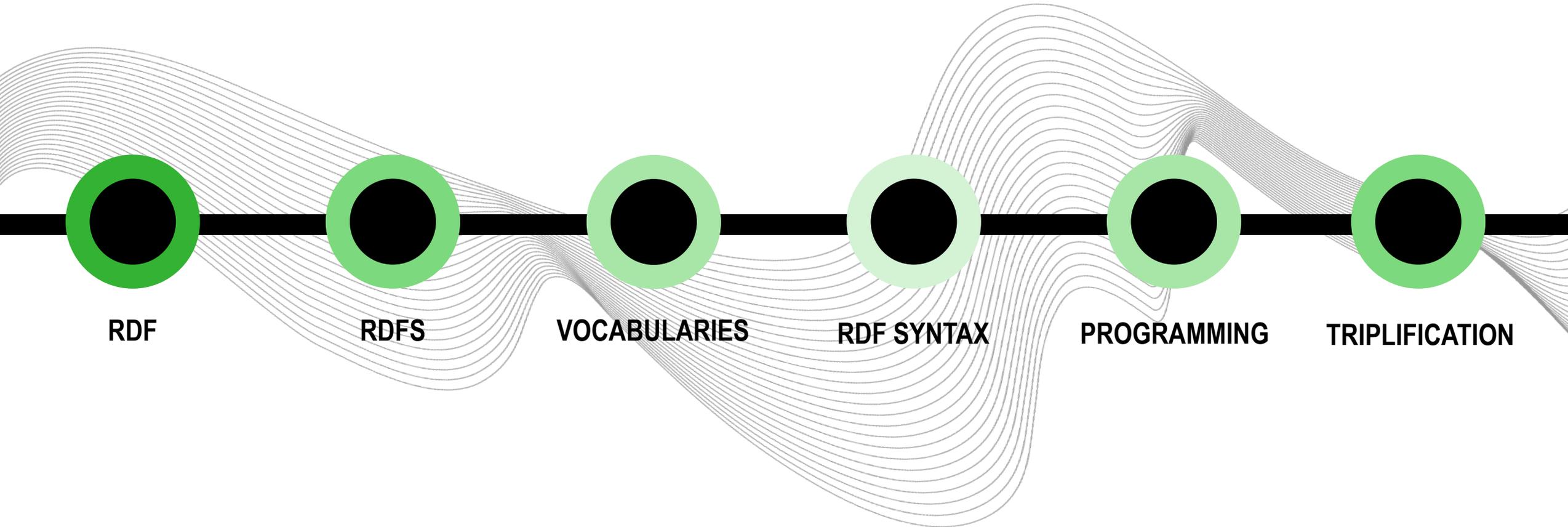


UNIVERSITY OF TWENTE.

LINKED DATA AND SEMANTIC WEB

LECTURE 3 – RESOURCE DESCRIPTION FRAMEWORK (RDF)

IN THIS PRESENTATION:



RDF

RDFS

VOCABULARIES

RDF SYNTAX

PROGRAMMING

TRIPLIFICATION

COURSE SCHEDULE

Week 3: <https://canvas.utwente.nl/courses/9784/pages/week-3>

today



Week	Topics	Lecture	Lab session	Lecturer
Week 1	Introduction	09/02 13:45 - 15:30	11/02 13:45 - 17:30	Luiz Olavo Bonino
Week 2	Ontology engineering	16/02 10:45 - 12:30	18/02 13:45 - 17:30	Joao Moreira / Luís Ferreira Pires
Week 3	W3C RDF, vocabularies, triplification	01/03 10:45 - 12:30	04/03 13:45 - 17:30	Joao Moreira
Week 4	Logics and W3C OWL	08/03 10:45 - 12:30	11/03 13:45 - 17:30	Luís Ferreira Pires
Week 5	Triple stores and W3C SPARQL	15/03 10:45 - 12:30	18/03 13:45 - 17:30	Erwin Folmer / Wouter Beek
Week 6	Ontology matching	22/03 10:45 - 12:30*	25/03 13:45 - 17:30	Shenghui Wang
Week 7	RDF validation with SHACL	29/03 10:45 - 12:30*	01/04 13:45 - 17:30	Luiz Olavo Bonino
Week 8	Applications	05/04 10:45 - 12:30*	08/04 13:45 - 17:30	Shenghui Wang
Week 9	Project presentations	13/04 13:45 - 17:30	-	
Week 10	Exam	20/04 13:45 - 16:30	-	

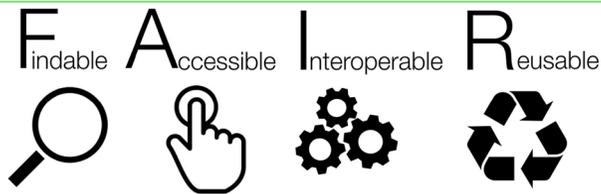
DEFINITIONS

Interoperability

“ability of two or more systems or components to exchange information and to use the information that has been exchanged” (IEEE, 1990)

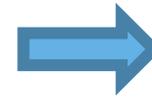
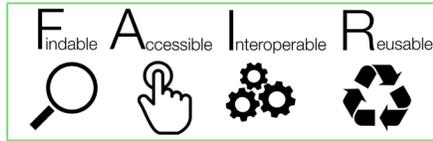
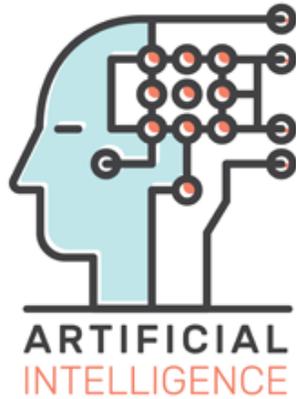
Smantic interoperability

“focus on interpretation of shared data in an unambiguously way, ensuring that the understanding of the information is the same for senders and receivers” (Heiler, 1995)



- I1.** (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2.** (meta)data use vocabularies that follow FAIR principles;
- I3.** (meta)data include qualified references to other (meta)data;

RDF FOR AI AND INTEROPERABILITY



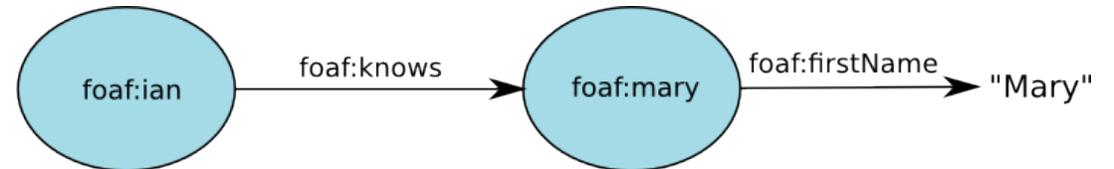
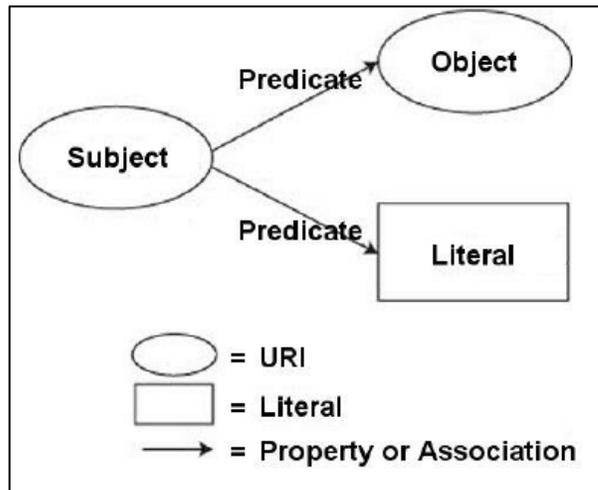
Fully AI Ready

“The carrot is that FAIR data allow much more *effective artificial intelligence* (...), which will open up unprecedented opportunities and increase *reproducibility*”

<https://www.nature.com/articles/d41586-020-00505-7>

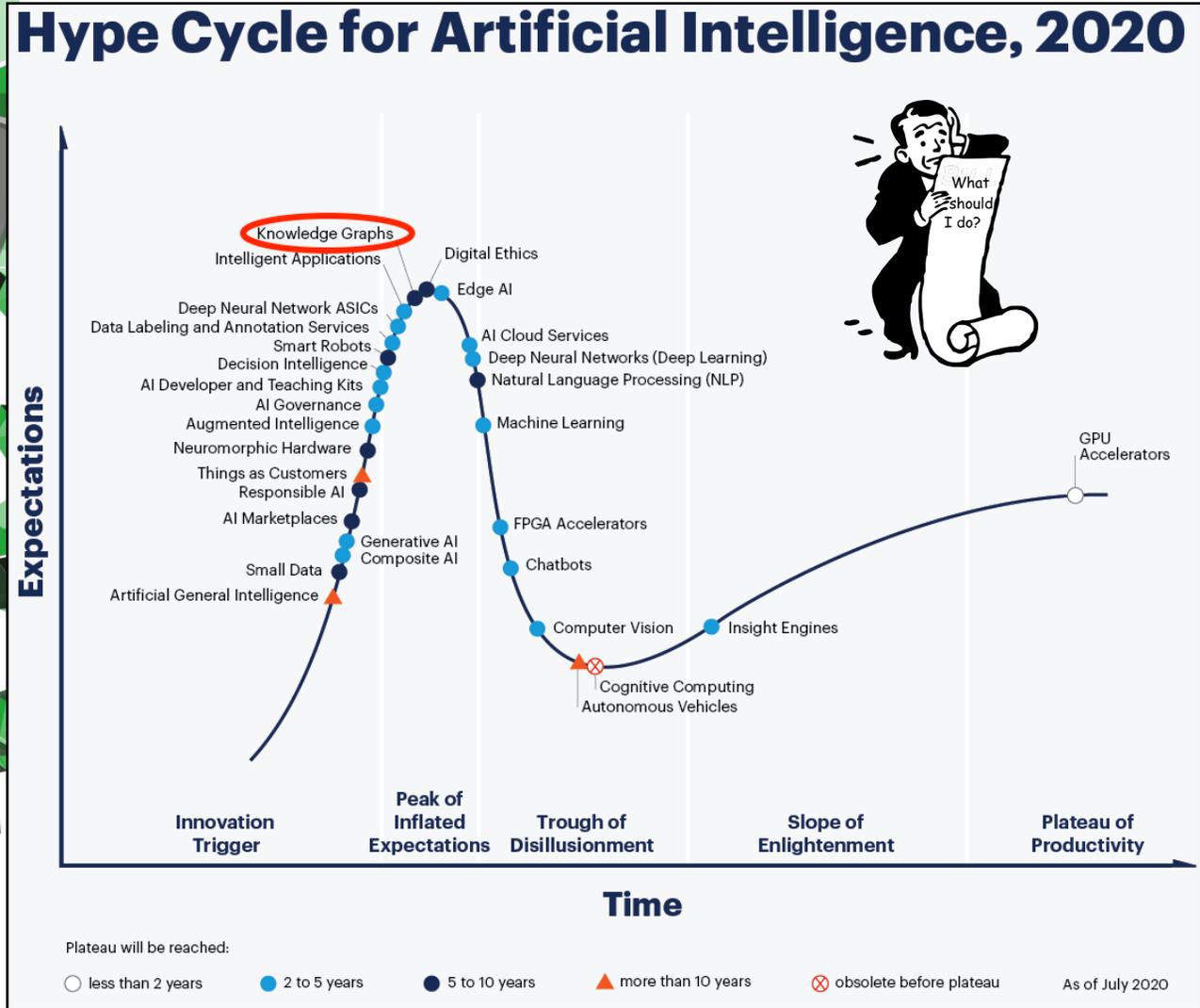


Resource Description Framework



```
foaf:ian foaf:knows foaf:mary.  
foaf:mary foaf:firstName "Mary".
```

KNOWLEDGE GRAPHS AND RDF TODAY



Industry (2020): “we need to use knowledge graphs!”

→ Nice name for an old idea... In 1993 there were people already using the term “ontology” for data representation!

→ How Google call it nowadays?

Google Structured Data

→ Who else in industry uses these technologies (in production)? In academia: everywhere...



Rijksoverheid

... and many many others....

→ SW technologies: already pointed as the future of data science



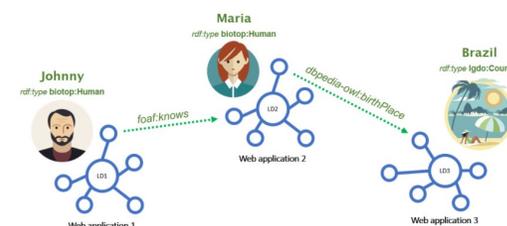
UNIVERSITY OF TWENTE.

RDF: RESOURCE DESCRIPTION FRAMEWORK

- **RDF stands for:**

- **Resource:** pages, concepts, ideas... (everything that can have a URI)
- **Description:** attributes, features and relations of the resources
- **Framework:** model, languages and syntaxes for these descriptions

“RDF is a standard model for data interchange on the Web”



An abstract syntax (a data model) for representing information in the Web.

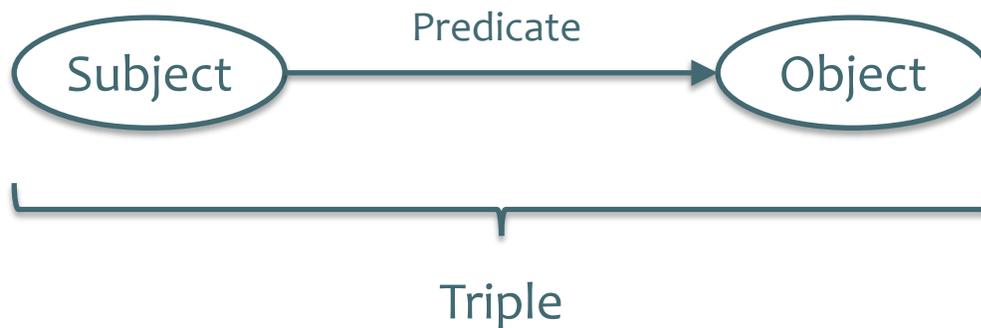
The abstract syntax has two key data structures:

- **RDF graphs:** sets of subject-predicate-object triples, where elements may be IRIs, blank nodes, or datatyped literals. Used to express descriptions of resources.
- **RDF datasets:** collections of RDF graphs, and comprise a default graph and zero or more named graphs.

Specification: <https://www.w3.org/TR/rdf11-concepts/>

RDF MODEL

RDF is a triple model, i.e., every piece of knowledge is broken down into *subject* (entity), *predicate* (attribute) and *object* (value)

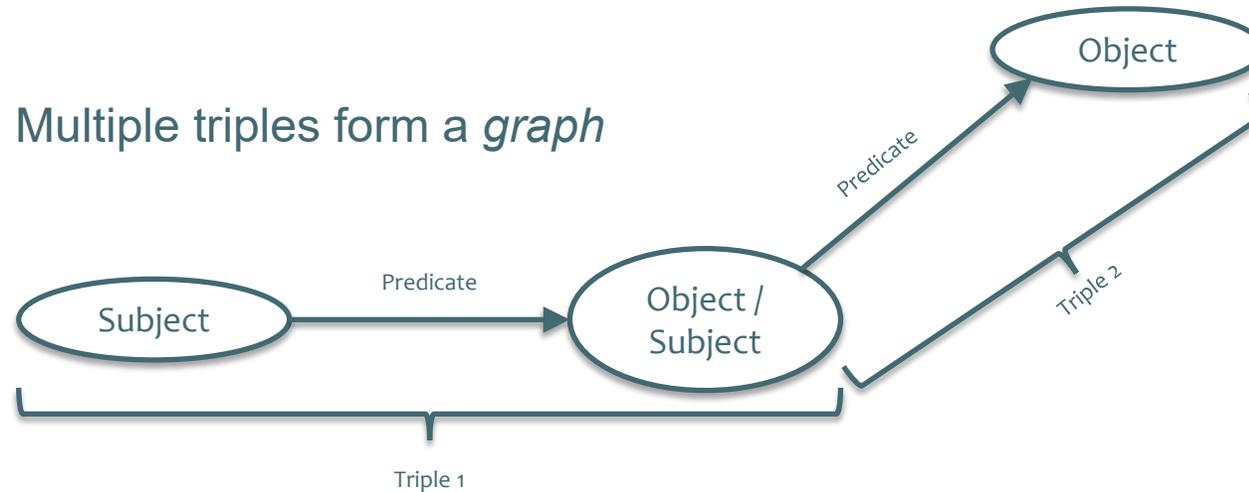


Subject: an [IRI](#) or a [blank node](#)

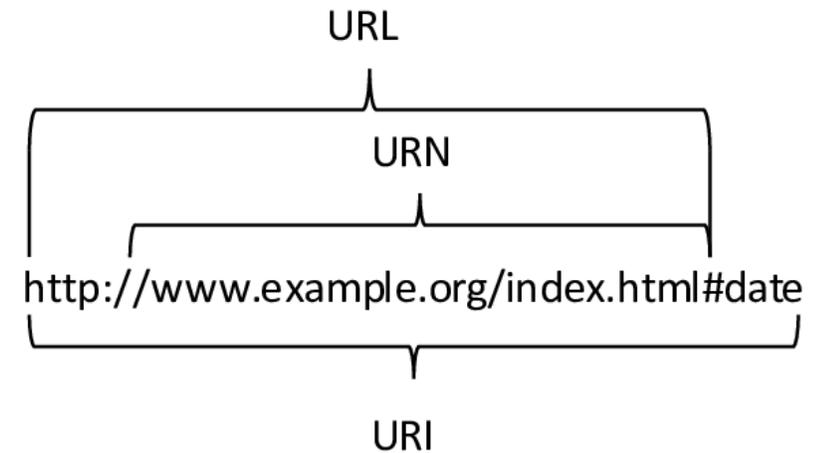
Predicate: an [IRI](#)

Object: an [IRI](#), a [literal](#) or a [blank node](#)

RDF GRAPH



- **IRI:** Internationalized Resource Identifier
- **URI:** Uniform Resource Identifier – *“unique sequence of characters that identifies a logical or physical resource used by web technologies”*
→ IRIs are a generalization of URIs.
- **URN:** Uniform Resource Name is a globally unique persistent identifier assigned within defined namespaces.



RDF MODEL

- **Literals:** used for datatypes - values such as strings, numbers, and dates
- **Blank node:** are local identifiers that are used in some concrete RDF syntaxes or RDF store implementations

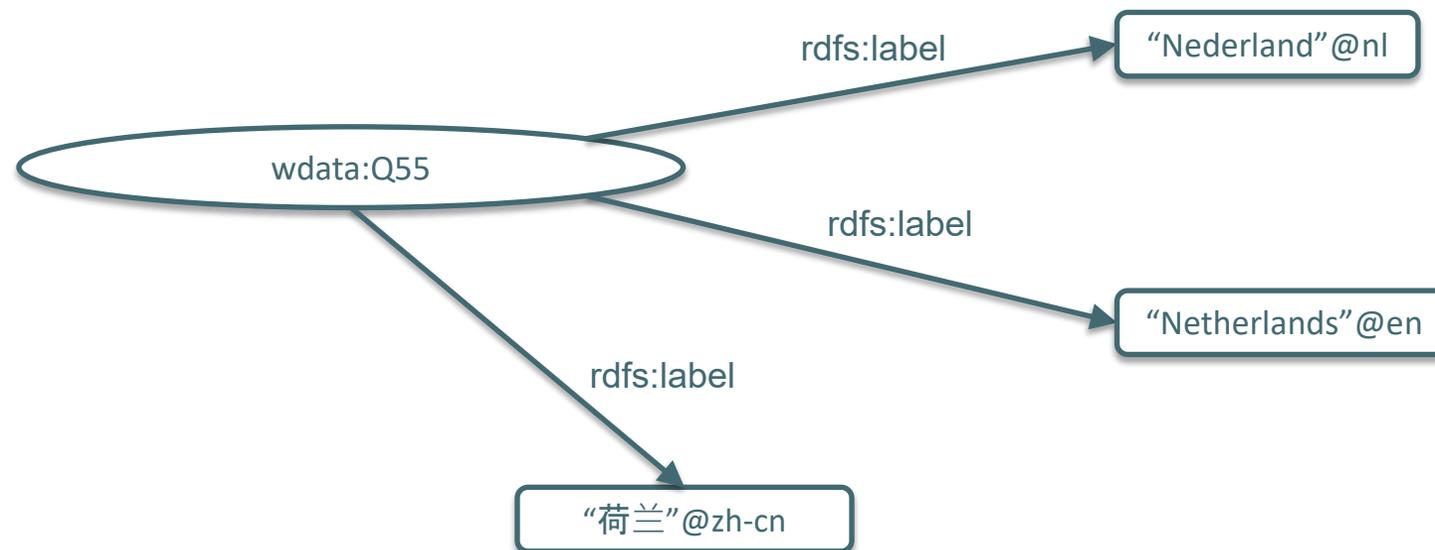
Datatypes are used with RDF literals. The datatype abstraction used in RDF is compatible with XML Schema. → check in Protégé!

A list of the RDF-compatible XSD types, with short descriptions"

	Datatype	Value space (informative)
Core types	xsd:string	Character strings (but not all Unicode character strings)
	xsd:boolean	true, false
	xsd:decimal	Arbitrary-precision decimal numbers
	xsd:integer	Arbitrary-size integer numbers
IEEE floating-point numbers	xsd:double	64-bit floating point numbers incl. $\pm\text{Inf}$, ± 0 , NaN
	xsd:float	32-bit floating point numbers incl. $\pm\text{Inf}$, ± 0 , NaN
Time and date	xsd:date	Dates (yyyy-mm-dd) with or without timezone
	xsd:time	Times (hh:mm:ss.sss...) with or without timezone
	xsd:dateTime	Date and time with or without timezone

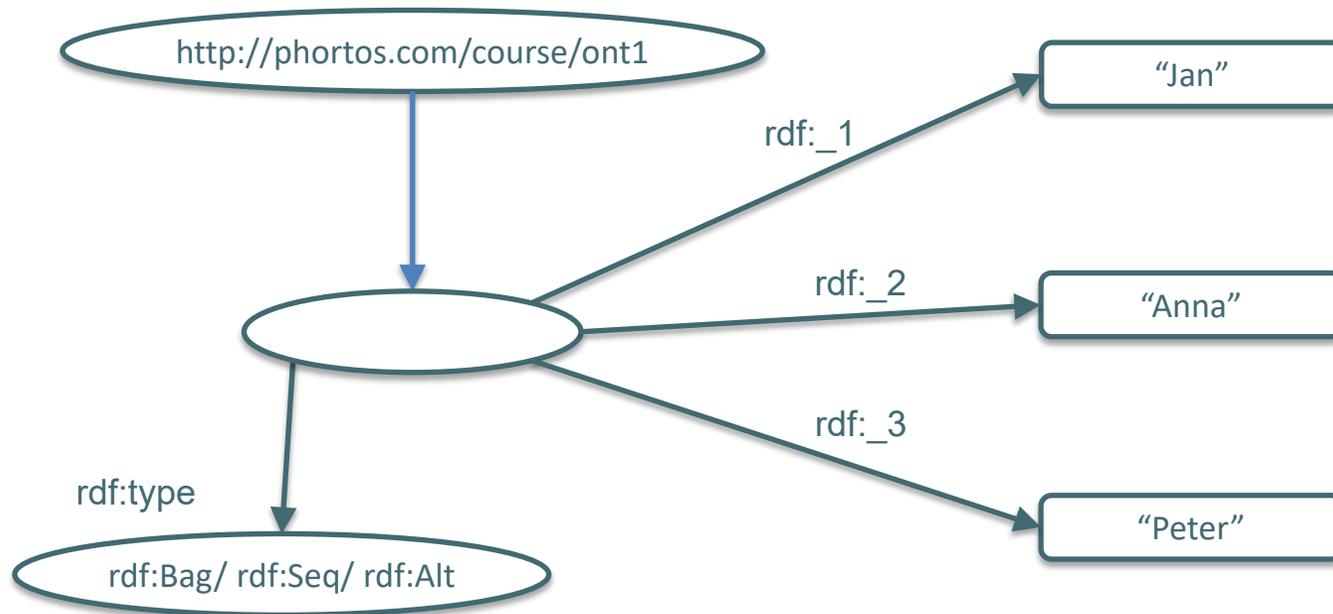
RDF LITERALS

- Literals can be plain or typed using arbitrary datatypes
- Support for multilingual labels
- Plain literals may have a language tag, as defined by RFC 3066



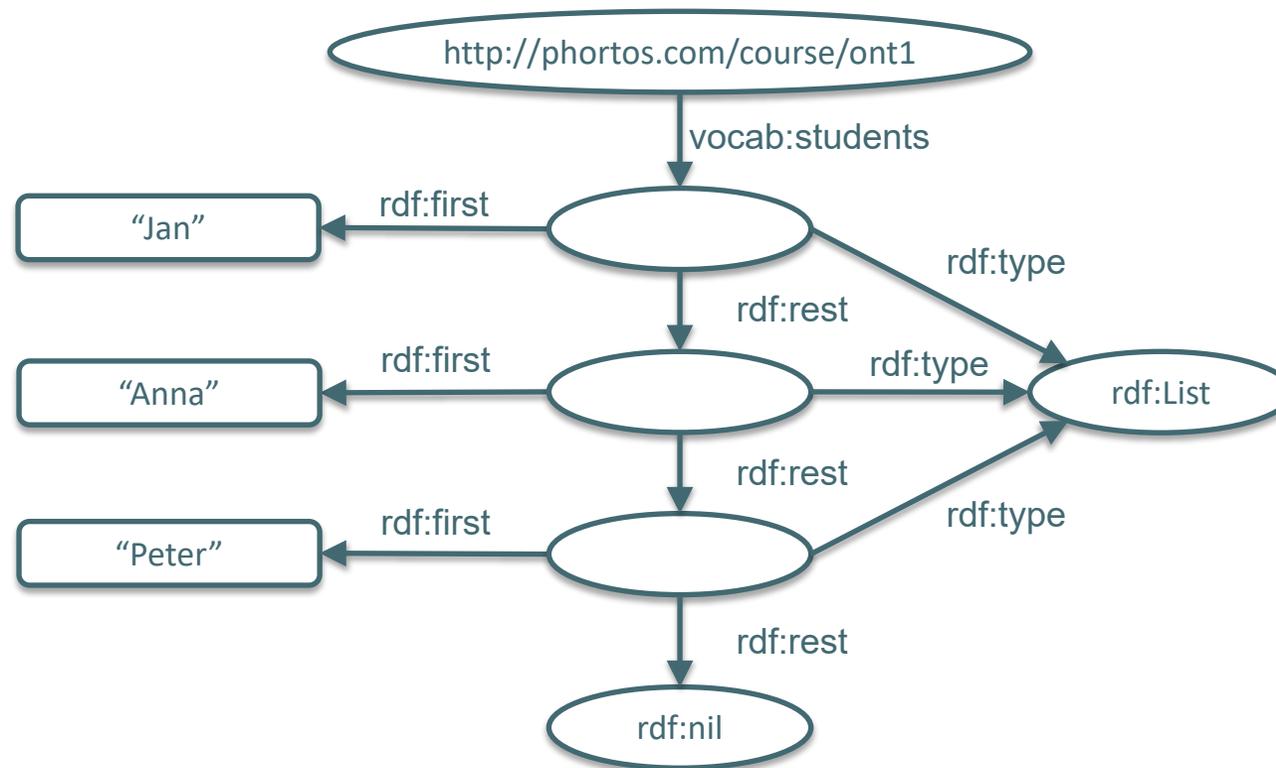
RDF SETS

- Containers allow to describe groups of things
- Containers are open. There is no mechanism to know the number of members
- Bag (unordered set), Seq (ordered set), Alt (choice)



RDF COLLECTIONS

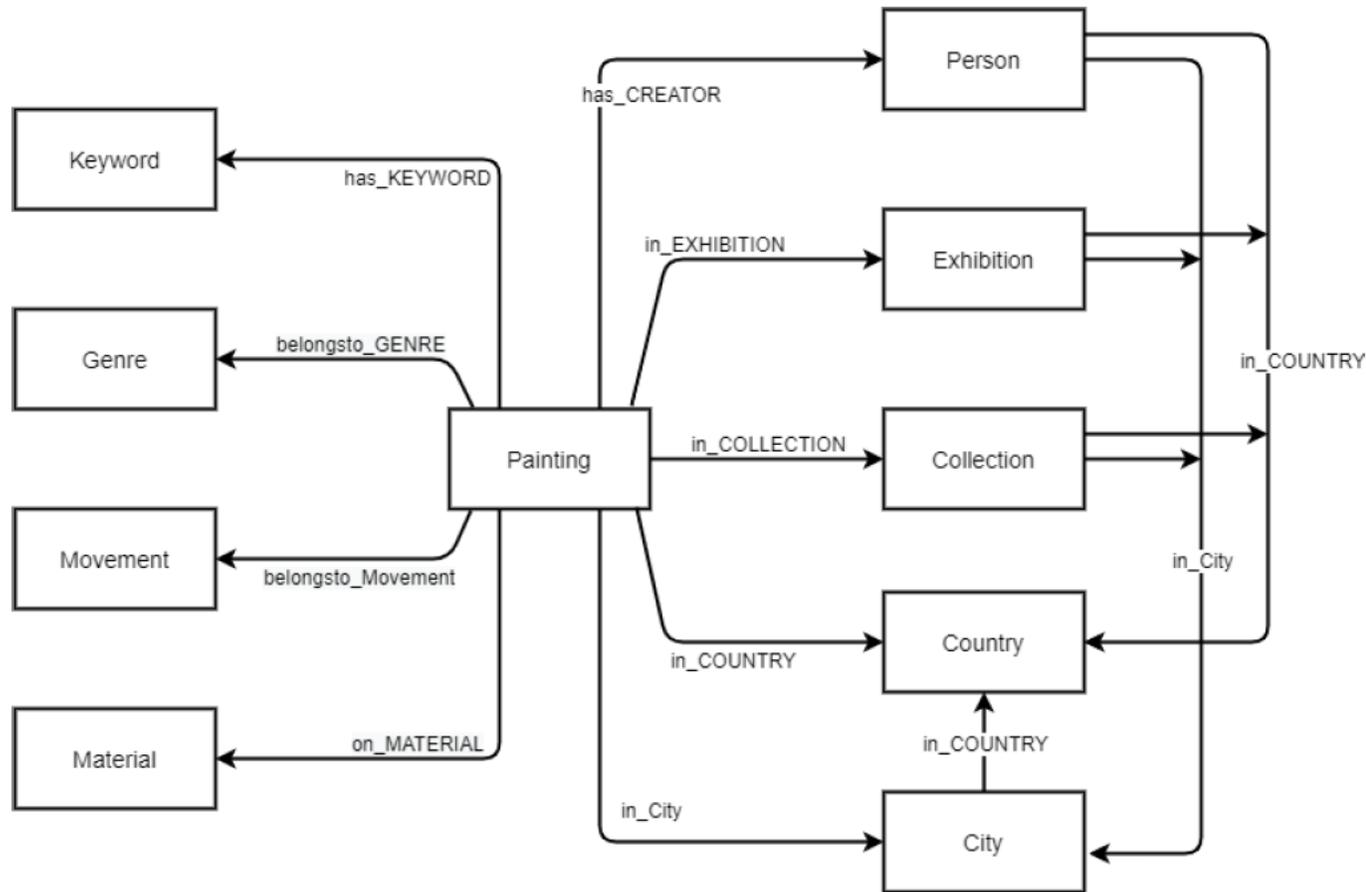
- Collections allow to describe groups of things
- Collections are closed, they can have no more members
- Type: ***rdf:List*** → Properties: *rdf:first*, *rdf:rest*, *rdf:nil*



RDF: EXAMPLE

Wikidata profile for *virtual museums*

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



Entity type	Property	Count
Painting	Name	2990
	Description	
	Image	
	Date	
	Height	
	Width	
	Sitelink URI	
Collection	Name	334
	Description	
	Image	
	Date	
	Website	
	Collection_size	
	Visitor_per_year Sitelink URI	
Genre	Name	29
	Description	
	Sitelink	
	URI	

Thanks to Dou Liu, Msc

RDFS: RDF SCHEMA

- Provides a data-modelling vocabulary for RDF data. RDF Schema is an extension of the basic RDF vocabulary.

→ *Different from XML Schema and JSON Schema!*

- ✓ Extends RDF with the possibility to define classes and associated properties
- ✓ Allows RDF apps to agree on a common data description vocabulary

2. Classes

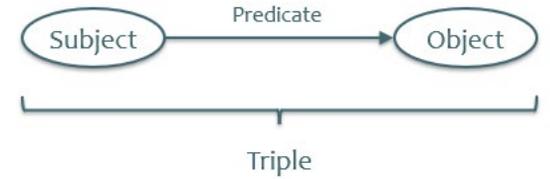
- 2.1 rdfs:Resource
- 2.2 rdfs:Class
- 2.3 rdfs:Literal
- 2.4 rdfs:Datatype
- 2.5 rdf:langString
- 2.6 rdf:HTML
- 2.7 rdf:XMLLiteral
- 2.8 rdf:Property

3. Properties

- 3.1 rdfs:range
- 3.2 rdfs:domain
- 3.3 rdf:type
- 3.4 rdfs:subClassOf
- 3.5 rdfs:subPropertyOf
- 3.6 rdfs:label
- 3.7 rdfs:comment

Specification: <https://www.w3.org/TR/rdf-schema/>

RDFS: MAIN CLASSES



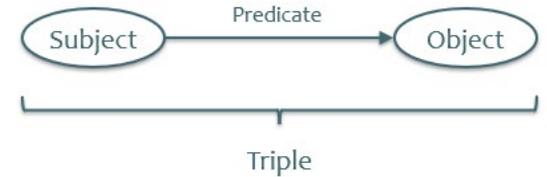
rdfs:Resource - all things described by RDF are called resources, and are instances of this class. All other classes are subclasses of this class. rdfs:Resource is an instance of rdfs:Class.

rdfs:Class - is the class of resources that are RDF classes. rdfs:Class is an instance of rdfs:Class.

rdf:Property - is the class of RDF properties. rdf:Property is an instance of rdfs:Class.

Observation: *RDF sets and collections are actually defined by RDFS! For example: rdfs:Container, rdf:Bag, rdf:Seq, rdf>List, etc*

RDFS: MAIN PROPERTIES



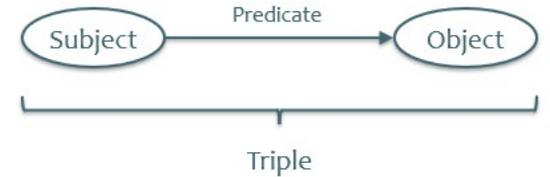
rdfs:range - an instance of `rdf:Property` used to state that the values of a property are instances of one or more classes. Example: ***P rdfs:range C***
→ P is an instance of the class `rdf:Property`, C is an instance of the class `rdfs:Class`, and the resources denoted by the objects of triples whose predicate is P are instances of the class C.

rdfs:domain - an instance of `rdf:Property` used to state that any resource that has a given property is an instance of one or more classes. Example: ***P rdfs:domain C***
→ P is an instance of the class `rdf:Property`, C is a instance of the class `rdfs:Class` and the resources denoted by the subjects of triples whose predicate is P are instances of the class C.

rdf:type - an instance of `rdf:Property` used to state that a resource is an instance of a class. Example: ***R rdf:type C*** → C is an instance of `rdfs:Class` and R is an instance of C.
→ The `rdfs:domain` of `rdf:type` is `rdfs:Resource`. The `rdfs:range` of `rdf:type` is `rdfs:Class`.

rdfs:subClassOf - an instance of `rdf:Property` used to state that all the instances of one class are instances of another. Example: ***C1 rdfs:subClassOf C2***
→ C1 is an instance of `rdfs:Class`, C2 is an instance of `rdfs:Class` and C1 is a subclass of C2. The `rdfs:subClassOf` property is transitive.

RDFS: MAIN PROPERTIES



rdf:type - an instance of `rdf:Property` used to state that a resource is an instance of a class.

Example: ***R rdf:type C*** \rightarrow C is an instance of `rdfs:Class` and R is an instance of C.

\rightarrow The `rdfs:domain` of `rdf:type` is `rdfs:Resource`. The `rdfs:range` of `rdf:type` is `rdfs:Class`.

rdfs:subClassOf - an instance of `rdf:Property` used to state that all the instances of one class are instances of another. Example: ***C1 rdfs:subClassOf C2***

\rightarrow C1 is an instance of `rdfs:Class`, C2 is an instance of `rdfs:Class` and C1 is a subclass of C2. The `rdfs:subClassOf` property is transitive.

Important: recall from UML - Object Orientation on generalization and specialization - the difference between: ***[instance] vs [subclass]***

Example:

\rightarrow João is an instance of the class Person

\rightarrow Lecturer is a subclass of the class Person (if all lecturers are persons! 😊)

And if we consider parallel universes, is João a class or an instance?



RDF EXPRESSIVENESS



Term List	Synonym Ring	Authority File	Taxonomy	Thesaurus	Ontology
Ambiguity control	Synonym control	Ambiguity control Synonym control	Ambiguity control (Synonym control) Hierarchical relationships	Ambiguity control Synonym control Hierarchical relationships Associative relationships	Ambiguity control Attributes Semantic relationships Classes

POPULAR VOCABULARIES

- FOAF: Friend of a friend
- DCT: Dublin Core Terms
- Schema.org
- SKOS: Simple Knowledge Organization System

```
<A> rdf:type skos:Concept ;  
    skos:prefLabel "love"@en ;  
    skos:altLabel "adoration"@en ;  
    skos:broader <B> ;  
    skos:inScheme <S> .
```

```
<B> rdf:type skos:Concept ;  
    skos:prefLabel "emotion"@en ;  
    skos:altLabel "feeling"@en ;  
    skos:topConceptOf <S> .
```

```
<S> rdf:type skos:ConceptScheme ;  
    dct:title "My First Thesaurus" ;  
    skos:hasTopConcept <B> .
```

Table 1. SKOS Vocabulary

URI	Definition
skos:Concept	Section 3. The skos:Concept Class
skos:ConceptScheme	Section 4. Concept Schemes
skos:inScheme	Section 4. Concept Schemes
skos:hasTopConcept	Section 4. Concept Schemes
skos:topConceptOf	Section 4. Concept Schemes
skos:altLabel	Section 5. Lexical Labels
skos:hiddenLabel	Section 5. Lexical Labels
skos:prefLabel	Section 5. Lexical Labels
skos:notation	Section 6. Notations
skos:changeNote	Section 7. Documentation Properties
skos:definition	Section 7. Documentation Properties
skos:editorialNote	Section 7. Documentation Properties
skos:example	Section 7. Documentation Properties
skos:historyNote	Section 7. Documentation Properties

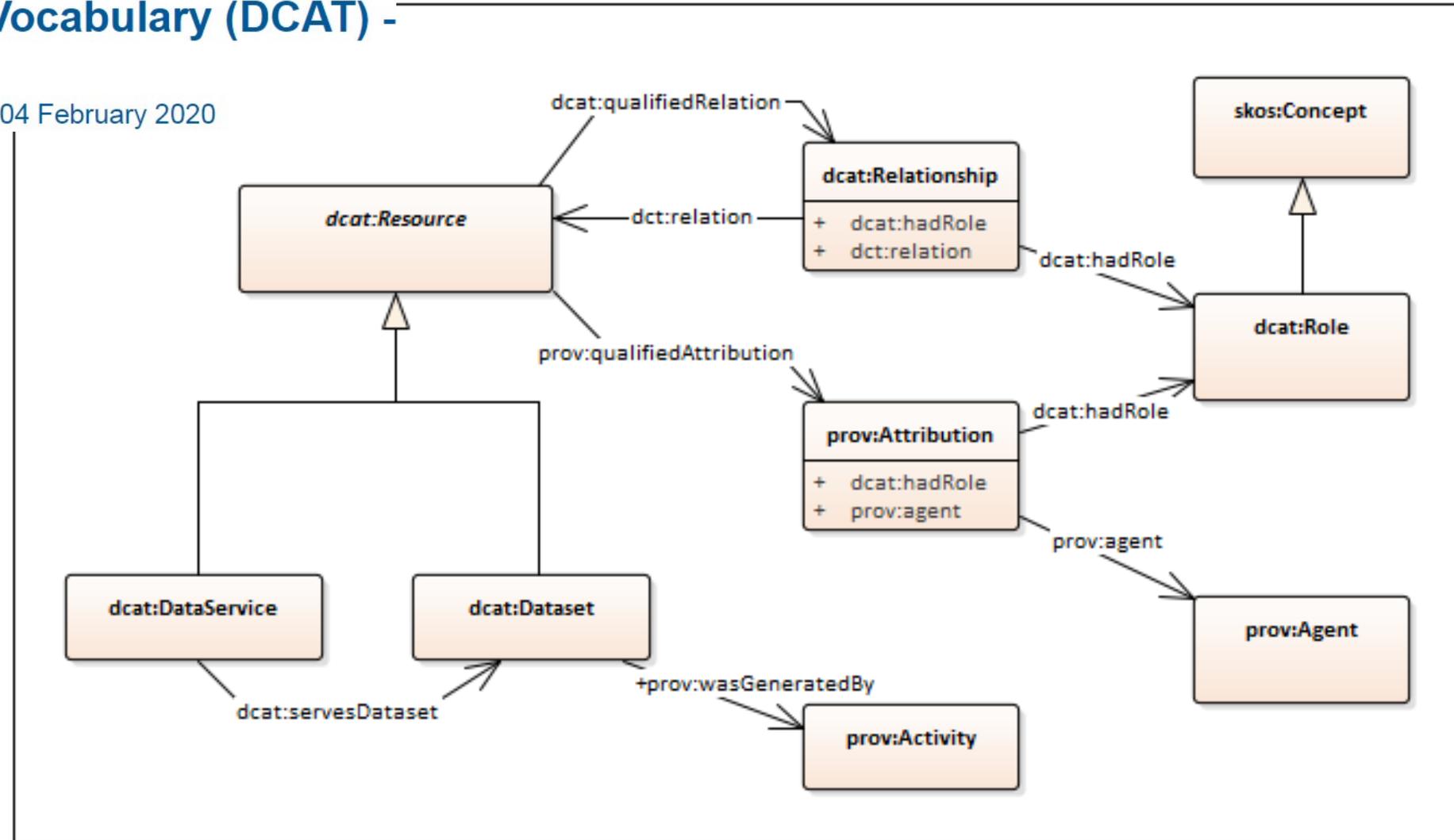
SOME RELEVANT STANDARDS IN RDF



Data Catalog Vocabulary (DCAT) - Version 2

W3C Recommendation 04 February 2020

<https://www.w3.org/TR/vocab-dcat-2/>



DCAT incorporates terms from pre-existing vocabularies where stable terms with appropriate meanings could be found:

- FOAF (friend of a friend)
- DCT (Dublin Core Terms)
- Schema.org

SOME RELEVANT STANDARDS IN RDF



<https://www.w3.org/TR/prov-overview/>

Expanded classes and properties provide additional terms that can be used to relate classes in the Starting Point category. The terms in the Starting Point category. Many of the terms in this category are subclasses or subproperties of those in the Starting Point category listed below and are discussed in [Section 3.2](#).

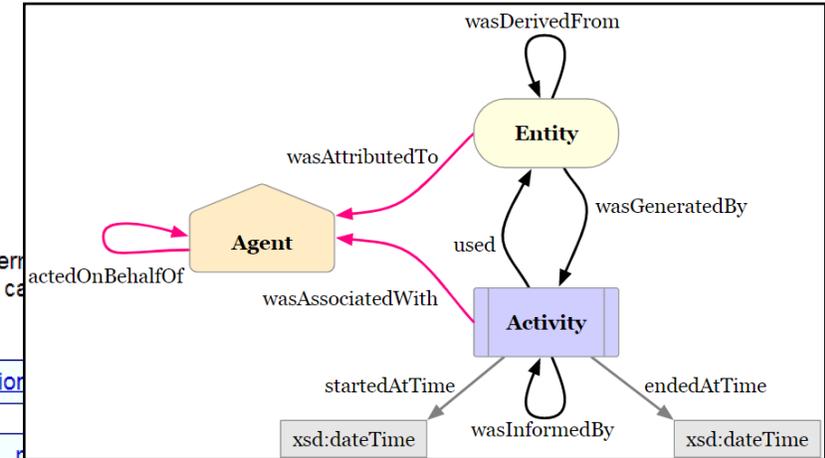
[prov:Collection](#) [prov:EmptyCollection](#) [prov:Bundle](#) [prov:Person](#) [prov:SoftwareAgent](#) [prov:Organization](#) [prov:Location](#)

[prov:alternateOf](#) [prov:specializationOf](#) [prov:generatedAtTime](#) [prov:hadPrimarySource](#) [prov:value](#)
[prov:invalidatedAtTime](#) [prov:wasInvalidatedBy](#) [prov:hadMember](#) [prov:wasStartedBy](#) [prov:wasEndedBy](#) [prov:invalidated](#) [prov:influenced](#) [prov:atLocation](#)
[prov:generated](#)

Qualified classes and properties provide elaborated information about binary relations asserted using Starting Point and Expanded properties. The terms in this category are applied using a pattern that differs from those in the Starting Point and Expanded categories. While the relations from the previous two categories are applied as direct, binary assertions, the terms in this category are used to provide additional attributes of the binary relations. The pattern used in this category allows users to provide elaborate details that are not available using only Starting Point and Expanded terms. The classes and properties in this category are listed below and are discussed in [Section 3.3](#).

[prov:Influence](#) [prov:EntityInfluence](#) [prov:Usage](#) [prov:Start](#) [prov:End](#) [prov:Derivation](#) [prov:PrimarySource](#) [prov:Quotation](#) [prov:Revision](#)
[prov:ActivityInfluence](#) [prov:Generation](#) [prov:Communication](#) [prov:Invalidation](#) [prov:AgentInfluence](#) [prov:Attribution](#) [prov:Association](#) [prov:Plan](#) [prov:Delegation](#)
[prov:InstantaneousEvent](#) [prov:Role](#)

[prov:wasInfluencedBy](#) [prov:qualifiedInfluence](#) [prov:qualifiedGeneration](#) [prov:qualifiedDerivation](#) [prov:qualifiedPrimarySource](#) [prov:qualifiedQuotation](#)
[prov:qualifiedRevision](#) [prov:qualifiedAttribution](#) [prov:qualifiedInvalidation](#) [prov:qualifiedStart](#) [prov:qualifiedUsage](#) [prov:qualifiedCommunication](#)
[prov:qualifiedAssociation](#) [prov:qualifiedEnd](#) [prov:qualifiedDelegation](#) [prov:influencer](#) [prov:entity](#) [prov:hadUsage](#) [prov:hadGeneration](#) [prov:activity](#) [prov:agent](#)
[prov:hadPlan](#) [prov:hadActivity](#) [prov:atTime](#) [prov:hadRole](#)

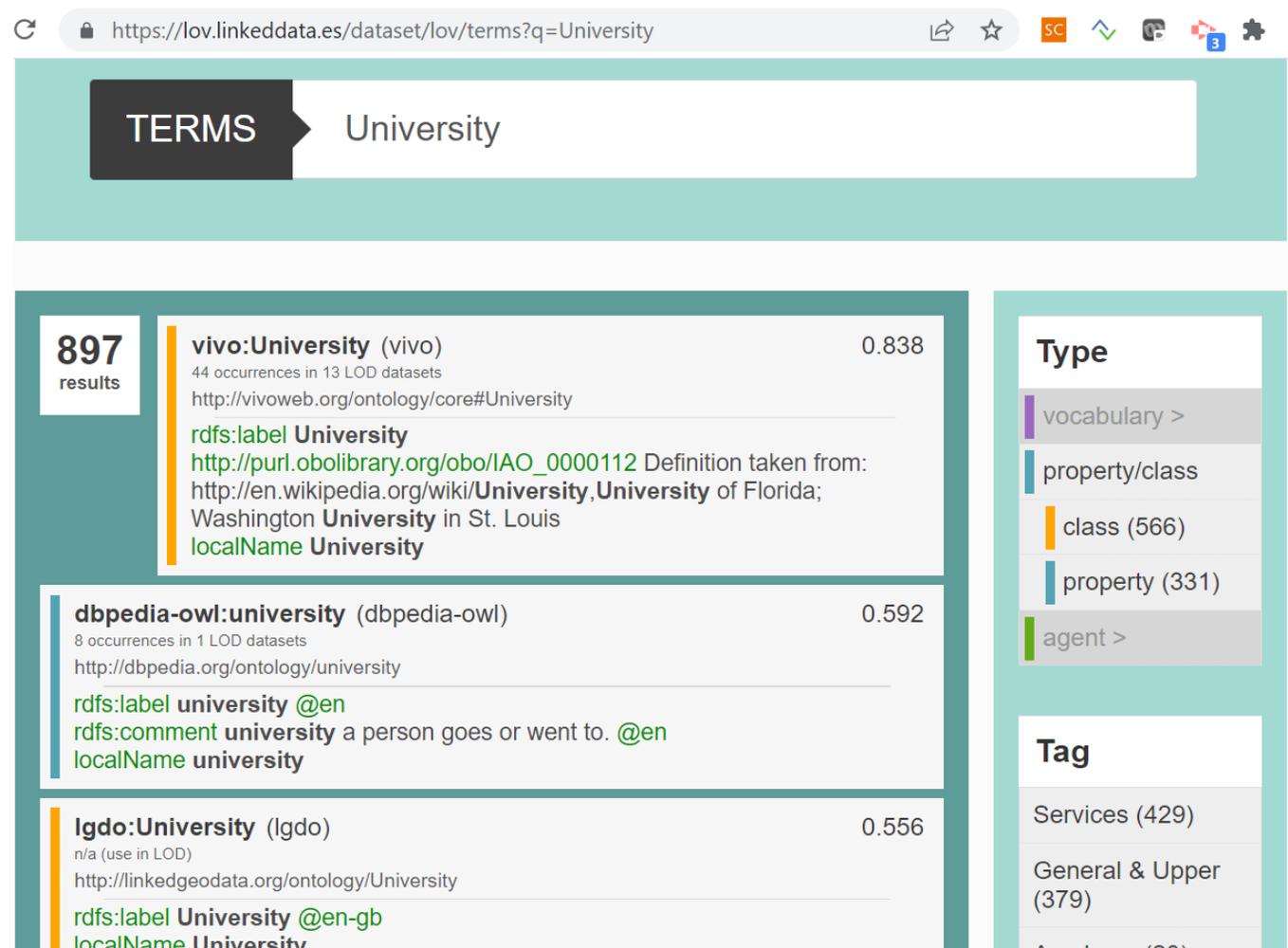


LINKED OPEN VOCABULARIES (LOV)

A vocabulary in LOV: definitions of a set of classes and properties (terms of the vocabulary = links) to describe specific types of things. The definitions of terms provided by the vocabularies bring clear semantics to descriptions and links, thanks to the formal language used (RDF).

→ vocabularies provide the semantic glue enabling Data to become meaningful Data.

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



The screenshot shows a web browser displaying the LOV website. The URL is <https://lov.linkeddata.es/dataset/lov/terms?q=University>. The page features a search bar with the term 'University' and a 'TERMS' button. Below the search bar, there are three search results for 'University'.

Results	URI	Score
897 results	vivo:University (vivo) 44 occurrences in 13 LOD datasets http://vivoweb.org/ontology/core#University rdfs:label University http://purl.obolibrary.org/obo/IAO_0000112 Definition taken from: http://en.wikipedia.org/wiki/University , University of Florida; Washington University in St. Louis localName University	0.838
	dbpedia-owl:university (dbpedia-owl) 8 occurrences in 1 LOD datasets http://dbpedia.org/ontology/university rdfs:label university @en rdfs:comment university a person goes or went to. @en localName university	0.592
	lgdo:University (lgdo) n/a (use in LOD) http://linkedgeodata.org/ontology/University rdfs:label University @en-gb localName University	0.556

On the right side of the page, there are two panels: 'Type' and 'Tag'. The 'Type' panel shows a hierarchy: 'vocabulary >' (selected), 'property/class', 'class (566)', 'property (331)', and 'agent >'. The 'Tag' panel shows 'Services (429)', 'General & Upper (379)', and 'Academy (20)'.

RDF AND FAIR PRINCIPLES

Interoperable:

I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.

>> **RDF is a broadly applicable formal language for semantic interoperability**

I2. (meta)data use vocabularies that follow FAIR principles;

>> **Preference to standardized controlled vocabularies and domain ontologies, e.g., from W3C (SSN/SOSA), ETSI (SAREF) , HL7 (FHIR), etc.**

I3. (meta)data include qualified references to other (meta)data;

>> **Meaningful links between (meta)data through data and object properties (RDF) based on W3C DCAT and PROV**



RDF SYNTAXES (SERIALIZATIONS)

- **RDF/XML**
 - RDF syntax based on XML
- **Turtle**
 - Superset of N-Triples, with lots of shortcuts to facilitate readability
- **JSON-LD**
 - Represent RDF using JSON notation
- **N-Triples**
 - Simple syntax for line-delimited triples
- **Notation3 (N3)**
 - Superset of Turtle and RDF. Contains non-RDF features like rules, scoping graphs, etc.
- **RDFa**
 - Syntax for embedding RDF directly into XHTML documents



RDF/XML SYNTAX SPEC

An XML syntax for RDF in terms of Namespaces

→ Initially (2004) the main RDF syntax

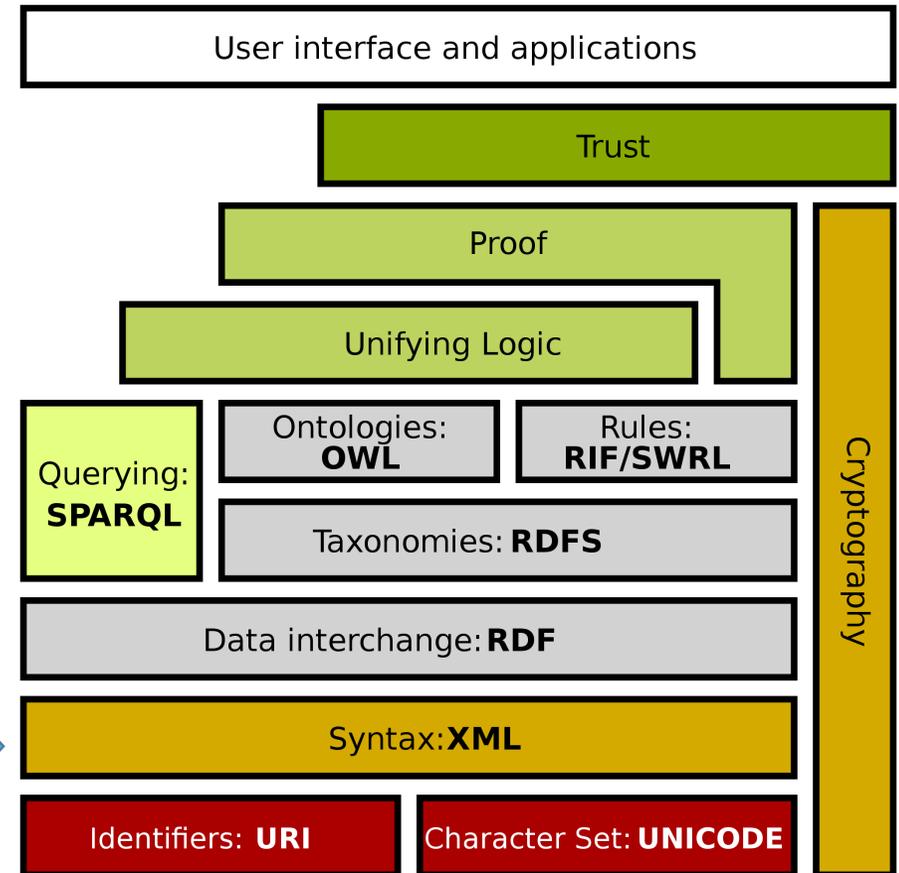
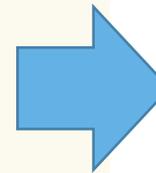
EXAMPLE 19

Complete example of a RDF collection of nodes using `rdf:parseType="Collection"` ([example19.rdf](#), output [example19.nt](#))

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/stuff/1.0/">

  <rdf:Description rdf:about="http://example.org/basket">
    <ex:hasFruit rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/banana"/>
      <rdf:Description rdf:about="http://example.org/apple"/>
      <rdf:Description rdf:about="http://example.org/pear"/>
    </ex:hasFruit>
  </rdf:Description>

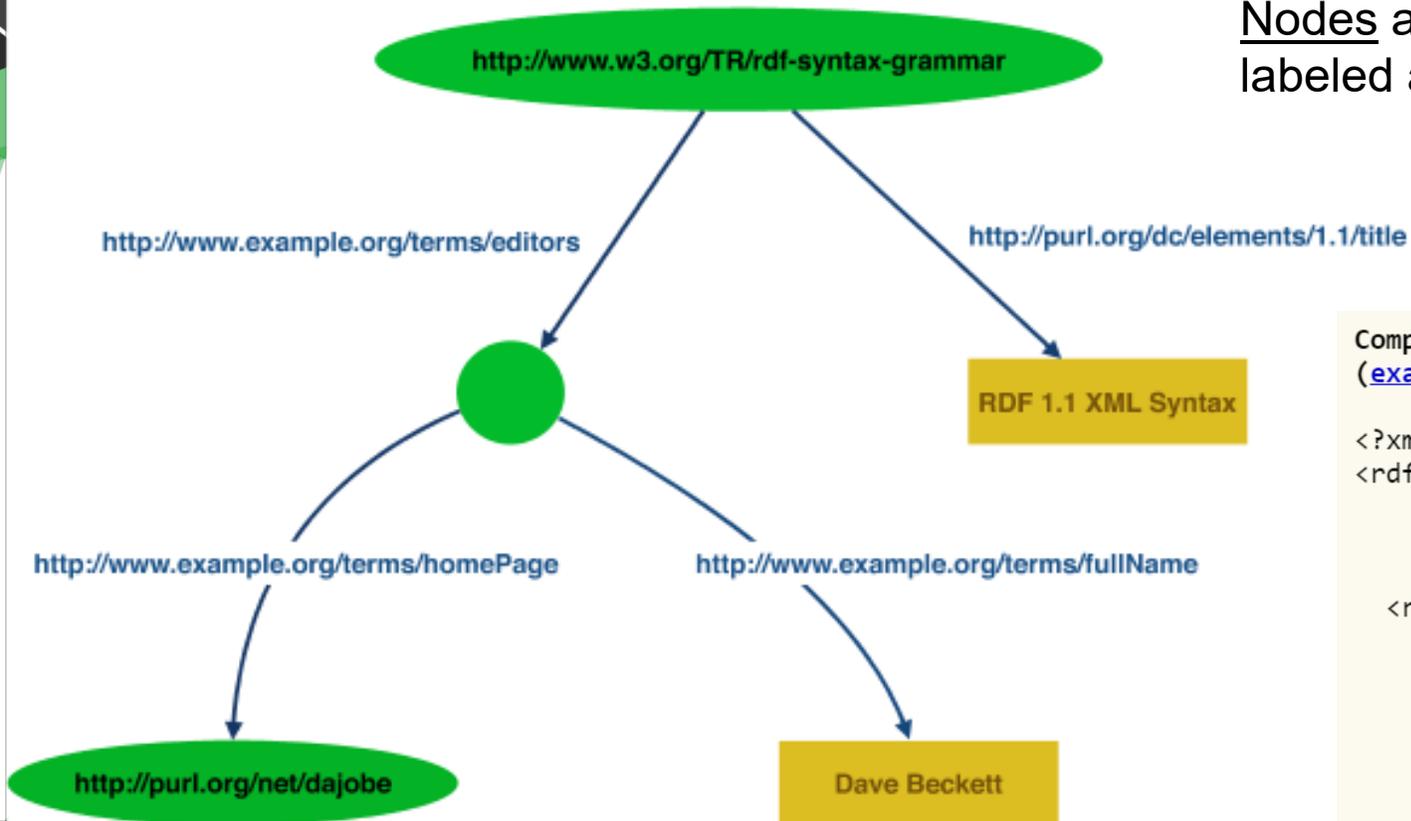
</rdf:RDF>
```



Specification: <https://www.w3.org/TR/rdf-syntax-grammar/>

RDF/XML SERIALIZATION SPEC

Nodes are represented as ovals (IRIs), and predicates as labeled arcs (IRIs) and string literals nodes in rectangles.



Complete RDF/XML description of Figure 1 graph
([example07.rdf](#), output [example07.nt](#))

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://example.org/stuff/1.0/">

  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
    dc:title="RDF1.1 XML Syntax">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett">
        <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
      </rdf:Description>
    </ex:editor>
  </rdf:Description>

</rdf:RDF>
```

RDF/TURTLE SERIALIZATION SPEC

Textual syntax for RDF that allows an RDF graph to be completely written in a compact and natural text form, with abbreviations for common usage patterns and datatypes.

RDF/Turtle (TTL)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [
    ex:fullname "Dave Beckett";
    ex:homePage <http://purl.org/net/dajobe/>
  ] .
```

RDF/XML

Complete RDF/XML description of Figure 1 graph
([example07.rdf](#), output [example07.nt](#))

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:ex="http://example.org/stuff/1.0/">

  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
                 dc:title="RDF1.1 XML Syntax">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett">
        <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
      </rdf:Description>
    </ex:editor>
  </rdf:Description>

</rdf:RDF>
```

Specification: <https://www.w3.org/TR/turtle/>

RDF/JSON-LD SERIALIZATION SPEC

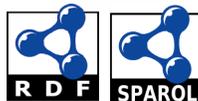
W3C Recommendation

JSON-LD 1.1

A JSON-based Serialization for Linked Data



W3C Recommendation 16 July 2020



This version:

<https://www.w3.org/TR/2020/REC-json-ld11-20200716/>

<https://json-ld.org/playground/>

```
{  
  "@context": "http://schema.org/",  
  "@type": "Person",  
  "name": "Hohnny",  
  "jobTitle": "Assistant Professor",  
  "telephone": "(425) 123-4567",  
  "url": "https://json-ld.org/"  
}
```

- A **lightweight syntax** to serialize RDF as JSON
- Primarily intended to be a way to use LD in **Web-based programming environments** for **interoperable Web services**
 - ✓ universal identifier mechanism for JSON objects (IRIs)
 - ✓ disambiguate keys shared among different JSON docs (context)
 - ✓ express directed graphs (e.g., a social network) in a single document

RDF/JSON-LD EXAMPLE WITH SCHEMA.ORG



```
{
  "@context": "https://schema.org",
  "@type": "Product",
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": "3.5",
    "reviewCount": "11"
  },
  "description": "0.7 cubic feet countertop microwave. Has six",
  "name": "Kenmore White 17\" Microwave",
  "image": "kenmore-microwave-17in.jpg",
  "offers": {
    "@type": "Offer",
    "availability": "https://schema.org/InStock",
    "price": "55.00",
    "priceCurrency": "USD"
  },
  "review": [
    {
      "@type": "Review",
      "author": "Ellie",
      "datePublished": "2011-04-01",
      "reviewBody": "The lamp burned out and now I have to repl",
      "name": "Not a happy camper",
      "reviewRating": {
        "@type": "Rating",
        "bestRating": "5",
        "ratingValue": "1",

```

<https://schema.org/Product>

GOOGLE DATASETS

Google: structured search



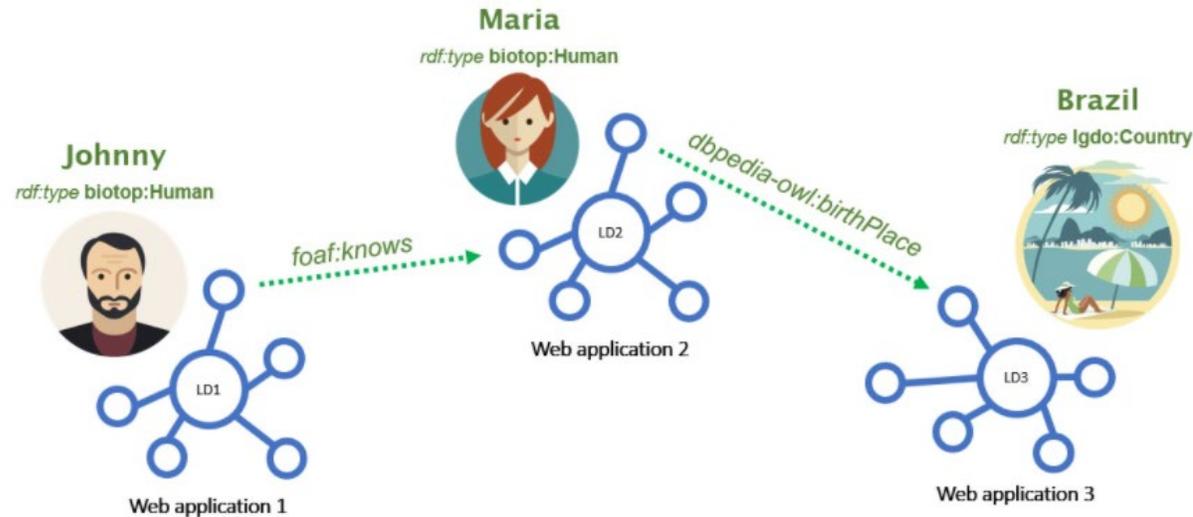
<https://developers.google.com/search/docs/guides/intro-structured-data>

<https://developers.google.com/search/docs/data-types/dataset>

```
<body><html>
<head>
  <title>NCDC Storm Events Database</title>
  <script type="application/ld+json">
    {
      "@context": "https://schema.org/",
      "@type": "Dataset",
      "name": "NCDC Storm Events Database",
      "description": "Storm Data is provided by the National Weather Service (NWS)",
      "url": "https://catalog.data.gov/dataset/ncdc-storm-events-database",
      "sameAs": "https://gis.ncdc.noaa.gov/geoportal/catalog/search/resource/details.do?i=10.1000/182",
      "identifier": ["https://doi.org/10.1000/182", "https://identifiers.org/ark:/12345/fk1234"],
      "keywords": [
        "ATMOSPHERE > ATMOSPHERIC PHENOMENA > CYCLONES",
        "ATMOSPHERE > ATMOSPHERIC PHENOMENA > DROUGHT",
        "ATMOSPHERE > ATMOSPHERIC PHENOMENA > FOG",
        "ATMOSPHERE > ATMOSPHERIC PHENOMENA > FREEZE"
      ],
      "license": "https://creativecommons.org/publicdomain/zero/1.0/",
      "hasPart": [
        {
          "@type": "Dataset",
          "name": "Sub dataset 01",
          "description": "Informative description of the first subdataset...",
          "license": "https://creativecommons.org/publicdomain/zero/1.0/"
        },
        {
          "@type": "Dataset",
          "name": "Sub dataset 02",
```

Exercise: Google's "Try the structured data codelab"

DISTRIBUTED DATA ARCHITECTURE



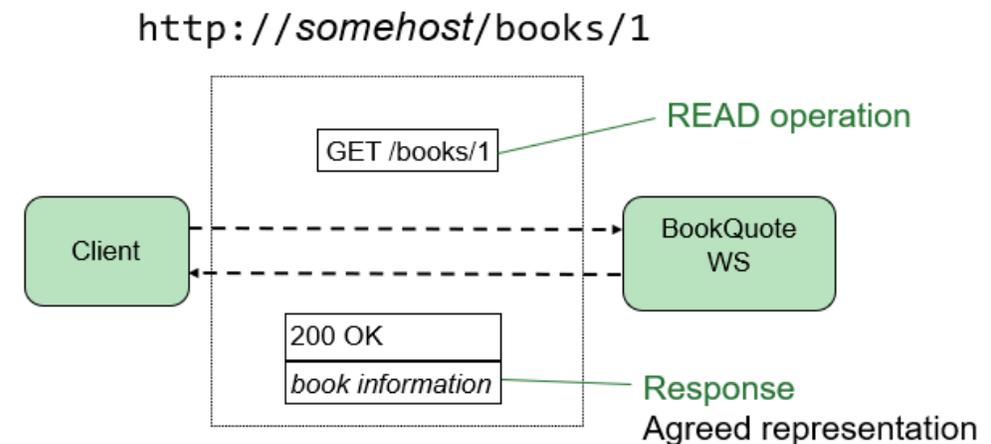
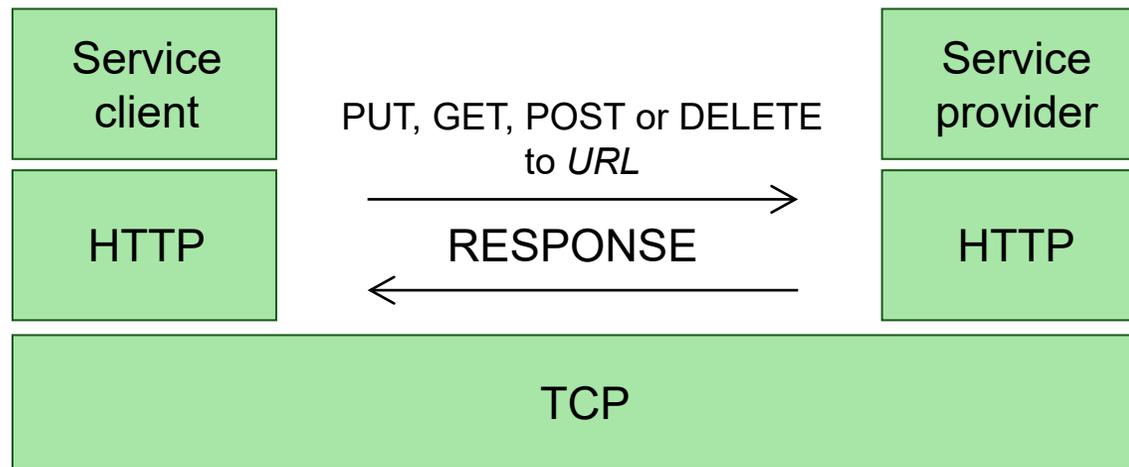
- **The Semantic Web (SW):** extension of the *World Wide Web* to make Internet data **machine-readable** in a way that can be consumed and **understood by machines**

→ How to enable this distributed data approach?

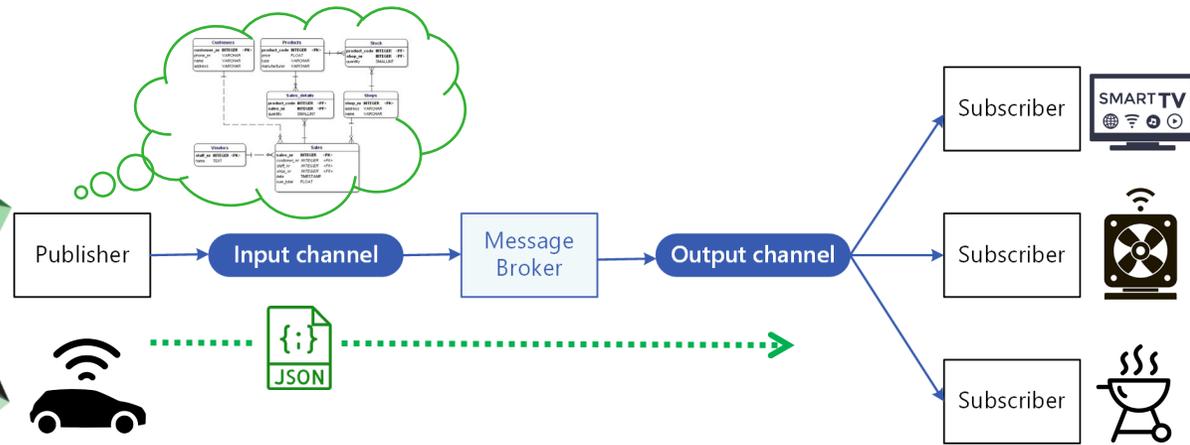
Web applications: Service-Oriented Architecture!

RESTFUL WEB SERVICES ARCHITECTURE

- No additional application protocol
- HTTP is used to convey application information



PUBLISH-SUBSCRIBE PATTERN



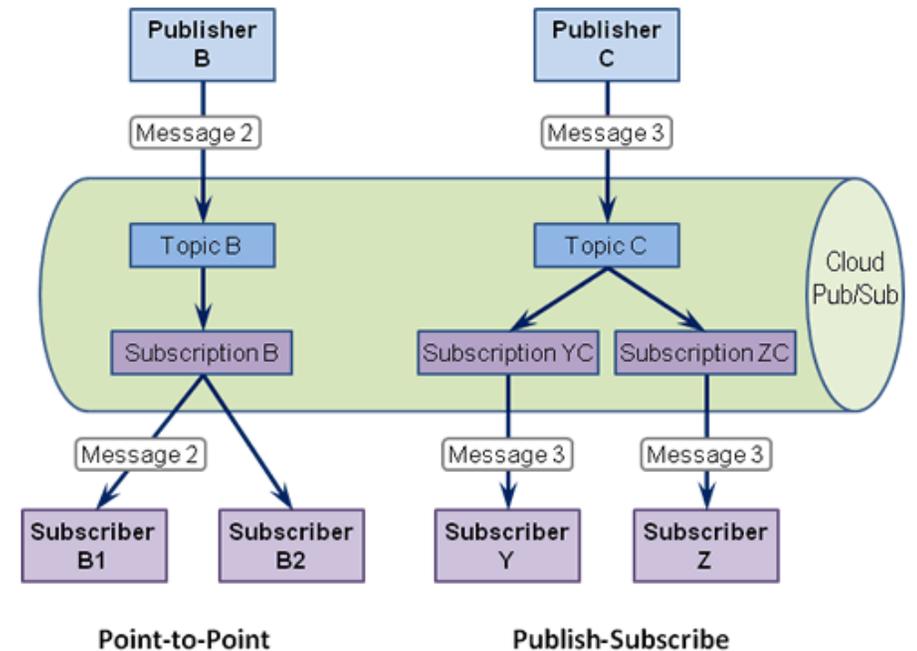
Send the **event** on a pub-sub channel (**message broker**), which delivers a copy of this event **to each receiver**



1. Input channel splits into multiple (each subscriber)
2. Broker delivers a copy of the message to output
3. Subscriber consumes message

Topic-based: **topic** is a named logical channel **defined by the publisher** (e.g., room-temperature)

Content-based: messages only delivered if the message matches constraints (**defined by subscriber**)



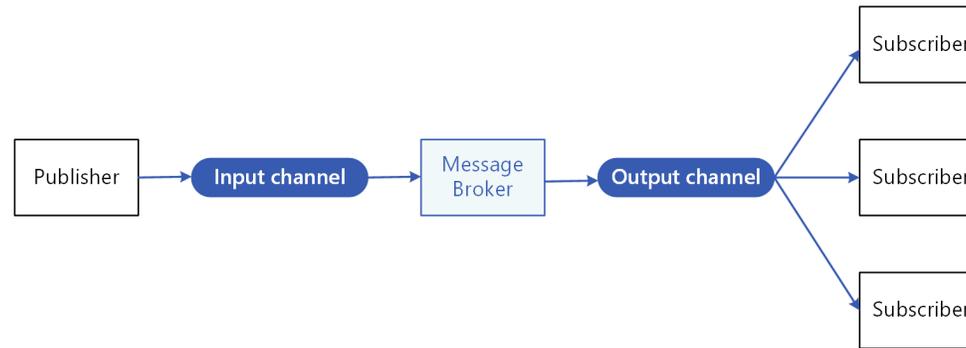
Point-to-Point

Publish-Subscribe

MICROSERVICES ARCHITECTURE

You Are Not Netflix (2020). Jakša Vučković

*“Updates that need to be performed as a consequence of the **user-triggered operation** (or some other **external event**) but are not required for keeping any form of **strong consistency** can be invoked using some **asynchronous communication** mechanism. Using a **publish-subscribe mechanism** decouples the downstream **microservices**”*



*“A **good practice** is to publish events using the **data model of the producer** including **all possible data about the event** and let each consumer transform the data according to its needs”*

- Why this is a “good practice”?
- What are possible consequences (problems)?

DISTRIBUTED DATA ARCHITECTURE

Verbosity problem: message size (payload) impacts the data exchange performance and cloud infrastructure costs

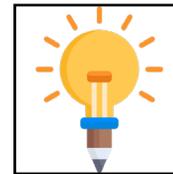
One possible solution: send only the parts that are most important for the consumer (like in GraphQL)?

➤ But publishers and subscribers are agnostic: publishers “don’t know” what subscribers need... 🤔

What if the producer publishes only the most relevant data about the event, i.e., dynamic data that changed with the event; and enables the consumer(s) to: (a) “understand” the producer’s data model and (b) a way to retrieve the static data?

A distributed data approach!

- Dynamic data: published in pub/sub broker
- Static data: available via RESTful services
- JSON-LD facilitates data distribution and semantic interoperability

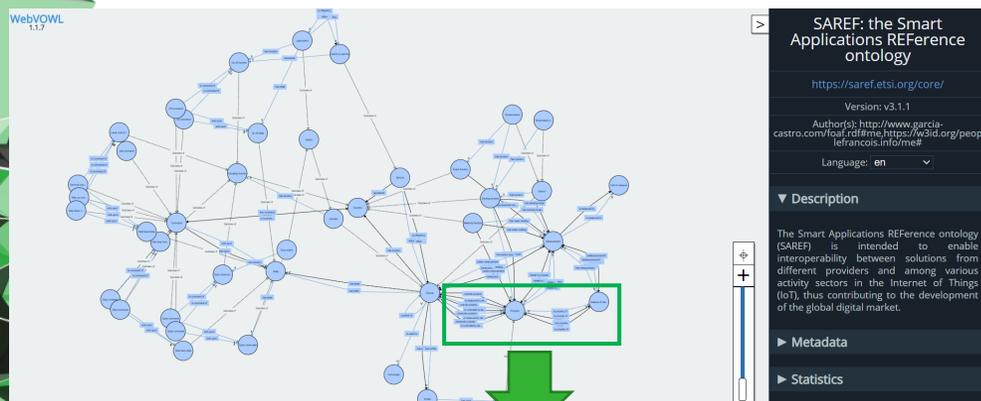


DISTRIBUTED DATA ARCHITECTURE

Watch video (2min): <https://youtu.be/pXSJmZcC2J8>

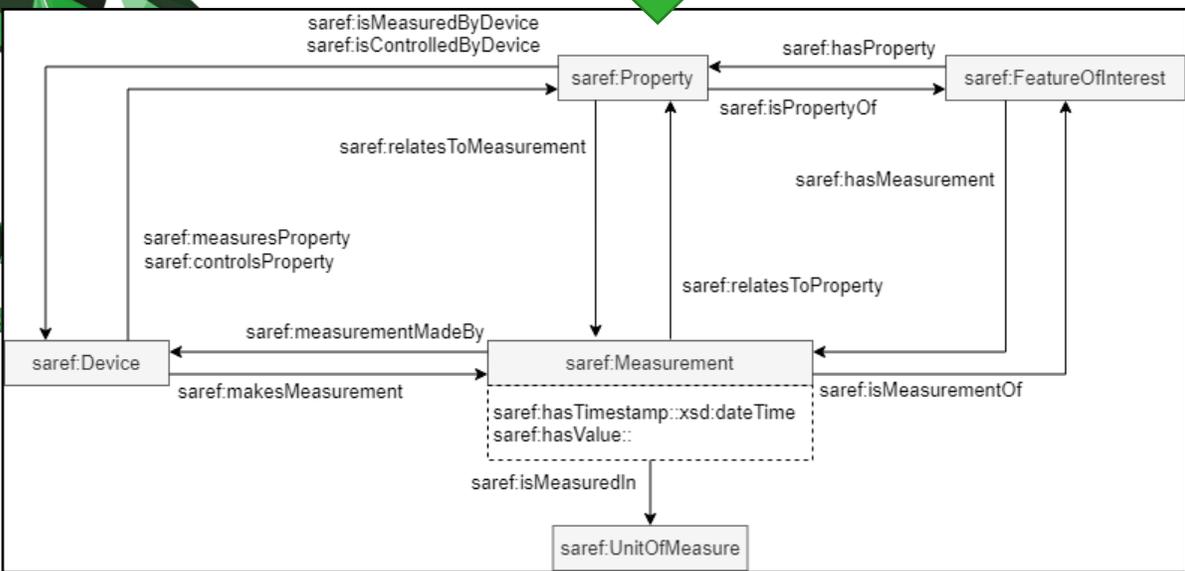
Semantic distributed data approach for the verbosity problem

- Reuse existing (standardized) data models (“ontologies”), e.g., ETSI SAREF (extensions)



```

"@context": {
  "saref": "https://saref.etsi.org/core/",
  "geo": "http://www.w3.org/2003/01/geo/wgs84_pos"
  "instance": "https://myapp-uri/"
}
"@id": "instance:MotoG5Plus_MobileDeviceId",
"@type": "saref:Device",
"geo:location": {
  "@id": "instance:DeviceCurrentLocation_MobileDeviceId_1536663581",
  "@type": "geo:Point",
  "geo:lat": 52.24145074,
  "geo:long": 6.85584719
},
"saref:consistsOf": [
  {
    "@id": "instance:MotoG5Plus_MobileDeviceId_AccelerometerSensorId",
    "@type": "saref:Sensor",
    "saref:makesMeasurement": {
      "@id": "instance:Acceleration_Average_AxisX_1.79769313486232E+308",
      "@type": "saref:Measurement",
      "saref:hasTimestamp": "2018-09-11T10:59:38.3963960Z",
      "saref:hasValue": 1.3323891985416412,
      "saref:isMeasuredIn": {
        "@id": "saref:SpeedUnit_MeterPerSecond",
        "@type": "saref:UnitOfMeasure"
      },
      "saref:relatesToProperty": {
        "@id": "instance:Acceleration_Average_AxisX",
        "@type": "saref:Property"
      }
    }
  }
]
  
```



RDF DATA GENERATION

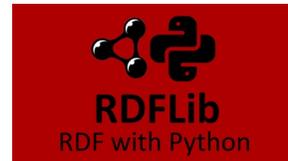
So, how to generate data in an RDF syntax format?

1. “From sketch”:

a. Manually with an ontology editor (e.g., Protégé)



b. Automatic with SW application developed with RDF libs: data is generated in RDF from the beginning!



2. Reusing existing data: **Triplification process**

a. Can be also manually (e.g., OpenRefine)

b. Can be “automatic” with RDF libs

SW APP WITH RDF LIBRARIES

Example in C# (similar to Jena)

```
public JObject GetECGDeviceJSON_SAREF4health(List<JObject> listDevicesOfDevice, JObject recordingECGSession)
{
    string deviceId = "sarefInst:Shimmer3ECG_unit_T9JRN42_DeviceId";
    JObject eCGDeviceJSON = JObject.FromObject(new
    {
        comment = "Shimmer3 ECG unit (T9J-RN42): INTER-IoT-EWS project",
        label = "Shimmer3 ECG unit T9J-RN42",
        seeAlso = "http://www.shimmersensing.com/products/ecg-development-kit#specifications-tab"
    });
    eCGDeviceJSON.Add("@id", deviceId);
    eCGDeviceJSON.Add("@type", JSONArray.Parse("[ 'saref4health:ECGDevice', 'saref:Device' ]"));
    eCGDeviceJSON.Add("saref4envi:hasFrequencyMeasurement", hasFrequencyMeasurement);
    eCGDeviceJSON.Add("saref:accomplishes", recordingECGSession);
    eCGDeviceJSON.Add("saref:hasManufacturer", "Shimmer");
    eCGDeviceJSON.Add("saref:hasTypicalConsumption", hasTypicalConsumption);
    eCGDeviceJSON.Add("saref:consistsOf", JToken.FromObject(listDevicesOfDevice));

    return eCGDeviceJSON;
}
```

JAVA: APACHE JENA FRAMEWORK





Apache Jena

A free and open source Java framework for building [Semantic Web](#) and [Linked Data](#) applications.

[Get started now!](#)
[Download](#)

RDF

RDF API

Interact with the core API to create and read [Resource Description Framework \(RDF\)](#) graphs. Serialise your triples using popular formats such as [RDF/XML](#) or [Turtle](#).

ARQ (SPARQL)

Query your RDF data using [ARQ](#), a [SPARQL 1.1](#) compliant engine. [ARQ](#) supports remote federated queries and free text search.

Triple store

TDB

Persist your data using [TDB](#), a native high performance triple store. [TDB](#) supports the full range of Jena APIs.

Fuseki

Expose your triples as a [SPARQL](#) end-point accessible over [HTTP](#). [Fuseki](#) provides REST-style interaction with your RDF data.

OWL

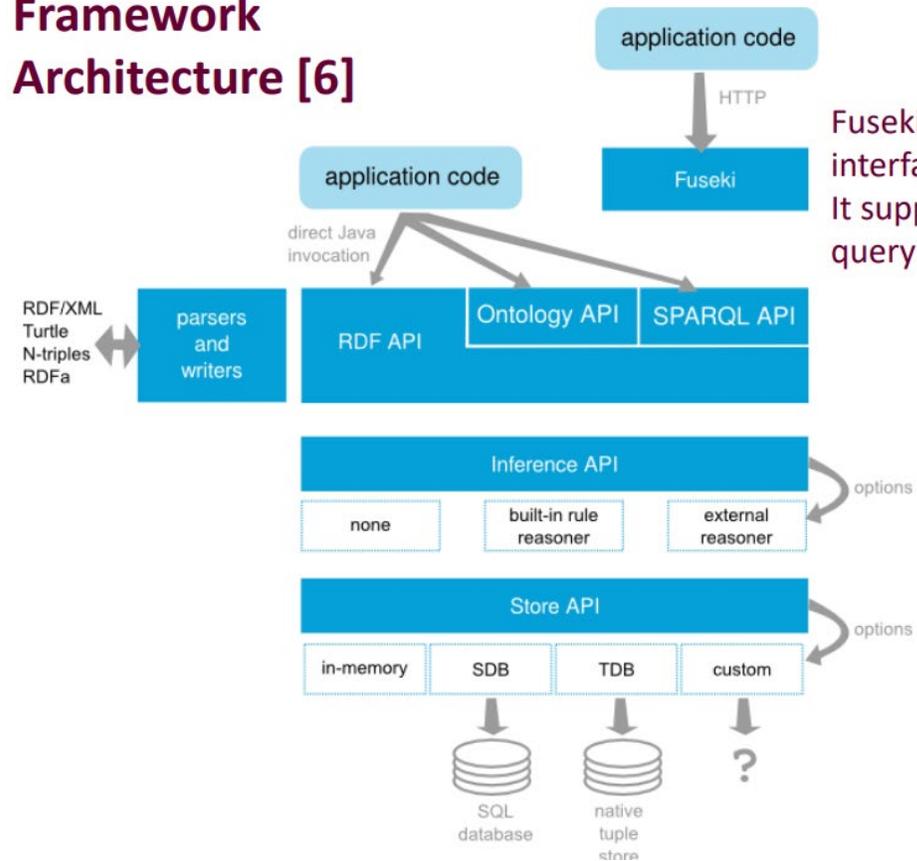
Ontology API

Work with models, [RDFS](#) and the [Web Ontology Language \(OWL\)](#) to add extra semantics to your RDF data.

Inference API

Reason over your data to expand and check the content of your triple store. Configure your own inference rules or use the built-in [OWL](#) and [RDFS](#) reasoners.

Framework Architecture [6]



Fuseki is an HTTP interface to RDF data. It supports SPARQL for querying and updating.

PYTHON: RDFLIB

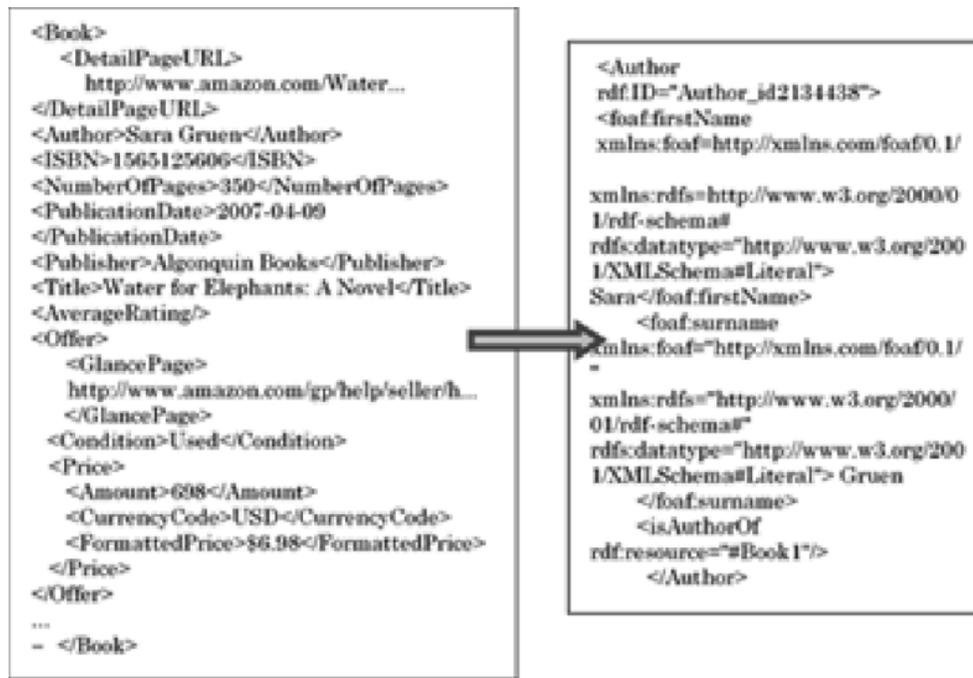
<https://github.com/jonimoreira/B2SHARE-FAIR>

```
265 # B2SHARE: Community
266 # Level 2: Catalog metadata layer
267 def translate_catalog(community):
268     context = {
269         # ontologies used in FDP according to spec
270         "rdf" : "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
271         "rdfs" : "http://www.w3.org/2000/01/rdf-schema#",
272         "dcat" : "http://www.w3.org/ns/dcat#",
273         "xsd" : "http://www.w3.org/2001/XMLSchema#",
274         "owl" : "http://www.w3.org/2002/07/owl#",
275         "dct" : "http://purl.org/dc/terms/",
276         "lang" : "http://id.loc.gov/vocabulary/iso639-1/",
277         "fdp" : "http://rdf.biosemantics.org/ontologies/fdp-o#",
278         "foaf" : "http://xmlns.com/foaf/",
279         # B2SHARE otology (internal terms)
280         "b2" : "https://b2share.eudat.eu/ontology/b2share/",
281         # Other ontologies (reused)
282         "pro" : "http://purl.org/spar/pro/" }
283
284     doc = {
285         "@type": "dcat:Catalog",
286         "http://purl.org/dc/terms/identifier": community.identifier,
287         "http://purl.org/dc/terms/title": community.name,
288         "http://purl.org/dc/terms/description": community.description,
289         "http://purl.org/dc/terms/issued": community.created,
290         "http://purl.org/dc/terms/modified": community.updated,
291         "http://xmlns.com/foaf/logo" : community.logo,
292         "https://b2share.eudat.eu/ontology/b2share/publication_workflow" : community.publication_workflow,
293         "https://b2share.eudat.eu/ontology/b2share/restricted_submission" : community.restricted_submission,
294         "@id": community.links.selflink
295     }
296     compacted = jsonld.compact(doc, context)
297     #print(compacted)
298     return compacted
299
```

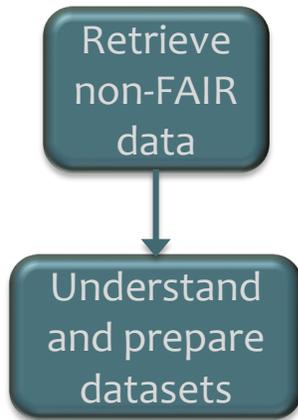
TRIPLIFICATION

Process of transforming non-RDF data, e.g., in CSV, XML, relational databases; to RDF data

Example:



FAIRIFICATION WORKFLOW



Standard format (XML, RDF, relational DB API, VCF, DICOM, etc.)?
What is the content?
Column/field names?
Relations?
Data domain and range?
Understand the data
Data munging

<https://github.com/FAIRDataTeam/FAIRifier>

FAIRIFIER A power tool for working with messy data.

Create Project « Start Over Configure Parsing Options

Open Project

Import Project

Language Settings

	Column 1	Column 2	Column 3
1.	1394	2778	literature
2.	100290337	4214	literature
3.	122704	54460	complexes
4.	4790	79155	binary;literature;signaling
5.	2597	70	signaling
6.	5923	7157	literature
7.	509	6122	literature
8.	4067	933	literature;signaling
9.	398	998	literature
10.	1748	5976	literature
11.	1537	55967	metabolic

Parse data as

CSV / TSV / separator-based files

Line-based text files

Fixed-width field text files

PC-Axis text files

JSON files

MARC files

RDF/N3 files

XML files

Open Document Format spreadsheets (.ods)

Character encoding

Columns are separated by

commas (CSV)

tabs (TSV)

custom \t

Escape special characters with \

FAIRIFIER human interactome barabasi Permalink

Facet / Filter Undo / Redo 1

141296 rows

Show as: rows records Show: 5 10 25 50 rows

All	gene_id_1	Column 2	Column 3
1.	1394		
2.	100290337		
3.	122704		
4.	4790		
5.	2597		
6.	5923		
7.	509		
8.	4067		
9.	398		
10.	1748		

Using facets and filters

Use facets and filters to select subsets of your data to act on. Choose facet and filter methods from the menus at the top of each data column.

Not sure how to get started? [Watch these screencasts](#)

- Facet
- Text filter
- Edit cells
- Edit column
- Transpose
- Sort...
- View
- Reconcile
- Split into several columns...
- Add column based on this column...
- Add column by fetching URLs...
- Rename this column
- Remove this column
- Move column to beginning
- Move column to end
- Move column left
- Move column right

Running example: dataset with protein-protein interactions in human interactome (Human-interactome-barabasi.tab)

Refine OPEN

Custom text transform on column gene_id_1

Expression Language Python / Jython

`return "http://identifiers.org/ncbigene/" + str(value)` No syntax error.

Preview History Starred Help

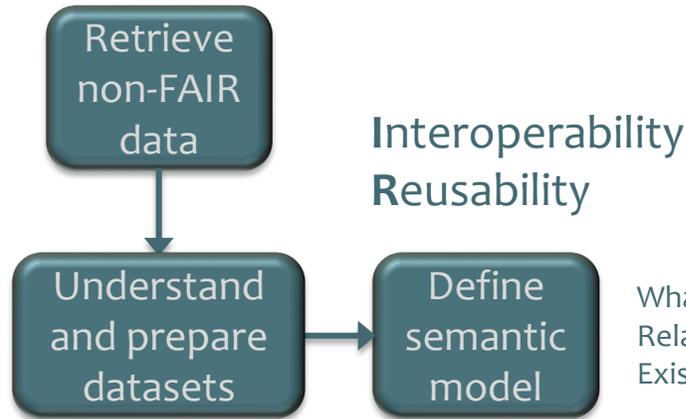
row	value	return "http://identifiers.org/ncbigene/" + str(value)
1.	1394	http://identifiers.org/ncbigene/1394
2.	100290337	http://identifiers.org/ncbigene/100290337
3.	122704	http://identifiers.org/ncbigene/122704
4.	4790	http://identifiers.org/ncbigene/4790
5.	2597	http://identifiers.org/ncbigene/2597
6.	5923	http://identifiers.org/ncbigene/5923
7.	509	http://identifiers.org/ncbigene/509
8.	4067	http://identifiers.org/ncbigene/4067
9.	398	http://identifiers.org/ncbigene/398
10.	1748	http://identifiers.org/ncbigene/1748

On error keep original set to blank store error

Re-transform up to 10 times until no change

OK Cancel

FAIRIFICATION WORKFLOW



What are the concepts involved?
Relations among concepts?
Existing vocabularies for the concepts and instances?

RDF Schema Alignment

The Semantic model below specifies how the RDF data that will get generated from your grid-shaped data. The cells in each record of your data will get placed into nodes within the model. Configure the model by specifying which column to substitute into which node.

Base URI: <http://127.0.0.1:3333/> [edit](#)

Semantic model [RDF Preview](#)

Available Prefixes: [rdf](#) [owl](#) [rdfs](#) [foaf](#) [+ add prefix](#) [manage prefixes](#)

(row index) URI [add property](#)
[add rdf:type](#) Search for class:

Select an item from the list:

- No suggested matches. (Shift + Enter) to add it

Your item not in the list?
Add it (Shift+Enter)

[Add another root node](#) [Load](#) [Share](#) [Save](#)

[OK](#) [Cancel](#)

RDF Schema Alignment

The Semantic model below specifies how the RDF data that will get generated from your grid-shaped data. The cells in each record of your data will get placed into nodes within the model. Configure the model by specifying which column to substitute into which node.

Base URI: <http://127.0.0.1:3333/> [edit](#)

Semantic model [RDF](#)

Available Prefixes: [rdf](#) [owl](#) [rdfs](#) [foaf](#) [+ add prefix](#) [manage prefixes](#)

(row index) URI [add property](#)
[add rdf:type](#) [http://edamontology.org/](#)

RDF Node

Use content from cell...

- (row index)
- gene_id_1
- gene_id_2
- source
- Constant Value

The cell's content is used ...

- as a URI
- as text
- as language-tagged text
- as integer number
- as non-integer number
- as date (YYYY-MM-DD)
- as dateTime (YYYY-MM-DD HH:MM:SS)
- as boolean
- as custom datatype (specify type URI)
- as a blank node

Use custom expression...

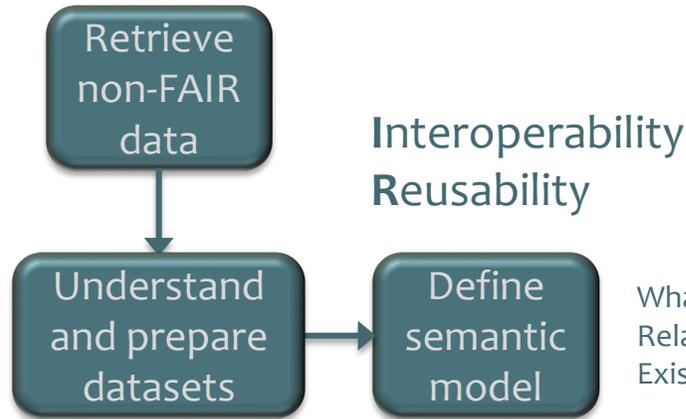
value
[preview/edit](#)

[OK](#) [Cancel](#)

[Add another root node](#) [Load](#) [Share](#) [Save](#)

[OK](#) [Cancel](#)

FAIRIFICATION WORKFLOW



What are the concepts involved?
Relations among concepts?
Existing vocabularies for the concepts and instances?

RDF Schema Alignment

The Semantic model below specifies how the RDF data that will get generated from your grid-shaped data. The cells in each record of your data will get placed into nodes within the model. Configure the model by specifying which column to substitute into which node.

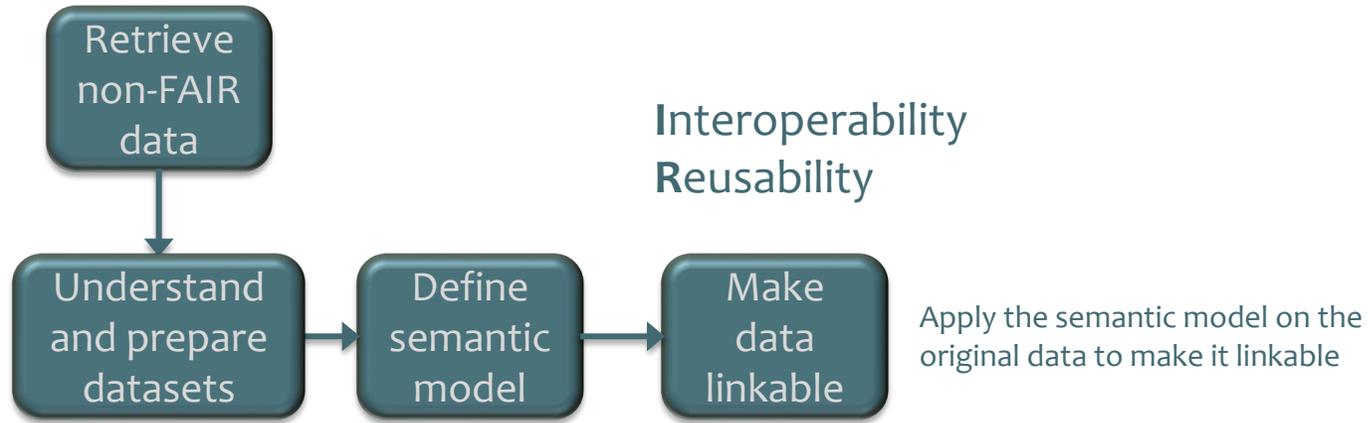
Base URI: <http://127.0.0.1:3333/> [edit](#)

Semantic model [RDF Preview](#)

Available Prefixes: rdf owl rdfs foaf [+ add prefix](#) [+ manage prefixes](#)

(row index) URI [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#) [⊜](#) [⊝](#) [⊞](#) [⊟](#) [⊠](#) [⊡](#) [⊢](#) [⊣](#) [⊤](#) [⊥](#) [⊦](#) [⊧](#) [⊨](#) [⊩](#) [⊪](#) [⊫](#) [⊬](#) [⊭](#) [⊮](#) [⊯](#) [⊰](#) [⊱](#) [⊲](#) [⊳](#) [⊴](#) [⊵](#) [⊶](#) [⊷](#) [⊸](#) [⊹](#) [⊺](#) [⊻](#) [⊼](#) [⊽](#) [⊾](#) [⊿](#) [⊘](#) [⊙](#) [⊚](#) [⊛](#)

FAIRIFICATION WORKFLOW



FAIRIFIER human interactome barabasi [Permalink](#)

Facet / Filter Undo / Redo 9

141296 rows

Show as: rows records Show: 5 10 25 50 rows

All	gene_id_1	gene_id_2	source
1.	http://identifiers.org/ncbigene/1394	http://identifiers.org/ncbigene/2778	literature
2.	http://identifiers.org/ncbigene/100290337	http://identifiers.org/ncbigene/4214	literature
3.	http://identifiers.org/ncbigene/122704	http://identifiers.org/ncbigene/54460	complexes
4.	http://identifiers.org/ncbigene/4790	http://identifiers.org/ncbigene/79155	binary;literature;signaling
5.	http://identifiers.org/ncbigene/2597	http://identifiers.org/ncbigene/70	signaling
6.	http://identifiers.org/ncbigene/5923	http://identifiers.org/ncbigene/7157	literature
7.	http://identifiers.org/ncbigene/509	http://identifiers.org/ncbigene/6122	literature
8.	http://identifiers.org/ncbigene/4067	http://identifiers.org/ncbigene/933	literature;signaling
9.	http://identifiers.org/ncbigene/398	http://identifiers.org/ncbigene/998	literature
10.	http://identifiers.org/ncbigene/1748	http://identifiers.org/ncbigene/5976	literature

Export project

- Tab-separated value
- Comma-separated value
- HTML table
- Excel (.xls)
- Excel 2007+ (.xlsx)
- ODF spreadsheet
- Triple loader
- MLWrite
- Custom tabular exporter...
- Templating...
- RDF as RDF/XML**
- RDF as Turtle**
- POST to FAIR Data Point

Using facets and filters

Use facets and filters to select subsets of your data to act on. Choose facet and filter methods from the menus at the top of each data column.

Not sure how to get started?
[Watch these screencasts](#)