IATA WEBINAR

ONE Record Insights

Episode 3

Crafting Ontologies: From physical freight to machine readable data

René Pietzsch
Head of Product Mgt, Linked & Semantic Data Consultant & Scrum Master, eccenca

Andra Blaj
Developer, ONE Record, IATA

IATA

Your hosts today

René Pietzsch
Head of Product Management,
Linked & Semantic Data Consultant &
Scrum Master, eccenca

Andra Blaj
Developer, ONE Record
IATA

ONE Record Insights

IATA WEBINAR

Episode 2
The data model: a digital twin of the air cargo industry
Tuesday, 30th June 11:00 – 12:30 (CEST)

Episode 3
Crafting ontologies: from physical freight to machine readable data
Tuesday, 7th July 11:00 – 12:30 (CEST)

Episode 4
The ONE Record API: an overview of the key features
Tuesday, 14th July 11:00 – 12:30 (CEST)

Episode 5
Data security: securing the Internet of Logistics
Tuesday, 21st July 11:00 – 12:30 (CEST)

Episode 6
Pilot testing: engaging with the cargo community
Tuesday, 28th July 11:00 – 12:30 (CEST)

ONE Record Insights

IATA WEBINAR

Episode 2
The data model: a digital twin of the air cargo industry
Tuesday, 30th June 11:00 – 12:30 (CEST)

Episode 3
Crafting ontologies: from physical freight to machine readable data
Tuesday, 7th July 11:00 – 12:30 (CEST)

Episode 4
The ONE Record API: an overview of the key features
Tuesday, 14th July 11:00 – 12:30 (CEST)

Episode 5
Data security: securing the Internet of Logistics
Tuesday, 21st July 11:00 – 12:30 (CEST)

Episode 6
Pilot testing: engaging with the cargo community
Tuesday, 28th July 11:00 – 12:30 (CEST)

Crafting Ontologies: From physical freight to machine readable data

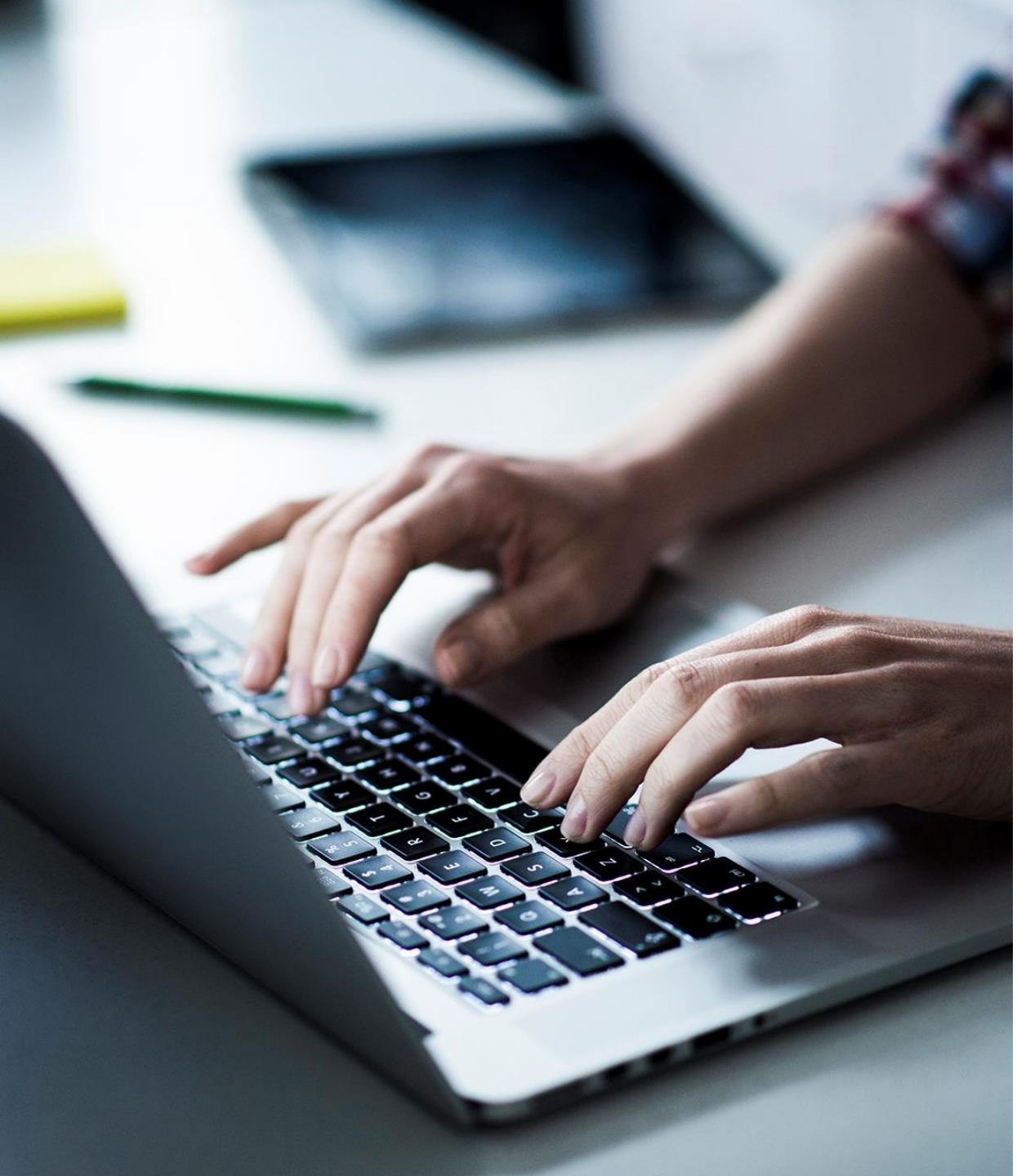
Part 1 History and context

Part 2 Let’s craft an ontology

Part 3 Operationalizing ontologies

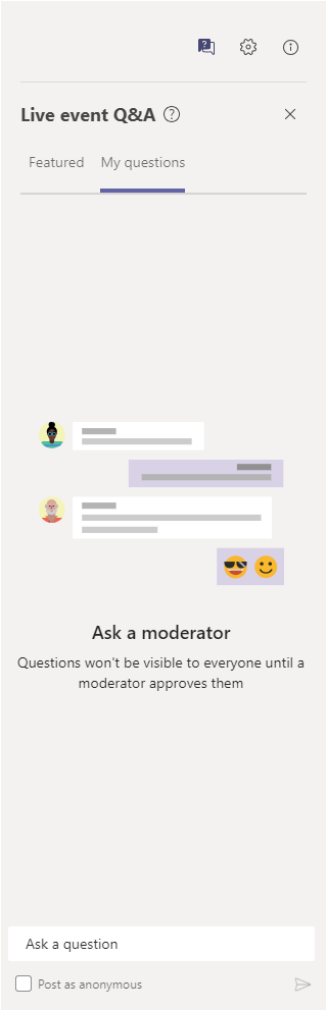
Part 4 The ONE Record ontology

Part 5 Q&A



How to participate during the meeting?

- **You can only hear the presenters**
- **Your microphones are disabled**
- **Use the questions box to interact**
- **Simply enter your questions in the chat box on the right**



This meeting is recorded for future use

The entire recording along with questions will be available shortly after this webinar finishes.

Simply click on the link in the invite for the live event to access it.

The presentations shown today will be available for download on our website:

www.iata.org/one-record

ONE Record Insights

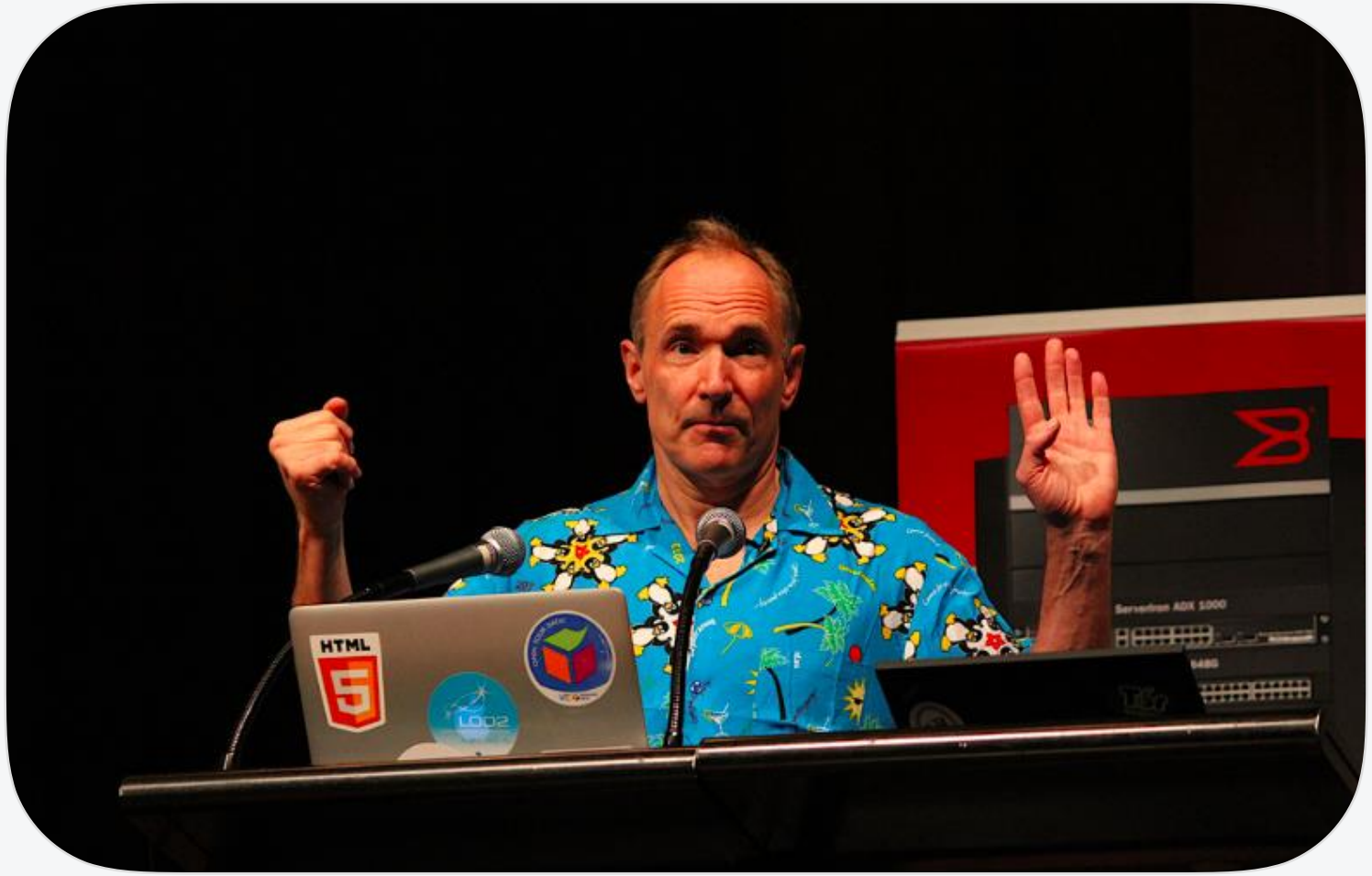
Part 1

History and context

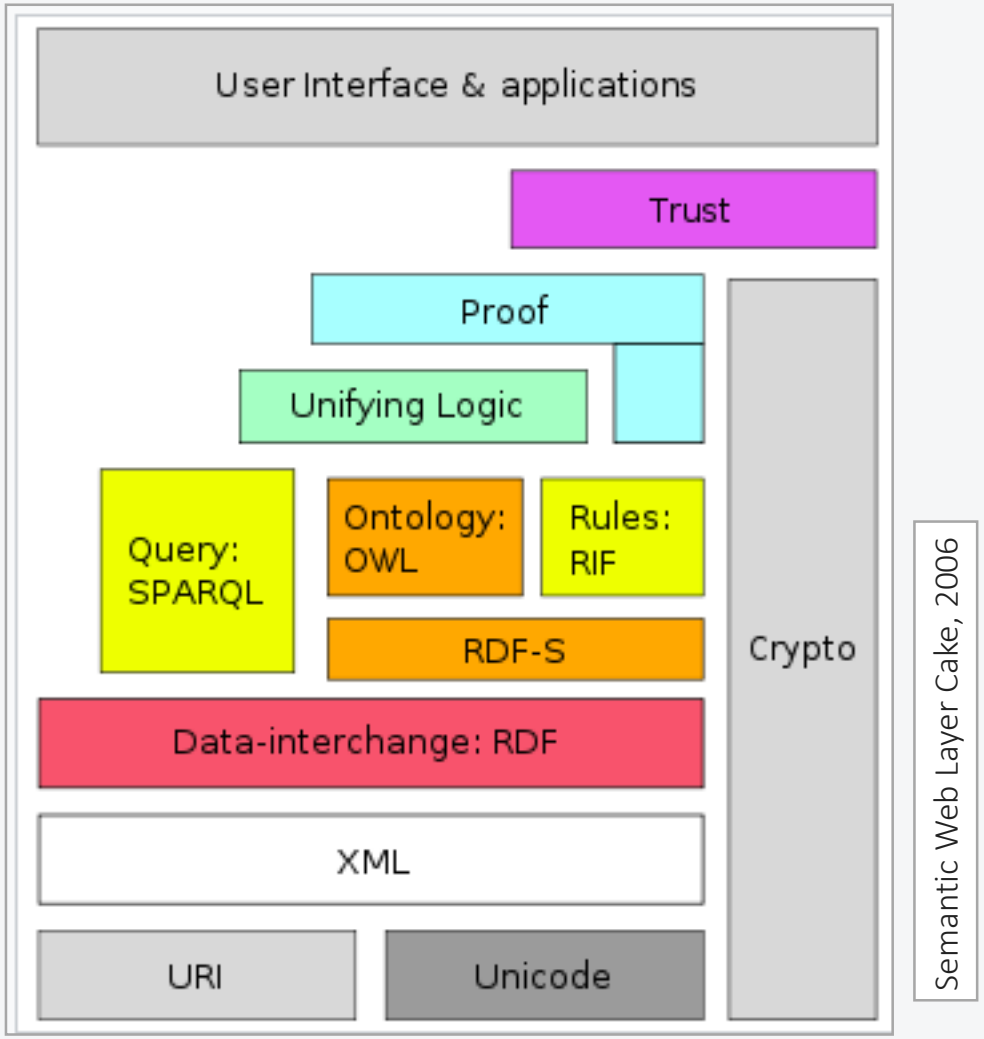
eccenca
mastering complexity

The Invention of the Internet

eccenca
mastering complexity

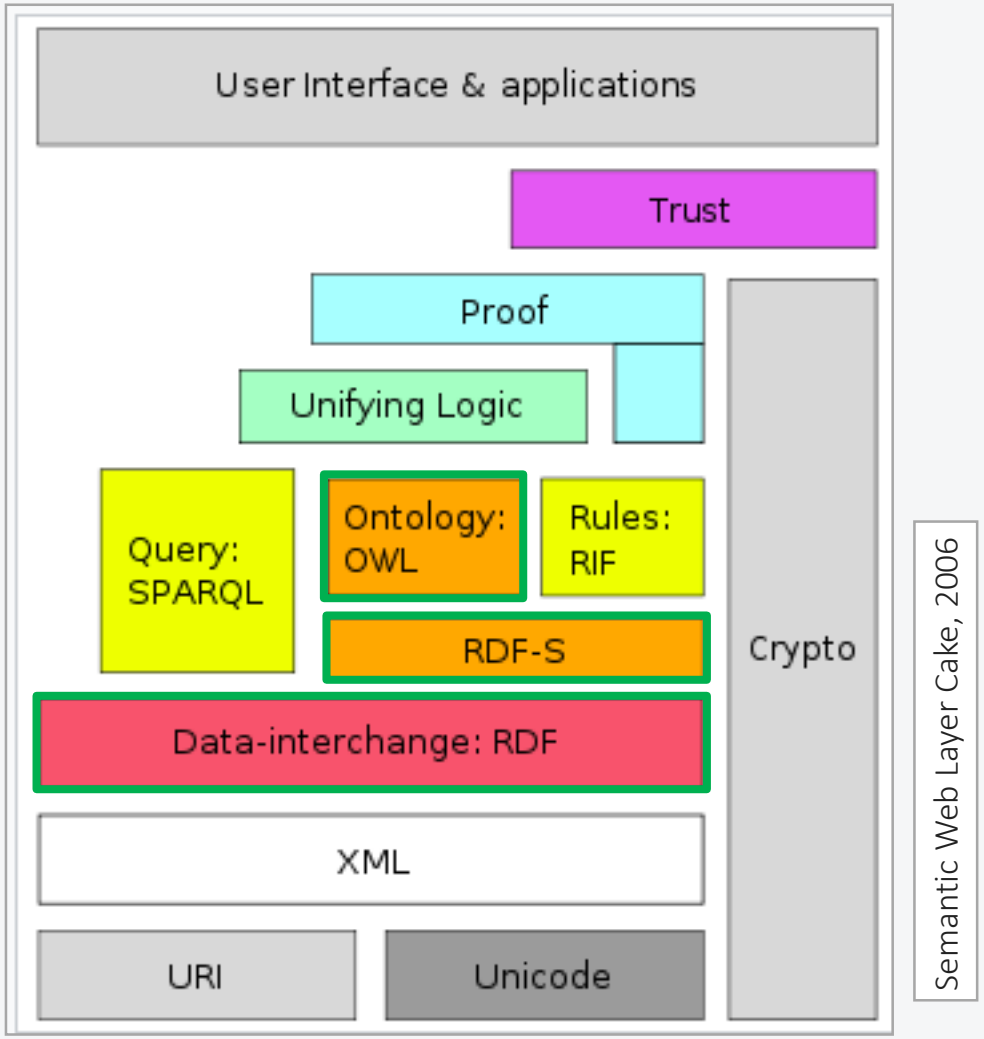


(Semantic Web) Standards



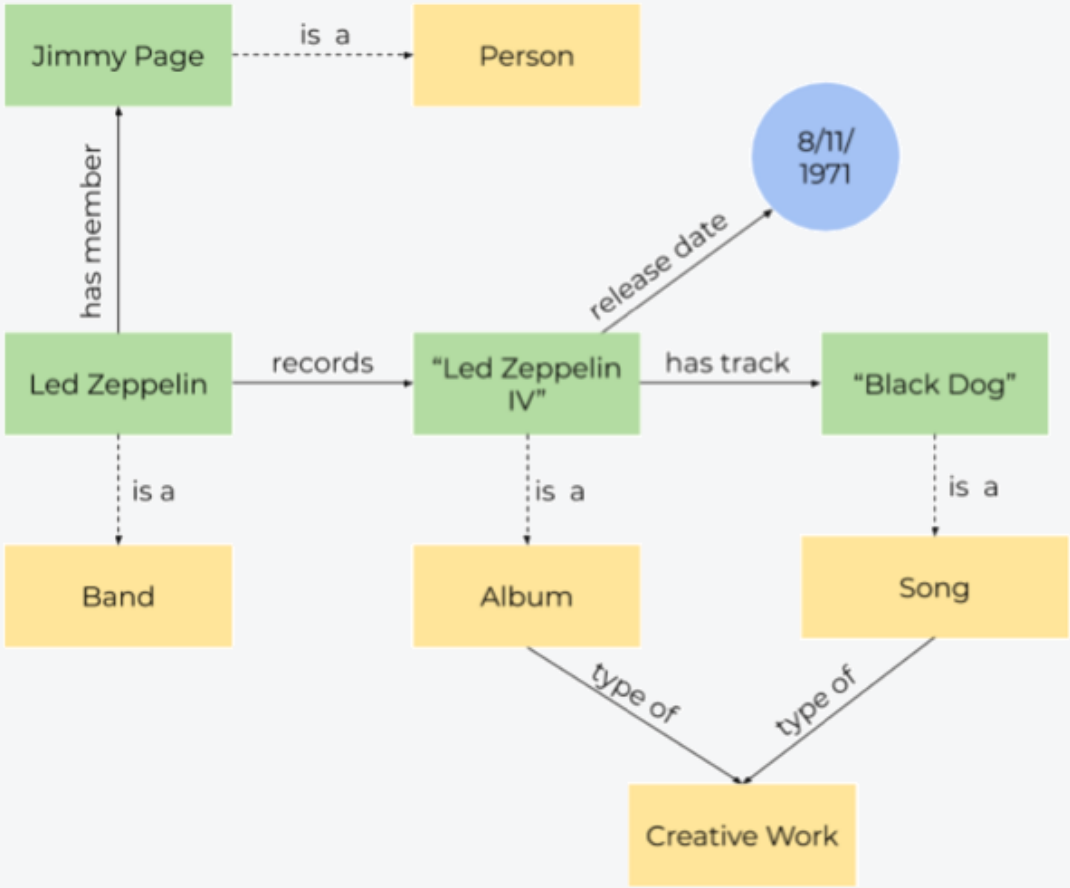
Semantic Web Layer Cake, 2006

(Semantic Web) Standards



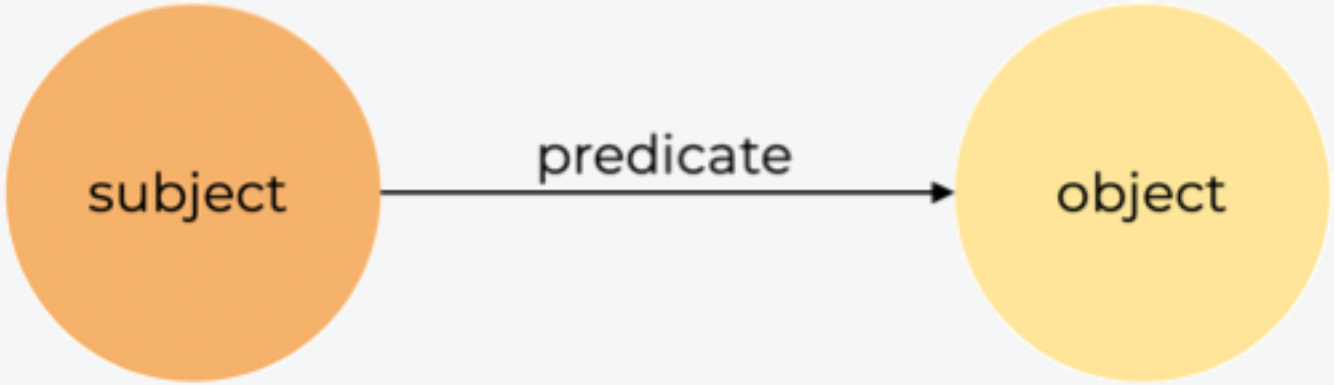
Semantic Web Layer Cake, 2006

RDF, RDFS, OWL... WTF...

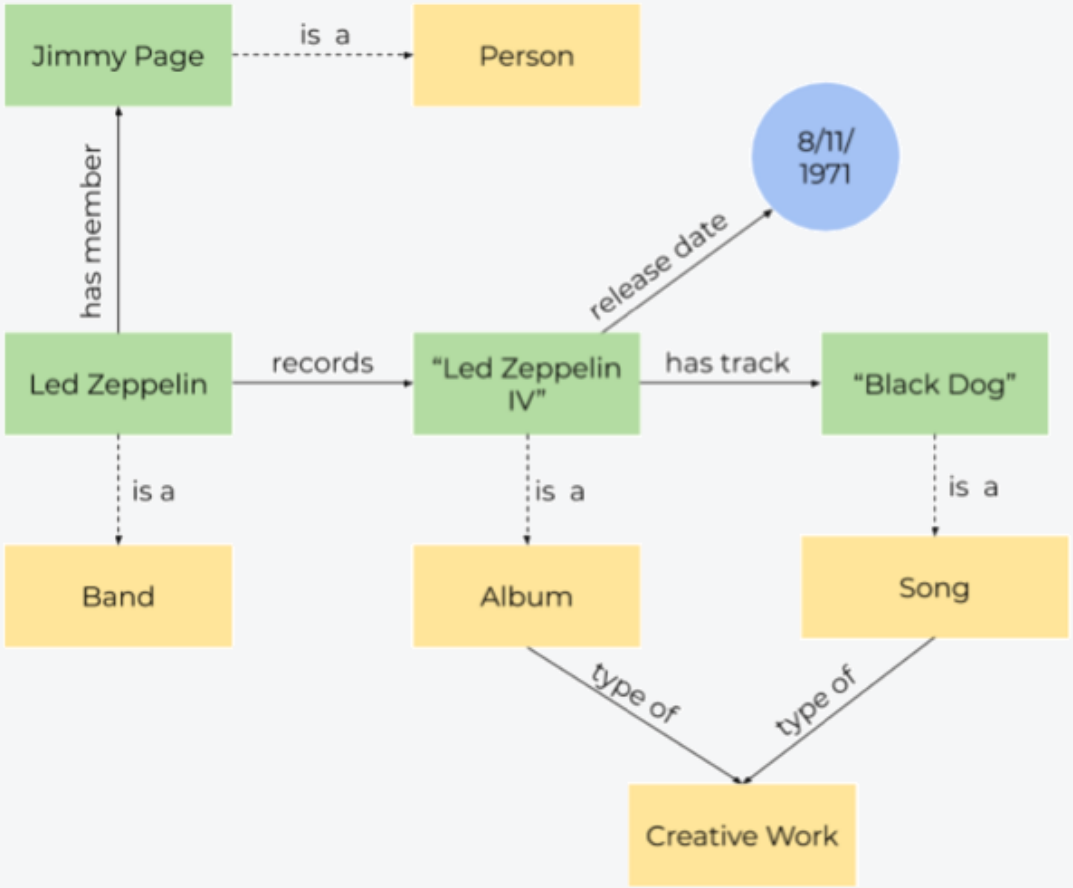


taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework

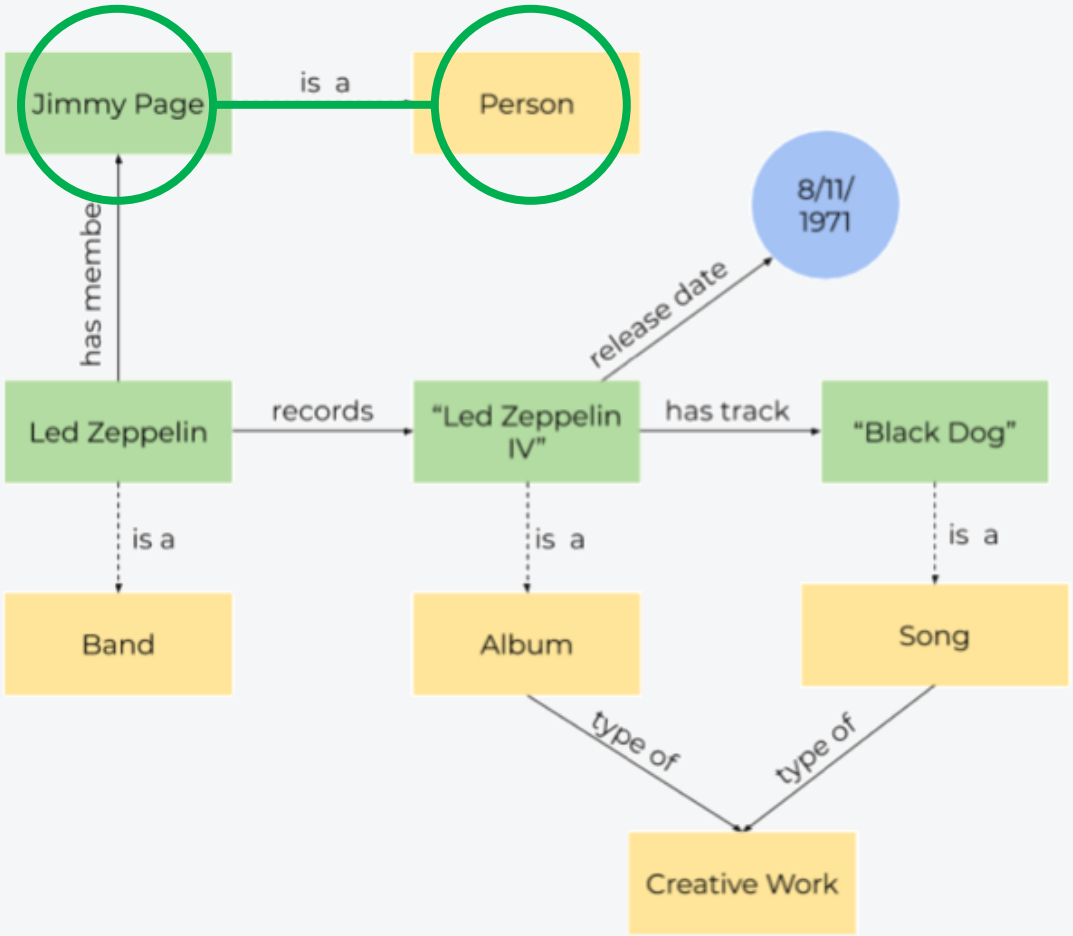


RDF - Resource Description Framework



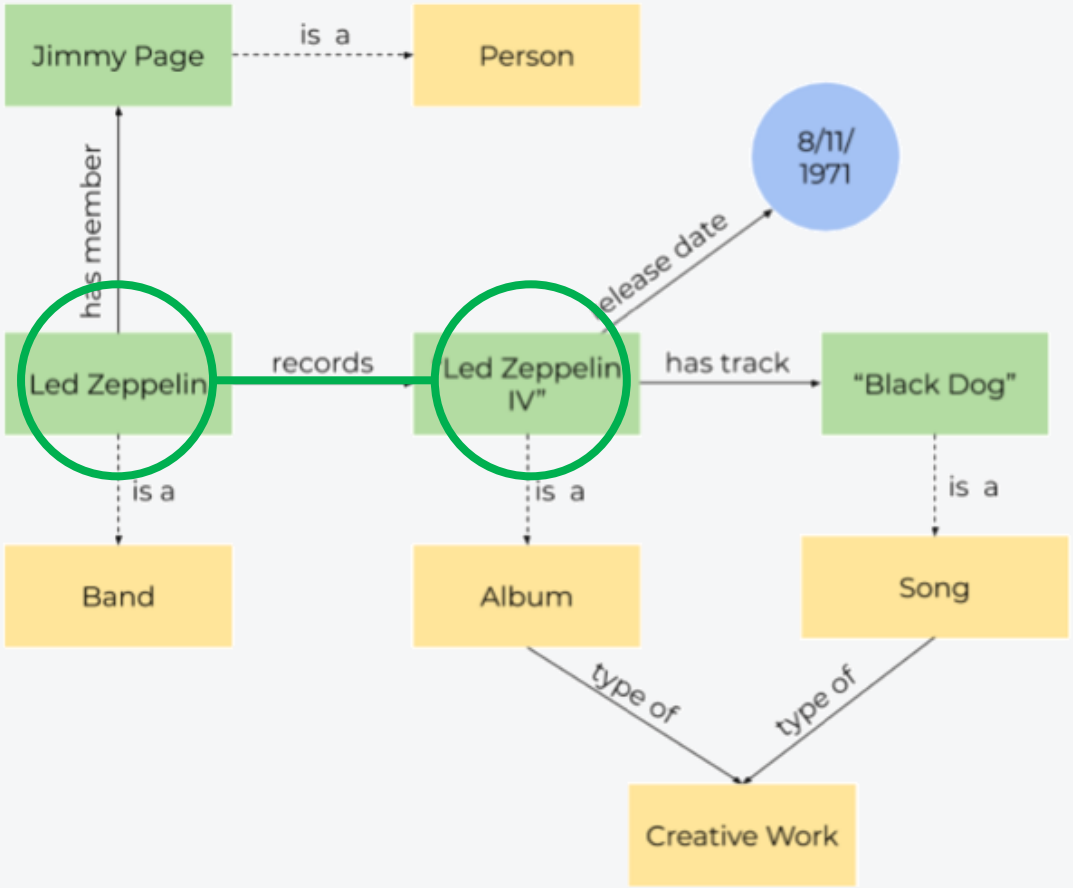
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework



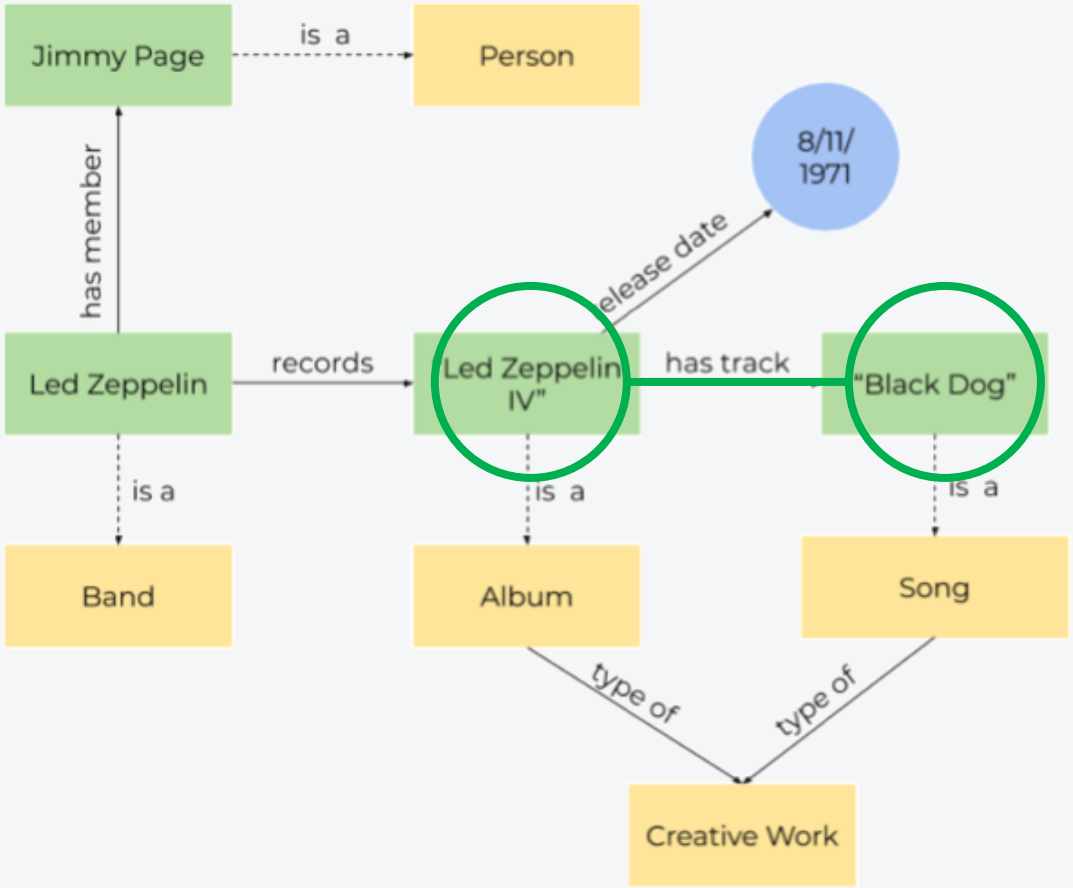
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework



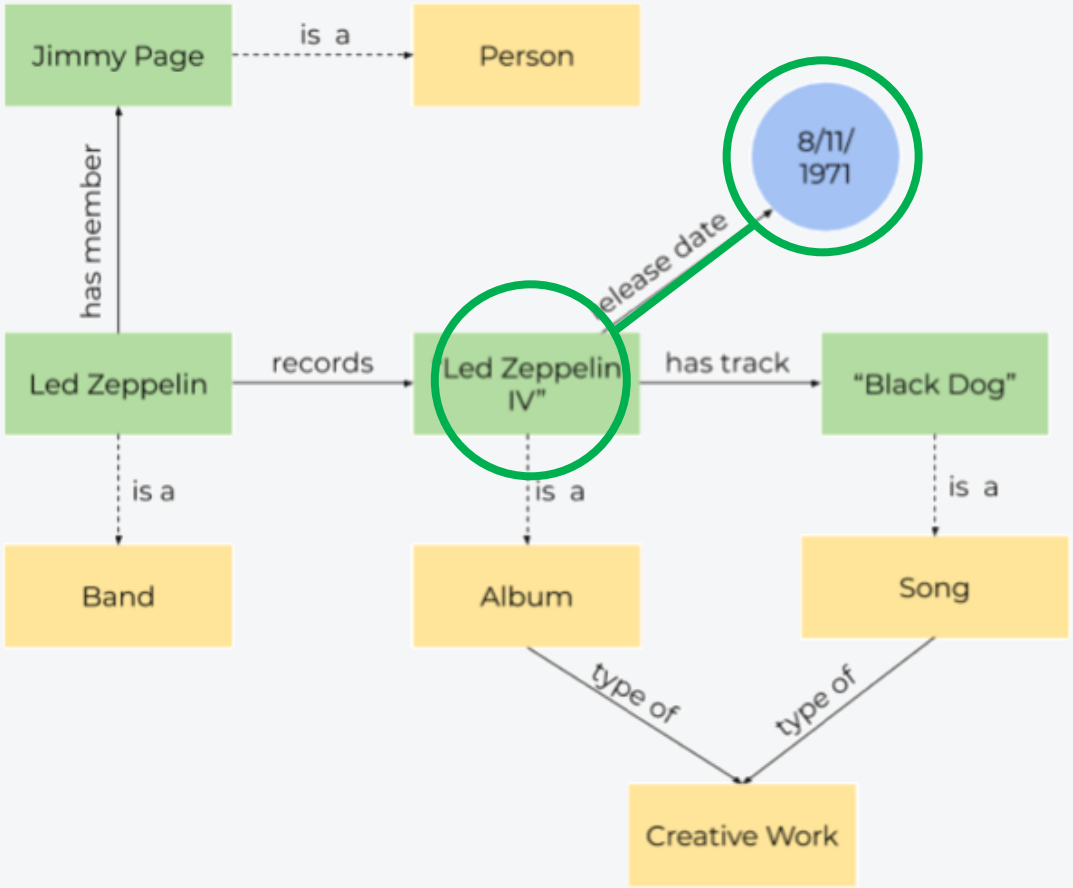
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework



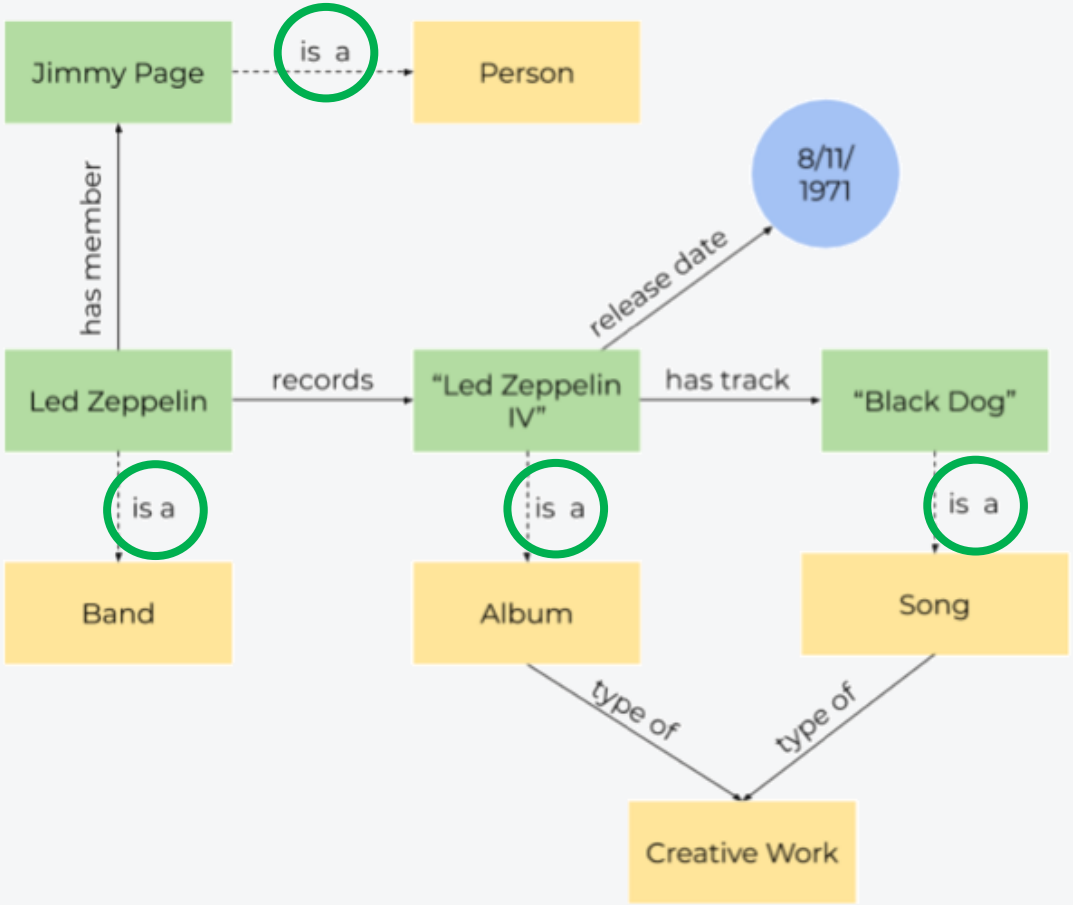
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework



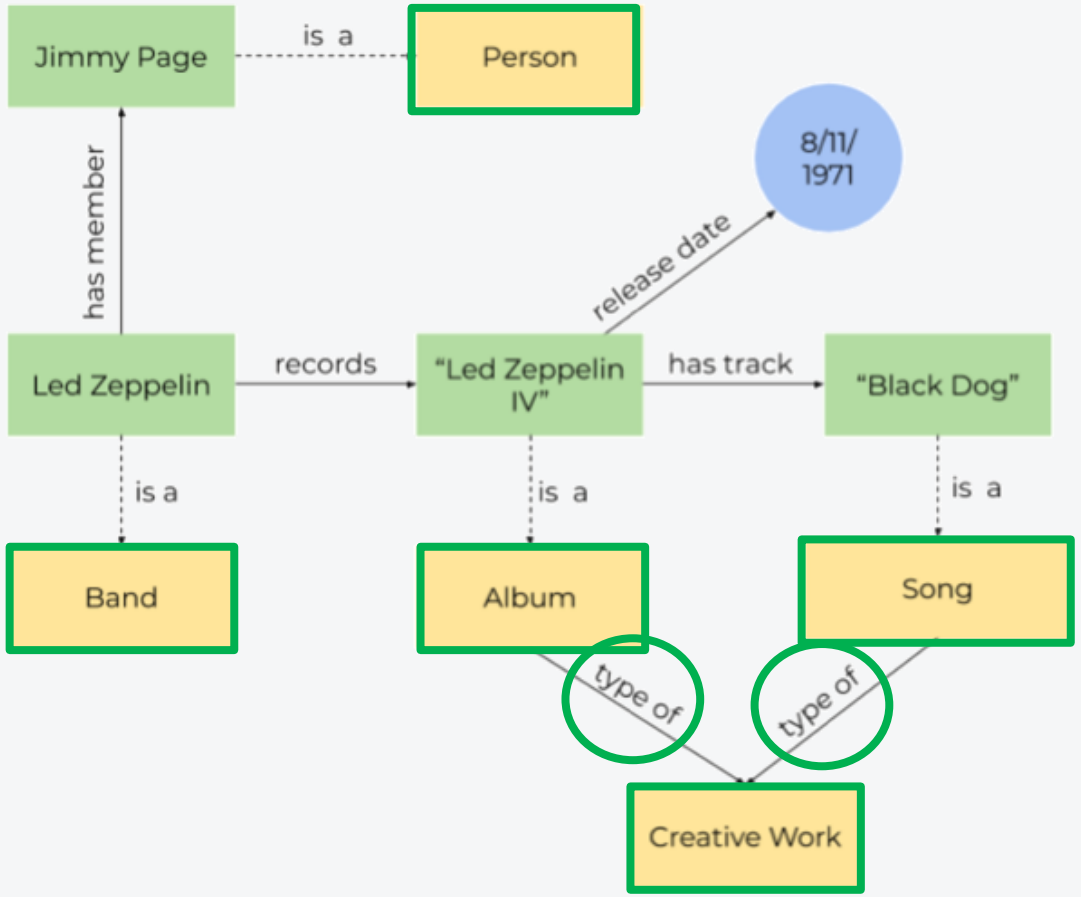
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF - Resource Description Framework



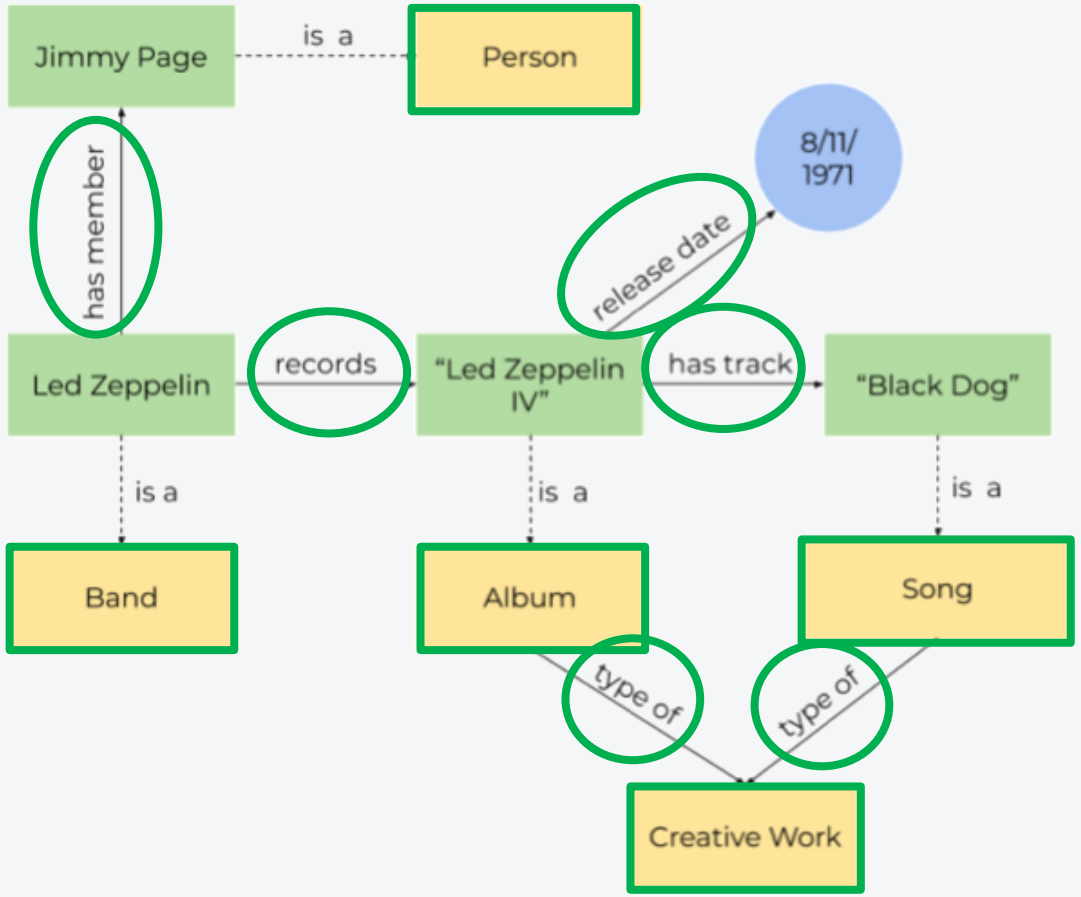
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF**S** - RDF Schema



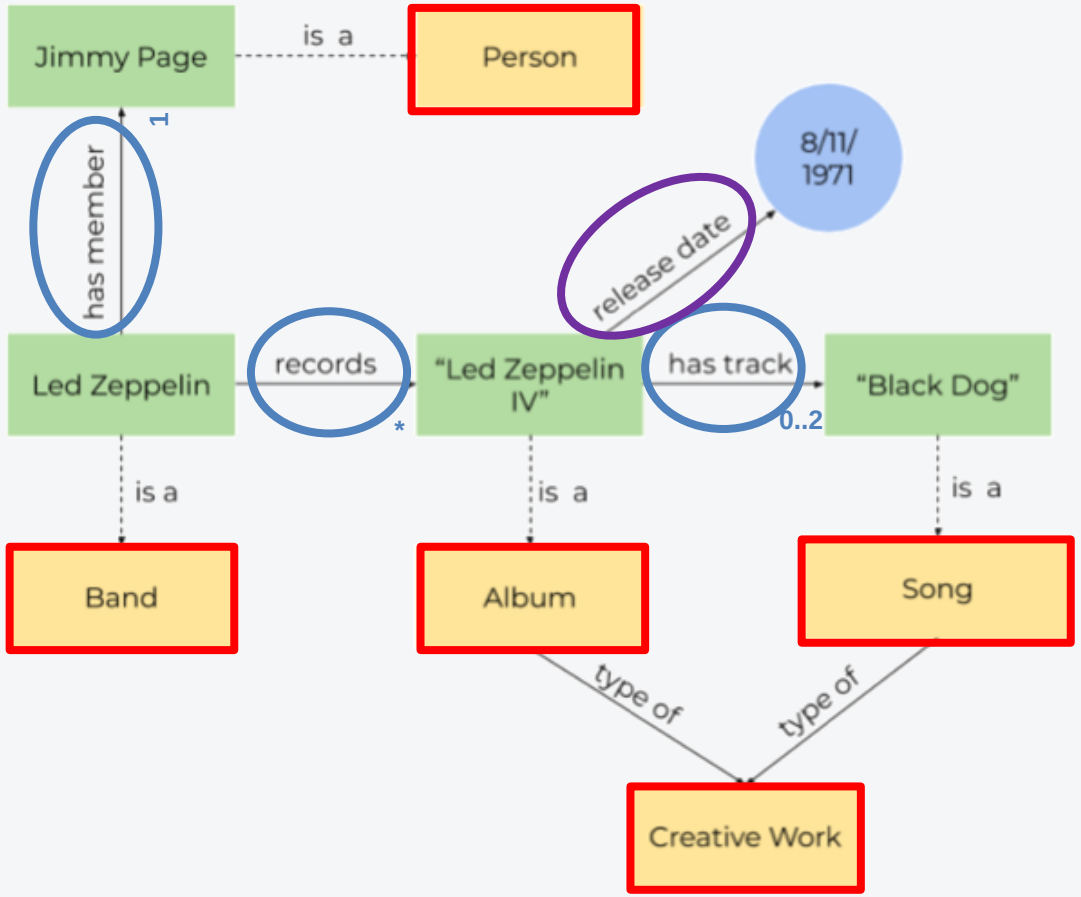
taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

RDF**S** - RDF Schema



taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

OWL - Web Ontology Language



taken from https://zenodo.org/record/3898519?hsCtaTracking=870ee1df-0d7b-4e50-8767-b79d2a8295f0%7C40a224e4-9b70-4886-9520-95036d56f67e#.XuoMVy97HRY

☆ ☆ ☆ ☆ ☆ Linked (Open) Data Principles

The Growth of the LOD Graph

Linked Open Data, 2007

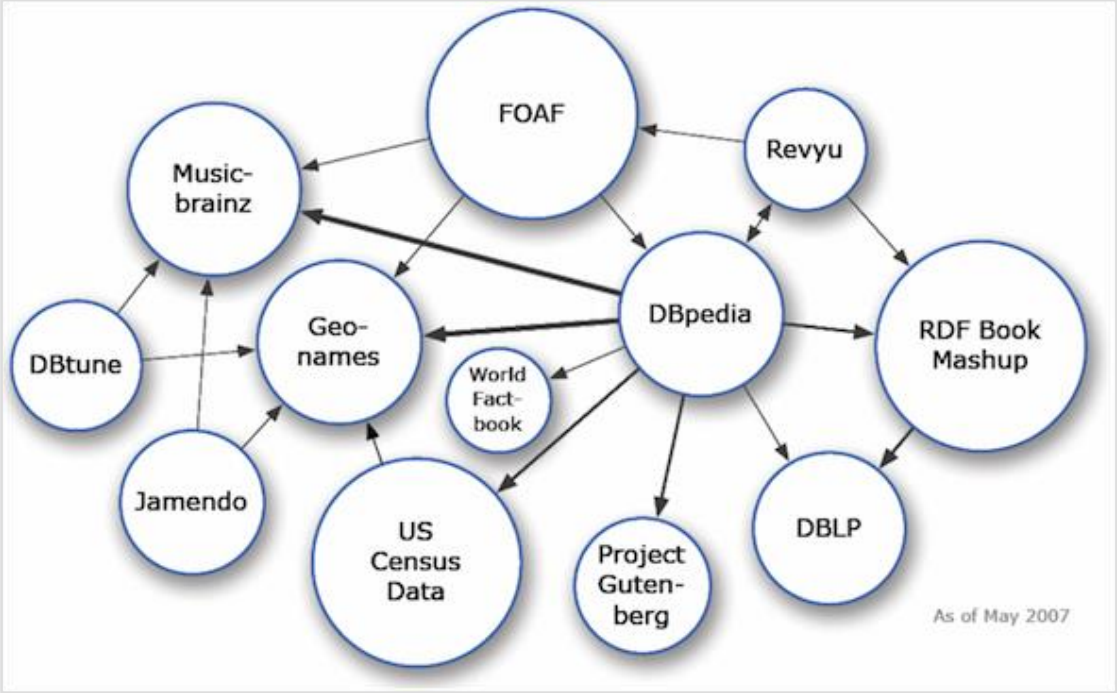
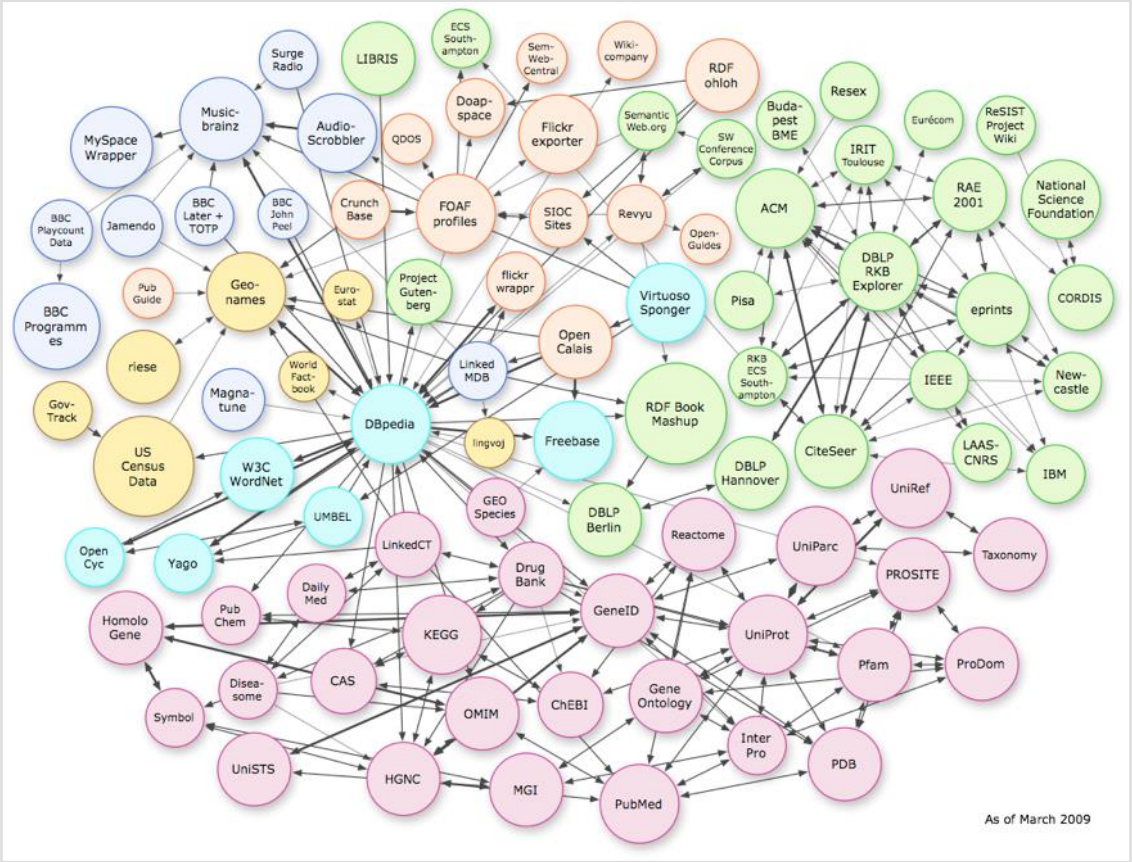


Image shows datasets that have been published in Linked Data format, by contributors to the Linking Open Data community project and other individuals and organizations.

The Growth of the LOD Graph

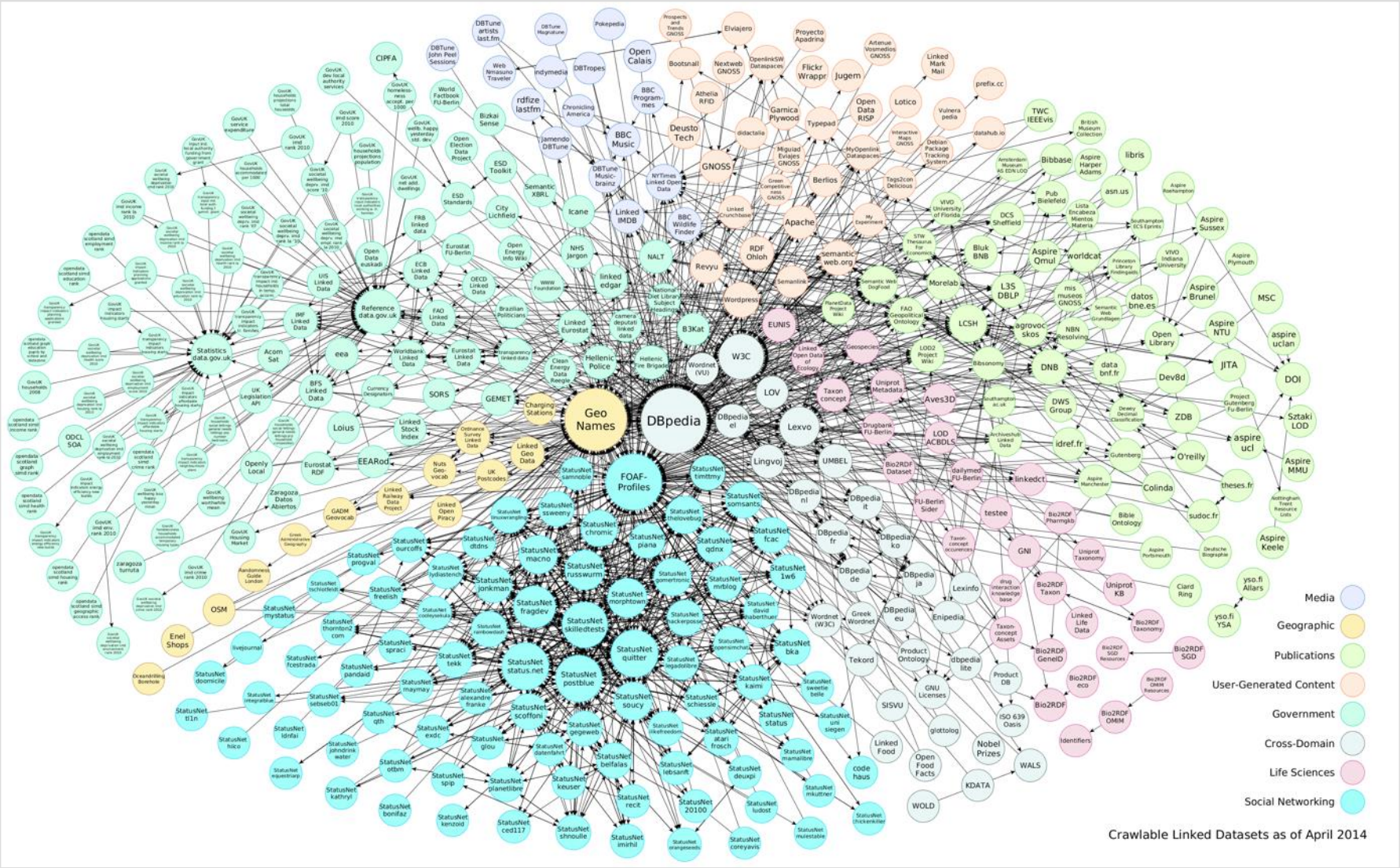
Linked Open Data, 2009

eccenca
mastering complexity



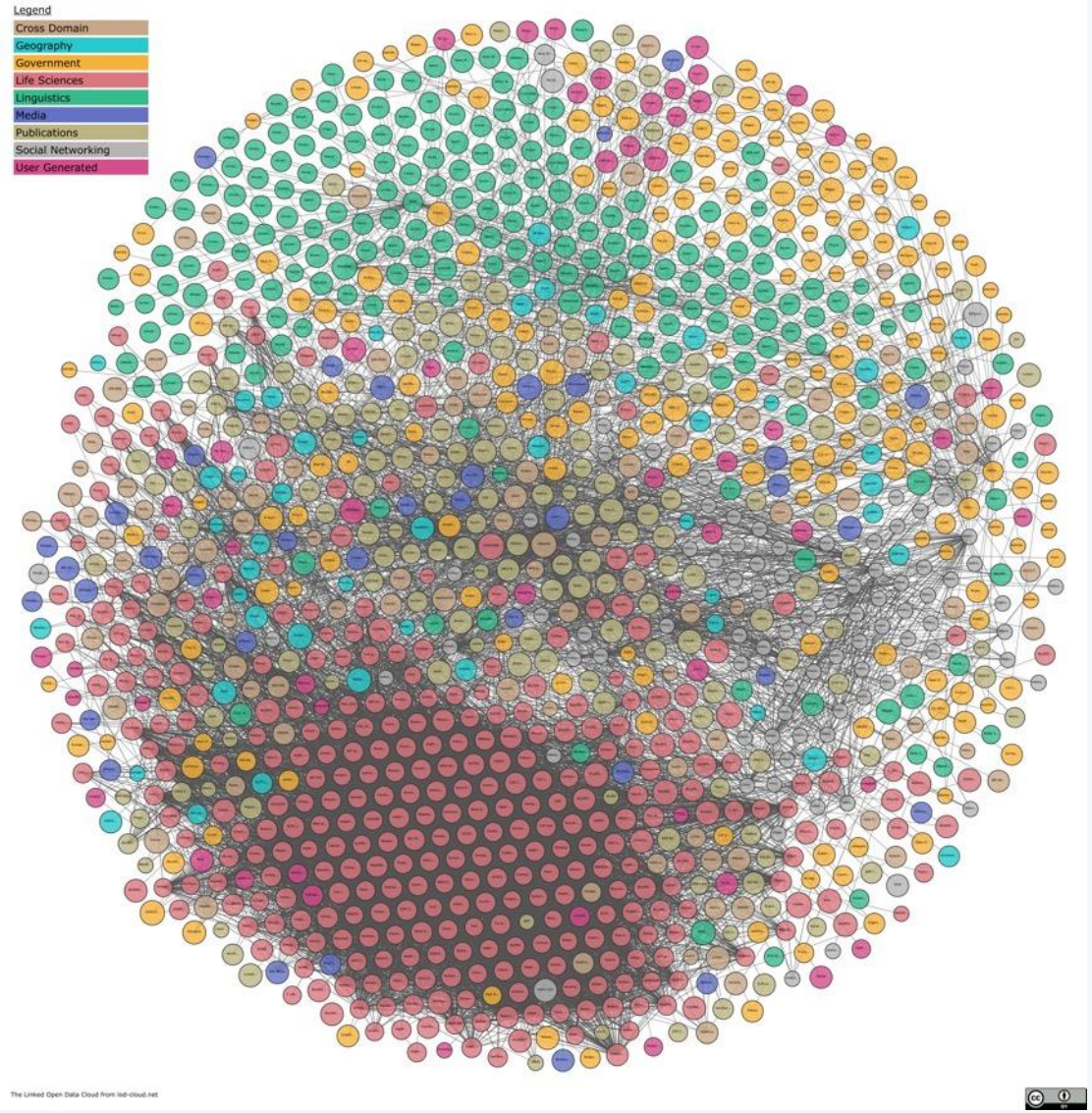
The Growth of the LOD Graph

Linked Open Data, 2014



The Growth of the LOD Graph

Linked Open Data, 2020



Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. http://lod-cloud.net/

ONE Record Insights

Part 2

Let’s craft an ontology

eccenca
mastering complexity

Introduction

Ontologies are shared, structured conceptualizations of domains.

Conceptualization:

- abstract, simplified view of the domain represented for some purpose
- captures objects, concepts and the relationships that hold among them

Structured:

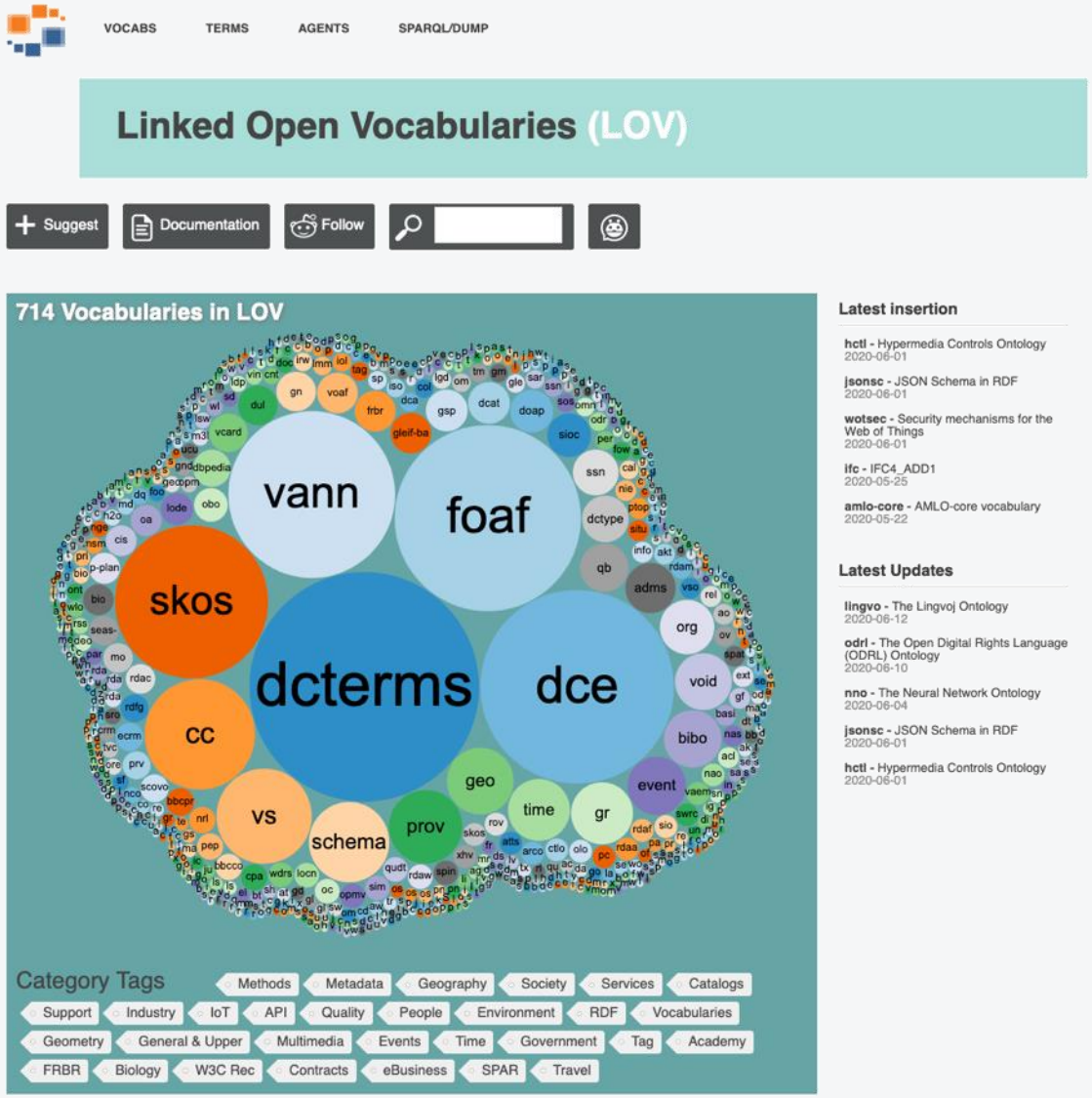
- According to standardized modelling paradigms like OWL and SKOS

Shared:

- Agreed upon by domain experts and acting as an extensible standard for data representation in the domain

Domains vary from most basic things (Persons, Places, Activities) to specialist areas like supply chains or manufacturing

Wait … “Google” for prior Work and / or Inspiration … LOV!



https://lov.linkeddata.es/dataset/lov/

Engineering Methodology

eccenca
mastering complexity

- No one correct way to ontology engineering, depends on the domain and application

- Generally:
  - Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.
  - Most likely, they are nouns (objects) and relations (verbs) in sentences that describe your domain.
  - Additionally, the scope of use and application determine which objects and relations we need to define

Necessarily iterative – created ontologies have to be discussed, applied and refined

Scope and requirements

- Scope depends on domain, application and use of the ontology
- Is there already data available, or are we modelling “on the green field”?
- If there is data available, what is it used for?
- Are we trying to create an easy to understand, logical data model that fits the domain?
- Are we trying to adhere to existing structures for legacy compatibility?
- How much meta data is needed?
- What other data sources are relevant?
- Are there already ontologies out there, that model the domain?

- Useful technique: **Ontology competency questions**
- Create a catalog of questions your ontology should answer
- After each development milestone / iteration, try to answer the questions by:
  - Creating example instance data according to your ontology that you can query or browse with the goal to answer the question

Source material

- Source material can be wide-ranging:
  - Existing, relational data sources (Excel, databases etc)
  - Slide decks that present the domain / train new users on existing concepts
  - Company taxonomy databases
  - Product catalogs
  - Books on the topic in question
  - Expert interviews

- Often a combination of these:
  - Receive a spreadsheet and a slide deck
  - Interview a domain expert (or be one yourself)
  - Create an initial ontology, then iterate with the expert
  - Receive more documentation material etc.

Example Domain: Product Data

Our company's physical products and services are managed by product managers. The practice of product management is distributed across all department in which they have a direct report. Products are structured into Categories for which expertise exists in product managers and responsible departments.

E.g. our P516-8211068 named Film Multiplexer Rheostat Warp is managed by Kristen Bauers in Procurement.

Example Domain: Product Data

eccenca
mastering complexity

Our company physical products and services are managed by product managers. The practice of product management is distributed across all department in which they have a direct report. Products are structured into Categories for which expertise exists in product managers and responsible departments.

E.g. our P516-8211068 named Film Multiplexer Rheostat Warp is managed by Kristen Bauers in Procurement.

1. Find concepts

Example Domain: Product Data

eccenca
mastering complexity

Our company physical products and services are managed by product managers. The practice of product management is distributed across all department in which they have a direct report. Products are structured into Categories for which expertise exists in product managers and responsible departments.

E.g. our P516-8211068 named Film Multiplexer Rheostat Warp is managed by Kristen Bauers in Procurement.

1. Find concepts
2. Find relationships

Example Domain: Product Data

eccenca
mastering complexity

Our company physical products and services are managed by product managers. The practice of product management is distributed across all department in which they have a direct report. Products are structured into Categories for which expertise exists in product managers and responsible departments.

E.g. our *P516-8211068* named *Film Multiplexer Rheostat Warp* is managed by *Kristen Bauers* in *Procurement*.

1. Find concepts
2. Find relationships
3. Find *instances* (typically in data sources, not in documentation material)

Example Domain: Product Data

eccenca
mastering complexity

Our company physical products and services are managed by product managers. The practice of product management is distributed across all department in which they have a direct report. Products are structured into Categories for which expertise exists in product managers and responsible departments.

E.g. our *P516-8211068* named *Film Multiplexer Rheostat Warp* is managed by *Kristen Bauers* in *Procurement*.

1. Find concepts
2. Find relationships
3. Find *instances* (typically in data sources, not in documentation material)
4. Find attributes

Manufacturing: Define classes

eccenca
mastering complexity

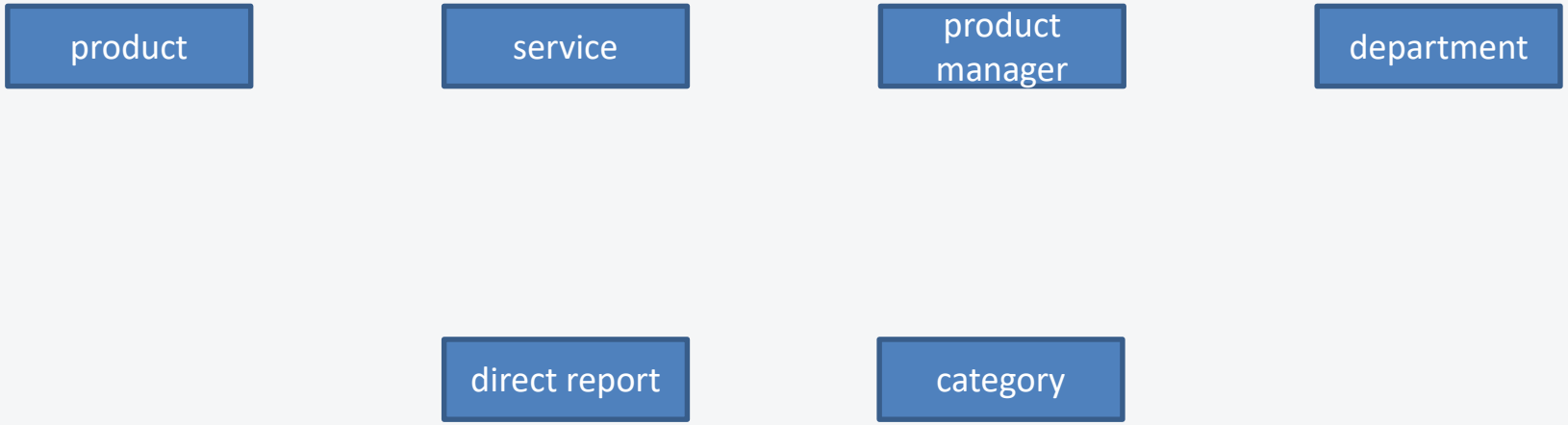
- Methodologies:
  - top-down: Define most general classes first, then specialize
  - bottom-up: Define most specific classes, then group them to generalize
  - middle-out: Define most useful, salient classes first, then specialize and generalize as appropriate

- Class candidates: product, service, product manager, department, direct report, categories

- Starting with these classes, often helpful to draw a diagram, presenting classes as boxes and relations as lines

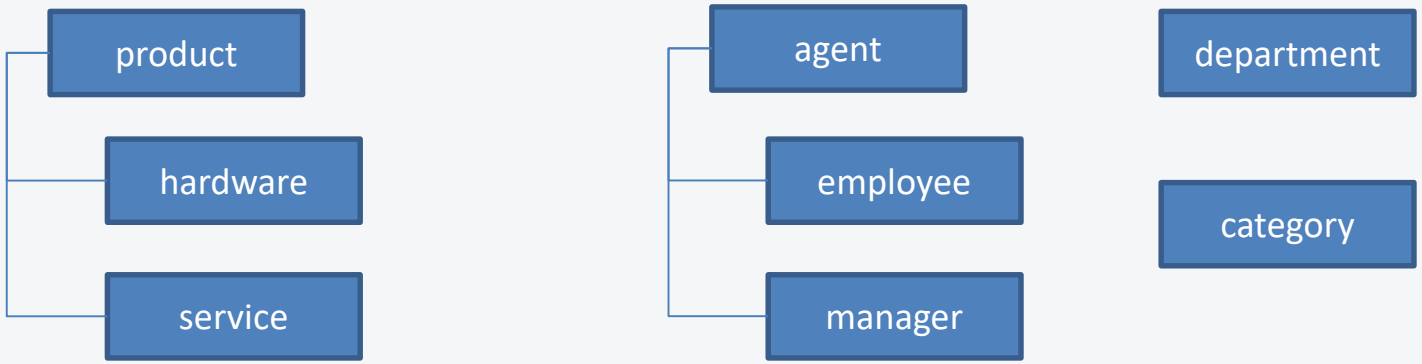
Manufacturing: Define classes

- Class candidates: product, service, product manager, department, direct report, categories



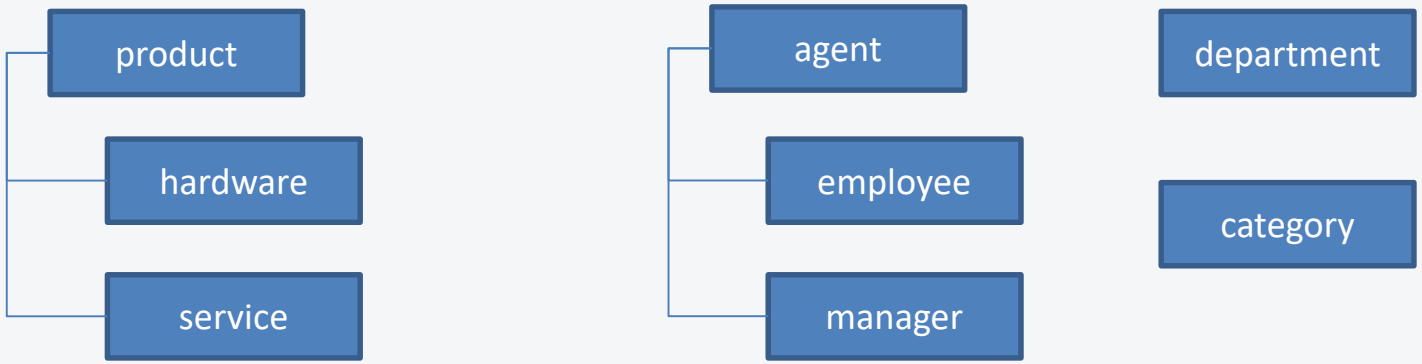
Manufacturing: Define classes

- Class candidates: product, service, product manager, department, direct report, categories



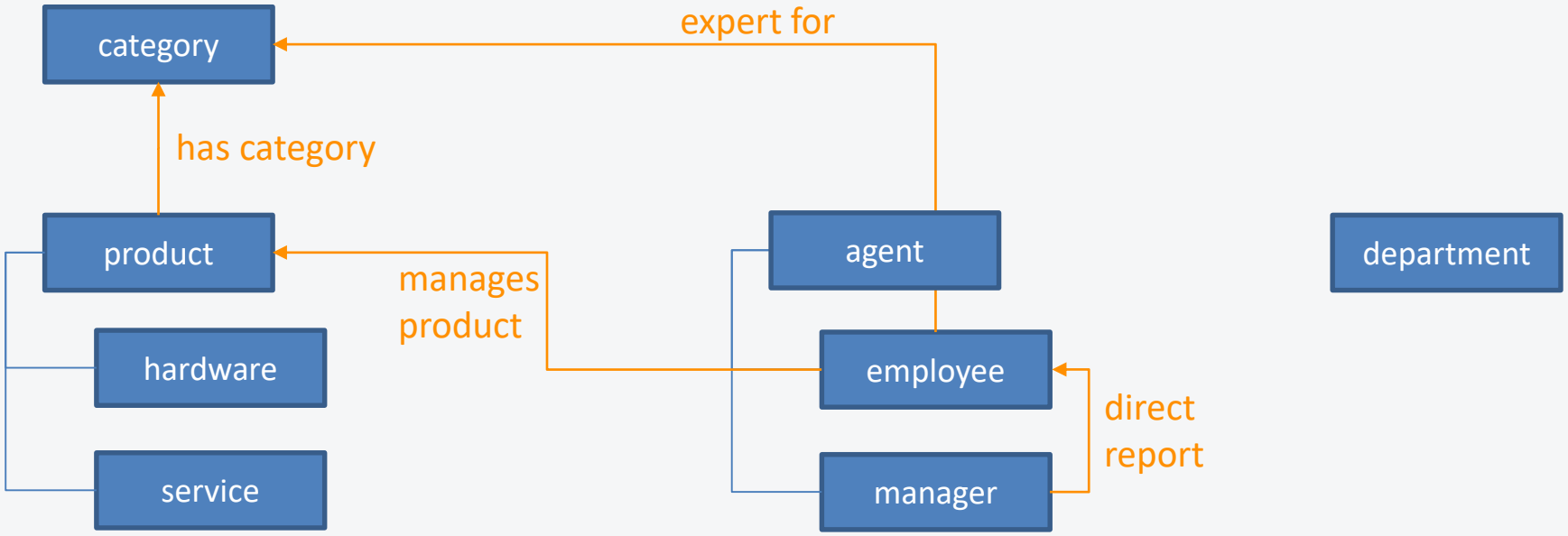
Manufacturing: Define classes

- Relation candidates: managed by, has (direct report), structured (into category), expert in, responsible for



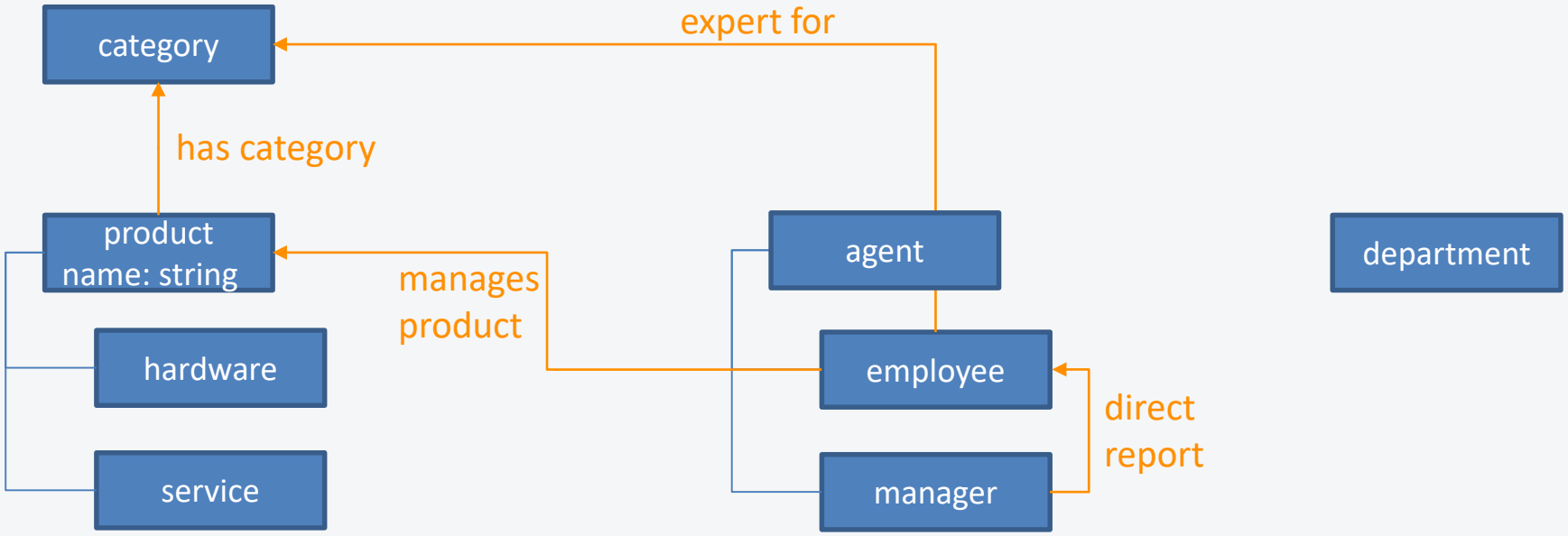
Manufacturing: Define classes

- Relation candidates: managed by, has (direct report), structured (into category), expert in, responsible for



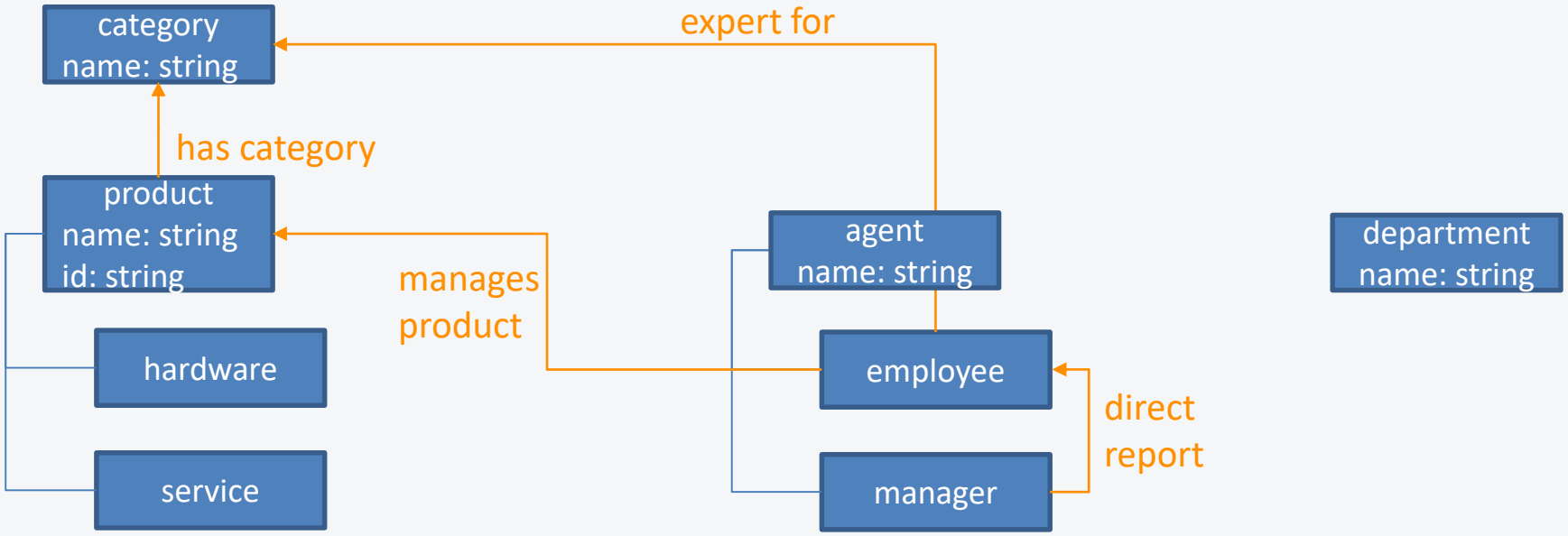
Manufacturing: Define classes

- Attribute candidates: named



Manufacturing: Define classes

- Attribute candidates: named



Manufa

Protégé Ontology editor

prod-vocab (http://ld.company.org/prod-vocab/0.0.1/) : [/Users/rene/Documents/git/gitlab.eccenca.com/devops/product-data-generator/products.ttl]

prod-vocab (http://ld.company.org/prod-vocab/0.0.1/)

Search...

Active Ontology | Entities | Individuals by class | OWLViz | Individual Hierarchy Tab | OntoGraf

Annotation properties | Datatypes | Individuals

Classes | Object properties | Data properties

Class hierarchy: Employee

Asserted

- owl:Thing
  - Address
  - Agent
    - Employee
      - Manager
  - Department
  - Product
    - Hardware
    - Service
  - 'Product Category'

Object property hierarchy: 'eligible for'

Asserted

- owl:topObjectProperty
  - 'eligible for'
  - 'expert for'
  - 'has address'
  - 'has category'
  - 'has direct report'
  - 'has manager'
  - 'has product manager'
  - 'manages product'
  - 'member of'
  - 'responsible for'

'eligible for' — http://ld.company.org/prod-vocab/eligibleFor

Annotations | Object Property Usage

Annotations: 'eligible for'

Annotations

rdfs:label

eligible for

rdfs:isDefinedBy

http://ld.company.org/prod-vocab/

Charac

- Functional
- Inverse functic
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Description: 'eligible for'

Equivalent To

SubProperty Of

owl:topObjectProperty

Inverse Of

Domains (intersection)

Service

Ranges (intersection)

Hardware

Disjoint With

SuperProperty Of (Chain)

No Reasoner set. Select a reasoner from the Reasoner menu

Show Inferences

eccenca

mastering complexity

Manufacturing

RDF Turtle Syntax

products.ttl — Untitled (Workspace)

orgmap.xml | products.ttl

```
@prefix : <http://ld.company.org/prod-vocab/> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

#@base <http://ld.company.org/prod-vocab/> .

<http://ld.company.org/prod-vocab/> rdf:type owl:Ontology ; ...
#################################################################
#    Object Properties
#################################################################

### http://ld.company.org/prod-vocab/eligibleFor
:eligibleFor rdf:type owl:ObjectProperty ;
             rdfs:subPropertyOf owl:topObjectProperty ;
             rdfs:domain :Service ;
             rdfs:range :Hardware ;
             rdfs:isDefinedBy <http://ld.company.org/prod-vocab/> ;
             rdfs:label "eligible for" .


### http://ld.company.org/prod-vocab/expertFor
:expertFor rdf:type owl:ObjectProperty ;
           rdfs:subPropertyOf owl:topObjectProperty ;
           rdfs:domain :Employee ;
           rdfs:range :ProductCategory ;
           rdfs:isDefinedBy <http://ld.company.org/prod-vocab/> ;
           rdfs:label "expert for" .


### http://ld.company.org/prod-vocab/hasAddress
:hasAddress rdf:type owl:ObjectProperty ;
            rdfs:subPropertyOf owl:topObjectProperty ;
            rdfs:domain :Agent ;
            rdfs:range :Address ;
            rdfs:isDefinedBy <http://ld.company.org/prod-vocab/> ;
            rdfs:label "has address" .
```

develop* | 0 0 | Ln 1, Col 1 | Spaces: 6 | UTF-8 | LF | TURTLE | [off] | 1

eccenca
mastering complexity

Manufacturing: Formalize

concepts.xlsx – Auf "meinem Mac" gespeichert

id	label	comment	parentClass	isRequiredBy	issued	modified
Cable	Cable	a cable		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
CableTest	Cable Test	a cable test		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
MeasurementSeries	Measurement Series	a Measurement Series		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
QualityCode	Quality Code	a Quality Code		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
Plant	Plant	a Plant		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
Material	Material	a Material		https://vocab.rfsworld.com/cable-testing/	2018-03-26	2018-03-26
CableSection	Cable Section	a Cable Section		https://vocab.rfsworld.com/cable-testing/	2018-03-27	2018-03-27
Measurement	Measurement	a Measurement		https://vocab.rfsworld.com/cable-testing/	2018-04-14	2018-04-14
Attenuation	Attenuation	a Attenuation		https://vocab.rfsworld.com/cable-testing/	2018-04-14	2018-04-14
Model	Model	a Model		https://vocab.rfsworld.com/cable-testing/	2018-04-26	2018-04-26

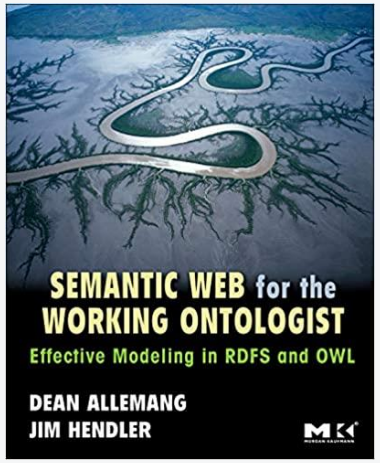
concepts.xlsx – Auf "meinem Mac" gespeichert

id	label	domain	range	comment	cardinality	isRequiredBy	superProperty	issued	m
plant	plant	Cable	Plant	having a plant		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
material	material	Cable	Material	having a material		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
measurementSeries	measurement series	CableTest	MeasurementSeries	having a measurement		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
cableTest	cable test	Cable	CableTest	having a test		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
qCode	quality code	CableTest	QualityCode	having a quality code		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
cableSection	cable section	Cable	CableSection	having a cable section		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
cableTest	cable test	CableSection	CableTest	having a test		https://vocab.rfsworld.com/cable-testing/		2018-03-26	20
measurementSeries	measurement series	Attenuation	MeasurementSeries	having a measurement series		https://vocab.rfsworld.com/cable-testing/		2018-04-03	20
measurement	measurement	MeasurementSeries	Measurement	having a measurement		https://vocab.rfsworld.com/cable-testing/		2018-04-12	20
measurementOf	measurement of	Measurement	MeasurementSeries	belonging to a measurement series		https://vocab.rfsworld.com/cable-testing/		2018-04-14	20
attenuation	attenuation	CableTest	Attenuation	having a attenuation		https://vocab.rfsworld.com/cable-testing/		2018-04-14	20
model	model	Cable	Model	having a model		https://vocab.rfsworld.com/cable-testing/		2018-04-26	20
model	model	Material	Model	having a model		https://vocab.rfsworld.com/cable-testing/		2018-04-26	20

using (Excel) Template and automatic mapping into RDFS/OWL

Links

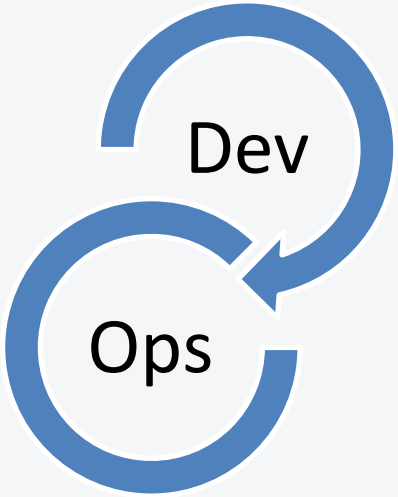
- Book: Dean Allemang, Jim Hendler - Working Ontologist
- Web: Leigh Dodds, Ian Davis - Linked Data Pattern



Git / CI CD of ontology

- Treat ontologies like code -> 

- Apply automatic integration and test procedures
- Automatic generation of visualization and documentation artifacts



Bamboo | My Bamboo | Build | Deploy | More | Create | Search

Build projects / Vocabularies

eccgeo-dataset

Run | Actions

Plan summary | Recent failures | History | Tests | Issues | Status Icons

Plan summary

Showing Last 25 builds

Current activity

No builds are currently running.

Recent history

#15	Changes by Martin Brümmer	1 month ago	351 passed
#14	Manual run by Martin Brümmer	2 months ago	351 passed
#13	Changes by Martin Brümmer and Priyanka Dank	2 months ago	Testless build
#12	Changes by Martin Brümmer	5 months ago	351 passed
#11	Changes by René Pietzsch	9 months ago	2 of 348 failed
#10	Changes by Lukas Eipert	10 months ago	348 passed
#9	Changes by Martin Brümmer	11 months ago	348 passed
#8	Changes by René Pietzsch	1 year ago	2 of 348 failed
#7	Changes by René Pietzsch	1 year ago	2 of 348 failed
#6	Changes by René Pietzsch	1 year ago	2 of 348 failed

Latest build | Last successful build

Feed for all builds or just the failed builds.

Plan statistics

10 builds

50% successful

1m average duration

Bamboo | My Bamboo | Build | Deploy | More | Create | Search

Build projects / Vocabularies / eccgeo-dataset

Build #15

Run | Actions

#15 was successful – Changes by Martin Brümmer

Stages & jobs

pre stage

inject git version

main stage

make bamboo-build

Summary | Tests | Commits | Artifacts | Logs | Metadata | Build Times | Issues

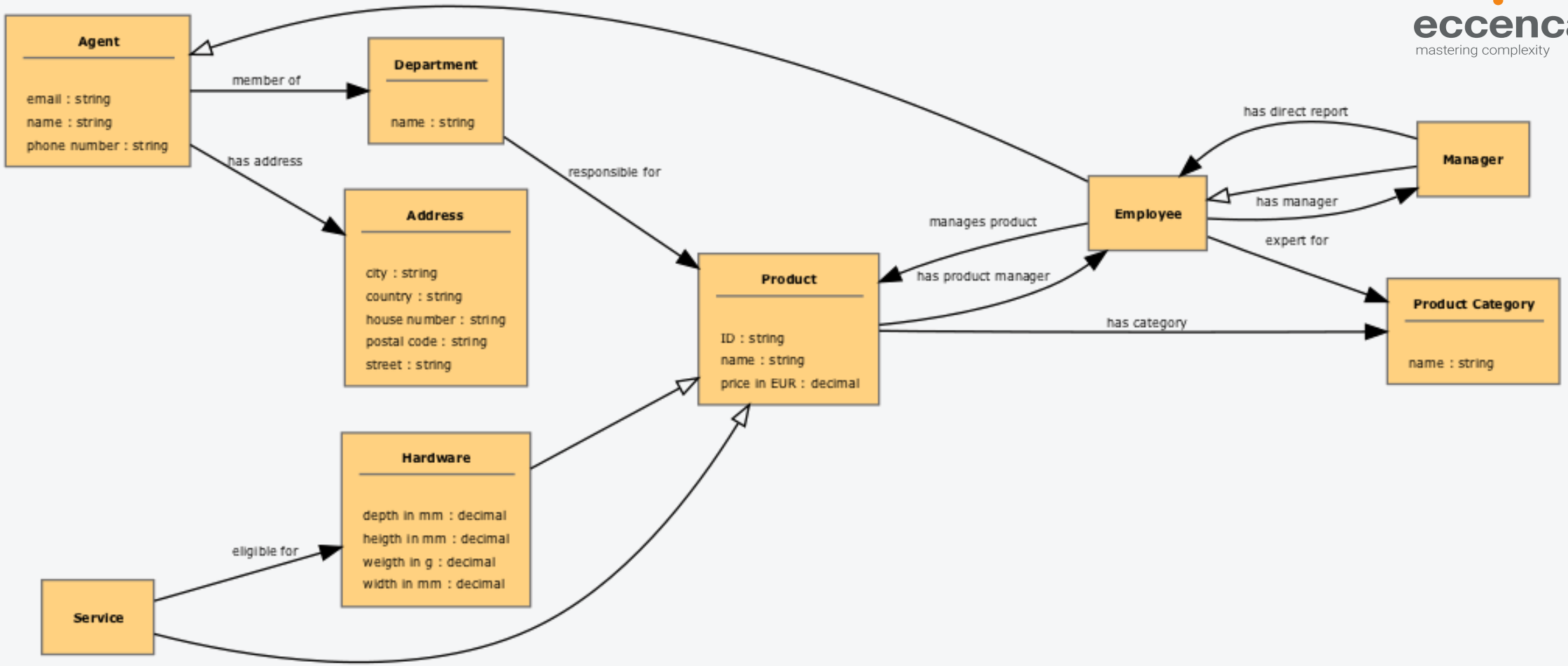
Logs

The following logs have been generated by the jobs in this plan. Expand all | Collapse all

Job	Logs
inject git version pre stage	Download or View
make bamboo-build main stage	Download or View

```
17-Nov-2017 10:33:33 [INFO  SimpleTestExecutorMonitor] Test 335/338 returned Errors: 0 / Prevalence:
                     251. Test: https://ns.eccenca.com/geo/currencyName is missing proper range
17-Nov-2017 10:33:33 [INFO  SimpleTestExecutorMonitor] Test 336/338 returned Errors: 0 / Prevalence:
                     156. Test: https://ns.eccenca.com/geo/postalCodeRegex is missing proper range
17-Nov-2017 10:33:33 [INFO  SimpleTestExecutorMonitor] Test 337/338 returned Errors: 0 / Prevalence:
                     246. Test: https://ns.eccenca.com/geo/countryCallingCode is missing proper range
17-Nov-2017 10:33:33 [INFO  SimpleTestExecutorMonitor] Test 338/338 returned Errors: 0 / Prevalence:
                     250. Test: https://ns.eccenca.com/geo/tld is missing proper range
17-Nov-2017 10:33:33 [INFO  SimpleTestExecutorMonitor] Tests run: 338, Failed: 0, Timeout: 0, Error:
                     0. Individual Errors: 0
17-Nov-2017 10:33:33 [WARN  ValidateCLI] Results folder (./results/) does not exist, creating it...
17-Nov-2017 10:33:33 [INFO  ValidateCLI] Results stored in:
                     ./results/._.instances.ttl.aggregatedTestCaseResult.*
17-Nov-2017 10:33:34 [INFO  RDFUnitUtils] Loaded 14 schema declarations from:
                     sun.net.www.protocol.jar.JarURLConnection$JarURLInputStream@5a10411
17-Nov-2017 10:33:43 [INFO  SimpleTestExecutorMonitor] Testing ./.manual.ttl
17-Nov-2017 10:33:57 [INFO  SimpleTestExecutorMonitor] Test 1/13 returned Errors: 0 / Prevalence: 258.
                     Test: Two defined resources should not have the same rdfs:comment
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 2/13 returned Errors: 0 / Prevalence:
                     61686. Test: All resources defined by an ontology should start with the ontology
                     URI and have rdfs:isDefinedBy and rdf:type
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 3/13 returned Errors: 0 / Prevalence: 0.
                     Test: All skos:Concepts should have a skos:broader statement OR be
                     skos:topConceptOf [a skos:ConceptScheme]
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 4/13 returned Errors: 0 / Prevalence: 0.
                     Test: All skos:Concepts should be skos:inScheme [a skos:ConceptScheme], have a
                     skos:definition or rdfs:comment and have a skos:prefLabel
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 5/13 returned Errors: 0 / Prevalence: 258.
                     Test: All defined resources should have rdfs:label different from rdfs:comment
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 6/13 returned Errors: 0 / Prevalence: 1.
                     Test: All ontologies should have owl:versionInfo
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 7/13 returned Errors: 0 / Prevalence: 258.
                     Test: All defined resources must not have literals with todo, foo, bar, lorem or
                     ipsum in it
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 8/13 returned Errors: 0 / Prevalence: 258.
                     Test: All defined resources should have different rdfs:comment in different
                     languages
17-Nov-2017 10:34:44 [INFO  SimpleTestExecutorMonitor] Test 9/13 returned Errors: 0 / Prevalence: 772.
                     Test: All defined resources should have at least one rdfs:label and an
```

https://bamboo.brox.de/browse/SCHEMA-GEODATA-15


Agent
email : string
name : string
phone number : string
member of
Department
name : string
has address
Address
city : string
country : string
house number : string
postal code : string
street : string
responsible for
Product
ID : string
name : string
price in EUR : decimal
Employee
has direct report
Manager
has manager
manages product
has product manager
expert for
Product Category
name : string
has category
Hardware
depth in mm : decimal
heigth in mm : decimal
weigth in g : decimal
width in mm : decimal
eligible for
Service

Ontology Specification Draft

2. Products vocabulary: Overview

back to ToC

This ontology has the following classes and properties.

Classes

Agent Department Employee Hardware Manager Price Product Product Category Service

Object Properties

eligible for expert for has category has direct report has manager has product manager manages product member of price responsible for

Data Properties

address text amount area of expertise currency depth in mm email height in mm ID name phone number weight in g width in mm

Annotation Properties

created creator issued license modified name preferred namespace prefix preferred namespace uri publisher rights vocabulary

Named Individuals

eccenca gmb h rpietzsch

3. Products vocabulary: Description

back to ToC

This is a placeholder text for the description of your ontology. The description should include an explanation and a diagram explaining how the classes are related, examples of usage, etc.

4. Cross reference for Products vocabulary classes, properties and dataproperties

back to ToC

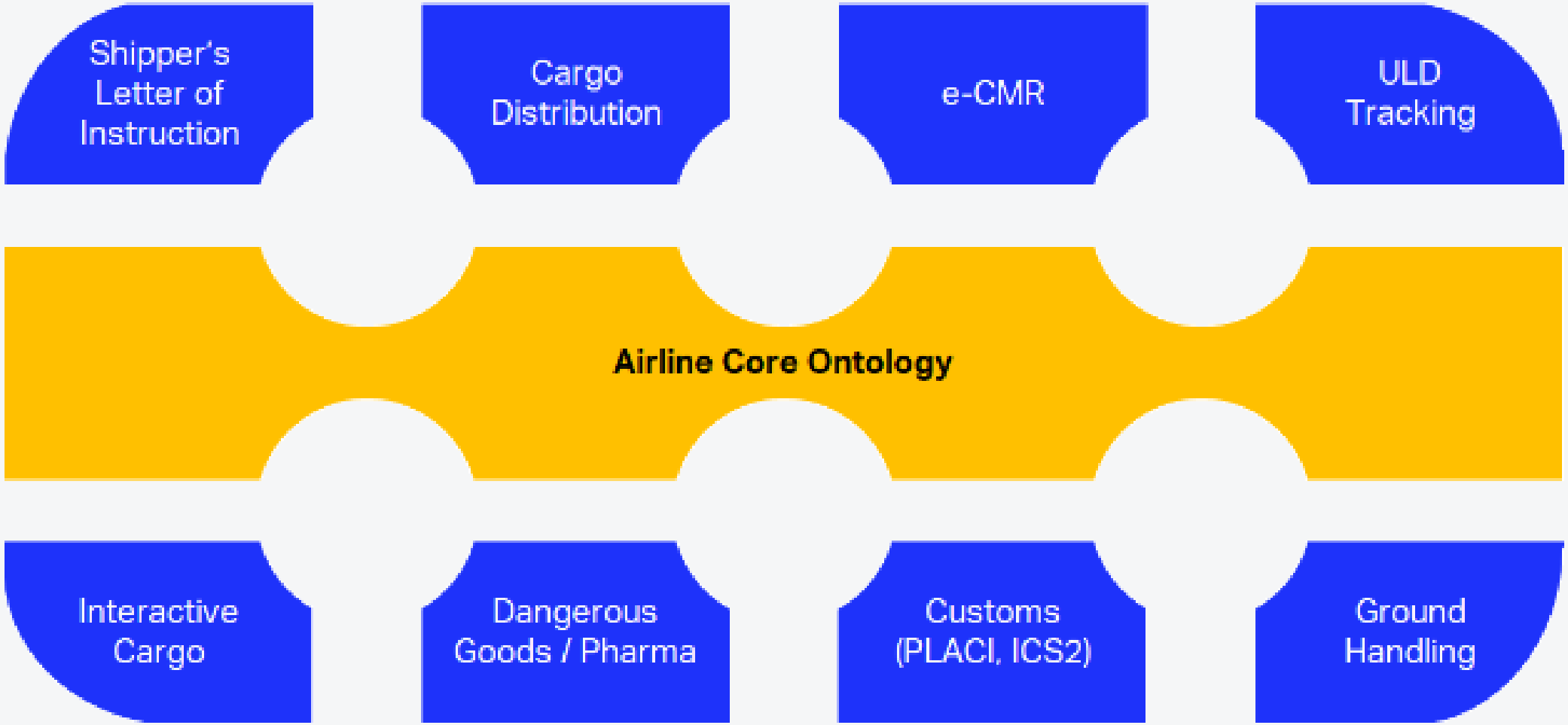
ONE Record Insights

Part 3

Operationalizing ontologies

eccenca
mastering complexity

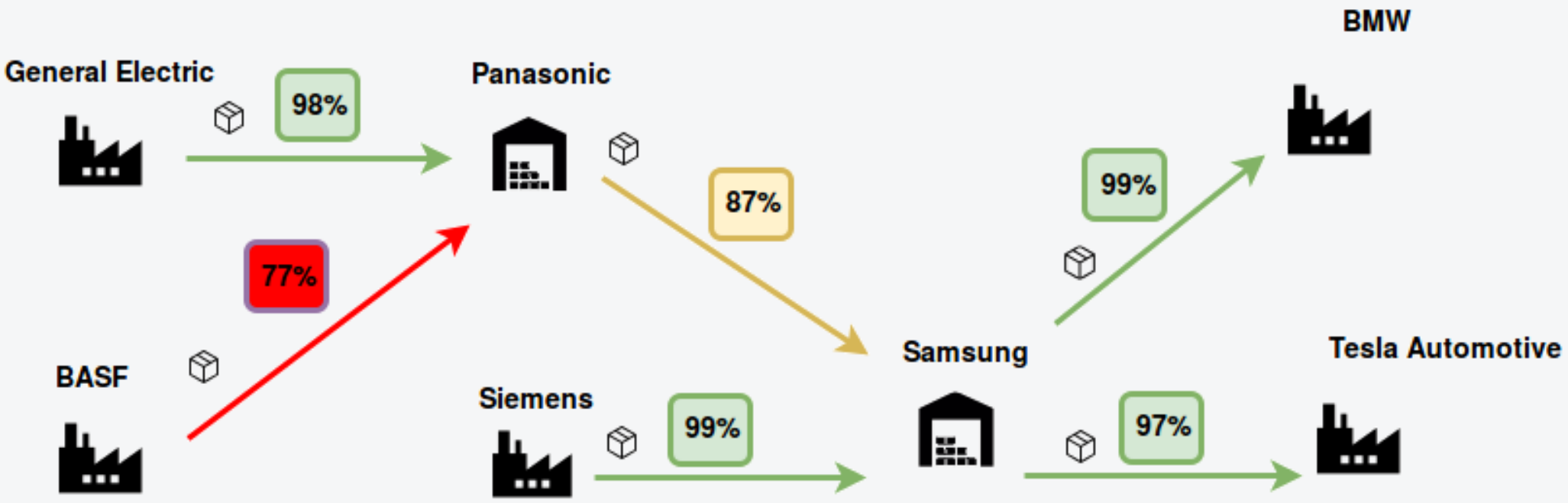
IATA: One Record Data Model



APICS + eccenca: SCOR-VOC

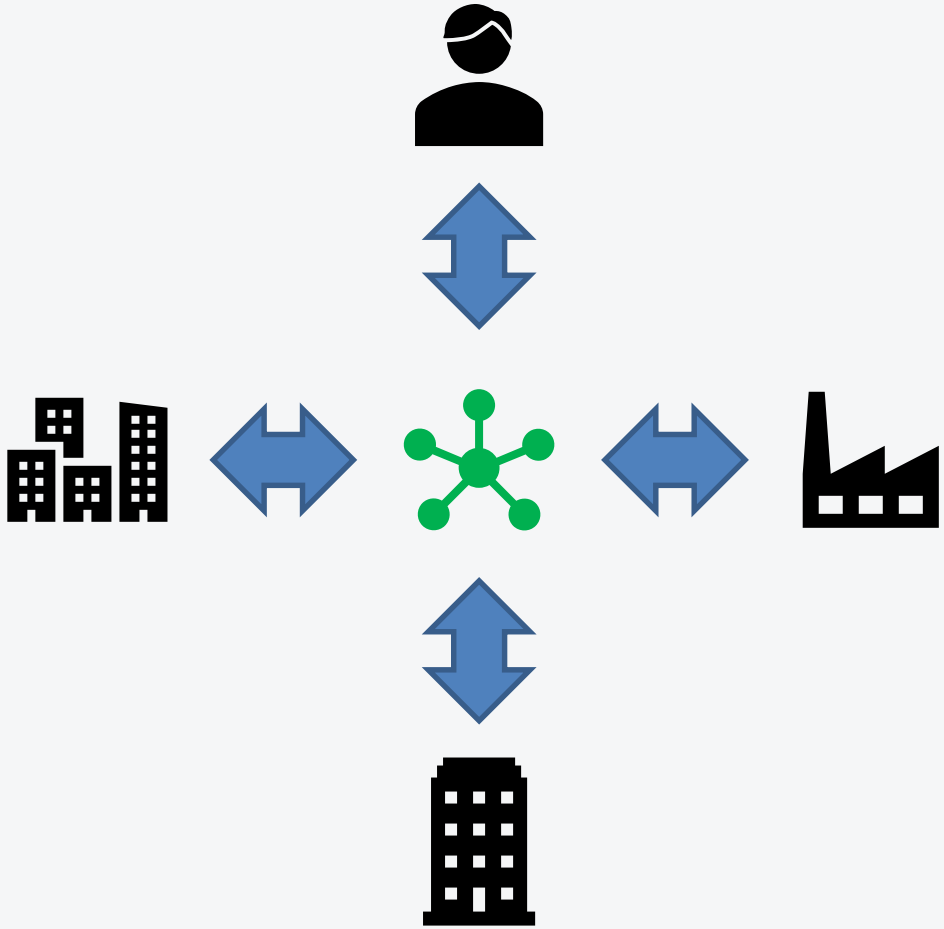
The value in the rectangle represents a SCOR Metric result.

For example: Orders delivered in time for certain goods.

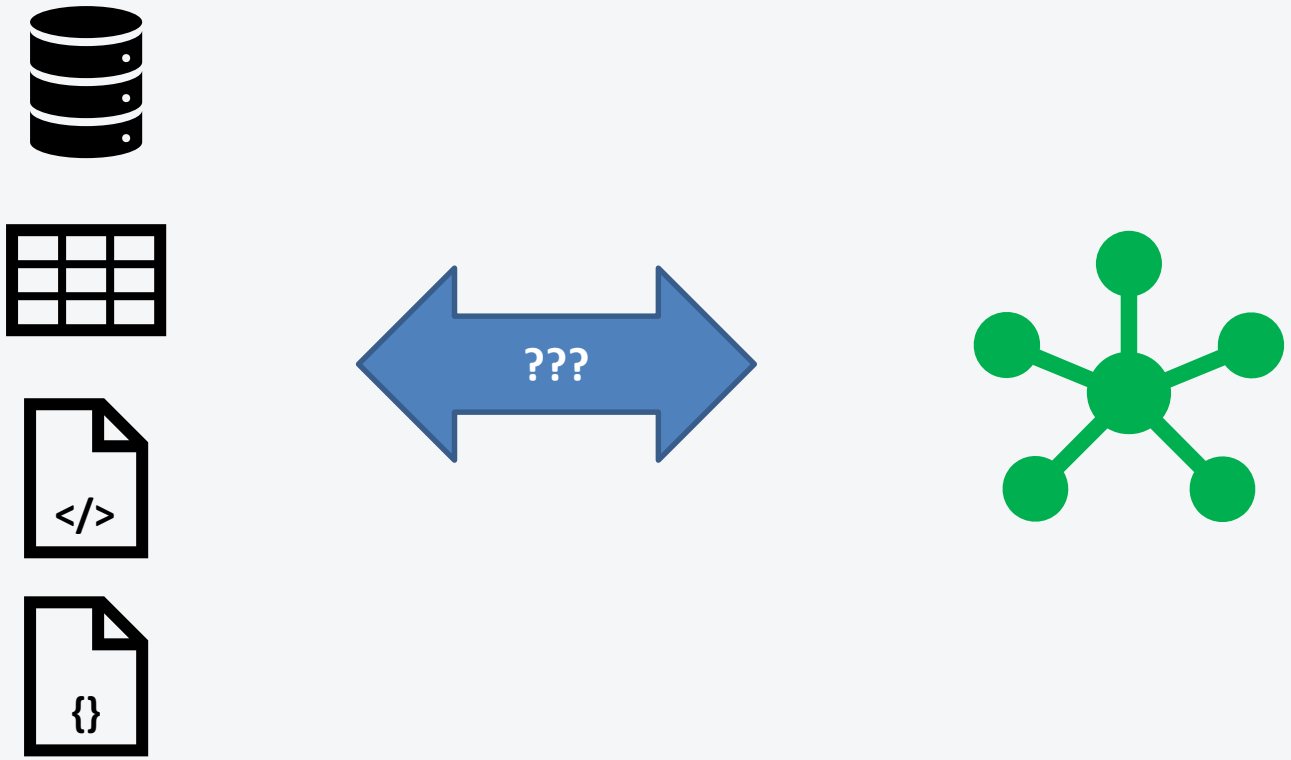


The universal Semantic Data Collaboration Paradigm

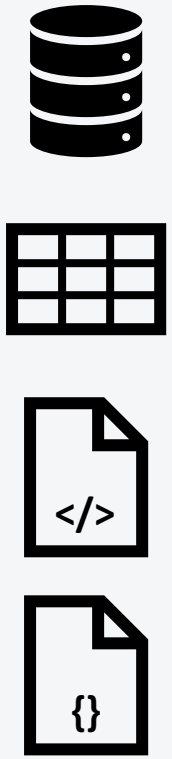
eccenca
mastering complexity

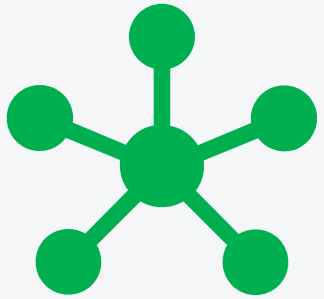


But … How …?



But … ~~How~~ … Now!

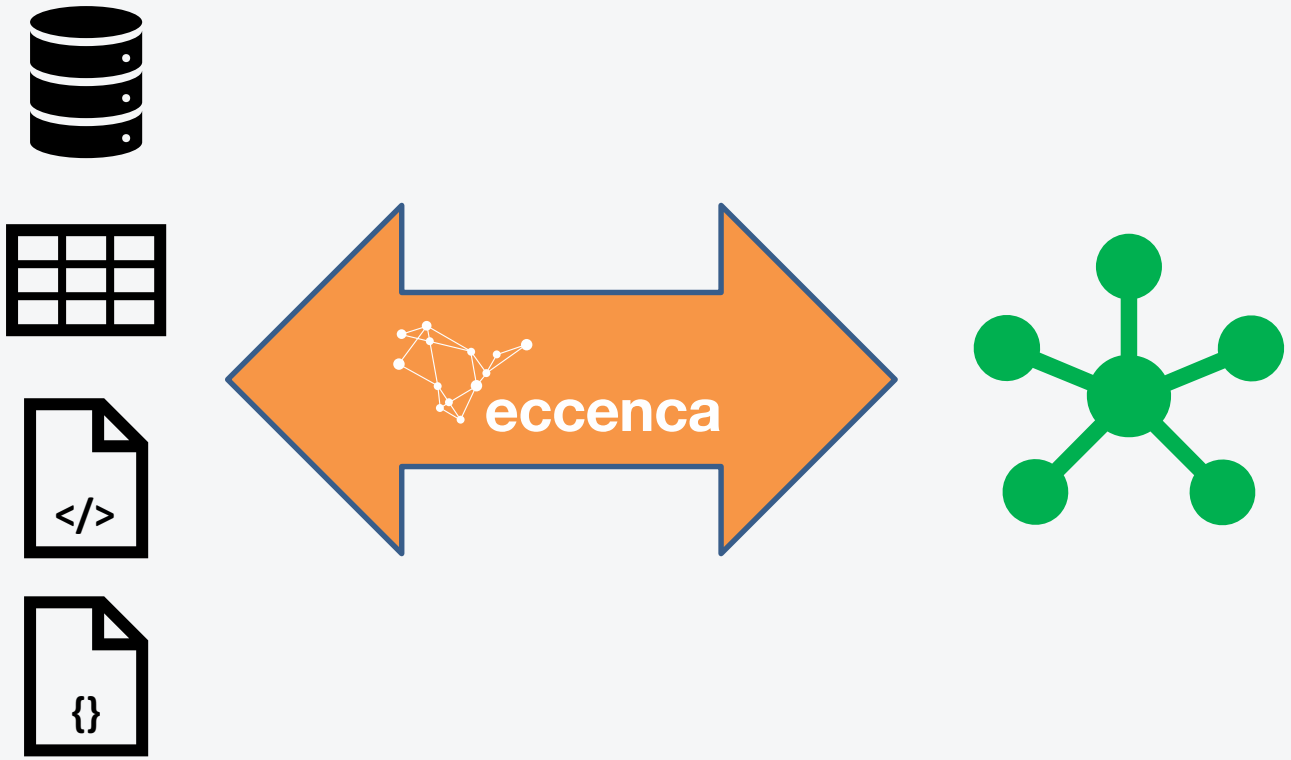




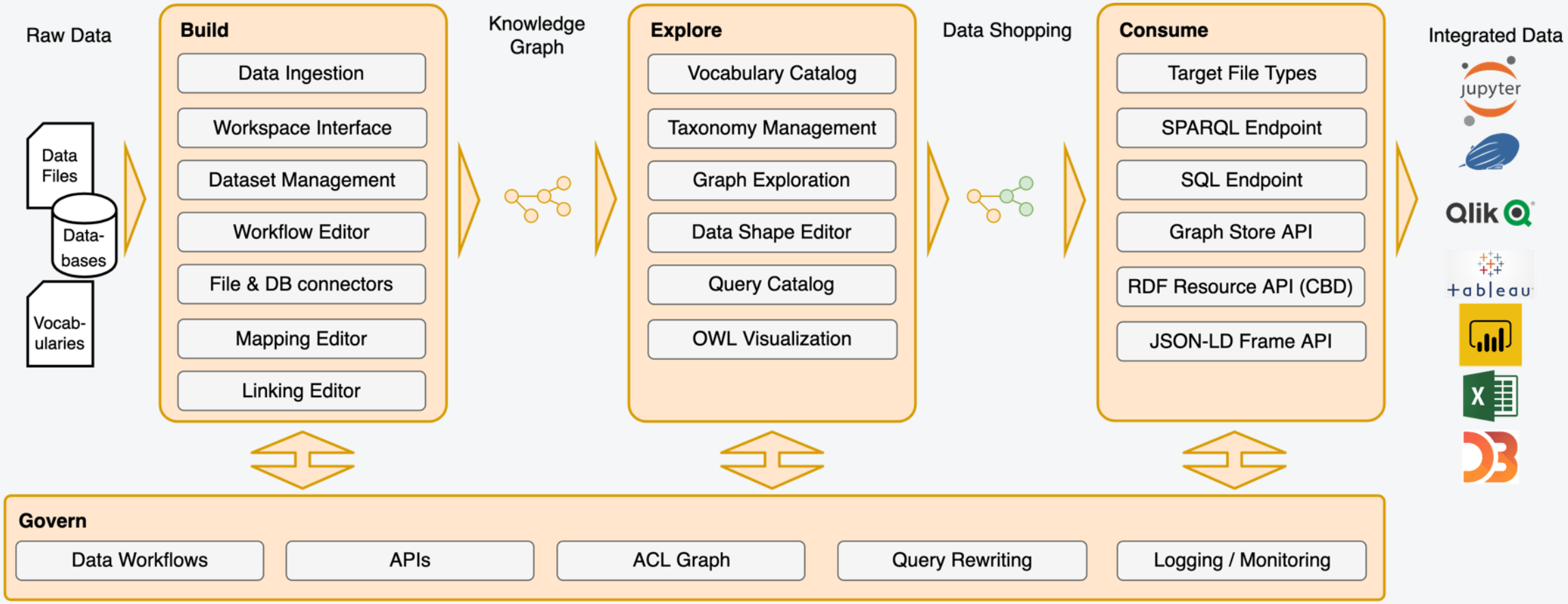
- Low level RDF libraries
- Mapping frameworks and mapping languages
- Triple Stores
- Enterprise knowledge graph platform

eccenca Corporate Memory an Enterprise Knowledge Graph Platform

eccenca
mastering complexity



Corporate Memory: User Journey and Functional Areas

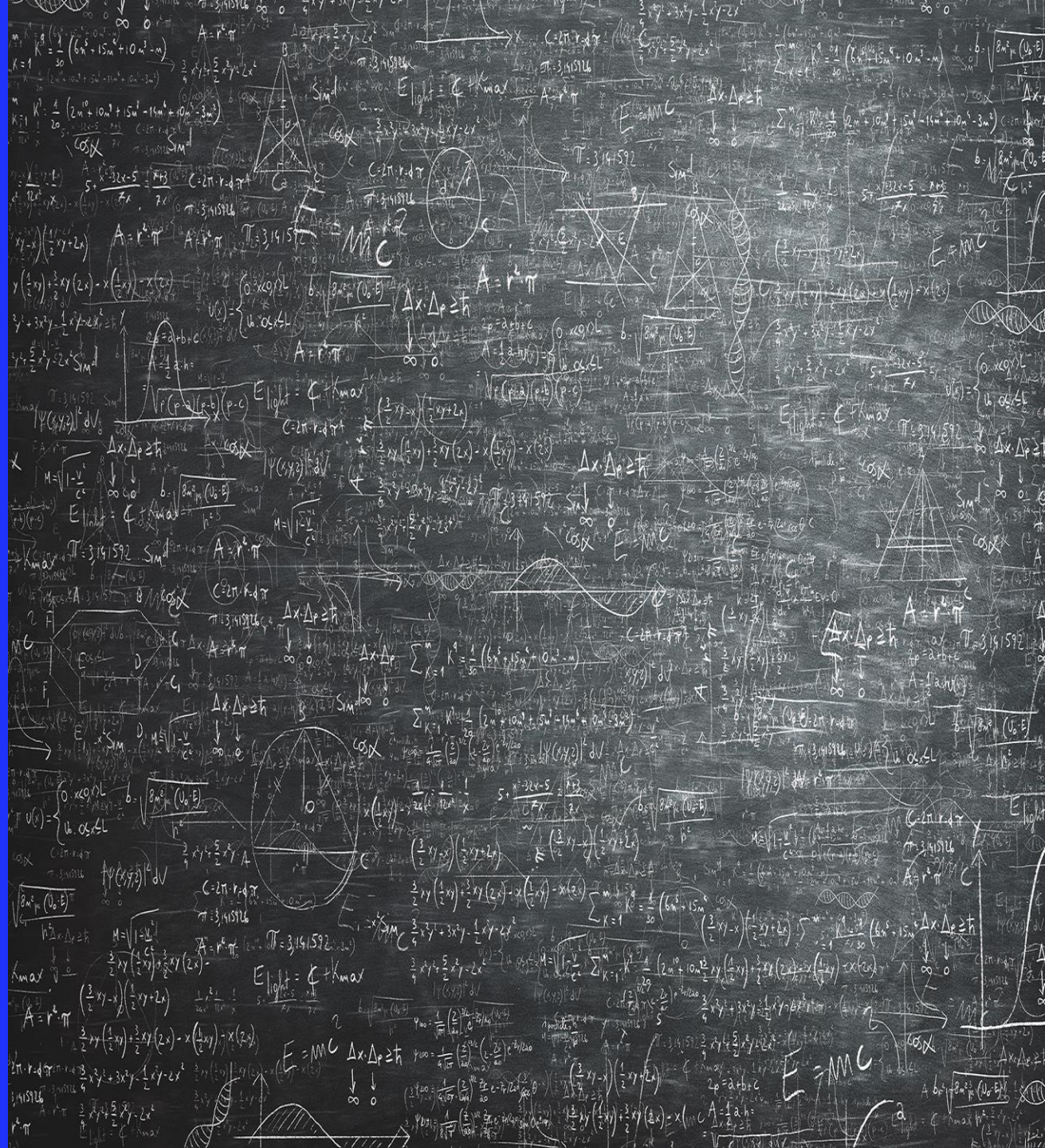


Findable Accessible Interoperable Reusable

ONE Record Insights

Part 4

The ONE Record ontology



ONE Record Data Model Components

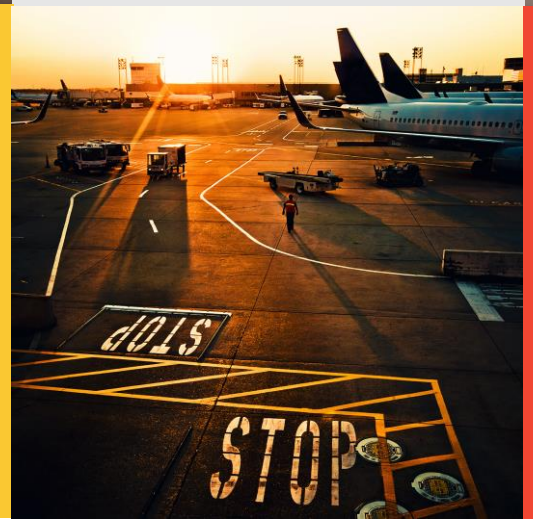
Determine the domain and scope of the ontology

What is the domain that the ontology will cover?

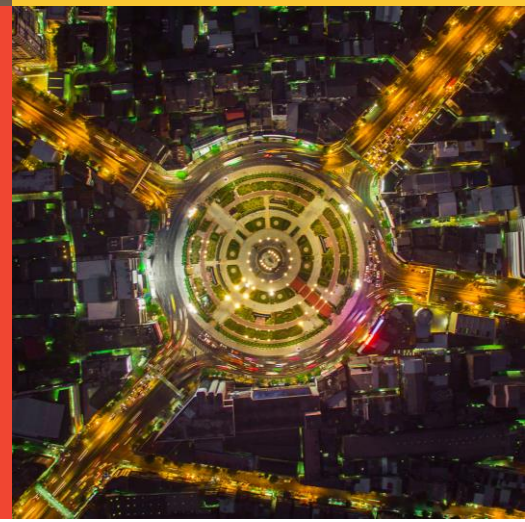
What information the ontology should cover?

How can we share this ontology?

For what we are going to use the ontology?



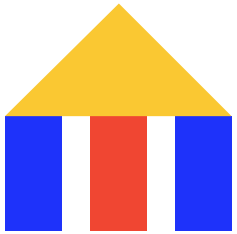
Who will create, use and maintain the ontology?



IATA

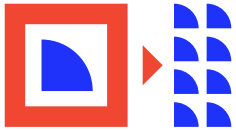
Data Model: Standard components

To support the deployment and the adoption of the ONE Record Data Model, IATA published a set of specification, guidance materials and tools

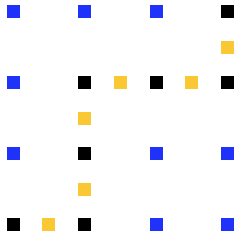


Design Principles

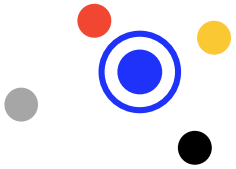
Conceptual Data Model



Logical Data Model



Use Cases

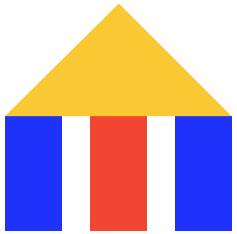


Ontology

https://github.com/IATA-Cargo/ONE-Record/tree/master/March-2020-standard-COTB-endorsed/Data-Model

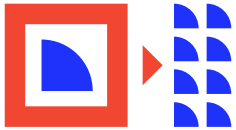
Data Model: Standard components

To support the deployment and the adoption of the ONE Record Data Model, IATA published a set of specification, guidance materials and tools

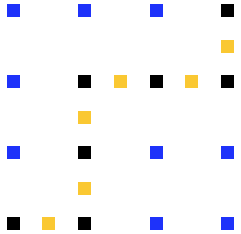


Design Principles

Conceptual Data Model



Logical Data Model



Use Cases

Ontology

https://github.com/IATA-Cargo/ONE-Record/tree/master/March-2020-standard-COTB-endorsed/Data-Model

Search or jump to... / Pull requests Issues Marketplace Explore

IATA-Cargo / ONE-Record

Unwatch 28 | Unstar 36 | Fork 24

Code | Issues 2 | Pull requests | Actions | Projects | Wiki | Security | Insights

Branch: master | ONE-Record / working_draft / ontology / | Go to file | Add file

andrablaj committed e6a148a 2 days ago ... History

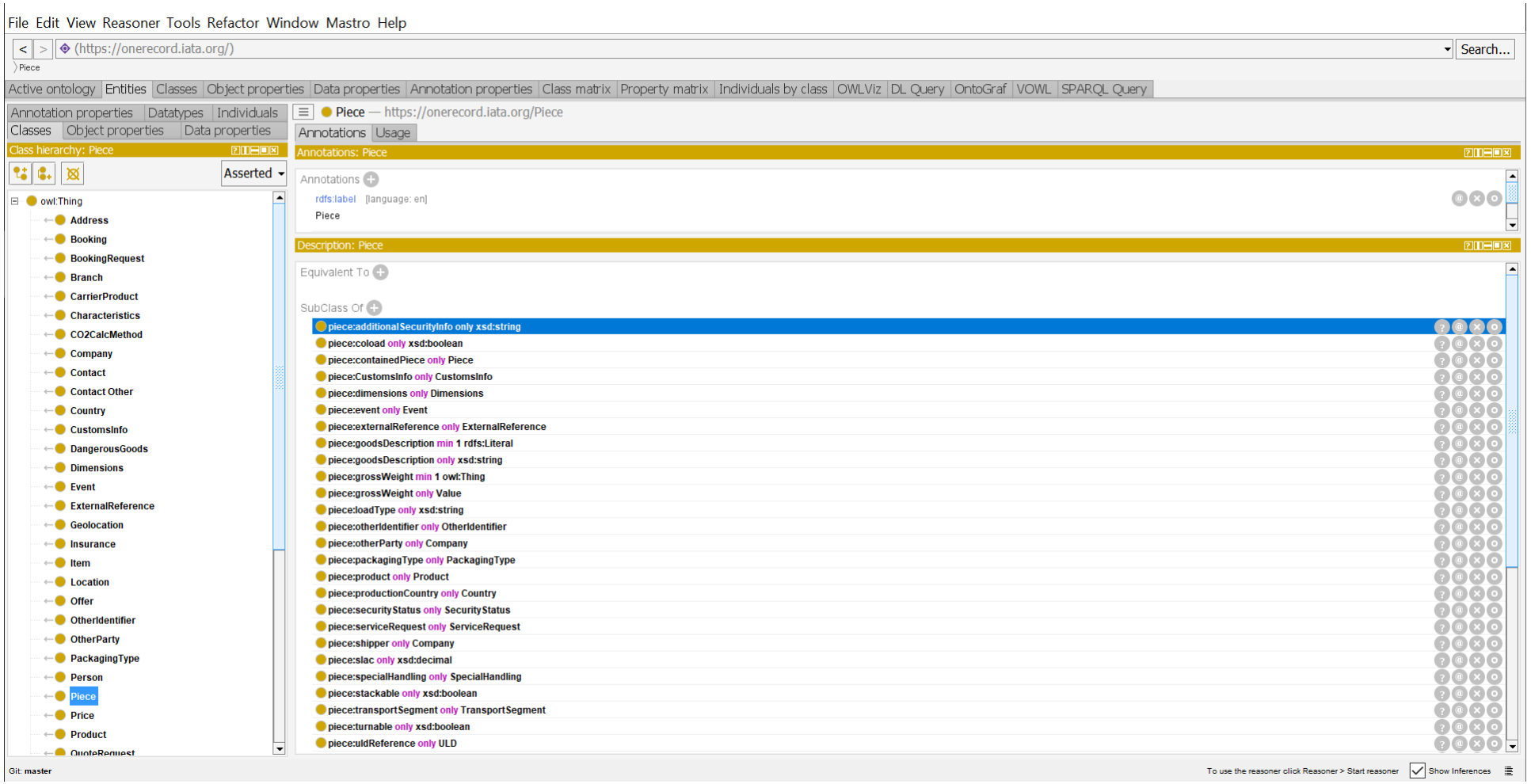
..

1R Data Model ontology - May 2020.ttl	Add more description to the ontology	2 days ago
1R Data Model ontology - old.ttl	Fix ontology	2 days ago
api_models.ttl	Add memento models	last month

http://github.com/IATA-Cargo/ONE-Record

Ontology Creation: Tools & Methodology

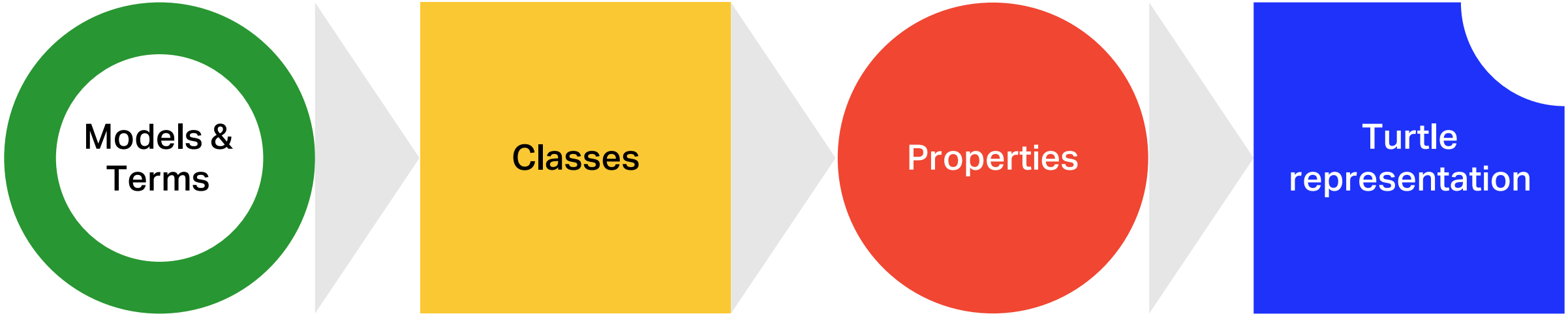
Ontology Creation with Protégé



Developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine, Protégé tool is one of the oldest and most widely deployed ontology modelling tools. It was originally conceived as a frame-based modelling tool for rich ontologies following the Open Knowledge Base Connectivity protocol. Later iterations of Protégé have expanded to include a plug-in that is now widely used for OWL and RDF modelling.

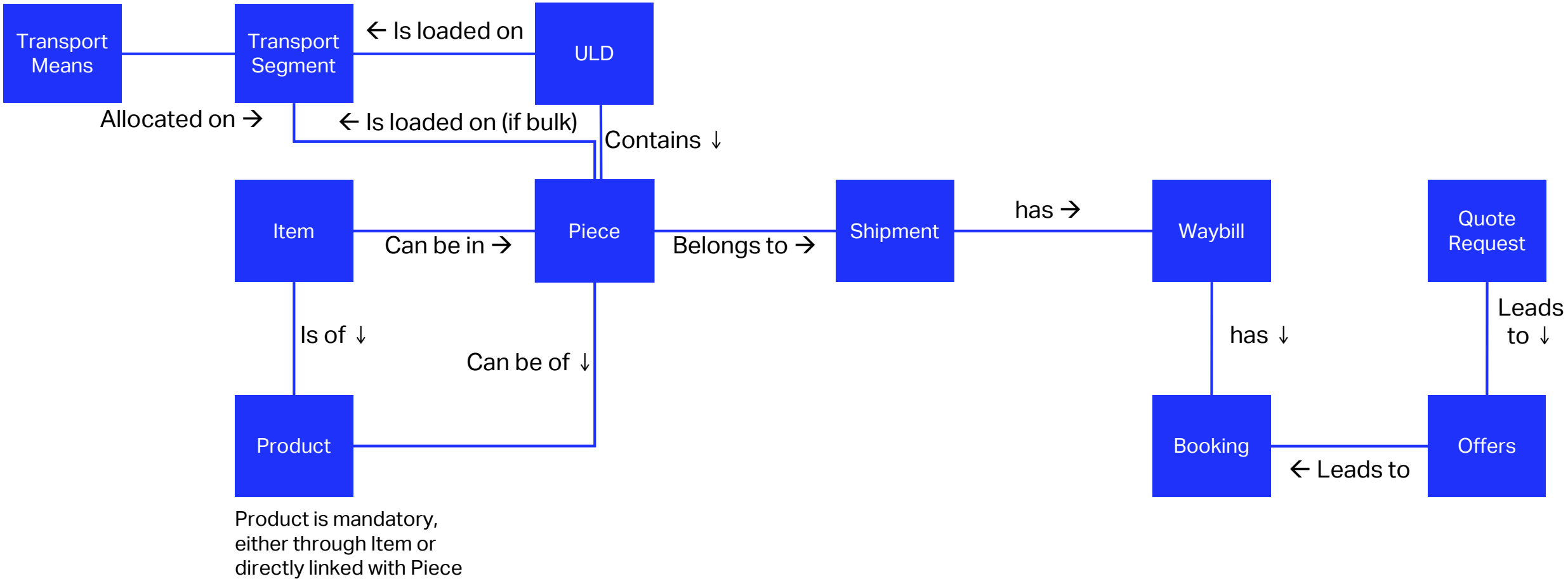
https://protege.stanford.edu/

Creating an ontology



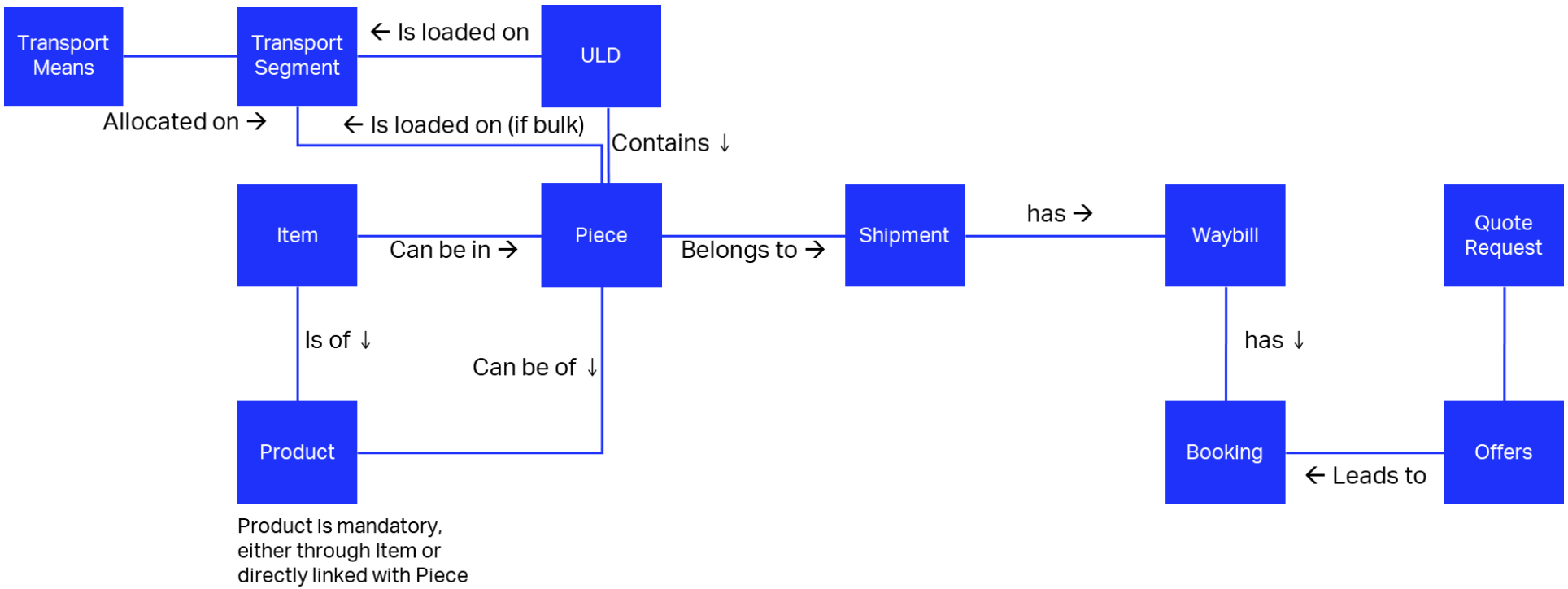
Iterative process

Models & Terms



Classes

- owl:Thing
  - Address
  - Booking
  - BookingRequest
  - Branch
  - CarrierProduct
  - Characteristics
  - CO2CalcMethod
  - Company
  - Contact
  - ContactOther
  - Country
  - CustomsInfo
  - DangerousGoods
  - Dimensions
  - Event
  - ExternalReference
  - Geolocation
  - Insurance
  - Item
  - Location
  - Offer
  - OtherIdentifier
  - OtherParty
  - PackagingType
  - Person
  - Piece
  - Price
  - Product
  - QuoteRequest
  - QuoteResponse
  - Ranges
  - Ratings
  - ReasonsForSecurityStatus
  - ReceivedFrom
  - Routing
  - SecurityStatus
  - ServiceRequest
  - Shipment
  - SpecialHandling
  - TransportMeans
  - TransportSegment
  - ULD
  - Value
  - VolumetricWeight
  - Waybill



Properties

Datatype properties

Object properties

Cardinality

Domain

Range

Turtle representation

```
xsd:date rdf:type rdfs:Datatype .

#################################################################
#    Classes
#################################################################

###  https://onerecord.iata.org/Address
:Address rdf:type owl:Class ;
         rdfs:subClassOf [ rdf:type owl:Restriction ;
                           owl:onProperty address:addressCode ;
                           owl:allValuesFrom xsd:string ;
                         ] ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty address:addressCodeType ;
                           owl:allValuesFrom xsd:string ;
                         ] ,
                         [ rdf:type owl:Restriction ;
                           owl:onProperty address:cityCode ;
                           owl:allValuesFrom xsd:string ;
```

.ttl

Ontology Tools

Ontology Visualization with Widoco

Ontology Specification Draft

language **en**

IATA ONE Record Ontology

Release 30-06-2020

Latest version:
https://onerecord.iata.org/
Revision:
1.0
Download serialization:
Format JSON LD | Format RDF/XML | Format N Triples | Format TTL
License:
License license name goes here
Visualization:
Visualize with WebVowl
Evaluation:
Evaluate with OOPS! (OntOlogy Pitfall Scanner!)
Cite as:
IATA ONE Record Ontology. Revision: 1.0.
Provenance of this page

Abstract

The ontology of IATA ONE Record standard is the core data model underlying the Linked Data solution drafted by the ONE Record standard: https://github.com/IATA-Cargo/ONE-Record. ONE Record is a standard for data sharing and creates a single record view of the shipment. This standard defines a common data model for the data that is shared via standardized and secured web API.

1. Introduction

back to ToC

ONE Record Data Model provides the air cargo industry with a standard data structure for data exchange using JSON-LD that facilitates data integration with existing and new data services.

1.1. Namespace declarations

Table 1: Namespaces used in the document

[Ontology NS Prefix]	<https://onerecord.iata.org/>
insurance	<https://onerecord.iata.org/Insurance>
country	<https://onerecord.iata.org/Country>
booking	<https://onerecord.iata.org/Booking>

Ontology Evaluation with oops!

IATA ONE Record Ontology

Title IATA ONE Record Ontology
URI https://onerecord.iata.org/
Version 1.0

The following evaluation results have been generated by the RESTFul web service provided by OOPS! (OntOlogy Pitfall Scanner!).

It is obvious that not all the pitfalls are equally important; their impact in the ontology will depend on multiple factors. For this reason, each pitfall has an importance level attached indicating how important it is. We have identified three levels:

- Critical — It is crucial to correct the pitfall. Otherwise, it could affect the ontology consistency, reasoning, applicability, etc.
- Important — Though not critical for ontology function, it is important to correct this type of pitfall.
- Minor — It is not really a problem, but by correcting it we will make the ontology nicer.

Evaluation results

P10. Missing disjointness	ontology *	Important
P12. Equivalent properties not explicitly declared	29 cases detected.	Important
P13. Inverse relationships not explicitly declared	127 cases detected.	Minor
P21. Using a miscellaneous class	3 cases detected.	Minor
P24. Using recursive definitions	2 cases detected.	Important

References:

- [1] Gómez-Pórez, A. Ontology Evaluation. Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.
- [2] Noy, N.F., McGuinness. D. L. Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Standford Medical Informatics. 2001.
- [3] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R.,; Wang, H., Wroe, C. "Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns". In Proc. of EKAW 2004, pp: 63-81. Springer. 2004.
- [4] Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A. Weaving the Pedantic Web. Linked Data on the Web Workshop LDOW2010 at WWW2010 (2010).
- [5] Archer, P., Goedertier, S., and Loutas, N. D7.1.3 - Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC. Deliverable. December 17, 2012.
- [6] Heath, T., Bizer, C.: Linked data: Evolving the Web into a global data space (1st edition). Morgan & Claypool (2011).

Developed by María Poveda
Built with Bootstrap
Integration with Widoco by Daniel Garijo

Developed with:

Object Triple Mapping with JOPA

Java OWL Persistence API (JOPA)

JOPA is a persistence API and implementation for accessing OWL ontologies. Its main features are:

- Object-ontological mapping based on integrity constraints,
- Explicit access to inferred knowledge,
- Access to unmapped properties and individual's types
- Transactions
- Separate storage access layer

Downloads and Links

The project is hosted at github.com: https://github.com/kbss-cvut/jopa

JOPA artifacts can be downloaded from Maven central:

JOPA implementation:

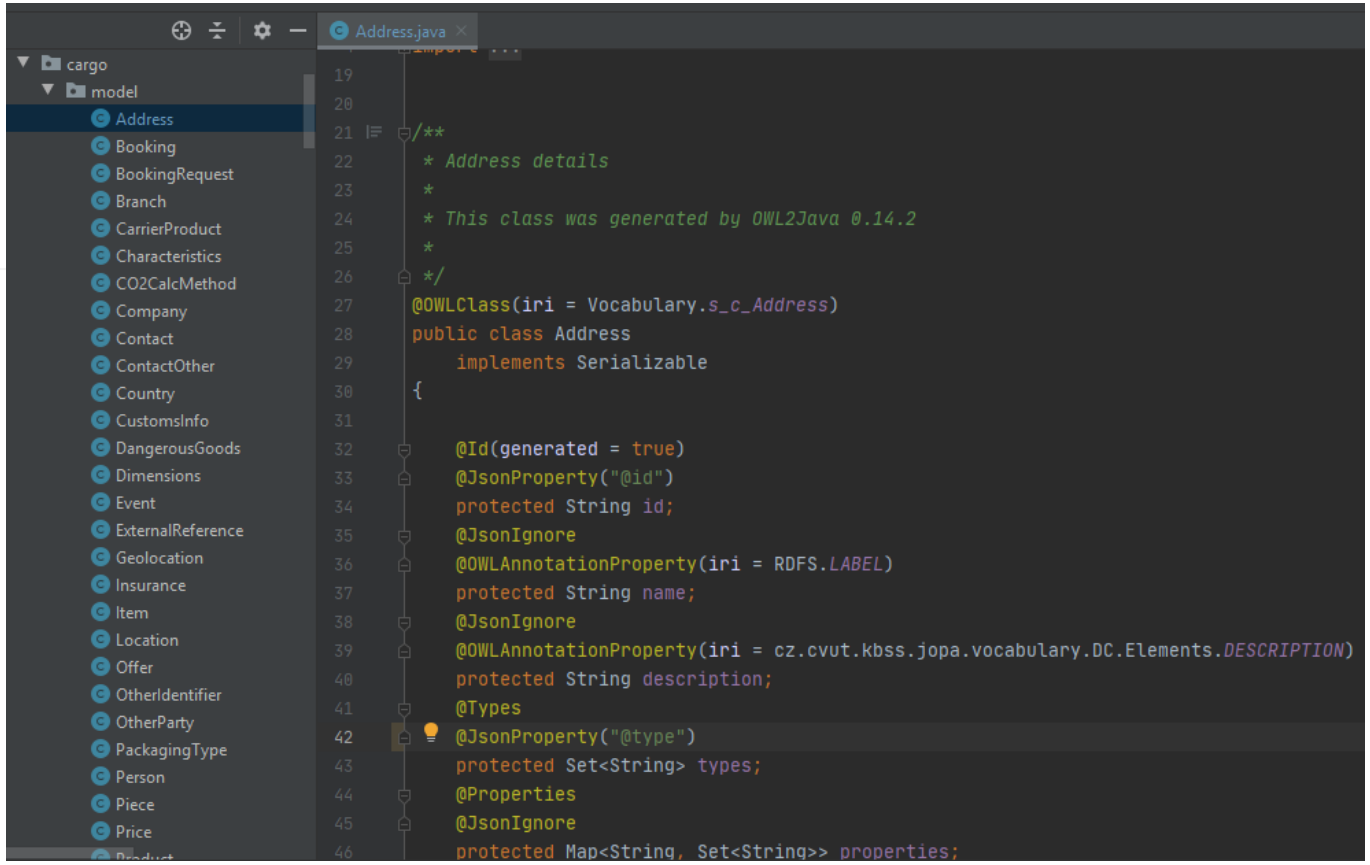
```
<dependency>
    <groupId>cz.cvut.kbss.jopa</groupId>
    <artifactId>jopa-impl</artifactId>
</dependecy>
```

Sesame (RDF4J) OntoDriver:

```
<dependency>
    <groupId>cz.cvut.kbss.jopa</groupId>
    <artifactId>ontodriver-sesame</artifactId>
</dependecy>
```

OWLAPI OntoDriver:

```
<dependency>
    <groupId>cz.cvut.kbss.jopa</groupId>
    <artifactId>ontodriver-owlapi</artifactId>
```



https://github.com/IATA-Cargo/one-record-server-java

There is no single correct ontology for a domain and there is no single correct way to create it.

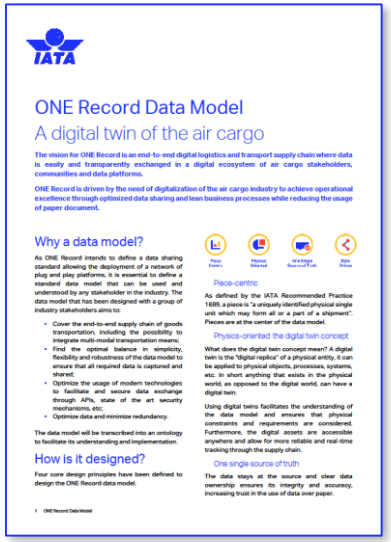
ONE Record Insights

Bonus

Deep dive into the ONE Record standard

ONE Record Insights and White Papers

Don’t miss our ONE Record Insights and White Papers



ONE Record Data Model

A digital twin of the air cargo

The vision for ONE Record is an end-to-end digital logistics and transport supply chain where data is easily and transparently exchanged in a digital ecosystem of air cargo stakeholders, communities and data platforms.

ONE Record is driven by the need of digitalization of the air cargo industry to achieve operational excellence through optimized data sharing and lean business processes while reducing the usage of paper document.

Why a data model?

As ONE Record intends to define a data sharing standard allowing the deployment of a network of plug and play platforms, it is essential to define a standard data model that can be used and understood by any stakeholder in the industry. The data model that has been designed with a group of industry stakeholders aims to:

- Cover the end-to-end supply chain of goods transportation, including the possibility to integrate multi-modal transportation means;
- Find the optimal balance in simplicity, flexibility and robustness of the data model to ensure that all required data is captured and shared;
- Optimize the usage of modern technologies to facilitate and secure data exchange through APIs, state of the art security mechanisms, etc;
- Optimize data and minimize redundancy.

The data model will be transcribed into an ontology to facilitate its understanding and implementation.

How is it designed?

Four core design principles have been defined to design the ONE Record data model.

Piece-centric

As defined by the IATA Recommended Practice 1689, a piece is “a uniquely identified physical single unit which may form all or a part of a shipment”. Pieces are at the center of the data model.

Physics-oriented the digital twin concept

What does the digital twin concept mean? A digital twin is the “digital replica” of a physical entity, it can be applied to physical objects, processes, systems, etc. In short anything that exists in the physical world, as opposed to the digital world, can have a digital twin.

Using digital twins facilitates the understanding of the data model and ensures that physical constraints and requirements are considered. Furthermore, the digital assets are accessible anywhere and allow for more reliable and real-time tracking through the supply chain.

One single source of truth

The data stays at the source and clear data ownership ensures its integrity and accuracy, increasing trust in the use of data over paper.

1 ONE Record Data Model

ONE Record Data Model

ONE Record & the power of ontologies

Looking beyond the hype for real solutions to real problems

ONE Record is a standard for data sharing that defines a common model for data that is shared via a standardized and secured API. The Data model specification provides the air cargo industry with a common language and a standard data structure for data exchange based on three concepts: Semantic Web, ontologies and Linked Data.

This document focuses on a high-level overview of what is Semantic Web and why the usage of concepts such as Linked Data and ontologies can be beneficial in a distributed end-to-end digital logistics and transport chain, such as the one that the ONE Record standard aims to support.

What is Semantic Web?

Semantic Web is an extension of the Web that adds semantics to the current format of data representation enabling considerable gains in the information treatment.

On the Web, you can post documents and photos about yourself, listen to podcasts, go shopping, and have video chats with friends. With Semantic Web, you can create data about a topic; remix it with data created by others and link together the most relevant of your information, establishing a world-wide standard of communication.

Why is an ontology important?

An ontology defines a common vocabulary for different stakeholders who need to share information in a domain. It includes machine-interpretable definitions of basic concepts within a domain and relations among them.

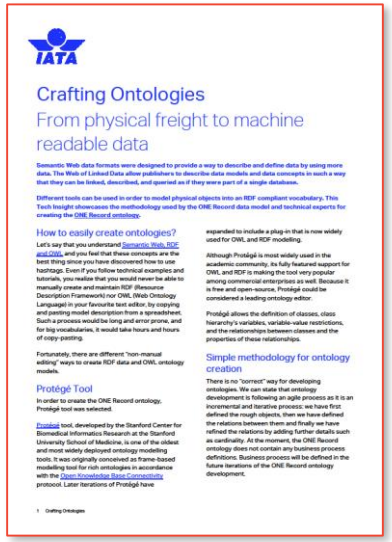
Some of the reasons for which someone would want to develop an ontology are:

- To share common understanding of the structure of data among stakeholders;
- To use accurate description of knowledge, unlike natural language in which words can have totally different semantic meaning depending on their context;
- To enable the re-use of domain knowledge. An example would be an ontology created to describe the domain of a bookstore, which once shared would allow other bookstores to build new catalogs without having to redo an analysis of their domain;
- To avoid misinterpretations and to contribute to greater reliability of knowledge representation;
- To enable automated reasoning about the data, as the relationships between the concepts are well defined.

Let’s take a simple example: An airline may want to integrate passenger data coming from other airlines. The data can be imported into a common RDF model by using, for example, converters to the airlines databases. However, one database may use the term “traveller”, whereas the other may use the term “passenger”. To make the integration complete, an extra definition should be added to the RDF data, describing the fact that the relationship described as “traveller” is the same as

1 ONE Record & the power of ontologies

ONE Record & The power of ontologies



Crafting Ontologies

From physical freight to machine readable data

Semantic Web data formats were designed to provide a way to describe and define data by using more data. The Web of Linked Data allow publishers to describe data models and data concepts in such a way that they can be linked, described, and queried as if they were part of a single database.

Different tools can be used in order to model physical objects into an RDF compliant vocabulary. This Tech Insight showcases the methodology used by the ONE Record data model and technical experts for creating the ONE Record ontology.

How to easily create ontologies?

Let’s say that you understand Semantic Web, RDF and OWL, and you feel that these concepts are the best thing since you have discovered how to use hashtags. Even if you follow technical examples and tutorials, you realise that you would never be able to manually create and maintain RDF (Resource Description Framework) nor OWL (Web Ontology Language) in your favourite text editor, by copying and pasting model description from a spreadsheet. Such a process would be long and error prone, and for big vocabularies, it would take hours and hours of copy-pasting.

Fortunately, there are different “non-manual editing” ways to create RDF data and OWL ontology models.

Protégé Tool

In order to create the ONE Record ontology, Protégé tool was selected.

Protégé tool, developed by the Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine, is one of the oldest and most widely deployed ontology modelling tools. It was originally conceived as frame-based modelling tool for rich ontologies in accordance with the Open Knowledge Base Connectivity protocol. Later iterations of Protégé have expanded to include a plug-in that is now widely used for OWL and RDF modelling.

Although Protégé is most widely used in the academic community, its fully featured support for OWL and RDF is making the tool very popular among commercial enterprises as well. Because it is free and open-source, Protégé could be considered a leading ontology editor.

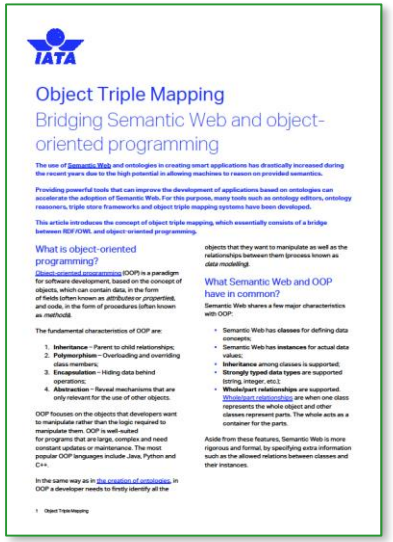
Protégé allows the definition of classes, class hierarchy’s variables, variable-value restrictions, and the relationships between classes and the properties of these relationships.

Simple methodology for ontology creation

There is no “correct” way for developing ontologies. We can state that ontology development is following an agile process as it is an incremental and iterative process: we have first defined the rough objects, then we have defined the relations between them and finally we have refined the relations by adding further details such as cardinality. At the moment, the ONE Record ontology does not contain any business process definitions. Business process will be defined in the future iterations of the ONE Record ontology development.

1 Crafting Ontologies

Crafting Ontologies



Object Triple Mapping

Bridging Semantic Web and object-oriented programming

The use of Semantic Web and ontologies in creating smart applications has drastically increased during the recent years due to the high potential in allowing machines to reason on provided semantics.

Providing powerful tools that can improve the development of applications based on ontologies can accelerate the adoption of Semantic Web. For this purpose, many tools such as ontology editors, ontology reasoners, triple store frameworks and object triple mapping systems have been developed.

This article introduces the concept of object triple mapping, which essentially consists of a bridge between RDF/OWL and object-oriented programming.

What is object-oriented programming?

Object-oriented programming (OOP) is a paradigm for software development, based on the concept of objects, which can contain data, in the form of fields (often known as *attributes* or *properties*), and code, in the form of procedures (often known as *methods*).

The fundamental characteristics of OOP are:

1. **Inheritance** – Parent to child relationships;
2. **Polymorphism** – Overloading and overriding class members;
3. **Encapsulation** – Hiding data behind operations;
4. **Abstraction** – Reveal mechanisms that are only relevant for the use of other objects.

OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. OOP is well-suited for programs that are large, complex and need constant updates or maintenance. The most popular OOP languages include Java, Python and C++.

In the same way as in the creation of ontologies, in OOP a developer needs to firstly identify all the objects that they want to manipulate as well as the relationships between them (process known as *data modelling*).

What Semantic Web and OOP have in common?

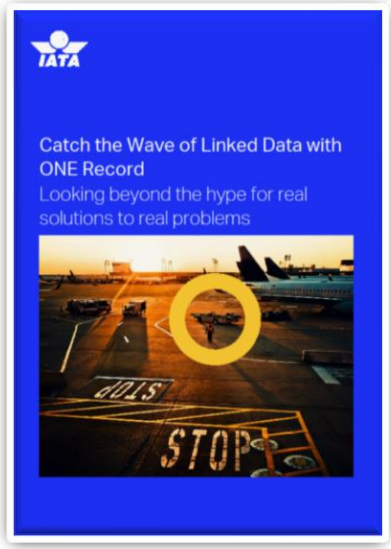
Semantic Web shares a few major characteristics with OOP:

- Semantic Web has **classes** for defining data concepts;
- Semantic Web has **instances** for actual data values;
- **Inheritance** among classes is supported;
- **Strongly typed data types** are supported (string, integer, etc.);
- **Whole/part relationships** are supported. Whole/part relationships are when one class represents the whole object and other classes represent parts. The whole acts as a container for the parts.

Aside from these features, Semantic Web is more rigorous and formal, by specifying extra information such as the allowed relations between classes and their instances.

1 Object Triple Mapping

Object Triple Mapping



Catch the Wave of the Linked Data

https://www.iata.org/one-record/#tab-2

ONE Record White Paper #1

The first white paper explores a different way of looking at Semantic Web from the air cargo ecosystem perspective. It investigates why the usage of concepts as Linked Data and ontologies can be beneficial in a distributed end-to-end digital logistics and transport chain, such as the one that the ONE Record standard enables.

Catch the Wave of Linked Data with ONE Record

Looking beyond the hype for real solutions to real problems

Advance your ONE Record knowledge & skills

Three great events to mark in your calendars

IATA WEBINAR

ONE Record Insights

Series of 6 webinars, every Tuesday from 23 June to 28 July, 11:00 to 12:30 CEST

HACKATHON

11-13 September

SPONSORED BY

Riege Software

Digital Cargo Conference 2020

Week of 14-18 Sept.

All digital! All action packed! All Complimentary!

Brought to you by IATA Digital Cargo team
More info: onerecord@iata.org

IATA

ONE Record Insights

Part 5

Q&A

Save the date

ONE Record Webinar
From June 23 to July 28
Every Tuesday 11:00-12:30

Hackathon
11-13 September

Digital Cargo Conference 2020
Week of 14-18 September

Thank You

More info

www.iata.org/one-record