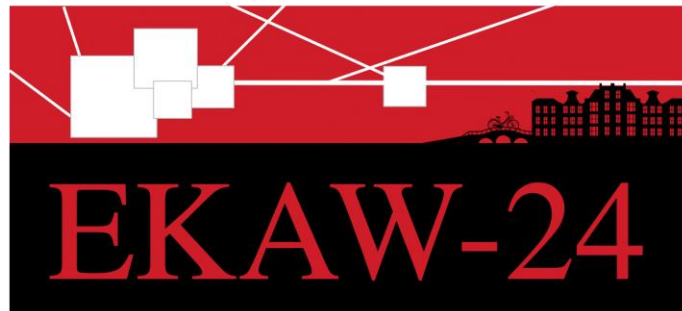




metaphacts

Smart Data. Smart Apps. Smart Decisions.

Tutorial: Semantic Knowledge Modeling - Ontologies & Vocabularies



Who we are



Dr. Peter Haase

Founder & Chief Scientific
Officer



Irina Schmidt

Chief Marketing Officer



OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)

W3C Recommendation 11 December 2012

This version:

<http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>

Latest version (series 2):

<http://www.w3.org/TR/owl2-syntax/>

Latest Recommendation:

<http://www.w3.org/TR/owl-syntax>

Previous version:

<http://www.w3.org/TR/2012/PER-owl2-syntax-20121018/>

Editors:

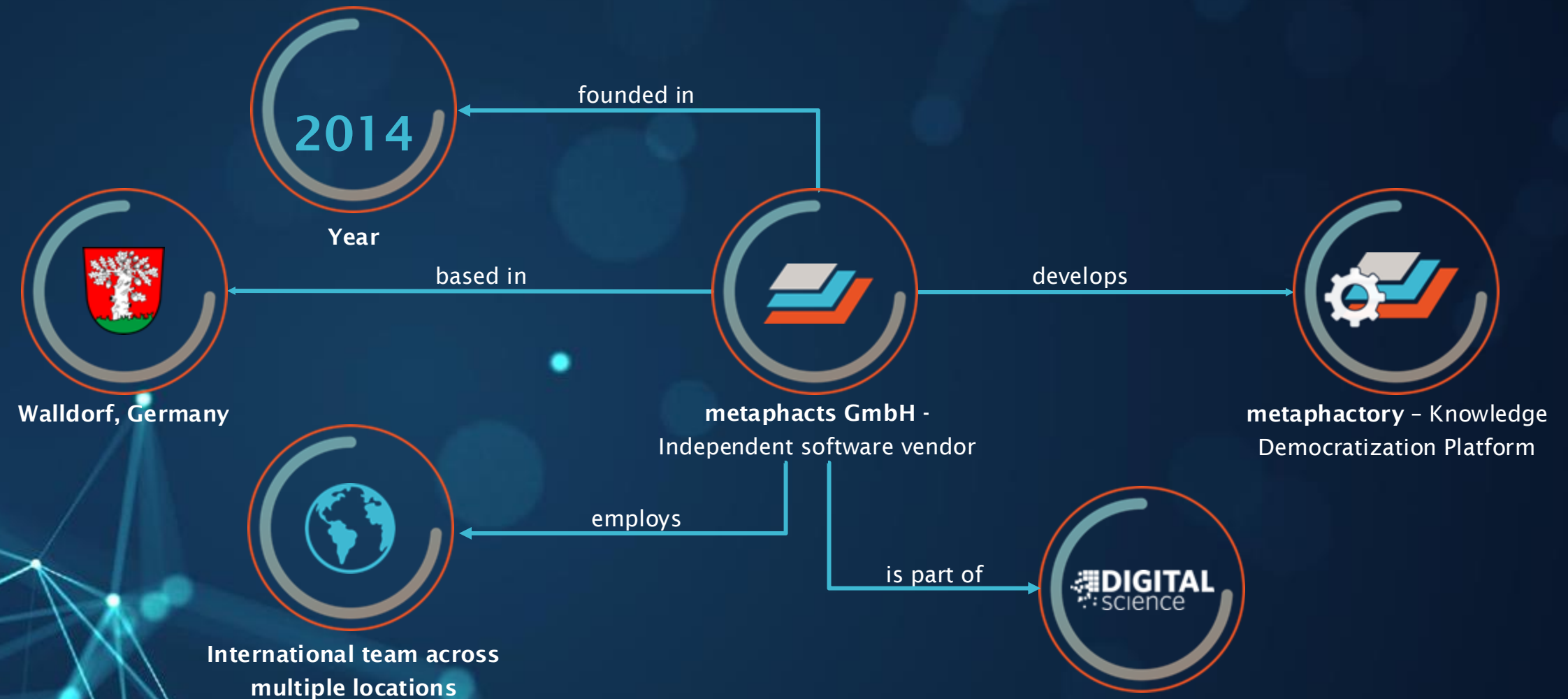
[Boris Motik](#), University of Oxford
 Peter F. Patel-Schneider, Nuance Communications
[Bijan Parsia](#), University of Manchester

Contributors: (in alphabetical order)

Conrad Bock, National Institute of Standards and Technology (NIST)
[Achille Fokoue](#), IBM Corporation
[Peter Haase](#), FZI Research Center for Information Technology
[Rinke Hoekstra](#), University of Amsterdam
[Ian Horrocks](#), University of Oxford
[Alan Ruttenberg](#), Science Commons (Creative Commons)
[Uli Sattler](#), University of Manchester
[Michael Smith](#), Clark & Parsia

metaphacts GmbH - Knowledge Democratization and AI

Company Snapshot



- Introduction
- What is semantic knowledge modeling – The metaphacts approach
- Visual modeling basics – Hands-on tutorial
- *Coffee break*
- Examples and experiences from industry
- Model-driven app building
- Advanced considerations in visual modelling: collaborative workflows, versioning
- Conclusion and Q&A

- The tutorial is an interactive learning experience with hands-on exercises
- Trainer will show live examples and guide participants through them
- Participants have their individual metaphactory setup for hands-on exercises
- Slides and exercise material will be made available to the participants for their personal use

Watch Out For...

■ “Documentation”

- Links content being presented to built-in metaphactory ‘Help’ sections for more technical details



Walkthrough: <https://#yourmetaphactory#.tutorial.metaphacts.cloud>

Documentation: <http://help.metaphacts.com/resource/Help:Start>

■ “Walkthrough”

- Links Example being presented to reusable and complete pre-defined code
- Description of planned Hands-on exercises
- Code Snippets to carry out exercise, and entire solutions





I'm Ready!

Raise a hand in Checkpoints to signify **readiness/completion**, e.g. ready to proceed, task complete, etc.



Speak out to flag **issues**, ask **questions**, request **support** with exercises, etc.



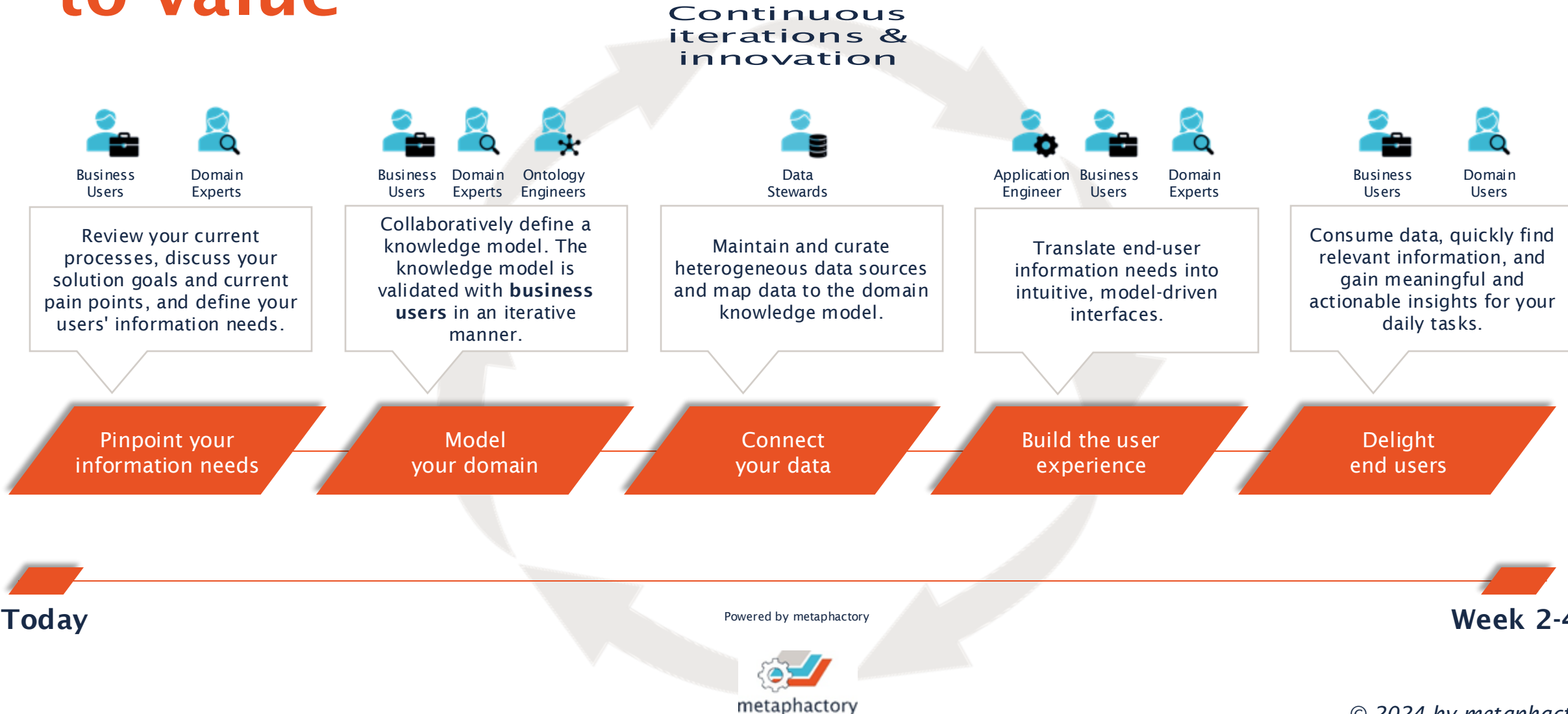
metaphacts

Smart Data. Smart Apps. Smart Decisions.

Semantic Modeling – The metaphacts Approach

Rapid application building – Fast path to value

Continuous iterations & innovation



Today

Powered by metaphactory



Week 2-4

metaphactory - Knowledge Democratization Platform



KNOWLEDGE GRAPH MANAGEMENT

Visual authoring, visualization, versioning & cataloging of ontologies, vocabularies, datasets & queries
Data validation, provenance & lineage

END-USER ORIENTED INTERACTION

Abstracted view
One-stop knowledge hub
Intuitive UI for knowledge discovery, exploration, analytics, editing

KNOWLEDGE GRAPH APPLICATION BUILDING

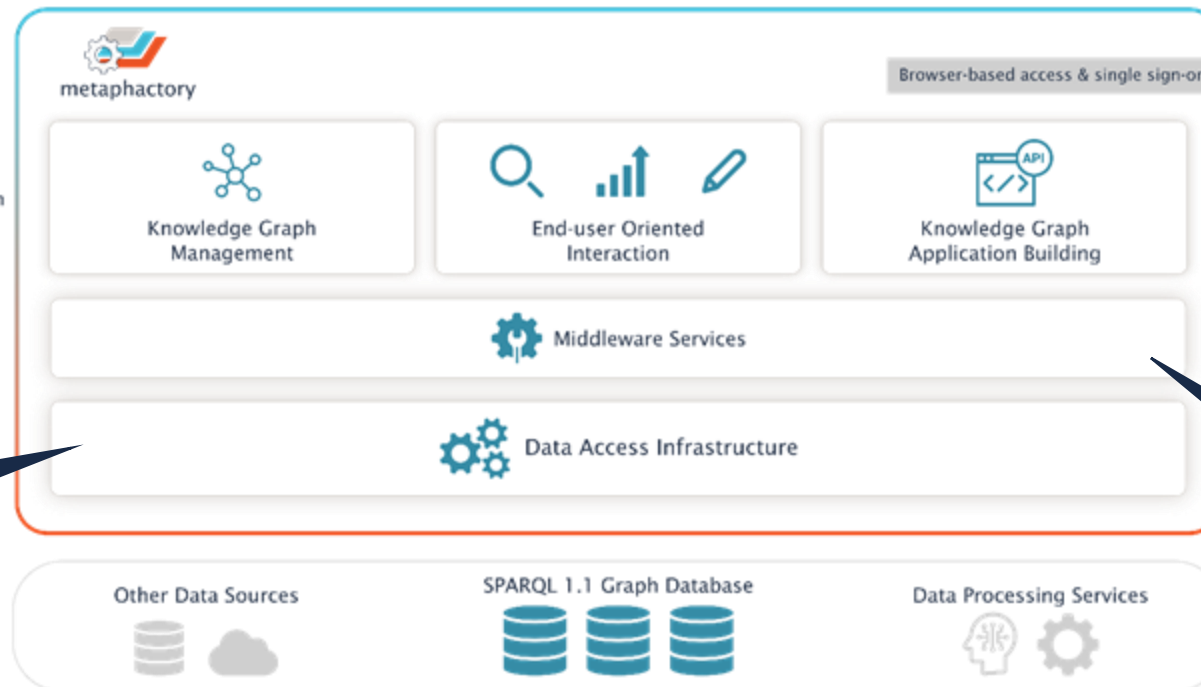
Low-code platform
Powerful template engine
Large library of Web components
Easy customization

DATA INTEGRATION & FEDERATION

Unified view on distributed and heterogenous data sources: graph databases, relational databases, REST APIs, machine learning algorithms
Transparent SPARQL federation

MIDDLEWARE SERVICES

Dynamic data-driven REST APIs based on queries
Role-based access control
Lookup & Reconciliation
Tableau - Web Data Connector Endpoint



✓ Run anywhere



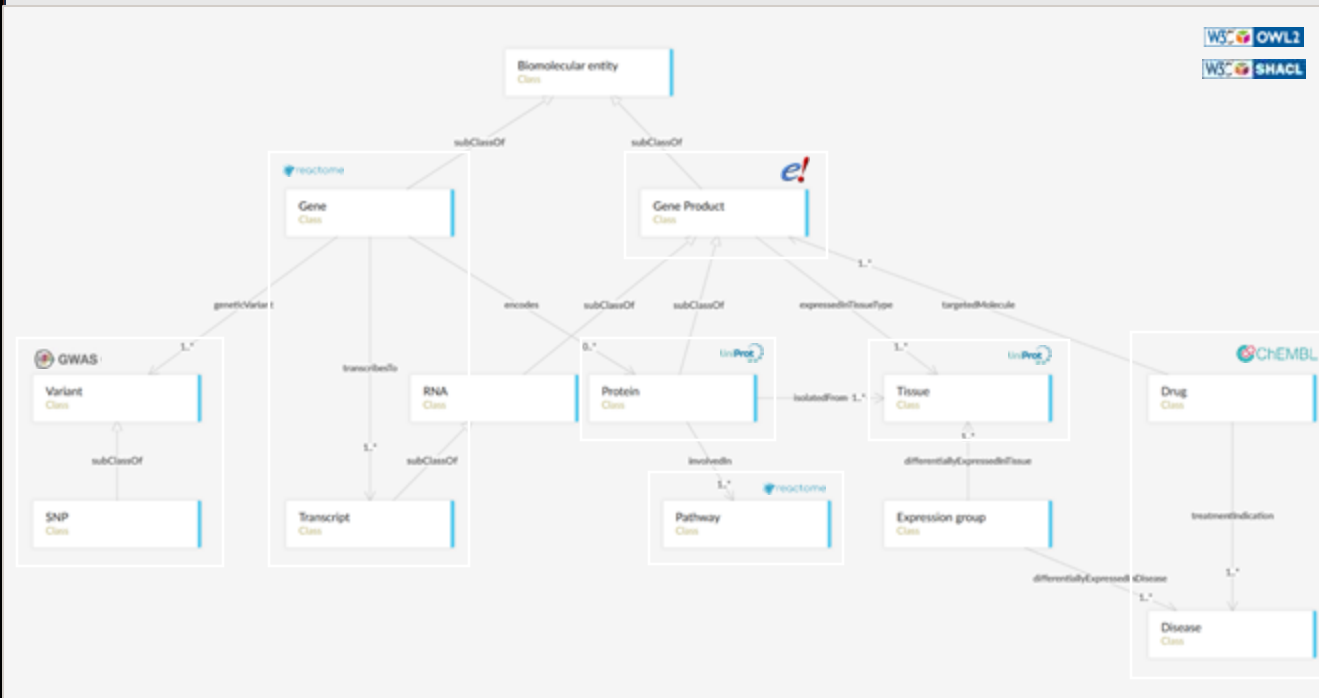
Building the knowledge graph

 Knowledge Graph Engineer
 Knowledge Steward (Domain Expert)
 Data Steward

Visual Ontology Modeling

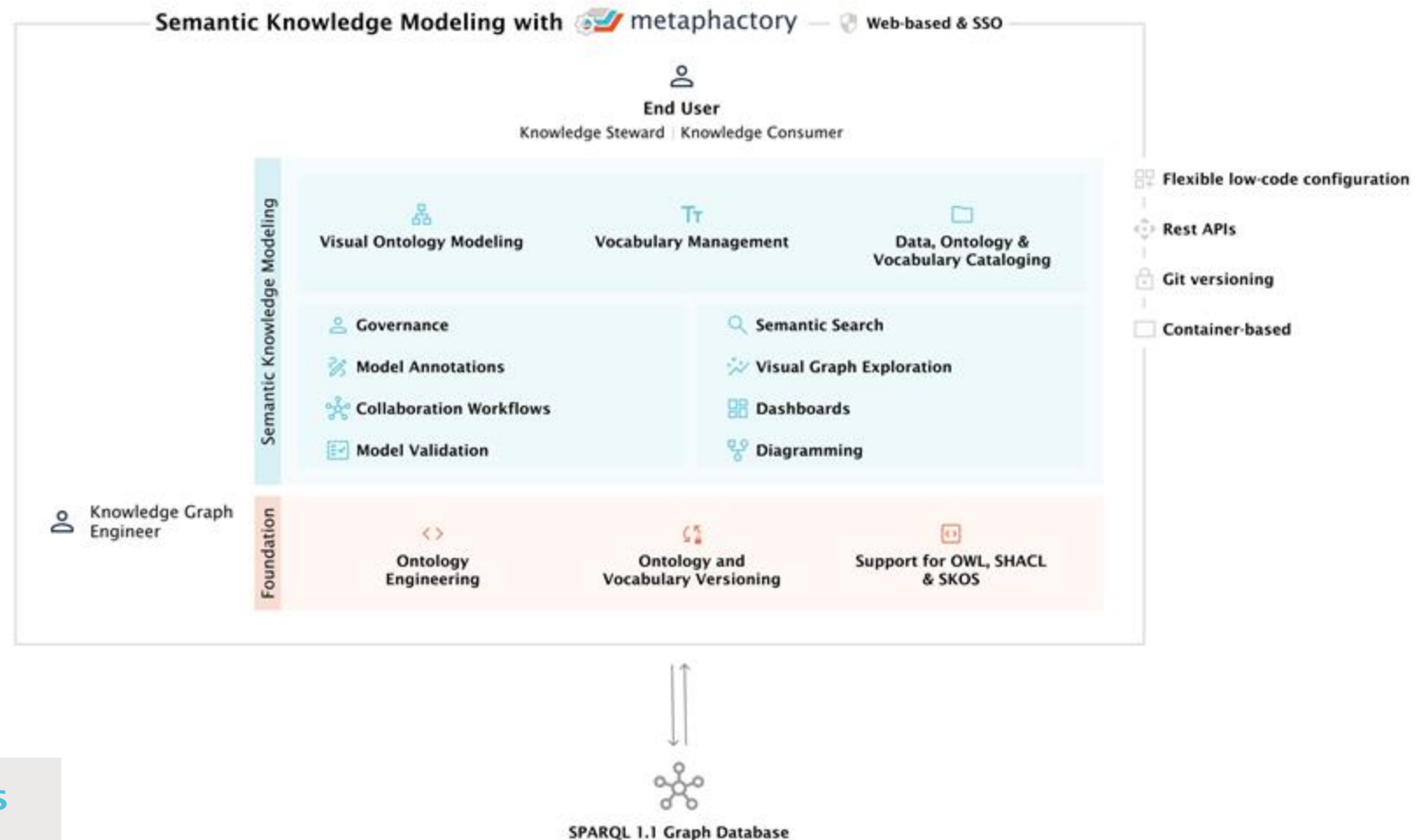
- All stakeholders are empowered to actively participate in the modeling process
- Agile processes for ontology design, implementation, documentation and governance

Example Ontology from the Life Sciences Domain



Semantic Knowledge Modeling

- » Capture domain knowledge & expertise in **reusable & extensible semantic models**
- » Boost data literacy by enabling **domain experts** & business users to contribute to the modeling process
- » Support **expert users** with advanced features for ontology engineering & governance workflows

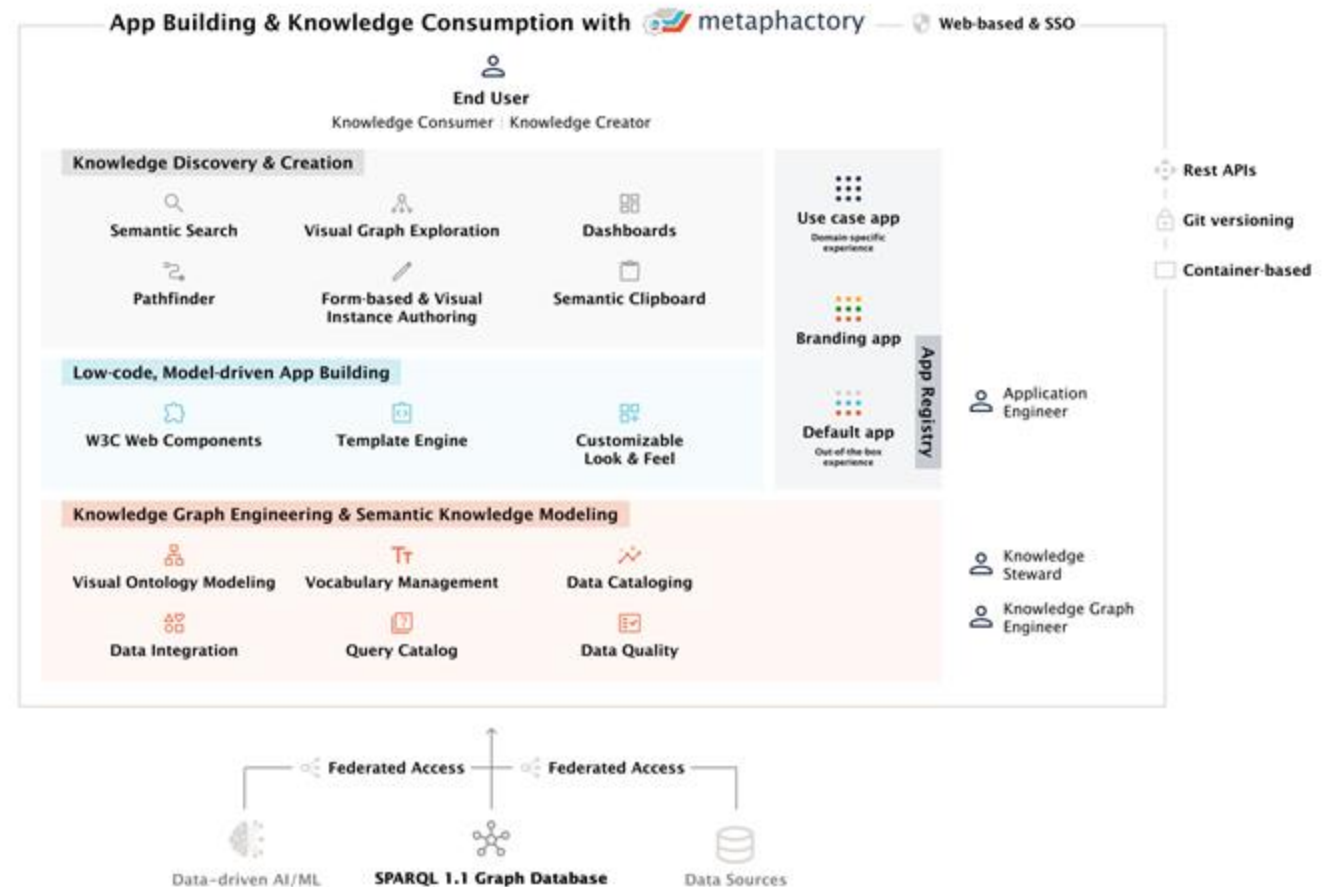


Based on open standards



Model-driven App Building & Knowledge Consumption

- » Build **use-case specific knowledge graph apps** that meet enterprise requirements
- » Drive knowledge democratization by empowering **end users** to discover, consume & create knowledge & insights in a self-service manner
- » Support **application engineers** with a model-driven, low-code approach for app building

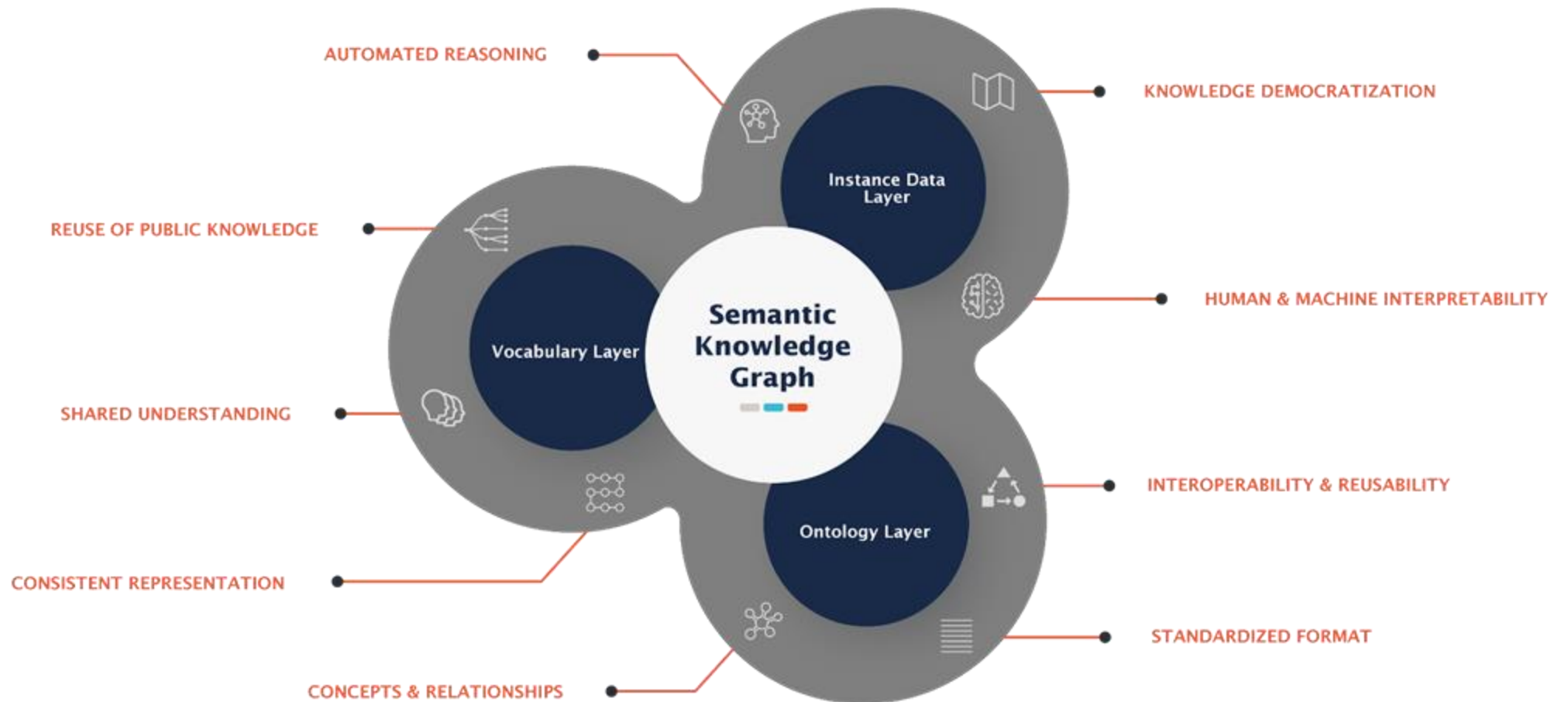


Based on open standards



The three layers of a semantic knowledge graph

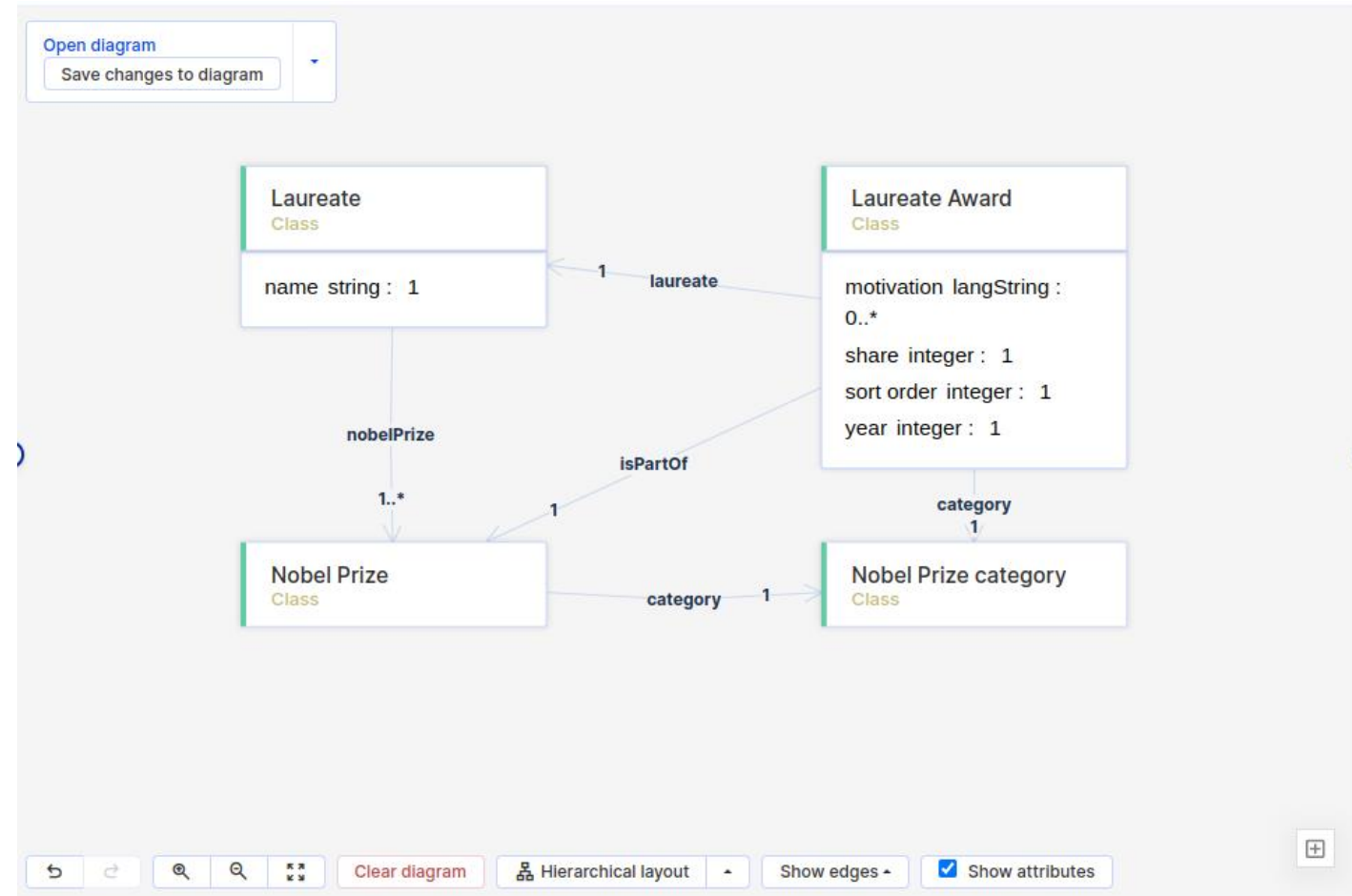
F indable A ccessible I nteroperable R eusable





- » Interlink ontologies & vocabularies to support reuse while separating management & governance tasks
- » Improve stakeholder communication, asset documentation & governance
- » Enable model-driven applications with e.g., auto-suggestions in semantic forms, runtime validation of user interaction, hierarchical facets in search, etc.
- » Ensure data quality by running checks & validations against business logic

- **Classes**
 - represent the abstract or concrete categories, types, or groups of entities within the domain of discourse.
- **Attributes**
 - are specific properties or characteristics assigned to an instance (individual) of a class.
- **Relations**
 - define how entities or classes are related to each other within the ontology.



- **Web Ontology Language (OWL)** provides more ontological constructs including:
 - distinction between OWL ObjectProperties (resources as values) and OWL DatatypeProperties (literals as values).
 - Able to define a named graph as an ontology, e.g.
 - `<http://data.nobelprize.org/terms/> a owl:Ontology;
dcterm:title "Linked Nobelprize Ontology".`
- **SHACL** is a language to describing and validating RDF graphs against a set of conditions through shapes:
 - **Node shapes**: constraints about a given focus (target) node.
 - **Property shapes**: constraints about a given property and its values for the focus node.

Complexity **hidden** behind
visual editing in
metaphactory!

Documentation: <https://www.w3.org/TR/owl2-overview/>
<https://www.w3.org/TR/shacl/>



Class definitions:

prefixes omitted

```
:Laureate a owl:Class ;  
    rdfs:label "Laureate" .
```

```
:LaureateAward a owl:Class ;  
    rdfs:label "Laureate Award" ;  
    rdfs:subClassOf dbpedia:Award .
```

Property definitions:

```
:isPartOf a owl:ObjectProperty;  
    rdfs:label "is part of" .
```

```
:name a owl:DatatypeProperty;  
    rdfs:label "name" .
```

SHACL definitions:

```
# prefixes omitted
:LaureateShape a sh:NodeShape ;
sh:targetClass :Laureate ;
```

Node shape

```
sh:property [
    sh:path :nobelPrize;
    sh:class :NobelPrize;
    sh:minCount 1 ;
] ;
sh:property [
    sh:path :name;
    sh:datatype xsd:string;
    sh:minCount 1 ;
    sh:maxCount 1 ;
].
```

Property shapes

Knowledge Graph Assets

A. Ontologies

- the model of your domain
- the schema/structure of the data
- Modeling language: OWL + SHACL

B. Vocabularies

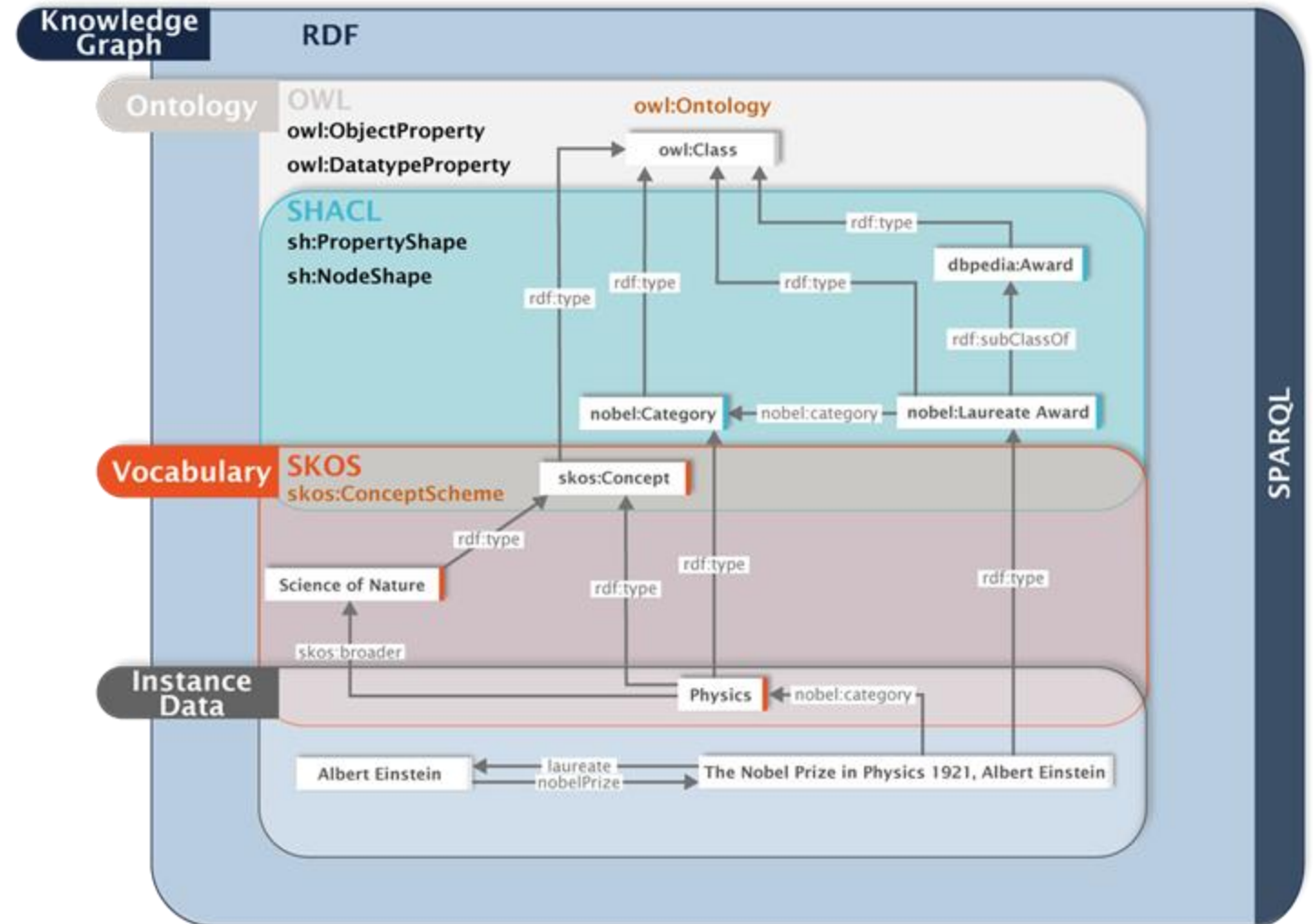
- Extends structure introduced by ontologies
- Definitions of relevant terms / concepts for a given domain
- Modeling language: SKOS

C. Instance Data

- Data about actual entities in the domain
- Concrete "ground level" information: particular people, specific products, or individual events with their associated attributes and relationships

• Notes:

- Knowledge Graph = collective assets
- SPARQL is the entry point to the Knowledge Graph as a whole (Read/Write at all levels)
- Vocabularies considered as Instance Data with special characteristics & function
- Vocabulary definition is very subjective, use-case dependent
- Blurred lines / overlap in asset role within the KG (what constitutes the model/structure VS what constitutes data)



- **Simple Knowledge Organization System (SKOS)** provides vocabulary definitions:

- Concept schemes, informal hierarchies and association networks

```
nobel:Physics a skos:Concept .
```

```
nobel:Chemistry a skos:Concept .
```

- **SKOS** allows for the definition of term hierarchies (**skos:hasTopConcept**, **skos:broader**, and **skos:narrower**) e.g.

```
nobel:vocabulary a skos:ConceptScheme ;
```

```
skos:hasTopConcept nobel:Categories.
```

```
nobel:Physics a skos:Concept ;
```

```
skos:inScheme nobel:Categories ;
```

```
skos:broader nobel:Science-of-Nature .
```

Complexity **hidden** behind
visual editing in
metaphactory!

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



Contents

1 Introduction	1
1.1 What is semantic modeling and how does it help?	1
1.2 What is involved in semantic modeling?	2
1.3 How does the model relate to data?	3
2 Getting started with modeling	4
2.1 Basic elements in the semantic modeling language	4
2.1.1 Class: How do you define the things you model?	4
2.1.2 Attribute: How do you describe your classes?	6
2.1.3 Relation: How do you relate your classes to one another?	7
2.2 Connecting elements together to form an ontology	9
3 Methodology and design principles for ontology modeling	11
3.1 Getting started: Modeling for the first time	11
3.1.1 Planning the scope of your ontology	11
3.1.2 Implementing the initial ontology design	12
3.1.3 Checking your work with competency questions	14
3.1.4 Documenting your work so far	17
3.1.5 Refining your ontology	18
3.2 Next steps: Advanced considerations	18
3.2.1 Modular development	18
3.2.2 Ontology reuse	20
4 Enhancing your modeling capabilities	21
4.1 Vocabulary	21
5 Conclusion and outlook	23
Quick Reference	i
Naming Tips	i
Modeling Checklists	iii

Class Naming Tips

- A class name is usually singular (e.g., **Person** not People).
- Class names are usually capitalized (e.g., **Organization**).
- Classes generally use nouns or noun phrases (e.g., **Organization** or **Software Business**).
- For classes named with noun phrases, you typically use title case (e.g., **Software Business**).

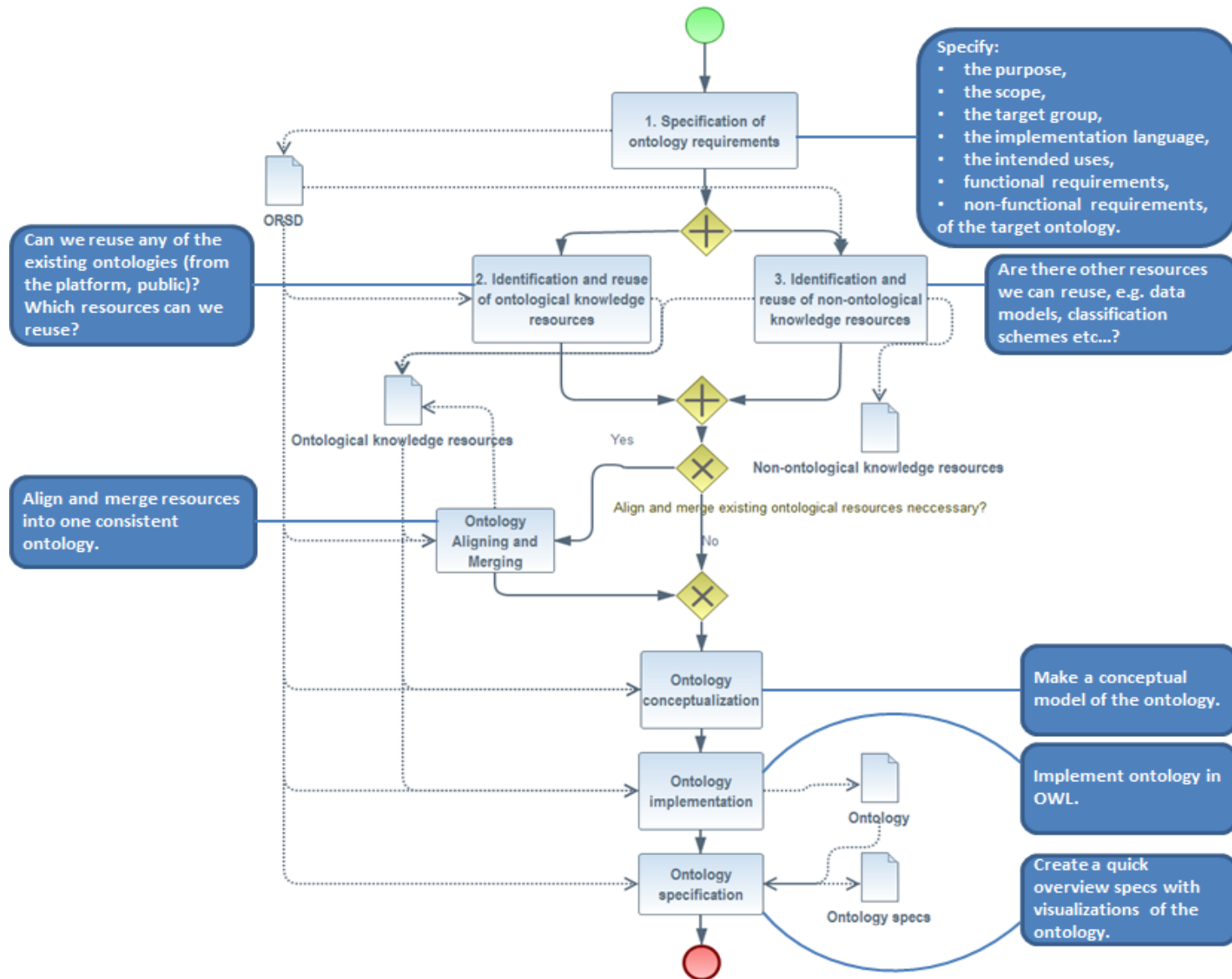
Attribute Naming Tips

- Attributes are typically named after the internal properties of a class (e.g., length, birth date, color, etc.).
- Attribute names are usually lowercase.
- Grammatically, attributes are usually either:
 - Nouns (e.g., length) - most common
 - Verbs/verb phrases (e.g., has length of) - less common
- When a verb is used for an attribute name, it should be present tense (e.g., has, not had).
- Multi-word attributes should use spaces to enable human-readability, although some prefer camel-case.
- No matter which style you choose, you should always be consistent!

Relation Naming Tips

- Relations use mostly the same naming conventions as attributes.
 - The exception is that grammatically, you usually want verbs and verb phrases for relations rather than nouns (e.g., “A **Person** knows a **Person**”, “A **Person** lives in a **House**”, etc.).
- A relation should derive its name from the intended connection between the two classes and indicate the semantics.
 - This allows us to distinguish between different connections between the same classes (e.g. “A **Person** eats **Food**” vs “A **Person** loves **Food**”).

Ontology Development Methodology – Inspired by NeOn





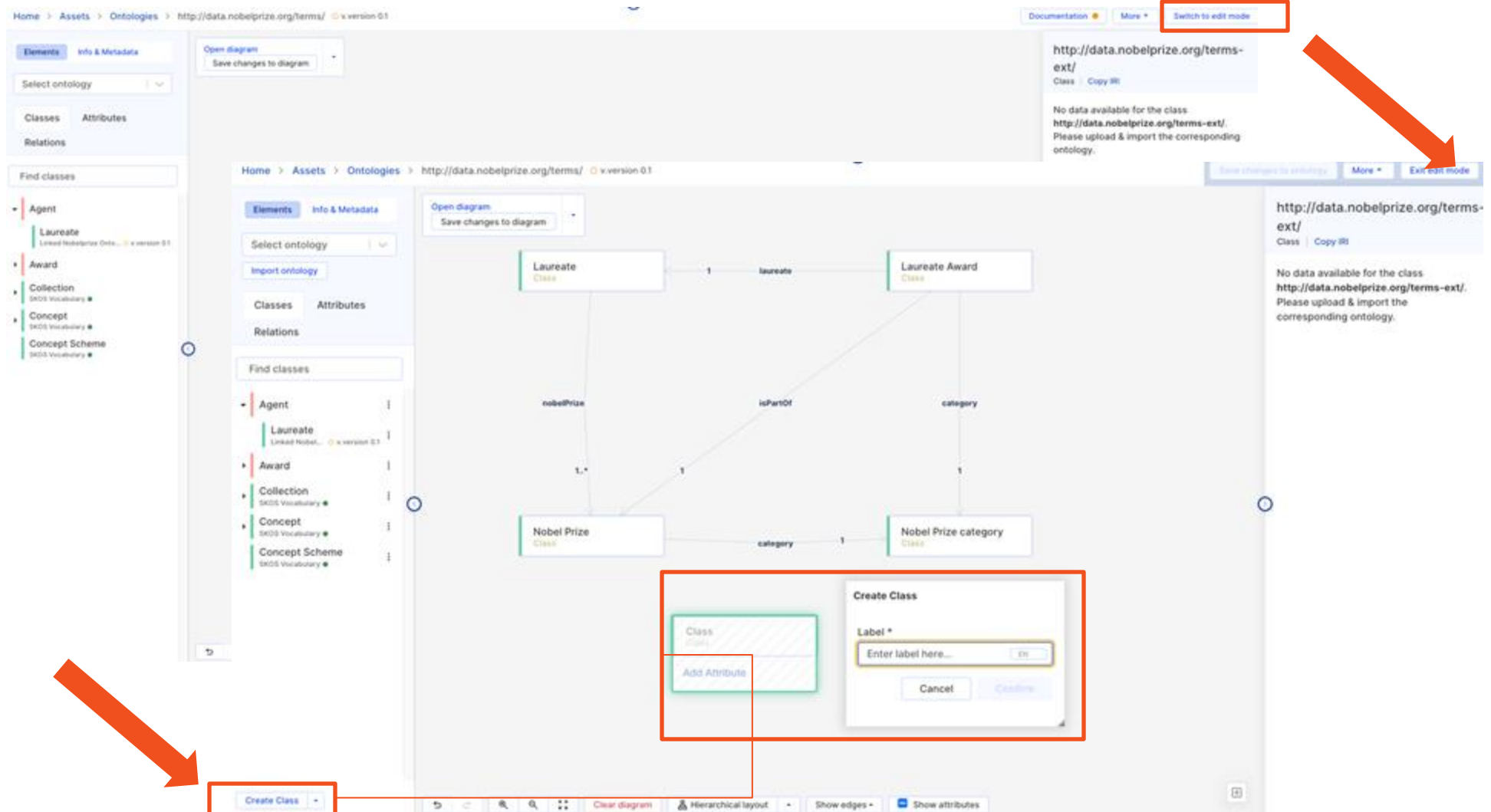
metaphacts

Smart Data. Smart Apps. Smart Decisions.

Visual ontology modeling

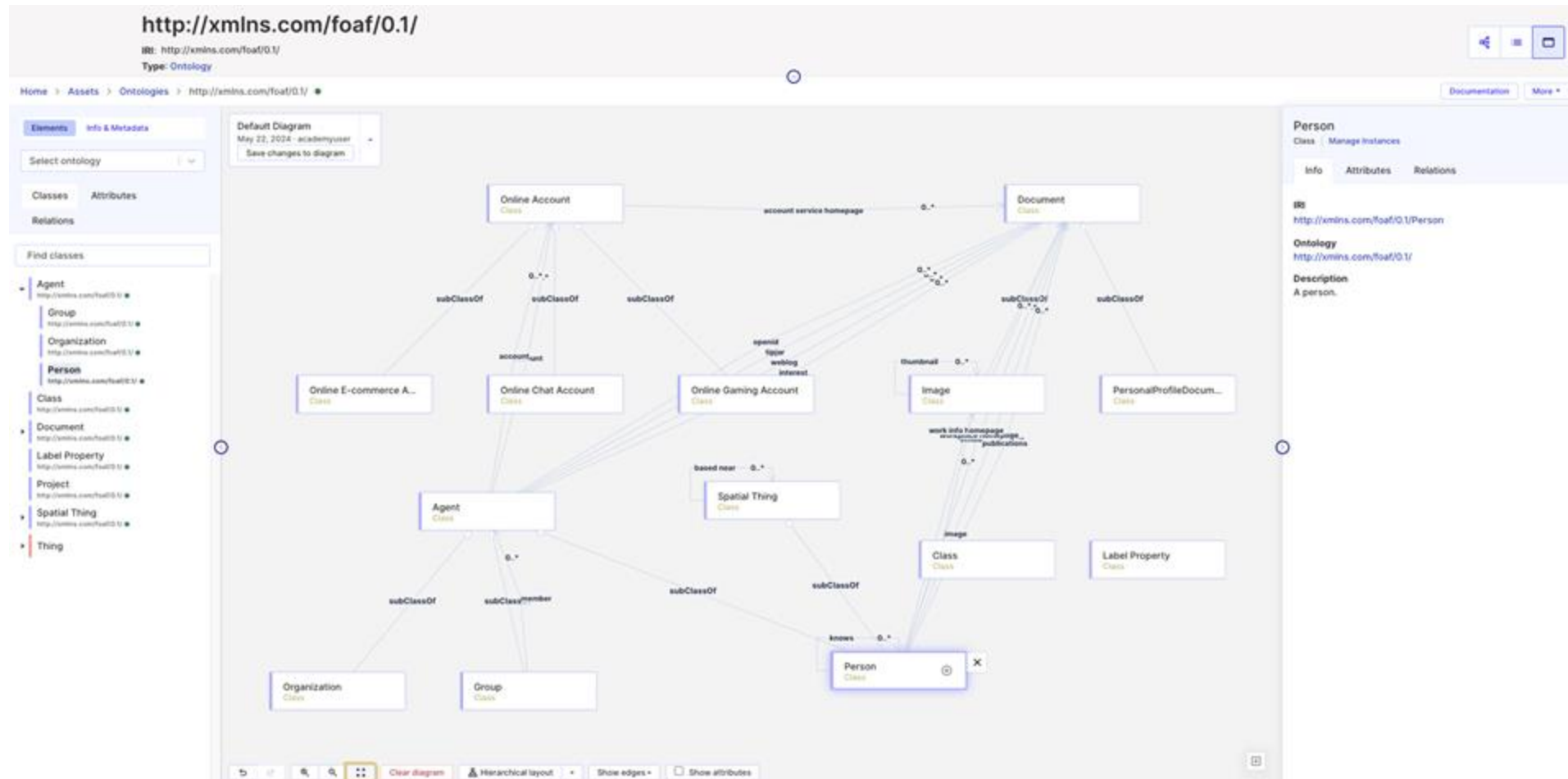
- Explore catalog
- Create a new ontology
- Create classes/relations/attributes
- Reuse
- Vocabulary restriction

Toggling Edit Mode



Visual Vocabulary & Ontology Management

- Visual diagrams as default means to interact with vocabularies & ontologies
 - exploration & documentation
 - modification



Vocabulary & Ontology Management: Basics

Home > Assets > Ontologies

Edit Page



Ontologies

Quality check 

This is the entry page to the ontology repository. Here you can create, import and manage ontologies.

Import

Create

create or import

foaf

Created by anyone

Last modified by anyone

Status 4

Clear All

Status:

In development

In review

Ready to be published

Published

view ontology

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

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

15 classes

27 attributes

33 relations


ontology summary

status


Create version,
Export, delete,
copy, RDF
serialized, validate
database, save to
git, set status,
Role Assignment

Vocabulary & Ontology Management: Basics

Home > Assets > Vocabularies Edit Page



Vocabularies

Quality check 

Import ▾Create

This is the entry page to the vocabulary repository. Here you can create, import and manage vocabularies.

Created by anyone ▾

Last modified by anyone ▾

Data theme

<http://publications.europa.eu/resource/authority/data-theme>

16 terms

14 top-level terms

0 collections

Last modified 4 months ago by academyuser

⋮

view vocabulary

vocabulary summary

create or import

download, delete,
Save to git,
Role assignment

© metaphacts GmbH. For personal use only. Do not redistribute.

Versioning Info as Metadata

Linked Nobelprize Ontology
<http://data.nobelprize.org/terms/>

Title *
Linked Nobelprize Ontology EN Add value

Label *
Linked Nobelprize Ontology EN Add value

Description
This ontology specifies the classes and properties to describe the Nobel Prize Linked Data dataset.
The Nobel Prize Linked Data dataset contains the authoritative information about Nobel Prizes and Nobel Laureates since 1901.
Please refer to <https://data.nobelprize.org/specification/> for details. EN Add value

Base Element Namespace
<http://data.nobelprize.org/terms-ext/> Add value

Version Info
version 0.1 Add value

Version IRI
Add value

Created
2013-02-12 23:00:00 Add value

Creator
nobelprize.org Add value

Contributor
Johannes Trame <jt@metaphacts.com> Add value
Jeen Broekstra <jb@metaphacts.com> Add value
sl@metaphacts.onmicrosoft.com Add value

Imported ontologies
<http://dbpedia.org/ontology/> Add value
<http://www.w3.org/2004/02/skos/core> Add value
<http://xmins.com/foaf/0.1/> Add value

License
Add value

ⓘ You must reload the page to apply the changes on imported ontologies

- Title (dcterms:title)
- Label (rdfs:label)
- Description (dcterms:description)
- Base Element Namespace (IRI) (vaem:namespace)
- Version info (owl:versionInfo)
- Version IRI (owl:versionIRI)
- Created (dcterms:created)
- Creator (dcterms:creator)
- Contributor (dcterms:contributor)
- Imported ontologies (owl:imports)
- License (dcterms:license)

Ontology VS Diagrams: What to keep in mind!

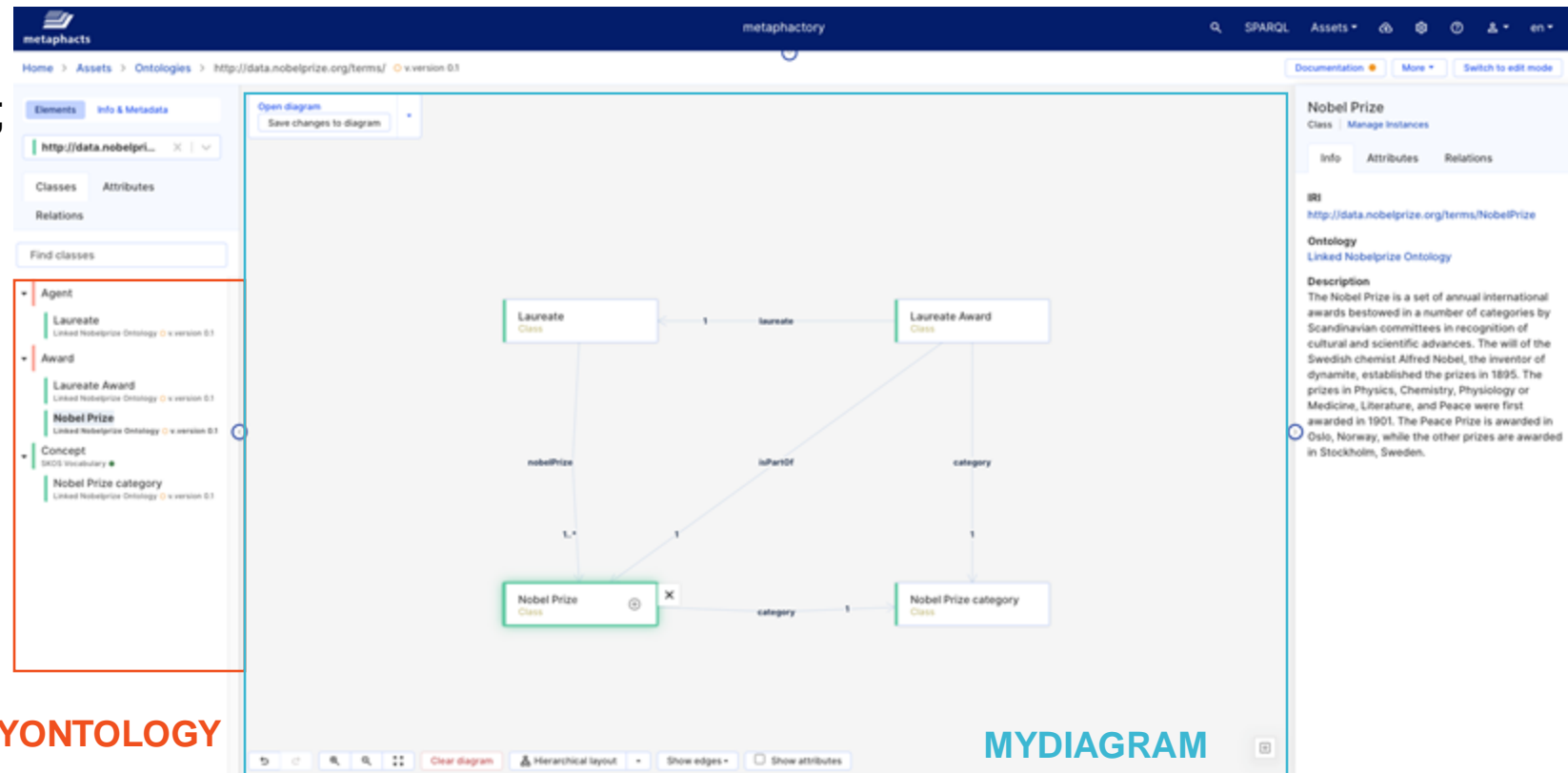
MYDIAGRAM – saved as a Diagram

- Apart from default diagrams; unlimited amount of diagrams can be created as views
- Each ontology can have 1+

MYONTOLOGY – model corresponding to MYDIAGRAM saved as Ontology

- elements inherited from other ontologies NOT included, only references
- Exactly 1

MYONTOLOGY

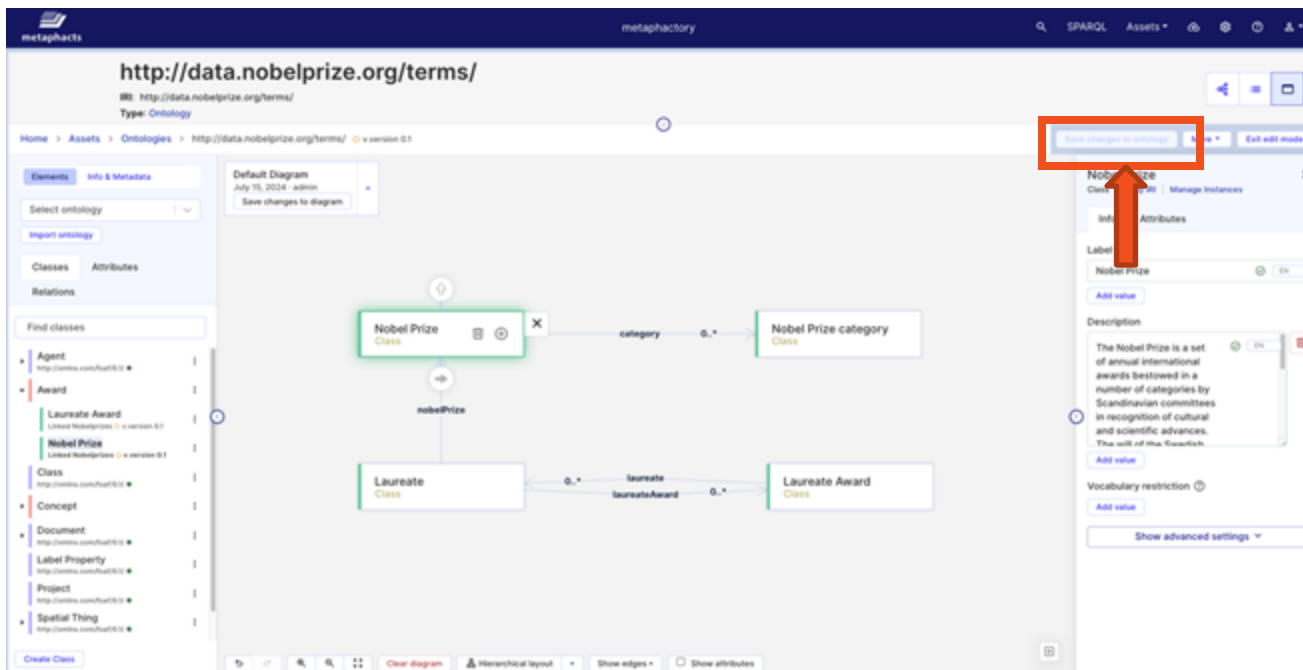


MYDIAGRAM

Ontology VS Diagrams: What to keep in mind!

Default Ontology View

❑ Save changes to Diagram!



ONTOLOGY EDIT MODE

❑ Saves changes to ontology AND current diagram!



Combining multiple ontologies

Combination supported and encouraged!

Side bar:

Switch between ontologies without resetting Canvas view

Canvas:

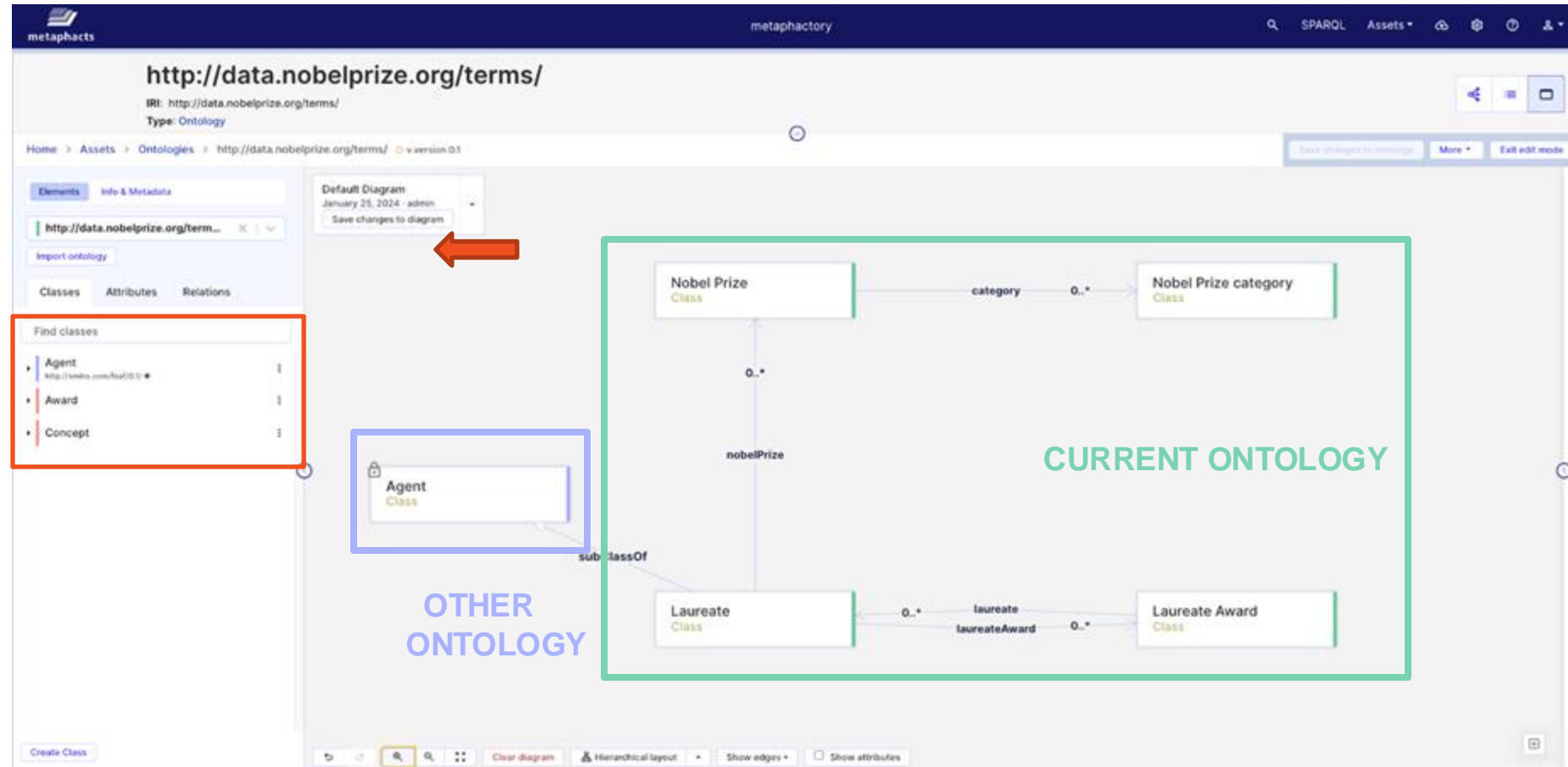
Drag/drop and connect entities from multiple ontologies

Colour scheme:

Same colour = same ontology!

Edit mode:

can only edit entities in current ontology





metaphacts

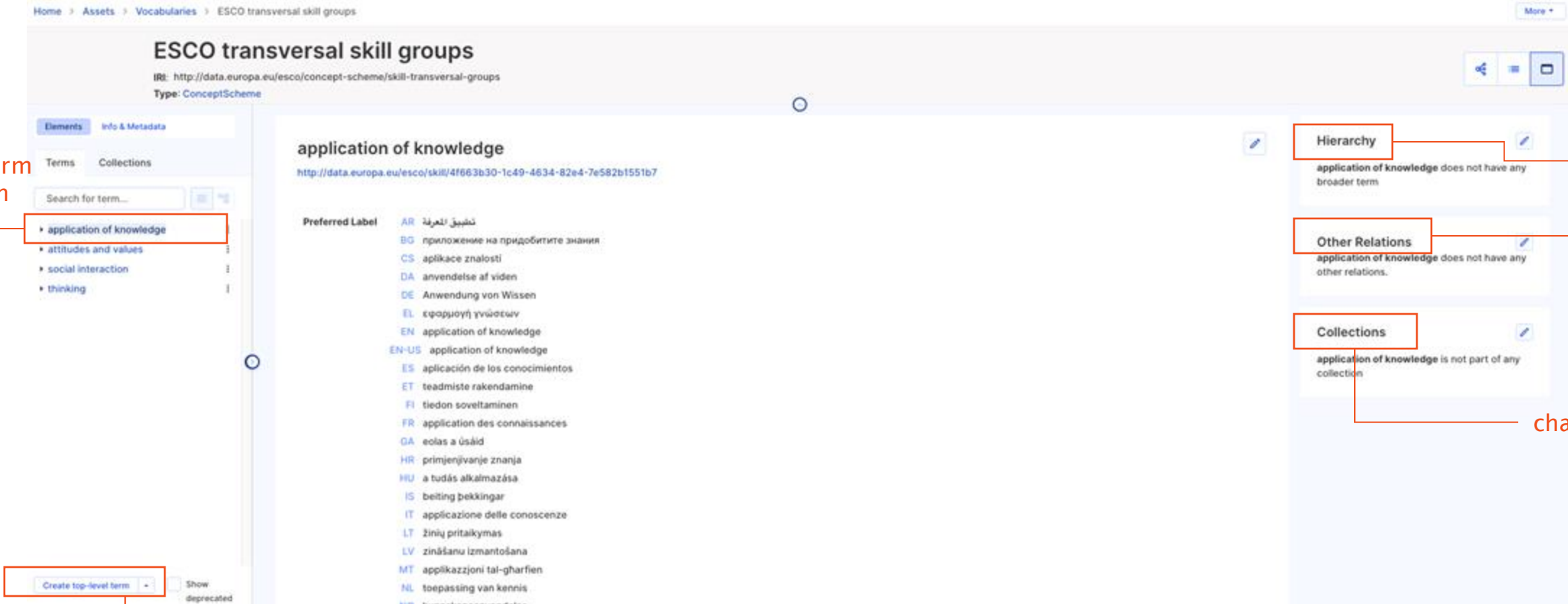
Smart Data. Smart Apps. Smart Decisions.

Vocabulary & taxonomy management

- Explore catalog
- Create a new vocabulary
- Create/edit/import terms
- Create a collection

Vocabulary & taxonomy management

create term
in branch



The screenshot shows the 'ESCO transversal skill groups' interface. The left sidebar contains a tree view with 'application of knowledge' selected. The main area displays the details for 'application of knowledge', including its URI and a list of translations in various languages. The right sidebar contains three sections: 'Hierarchy', 'Other Relations', and 'Collections'. Red boxes and lines highlight specific features and annotations:

- create term in branch**: Points to the 'application of knowledge' item in the left sidebar.
- create a top level term**: Points to the 'Create top-level term' button in the bottom left.
- change hierarchy**: Points to the 'Hierarchy' section in the right sidebar.
- set additional skos relation**: Points to the 'Other Relations' section in the right sidebar.
- change collection**: Points to the 'Collections' section in the right sidebar.

Vocabulary & taxonomy management

- Create a new 'Occupations' Vocabulary
- Create top level terms
- Import narrower terms

Import

Import terms under "Manager". If a term with the given label exists, a broader relationship is established.

General Manager
Branch Manager

Delimiter Options: ☒ Linebreak ☐ Comma ☐ Semicolon

Reset

Save

Home > Assets > Vocabularies > Occupations

Elements

Info & Metadata

Terms

Collections

Search for term...

▼ Engineer

Data Engineer

▸ Software Engineer

Manager



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Hands-On: Go to your tutorial instance!



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Coffee Break



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Examples and Experiences from Industry



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Resource Hub

Resource Type 1

Topic 1

Use Case

Product Feature

Publication Date

Customer

Author

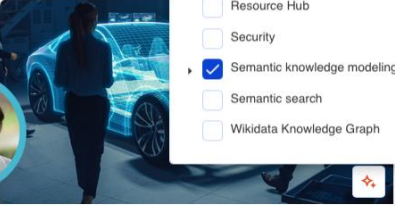
Resource Type: Blog

Quick search

☐ RDF & SPARQL 3
 ☐ Resource Hub 1
 ☐ Security 3
 ☒ Semantic knowledge modeling 18
 ☐ Semantic search 3
 ☐ Wikidata Knowledge Graph 1

modeling X

Show All



Leveraging Knowledge Graphs at Scania and TRATON: A Revolution in Data Management


In this guest blog post, Tanuja Gupta, Manager: Knowledge Graphs and Explainable AI at MAN and previously Solutions Architect and Knowledge Graph Ambassador at Scania (both part of TRATON GROUP), explains how knowledge...

B...

SEMANTIC KNOWLEDG... +1 >

November 14, 2024

Tanuja Gupta



BIBFRAME dilemmas for libraries: Challenges and opportunities


In this article, Richard Wallis, a distinguished Linked Data and Semantic Web expert and thought leader, shares insights from the recent BIBFRAME Workshop in Europe where he presented a novel knowledge graph-powered...

B...

KNOWLEDGE GRAPH M... +4 >

November 6, 2024

Richard Wallis



A guide to ontology governance in metaphactory


In this blog post, we explore the importance of establishing policies and frameworks that govern the creation and management of ontologies within your organization. We also look at how metaphactory's ontology management helps to...

B...

SEMANTIC KNOWLEDG... +2 >

October 25, 2024

Johannes Trame +1 >



How a semantic model can elevate your enterprise information architecture


If your organization doesn't already have an enterprise information architecture in place—it should, and if you do have one, it should be based on a semantic model. In this article, we'll explain what an "enterprise information..."

B...


SEMANTIC KNOWLEDG... +2 >

October 16, 2024


Sebastian Schmidt +1 >




How to approach semantic modeling: Perspectives from a metaphacts friend



We used a knowledge graph to enhance our CMS. Here's how it went.

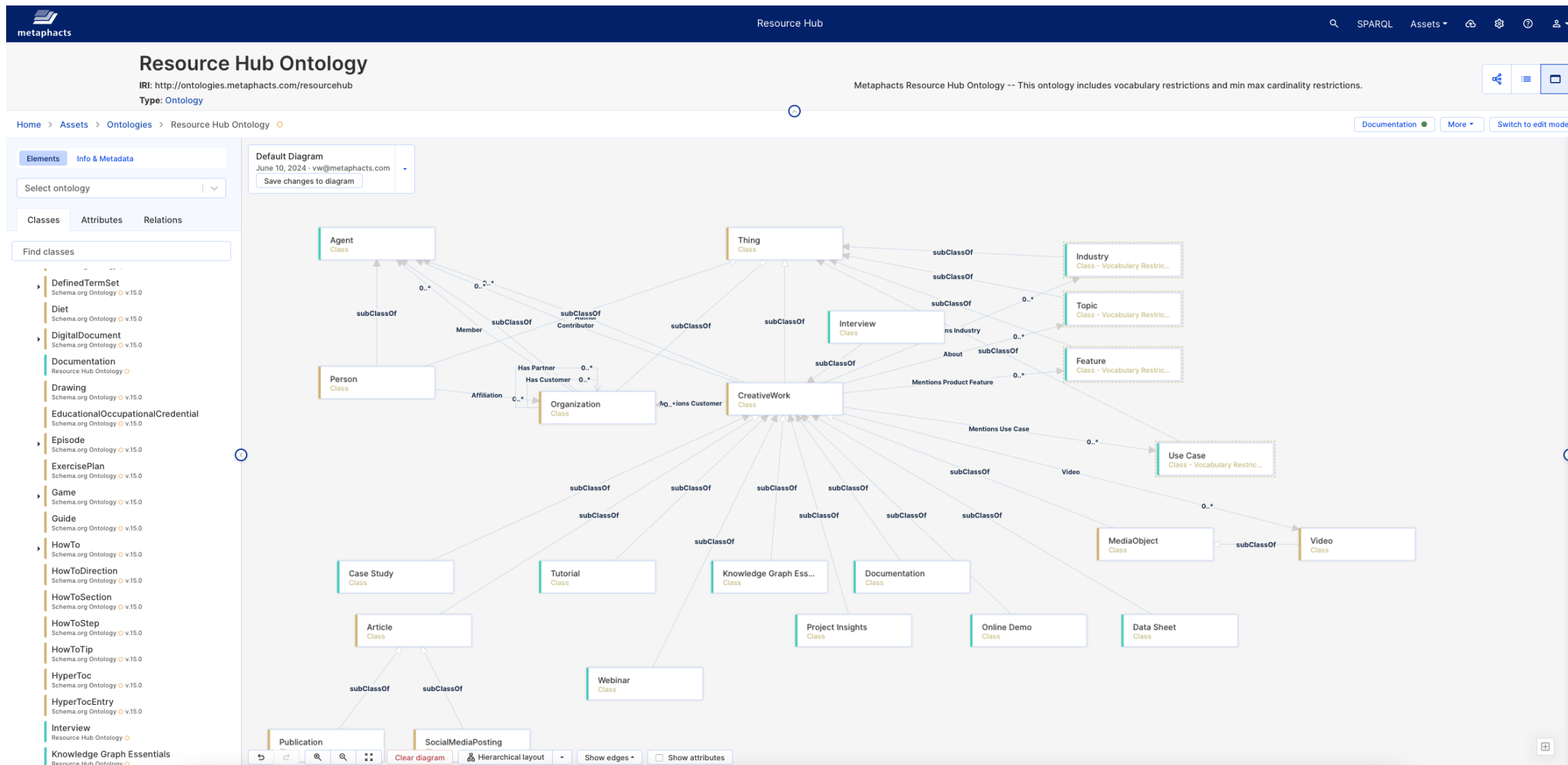



Building massive knowledge graphs using automated ETL pipelines



Fragmented knowledge in pharma: Bridging the divide between private and public data

Resource Hub Ontology



Resource Hub

Home > Assets > Vocabularies > Topics

More

Topics

IRI: <http://vocabularies.metaphacts.com/topics>
Type: ConceptScheme

ElementsInfo & Metadata

TermsCollections

Search for term...

- Data fabric
- Decision Intelligence
- Dimensions Knowledge Graph
- End-user empowerment
- ETL Pipelines
- FAIR data
- Federation
- Getting started with knowledge graphs
- Graph processing
- Knowledge democratization
- Knowledge discovery
- Knowledge graph exploration
- Knowledge graph management
- Knowledge-centric culture
- Linked Data
- Low-code application building
- Nobel Prize Dataset
- Pathfinding
- RDF & SPARQL
- Resource Hub
- Security
- Semantic knowledge modeling
 - Data cataloging
 - Ontology modeling
 - Vocabulary management
- Semantic search
- Wikidata Knowledge Graph

Create top-level termShow deprecated

Ontology modeling

<http://vocabularies.metaphacts.com/topics/2e628743-3be9-4442-ba52-310287c868cd>

Preferred Label	Ontology modeling
Additional Types	Topic

Hierarchy

- Ontology modeling has the following broader term:
 - Semantic knowledge modeling

Other Relations

- Ontology modeling does not have any other relations.

Collections

- Ontology modeling is not part of any collection



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Enterprise Information Architecture

What is Enterprise Information Architecture (EIA) ?

Critical component for (strategic) data, IT and technology management and planning in support of the business.
(often embedded into Enterprise Architecture – EA)



model, assess, structure, analyze, organize, manage & visualize an organization's data & information assets across various systems, technologies, departments, processes & stakeholders



enable efficient decision-making, information sharing, & process optimization within an organization while ensuring security and privacy



common elements include metadata, taxonomies, content management, data modeling, data governance, & API integration



enable stakeholders within to organization to communicate on data (with a common business language)

Why Semantics in EIA*?

- » Connect business and IT (drive data democratization)
- » Connect data governance silos (data catalog, GRC, business glossary, process catalog, ...)
- » Connect company strategy and digital execution



* see also <https://www.forrester.com/blogs/enterprise-architecture-from-design-driven-to-data-driven/>

Semantic layer approach to EIA

Modeling of the data landscape in **two layers**:

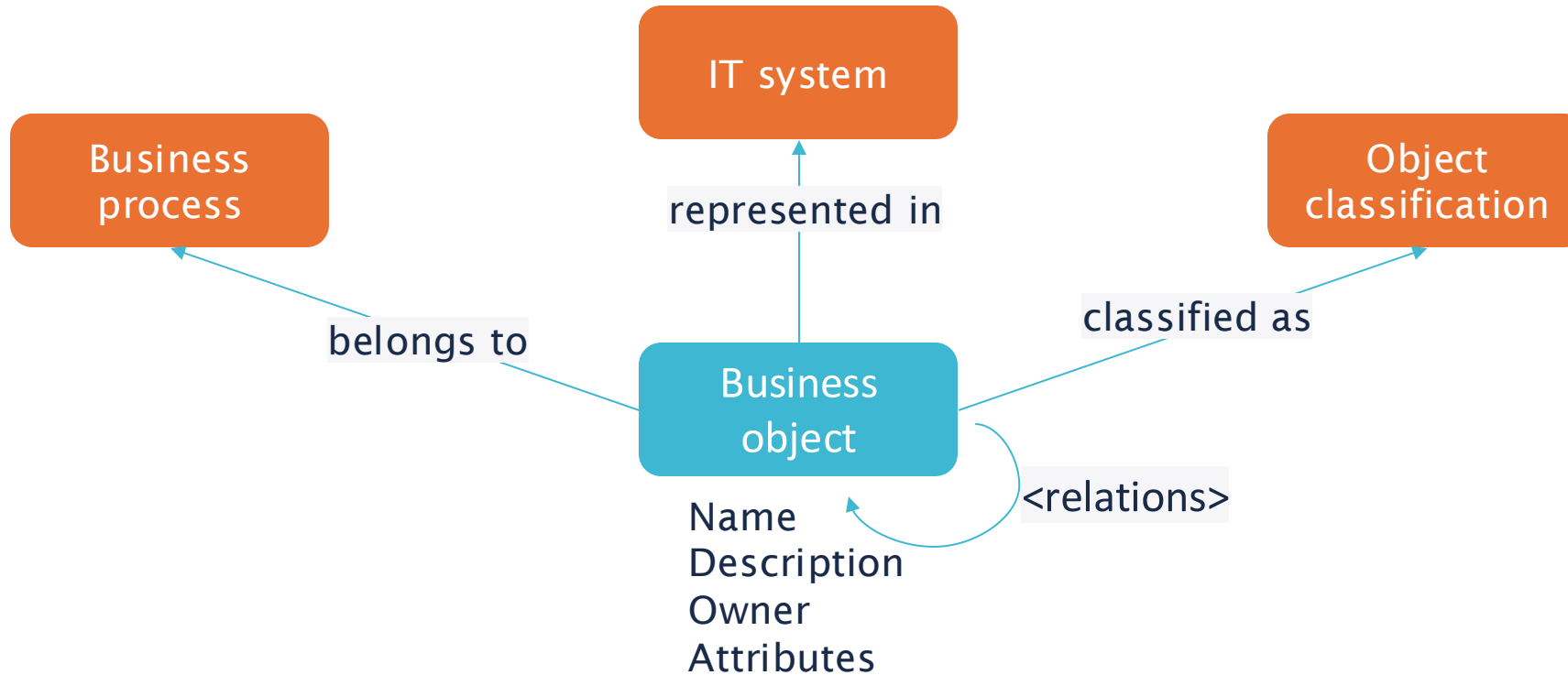
Business objects

- » Real or virtual objects of the business world (e.g., sales order, packaging material, plant)
- » Input, output or interim result of a business process
- » Named by explicit and harmonized terms
- » Represented and persisted by data within IT systems

Physical representations

- » Physical data models representing the physical structure of data
- » Part of the IT landscape
- » Cover complete or parts of business objects (in a specific context = IT system, database, data warehouse, data lake, data transport technology)

The business object in the semantic layer

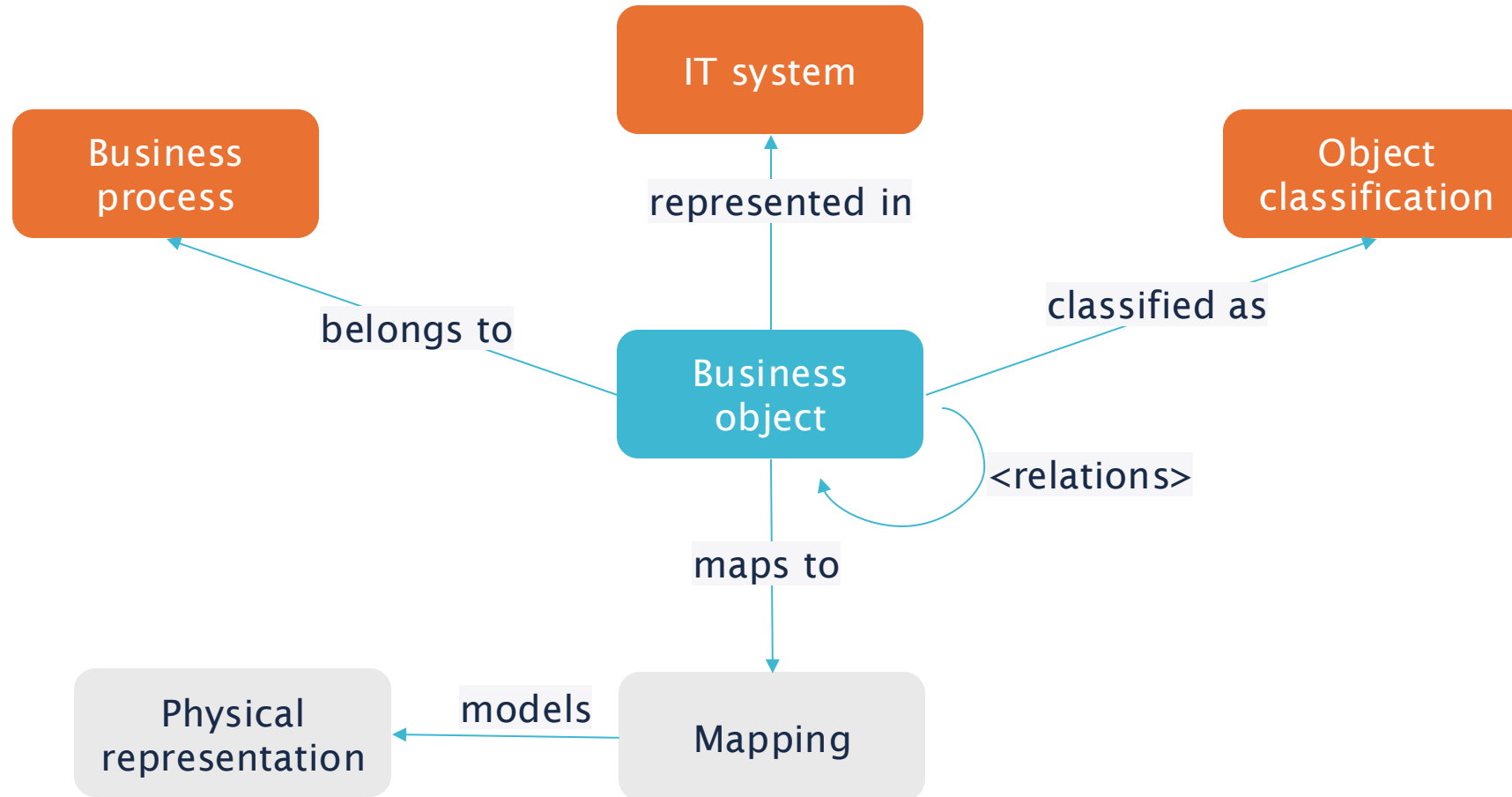


Defined terms

- Data Owner
- Data Steward
- Data Engineer
- Information Architect
- Information Architecture Governance
- User

Expanding the semantic layer to physical layer

Defined terms



- Data Owner
- Data Steward
- Data Engineer
- Information Architect
- Information Architecture Governance
- User



metaphacts

Smart Data. Smart Apps. Smart Decisions.

SemOpenAlex

SemOpenAlex Ontology

ElementsInfo & Metadata

Select ontology

ClassesAttributesRelations

Find classes

ArticleProcessingCharge

SemOpenAlex Ontology v.3.0.0

Author

SemOpenAlex Ontology v.3.0.0

Authorship

SemOpenAlex Ontology v.3.0.0

Concept

SemOpenAlex Ontology v.3.0.0

CountsByYear

SemOpenAlex Ontology v.3.0.0

Funder

SemOpenAlex Ontology v.3.0.0

Geo

SemOpenAlex Ontology v.3.0.0

Institution

SemOpenAlex Ontology v.3.0.0

Location

SemOpenAlex Ontology v.3.0.0

OpenAccess

SemOpenAlex Ontology v.3.0.0

Publisher

SemOpenAlex Ontology v.3.0.0

Source

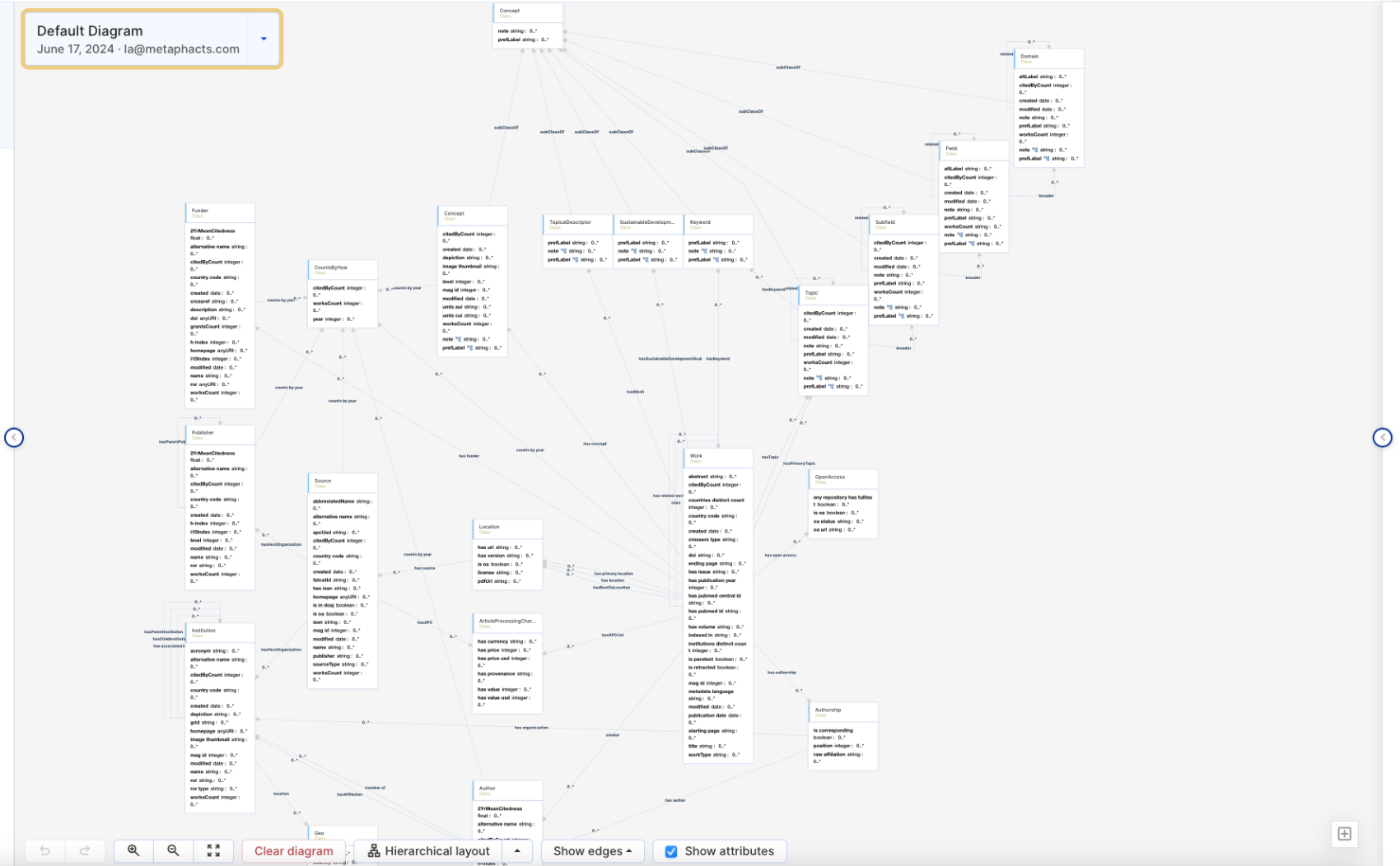
SemOpenAlex Ontology v.3.0.0

Work

SemOpenAlex Ontology v.3.0.0

Default Diagram

June 17, 2024 · la@metaphacts.com



© metaphacts GmbH. For personal use only. Do not redistribute.

SemOpenAlex Explorer

Welcome to [SemOpenAlex](#), the world's most extensive scholarly knowledge graph with over **26 billion RDF & RDF-star triples**. SemOpenAlex provides comprehensive information on **scientific publications** and related entities. It is built upon **OpenAlex** and is licensed under CC0, making it free for use in any context. Use SemOpenAlex to semantically navigate the scholarly space, seamlessly integrate your own data with academic publishing information, and leverage the power of machine learning to identify patterns, make predictions, and generate recommendations based on our [SemOpenAlex entity embeddings](#). [Icon attribution](#)

 Start

 Author ~95 million	 Work ~256 million	 Source ~255 thousand	 Institution ~108 thousand	 Publisher ~10 thousand
 Funder ~32 thousand	 Topic Domain 4	 Topic Field 26	 Topic Subfield 252	 Topic 4,516
 Keyword ~27 thousand	 Concept ~65 thousand	 Topic (Sustainable Development Goals) 17	 Topic (Medical Subject Headings) ~30 thousand	



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Ontology Repository



Ontologies

This is the entry page to the ontology repository. Here you can create, import and manage ontologies.

[Created by anyone](#) ▾[Last modified by anyone](#) ▾[Status](#) 4 ▾[Clear All](#)**Status:**[In development](#) ✕[In review](#) ✕[Ready to be published](#) ✕[Published](#) ✕[Show All](#) ▾

BIBFRAME vocabulary

● v.2.2.0 ⋮<http://id.loc.gov/ontologies/bibframe/>

The Bibframe vocabulary consists of RDF classes and properties used for the description of items cataloged principally by libraries, but may also be used to describe items cataloged by museums and archives. Classes include the three core classes - Work, Instance, and Item - in addition to many more classes to support description. Properties describe characteristics of the resource being described as well as relationships among resources. For example: one Work might be a "translation of" another Work; an Instance may be an "instance of" a particular Bibframe Work. Other properties describe attributes of Works an... [read more](#)

[203 classes](#) [63 attributes](#) [144 relations](#)

Last modified on October 18, 2022

Document Annotation

● ⋮<http://ontologies.metaphacts.com/document-annotation>[0 classes](#) [0 attributes](#) [0 relations](#)

fabio

● v.2.1 ⋮<http://purl.org/spar/fabio>

<https://ontologies.metaphacts.com/>

SPARQL/DUMP

Academy Tag **FRBR** eBusiness Contracts Travel Health

vair - Vocabulary of AI Risks
2024-09-17



SPARQL/DUMP

► Spatial

localName SpatialRepresentation

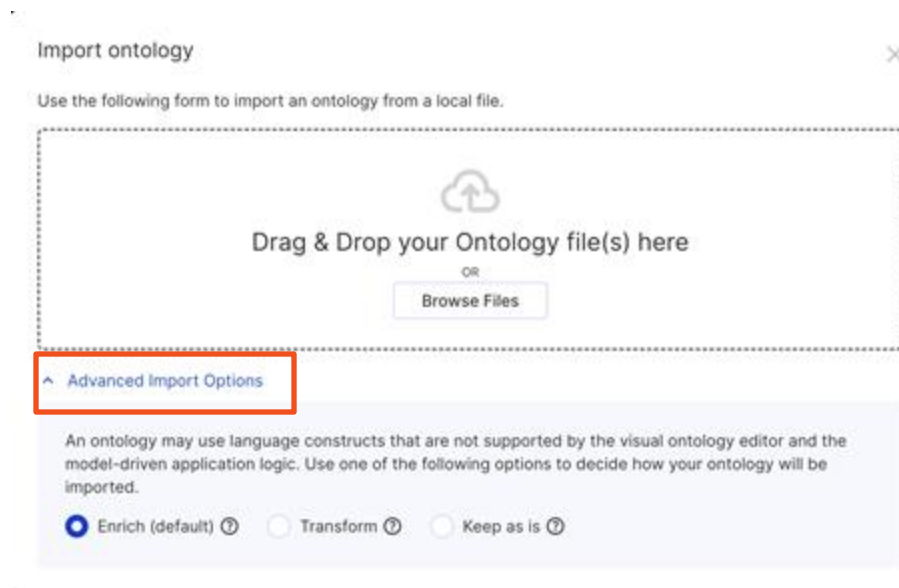
donto (25)

Import existing Ontology/Vocabulary - Ontology transformations

Warning! transformation to non-SHACL ontology in metaphactory can be automatically applied.

Check *Admin > Assets System Configuration* to set the appropriated option. The following three options are available for import:

- **enrich**: Augment the original ontology with language constructs that are compatible with the visual ontology editor and the model-driven application logic.
- **transform**: Transform the original ontology to language constructs that are compatible with the visual ontology editor and the model-driven application logic. Original language constructs may be dropped as part of transformation.
- **keep**: Import the ontology as is. Incompatible language constructs will not be transformed.



Import ontology

Use the following form to import an ontology from a local file.

Drag & Drop your Ontology file(s) here

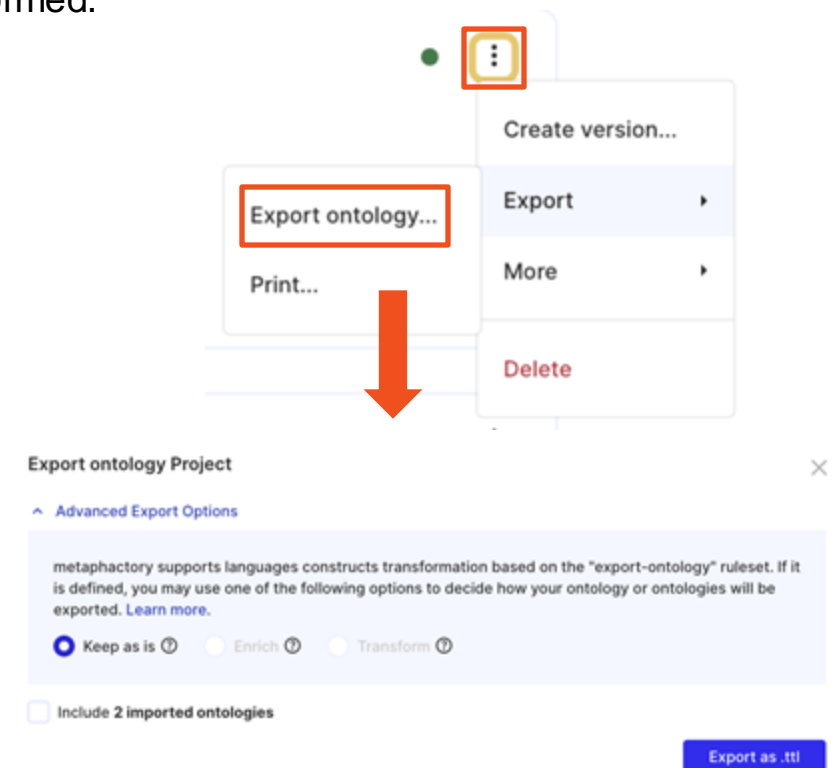
OR

Browse Files

Advanced Import Options

An ontology may use language constructs that are not supported by the visual ontology editor and the model-driven application logic. Use one of the following options to decide how your ontology will be imported.

☒ Enrich (default) ☐ Transform ☐ Keep as is



Create version...

Export

More

Delete

Export ontology...

Print...

Export ontology Project

Advanced Export Options

metaphactory supports languages constructs transformation based on the "export-ontology" ruleset. If it is defined, you may use one of the following options to decide how your ontology or ontologies will be exported. [Learn more.](#)

☒ Keep as is ☐ Enrich ☐ Transform

☐ Include 2 imported ontologies

Export as .ttl

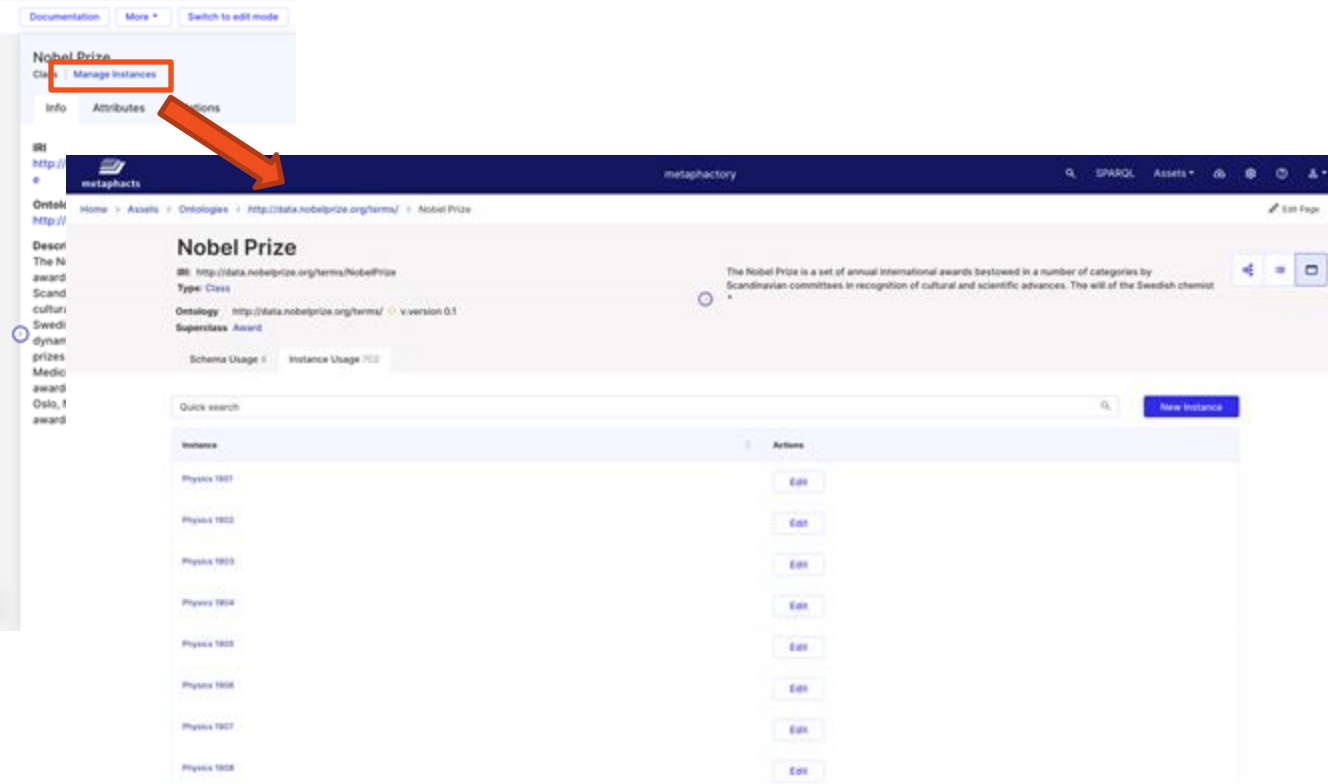
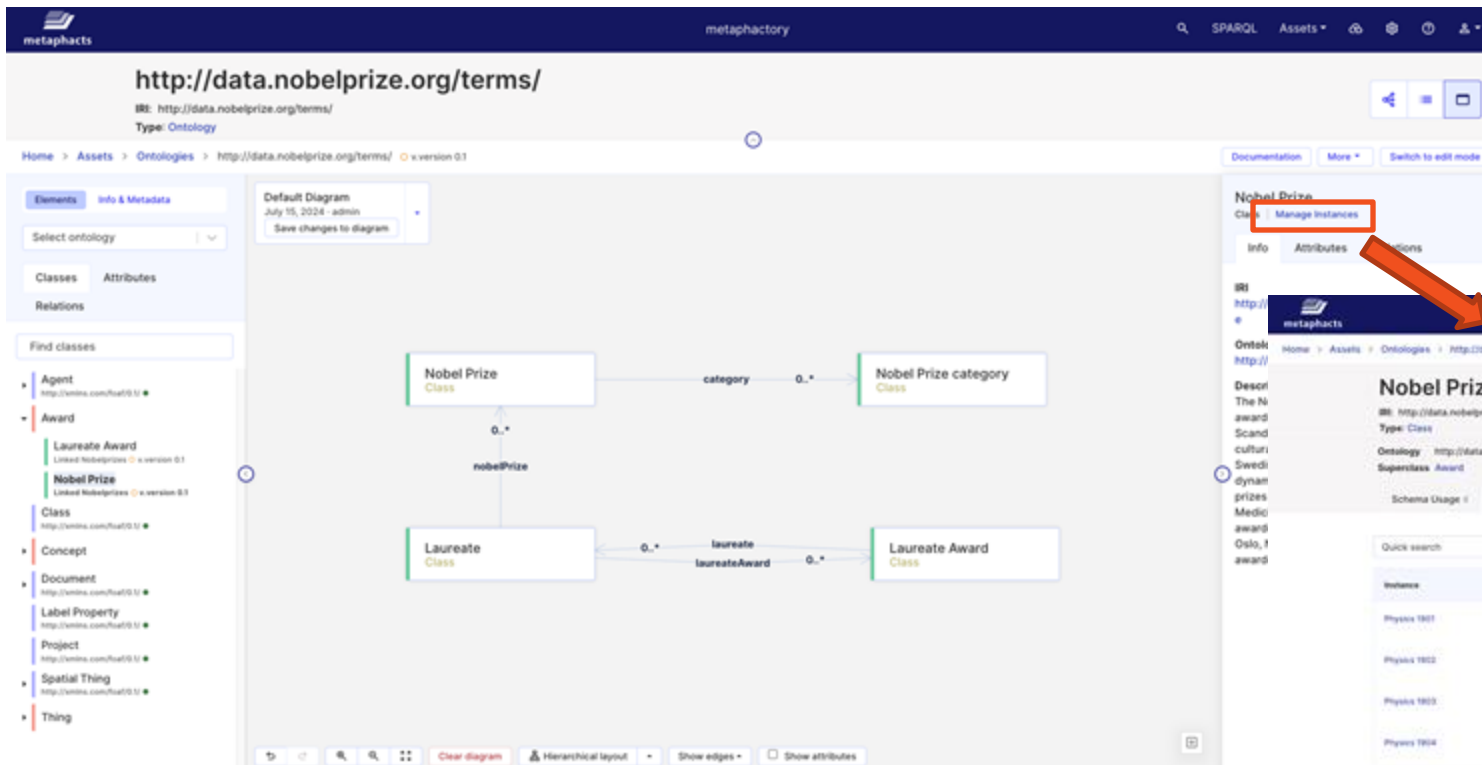


metaphacts

Smart Data. Smart Apps. Smart Decisions.

Instance Data Management

Instance data management



The screenshot shows the 'Nobel Prize' class details page in the metaphactory interface. The page includes a description of the Nobel Prize and a list of instances.

Nobel Prize
Class
July 15, 2024 - admin
Save changes to diagram

IRI: <http://data.nobelprize.org/terms/NobelPrize>
Type: Class
Ontology: <http://data.nobelprize.org/terms/> v:version 0.1
Superclass: Award

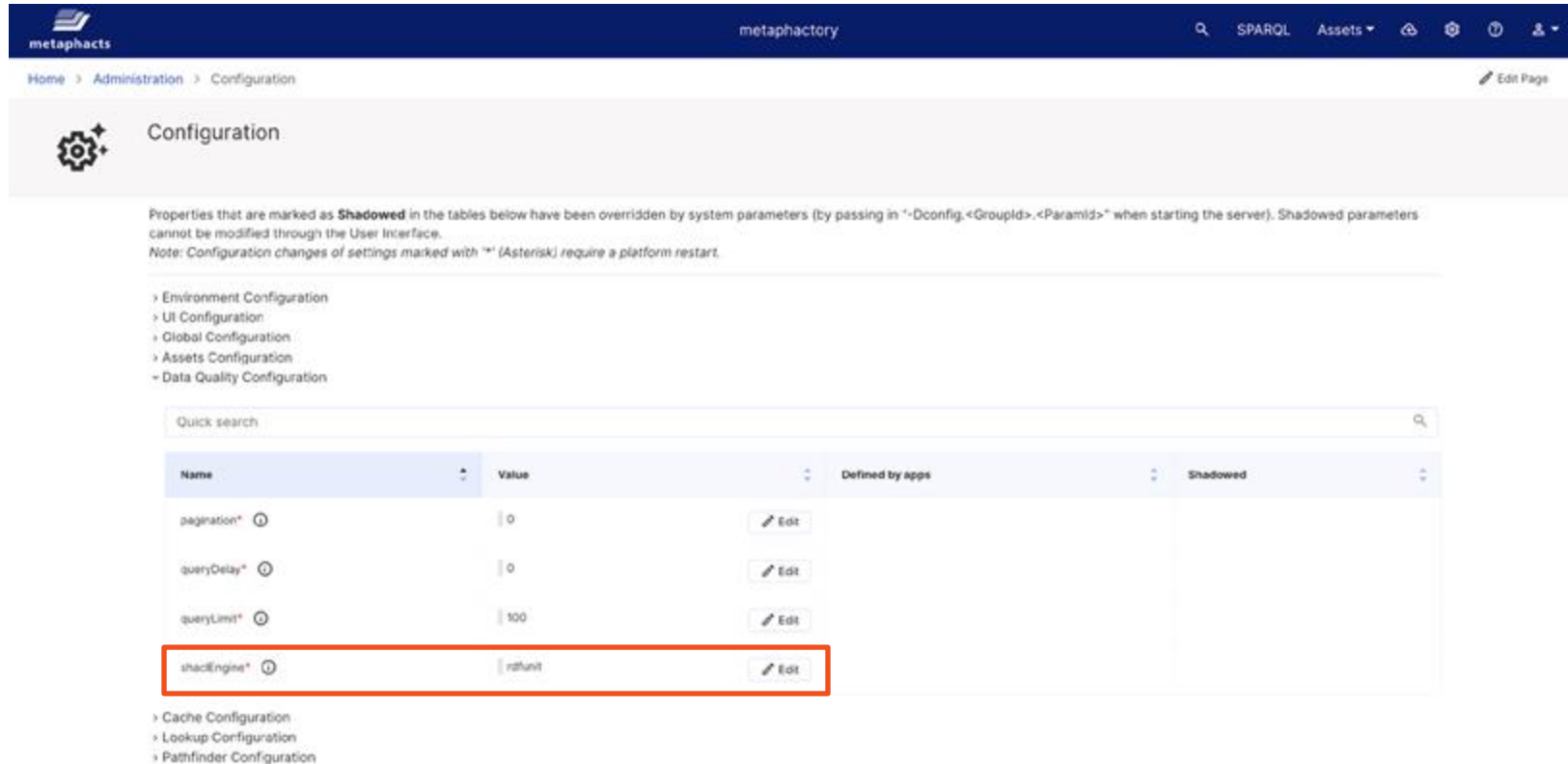
Schema Usage: Instance Usage

Quick search:

Instance	Actions
Physics 1901	<input type="button" value="Edit"/>
Physics 1902	<input type="button" value="Edit"/>
Physics 1903	<input type="button" value="Edit"/>
Physics 1904	<input type="button" value="Edit"/>
Physics 1905	<input type="button" value="Edit"/>
Physics 1906	<input type="button" value="Edit"/>
Physics 1907	<input type="button" value="Edit"/>
Physics 1908	<input type="button" value="Edit"/>

Model-driven Data Validation

- metaphactory uses an internal SHACL engine by default
 - Possible to set up database's own engine (Stardog or GraphDB)



The screenshot shows the 'Configuration' page in the metaphactory interface. The page has a dark blue header with the 'metaphacts' logo and 'metaphactory' text. Below the header, there's a breadcrumb trail: 'Home > Administration > Configuration'. The main content area is titled 'Configuration' with a gear icon. A note explains that properties marked as 'Shadowed' are overridden by system parameters and cannot be modified through the UI. Below this, there's a list of configuration categories: Environment Configuration, UI Configuration, Global Configuration, Assets Configuration, and Data Quality Configuration. A 'Quick search' bar is present. A table lists configuration items with columns for Name, Value, Defined by apps, and Shadowed. The 'shaclEngine*' row is highlighted with a red box, showing a value of 'rdflib' and an 'Edit' button. Other rows include 'pagination*', 'queryDelay*', and 'queryLimit*'. At the bottom, there are links for Cache Configuration, Lookup Configuration, and Pathfinder Configuration.

Properties that are marked as **Shadowed** in the tables below have been overridden by system parameters (by passing in "--Dconfig.<Groupid>.<Paramid>" when starting the server). Shadowed parameters cannot be modified through the User Interface.
Note: Configuration changes of settings marked with "*" (Asterisk) require a platform restart.

> Environment Configuration
> UI Configuration
> Global Configuration
> Assets Configuration
- Data Quality Configuration

Quick search

Name	Value	Defined by apps	Shadowed
pagination*	0	Edit	
queryDelay*	0	Edit	
queryLimit*	100	Edit	
shaclEngine*	rdflib	Edit	

> Cache Configuration
> Lookup Configuration
> Pathfinder Configuration

Documentation: <https://help.metaphacts.com/resource/Help:DataQuality/>





metaphacts

Smart Data. Smart Apps. Smart Decisions.

Model-driven Development

Configuration process

1. Select search features

☒ **Select classes**
3 classes

3. Select relations and attributes

4. Generate configuration

[Download](#)

[Discard all](#)

This wizard helps you create configuration a for your search, which you can then insert into the page and fine-tune to your specific needs by following the examples in the metaphactory documentation.

Based on your selection, the wizard will generate the configuration for the metaphactory search component.

Select search features (1 of 4)

Configure your search based on your needs and available data.

☒ Enable facets

If enabled, additional contextual filters will be generated on top of the results which allow the user to refine them without changing the initial query.

☒ Show exploration facets

If enabled, facets will be shown when exploring all entities of a domain without further search conditions being formulated. Consider to disable this option to improve performance on large datasets with millions of entities.

☒ Model-Driven Search Profile

If enabled, you will only be asked to identify and select the classes relevant as domain for your search. The selection of relations and attributes will be skipped as the search profile is created automatically at runtime based on the selected classes.

Count Query Optimization

Some database vendors offer built-in optimizations for count queries (e.g. for instances of a given type). In order to efficiently render the counts in the domain cards, it is recommended to use the database specific optimization.

- ☐ GraphDB
- ☐ Stardog
- ☒ Generic SPARQL

[Next: Select classes](#)

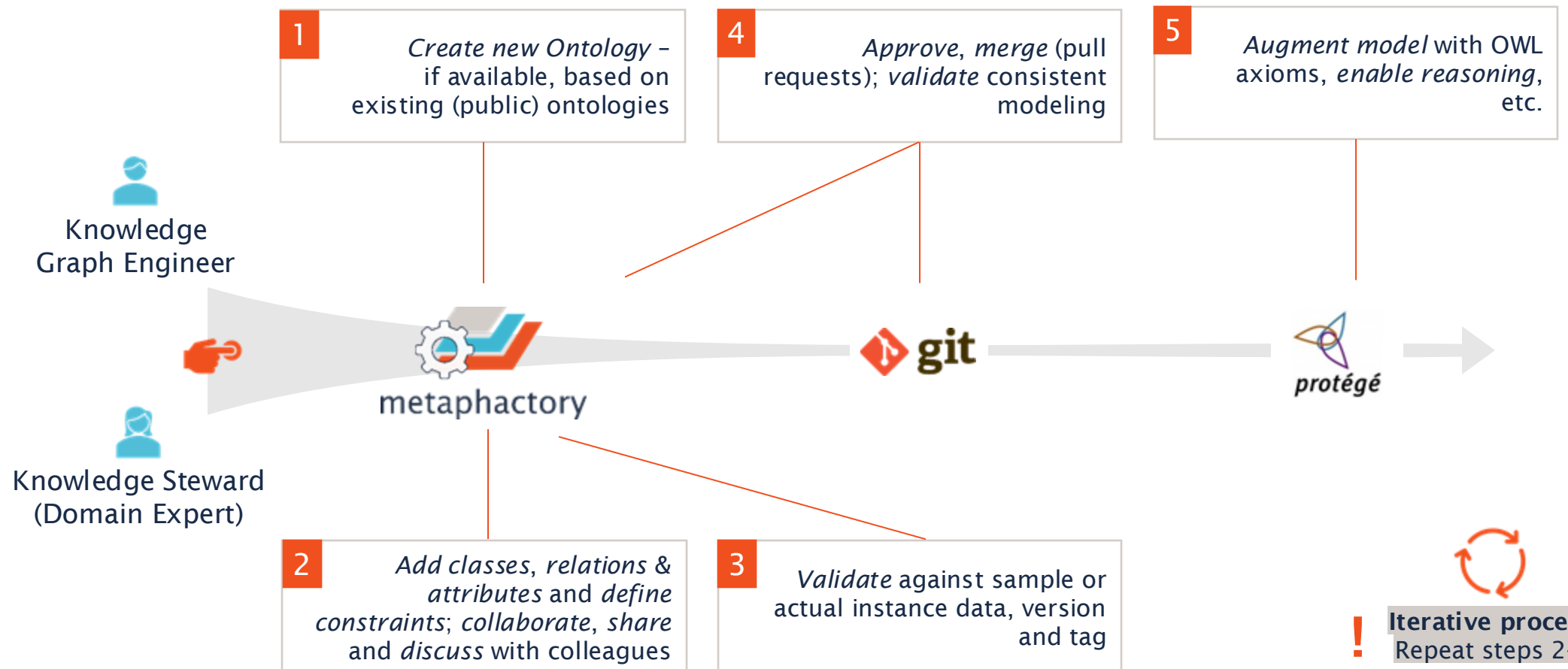


metaphacts

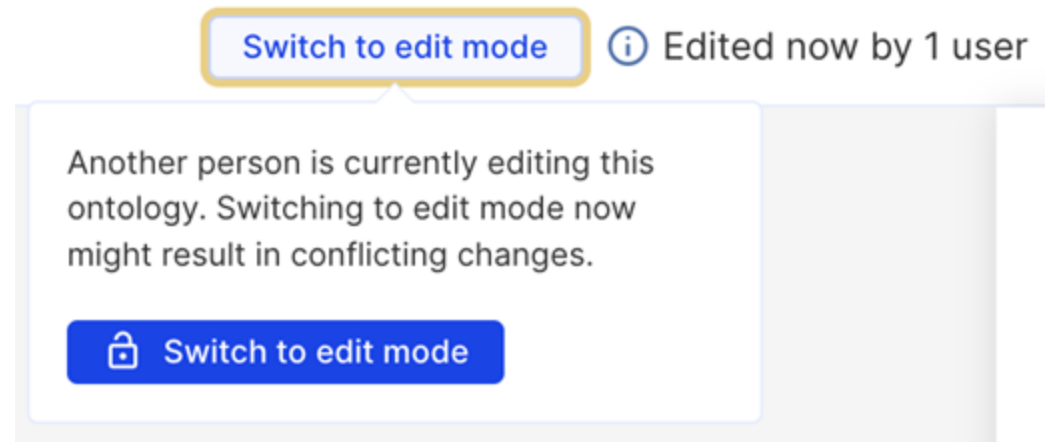
Smart Data. Smart Apps. Smart Decisions.

Advanced Modeling Features

Collaborative semantic knowledge modeling

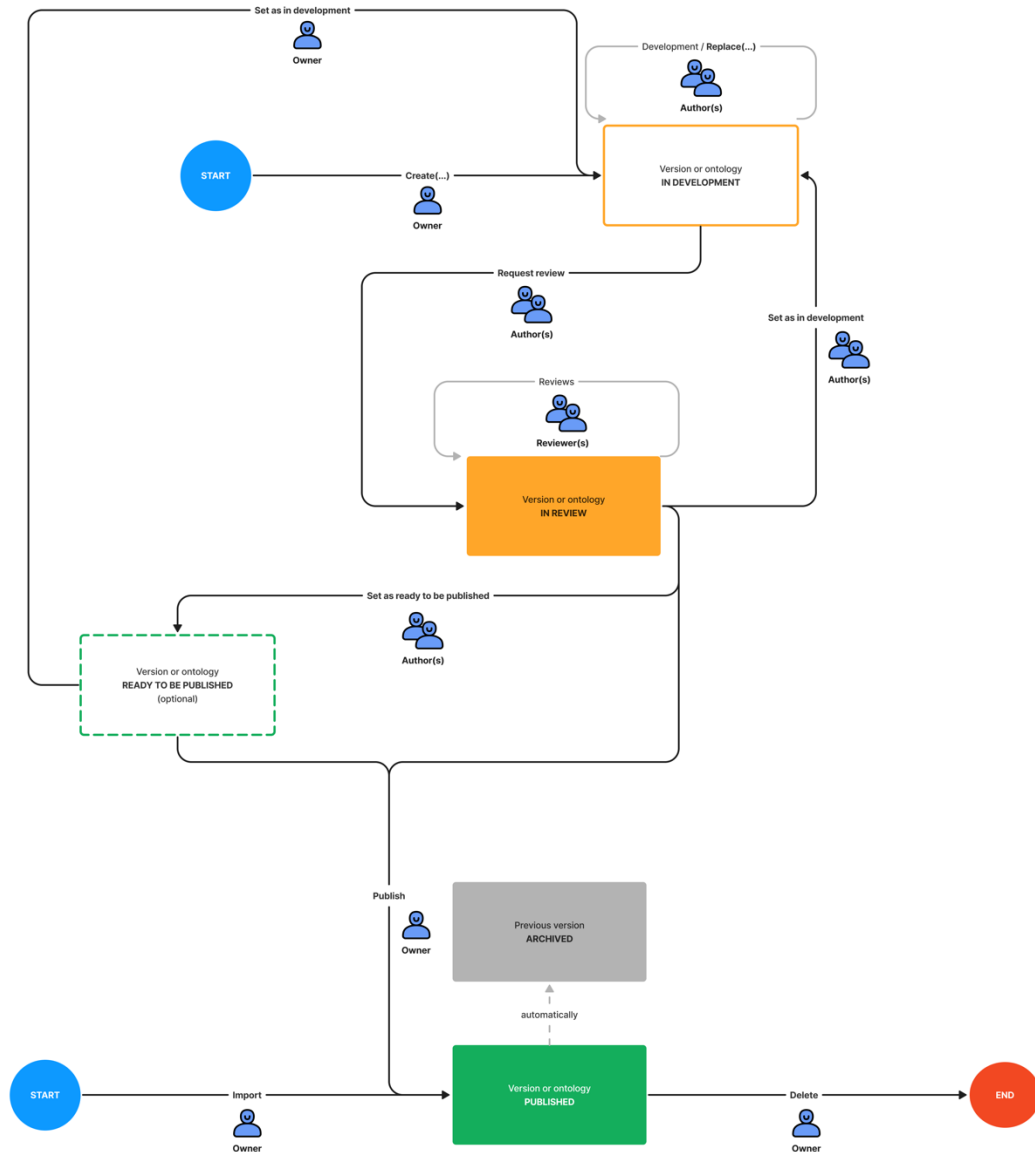


- ‘Soft Locking mechanism’: Alerts that other engineers are editing the asset
 - No. of active editors/edit sessions
 - Users are anonymized by default (data privacy) – can be overridden
 - Admin > Sys. Settings > UI config (anonymizeAssetLocks = false)



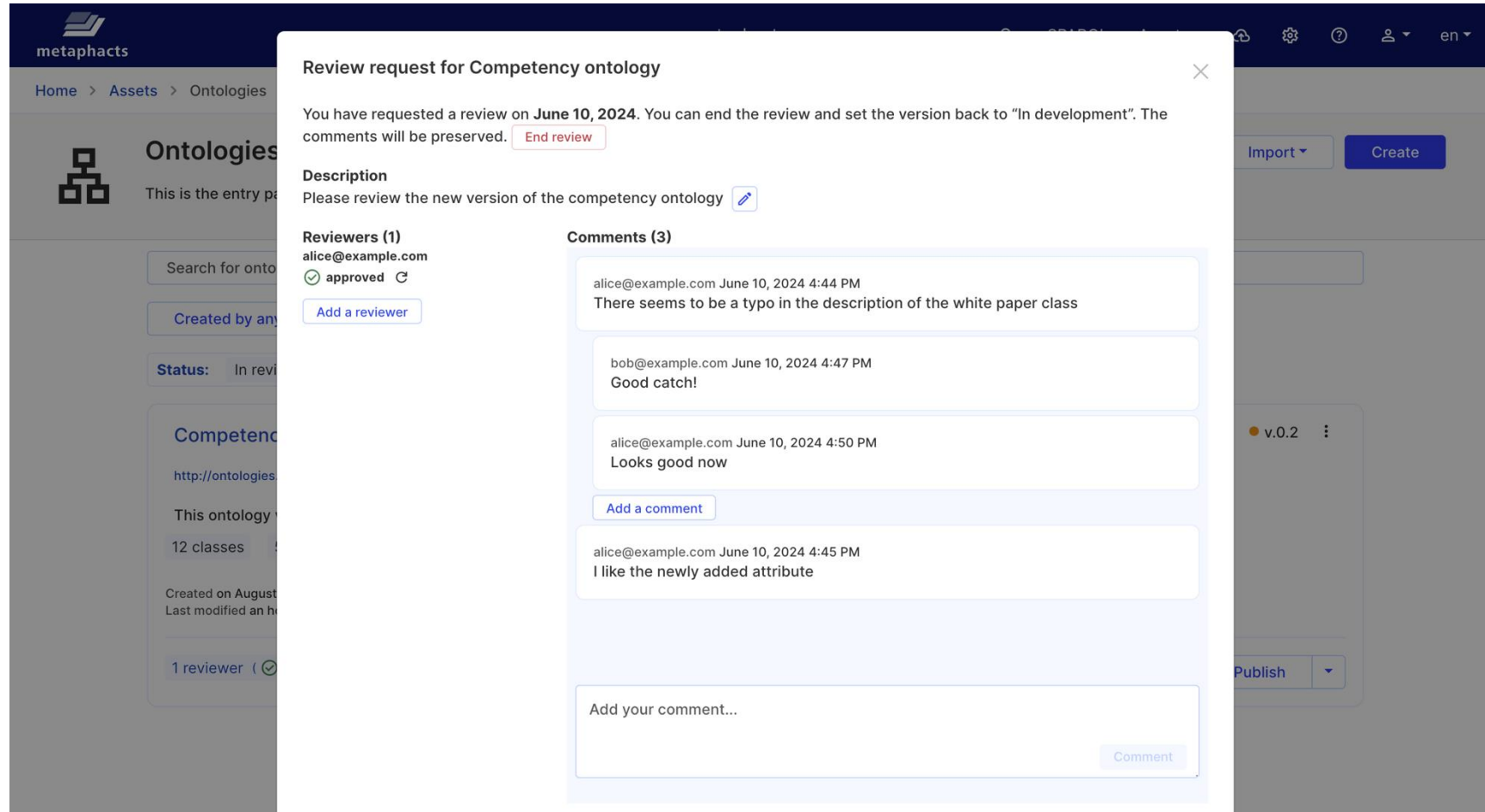
- Important Notes: 1 editing session per ontology supported
 - Notifications serve informational purpose, no hard locking
 - Saving changes persists/replaces the entire graph of the ontology
 - Offline checks amongst engineers required in case of questionable locks (e.g. other user has a stale session, or same user in another Web browser tab)

Collaborative Editorial Workflows



<https://help.metaphacts.com/resource/Help:EditorialWorkflowsAndVersioning>

Review Requests



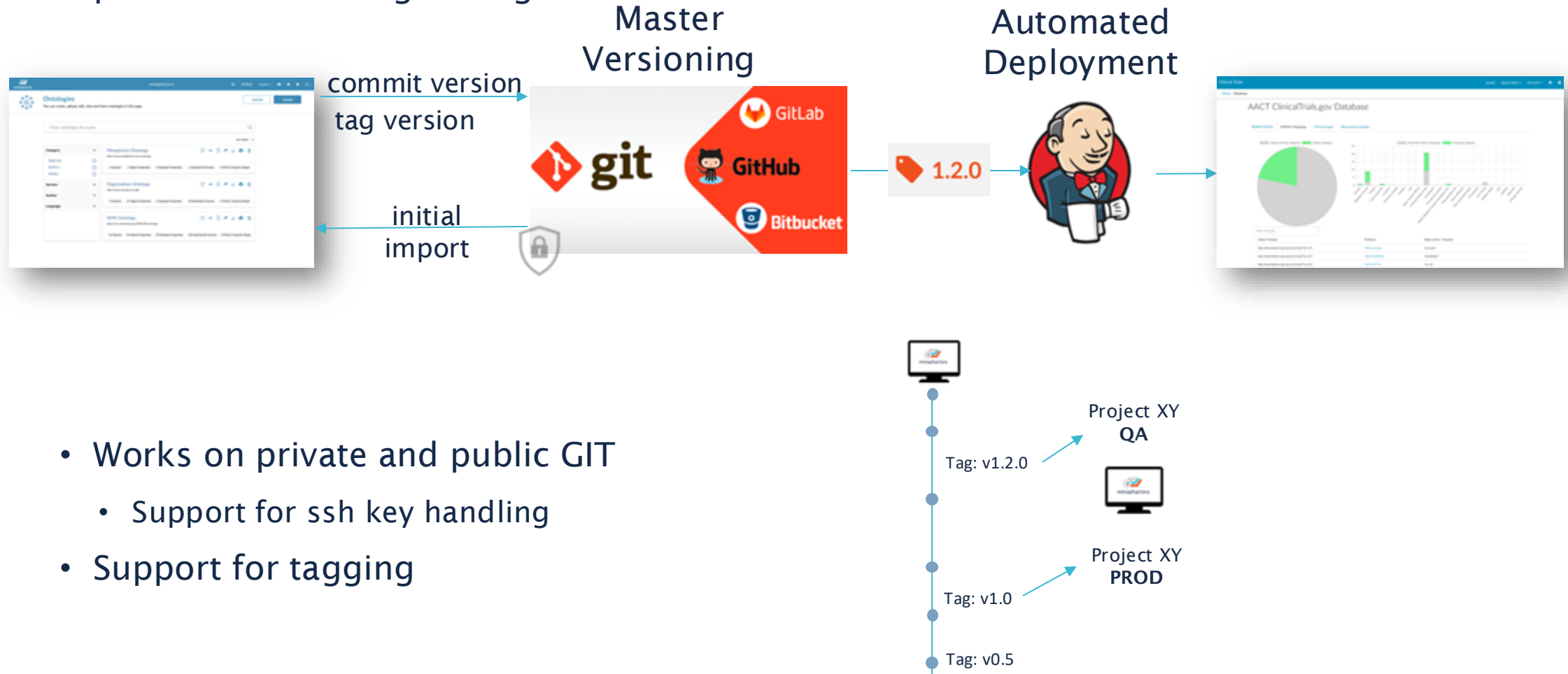
The screenshot displays the metaphacts web application interface. A modal window titled "Review request for Competency ontology" is open, overlaying the main content. The modal contains the following information:

- Header:** "Review request for Competency ontology" with a close button (X).
- Text:** "You have requested a review on **June 10, 2024**. You can end the review and set the version back to 'In development'. The comments will be preserved. [End review](#)".
- Description:** "Please review the new version of the competency ontology" with an edit icon.
- Reviewers (1):** A list showing "alice@example.com" with a green checkmark and the word "approved", and an "Add a reviewer" button.
- Comments (3):** A list of three comments:
 - alice@example.com June 10, 2024 4:44 PM: "There seems to be a typo in the description of the white paper class"
 - bob@example.com June 10, 2024 4:47 PM: "Good catch!"
 - alice@example.com June 10, 2024 4:50 PM: "Looks good now"An "Add a comment" button is located below the comments.
- Comment Input:** A text area with the placeholder "Add your comment..." and a "Comment" button.

In the background, the main interface shows the "Ontologies" section with a search bar, a "Created by any" filter, and a "Status" dropdown set to "In review". A specific ontology, "Competency", is highlighted, showing its URL, a description, and a list of classes. A "1 reviewer" status is also visible.

Ontology & Vocabulary Versioning with Git

- Import and versioning through GIT



- Works on private and public GIT
 - Support for ssh key handling
- Support for tagging

Saving Changes & Documentation, History

Git versioning



Save to Git

History

Base Revision

18550fa5

Storage ?

assetStorage

Location * ?

nobel.ttl

Branch ?

☒ develop-20240927-1157-nobel.ttl

☐ main

Commit message *

Added new class <http://data.nobelprize.org/terms-ext/Class>

Discard pending changes

Save to Git

Git Versioning

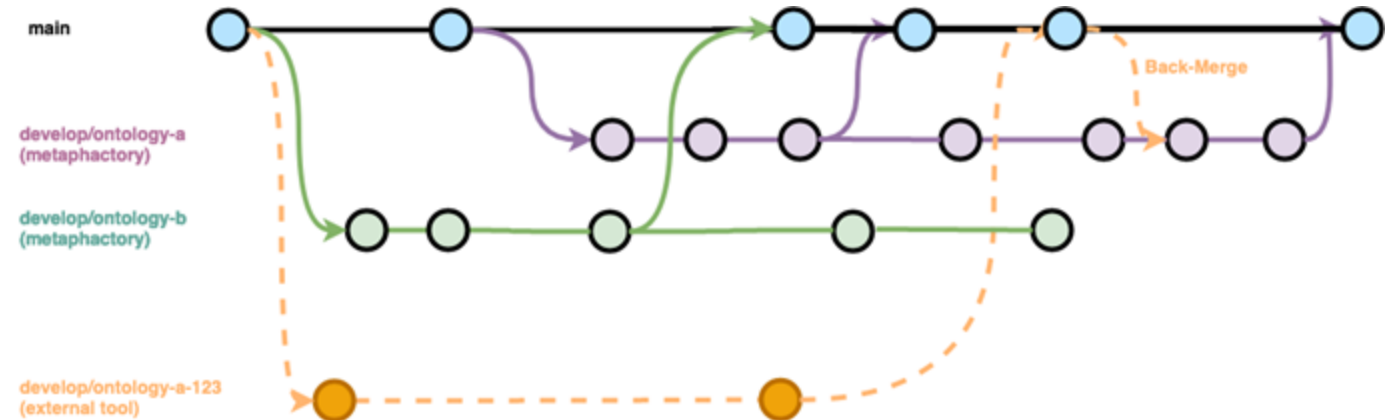
Save to Git

History

Git Hash	Summary	Date	Modified by	Actions
145b4d59	Modified toy ontology metadata	2023-10-05	domaine	Restore Tag
1b517a3d	Modified ontology metadata	2023-10-05	domaine	Restore Tag
9f71ca2c	Initial version of http://ontologies.metaphacts.com/animals	2023-10-05	admin	Restore Tag

- **Recommended workflow**

- Main Branch: contains stable (release) versions of assets
- Development Branch(es): Ontology editor operates on revisions
- Small badge indicates asset connection to main branch (Main) or develop branch (Dev).



- **One branch per Ontology or Vocabulary ensures isolated/reviewable changes**

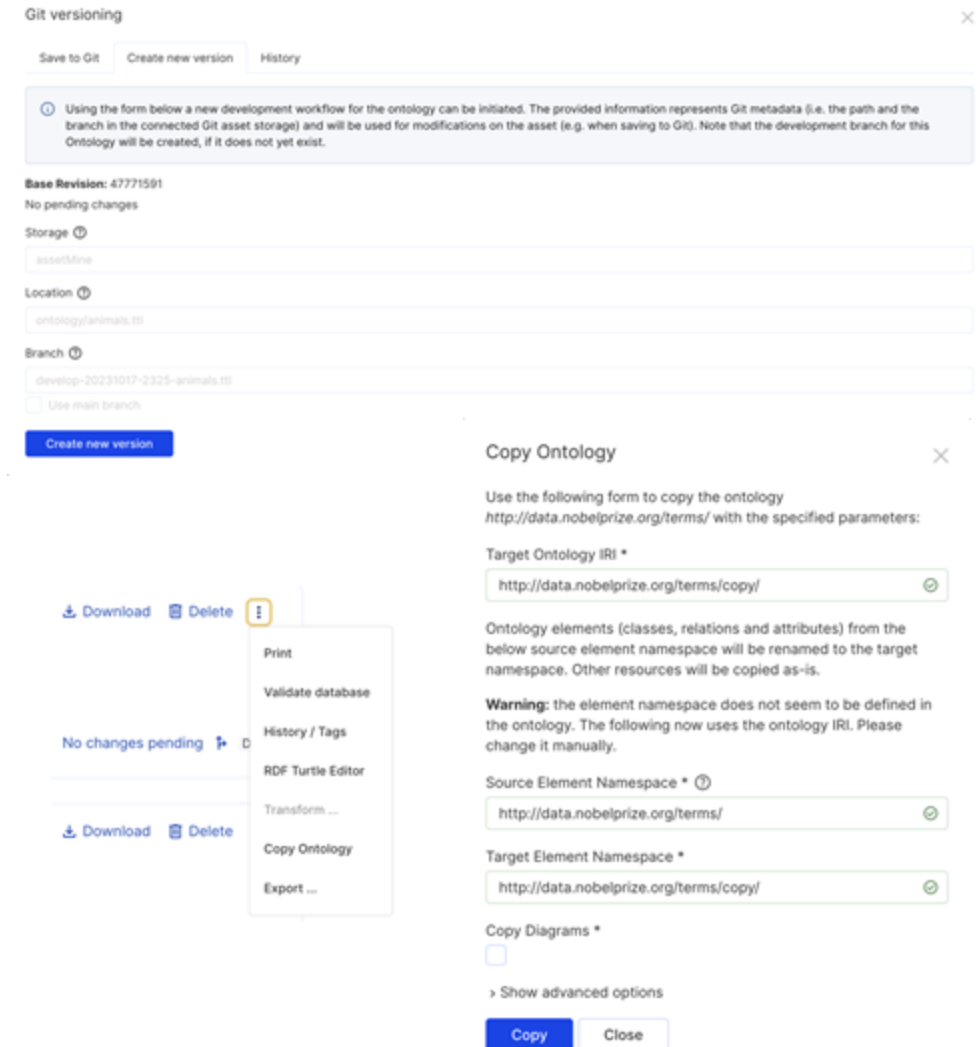
- easy integration in external Git workflows
 - See ontology changes as diff with main branch
 - PRs can be used for reviews
 - Merge + Git approvals can be employed for publishing
- Automation with Git-Actions or Webhooks possible , e.g., consistency checks, publish metadata to other DBs, injecting asset to another system - staging systems.

- **Important Notes**

- Asset scanning on startup operates on main branch, i.e., only ontologies/vocabularies in main branch can be imported
- Main branch = Configured branch in the Asset Storage (independent of name)

Documentation: <http://help.metaphacts.com/resource/Help:AssetManagement#git-branching-model>

- **Creating new version (branch) of Ontology/Vocabulary**
 1. Existing ontology imported from main branch
 2. Specify and confirm the Development Git metadata (location, branch, etc).
 3. Branch is created in the Git repository and is applied on save
- **Entry points & implications:**
 - Status badge of an asset (Main branch) opens a form to create a new dev version
 - Implication: Asset identity not changed – Ontology/Vocabulary editor operates on existing loaded Asset
 - Copying an existing ontology
 - Implication: New asset identity (i.e., separate/new ontology identifier etc).
 - Parameters pre-set with computed defaults based on original



The screenshot displays two overlapping windows from the metaphacts application. The background window is titled 'Git versioning' and contains a form for creating a new development version. It includes tabs for 'Save to Git', 'Create new version', and 'History'. A message box explains that the form initiates a new development workflow. Below this, there are input fields for 'Base Revision' (47771591), 'Storage' (assetMine), 'Location' (ontology/animals.ttl), and 'Branch' (develop-20231017-2325-animals.ttl). A 'Create new version' button is at the bottom. The foreground window is titled 'Copy Ontology' and provides instructions on how to copy an ontology using a specific IRI. It includes input fields for 'Target Ontology IRI' and 'Source Element Namespace', both set to 'http://data.nobelprize.org/terms/'. There are also checkboxes for 'Copy Diagrams' and a 'Show advanced options' link. A 'Copy' button is at the bottom right.

Documentation: <http://help.metaphacts.com/resource/Help:AssetManagement#git-branching-model>



metaphacts

Smart Data. Smart Apps. Smart Decisions.

Free Exercises

- We will keep the tutorial instances running for one week
- Upon email request we can extend to four weeks
- Then there are various options to continue / get started with your own metaphactory

To Continue with metaphactory: <https://metaphacts.com/get-started>



Product

Solutions

Company

Resource Hub

Blog

Get Started

TRY METAPHACTORY TODAY

Create, manage, extract & present domain knowledge all in one powerful knowledge graph platform. Get started with metaphactory by selecting one of our free trial options below!

Trial with guided support

Deploy-it-yourself



Select your trial option

Whether you prefer to do-it-yourself or receive guided support, we have a range of trial options to choose from.

Your learning path..

Building Knowledge Graph Applications

metaphactory Basics

Covers out-of-the-box features and introduces you to Knowledge Graph search, exploration and navigation using metaphactory.

- Intro to metaphactory
- Overview of platform features
- Basics

App Building Basics

Covers platform customization topics to support build apps for data visualization and discovery.

- » Templates & Customization
- » Semantic components
- » Events

App Building Advanced

Covers advanced features for templating and interactions involving multiple components.

- » Security & permissions
- » App packing and lifecycle
- » Advanced data authoring, search framework and federation.

Data Authoring

Focuses on creating forms for data authoring and editing, and how to use them in data curation workflows.

- » Forms & Model-driven Forms
- » Outlook Visual Editing
- » Workflows

Search

Provides a deep dive into configuring search interfaces and search results.

- » Semantic search framework
- » Facet filtering
- » Lookup Service
- » Result visualization

Federation

Focuses on the setup and querying of external data sources to augment the knowledge graph beyond the main graph database.

- » Ephedra & FedX
- » Rest API
- » SQL

Platform extensions

Covers the development of custom extension for custom UI or backend requirements.

- » App packing and deployment
- » Frontend extensions as custom components in JS
- » Backend extensions in Java

Building models for domain experts and data engineers

Visual Modelling Basics
~ Ontology & Vocabulary ~

A self-guided business user tutorial to semantic knowledge modeling.

- » Vocabulary & Taxonomy
- » Ontology
- » Data cataloging

Visual Modelling Advanced
~ Ontology & Vocabulary ~

Advanced features for knowledge modeling & assets management.

- » Editorial workflows
- » Management
- » Versioning

Certification (optional)

- metaphactory
- ★ Knowledge Graph Engineer (Associate)
- ★ Knowledge Graph Application Engineer (Associate)
- ★★ Knowledge Graph Solution Architect (Advanced)
- ★★★ Knowledge Graph Solution Architect (Expert)

The concepts and standards behind Knowledge Graphs

OWL + SHACL

Covers ontology modelling with OWL and the creation of SHACL constraints using metaphactory.

- » Classes, Properties
- » OWL in RDF syntax
- » SHACL node shapes

RDF + SPARQL

Provides an overview of the RDF data model and the use of the SPARQL query language with metaphactory.

- » RDF triples, Named Graphs
- » SPARQL query forms
- » Outlook RDF-star, SPARQL-star

Production readiness

DevOps

Covers the deployment and configuration of metaphactory for production setups.

- » Docker deployment
- » Prod-QA-DEV setup, Git storage
- » Connect to graph databases
- » Single-Sign-On

➤ A training for your custom topic can be arranged upon request.

