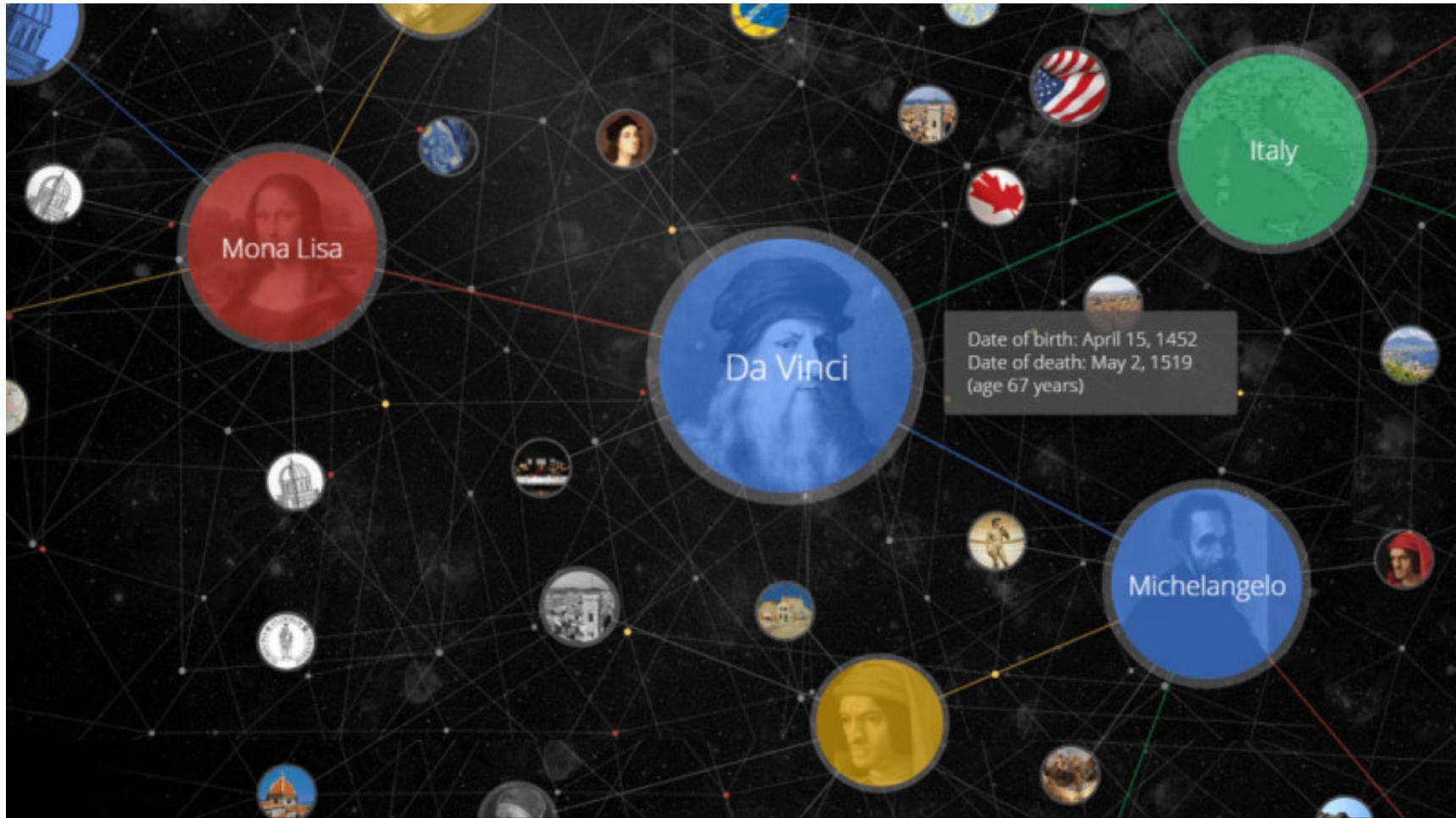


# metaphacts

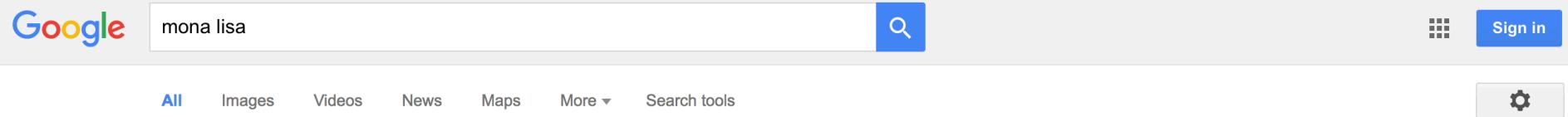
Smart Data. Smart Apps. Smart Decisions.

## Examples and Applications

## Google Knowledge Graph



## Entity Search and Summarizations



A screenshot of a Google search results page for the query "mona lisa". The search bar at the top contains "mona lisa". Below it, a navigation bar offers options: All (selected), Images, Videos, News, Maps, More ▾, and Search tools. To the right is a settings gear icon and a "Sign in" button. The main content area shows search results. At the top left of the results is the Google logo.

About 21,500,000 results (1.11 seconds)

### Mona Lisa - Wikipedia

[https://en.wikipedia.org/wiki/Mona\\_Lisa](https://en.wikipedia.org/wiki/Mona_Lisa) ▾

The Mona Lisa is a half-length portrait of a woman by the Italian Renaissance artist Leonardo da Vinci, which has been acclaimed as "the best known, the most ...".  
Louvre · Mona Lisa · Mona Lisa (disambiguation) · The Theft of the Mona Lisa

### Mona Lisa – Portrait of Lisa Gherardini, wife of Francesco del ... - Louvre

[www.louvre.fr/en/.../mona-lisa-portrait-lisa-gherardini-wife-francesco-del-giocondo](http://www.louvre.fr/en/.../mona-lisa-portrait-lisa-gherardini-wife-francesco-del-giocondo) ▾

The history of the Mona Lisa is shrouded in mystery. Among the aspects which remain unclear are the exact identity of the sitter, who commissioned the portrait, ...

### Mona Lisa | painting by Leonardo da Vinci | Britannica.com

<https://www.britannica.com/topic/Mona-Lisa-painting> ▾

Aug 17, 2016 - Mona Lisa, oil painting on a poplar wood panel by the Italian painter, draftsman, sculptor, architect, and engineer Leonardo da Vinci, probably the world's most-famous painting. ... Mona Lisa, oil painting on a poplar wood panel by Leonardo da Vinci, .... A third suggestion was that ...

### Mona Lisa | Leonardo da Vinci | High Renaissance: Florence and ...

<https://www.khanacademy.org/.../celebrity-art-leonardo-s-mona-l...> ▾

Learn about the history and key elements of the Mona Lisa, one of the most famous paintings in the world.

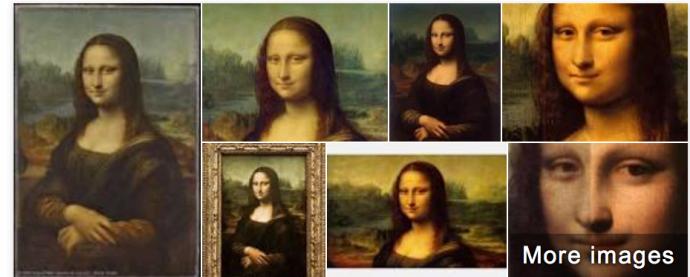


▶ 8:39

### Is the Mona Lisa Hiding Another Portrait? - artnet News

<https://news.artnet.com/art-world/leonardo-da-vinci-mona-lisa-portrait-hidden-388235> ▾

Dec 8, 2015 - According to French scientist Pascal Cotte, who has analyzed the painting for over 10 years, da Vinci's 'Mona Lisa' is hiding another portrait ...



## Mona Lisa

Work of art

The Mona Lisa is a half-length portrait of a woman by the Italian Renaissance artist Leonardo da Vinci, which has been acclaimed as "the best known, the most visited, the most written about, the most ...".  
[Wikipedia](#)

**Artist:** [Leonardo da Vinci](#)

**Dimensions:** 77 cm x 53 cm

**Location:** [The Louvre](#) (since 1797)

**Created:** 1503

**Media:** Oil paint

**Periods:** High Renaissance, Renaissance

## Discovering Related Entities

Google leonardo da vinci paintings  Sign in

All Images Videos Books News More ▾ Search tools 

Leonardo da Vinci / Artwork Most popular first ▾



Mona Lisa  
1503      The Last Supper  
1498      The Virgin of the  
Rocks  
1485      Vitruvian Man  
Annunciation  
1472      Lady with an  
Ermine  
1490      Adoration of the  
Magi  
1481      St. John the  
Baptist  
1513      The Virgin and  
Child with St. Anne  
1505      The Battle of  
Anghiari

**List of works by Leonardo da Vinci - Wikipedia**  
[https://en.wikipedia.org/wiki/List\\_of\\_works\\_by\\_Leonardo\\_da\\_Vinci](https://en.wikipedia.org/wiki/List_of_works_by_Leonardo_da_Vinci) ▾

The Last Supper is Leonardo da Vinci's most famous work of religious art and his only surviving mural. Leonardo da Vinci (baptised Leonardo di ser Piero da Vinci) ( pronunciation (help·info)), (April 15, 1452 – May 2, 1519) was one of the leading artists of the High Renaissance.

[Major extant works](#) · [Disputed attributions](#) · [Lost works](#) · [Some recent attributions](#)

**Leonardo da Vinci - Wikipedia**  
[https://en.wikipedia.org/wiki/Leonardo\\_da\\_Vinci](https://en.wikipedia.org/wiki/Leonardo_da_Vinci) ▾

Jump to [Painting](#) - Leonardo was, and is, renowned primarily as a painter. Among his works, the [Mona Lisa](#) is the most famous and most parodied portrait and [The Last Supper](#) the most reproduced religious painting of all time.

**Leonardo Da Vinci - The complete works**  
[www.leonardoda-vinci.org/](http://www.leonardoda-vinci.org/) ▾

Leonardo Da Vinci - Homepage. The complete [works](#), large resolution images, ecard, rating, slideshow and more! One of the largest [Leonardo Da Vinci](#) ...

[Biography](#) · [Self Portrait](#) · [The Works](#) · [The Last Supper 1498](#)



**Leonardo da Vinci**

Mathematician

Leonardo di ser Piero da Vinci, more commonly Leonardo da Vinci or simply Leonardo, was an Italian polymath whose areas of interest included invention, painting, sculpting, architecture, science, music, ...

[Wikinews](#)

## Factual Answers

when was leonardo da vinci born

All Images News Videos Shopping More ▾ Search tools

About 2,960,000 results (0.97 seconds)

Leonardo da Vinci / Born

April 15, 1452, Anchiano

Michelangelo March 6, 1475 Leonardo DiCaprio November 11, 1974 William Shakespeare April 1564

Feedback

People also ask

How old was Leonardo da Vinci when he died?

What did Leonardo da Vinci do?

Why is Leonardo da Vinci well known?

What things did Leonardo da Vinci study?

Leonardo da Vinci

Mathematician

Leonardo di ser Piero da Vinci, more commonly Leonardo da Vinci or simply Leonardo, was an Italian polymath whose areas of interest included invention, painting, sculpting, architecture, science, music, ... [Wikipedia](#)

**Born:** April 15, 1452, [Anchiano](#)

**Died:** May 2, 1519, [Clos Lucé, Amboise, France](#)

**Buried:** [Chapel of Saint-Hubert](#)

**Siblings:** [Giovanni Ser Piero](#), [Guglielmo Ser Piero](#), more

**Quotes**

*Simplicity is the ultimate sophistication.*

*As a well-spent day brings happy sleep, so a life well spent brings happy death.*

*Learning never exhausts the mind.*

**Artwork**

[View 15+ more](#)





Get Started

Introduction

Prerequisites

How To...

Install Client Libraries

Authorize Requests

Use the Knowledge Graph Search Widget

Terms of Service

## Google Knowledge Graph Search API



The Knowledge Graph Search API lets you find entities in the [Google Knowledge Graph](#). The API uses standard [schema.org](#) types and is compliant with the [JSON-LD](#) specification.

### Typical use cases

Some examples of how you can use the Knowledge Graph Search API include:

- Getting a ranked list of the most notable entities that match certain criteria.
- Predictively completing entities in a search box.
- Annotating/organizing content using the Knowledge Graph entities.

 **Note:** The Knowledge Graph Search API is a read-only API.

For detailed information about the API methods and parameters, see the [API Reference](#).

### Sample request

The following example shows one kind of request you can send to the API. (But check the [Prerequisites](#) section first. You'll also need to insert your own API key.)

<https://developers.google.com/knowledge-graph/>

Contents

Typical use cases

Sample request

Knowledge Graph entities

# THE ECONOMIC GRAPH





# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## Examples and Applications

Wikidata

# Open Knowledge Graphs



# Query the Knowledge of Wikipedia like a Database



## Wikipedia page

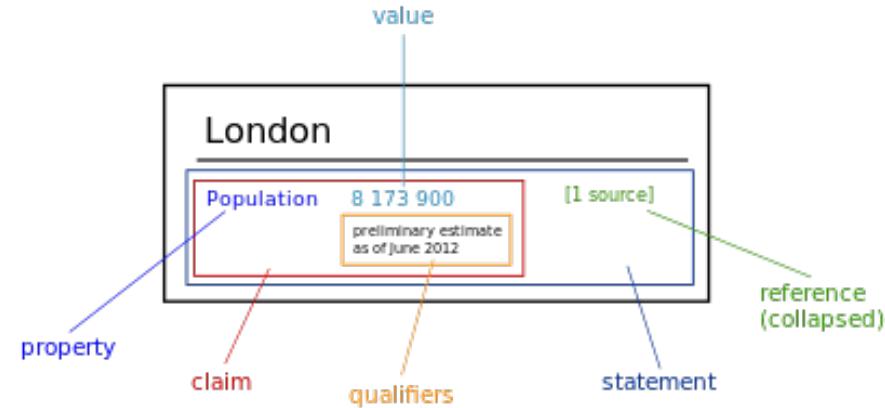
The screenshot shows the Wikipedia page for London. It includes a sidebar with navigation links like Main page, Contents, and Interaction. The main content area features a large image of the London skyline and text about its history as the capital of England and the United Kingdom, mentioning its ancient Roman roots and modern status as a global financial center. It also highlights its role as a cultural capital, mentioning the Royal Opera House and the British Museum.

## A query against Wikipedia

The screenshot shows a query results table from the metaphacts platform. The table lists the top 15 largest cities in the world where the mayor is a woman. The columns are labeled 'city', 'mayor', and 'population'. The data includes cities like Madrid, Paris, and London, along with their respective mayors and populations. The interface also features a search bar and a 'Filter Results' button.

city	mayor	population
Madrid	Ana Botella	3165235
Paris	Anne Hidalgo	2240621
Houston	Annise Parker	2195914
Warsaw	Hanna Gronkiewicz-Waltz	1726581
Prague	Adriana Krnáčová	1243201
Valencia	Rita Barberá Nolla	792303
Taguig	Lani Cayetano	644473
Dresden	Helma Orosz	531982
Zürich	Corine Mauch	400028
Resistencia	Aída Beatriz Máxima Ayala	300000

- **Collecting structured data.** Unlike the Wikipedias, which produce encyclopedic articles, Wikidata collects data, in a structured form.
- **Collaborative.** The data in Wikidata is entered and maintained by Wikidata editors, who decide on the rules of content creation and management in Wikidata supporting the notion of verifiability.
- **Free.** The data in Wikidata is published under the [Creative Commons](#)
- **Large.**
  - 25 million entities
  - 130 million statements
  - 130 million labels
  - 350 languages
  - >1500 million triples



# London (Q84)

capital city of England and the United Kingdom

London, England | London, UK | London, United Kingdom

[edit]

[Collapse]

► In more languages

Wikipedia (225 entries) [edit]

ab Лондан

ace London

af Londen

als London

am Անջն

ang Lunden

an Londres

arc ლონდონი

ar لندن

arz لندن

ast Londres

ay London

az London

bar London

bat\_smg Londons

ba Лондон

bcl Londres

be\_x\_old Лёндан

be Горад Лондан

bg Лондон

bh লংদন

bi London

bn লন্ডন

bo ښکوټنۍ

br Londrez

bs London

bxr Лондон

ca Londres

cbk\_zam London

cdo Lüng-düng

ceb London

ce Лондон

chr Ід Առահմէծ

ckb սահպ

co Londra

crh London

## Statements

topic's main Wikimedia portal

Portal:London

[edit]

▼ 0 references

[add reference]

[add]

instance of

metropolitan area

[edit]

▼ 0 references

[add reference]

city with millions of inhabitants

[edit]

► 1 reference

metropolis

[edit]

▼ 0 references

[add reference]

financial centre

[edit]

▼ 0 references

[add reference]

capital

[edit]

country

United Kingdom

England

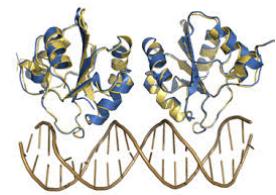
▼ 0 references

[add reference]

[add]

# Use Cases for the Wikidata Knowledge Graph

- **Build your applications** using Wikidata
  - Free corpus of structured knowledge
  - Easily accessible and standards-based
  - See <http://query.wikidata.org/>
- **Contextualize** your enterprise data
  - Wikidata provides stable identifiers into the open data world
  - Seamless integration of private data with open data
- **Enrich** Wikidata with your data
  - Contribute your data to Wikidata
  - Link to your own data, make it visible
  - Examples:
    - Open biomedical databases – Wikidata as a central hub
    - Cultural heritage



# Inventory

[add a book](#)[friends &  
groups](#)[books  
exchanges](#)

[Dan Brown](#)

American author

1964 -

country of citizenship: United States of America

native language: English

educated at: Amherst College

occupation: writer, novelist, teacher



Daniel "Dan" Brown (born June 22, 1964) is an American author of thriller fiction who is best known for the 2003

## • [The Da Vinci Code](#)

The Da Vinci Code is a 2003 mystery-detective novel by Dan Brown. It follows symbologist Robert Langdon and cryptologist Sophie Neveu after a murder in the Louvre Museum in Paris, when they become involved in a ...

author: [Dan Brown](#)

2003

[more details](#)

[+ add to my inventory](#)

**1** [in your network](#)

## [Angels & Demons](#)

author: [Dan Brown](#)

[more details](#)

[+ add to my inventory](#)

**0** [in your network](#)

## [Deception Point](#)

Deception Point is a 2001 techno-thriller novel by Dan Brown.

author: [Dan Brown](#)

[more details](#)

[+ add to my inventory](#)

**0** [in your network](#)

## [Ψηφιακό Οχυρό](#)

Digital Fortress is a techno-thriller novel written by American author Dan Brown and published in 1998 by St. Martin's Press.

author: [Dan Brown](#)

[more details](#)

[+ add to my inventory](#)

**0** [in your network](#)

## [The Lost Symbol](#)

The Lost Symbol is a 2009 novel written by American writer Dan Brown. It is a thriller set in Washington, D.C., after the events of The Da Vinci Code, and relies on Freemasonry for both its recurring theme and its major characters...

author: [Dan Brown](#)

[more details](#)

[+ add to my inventory](#)

**0** [in your network](#)



Top 20 Timelines

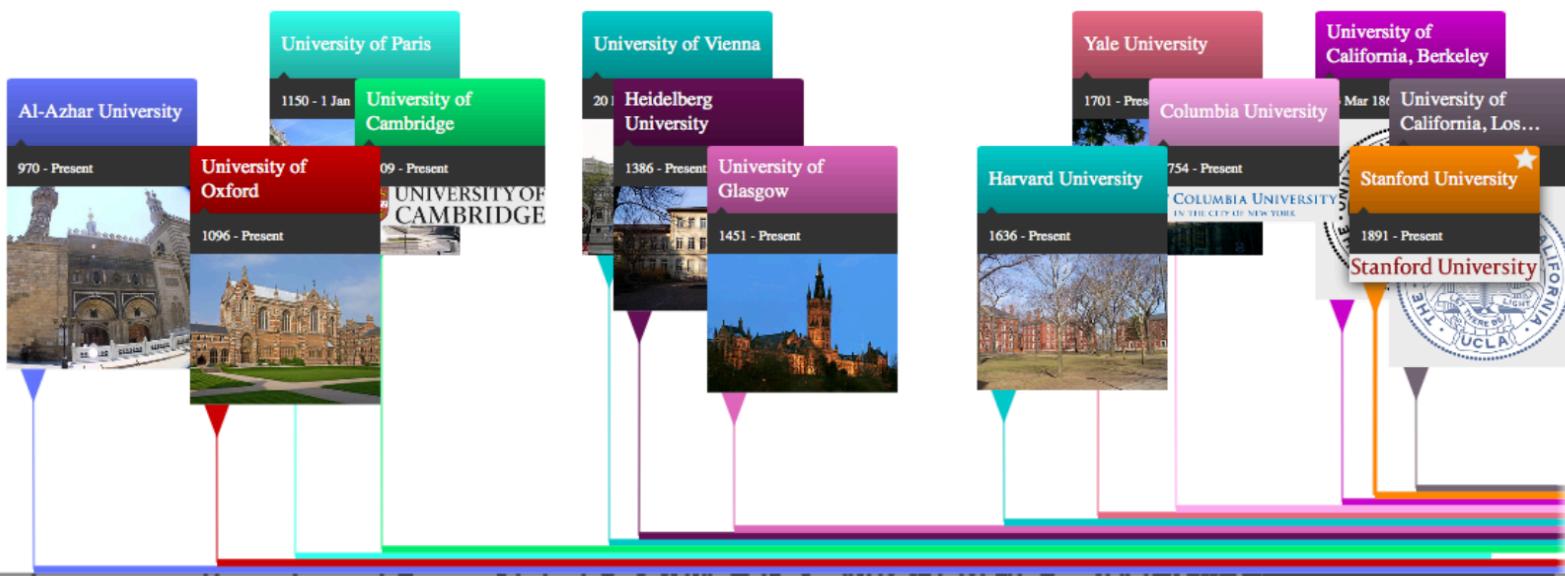
Start searching for your topic...



Compact

Save

Sign In





# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## Examples and Applications

Cultural Heritage

- Challenge:
  - Very context-rich data
  - Multi-disciplinary data, e.g. archaeologists, historians, librarians
  - Multi-institutional data
  - Complex domain, relationships, e.g. temporal, spatial, historical, political
- Benefits of Knowledge Graphs
  - Integration and interchange of heterogeneous cultural heritage information
  - Rich ontologies for knowledge representation
  - Deep semantics for true conceptual merging
  - Multi-lingual knowledge representation
  - Knowledge access across museums and organizations
  - Enabling knowledge sharing and collaboration

- Collaboration environment for researchers in Cultural Heritage
  - Expert users: researchers, curators
- Based on CIDOC-CRM: very rich, expressive ontology
- Large, cross-museum data sets
  - E.g. British Museum: 100s millions of triples
- Advanced search capabilities
  - Supporting query construction
  - Sharing of searches, results, visualizations
- Knowledge sharing
  - Discussions around cultural heritage annotations
  - Argumentation support:  
Representation of conflicting views and opinions



The British  
Museum



<http://researchspace.org/>

ResearchSpace

Example Records ▾ Login ?

Home

Thing refers to Place Vesuvius, Mount remove

has material type (120) Hide Filter

- onyx (1)
- opal (1)
- organic (3)
- ostrich eggshell (1)
- palm nut (1)
- palm stem (1)
- paper (120)
- pear wood (1)
- pearl (1)
- pearlshell (2)

has technique type (120)

has type (120)

Found 120 matches

Grid Table Chart



Montis Vesvii fauces et ...



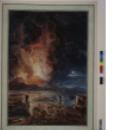
View of Mount Vesuvius, ...



Eruption of a volcano at n...



View of the Bay of Naple...



Eruption of Vesuvius, 8th ...



Mount Vesuvius from San...



Crater of Vesuvius, paper ...



Eruption of the Soufrière ...





# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## Examples and Applications

Life Sciences

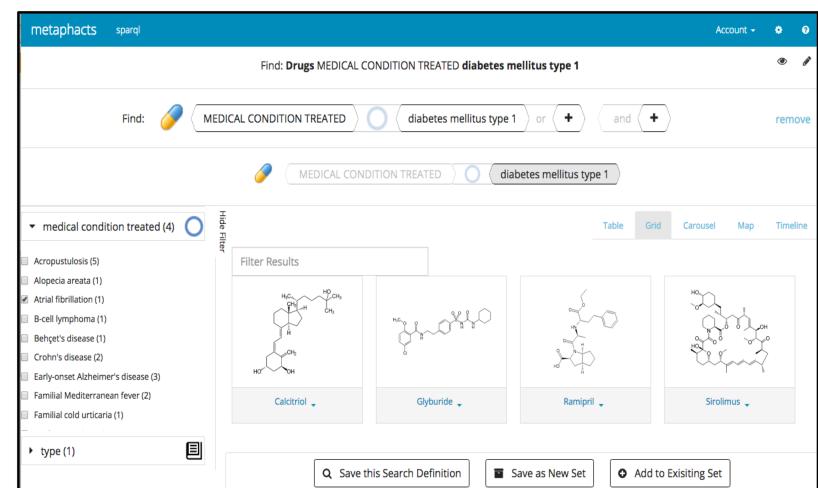
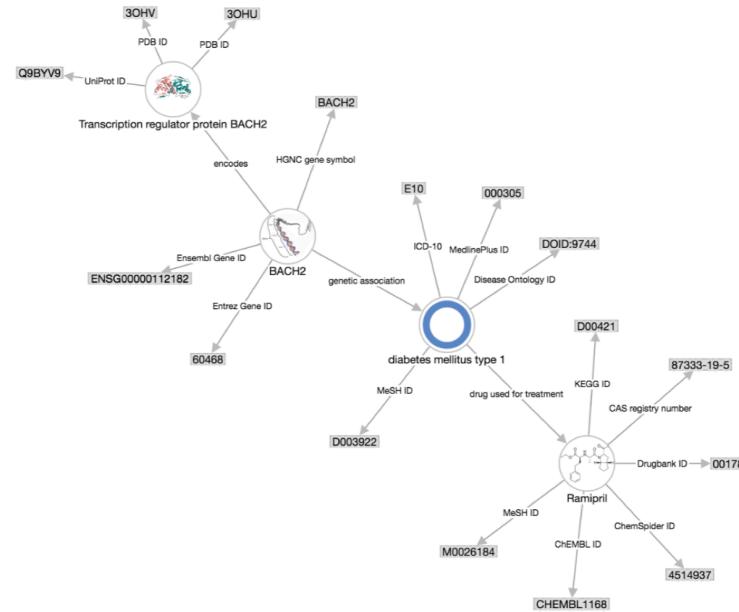
## Challenge:

- Much of the relevant knowledge in external databases
- Many disparate databases / data silos
- Many different data formats
- Complex domain, complex relationships  
e.g. *compounds, targets, pathways, diseases and tissues*



# Benefits of Knowledge Graphs in the Life Sciences

- Integrated knowledge representation
  - Common format
  - Stable, global identifiers
  - Federated queries
- Integrated knowledge access: One-stop portals
  - Rich semantic search on a conceptual level
  - Entry points to further data, in-house and external
  - Crossing boundaries between private and open data



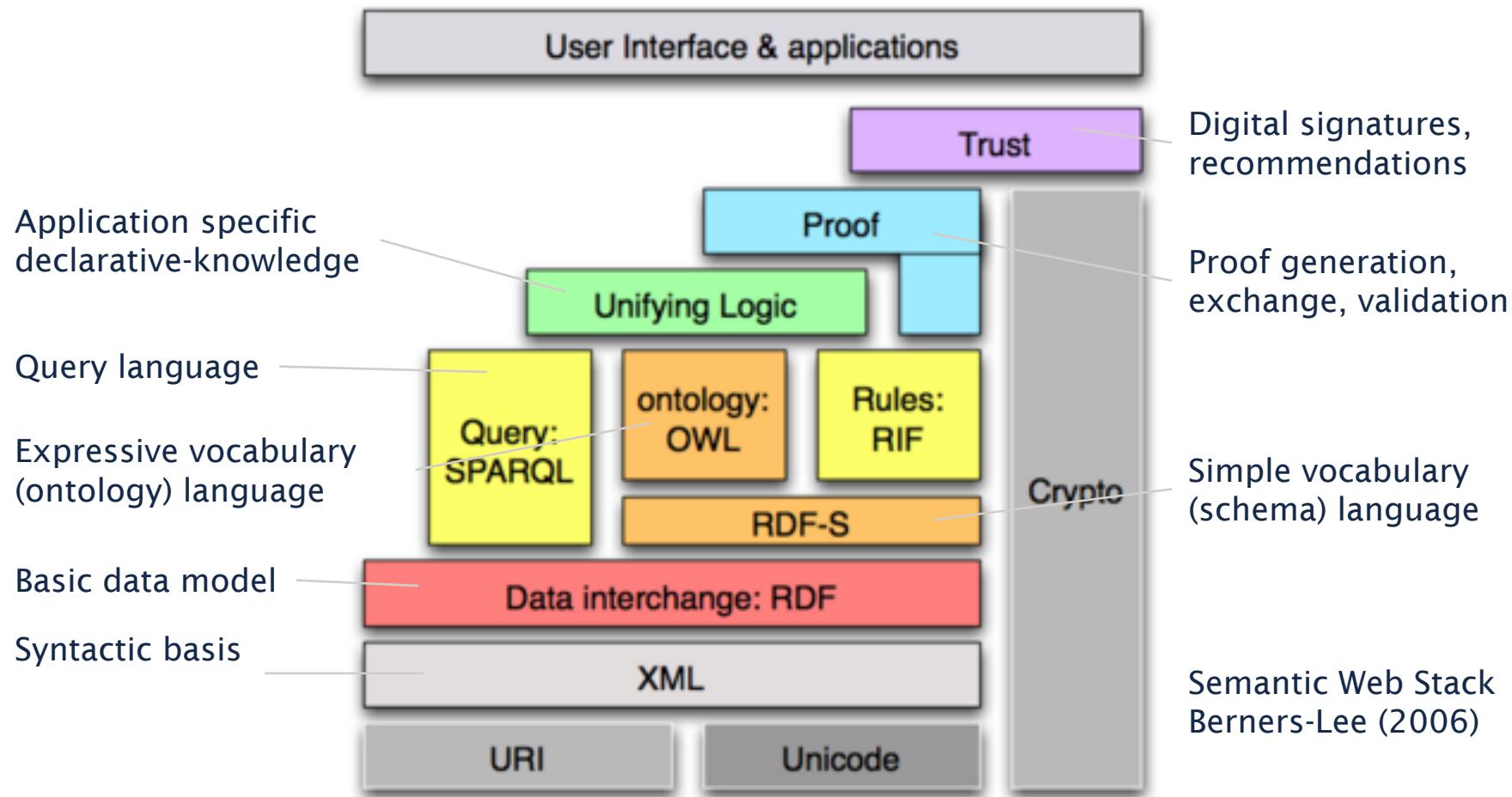


# metaphacts

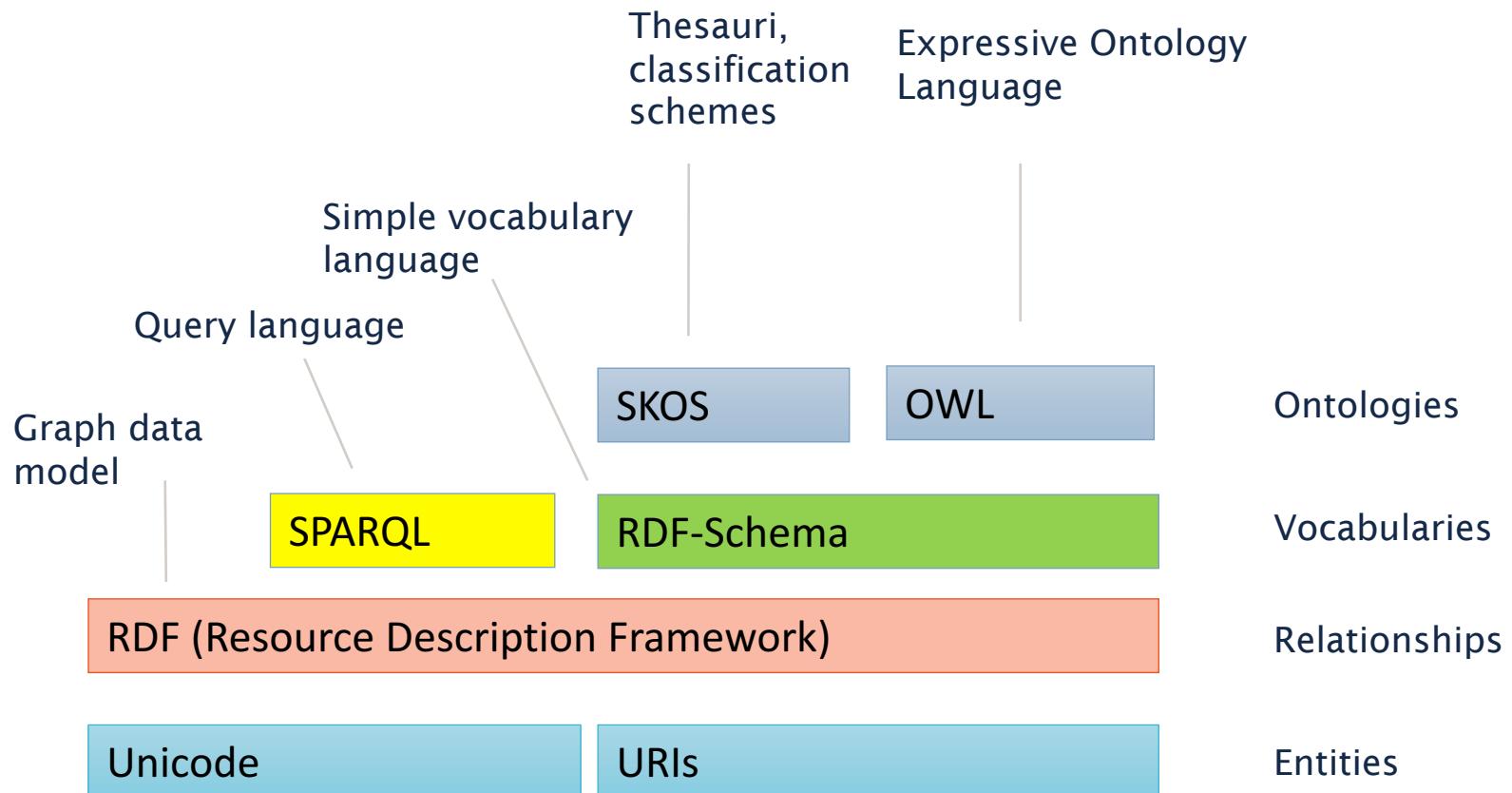
Smart Data. Smart Apps. Smart Decisions.

## Standards and Principles





# Knowledge Graphs Built on the Semantic Web Layer Cake

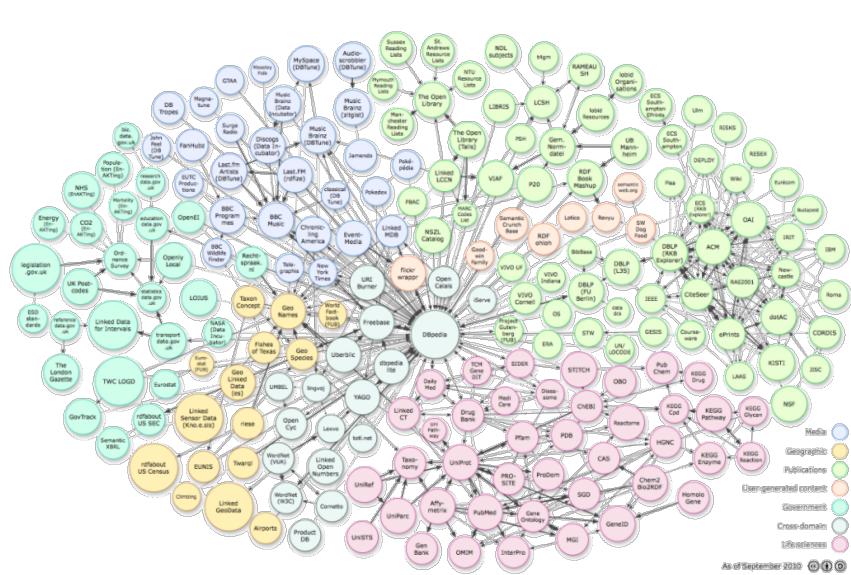


## Linked Data

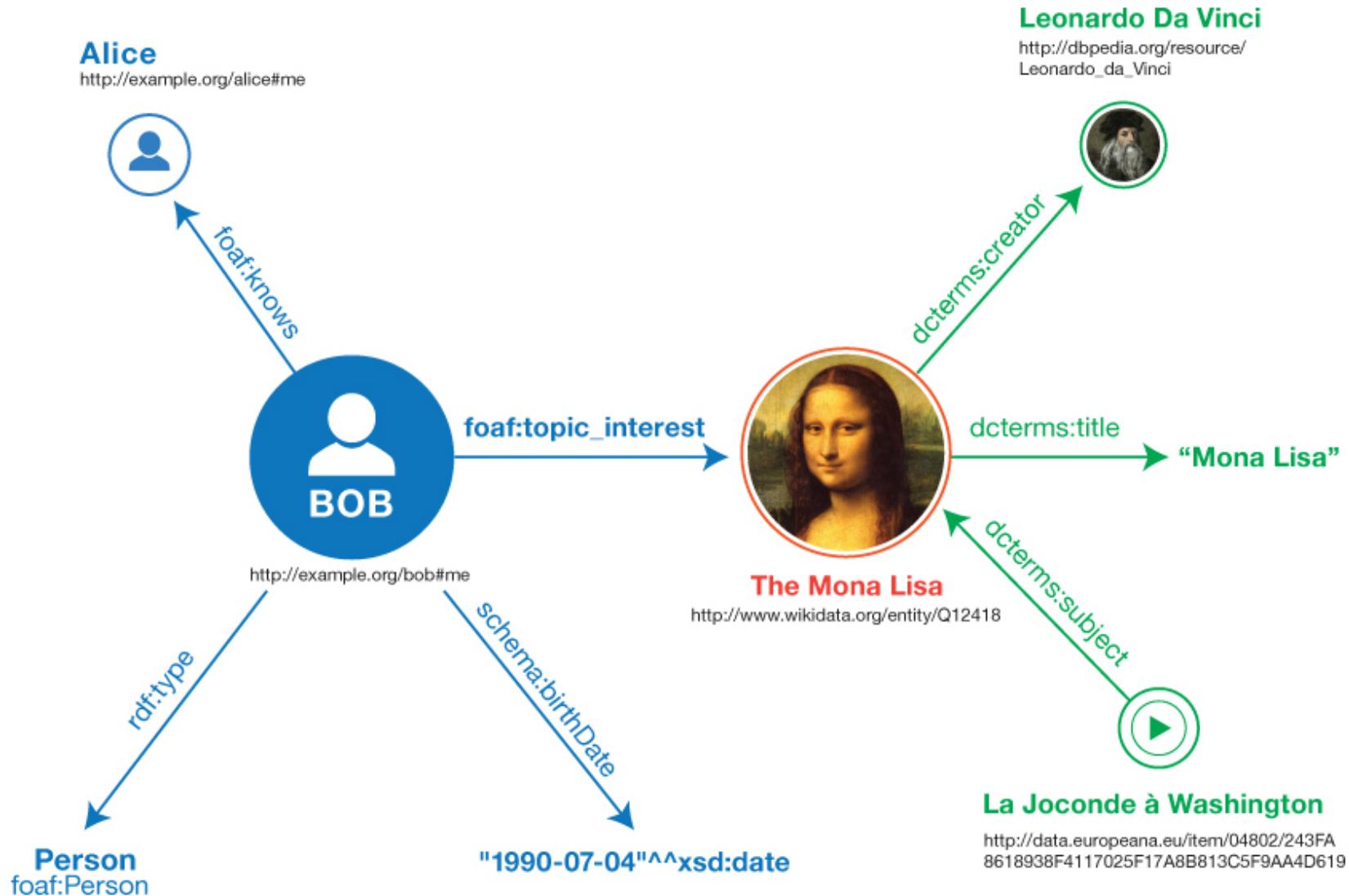
- Set of standards, principles for publishing, sharing and interrelating structured knowledge
- From data silos to interconnected knowledge graphs

## Linked Data Principles

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards: RDF, SPARQL.
- Include links to other URIs, so that they can discover more things.



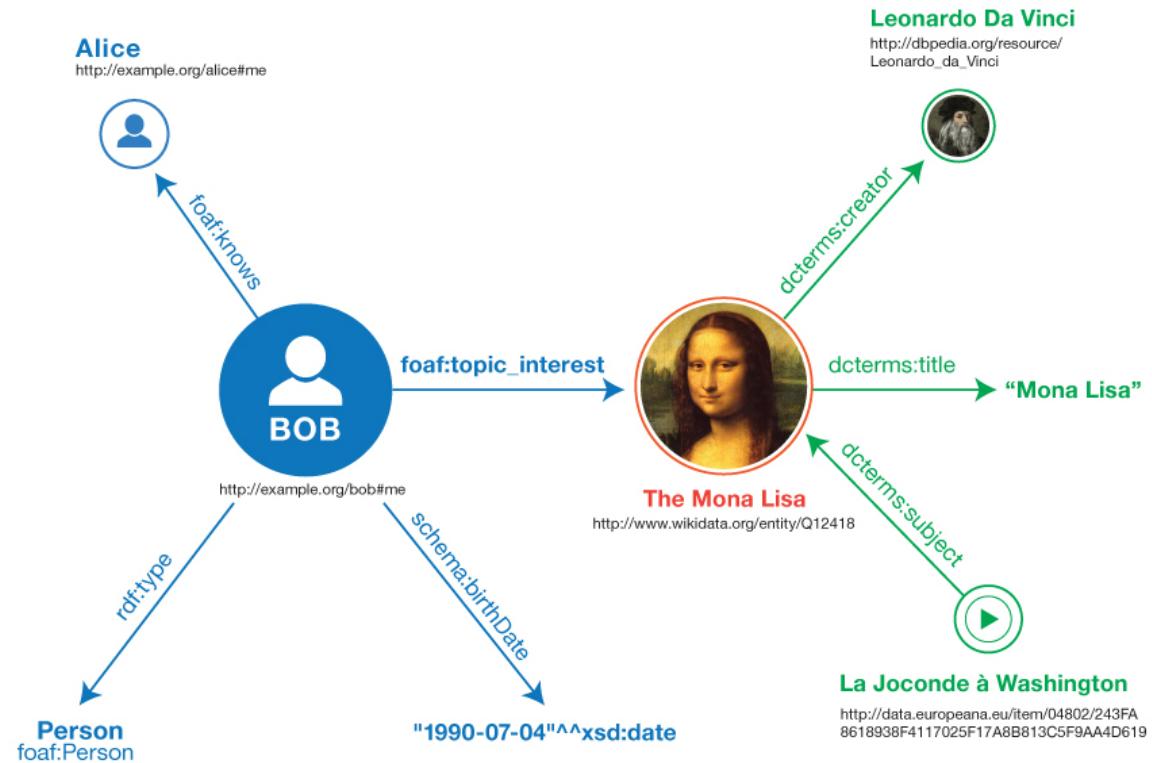
# Our Knowledge Graph again (a bit more technical)



# Our Knowledge Graph again (a bit more technical)

Graph consists of:

- Resources (identified via URIs)
- Literals: data values with data type (URI) or language (multilinguality integrated)
- Attributes of resources are also URI-identified (from *vocabularies*)
- Various data sources and vocabularies can be arbitrarily mixed and meshed
- URIs can be shortened with namespace prefixes; e.g. schema: → <http://schema.org/>



*Allows one to talk about anything*

Uniform Resource Identifier (URI) can be used to identify entities

[http://dbpedia.org/resource/Leonardo\\_da\\_Vinci](http://dbpedia.org/resource/Leonardo_da_Vinci)

is a name for **Leonardo da Vinci**

dbpedia:  
Leonardo\_da\_Vinci

<http://www.wikidata.org/entity/Q12418>

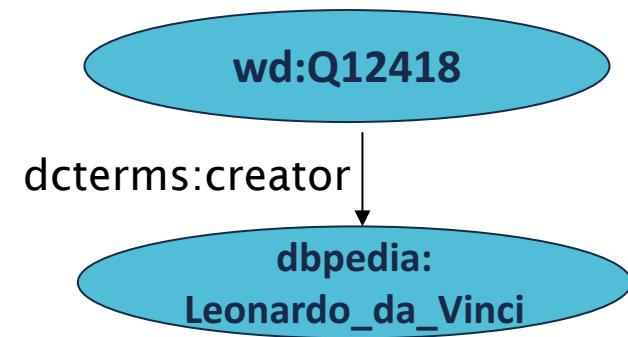
is a name for the **Mona Lisa** painting

wd:Q12418

## *Allows one to express statements*

An RDF statement consists of:

- Subject: resource identified by a URI
- Predicate: resource identified by a URI
- Object: resource or literal



Variety of RDF syntaxes,  
e.g. Turtle (Terse RDF Triple Language):

```
wd:Q12418 dcterms:creator dbpedia:Leonardo_da_Vinci .
```

- Language for two tasks w.r.t. the RDF data model:
  - Definition of vocabulary – nominate:
    - the ‘types’, i.e., classes, of things we might make assertions about, and
    - the properties we might apply, as predicates in these assertions, to capture their relationships.
  - Inference – given a set of assertions, using these classes and properties, specify what should be inferred about assertions that are implicitly made.

- **rdfs:Class** – Example:  
`foaf:Person` – *Represents the class of persons*
- **rdf:Property** – Class of RDF properties. Example:  
`foaf:knows` – *Represents that a person “knows” another*
- **rdfs:domain** – States that any resource that has a given property is an instance of one or more classes  
`foaf:knows rdfs:domain foaf:Person`
- **rdfs:range** – States that the values of a property are instances of one or more classes  
`foaf:knows rdfs:range foaf:Person`

Schema    foaf:knows  
            rdfs:range  
            foaf:Person .

Existing fact    <http://example.org/bob#me>  
                  foaf:knows  
                  <http://example.org/alice#me>.

Inferred fact    <http://example.org/alice#me>  
                  rdf:type  
                  foaf:Person.

We *expect* to use this vocabulary to make assertions about persons.

Having made such an assertion...

***Inferences*** can be drawn that we did not explicitly make

- RDFS provides a simplified ontological language for defining vocabularies about specific domains.
- OWL provides more ontological constructs for knowledge representation.
- Semantics grounded in **Description Logics**.
- OWL 2 is divided into sub-languages denominated profiles:
  - OWL 2 EL: Limited to basic classification, but with polynomial-time reasoning
  - OWL 2 QL: Designed to be translatable to relational database querying
  - OWL 2 RL: Designed to be efficiently implementable in rule-based systems
- Most graph databases concentrate on the use of RDFS with a subset of OWL features.

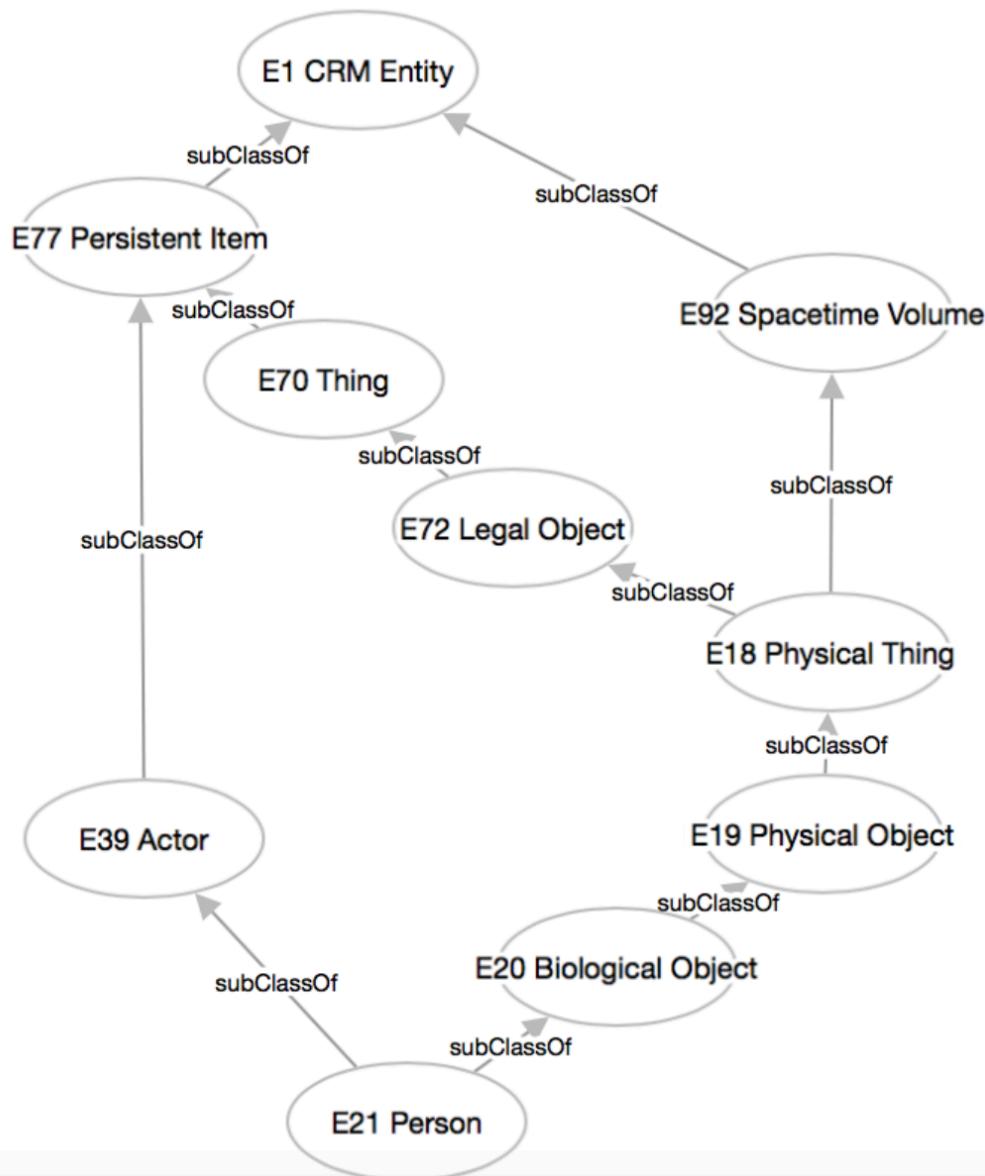


More restrictive  
than OWL DL

OWL is made up of **terms** which provide for:

- **Class construction:** forming new classes from membership of existing ones (e.g., unionOf, intersectionOf, etc.).
- **Property construction:** distinction between OWL ObjectProperties (resources as values) and OWL DatatypeProperties (literals as values).
- **Class axioms:** sub-class, equivalence and disjointness relationships.
- **Property axioms:** sub-property relationship, equivalence and disjointness, and relationships between properties.
- **Individual axioms:** statements about individuals (sameIndividual, differentIndividuals).

## Example: CIDOC-CRM Ontology



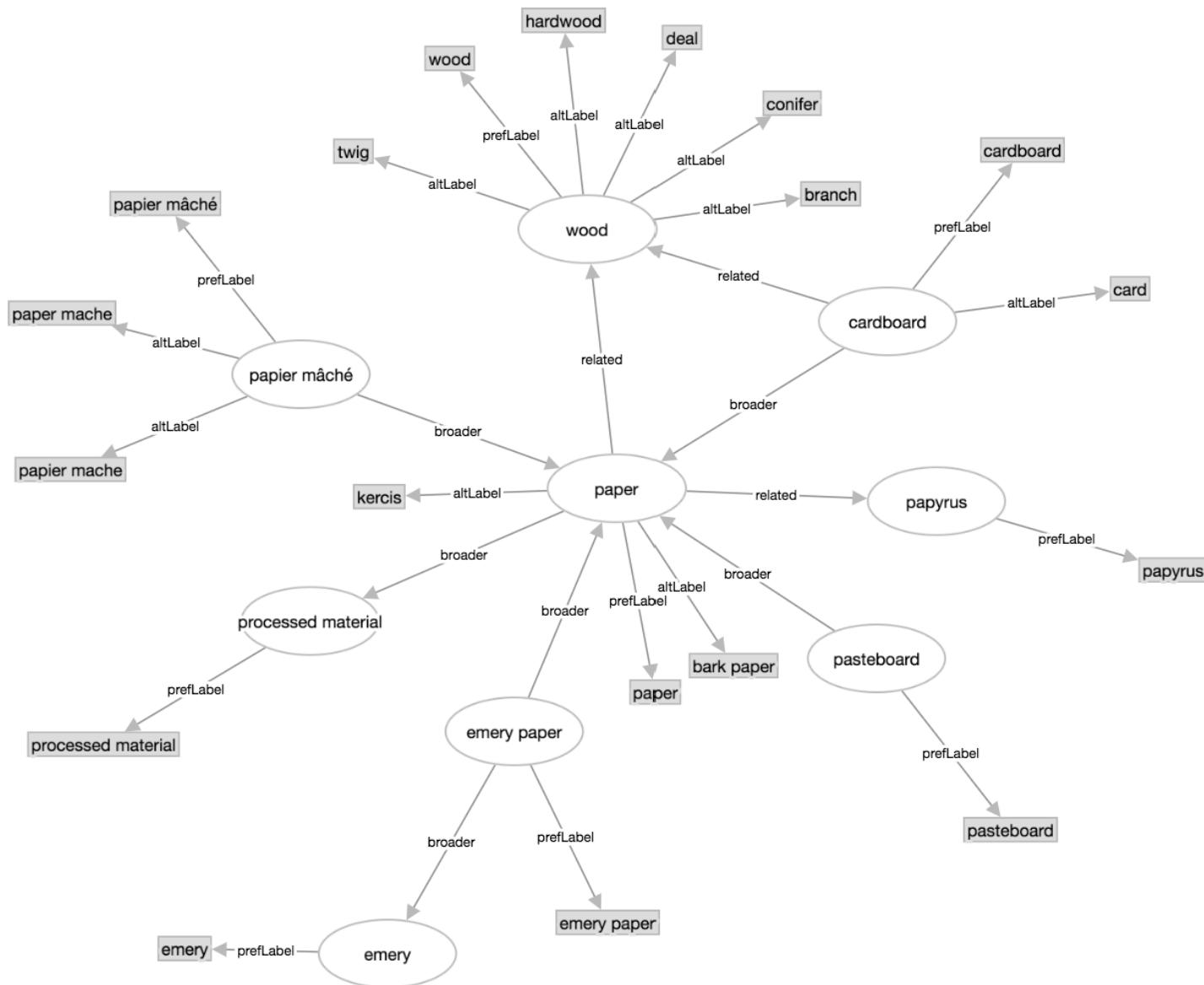
**Class: Physical Thing**  
**SubClassOf:** Legal Object  
**SubClassOf:** Spacetime Volume  
**DisjointWith:** Conceptual Object  
**SubClassOf:** consists\_of some Material

**Class: Person**  
**SubClassOf:** Actor  
**SubClassOf:** Biological Object  
**SubClassOf:** was\_born exactly 1  
**SubClassOf:** has\_parent min 2

- Data model for knowledge organization systems (thesauri, classification scheme, taxonomies)
- Conceptual resources (concepts) can be
  - identified with URIs,
  - labeled with lexical strings in natural language,
  - documented with various types of note,
  - semantically related to each other in informal hierarchies and association networks and
  - aggregated into concept schemes.

<http://www.w3.org/TR/skos-reference/>

# Example: Concept Definition for Paper



- Query language for RDF-based knowledge graphs.
- Designed to use a syntax similar to SQL for retrieving data from relational databases.
- Different query forms:
  - **SELECT** returns variables and their bindings directly.
  - **CONSTRUCT** returns a single RDF graph specified by a graph template.
  - **ASK** test whether or not a query pattern has a solution. Returns yes/no.
  - **DESCRIBE** returns a single RDF graph containing RDF data about resources.

## Main idea: Pattern matching

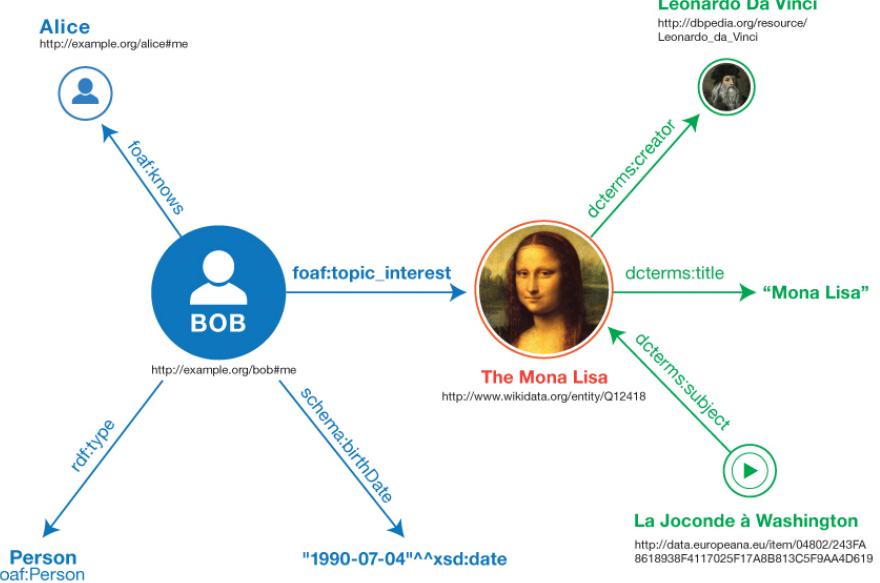
- Queries describe sub-graphs of the queried graph
- **Graph patterns** are RDF graphs specified in Turtle syntax, which contain variables (prefixed by either "?" or "\$")



- Sub-graphs that match the graph patterns yield a **result**
- The syntax of a **SELECT** query is as follows:
  - **SELECT** nominates which components of the matches made against the data should be returned.
  - **FROM** (optional) indicates the sources for the data against which to find matches.
  - **WHERE** defines patterns to match against the data.
  - **ORDER BY** defines a means to order the selected matches.

## Example:

*Select the creator of the things that Bob is interested in.*



PREFIX foaf: <<http://xmlns.com/foaf/0.1/>>

PREFIX dcterms: <<http://purl.org/dc/terms/>>

SELECT ?creator

WHERE {

```

<http://example.org/bob#me> foaf:topic_interest ?interest .
?interest dcterms:creator ?creator
}
```

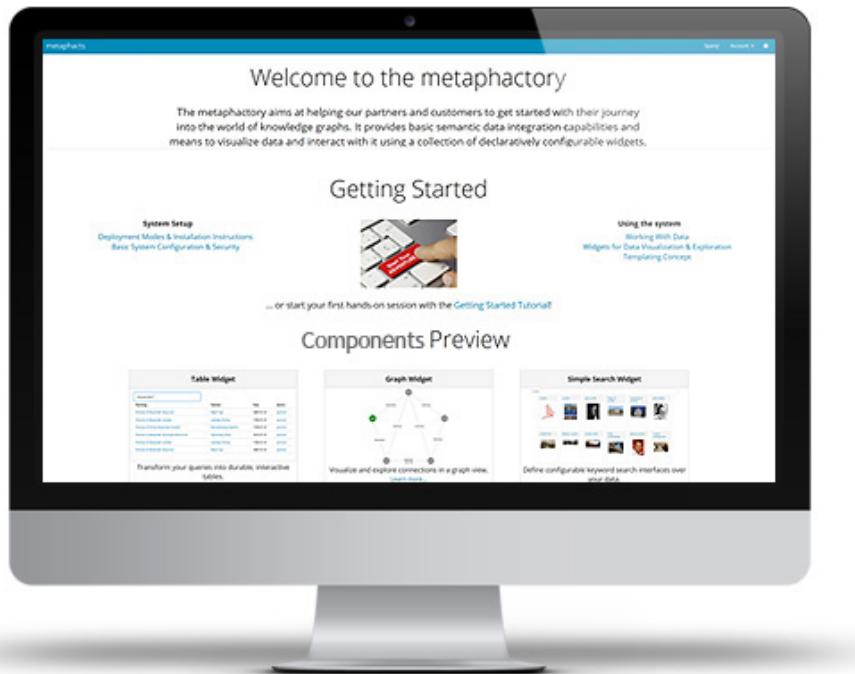
Results: [dbpedia:Leonardo\\_da\\_Vinci](http://dbpedia.org/resource/Leonardo_da_Vinci)



# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## metaphactory Knowledge Graph Platform



The metaphacts team offers an unmatched experience and know-how around enterprise knowledge graphs for our clients in areas such as business, finance, life science, and cultural heritage.

The metaphactory is our end-to-end platform to create and utilize enterprise knowledge graphs - from semantic graph data management to data-driven application development.

Built entirely on open standards and technologies, our platform covers the entire lifecycle of dealing with knowledge graphs.

As a main benefit our platform enables knowledge workers to create and gain meaningful insight into their data with one comprehensive software solution.



## KNOWLEDGE GRAPH BACKEND

- Scalable data processing
- Easy-to-use interface
- High-performance querying and analytics
- Built-in inferencing and custom services
- Standard connectors for a variety of data formats
- Single server, embedded mode, high availability, and scale out



## KNOWLEDGE GRAPH CREATION

- Semi-automatic creation of knowledge graphs
- Curation and interlinking of data from heterogeneous sources
- Collaborative management and authoring
- Custom query and templates catalogs
- Data annotation
- Capturing of provenance information



## KNOWLEDGE GRAPH APPLICATIONS

- Rapid development of end-user oriented applications
- Web components for end-user friendly presentation and interaction
- Interactive visualization
- Rich semantic search with visual query construction and faceting
- Customizable semantic clipboard

## BUILT IN OPEN SOURCE

- ✓ Dual licensing (LGPL & commercial license)
- ✓ Open Platform API and SDK
- ✓ Integration of external tools and application via APIs
- ✓ Easy development of own web components and services
- ✓ Full HTML5 compliance
- ✓ Re-usable, declaratively configurable Web Components



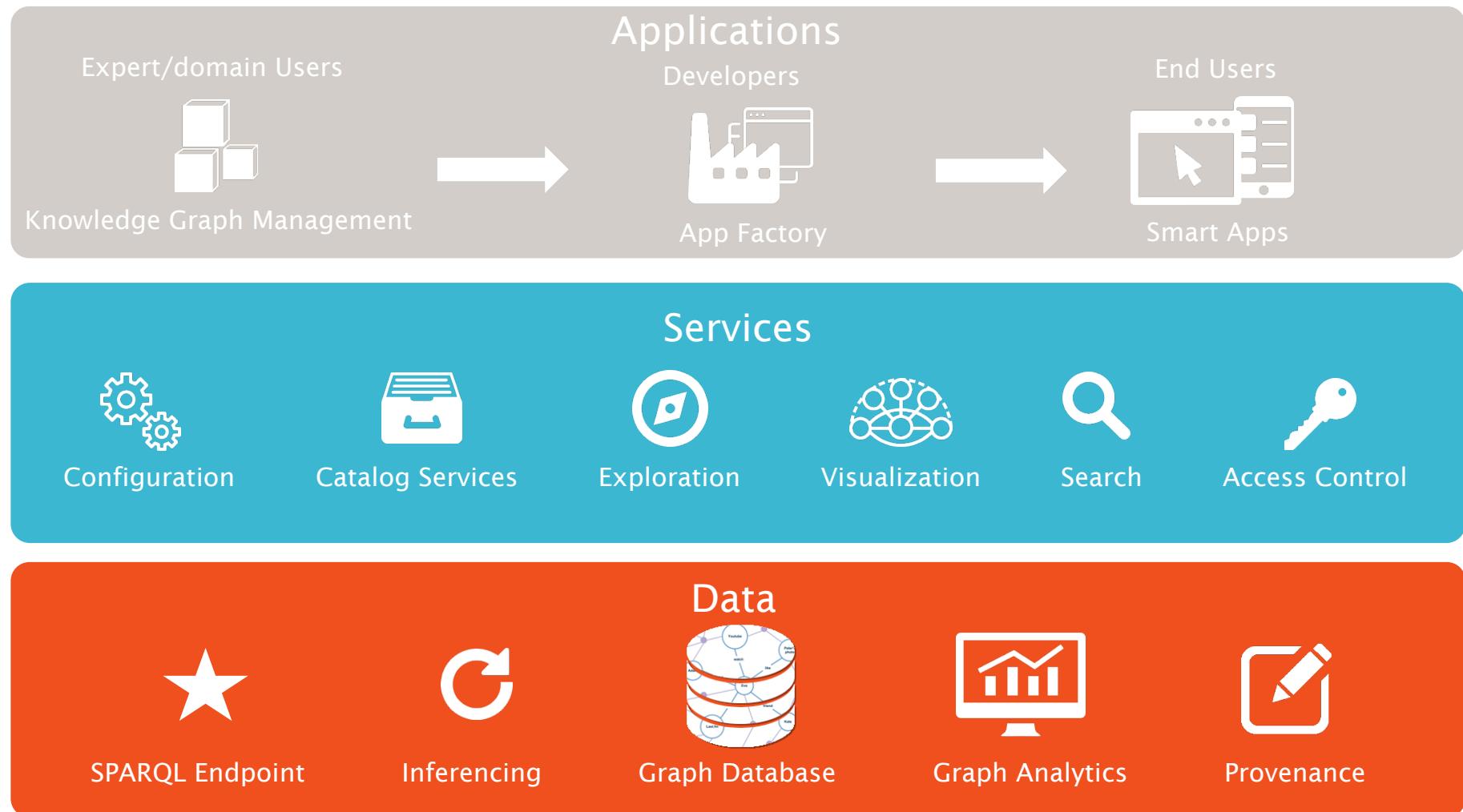
= Easy modification, customization, and extensibility

## BUILT ON OPEN STANDARDS

- ✓ W3C Web Components
- ✓ W3C Open Annotation Data Model
- ✓ W3C Linked Data Platform Containers
- ✓ Data processing based on W3C standards such as RDF, SPARQL
- ✓ Expressive ontologies for schema modeling based on OWL 2, SKOS
- ✓ Rules, constraints, and query specification based on SPIN and RDF Data Shapes



= Sustainable Solution





## EXPERT USERS

- Collaboratively construct and manage knowledge graphs
- Integrate data from heterogeneous sources
- Use standard connectors for a variety of data formats
- Benefit from scalable data processing for big graphs
- Conduct high-performance querying and analytics



## DEVELOPERS

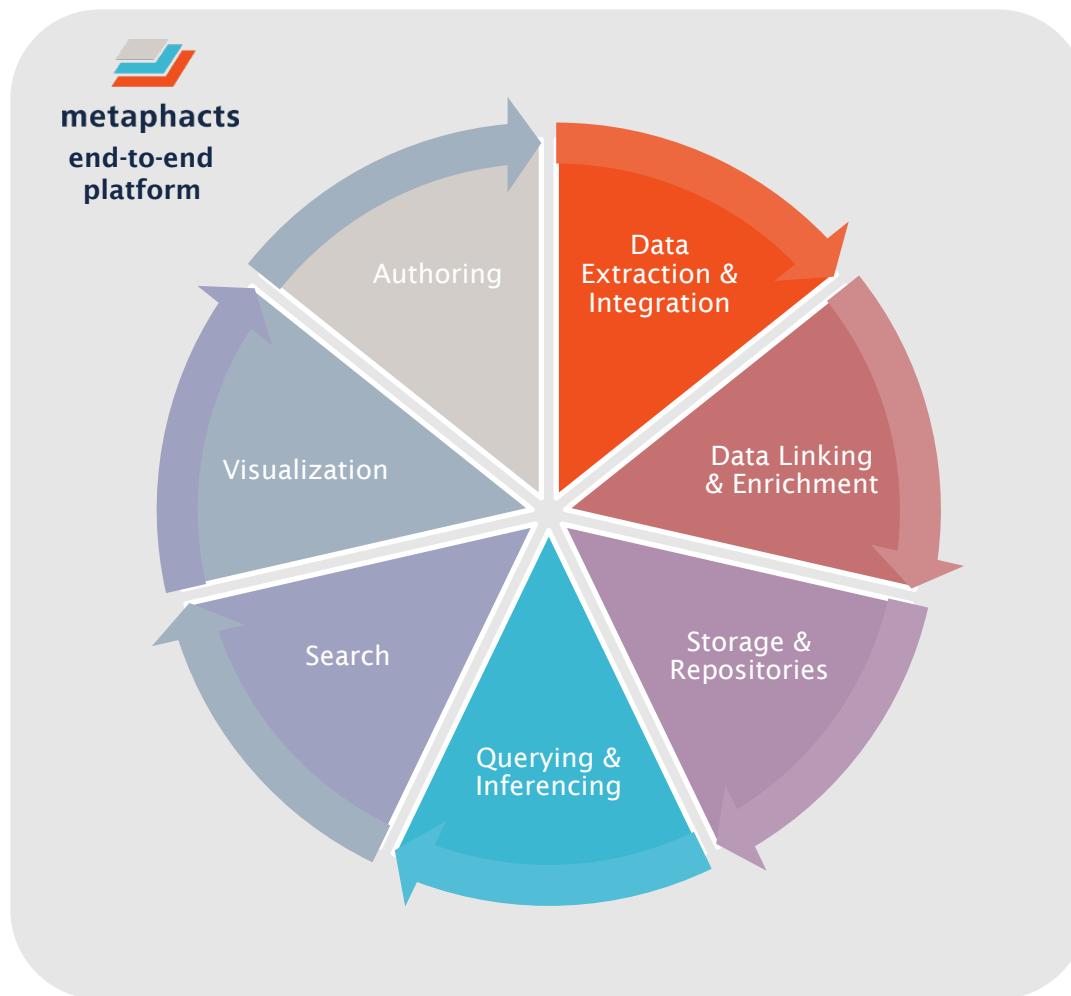
- Rapidly develop Web and mobile end-user oriented applications
- Benefit from various deployment modes: stand-alone, HA, scale-up, scale-out
- Interact with an easy-to-use interface
- Collaboratively manage, annotate and author data
- Use large set of custom query and templates catalogs
- Capture of provenance information

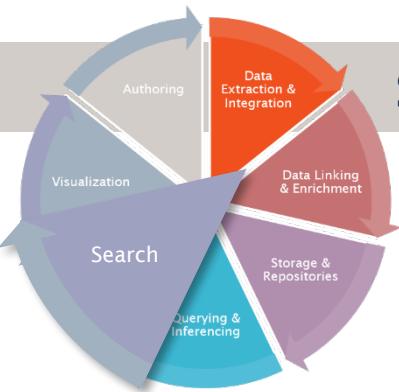


## END USERS

- Benefit from user-friendly interaction with data
- Gain insights into complex relationships
- Enable transparency and extract value
- Ask questions and obtain precise results
- Reduce effort for data analysis
- Reduce noise – obtain targeted, high quality results
- Enhance quality of business decisions

# metaphacts Supports the Whole Data Lifecycle



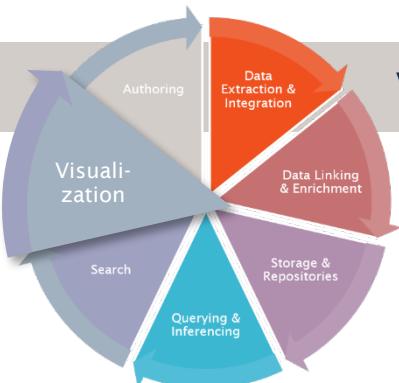


# Search

- Domain independent, fully customizable search widget
- Satisfy complex information needs without learning SPARQL
- Search functionalities
  - Graphical query construction
  - End user friendly search interfaces for building and sharing complex queries
  - Semantic auto suggestion
  - Interactive result visualization
  - Faceted search and exploration of item collections
- Ability to invoke external full text search indices such as Solr including the possibility to score, rank and limit the results for responsive autosuggestion
- Saving and sharing of queries and search results

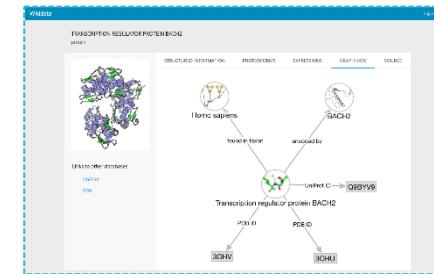
The screenshot illustrates the ResearchSpace search interface. At the top, there's a navigation bar with 'Example Records' and 'Login' buttons. Below the navigation is a search bar with a breadcrumb trail: 'Thing' > 'refers to' > 'Place' > 'Vesuvius, Mount'. To the right of the search bar is a 'remove' button. On the left side, there's a sidebar with filters for 'has material type' (120) and 'has technique type' (120), each with a semantic auto-suggestion dropdown. The main content area shows a grid of 120 search results, each with a thumbnail image and a caption. Some captions include ellipses, indicating truncated text.

# Visualization



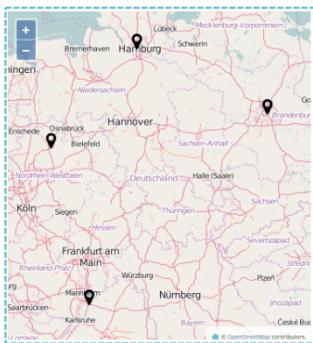
## Carousel

Animated browsing through a list of result items



## Graph

Visualize and explore connections in a graph view  
Custom styling of the graph  
Variety of graph layouts



## Map

Displaying spatial data on a geographic map

## Registered Events

- ▼ Hazards
  - Environmental\_Hazards
  - Human\_Induced\_Hazards
    - Communication
    - Infrastructure
    - Transportation
    - War
  - Natural\_Hazards
  - Technological\_Hazards



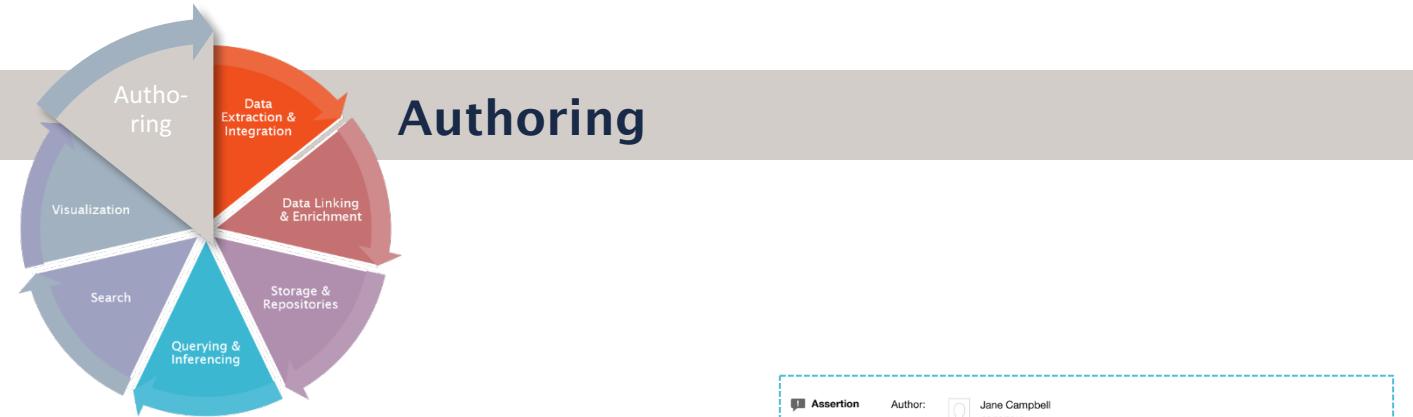
## Chart

Visualize trends and relationships between numbers, ratios, or proportions

Filter Results		
Subject	Predicate	Object
alice	foafknows	bob
alice	foafknows	carol
carol	foafknows	mike
mike	foafknows	carol
bob	foafknows	carol

## Table

Transform your queries into durable, interactive tables  
Many customization possibilities, e.g. pagination, filters and cell templates



## Authoring

- Annotations
  - Based on W3C Open Annotation Data Model
  - Automated semantic link extraction
- Form based authoring
  - Manually author and update instance data, backed by query templates, data dependencies, and type constraints
  - Rich editing components for special data types
  - Customizable flexible forms
  - Autosuggestion and validation against the knowledge graph
- Capturing of provenance information
- User group management

**Assertion**

Author: Jane Campbell  
20/09/2015

Title: Identity of paper supplier is clearly established

Proposition:

- Surimono, Woodblock Print → consists of → Paper Agree
- Surimono, Woodblock Print → consists of → Ink Disagree

**Supporting Argument**

Observation

Belief Adoption

Inference Narrower Term

Open Clipboard

Japanese Painting: Kokin waka shu

Basic Information Acquisition Other

CRM Preferred Identifier: J016009

All identifiers: 8333  
[ja:PgAdd102 Japanese Painting Additional Number] 1992.0701.0.1F

Acquisition Begin Date: 1/2/2015  
01/01/1992

January 1992

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Reset Submit

Sphaera Books Parts Persons Places

HOME / BOOKS / LECTIOINES ASTRONOMICAE IN SPHAERAM IOANNIS DE SACRO BOSCO

Lectioines astronomicae in sphaeram loannis de Sacro Bosco

**Description**

Short Title\*

Publication

Book Title

Authors

Printing

Locators

Parts

Meta

Delete

Statements

Book Type

Language

URL To PDF

Number Of Pages

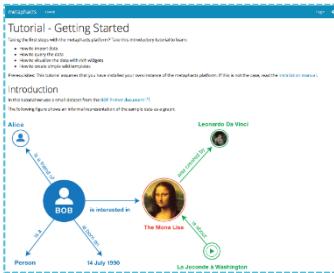
Save

# Install & Go: Out-of-the-Box Functionality



## Get started

**Getting Started Tutorial**  
to guide you through your  
first steps with metaphactory

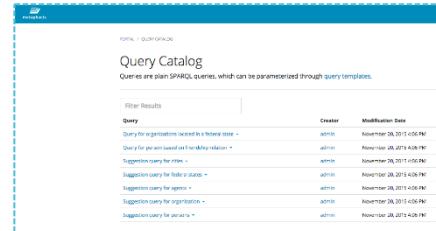


## Search

**Keyword Search Interface**  
with semantic autosuggestion,  
driven by SPARQL

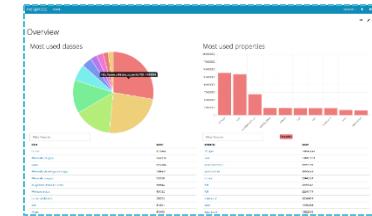


**Management of Queries in Catalog**  
for easy reuse and updating



## Explore

**Data Overview Pages**  
with Web components for  
end-user friendly data  
presentation and interaction



**Template-based Data Browser**  
used to define generic views  
which are automatically applied  
to entire sets of instances



## Example: Simple Semantic Search

Keyword search with semantic autosuggestion, driven by SPARQL

Set up in ~2 minutes!

1

Developer embeds  
'semantic-simple-search'  
into the page

```
<semantic-simple-search data-  
config='{  
  "query": "  
    SELECT ?result ?label ?desc  
    ?img WHERE {  
      ?result rdfs:label ?label .  
      ?result rdfs:comment ?desc .  
      ?result foaf:thumbnail ?img .  
      FILTER(CONTAINS(?label,  
      ?token))  
    }",  
    "searchTermVariable": "token", //  
    user input  
    "template": "  
      <span title=\"{{result}}\">  
        <img src=\"{{img}}\"  
        height=\"30\"/>  
        {{label}} {{desc}}</span>  
    \"' />
```

2

Rendered component is  
displayed to the user and  
can be used right away

Search for something

3

Autosuggestions are  
dynamically computed  
based on query and user  
input

Amazons

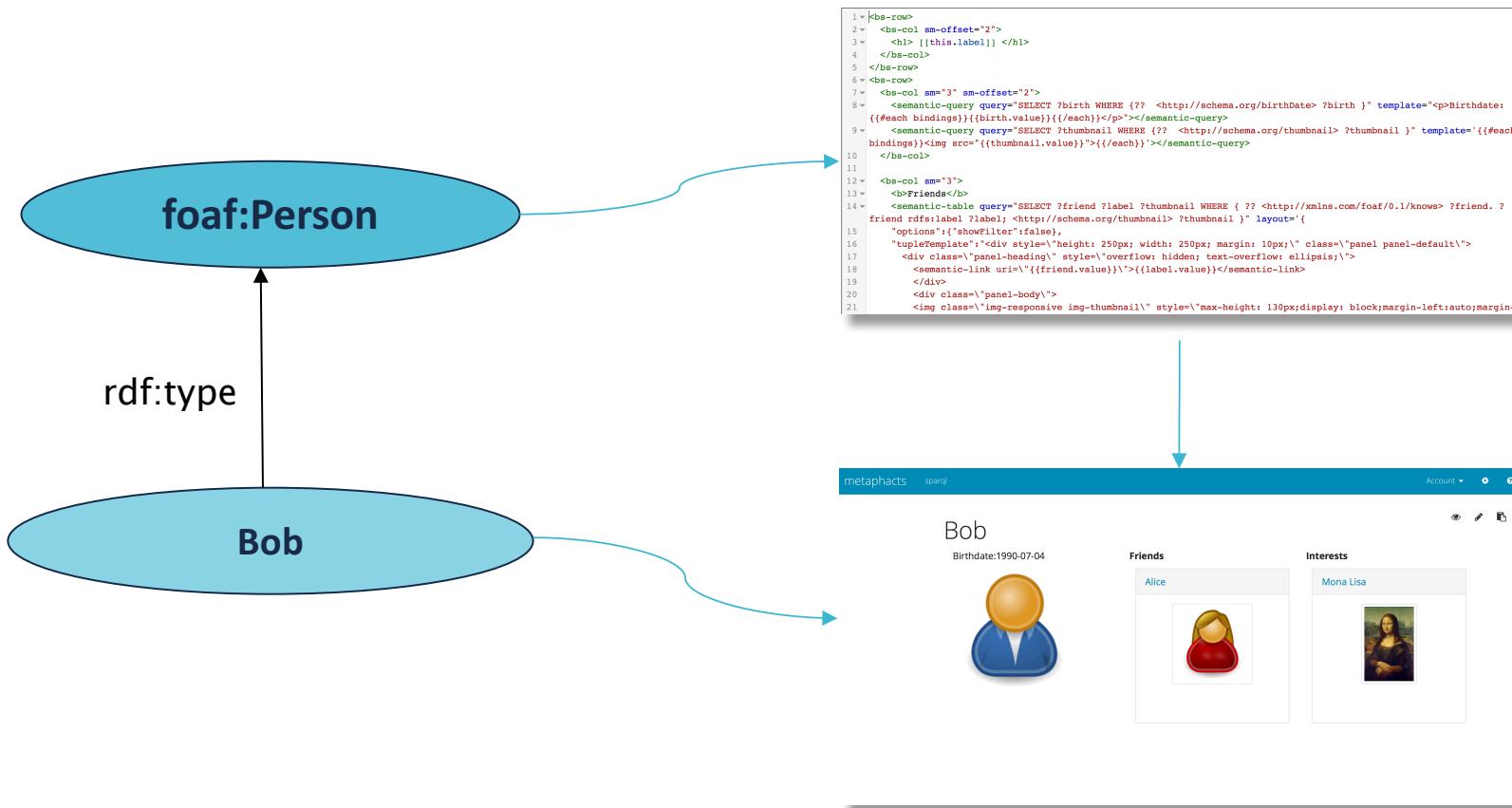


Amazonas (longest river in  
South America)



Amazon.com (American  
electronic commerce and cloud  
computing company)

- Associate a class in the knowledge graph with a template
  - The template is applied to instances of the class

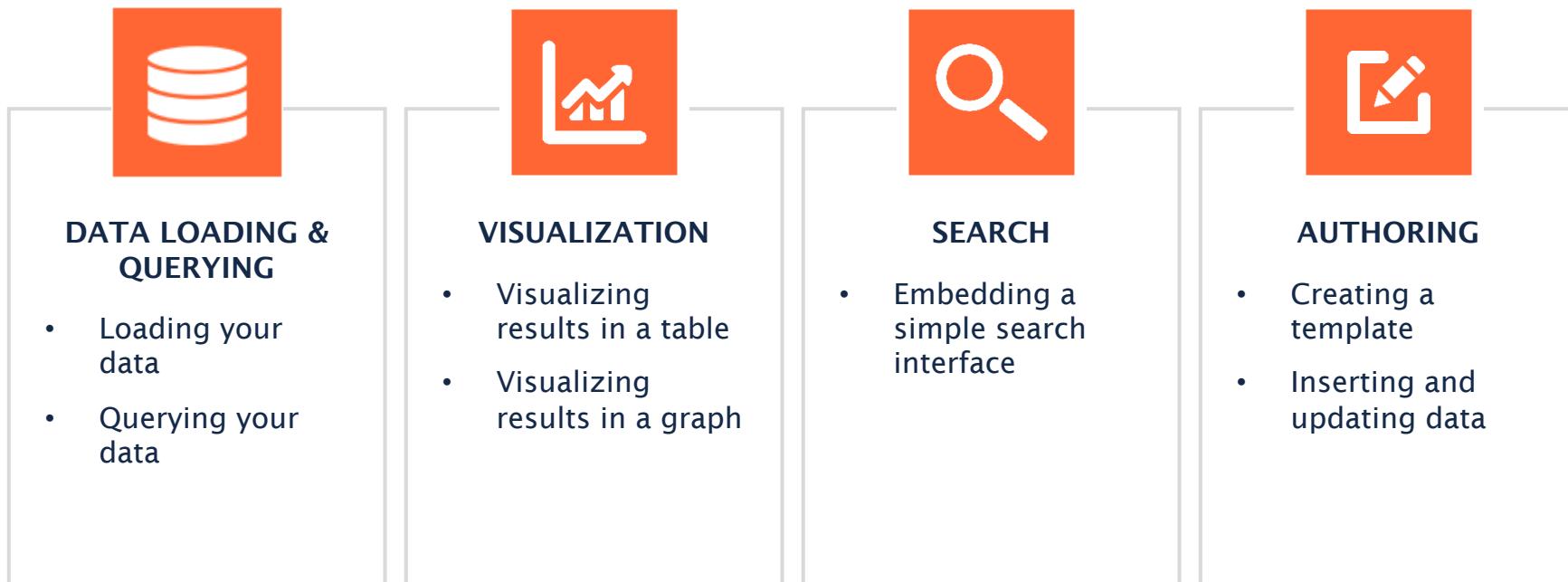




# metaphacts

Smart Data. Smart Apps. Smart Decisions.

## Hands-on Exercises

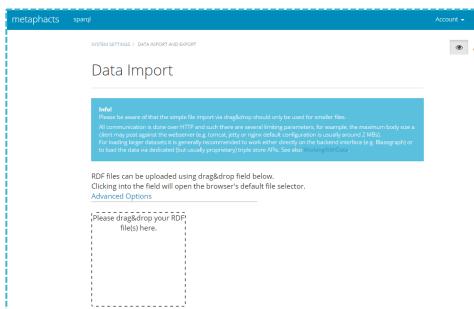


- Download metaphactory (or copy from USB stick)  
<http://www.knowledgegraph.info/>
- Follow README, start metaphactory  
start.sh / start.bat
- Open start page  
<http://localhost:10214>
- Follow “Getting started tutorial”  
<http://localhost:10214/resource/Help:Tutorial>
- Have fun and ask questions ;-)

# Data Loading & Querying

1

Load data into the store via the data import and export administration page



The screenshot shows the 'Data Import' section of the metaphacts interface. It includes a note about the limitations of the drag-and-drop feature, a file input field for RDF files, and an 'Advanced Options' link.

Data Export & Deletion

The repository contains 2 distinct named graphs.			
Named Graph	Triples	Export	Delete
file:/sample.ttl-21-01-2017-10-19-52	17		
http://www.bigdata.com/rdf#nullGraph	1		

2

Query the data via the SPARQL endpoint.

E.g.: issue a query for all statements made about Bob as a subject



The screenshot shows the SPARQL endpoint interface with a query editor containing the following code:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE { <http://example.org/bob#me> ?predicate ?object
}
4 LIMIT 10
```

A 'Execute' button is at the bottom.

3

Visualize results in a table

predicate	object
http://schema.org/date	1990-07-24
http://schema.org/genre	http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg
http://schema.org/label	Bob
http://schema.org/name	http://example.org/bob
http://schema.org/note	http://wikidata.entity/012418

... or as raw data

Raw Response	Table	JSON										
<pre>i = ("head": {"vars": [{"predicate": "object"}]}, "results": {"bindings": [{"predicate": "object": {"type": "uri", "value": "http://schema.org/birthDate"}, "object": {"type": "literal", "value": "1990-07-24"}, "label": "1990-07-24"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/genre"}, "object": {"type": "uri", "value": "http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg"}, "label": "http://schema.org/genre"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/name"}, "object": {"type": "uri", "value": "http://example.org/bob"}, "label": "Bob"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/note"}, "object": {"type": "uri", "value": "http://wikidata.entity/012418"}, "label": "http://wikidata.entity/012418"}])</pre>	<table border="1"><thead><tr><th>predicate</th><th>object</th></tr></thead><tbody><tr><td>http://schema.org/date</td><td>1990-07-24</td></tr><tr><td>http://schema.org/genre</td><td>http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg</td></tr><tr><td>http://schema.org/name</td><td>http://example.org/bob</td></tr><tr><td>http://schema.org/note</td><td>http://wikidata.entity/012418</td></tr></tbody></table>	predicate	object	http://schema.org/date	1990-07-24	http://schema.org/genre	http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg	http://schema.org/name	http://example.org/bob	http://schema.org/note	http://wikidata.entity/012418	<pre>i = ("head": {"vars": [{"predicate": "object"}]}, "results": {"bindings": [{"predicate": "object": {"type": "uri", "value": "http://schema.org/birthDate"}, "object": {"type": "literal", "value": "1990-07-24"}, "label": "1990-07-24"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/genre"}, "object": {"type": "uri", "value": "http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg"}, "label": "http://schema.org/genre"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/name"}, "object": {"type": "uri", "value": "http://example.org/bob"}, "label": "Bob"}, {"predicate": "object": {"type": "uri", "value": "http://schema.org/note"}, "object": {"type": "uri", "value": "http://wikidata.entity/012418"}, "label": "http://wikidata.entity/012418"}])</pre>
predicate	object											
http://schema.org/date	1990-07-24											
http://schema.org/genre	http://upload.wikimedia.org/wikipedia/commons/thumb/1/12/user_icon_2.svg/20px-user_icon_2.svg											
http://schema.org/name	http://example.org/bob											
http://schema.org/note	http://wikidata.entity/012418											

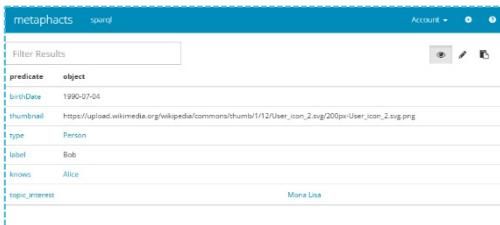
# Visualizing Results in a Table

1

Embed 'semantic-table' component

```
<semantic-table config='{
  "query": "SELECT * WHERE {
<http://example.org/bob#me>
?predicate ?object }"
  }'>
</semantic-table>
```

to visualize previous query as a table in a page



The screenshot shows a table with the following data:

predicate	object
birthDate	1990-07-04
thumbnail	<a href="https://upload.wikimedia.org/wikipedia/commons/thumb/1/12/User_icon_2.svg/200px-User_icon_2.svg.png">https://upload.wikimedia.org/wikipedia/commons/thumb/1/12/User_icon_2.svg/200px-User_icon_2.svg.png</a>
type	Person
label	Bob
knows	Alice
topic_interest	<a href="#">More Lisa</a>

2

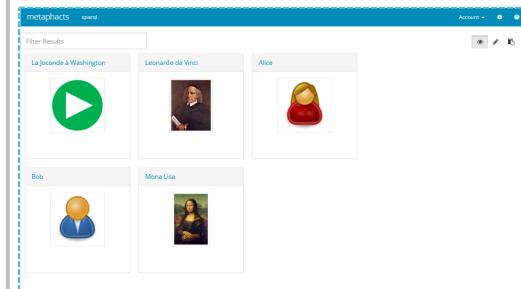
Customize the query to embed thumbnail images in the result visualization

```
SELECT ?uri ?label
?thumbnail WHERE { ?uri
rdfs:label ?label;
<http://schema.org.thumbnail> ?thumbnail }
```

Use **tupleTemplate** to define a template for displaying the new table

3

Visualize results in a table displaying thumbnails as images, the labels of the resources as captions, and links to the individual resource pages



# Visualizing Results in a Graph

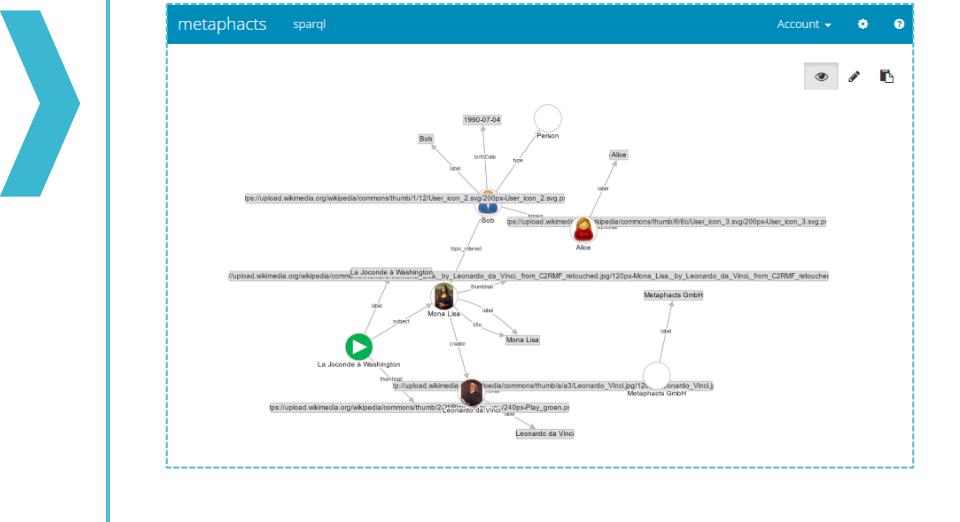
1

Embed 'semantic-graph'  
component

```
<semantic-graph  
query="CONSTRUCT WHERE { ?s ?p ?o  
} ">  
/</semantic-graph>
```

2

Visualize results in a graph



# Embedding a Simple Search Interface

1

Embed 'semantic-simple-search' into the page

```
<semantic-simple-search
config='{
    "query":'
        SELECT ?uri ?label
    WHERE {
        FILTER REGEX(?label,
        \"?token\", \"i\")
            ?uri rdfs:label
        ?label
        } LIMIT 10
    ",
    "searchTermVariable": "token",
    "resourceSelection": {

        "resourceBindingName": "uri",
        "template": "<span
style=\"color: blue;\""
        title=\"{{uri.value}}\">{{label.
value}}</span>"
        },
        "inputPlaceholder": "Search for
something e.g. \"Bob\""
    }
}>
</semantic-simple-search>
```

2

Rendered component is displayed and can be used right away

Search for something e.g. "Bob"

3

Autosuggestions are dynamically computed based on query and user input



# Creating a Template

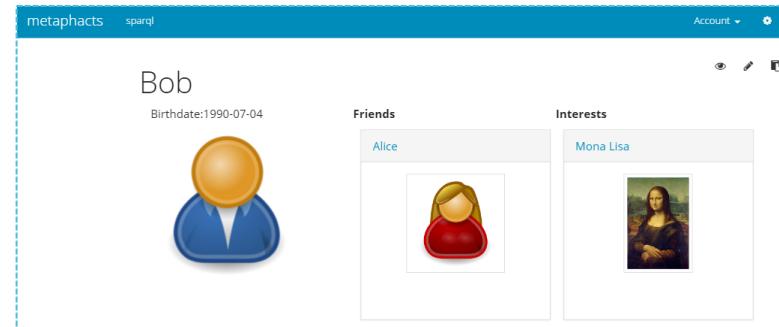
1

2

Use the templating mechanism to create a template for the resource type 'Person', to display:

- the person's name
- an image, if available
- his interests
- his friendship relationship

Visualize the result on Bob's instance page



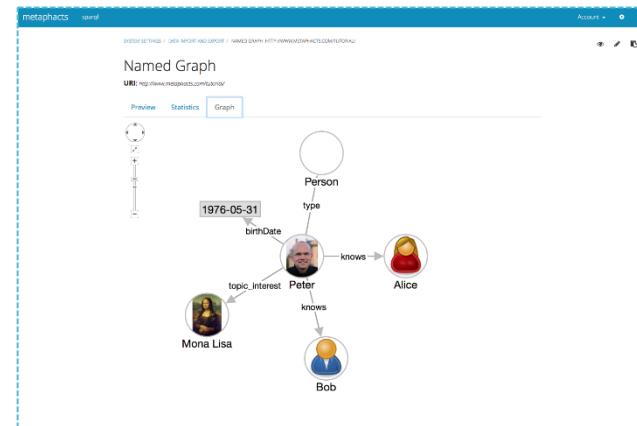
1

2

Use a SPARQL UPDATE operation  
against the SPARQL endpoint to  
create and add new instance data

- the person's name
- an image, if available
- his interests
- his friendship relationship

Visualize the result



- **Knowledge graphs** as a flexible model for data integration and knowledge representation
- Standards for “semantic” knowledge graphs
  - RDF as graph-based data model
  - OWL as expressive ontology language
  - SKOS for taxonomic knowledge
  - SPARQL as query language
- **Application areas**
  - Open knowledge graphs, e.g. Wikidata
  - Cultural Heritage
  - Life Sciences
  - And many more
- Get started with the metaphactory Knowledge Graph platform today!

Get in Touch!



# metaphacts GmbH

Industriestraße 41

69190 Walldorf

Germany

p +49 6227 6989965

m +49 157 50152441

e [info@metaphacts.com](mailto:info@metaphacts.com)

 [@metaphacts](https://twitter.com/metaphacts)

[www.metaphacts.com](http://www.metaphacts.com)