DBpedia

DBpedia Tutorial @ Data Week 2024

Milan Dojchinovski, Sebastian Hellmann, Jan Forberg, Jonathan Justavino Lüderitz, Johannes Frey, Kirill Yankov and Julia Holze

dbpedia.org

InfAI®
Institut für Angewandte Informatik

Meet the Organizers

All members of the DBpedia core team hosted by:

Institute of Applied Informatics / DBpedia Association, Leipzig, DE

Milan Dojchinovski

Sebastian Hellmann

Jan Forberg

Johannes Frey

Julia Holze

Kirill Yankov

Jonathan Lüderitz

About the tutorial

- Get familiar with DBpedia
  - history of DBpedia, community, DBpedia KG release process
  - how a DBpedia triple is born
  - ontology, endpoints

- Learn about the DBpedia technology stack
  - DBpedia Databus
  - DBpedia Spotlight
  - DBpedia Lookup

- Learn best practices via several practical showcases
  - Semantic Text Annotation and Search using Databus and DBpedia Spotlight
  - CI and Databus publishing using Jenkins
  - Databus Metadata Overlay Search System
  - Terminology Server using DBpedia Lookup

Agenda

- 11:00 - 11:05 **Opening**, by the tutorial organizers
- 11:05 - 12:30 **Session 1: DBpedia Tech Overview and DBpedia Databus**

DBpedia overview, Databus Use Cases, DBpedia Databus, Databus collections, deploying own DBpedia KG

Lunch break (60 mins)

- 13:30 - 15:00 **Session 2: DBpedia and Databus Showcases**

Semantic Indexing and Search using DBpedia Spotlight and Databus, CI and Databus publishing using Jenkins

Coffee break (30 mins)

- 15:30 - 16:50 **Session 3: Bpedia and Databus Showcases (cont.)**

Databus Metadata Overlay Search System (MOSS), Terminology Server & Archivo

- 16:50 - 17:00 **Closing session**

* all times are in CEST time zone

Guidelines

- Feel free to engage/ask questions:
  - after each session
- The slides are made public
  - see the footer placeholder: http://tinyurl.com/DBpediaDataWeek2024

DBpedia

DBpedia Overview

by Milan Dojchinovski

DBpedia Mission

2007 - A crowd-sourced community effort to **extract structured information from Wikipedia** and make this information available on the Web.

- benefit: query Wikipedia as a DB

2024 - Current mission: Global and Unified Access to Knowledge Graphs

- Original definition still holds true, moreover ...
- **Global DBpedia -> data beyond Wikipedia**
    1. offer **links** to other sources
    2. **platform** (i.e. databus) to integrate your data with all other data

DBpedia Milestones

DBpedia

2007
formation of the Linked Data Cloud

2010
Open editing of DBpedia Ontology
A new type of Cyc?

2012-2016
Covering all 140 Wikipedias, Commons, Wikidata
14.4 B facts extracted

2018

2020 - 22 B facts per month

Huge Linked Data - derived Open Knowledge Graphs (OKG)

2007
first Wikipedia extraction, SPARQL Linked Data

OPENLINK SOFTWARE Making Technology Work For You®
UNIVERSITY OF MANNHEIM
UNIVERSITAT LEIPZIG
InfAI Institut für Angewandte Informatik

2009
Major boost in KG and Linking Research

2011
Industry adoption

IBM Watson
YAHOO!
BBC
UNICODE
SIEMENS

2014
Foundation of DBpedia Association Leipzig

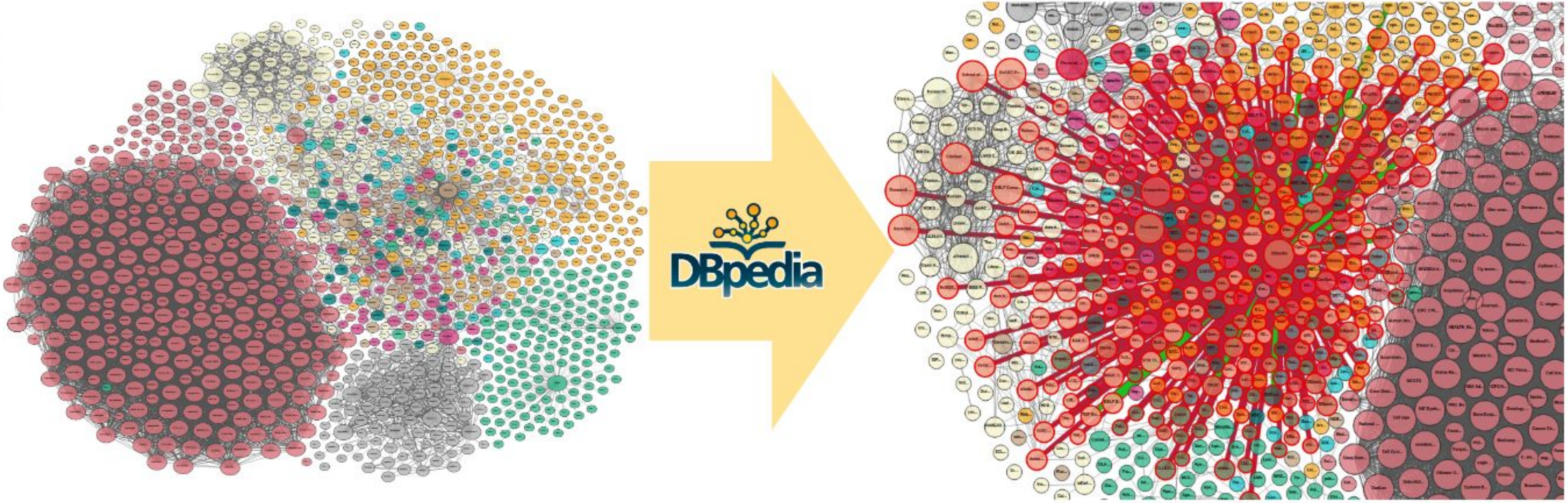
2017
SHACL W3C Standard by Uni Leipzig
Test-driven KG development

2019
DBpedia Innovation Platform - Central hub for Linked Data Technology and Ecosystem

2020 - FAIR Linked Data

Findable
Accessible
Interoperable
Reusable

DBpedia is not only connecting & publishing data

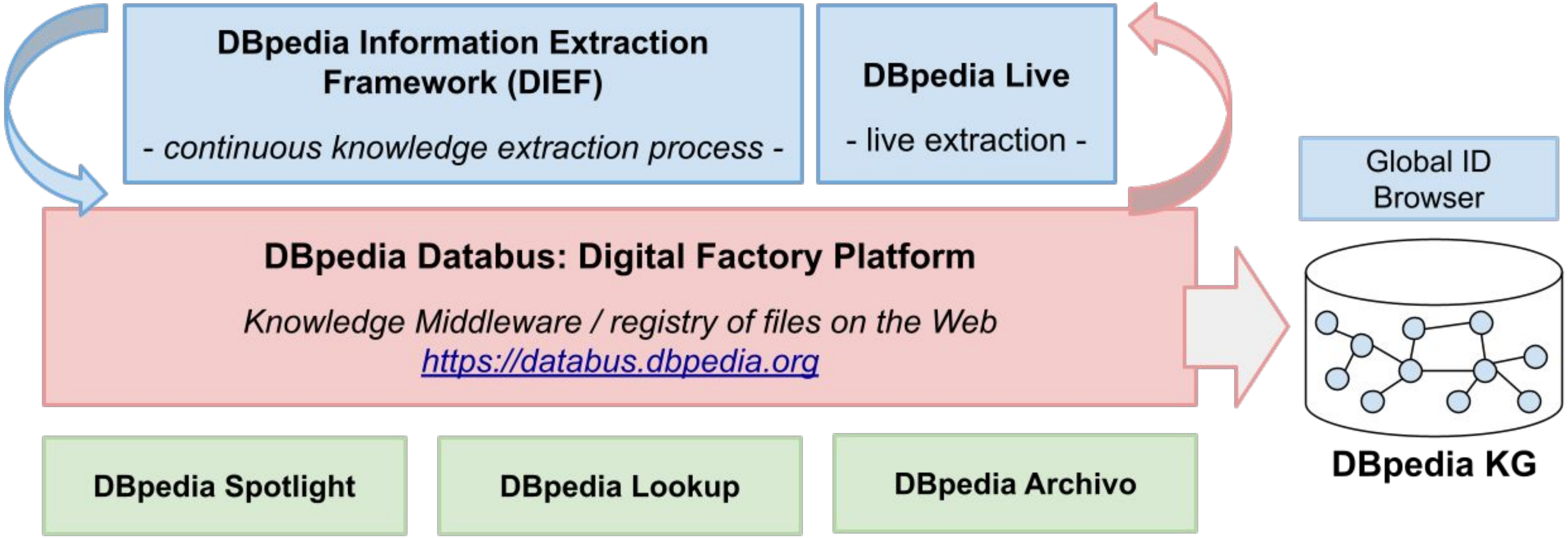


… but also people and orgs

Organizational Structure in Numbers

- Around 20 DBpedia Chapters
  - **language chapters**, English, German, Dutch, Czech, Polish, Hungarian, ...
  - **regional chapters**, e.g. for cities or individual countries
  - **domain chapters**, e.g. for law, medicine, media and science
  - each chapter hosts and maintains localized DBpedia version
  - more about DBpedia chapters at https://www.dbpedia.org/members/chapter-overview/
- 30+ DBpedia members
  - 41% industry and start-up, 37% non-profit, 22% tiny & self-employed
  - join the network of pioneers to shape the future of knowledge graphs
  - apply via https://www.dbpedia.org/members/membership/

The DBpedia Tech Ecosystem



DBpedia Information Extraction Framework (DIEF)

- continuous knowledge extraction process -

DBpedia Live

- live extraction -

Global ID Browser

DBpedia Databus: Digital Factory Platform

Knowledge Middleware / registry of files on the Web
https://databus.dbpedia.org

DBpedia KG

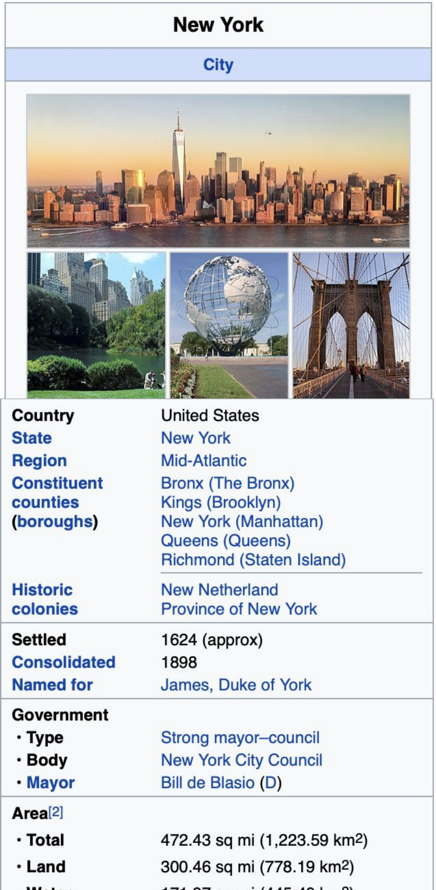
DBpedia Spotlight

DBpedia Lookup

DBpedia Archivo

How a DBpedia triple is born?

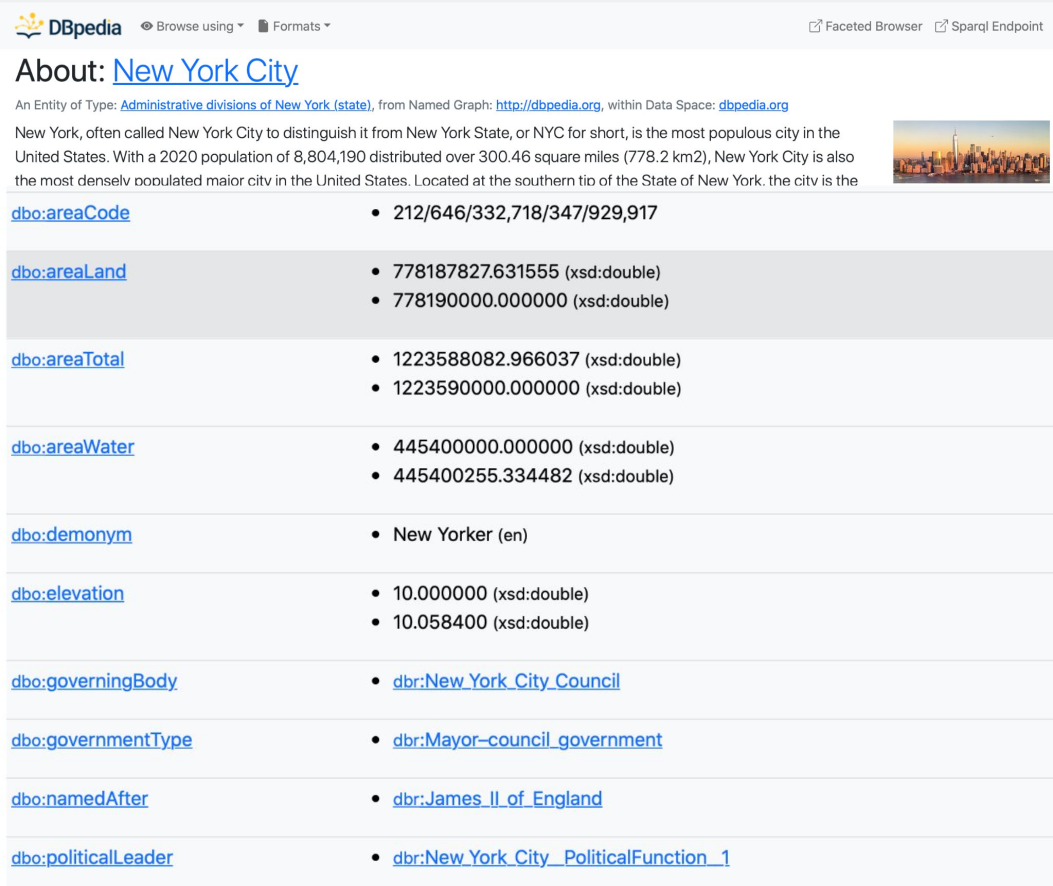
New York City in DBpedia



New York

City

Country	United States
State	New York
Region	Mid-Atlantic
Constituent counties (boroughs)	Bronx (The Bronx) Kings (Brooklyn) New York (Manhattan) Queens (Queens) Richmond (Staten Island)
Historic colonies	New Netherland Province of New York
Settled	1624 (approx)
Consolidated	1898
Named for	James, Duke of York
Government	
· **Type**	Strong mayor–council
· **Body**	New York City Council
· **Mayor**	Bill de Blasio (D)
Area[2]	
· **Total**	472.43 sq mi (1,223.59 km^2)
· **Land**	300.46 sq mi (778.19 km^2)
· **Water**	171.07 sq mi (445.40 km^2)



DBpedia   Browse using ▾   Formats ▾   Faceted Browser   Sparql Endpoint

About: New York City

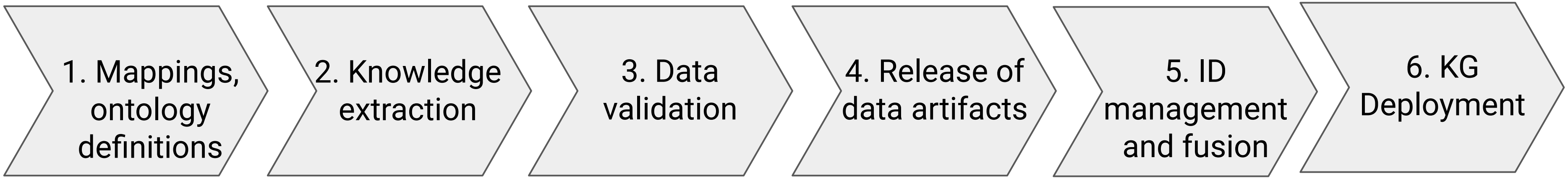
An Entity of Type: Administrative divisions of New York (state), from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

New York, often called New York City to distinguish it from New York State, or NYC for short, is the most populous city in the United States. With a 2020 population of 8,804,190 distributed over 300.46 square miles (778.2 km2), New York City is also the most densely populated major city in the United States. Located at the southern tip of the State of New York, the city is the

dbo:areaCode	• 212/646/332,718/347/929,917
dbo:areaLand	• 778187827.631555 (xsd:double) • 778190000.000000 (xsd:double)
dbo:areaTotal	• 1223588082.966037 (xsd:double) • 1223590000.000000 (xsd:double)
dbo:areaWater	• 445400000.000000 (xsd:double) • 445400255.334482 (xsd:double)
dbo:demonym	• New Yorker (en)
dbo:elevation	• 10.000000 (xsd:double) • 10.058400 (xsd:double)
dbo:governingBody	• dbr:New_York_City_Council
dbo:governmentType	• dbr:Mayor–council_government
dbo:namedAfter	• dbr:James_II_of_England
dbo:politicalLeader	• dbr:New_York_City__PoliticalFunction__1

https://t

Overarching DBpedia KG Release Process



1. Mappings, ontology definitions

2. Knowledge extraction

3. Data validation

4. Release of data artifacts

5. ID management and fusion

6. KG Deployment

1. Definition of mappings and ontology definition
2. Execution of the knowledge extraction process over wikipedia dumps
3. Parsing and validation of the data against strict rules
4. Release of (intermediate) data artifacts
5. ID management and knowledge fusion from all language editions
6. Deployment of the resulting KG

DBpedia Datasets Partitions

Available extractions, 22 billion facts total (500GB without text)

- **Mapping-based** (rule-based)
- **Generic** (automatic)
- **Text**
- **Wikidata**

… bonus:

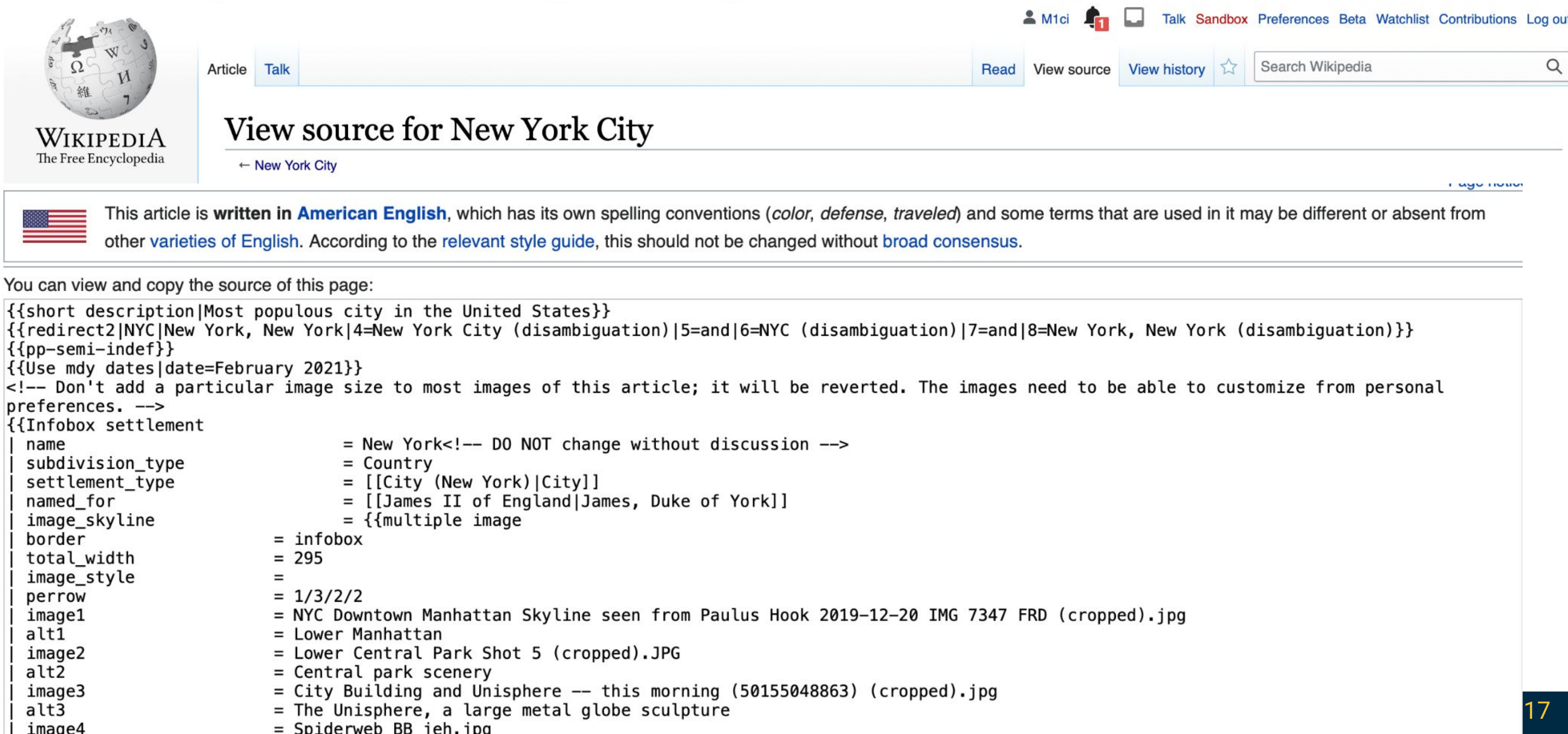
- **Fusion** - fused version of all wikipedia languages
- **Global IDs** - unique URIs across all languages (https://global.dbpedia.org)

… data derived based on the Wikimedia XML dumps

How a DBpedia triple is born

… using **mappings-based extraction?**

Structure of Wikipedia articles



View source for New York City

← New York City

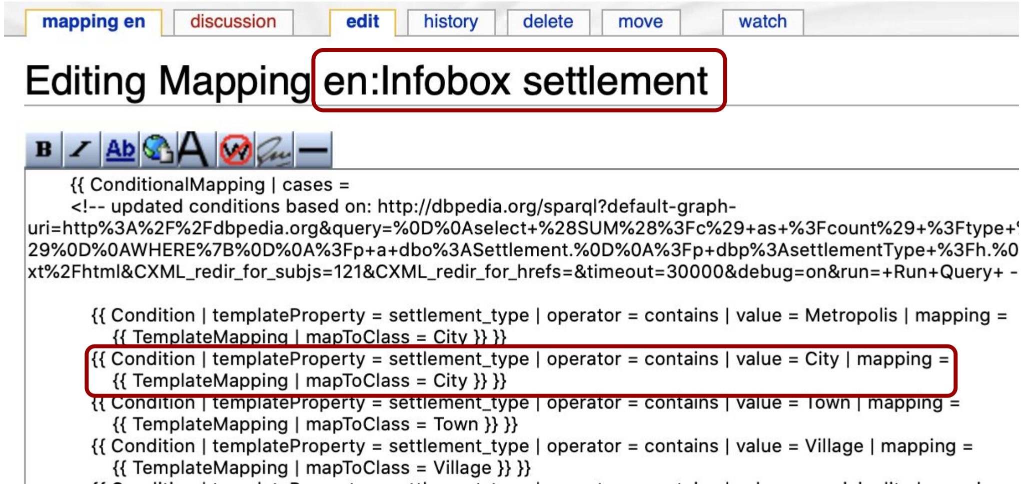
This article is **written in American English**, which has its own spelling conventions (*color*, *defense*, *traveled*) and some terms that are used in it may be different or absent from other varieties of English. According to the relevant style guide, this should not be changed without broad consensus.

You can view and copy the source of this page:

```
{{short description|Most populous city in the United States}}
{{redirect2|NYC|New York, New York|4=New York City (disambiguation)|5=and|6=NYC (disambiguation)|7=and|8=New York, New York (disambiguation)}}
{{pp-semi-indef}}
{{Use mdy dates|date=February 2021}}
<!-- Don't add a particular image size to most images of this article; it will be reverted. The images need to be able to customize from personal
preferences. -->
{{Infobox settlement
| name                             = New York<!-- DO NOT change without discussion -->
| subdivision_type                 = Country
| settlement_type                  = [[City (New York)|City]]
| named_for                        = [[James II of England|James, Duke of York]]
| image_skyline                    = {{multiple image
| border                    = infobox
| total_width               = 295
| image_style               =
| perrow                    = 1/3/2/2
| image1                    = NYC Downtown Manhattan Skyline seen from Paulus Hook 2019-12-20 IMG 7347 FRD (cropped).jpg
| alt1                      = Lower Manhattan
| image2                    = Lower Central Park Shot 5 (cropped).JPG
| alt2                      = Central park scenery
| image3                    = City Building and Unisphere -- this morning (50155048863) (cropped).jpg
| alt3                      = The Unisphere, a large metal globe sculpture
| image4                    = Spiderweb BB jeh.jpg
```

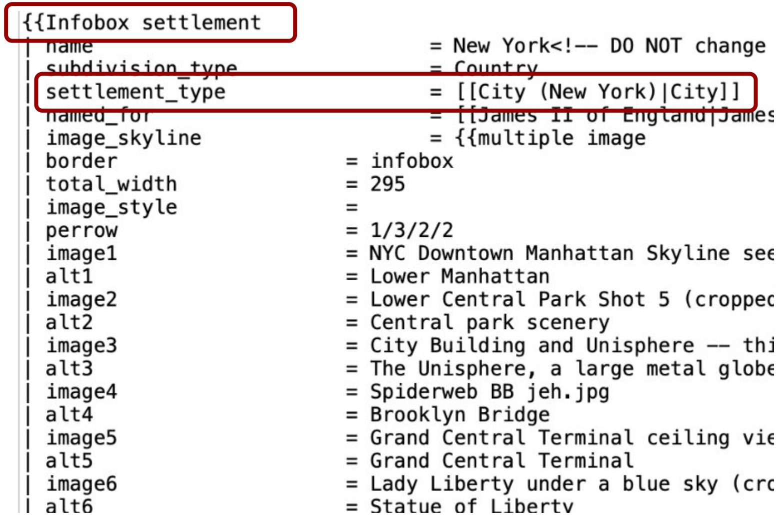
Triple Generation using Mappings

- Mappings maintained on the mappings server: http://mappings.dbpedia.org/
- Mappings for approx. 40 languages, 6 datasets
- http://mappings.dbpedia.org/index.php/Mapping_en:Infobox_settlement



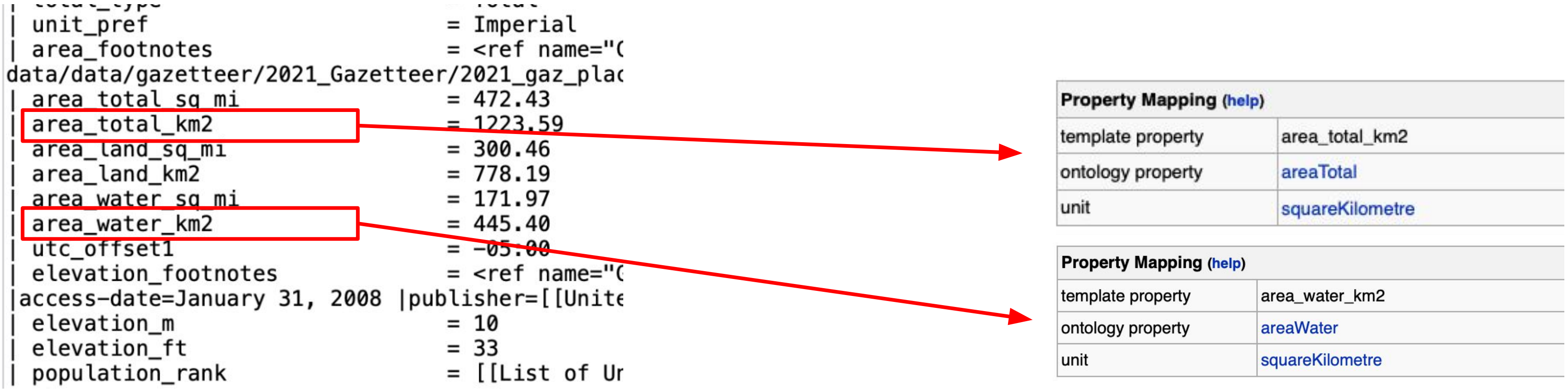
Editing Mapping en:Infobox settlement

```
        {{ ConditionalMapping | cases =
        <!-- updated conditions based on: http://dbpedia.org/sparql?default-graph-
uri=http%3A%2F%2Fdbpedia.org&query=%0D%0Aselect+%28SUM%28%3Fc%29+as+%3Fcount%29+%3Ftype+
29%0D%0AWHERE%7B%0D%0A%3Fp+a+dbo%3ASettlement.%0D%0A%3Fp+dbp%3AsettlementType+%3Fh.%0
xt%2Fhtml&CXML_redir_for_subjs=121&CXML_redir_for_hrefs=&timeout=30000&debug=on&run=+Run+Query+ -

            {{ Condition | templateProperty = settlement_type | operator = contains | value = Metropolis | mapping =
                {{ TemplateMapping | mapToClass = City }} }}
            {{ Condition | templateProperty = settlement_type | operator = contains | value = City | mapping =
                {{ TemplateMapping | mapToClass = City }} }}
            {{ Condition | templateProperty = settlement_type | operator = contains | value = Town | mapping =
                {{ TemplateMapping | mapToClass = Town }} }}
            {{ Condition | templateProperty = settlement_type | operator = contains | value = Village | mapping =
                {{ TemplateMapping | mapToClass = Village }} }}
```



```
{{Infobox settlement
| name                                = New York<!-- DO NOT change
| subdivision_type                    = Country
| settlement_type                     = [[City (New York)|City]]
| named_for                           = [[James II of England|James
| image_skyline                       = {{multiple image
| border                  = infobox
| total_width             = 295
| image_style             =
| perrow                  = 1/3/2/2
| image1                  = NYC Downtown Manhattan Skyline see
| alt1                    = Lower Manhattan
| image2                  = Lower Central Park Shot 5 (croppec
| alt2                    = Central park scenery
| image3                  = City Building and Unisphere -- thi
| alt3                    = The Unisphere, a large metal globe
| image4                  = Spiderweb BB jeh.jpg
| alt4                    = Brooklyn Bridge
| image5                  = Grand Central Terminal ceiling vie
| alt5                    = Grand Central Terminal
| image6                  = Lady Liberty under a blue sky (crc
| alt6                    = Statue of Liberty
```

Mappings Example



```
| total_type                          = Total
| unit_pref                           = Imperial
| area_footnotes                      = <ref name="(
data/data/gazetteer/2021_Gazetteer/2021_gaz_plac
| area_total_sq_mi                    = 472.43
| area_total_km2                      = 1223.59
| area_land_sq_mi                     = 300.46
| area_land_km2                       = 778.19
| area_water_sq_mi                    = 171.97
| area_water_km2                      = 445.40
| utc_offset1                         = -05:00
| elevation_footnotes                 = <ref name="(
|access-date=January 31, 2008 |publisher=[[Unite
| elevation_m                         = 10
| elevation_ft                        = 33
| population_rank                     = [[List of Ur
```

Property Mapping (help)	
template property	area_total_km2
ontology property	areaTotal
unit	squareKilometre

Property Mapping (help)	
template property	area_water_km2
ontology property	areaWater
unit	squareKilometre

{{ PropertyMapping | templateProperty = area_total_km2 | ontologyProperty = areaTotal | unit = squareKilometre }}

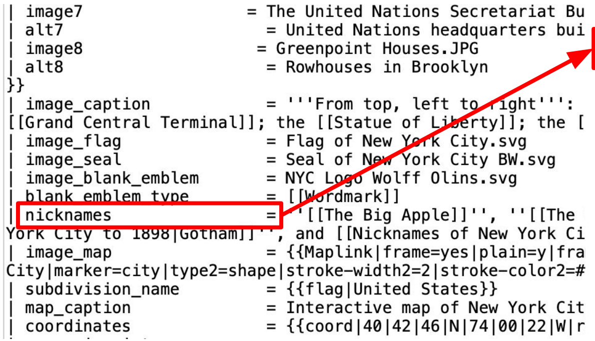
{{ PropertyMapping | templateProperty = area_water_km2 | ontologyProperty = areaWater | unit = squareKilometre }}

```
dbr:New_York_City
    dbo:areaTotal     1223590000.000000 ;
    dbo:areaWater     445400000.000000 .
```

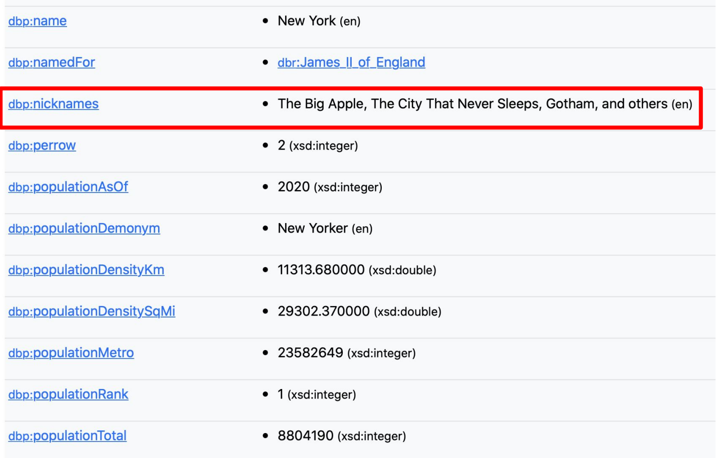
How a DBpedia triple is born

… using **generic extraction?**

Generic Extraction Example



```
| image7                       = The United Nations Secretariat Bu
| alt7                          = United Nations headquarters bui
| image8                       = Greenpoint Houses.JPG
| alt8                          = Rowhouses in Brooklyn
}}
| image_caption                 = '''From top, left to right''':
[[Grand Central Terminal]]; the [[Statue of Liberty]]; the [
| image_flag                    = Flag of New York City.svg
| image_seal                    = Seal of New York City BW.svg
| image_blank_emblem            = NYC Logo Wolff Olins.svg
| blank_emblem_type             = [[Wordmark]]
| nicknames                     = ''[[The Big Apple]]'', ''[[The
York City to 1898|Gotham]]'', and [[Nicknames of New York Ci
| image_map                     = {{Maplink|frame=yes|plain=y|fra
City|marker=city|type2=shape|stroke-width2=2|stroke-color2=#
| subdivision_name              = {{flag|United States}}
| map_caption                   = Interactive map of New York Cit
| coordinates                   = {{coord|40|42|46|N|74|00|22|W|r
```



dbp:name	• New York (en)
dbp:namedFor	• dbr:James_II_of_England
dbp:nicknames	• The Big Apple, The City That Never Sleeps, Gotham, and others (en)
dbp:perrow	• 2 (xsd:integer)
dbp:populationAsOf	• 2020 (xsd:integer)
dbp:populationDemonym	• New Yorker (en)
dbp:populationDensityKm	• 11313.680000 (xsd:double)
dbp:populationDensitySqMi	• 29302.370000 (xsd:double)
dbp:populationMetro	• 23582649 (xsd:integer)
dbp:populationRank	• 1 (xsd:integer)
dbp:populationTotal	• 8804190 (xsd:integer)

Output triples:

dbr:New_York_City
    dbp:nicknames   "The Big Apple, The City That Never Sleeps, Gotham, and others”

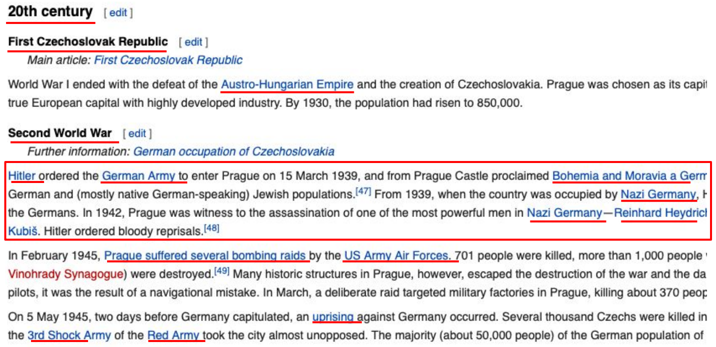
Generic Extraction

- Automatic extraction and export of information
  - Covers 130+ languages and exports 30 different datasets
  - https://databus.dbpedia.org/dbpedia/generic/
- Extraction of:
  - unmapped information in infoboxes
  - other structured information found on the Wikipedia pages

1. Automatic extraction of unmapped properties from infoboxes
   - covers all infobox types along with their attributes
   - http://dbpedia.org/property/ + the name of the infobox attribute
   - e.g. http://dbpedia.org/property/birthplace for the Wikipedia attribute “birthplace”
   - objects are created from the attribute values
2. Automatic extraction of other structured information
   - set of extractors
   - https://github.com/dbpedia/extraction-framework/tree/master/core/src/main/scala/org/dbpedia/extraction/mappings
   - categories, interlanguage links, labels, and many others

Text Extraction

- Wikipedia articles texts
- https://databus.dbpedia.org/dbpedia/text/
- 132 languages, 8 datasets
  - Short and long abstracts
  - content/text + structure
    - sections, sub-sections, paragraphs
    - links
- Information modeled using the NIF Format
- Use cases
  - Training data for text mining
  - Fact extraction



20th century [edit]

First Czechoslovak Republic [edit]

Main article: First Czechoslovak Republic

World War I ended with the defeat of the Austro-Hungarian Empire and the creation of Czechoslovakia. Prague was chosen as its capi
true European capital with highly developed industry. By 1930, the population had risen to 850,000.

Second World War [edit]

Further information: German occupation of Czechoslovakia

Hitler ordered the German Army to enter Prague on 15 March 1939, and from Prague Castle proclaimed Bohemia and Moravia a Gerr
German and (mostly native German-speaking) Jewish populations.[47] From 1939, when the country was occupied by Nazi Germany, H
the Germans. In 1942, Prague was witness to the assassination of one of the most powerful men in Nazi Germany—Reinhard Heydrict
Kubiš. Hitler ordered bloody reprisals.[48]

In February 1945, Prague suffered several bombing raids by the US Army Air Forces. 701 people were killed, more than 1,000 people
Vinohrady Synagogue) were destroyed.[49] Many historic structures in Prague, however, escaped the destruction of the war and the da
pilots, it was the result of a navigational mistake. In March, a deliberate raid targeted military factories in Prague, killing about 370 peop

On 5 May 1945, two days before Germany capitulated, an uprising against Germany occurred. Several thousand Czechs were killed in
the 3rd Shock Army of the Red Army took the city almost unopposed. The majority (about 50,000 people) of the German population of

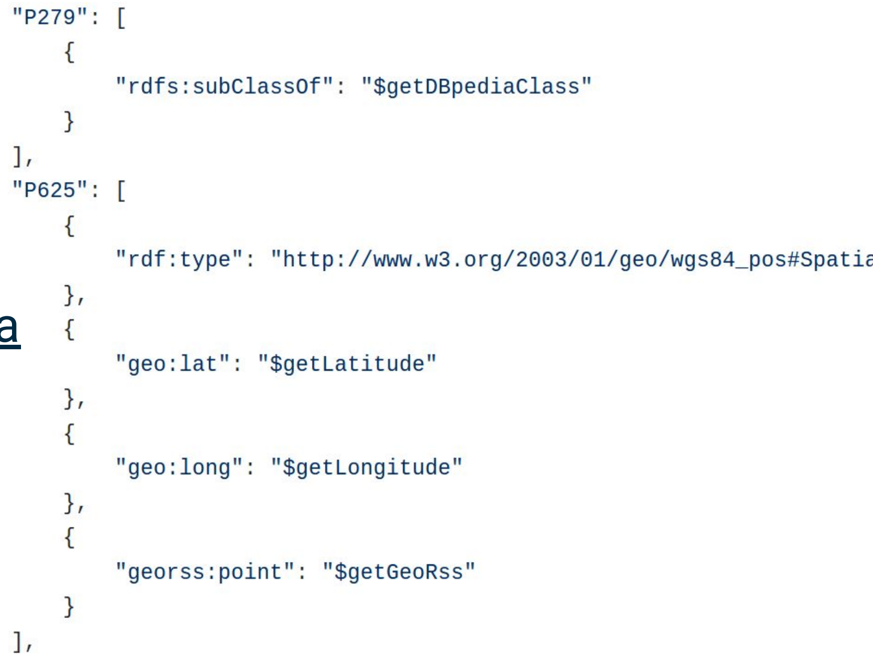
https://en.wikipedia.org/wiki/Prague

Wikidata Extraction

- Same approach as for Wikipedia
- Generic and mappings-based
- Mappings in JSON

https://databus.dbpedia.org/dbpedia/wikidata

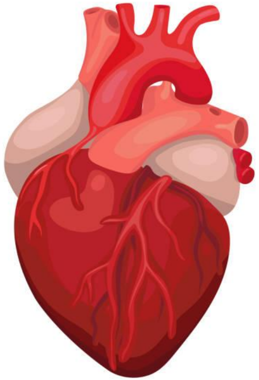
Benefit: Unified access over Wikipedia and Wikidata

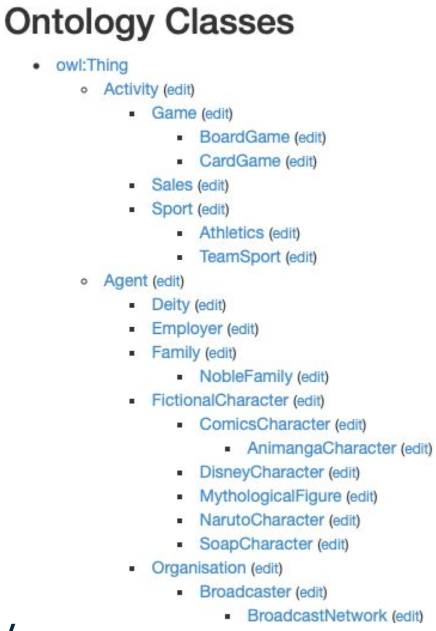


```
"P279": [
    {
        "rdfs:subClassOf": "$getDBpediaClass"
    }
],
"P625": [
    {
        "rdf:type": "http://www.w3.org/2003/01/geo/wgs84_pos#Spatia
    },
    {
        "geo:lat": "$getLatitude"
    },
    {
        "geo:long": "$getLongitude"
    },
    {
        "georss:point": "$getGeoRss"
    }
],
```

DBpedia Ontology

- The heart of DBpedia
- A shallow cross-domain ontology
  - model information extracted from Wikipedia
  - … BUT goes beyond Wikipedia
  - e.g. mappings for the Dutch National KG
- Generated on-the-fly
  - when changes in the mappings wiki are introduced
- Stats
  - over 700 classes and more than 3,000 properties
- Since v3.7: a directed-acyclic graph, not a tree
  - classes may have multiple superclasses
- Get it from the Databus
  - https://databus.dbpedia.org/ontologies/dbpedia.org/ontology--DEV
  - published via DBpedia Archivo





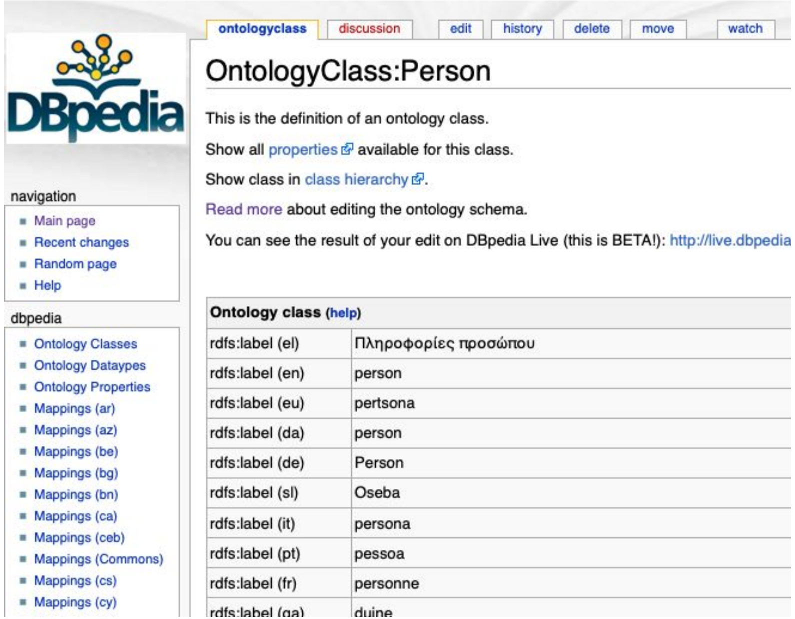
Ontology Classes

- owl:Thing
  - Activity (edit)
    - Game (edit)
      - BoardGame (edit)
      - CardGame (edit)
    - Sales (edit)
    - Sport (edit)
      - Athletics (edit)
      - TeamSport (edit)
  - Agent (edit)
    - Deity (edit)
    - Employer (edit)
    - Family (edit)
      - NobleFamily (edit)
    - FictionalCharacter (edit)
      - ComicsCharacter (edit)
        - AnimangaCharacter (edit)
      - DisneyCharacter (edit)
      - MythologicalFigure (edit)
      - NarutoCharacter (edit)
      - SoapCharacter (edit)
    - Organisation (edit)
      - Broadcaster (edit)
        - BroadcastNetwork (edit)

DBpedia Ontology (cont.)

Edit via the mappings server

http://mappings.dbpedia.org/index.php/OntologyClass:Person



OntologyClass:Person

This is the definition of an ontology class.

Show all properties available for this class.

Show class in class hierarchy.

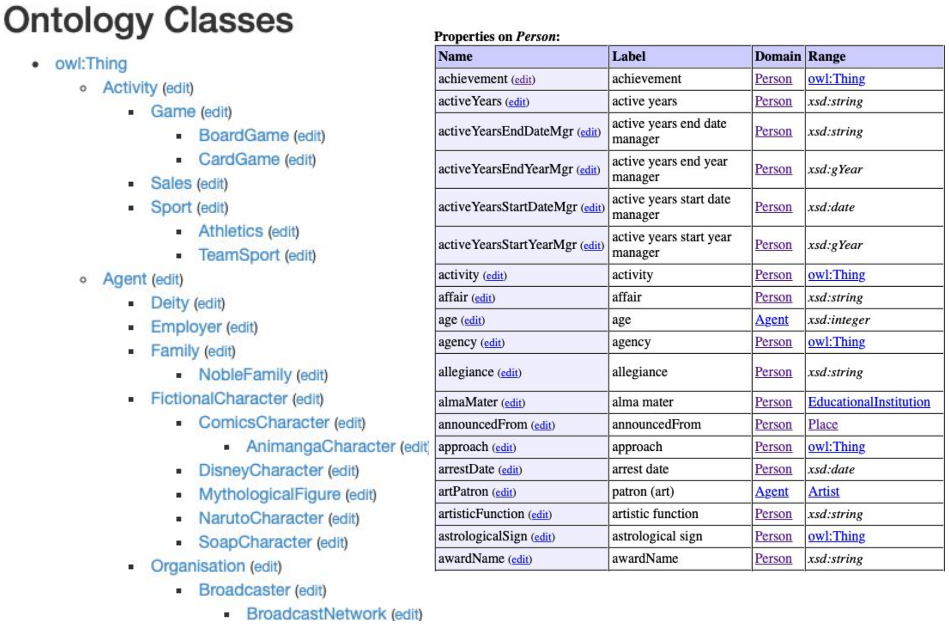
Read more about editing the ontology schema.

You can see the result of your edit on DBpedia Live (this is BETA!): http://live.dbpedia

Ontology class (help)	
rdfs:label (el)	Πληροφορίες προσώπου
rdfs:label (en)	person
rdfs:label (eu)	pertsona
rdfs:label (da)	person
rdfs:label (de)	Person
rdfs:label (sl)	Oseba
rdfs:label (it)	persona
rdfs:label (pt)	pessoa
rdfs:label (fr)	personne
rdfs:label (ga)	duine

Browse the ontology

http://mappings.dbpedia.org/server/ontology/classes/



Ontology Classes

- owl:Thing
  - Activity (edit)
    - Game (edit)
      - BoardGame (edit)
      - CardGame (edit)
    - Sales (edit)
    - Sport (edit)
      - Athletics (edit)
      - TeamSport (edit)
  - Agent (edit)
    - Deity (edit)
    - Employer (edit)
    - Family (edit)
      - NobleFamily (edit)
    - FictionalCharacter (edit)
      - ComicsCharacter (edit)
        - AnimangaCharacter (edit)
      - DisneyCharacter (edit)
      - MythologicalFigure (edit)
      - NarutoCharacter (edit)
      - SoapCharacter (edit)
    - Organisation (edit)
      - Broadcaster (edit)
        - BroadcastNetwork (edit)

Properties on *Person*:

Name	Label	Domain	Range
achievement (edit)	achievement	Person	owl:Thing
activeYears (edit)	active years	Person	*xsd:string*
activeYearsEndDateMgr (edit)	active years end date manager	Person	*xsd:string*
activeYearsEndYearMgr (edit)	active years end year manager	Person	*xsd:gYear*
activeYearsStartDateMgr (edit)	active years start date manager	Person	*xsd:date*
activeYearsStartYearMgr (edit)	active years start year manager	Person	*xsd:gYear*
activity (edit)	activity	Person	owl:Thing
affair (edit)	affair	Person	*xsd:string*
age (edit)	age	Agent	*xsd:integer*
agency (edit)	agency	Person	owl:Thing
allegiance (edit)	allegiance	Person	*xsd:string*
almaMater (edit)	alma mater	Person	EducationalInstitution
announcedFrom (edit)	announcedFrom	Person	Place
approach (edit)	approach	Person	owl:Thing
arrestDate (edit)	arrest date	Person	*xsd:date*
artPatron (edit)	patron (art)	Agent	Artist
artisticFunction (edit)	artistic function	Person	*xsd:string*
astrologicalSign (edit)	astrological sign	Person	owl:Thing
awardName (edit)	awardName	Person	*xsd:string*

DBpedia SPARQL Endpoints

Three core SPARQL endpoints:

1) DBpedia main SPARQL endpoint
   a) https://dbpedia.org/sparql
   b) hosts the DBpedia latest core release (tiny diamond, see next slide on the KG diamonds)
   c) see https://databus.dbpedia.org/dbpedia/collections/latest-core

2) Databus SPARQL endpoint
   a) hosts the data artifacts metadata
   b) https://databus.dbpedia.org/sparql

3) DBpedia Live endpoint*
   a) serves live extracted data
   b) http://live.dbpedia.org/sparql

* DBpedia live is under maintenance currently.

The Power of the DBpedia Knowledge Graph

Main SPARQL endpoint: https://dbpedia.org/sparql

Simple example: *“persons, their names in English, their birth country and country population”*

```
SELECT ?person ?name ?country ?population WHERE {
    ?person a dbo:Person .
    ?person rdfs:label ?name .
    ?person dbo:birthPlace ?country .
    ?country dbo:populationTotal ?population .
    FILTER (langMatches( lang(?name), "en" ) )
}
```

The Power of the DBpedia Knowledge Graph

Main SPARQL endpoint: https://dbpedia.org/sparql

More complex query:

- *soccer players,*
- *born in a country with more than 10 million inhabitants,*
- *played as goalkeeper*
- *for a club*
- *that has a stadium*
- *with more than 30.000 seats.*

The Power of the DBpedia Knowledge Graph

Main SPARQL endpoint: https://dbpedia.org/sparql

```
SELECT DISTINCT ?personIRI ?name ?countryOfBirth ?population ?team ?stadium ?stadiumCapacity
WHERE {
   ?personIRI a dbo:Person .
   ?personIRI rdfs:label ?name .
   ?personIRI dbo:birthPlace ?countryOfBirth .
   ?countryOfBirth dbo:populationTotal ?population .
   ?personIRI dbo:team ?team .
   ?personIRI dbo:position|dbp:position <http://dbpedia.org/resource/Goalkeeper_(association_football)> .
   ?team dbo:stadium ?stadium .
   ?stadium dbo:seatingCapacity ?stadiumCapacity .

   FILTER (langMatches( lang(?name), "EN" ) )
   FILTER (?stadiumCapacity > 30000)
   FILTER (?population > 10000000)

} ORDER BY DESC(?stadiumCapacity)
```

DBpedia

Q&A

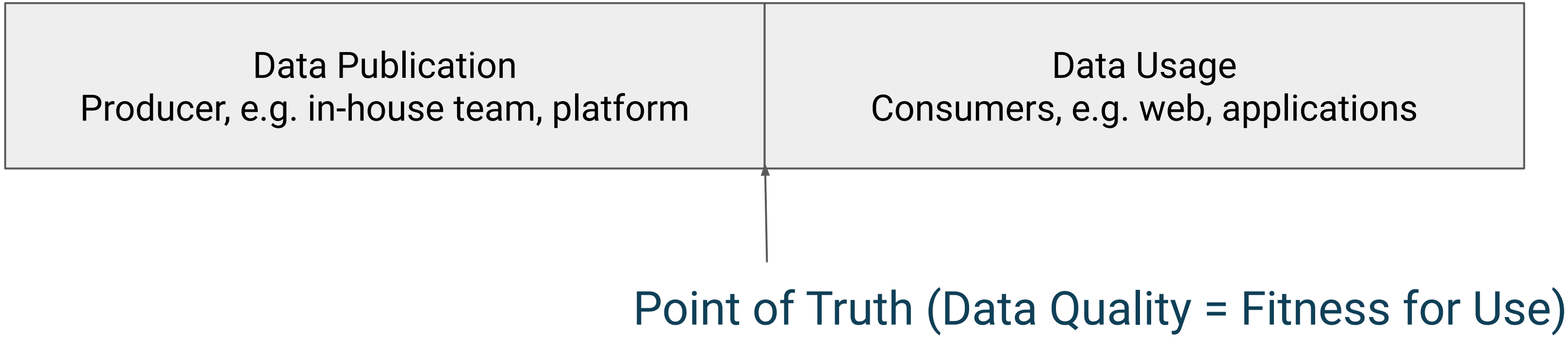
Introduction to Databus Use Cases

by Sebastian Hellmann

Databus Vision

DBpedia Databus tackles the challenges of **data acquisition** and **reuse** by offering a **comprehensive catalog** that simplifies **finding, accessing, and building** on data.

- coded on the pain points, by data engineers, for data engineers
- powers almost every aspect of DBpedia (internally and externally)
- tackles the “Data Quality” challenge



Data Publication
Producer, e.g. in-house team, platform

Data Usage
Consumers, e.g. web, applications

Point of Truth (Data Quality = Fitness for Use)

Databus - DCAT on Steroids I

Databus - lightweight, scalable, adaptable, powerful Data Catalog Platform

- Open Source (Apache 2) implementation
- Built on Data Catalog Vocabulary (DCAT) W3C standard, but fixes problems in the DCAT model

Importance of Metadata

- Imagine a library without a catalog and systematic numbers on the shelfs
  - Which book should I get? findability
  - Where is the book? accessibility
  - Operations: borrow / return / add a book
  - Data Quality = Fitness for Use -> Anything that Impacts Usability

Databus - DCAT on Steroids II (Details Matter)

	Feb 2020	Jan 2024	2014 DataID - 2024	
Feature	**DCAT 2**	**DCAT 3**	**Databus Ontology**	**Difference**
Abstract Dataset	X	~	✓	same dataset, different version
Versioning	X	~	✓	easier to query versions
Format/Compression	X	X	✓	.csv.gz
SHACL	X	X	✓	validated, consistent fields
Multi-file Distribution	X	X	✓	not only different variants
Identifier Scheme	X	X	✓	navigation & persistence
Collections	Publisher	Publisher	User	usage-centric

> 3 years ahead of DCAT

Databus Use Cases for DBpedia

Covered by this tutorial (focused on Data Usage):

- Making your own DBpedia KG collection
- Building data-rich applications with Docker and Databus
  - Virtuoso SPARQL Database, Terminology Server / Lookup, Spotlight
- Building a workflow for text enrichment & entity linking with Spotlight
- Add Custom Metadata and Search (MOSS)
- Continuous Integration (Data production and Quality Control)

Not covered (Creation):

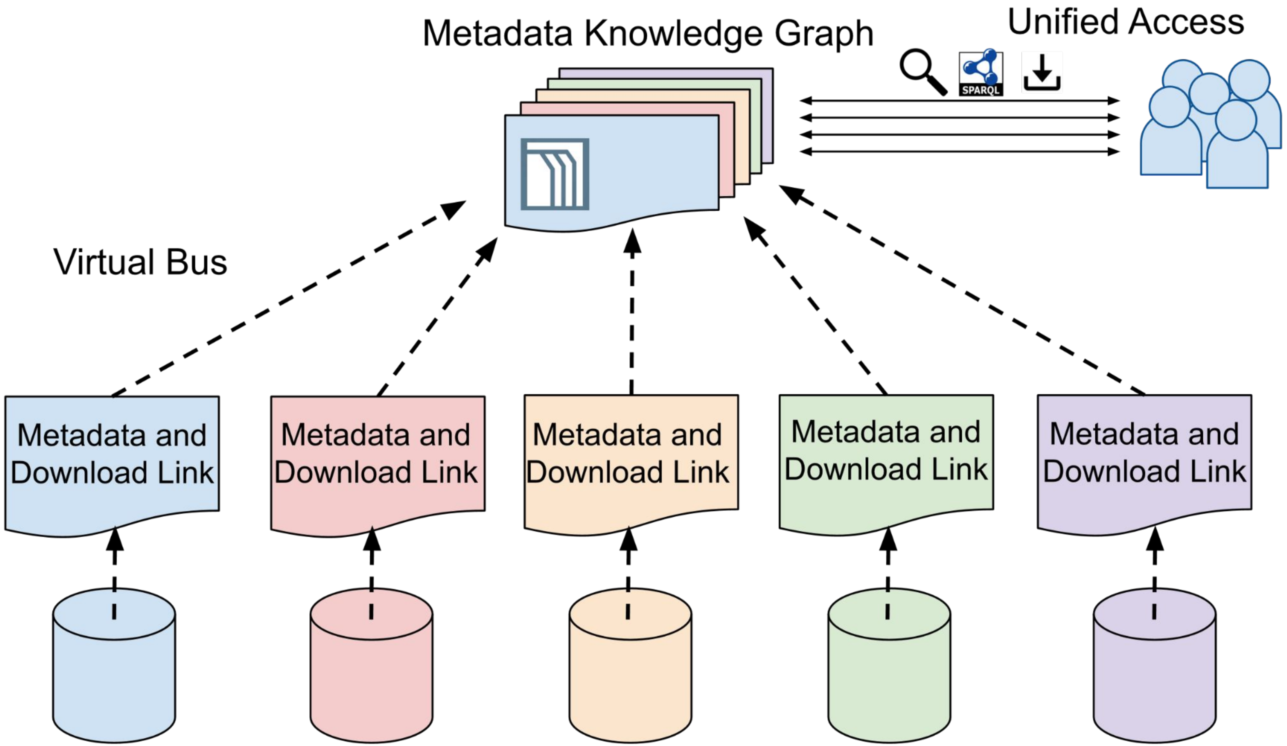
- Creating a Community Extension (links, cleaned data, additional KG data)
- Deploying your own Databus in your project/research group/company

Beyond DBpedia I : APIfying Decentral Files

- Crawling, structuring and archiving web data (Archivo)
  - ontologies down? https://databus.dbpedia.org
- Designating a Databus account, e.g. databus.dbpedia.org/ontologies
- Build a feeder that registers data on the bus
- Query with Linked Data, SPARQL, API

```
curl -H "Accept: application/ld+json"
https://databus.dbpedia.org/ontologies/georss.org/georss/2020.08.10-110000
```

Beyond DBpedia II: Access Control


Metadata Knowledge Graph
Unified Access
SPARQL
Virtual Bus
Metadata and Download Link
Metadata and Download Link
Metadata and Download Link
Metadata and Download Link
Metadata and Download Link

Metadata KG Access:
databus.coypu.org

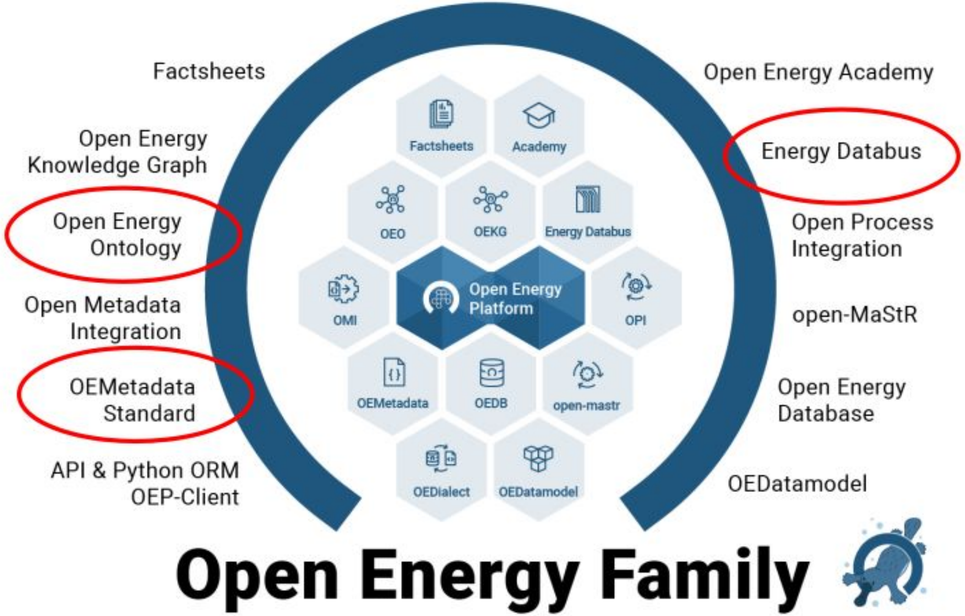
Data Storage Access:

Keycloak
JSON Web Token

Towards a Gold Metadata standard in Energy System Research

- The **Open Energy Family** is an initiative for open and FAIR data in the domain of energy systems research

- Development of a FAIR infrastructure within the Open Energy Family



Factsheets

Open Energy Knowledge Graph

Open Energy Ontology

Open Metadata Integration

OEMetadata Standard

API & Python ORM OEP-Client

Open Energy Academy

Energy Databus

Open Process Integration

open-MaStR

Open Energy Database

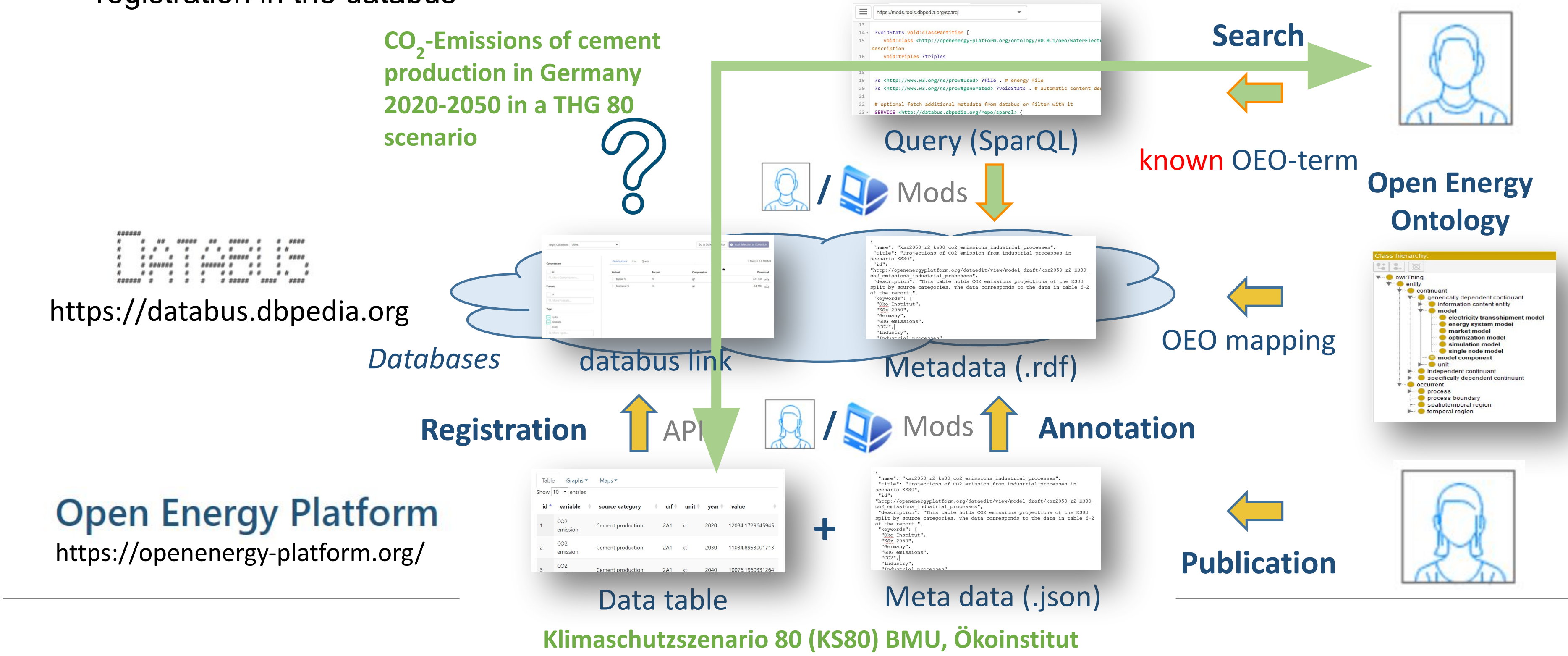
OEDatamodel

Open Energy Family

Research Grant 03EI1005A-G

Demonstrator: Publication of a Data Set Using the databus

- **Goal**: Demonstration of the improved visibility and improved discovery of a data set through the registration in the databus



https://mods.tools.dbpedia.org/sparql

```
?voidStats void:classPartition [
   void:class <http://openenergy-platform.org/ontology/v0.0.1/oeo/WaterElectr
description
   void:triples ?triples

?s <http://www.w3.org/ns/prov#used> ?file . # energy file
?s <http://www.w3.org/ns/prov#generated> ?voidStats . # automatic content des

# optional fetch additional metadata from databus or filter with it
SERVICE <http://databus.dbpedia.org/repo/sparql> {
```

Target Collection: cities — Go to Collection Editor — Add Selection to Collection

Compression: ☐ gz — More Compressions...

Format: ☐ nt — More Formats...

Type: ☑ hydro ☑ biomass ☐ wind — More Types...

Distributions | List | Query — 2 file(s) / 2.8 MB MB

Variant	Format	Compression	Download
hydro, rli	nt	gz	651 KB
biomass, rli	nt	gz	2.1 MB

```
{
 "name": "ksz2050_r2_ks80_co2_emissions_industrial_processes",
 "title": "Projections of CO2 emission from industrial processes in
scenario KS80",
 "id":
"http://openenergyplatform.org/dataedit/view/model_draft/ksz2050_r2_KS80_
co2_emissions_industrial_processes",
 "description": "This table holds CO2 emissions projections of the KS80
split by source categories. The data corresponds to the data in table 6-2
of the report.",
 "keywords": [
  "Öko-Institut",
  "KSz 2050",
  "Germany",
  "GHG emissions",
  "CO2",
  "Industry",
  "Industrial processes"
```

Class hierarchy:

- owl:Thing
  - entity
    - continuant
      - generically dependent continuant
        - information content entity
        - **model**
          - **electricity transshipment model**
          - **energy system model**
          - **market model**
          - **optimization model**
          - **simulation model**
          - **single node model**
        - **model component**
        - unit
      - independent continuant
      - specifically dependent continuant
    - occurrent
      - process
      - process boundary
      - spatiotemporal region
      - temporal region

Table | Graphs | Maps

Show 10 entries

id	variable	source_category	crf	unit	year	value
1	CO2 emission	Cement production	2A1	kt	2020	12034.1729645945
2	CO2 emission	Cement production	2A1	kt	2030	11034.8953001713
3	CO2	Cement production	2A1	kt	2040	10076.1960331264

```
{
 "name": "ksz2050_r2_ks80_co2_emissions_industrial_processes",
 "title": "Projections of CO2 emission from industrial processes in
scenario KS80",
 "id":
"http://openenergyplatform.org/dataedit/view/model_draft/ksz2050_r2_KS80_
co2_emissions_industrial_processes",
 "description": "This table holds CO2 emissions projections of the KS80
split by source categories. The data corresponds to the data in table 6-2
of the report.",
 "keywords": [
  "Öko-Institut",
  "KSz 2050",
  "Germany",
  "GHG emissions",
  "CO2",
  "Industry",
  "Industrial processes"
```

DBpedia Databus + Collections

by Jan Forberg

Chapter Outline

- Databus
  - Concepts
  - Interface
  - API
  - Deployment and Customization
- Collections
  - Concept
  - The DBpedia Collections
  - Creating your own Collection

Databus

- Decentralized RDF-metadata storage
- Holds basic description of files as **RDF**
  - License information
  - Checksums
  - Formats/Compressions
  - “Structural” Information
- Offers stable identifiers for files
- Extension point for more complex metadata
- The soil on which more RDF-metadata can grow

Databus - Concepts

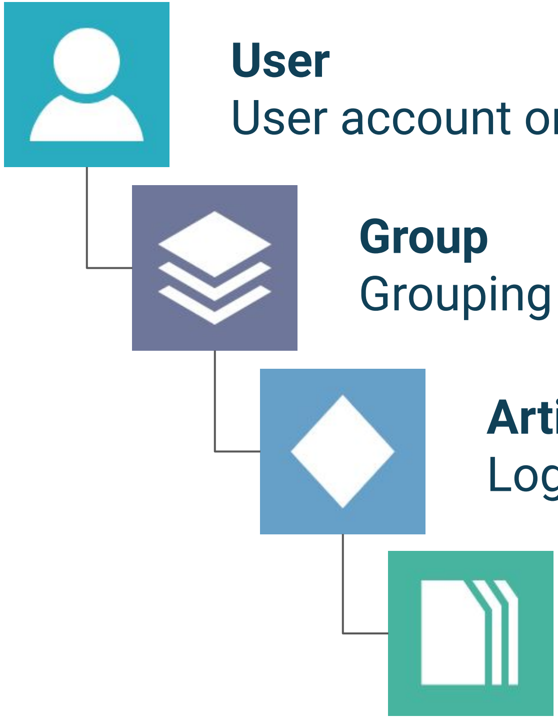
User
User account on the Databus

Group
Grouping element for artifacts

Artifact
Logical dataset, may have multiple versions (e.g. “DBpedia Labels”)

Version
Version of an artifact, a set of files

Databus - Hierarchy



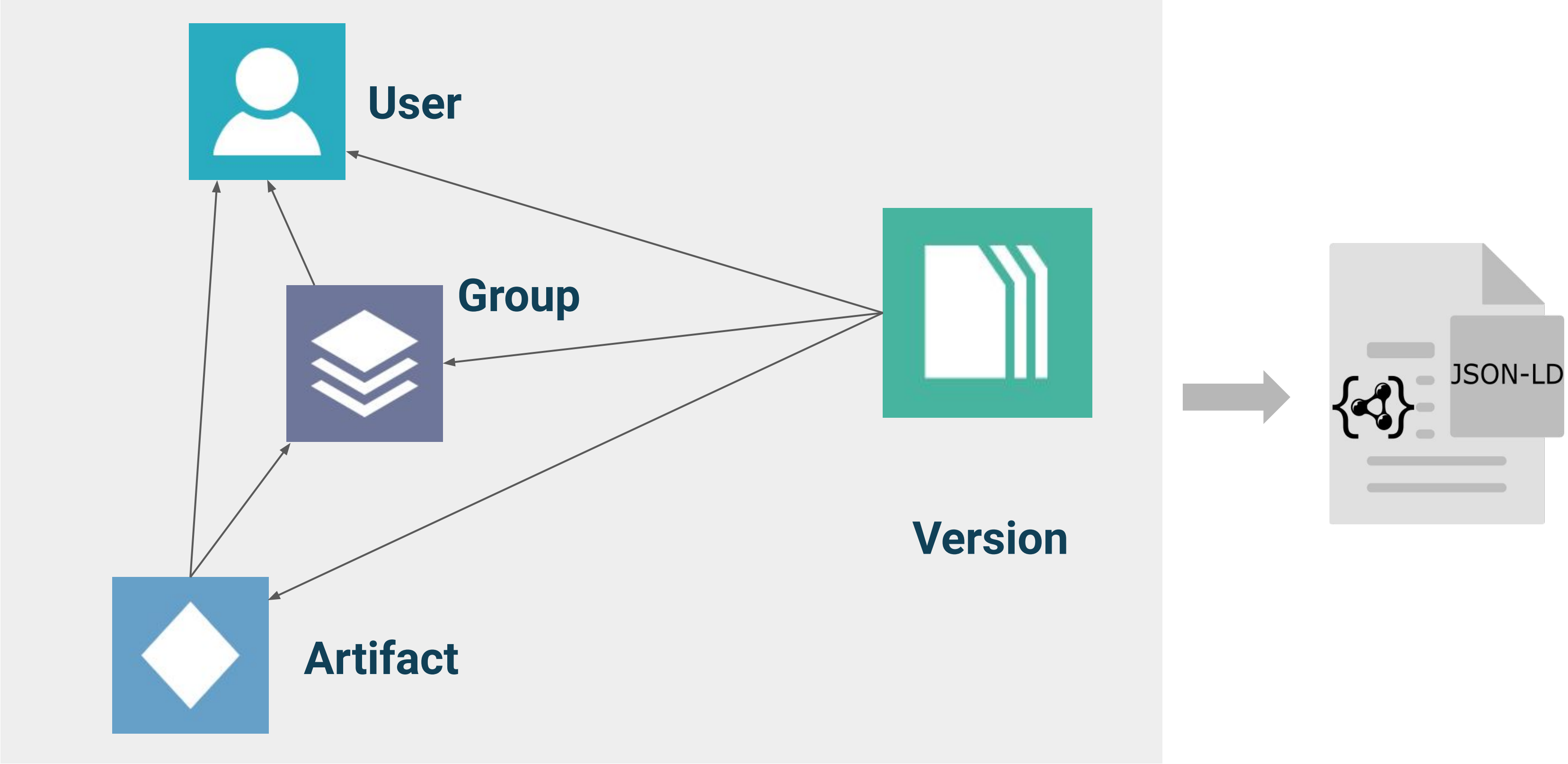
User
User account on the Databus

Group
Grouping element for artifacts

Artifact
Logical dataset

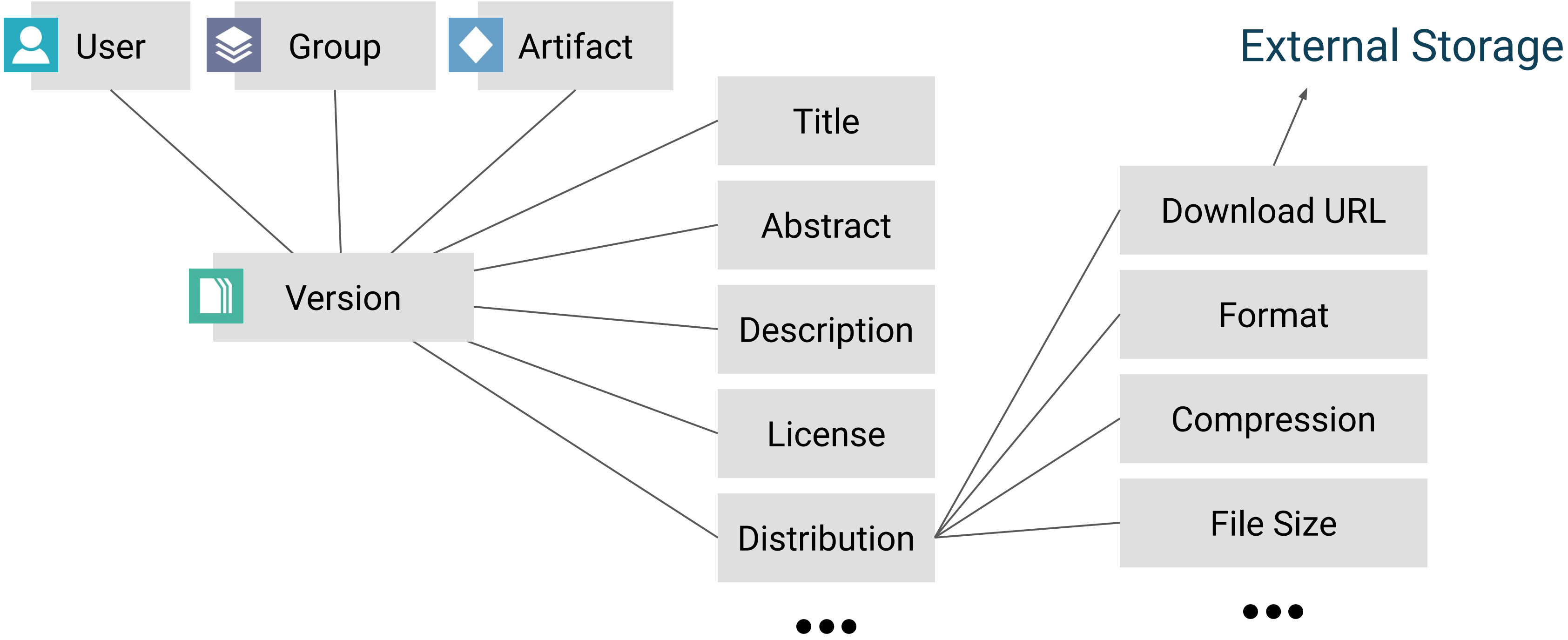
Version
Version of an artifact, a set of files

Databus - Links

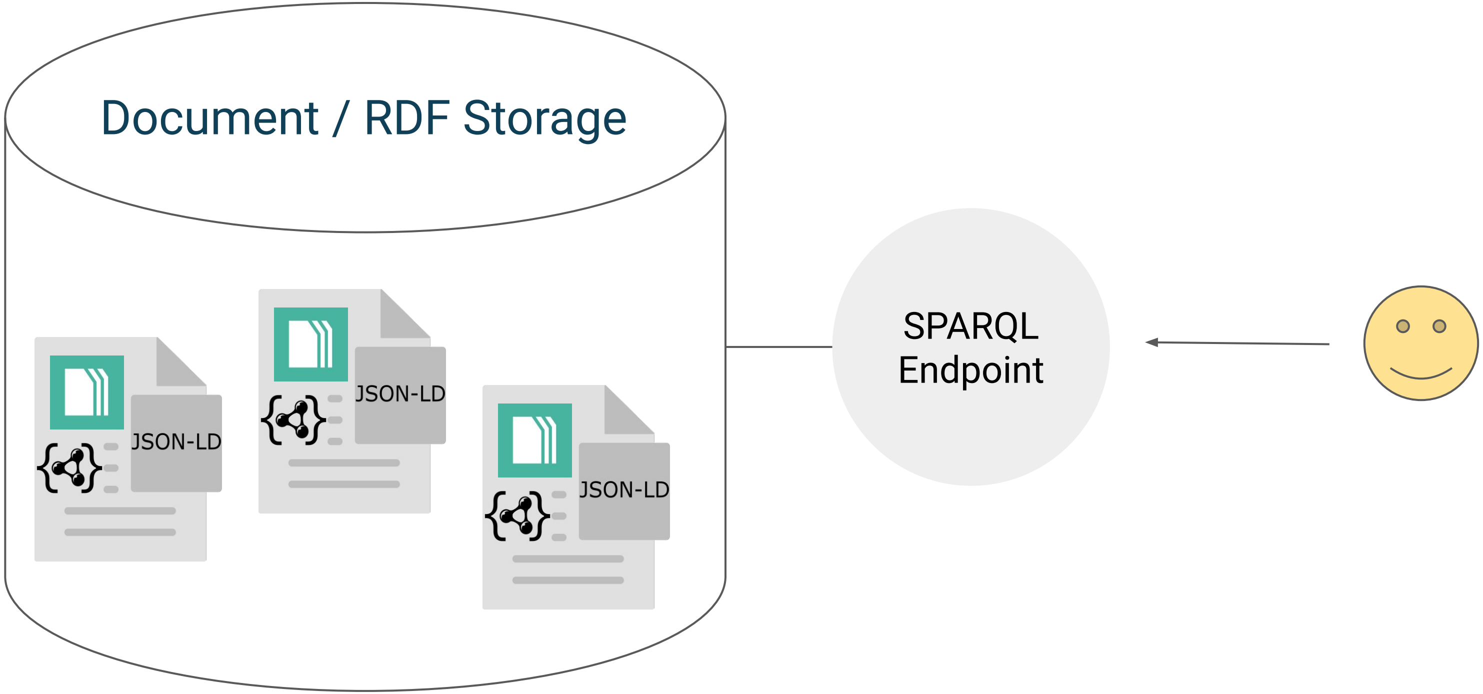


```
{
   "@context": "https://databus.dbpedia.org/res/context.jsonld",
   "@graph": [
      {
         "@id": "https://databus.dbpedia.org/dbpedia/prefusion",
         "@type": "Group",
         "account": "https://databus.dbpedia.org/dbpedia"
      },
      {
         "@id": "https://databus.dbpedia.org/dbpedia/prefusion/labels",
         "@type": "Artifact",
         "account": "https://databus.dbpedia.org/dbpedia",
         "group": "https://databus.dbpedia.org/dbpedia/prefusion"
      },
      {
         "@id": "https://databus.dbpedia.org/dbpedia/prefusion/labels/2019.03.01",
         "@type": "Version",
         "title": "DBpedia PreFusion labels",
         "abstract": "DBpedia PreFusion labels",
         "account": "https://databus.dbpedia.org/dbpedia",
         "group": "https://databus.dbpedia.org/dbpedia/prefusion",
         "artifact": "https://databus.dbpedia.org/dbpedia/prefusion/labels",
         "distribution": [
            "https://databus.dbpedia.org/dbpedia/prefusion/labels/2019.03.01#labels_sources=dbpw_tag=context.jsonld",
            "https://databus.dbpedia.org/dbpedia/prefusion/labels/2019.03.01#labels_sources=dbpw_tag=median.tsv.bz2",
            "https://databus.dbpedia.org/dbpedia/prefusion/labels/2019.03.01#labels_sources=dbpw.jsonld.bz2"
         ],
         ...
      },
     ...
```

Databus Versions



Storage


Document / RDF Storage
JSON-LD
JSON-LD
JSON-LD
SPARQL Endpoint

Interface

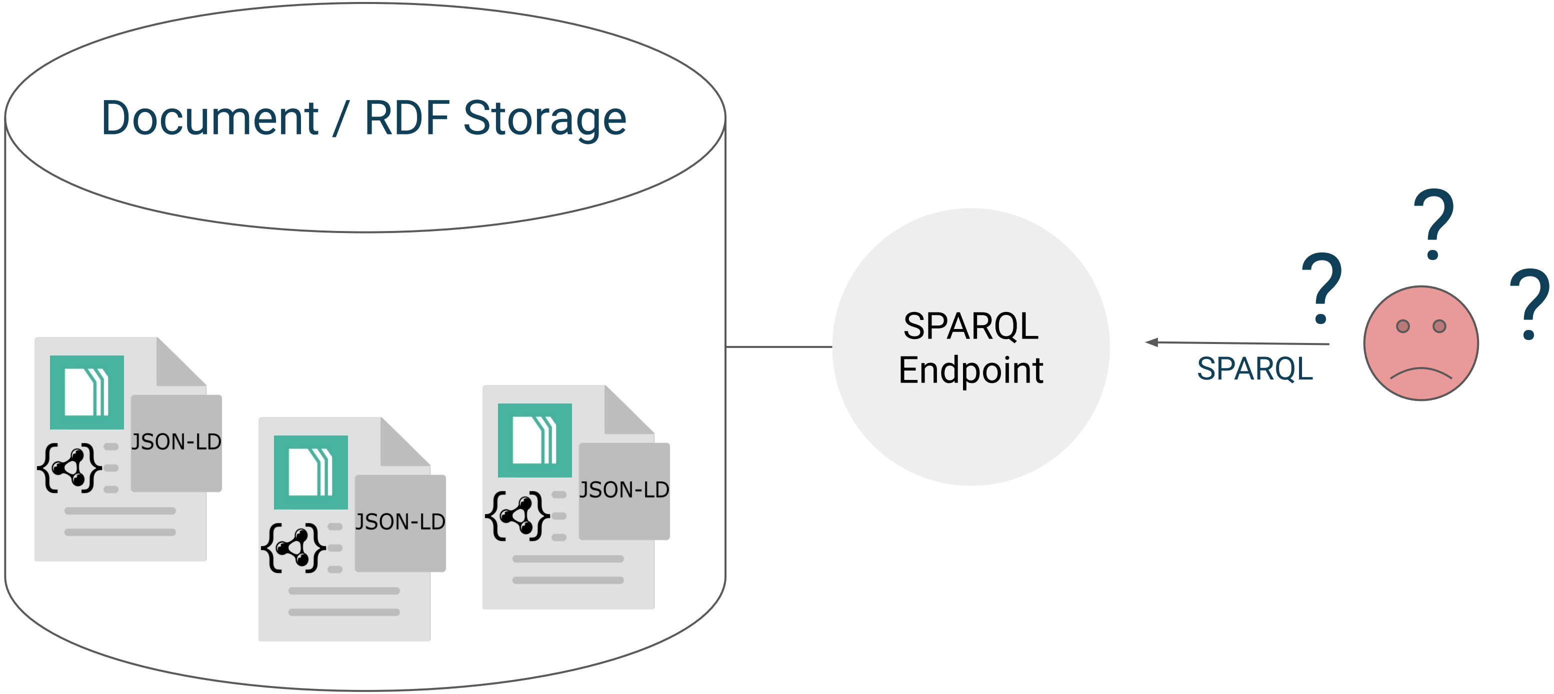
https://databus.dbpedia.org/

Deployment & Customization

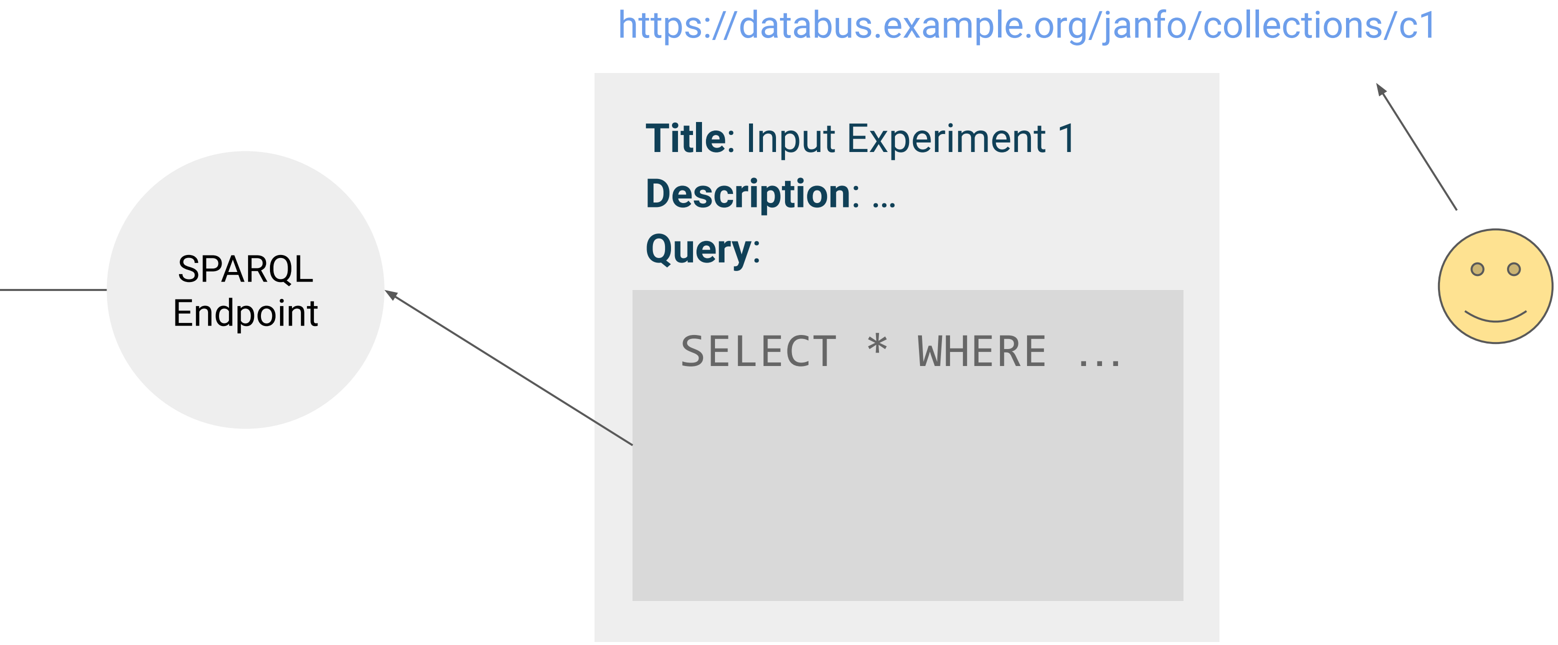
- Dockerized Deployment
  - Virtuoso Store
  - Gstore (adds document-store layer)
  - Databus API and Webapp
  - Lookup (for indexed search)
- Process is documented in the Git Repository
- Need help? Contact us!
- Customization Example: https://dev.databus.dbpedia.org

Databus Collections

Databus Collections



Databus Collections



https://databus.example.org/janfo/collections/c1

Title: Input Experiment 1

Description: …

Query:

```
SELECT * WHERE ...
```

DBpedia Data

https://databus.dbpedia.org/dbpedia/collections/latest-core

Our Own Data Collection

https://databus.dbpedia.org/

DBpedia

Making your own DBpedia KG

by Jan Forberg

Preliminaries

- Docker and Docker-Compose
- An internet connection

The Technology

- Virtuoso Opensource Triple Store
  https://hub.docker.com/r/openlink/virtuoso-opensource-7
- Databus Collection Downloader
  https://github.com/dbpedia/dbpedia-databus-collection-downloader
- DBpedia Virtuoso Quickstarter
  https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart

Virtuoso Opensource Triple Store

https://hub.docker.com/r/openlink/virtuoso-opensource-7

- Highly **scalable** and **performant** triplestore solution.
- Offers robust support for RDF data **storage** and **querying**.
- Easily deployable through Docker, enabling seamless integration into various environments.

Databus Collection Downloader

https://github.com/dbpedia/dbpedia-databus-collection-downloader

- **Lightweight** Downloader
- No File Conversion, Decompression or Mapping (see Databus Client)
- Simple GET requests to retrieve
  - Collection query
    `GET -H “Accept: text/sparql” [COLLECTION_URI]`
  - Download URLs
  - Files

DBpedia Virtuoso Quickstarter

https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart

- Waits for the Virtuoso to initialize
- Waits for the downloader to finish

Then

- Tells Virtuoso to load local data from disk
- Installs the Virtuoso DBpedia Plugin
    - Includes the DBpedia HTML pages

DEMO TIME

Q&A

Lunch Break

… we will continue at 13:30

Session 2: DBpedia and Databus Showcases

Semantic Text Annotation and Search using Spotlight and Databus

CI and Databus publishing using Jenkins

Semantic Text Annotation and Search using DBpedia Spotlight and Databus

by Jan Forberg

Use Cases

- SPARQL/Linked Data: Get more data
- SPARQL/Linked Data: filter documents
- prompt enrichment / RAG

Goal

Improve information requests using the DBpedia Technology Stack:

- DBpedia Databus
- DBpedia Spotlight
- Virtuoso Quickstarter

The Idea

We want to filter documents based on their content. Can we do more than just searching for words?

- Entity Classification and Tagging
- Link to additional data

Let’s try to apply **geospatial** queries to topics in text documents!

- Link entities in text to geospatial data
- Retrieve documents with topics about things in a certain area

Tools

- bash-able CLI
- A Github account
- Docker
- Docker Compose
- A web browser

Preparations

- Create a Databus Account
- Create a Databus API Key

Resources

https://github.com/dbpedia/tutorials/tree/master/dataweek24/

RUN:

```
git clone https://github.com/dbpedia/tutorials.git
cd tutorials/dataweek24/use-case
```

Tasks

- Create a Data Producer
  - Create file annotations
  - Upload file annotations
  - Publish file annotations on the Databus
- Create a Data Consumer
  - Start a triple store with a mix of **our** data and **DBpedia** data

Steps

- Get some text **documents**
- **Annotate** the text documents with DBpedia identifiers
- **Upload** the annotations
- **Publish** the annotations on our Databus
- Create a **Databus Collection** with annotations and geodata from another Publisher
- **Load** the Databus Collection into a triple store
- Send a **SPARQL query** to the triple store

Steps

- Get some text **documents**
- **Annotate** the text documents with DBpedia identifiers
- **Upload** the annotations
- **Publish** the annotations on our Databus
- Create a **Databus Collection** with annotations and geodata from another Publisher
- **Load** the Databus Collection into a triple store
- Send a **SPARQL query** to the triple store

Data Producer

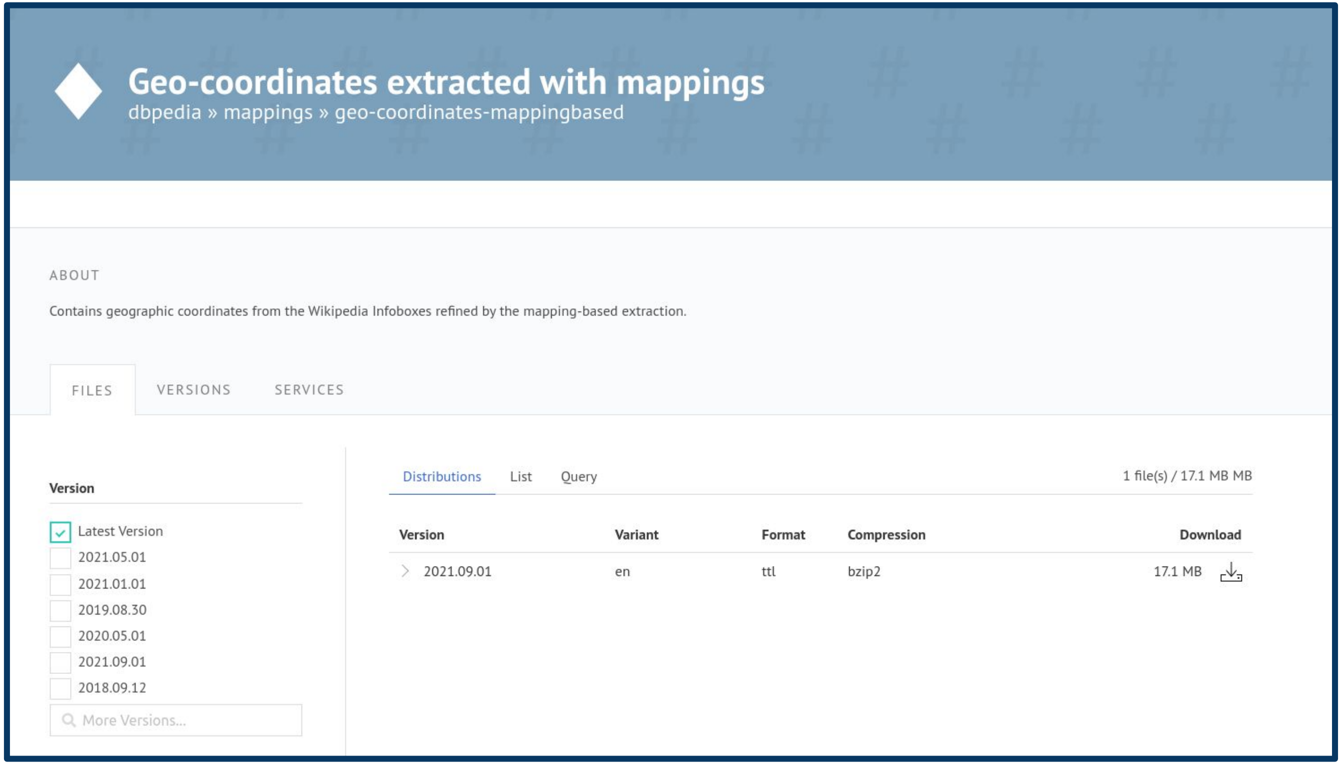
Data Consumer

Choosing the complementary data

- Geo-coordinates!
- Mapping-based or Generic Geo-coordinates?
    - Mappings: more precision
    - Generic: more recall
- Latest version generally a good idea

https://databus.dbpedia.org/dbpedia/mappings/geo-coordinates-mappingbased/

Choosing the complementary data



Geo-coordinates extracted with mappings

dbpedia » mappings » geo-coordinates-mappingbased

ABOUT

Contains geographic coordinates from the Wikipedia Infoboxes refined by the mapping-based extraction.

Version

- ☑ Latest Version
- □ 2021.05.01
- □ 2021.01.01
- □ 2019.08.30
- □ 2020.05.01
- □ 2021.09.01
- □ 2018.09.12

1 file(s) / 17.1 MB MB

Version	Variant	Format	Compression	Download
2021.09.01	en	ttl	bzip2	17.1 MB

Step 1

Find some text documents

https://github.com/dbpedia/tutorials/tree/master/dataweek24/use-case/automobile-industry-texts

** we have prepared some*

Step 2

Build a Data Producer

https://github.com/dbpedia/tutorials/blob/master/dataweek24/use-case/annotate.sh

Step 3

Create a Databus Collection

https://databus.dbpedia.org

https://databus.dbpedia.org/USERNAME/collections

Step 4

Load the Collection to a local triple store.

clone:
https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart

run:
COLLECTION_URI=https://databus.dbpedia.org/m1ci/collections/dataweek2024 VIRTUOSO_ADMIN_PASSWD=YourSecretPassword docker-compose up

Step 5

Send some queries.

http://localhost:8890/sparql

Queries

```
SELECT DISTINCT ?s ?o WHERE {
   ?s <http://www.w3.org/2005/11/its/rdf#taIdentRef> ?o .
}
```

Queries

```
SELECT * WHERE {
   ?s <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?o .
}
```

Queries

```
SELECT DISTINCT ?s ?o WHERE {
   ?s <http://www.w3.org/2005/11/its/rdf#taIdentRef> ?o .
   ?o <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat .
   FILTER(?lat > 0)
}
```

Queries

```
SELECT DISTINCT ?s ?o WHERE {
   ?s <http://www.w3.org/2005/11/its/rdf#taIdentRef> ?o  .
   ?o <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?long .
   FILTER(?long > 0 && ?long < 20)
}
```

Queries

```
SELECT DISTINCT ?s ?o ?p ?x WHERE {
   ?s <http://www.w3.org/2005/11/its/rdf#taIdentRef> ?o .
   ?o ?p ?x .
   ?x <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat .
   FILTER(?lat > 60)
}
```

DBpedia

Q&A

CI and Databus publishing using Jenkins

by Kirill Yankov

Introduction

Continuous integration and continuous delivery tools help in automation of your software or data release processes.

and others…

Introduction

Combination of CI tools and Databus can simplify the automation and make pipelines more reliable, especially when you work with versioned data.

Two main scenarios:

➔ you produce data and want to publish it

➔ you use some versioned data in your builds (for example: for testing or for generation of your own data)

Introduction

Demo using Jenkins Pipelines

The code from examples is available in our Gitbook @ databus.dbpedia.org

Usage -> Integration with CI: https://dbpedia.gitbook.io/databus/usage/ci

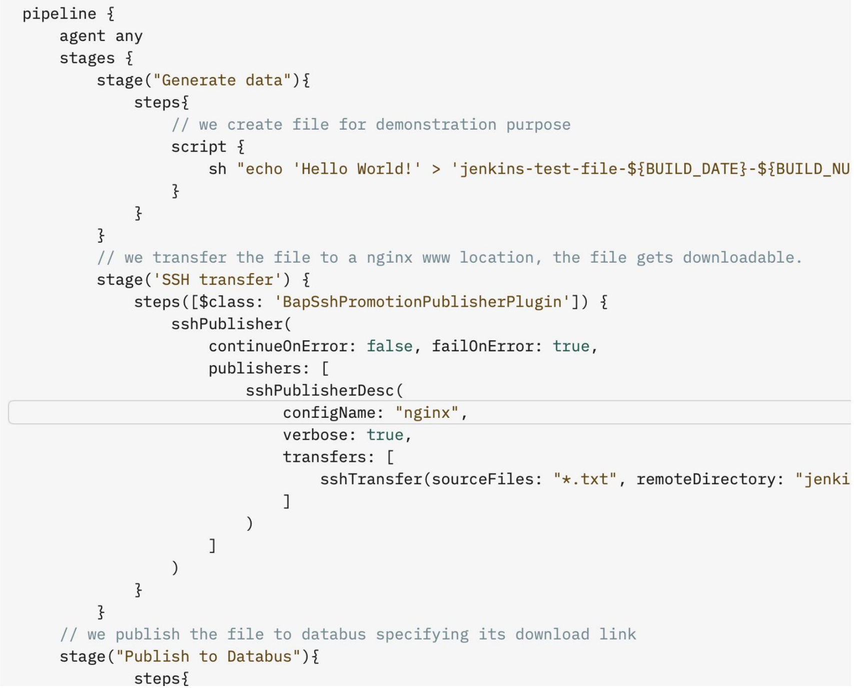
Publishing data in a Databus

Scenario description (what is going on in the pipeline):

1. we generate some data in a pipeline
2. make it available for download at some location (uploading it to nginx)
3. publish the download link to Databus

Publishing data in a Databus

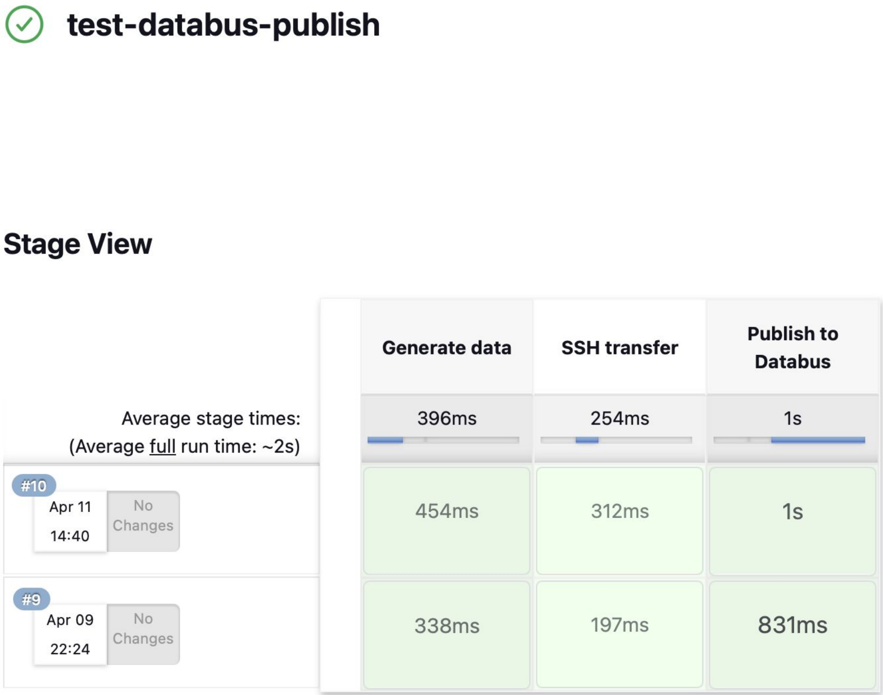
Demo code: https://dbpedia.gitbook.io/databus/usage/ci#publishing-your-data-files-datasets-into-databus



```
pipeline {
    agent any
    stages {
        stage("Generate data"){
            steps{
                // we create file for demonstration purpose
                script {
                    sh "echo 'Hello World!' > 'jenkins-test-file-${BUILD_DATE}-${BUILD_NU
                }
            }
        }
        // we transfer the file to a nginx www location, the file gets downloadable.
        stage('SSH transfer') {
            steps([$class: 'BapSshPromotionPublisherPlugin']) {
                sshPublisher(
                    continueOnError: false, failOnError: true,
                    publishers: [
                        sshPublisherDesc(
                            configName: "nginx",
                            verbose: true,
                            transfers: [
                                sshTransfer(sourceFiles: "*.txt", remoteDirectory: "jenki
                            ]
                        )
                    ]
                )
            }
        }
    // we publish the file to databus specifying its download link
    stage("Publish to Databus"){
            steps{
```

Publishing data in a Databus

Demo in Jenkins



✓ **test-databus-publish**

Stage View

	Generate data	SSH transfer	Publish to Databus
Average stage times: (Average full run time: ~2s)	396ms	254ms	1s
#10 Apr 11 14:40 No Changes	454ms	312ms	1s
#9 Apr 09 22:24 No Changes	338ms	197ms	831ms

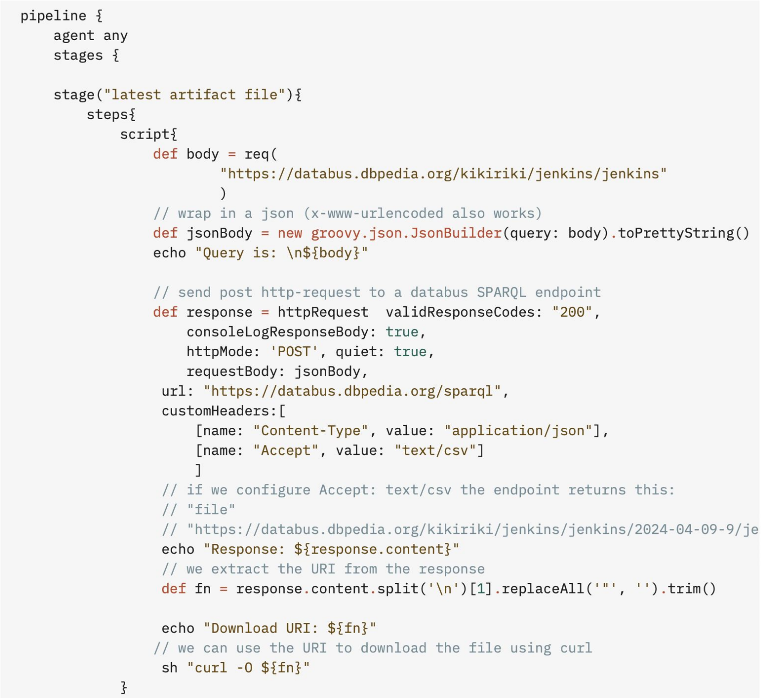
Downloading data from Databus

Pipeline scenario description:

1. we have some artifact published in databus
2. we execute a SPARQL query in Databus SPARQL endpoint to retrieve links for downloading files of an artifact
3. we use download link to download the artifacts (for example with curl)

Downloading data from Databus

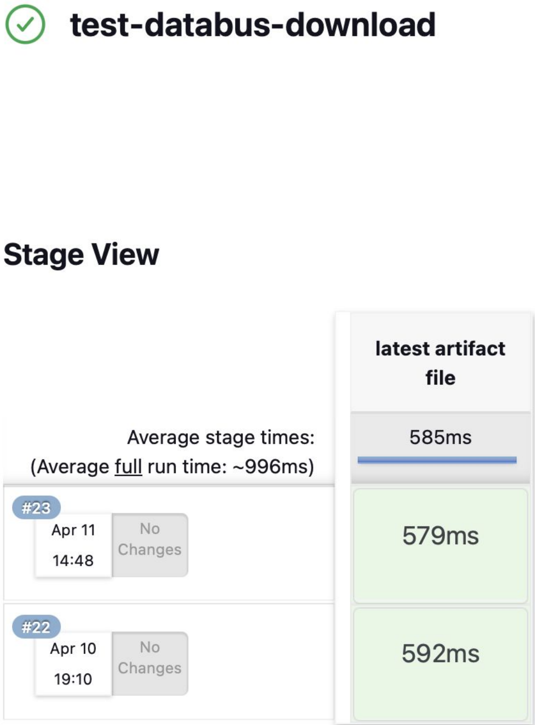
Demo code: https://dbpedia.gitbook.io/databus/usage/ci#downloading-data-files-datasets-from-databus



```
pipeline {
    agent any
    stages {

    stage("latest artifact file"){
        steps{
            script{
                def body = req(
                        "https://databus.dbpedia.org/kikiriki/jenkins/jenkins"
                        )
                // wrap in a json (x-www-urlencoded also works)
                def jsonBody = new groovy.json.JsonBuilder(query: body).toPrettyString()
                echo "Query is: \n${body}"

                // send post http-request to a databus SPARQL endpoint
                def response = httpRequest  validResponseCodes: "200",
                    consoleLogResponseBody: true,
                    httpMode: 'POST', quiet: true,
                    requestBody: jsonBody,
                 url: "https://databus.dbpedia.org/sparql",
                 customHeaders:[
                     [name: "Content-Type", value: "application/json"],
                     [name: "Accept", value: "text/csv"]
                     ]
                 // if we configure Accept: text/csv the endpoint returns this:
                 // "file"
                 // "https://databus.dbpedia.org/kikiriki/jenkins/jenkins/2024-04-09-9/je
                 echo "Response: ${response.content}"
                 // we extract the URI from the response
                 def fn = response.content.split('\n')[1].replaceAll('"', '').trim()

                 echo "Download URI: ${fn}"
                // we can use the URI to download the file using curl
                 sh "curl -O ${fn}"
            }
```

Downloading data from Databus

Demo in jenkins



✓ **test-databus-download**

Stage View

	latest artifact file
Average stage times: (Average full run time: ~996ms)	585ms
#23 Apr 11 14:48 No Changes	579ms
#22 Apr 10 19:10 No Changes	592ms

Wrap Up

Databus can be a useful instrument for your CI/CD automation:

- for storing structured metadata about the data you use in pipelines
- for fine-grained and flexible file retrieval using SPARQL queries

The code from the demo is available in our Databus Gitbook

Usage -> Integration with CI: https://dbpedia.gitbook.io/databus/usage/ci

Q&A

Coffee Break

… we will continue at 15:30

Session 3: DBpedia and Databus Showcases (cont.)

Databus Metadata Overlay Search System (MOSS)

Terminology Server using Databus, Lookup and Archivo

Databus Metadata Overlay Search System (MOSS)

by Jonathan Justavino Lüderitz

Questions

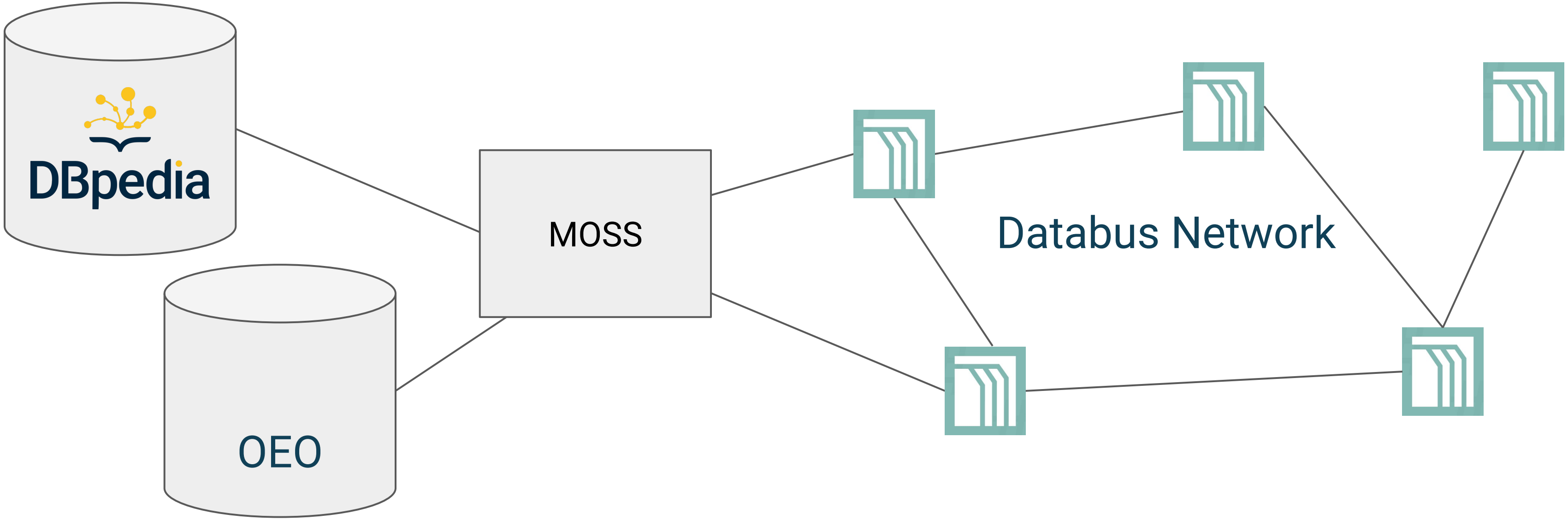
- What is MOSS?
- Why use MOSS?
- How to get started with MOSS?

What is MOSS?

- Metadata Overlay Search System
- Based on Databus technology stack components
  - Gstore
  - Lookup
- Storage and indexer of additional metadata graphs
  - Stores Databus metadata extensions
  - Offers enhanced search over files

Why use MOSS?

- Leverage RDF Metadata Interconnectivity
- Decentralized Integration with the Databus Network

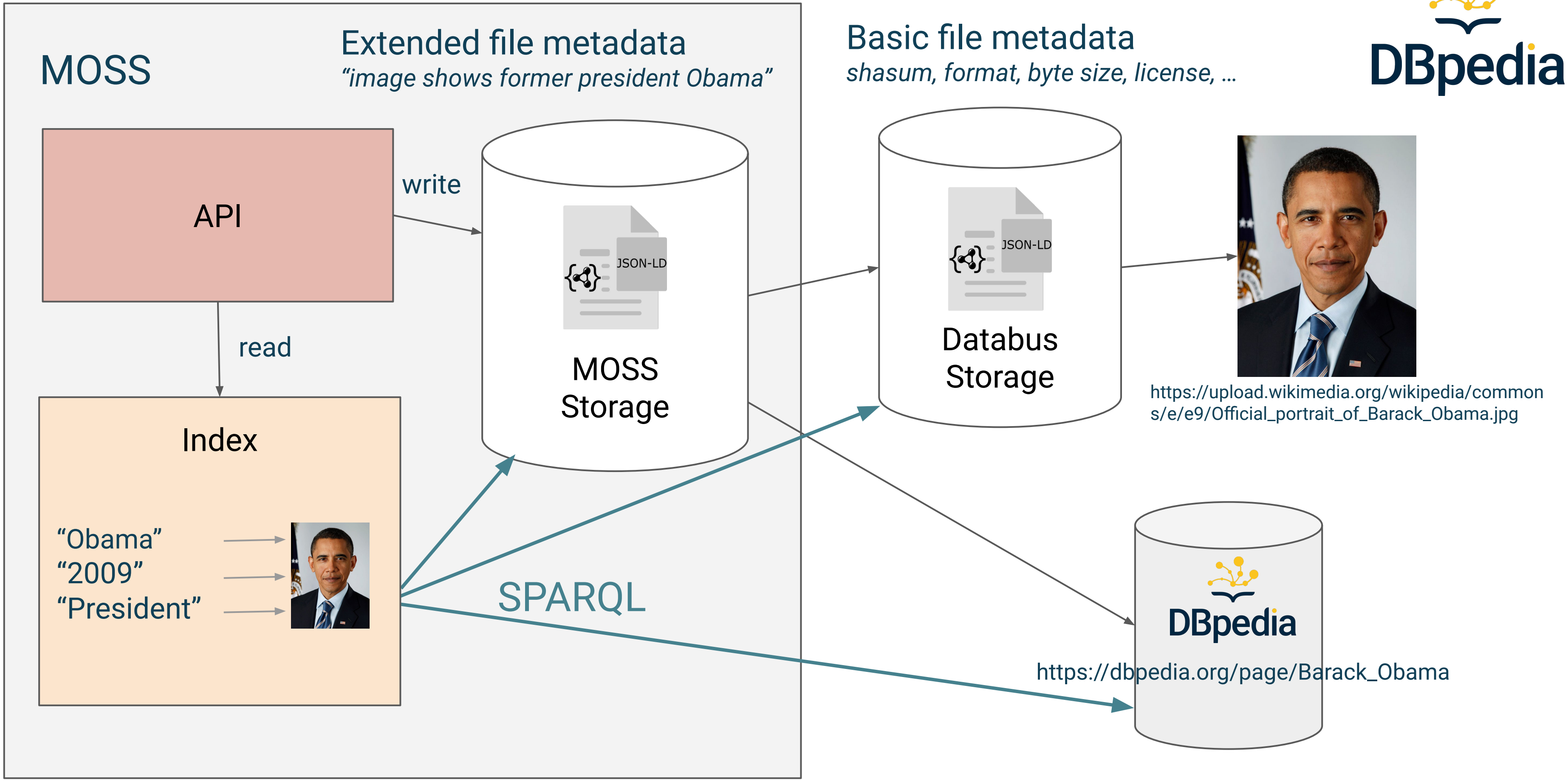


Why use MOSS?

- Databus Metadata is limited
  - Minimal Metadata: Format, Compression, Download URL
  - Need for Additional RDF Metadata Storage
- We might have additional Metadata, e.g.
  - Content Description
  - Formatting Description

“image shows former US president Obama”

“image is 1000x2500 pixels”



MOSS

Extended file metadata

“image shows former president Obama”

Basic file metadata

shasum, format, byte size, license, …

API

write

read

Index

“Obama”

“2009”

“President”

MOSS Storage

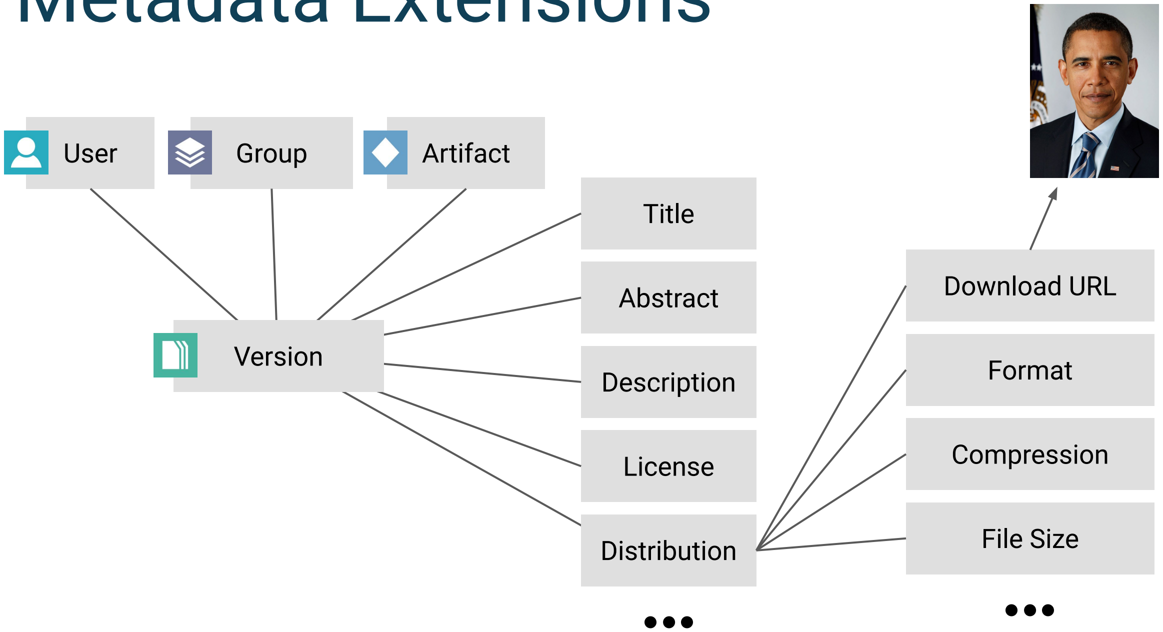
Databus Storage

SPARQL

https://upload.wikimedia.org/wikipedia/commons/e/e9/Official_portrait_of_Barack_Obama.jpg

https://dbpedia.org/page/Barack_Obama

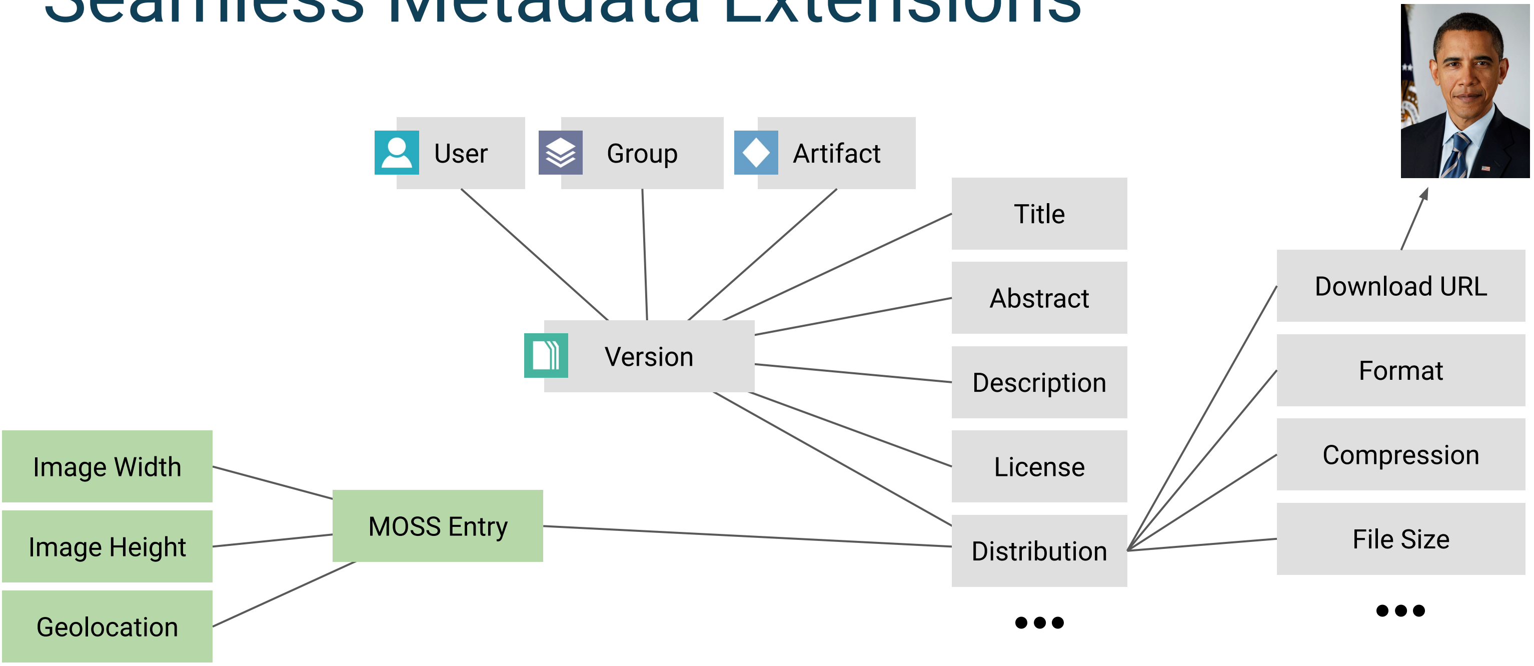
Seamless Metadata Extensions



🟩 custom metadata schema 1

🟧 custom metadata schema 2

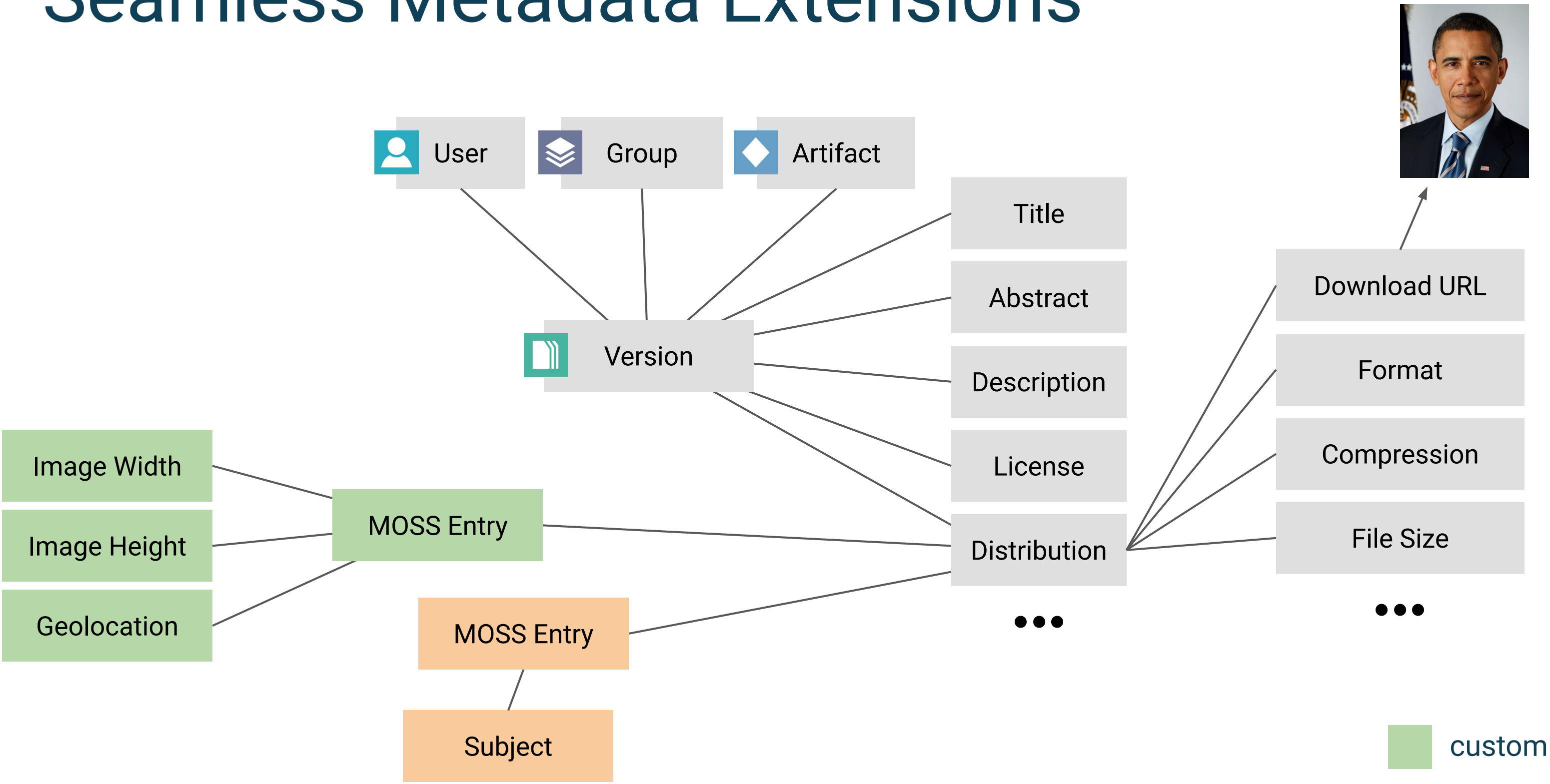
Seamless Metadata Extensions



🟩 custom metadata schema 1

🟧 custom metadata schema 2

Seamless Metadata Extensions

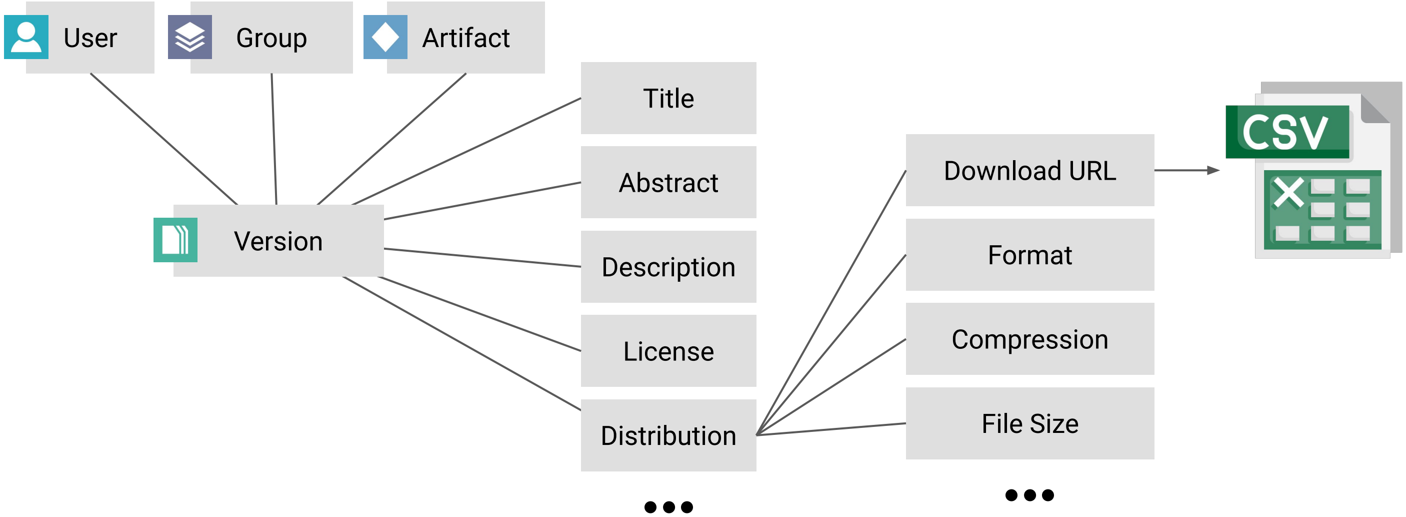


e.g. https://dbpedia.org/resource/Barack_Obama

🟩 custom metadata layer 1

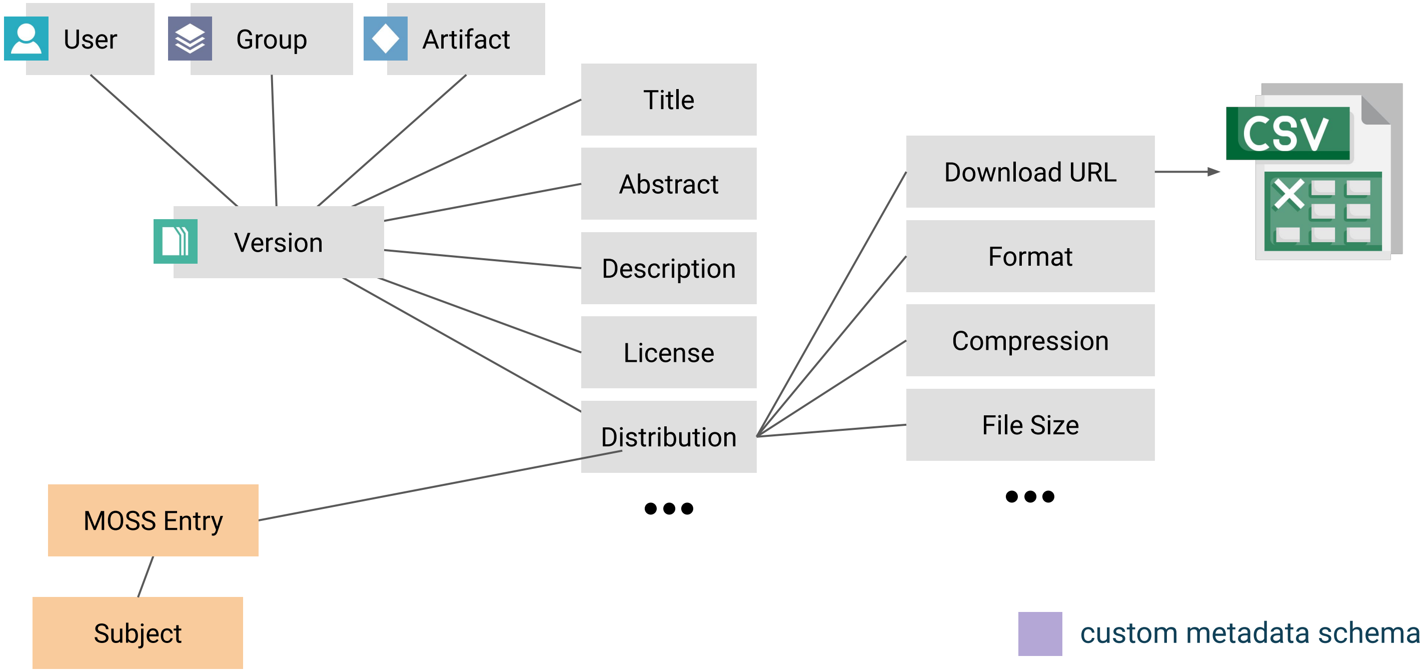
🟧 custom metadata layer 2

Seamless Metadata Extensions (Example 2)



🟪 custom metadata schema 3

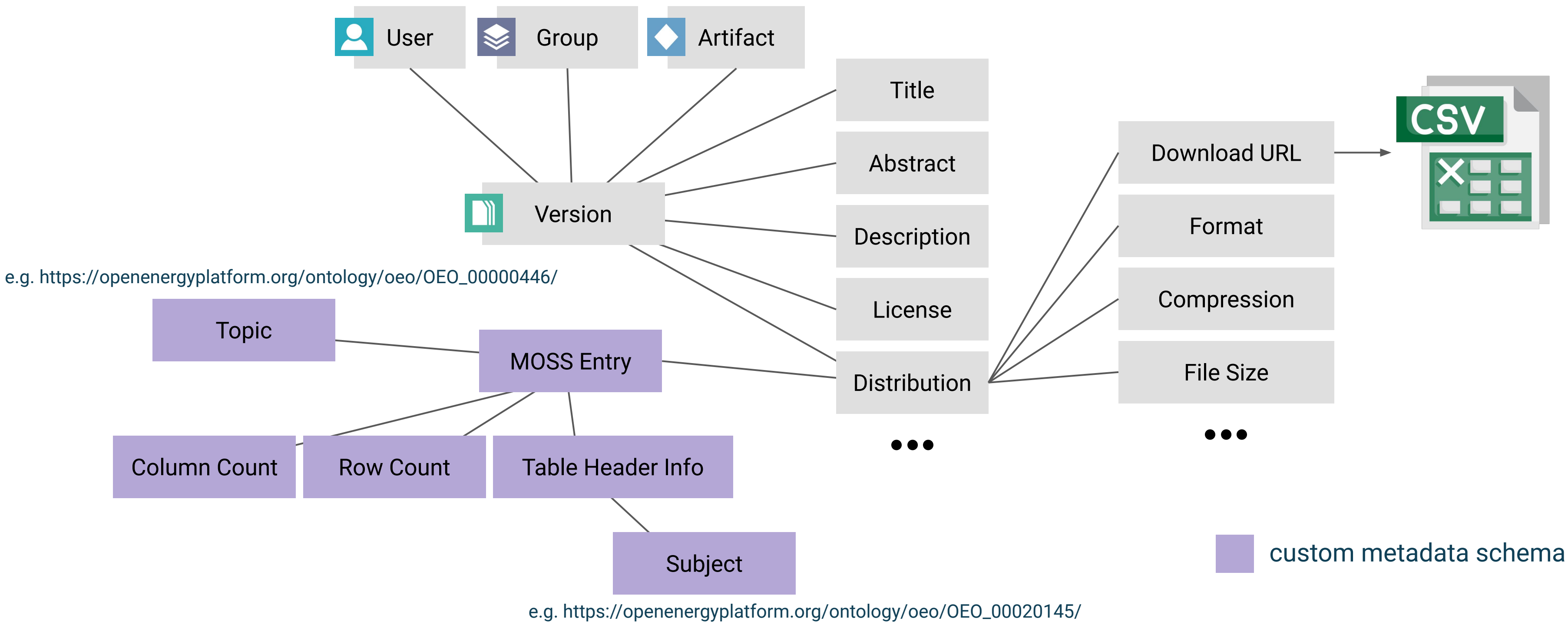
Seamless Metadata Extensions (Example 2)



e.g. https://dbpedia.org/resource/Barack_Obama

🟪 custom metadata schema 3

Seamless Metadata Extensions (Example 2)



▨ custom metadata schema 3

How to get started with MOSS

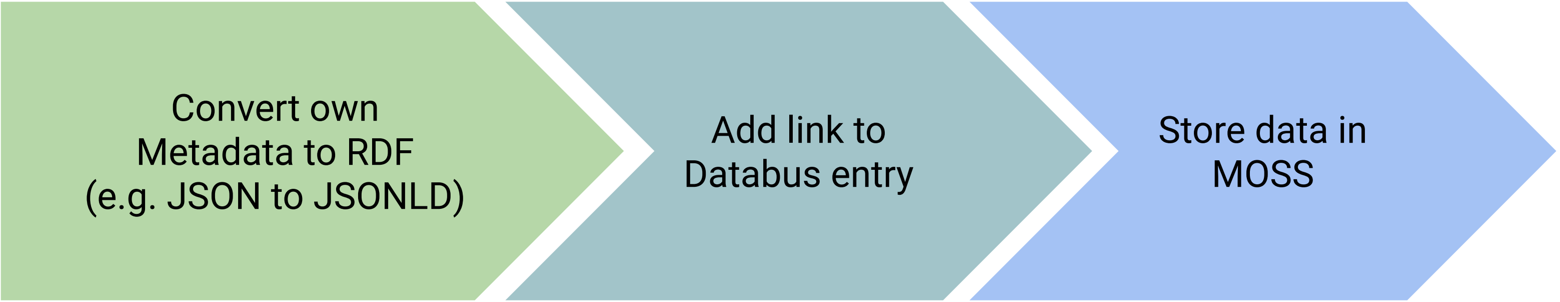
- Deploy a MOSS instance
  https://github.com/dbpedia/databus-moss/tree/dev

How to get started with MOSS

- Deploy a MOSS instance
  https://github.com/dbpedia/databus-moss/tree/dev
- Add your metadata

How to get started with MOSS

- Deploy a MOSS instance
  https://github.com/dbpedia/databus-moss/tree/dev
- Add your metadata



Convert own Metadata to RDF (e.g. JSON to JSONLD)

Add link to Databus entry

Store data in MOSS

How to get started with MOSS

- Deploy a MOSS instance
  https://github.com/dbpedia/databus-moss/tree/dev
- Add your metadata
- OPTIONAL: Add additional RDF data (e.g. an ontology)

How to get started with MOSS

- Deploy a MOSS instance
  https://github.com/dbpedia/databus-moss/tree/dev
- Add your metadata
- OPTIONAL: Add additional RDF data (e.g. an ontology)
- Add Indexer for your Metadata format
  - Can index any RDF metadata schema
  - Can index as many schema as needed
  - Can include additional RDF during indexing

How to MOSS?

- Enjoy great search results:

DEMO TIME

Capabilities of MOSS

- Storage for structured metadata
- Automated, configurable indexing of heterogeneous metadata
- Potential for advanced data retrieval via SPARQL
- Built-in flexible text search engine

- Future Work:
  - Collaborative Editing of Metadata (Wiki Approach)

Conclusion

- MOSS Enhances Databus Metadata
- Facilitates Richer Metadata Storage
- Enables Advanced Data Retrieval and Search

Q&A

Terminology Server using Databus, Lookup and Archivo

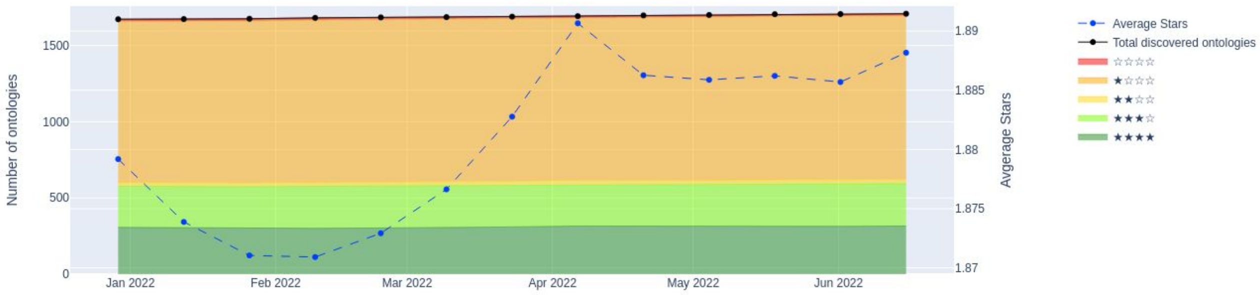
by Johannes Frey

DBpedia Archivo

Augmented Ontology Archive

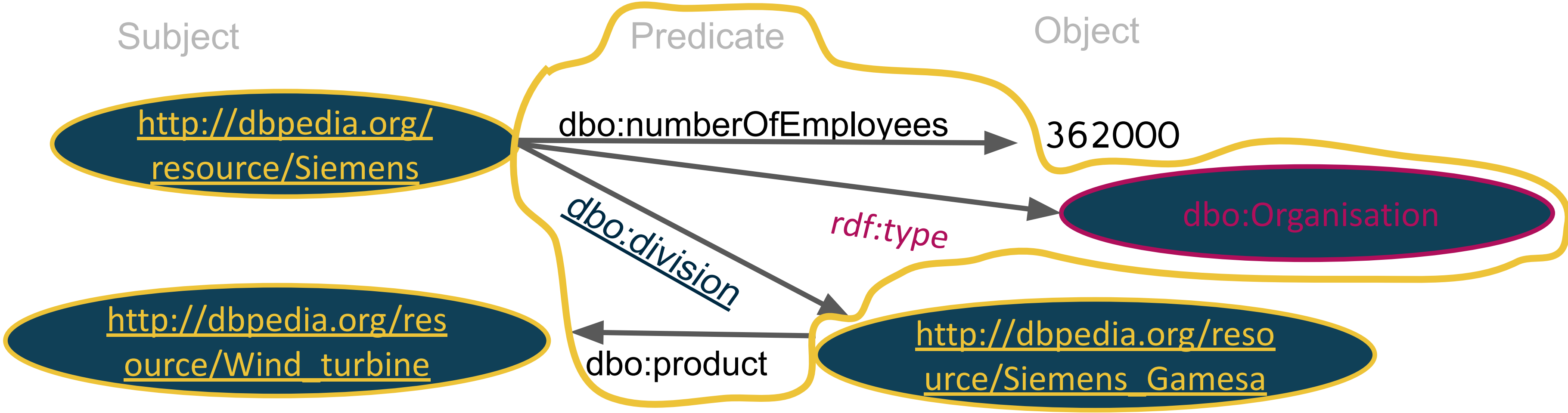
DBpedia Archivo in a nutshell

- **Augmented Ontology Archive** for Improving FAIRness of OWL Ontologies & SKOS concept schemes
- fully automated: discovery, versioning & testing for web-scale crawling
- unified + persistent access to ontology (meta)data >1800 Ontologies → the most exhaustive unified ontology space
- augmentation with different serialization formats, LODE docu, stats, reports
- 4-star rating and badges measure fitness for use fundamental FAIRness



Basics: LOD & Ontologies

- Ontologies provide identifier spaces for terms schema information for properties and classes of Linked Open Data (distributed KGs)
- “Common language”: ontologies reuse and specialize/generalize from existings terms in other ontologies → interoperability (matching on different granularity levels)


Subject
Predicate
Object
http://dbpedia.org/resource/Siemens
dbo:numberOfEmployees
362000
rdf:type
dbo:Organisation
dbo:division
http://dbpedia.org/resource/Siemens_Gamesa
dbo:product
http://dbpedia.org/resource/Wind_turbine

Motivation: Importance of Ontologies

Ontologies provide context crucial for interpretation and use of LOD

- Declaration of identifiers (what are valid properties / classes)
- Basic schema information (e.g. Object vs. Datatype property)
- Human readable semantics (e.g. label, comments, definition)
- Interoperability information to other ontologies (e.g. equivalentClass)
- **Formalizes impl. knowledge for machines (e.g. subClassOf / subPropertyOf)**

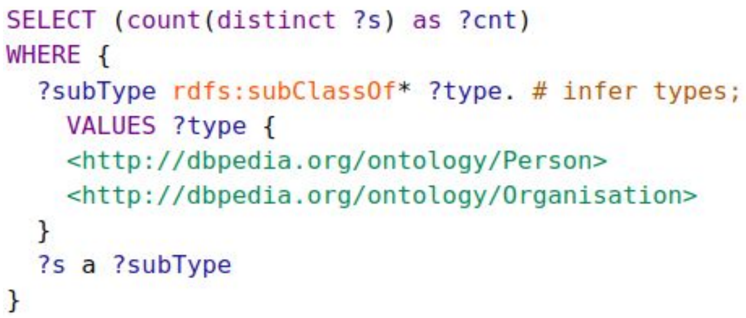
→ important artifact for reprod. experiments & workflows that are based on LOD

Motivation: Access to Ontologies affects Reprod.

Excerpt of rdfs:subClassOf (is-A) hierarchy DBpedia Ontology

```
∟   owl:Thing
      ∟   dbo:Person
            ∟   dbo:Scientist
                  ∟   dbo:Professor
```

Example analysis SPARQL query using ont.



```
SELECT (count(distinct ?s) as ?cnt)
WHERE {
  ?subType rdfs:subClassOf* ?type. # infer types;
    VALUES ?type {
    <http://dbpedia.org/ontology/Person>
    <http://dbpedia.org/ontology/Organisation>
  }
  ?s a ?subType
}
```

Example Knowledge Graph

```
:JF   a   dbo:Person
:ER   a   dbo:Professor
```

(FAIRness) Problems of Ontologies

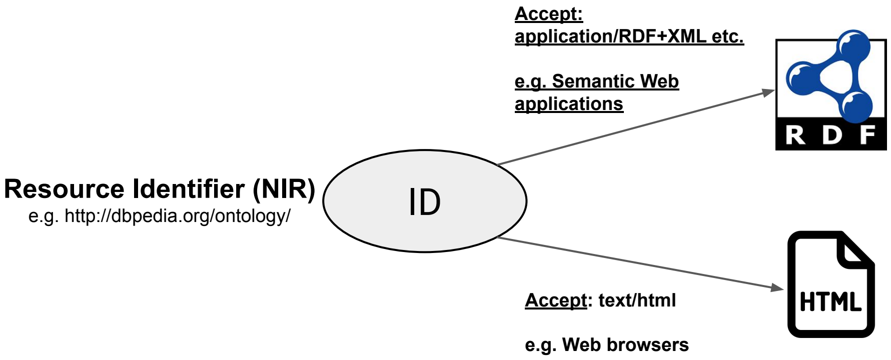
FAIR recursion problem (I2): FAIR (meta)data needs FAIR vocabularies & ontologies

Ontology Access Problems:

- Incorrect linked data deployment
- Unavailable/unresolvable ontologies

Ontology Interoperability + Reusability Problems:

- Missing / unclear / bad licensing
- File parsing errors / warnings
- Logical inconsistencies
- Bad/incomplete basic metadata /documentation



Resource Identifier (NIR)
e.g. http://dbpedia.org/ontology/

ID

Accept: application/RDF+XML etc.

e.g. Semantic Web applications

R D F

Accept: text/html

e.g. Web browsers

HTML

Findability

- no stable citation (ID) of a particular version of an ontology → missing terms due to evolution
- search for FAIR ontologies ??

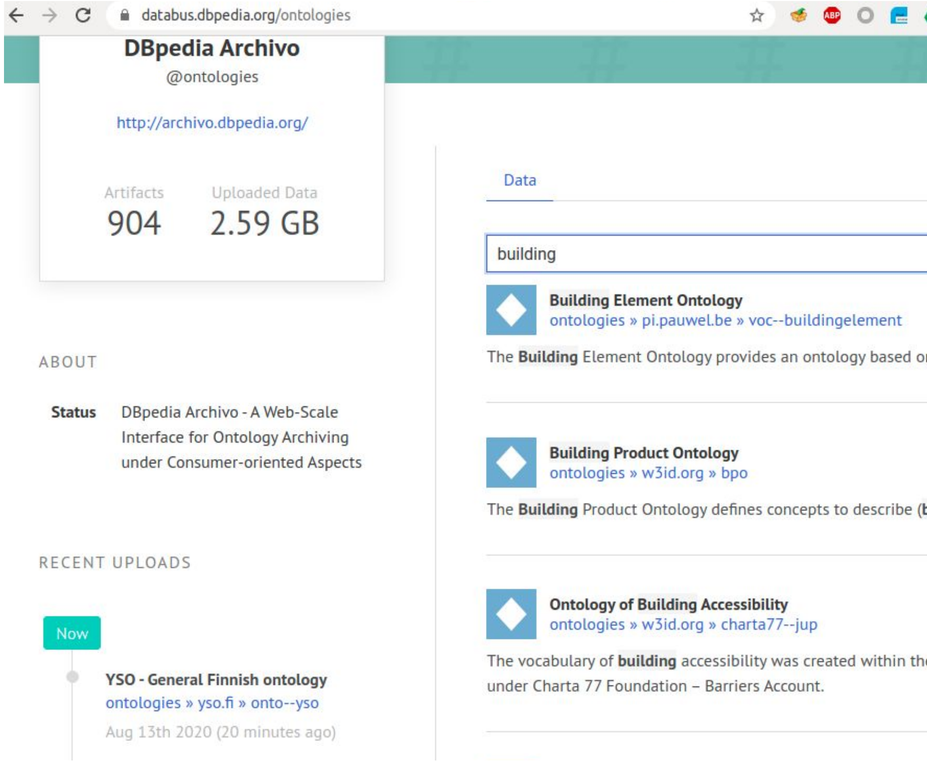
Archivo: An Ont. Interf. on DBpedia Databus

Solution: a web-scale ***Augmented Ontology Archive*** offering a ***unified interface*** for ontology consumption

Archivo is a dedicated publishing agent (user) on DBpedia Databus

→ persistent, unified versioning & archiving of ontologies based on Databus IDs

→ access archiving metadata via SPARQL & Linked Data



databus.dbpedia.org/ontologies

DBpedia Archivo
@ontologies

http://archivo.dbpedia.org/

Artifacts 904

Uploaded Data 2.59 GB

ABOUT

Status DBpedia Archivo - A Web-Scale Interface for Ontology Archiving under Consumer-oriented Aspects

RECENT UPLOADS

Now

YSO - General Finnish ontology
ontologies » yso.fi » onto--yso
Aug 13th 2020 (20 minutes ago)

Data

building

Building Element Ontology
ontologies » pi.pauwel.be » voc--buildingelement
The Building Element Ontology provides an ontology based o

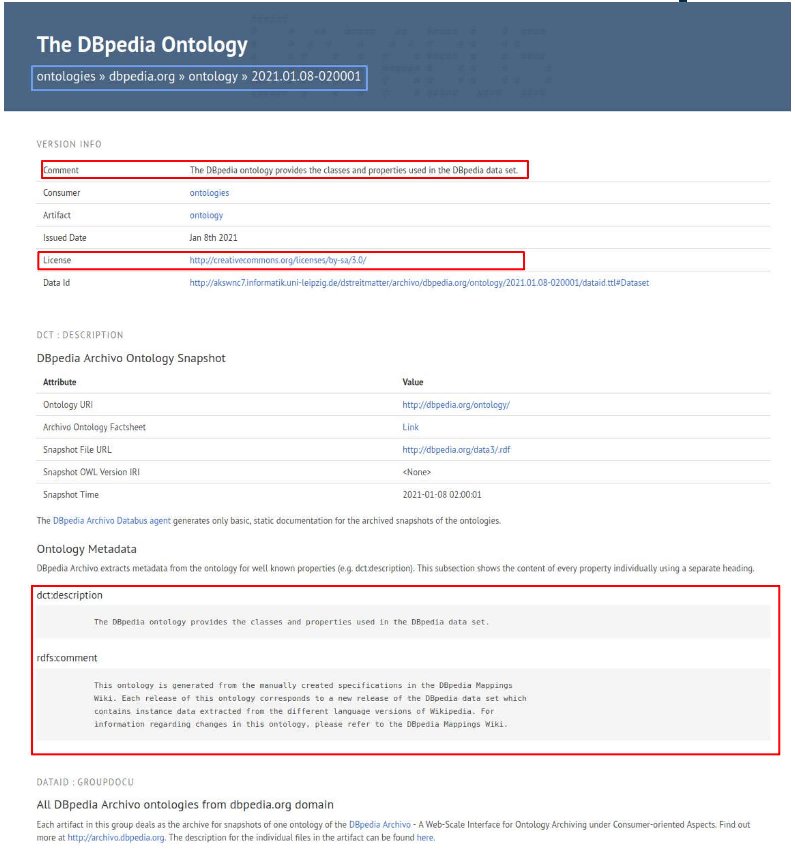
Building Product Ontology
ontologies » w3id.org » bpo
The Building Product Ontology defines concepts to describe (b

Ontology of Building Accessibility
ontologies » w3id.org » charta77--jup
The vocabulary of building accessibility was created within th under Charta 77 Foundation – Barriers Account.

Archivo Ontologies on the Databus

- Creates Databus IDs based on the ontology IRI for identification:
  - publisher → dedicated Databus agent “ontologies”
  - group → domain of the ontology
  - artifact → path of IRI
  - version → timestamp of discovery/update

- Maps multiple Ontology metadata properties to annotate ontologies on the Databus



The DBpedia Ontology

ontologies » dbpedia.org » ontology » 2021.01.08-020001

VERSION INFO

Comment	The DBpedia ontology provides the classes and properties used in the DBpedia data set.
Consumer	ontologies
Artifact	ontology
Issued Date	Jan 8th 2021
License	http://creativecommons.org/licenses/by-sa/3.0/
Data Id	http://akswnc7.informatik.uni-leipzig.de/dstreitmatter/archivo/dbpedia.org/ontology/2021.01.08-020001/dataid.ttl#Dataset

DCT : DESCRIPTION

DBpedia Archivo Ontology Snapshot

Attribute	Value
Ontology URI	http://dbpedia.org/ontology/
Archivo Ontology Factsheet	Link
Snapshot File URL	http://dbpedia.org/data3/.rdf
Snapshot OWL Version IRI	<None>
Snapshot Time	2021-01-08 02:00:01

The DBpedia Archivo Databus agent generates only basic, static documentation for the archived snapshots of the ontologies.

Ontology Metadata

DBpedia Archivo extracts metadata from the ontology for well known properties (e.g. dct:description). This subsection shows the content of every property individually using a separate heading.

dct:description

```
The DBpedia ontology provides the classes and properties used in the DBpedia data set.
```

rdfs:comment

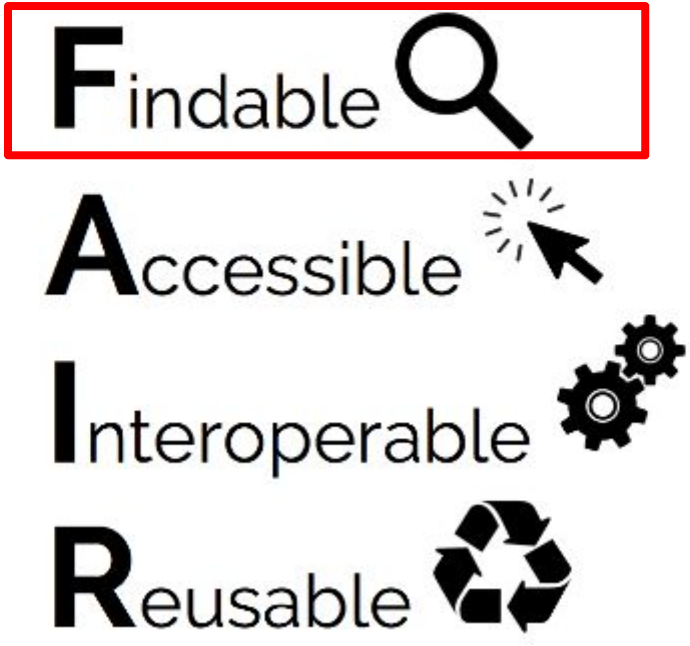
```
This ontology is generated from the manually created specifications in the DBpedia Mappings
Wiki. Each release of this ontology corresponds to a new release of the DBpedia data set which
contains instance data extracted from the different language versions of Wikipedia. For
information regarding changes in this ontology, please refer to the DBpedia Mappings Wiki.
```

DATAID : GROUPDOCU

All DBpedia Archivo ontologies from dbpedia.org domain

Each artifact in this group deals as the archive for snapshots of one ontology of the DBpedia Archivo - A Web-Scale Interface for Ontology Archiving under Consumer-oriented Aspects. Find out more at http://archivo.dbpedia.org. The description for the individual files in the artifact can be found here.

Finding an Ontology



Findable

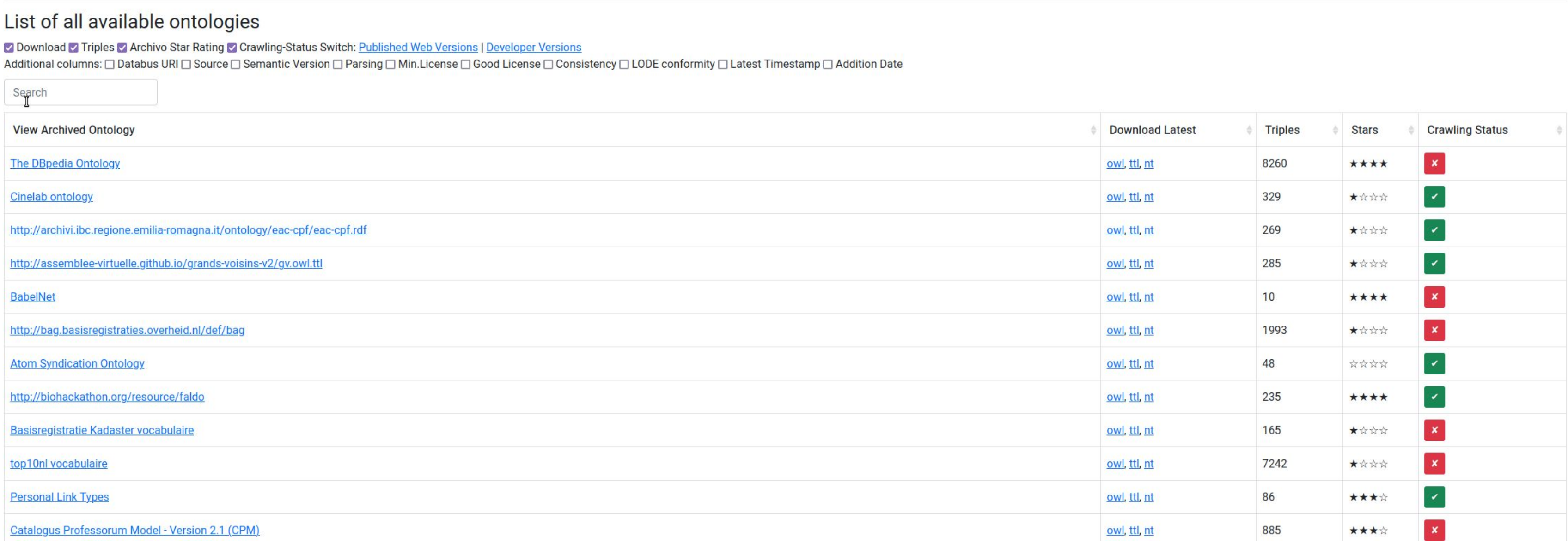
Accessible

Interoperable

Reusable

Finding an Ontology

- Search/Filter by Name and Sort via Ontology Overview Table on Archivo Web Frontend



List of all available ontologies

☑ Download ☑ Triples ☑ Archivo Star Rating ☑ Crawling-Status Switch: Published Web Versions | Developer Versions

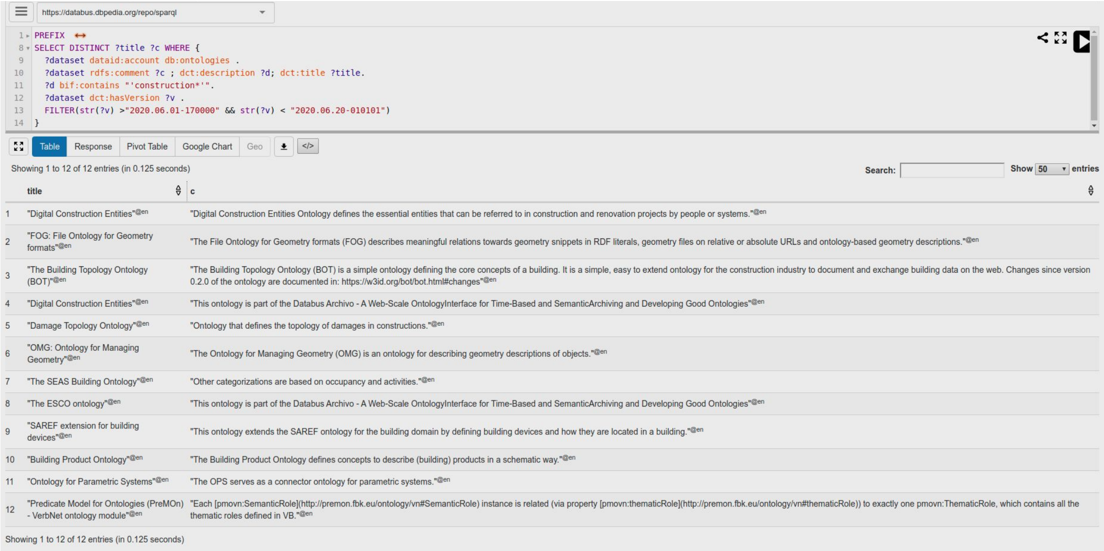
Additional columns: □ Databus URI □ Source □ Semantic Version □ Parsing □ Min.License □ Good License □ Consistency □ LODE conformity □ Latest Timestamp □ Addition Date

Search

View Archived Ontology	Download Latest	Triples	Stars	Crawling Status
The DBpedia Ontology	owl, ttl, nt	8260	★★★★	✗
Cinelab ontology	owl, ttl, nt	329	★☆☆☆	✓
http://archivi.ibc.regione.emilia-romagna.it/ontology/eac-cpf/eac-cpf.rdf	owl, ttl, nt	269	★☆☆☆	✓
http://assemblee-virtuelle.github.io/grands-voisins-v2/gv.owl.ttl	owl, ttl, nt	285	★☆☆☆	✓
BabelNet	owl, ttl, nt	10	★★★★	✗
http://bag.basisregistraties.overheid.nl/def/bag	owl, ttl, nt	1993	★☆☆☆	✗
Atom Syndication Ontology	owl, ttl, nt	48	☆☆☆☆	✓
http://biohackathon.org/resource/faldo	owl, ttl, nt	235	★★★★	✓
Basisregistratie Kadaster vocabulaire	owl, ttl, nt	165	★☆☆☆	✗
top10nl vocabulaire	owl, ttl, nt	7242	★☆☆☆	✗
Personal Link Types	owl, ttl, nt	86	★★★☆	✓
Catalogus Professorum Model - Version 2.1 (CPM)	owl, ttl, nt	885	★★★☆	✗

Finding an Ontology

- search on Title or Core Ontology Metadata: via Databus SPARQL, or via Databus Search



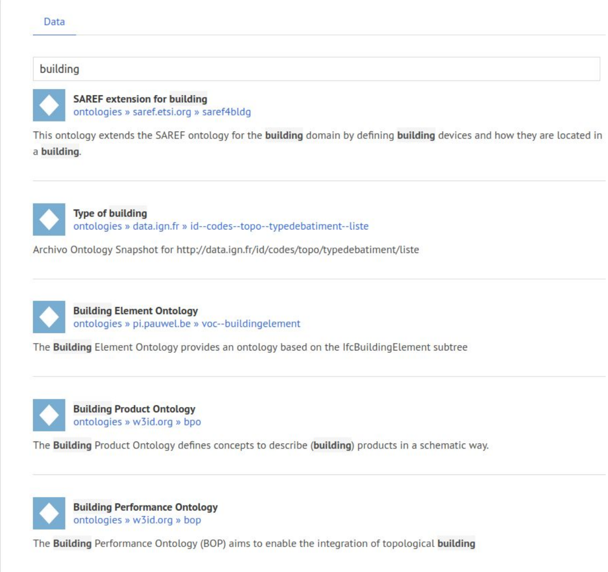
```sparql
PREFIX ↔
SELECT DISTINCT ?title ?c WHERE {
  ?dataset dataid:account db:ontologies .
  ?dataset rdfs:comment ?c ; dct:description ?d; dct:title ?title.
  ?d bif:contains "'construction*'".
  ?dataset dct:hasVersion ?v .
  FILTER(str(?v) >"2020.06.01-170000" && str(?v) < "2020.06.20-010101")
}
```

Showing 1 to 12 of 12 entries (in 0.125 seconds)

Search: Show 50 entries

	title	c
1	"Digital Construction Entities"@en	"Digital Construction Entities Ontology defines the essential entities that can be referred to in construction and renovation projects by people or systems."@en
2	"FOG: File Ontology for Geometry formats"@en	"The File Ontology for Geometry formats (FOG) describes meaningful relations towards geometry snippets in RDF literals, geometry files on relative or absolute URLs and ontology-based geometry descriptions."@en
3	"The Building Topology Ontology (BOT)"@en	"The Building Topology Ontology (BOT) is a simple ontology defining the core concepts of a building. It is a simple, easy to extend ontology for the construction industry to document and exchange building data on the web. Changes since version 0.2.0 of the ontology are documented in: https://w3id.org/bot/bot.html#changes"@en
4	"Digital Construction Entities"@en	"This ontology is part of the Databus Archivo - A Web-Scale OntologyInterface for Time-Based and SemanticArchiving and Developing Good Ontologies"@en
5	"Damage Topology Ontology"@en	"Ontology that defines the topology of damages in constructions."@en
6	"OMG: Ontology for Managing Geometry"@en	"The Ontology for Managing Geometry (OMG) is an ontology for describing geometry descriptions of objects."@en
7	"The SEAS Building Ontology"@en	"Other categorizations are based on occupancy and activities."@en
8	"The ESCO ontology"@en	"This ontology is part of the Databus Archivo - A Web-Scale OntologyInterface for Time-Based and SemanticArchiving and Developing Good Ontologies"@en
9	"SAREF extension for building devices"@en	"This ontology extends the SAREF ontology for the building domain by defining building devices and how they are located in a building."@en
10	"Building Product Ontology"@en	"The Building Product Ontology defines concepts to describe (building) products in a schematic way."@en
11	"Ontology for Parametric Systems"@en	"The OPS serves as a connector ontology for parametric systems."@en
12	"Predicate Model for Ontologies (PreMOn) - VerbNet ontology module"@en	"Each [pmovn:SemanticRole](http://premon.fbk.eu/ontology/vn#SemanticRole) instance is related (via property [pmovn:thematicRole](http://premon.fbk.eu/ontology/vn#thematicRole)) to exactly one pmovn:ThematicRole, which contains all the thematic roles defined in VB."@en

Showing 1 to 12 of 12 entries (in 0.125 seconds)



building

SAREF extension for building
ontologies » saref.etsi.org » saref4bldg

This ontology extends the SAREF ontology for the **building** domain by defining **building** devices and how they are located in a **building**.

Type of building
ontologies » data.ign.fr » id--codes--topo--typedebatiment--liste

Archivo Ontology Snapshot for http://data.ign.fr/id/codes/topo/typedebatiment/liste

Building Element Ontology
ontologies » pi.pauwel.be » voc--buildingelement

The **Building** Element Ontology provides an ontology based on the IfcBuildingElement subtree

Building Product Ontology
ontologies » w3id.org » bpo

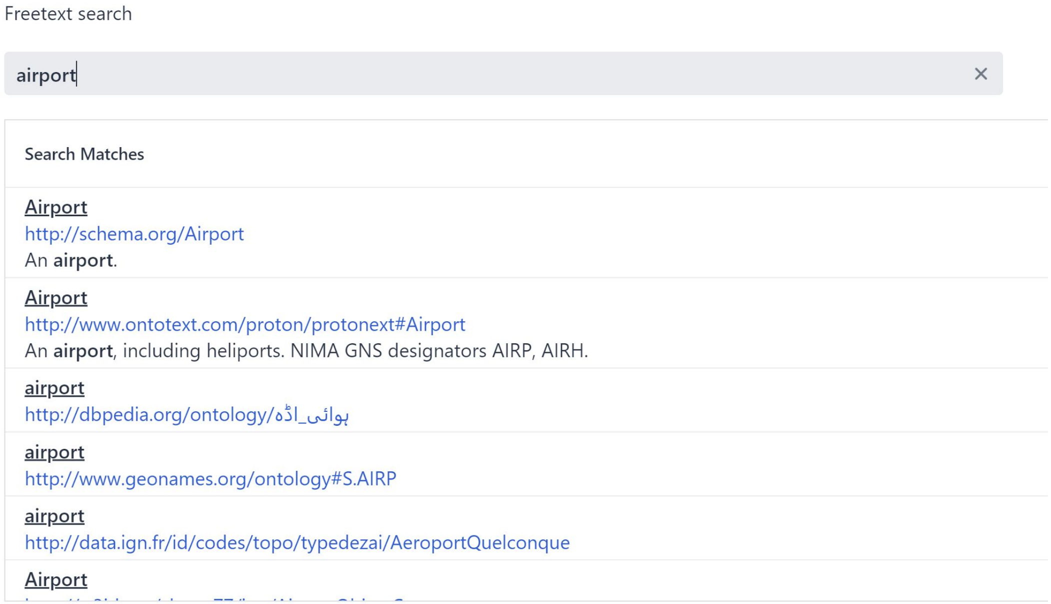
The **Building** Product Ontology defines concepts to describe (**building**) products in a schematic way.

Building Performance Ontology
ontologies » w3id.org » bop

The **Building** Performance Ontology (BOP) aims to enable the integration of topological **building**

Searching for Terms from Ontologies (A)

Ranked Fuzzy Term Search via (Lucene) Index powered by DBpedia Lookup indexing configuration (public service alpha)



Freetext search

airport ✕

Search Matches

Airport
http://schema.org/Airport
An **airport**.

Airport
http://www.ontotext.com/proton/protonext#Airport
An **airport**, including heliports. NIMA GNS designators AIRP, AIRH.

airport
http://dbpedia.org/ontology/ہوائی_اڈہ

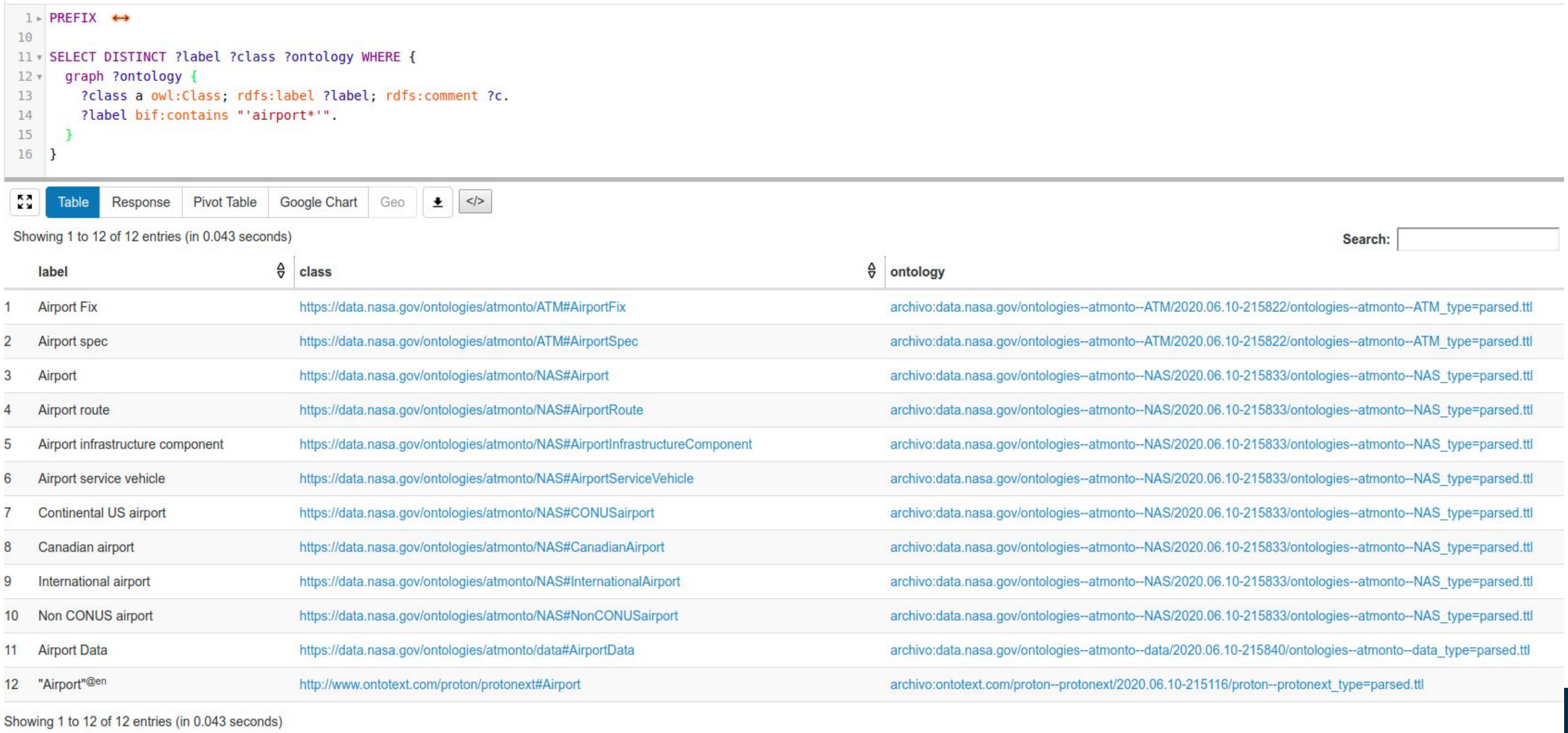
airport
http://www.geonames.org/ontology#S.AIRP

airport
http://data.ign.fr/id/codes/topo/typedezai/AeroportQuelconque

Airport

Searching for Terms from Ontologies (B)

Term Search via (self-hosted) SPARQL Index on full Ontology Content



```
PREFIX ↔

SELECT DISTINCT ?label ?class ?ontology WHERE {
  graph ?ontology {
    ?class a owl:Class; rdfs:label ?label; rdfs:comment ?c.
    ?label bif:contains "'airport*'".
  }
}
```

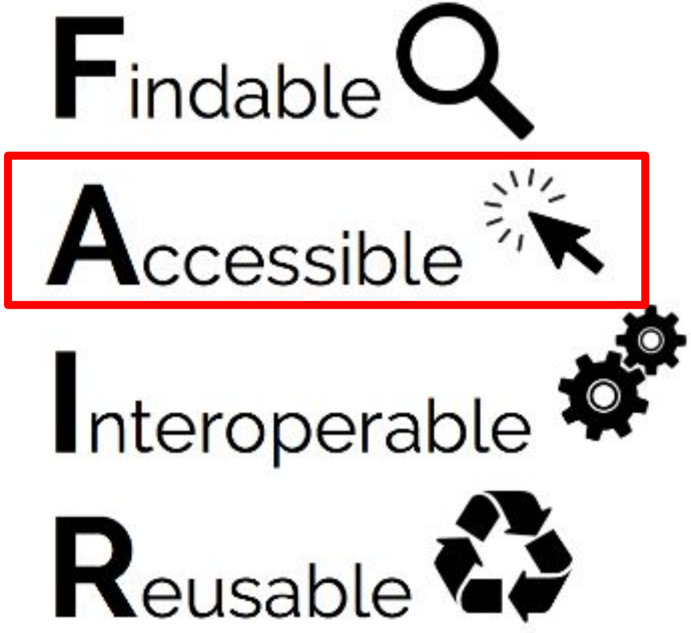
Showing 1 to 12 of 12 entries (in 0.043 seconds)

Search:

	label	class	ontology
1	Airport Fix	https://data.nasa.gov/ontologies/atmonto/ATM#AirportFix	archivo:data.nasa.gov/ontologies--atmonto--ATM/2020.06.10-215822/ontologies--atmonto--ATM_type=parsed.ttl
2	Airport spec	https://data.nasa.gov/ontologies/atmonto/ATM#AirportSpec	archivo:data.nasa.gov/ontologies--atmonto--ATM/2020.06.10-215822/ontologies--atmonto--ATM_type=parsed.ttl
3	Airport	https://data.nasa.gov/ontologies/atmonto/NAS#Airport	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
4	Airport route	https://data.nasa.gov/ontologies/atmonto/NAS#AirportRoute	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
5	Airport infrastructure component	https://data.nasa.gov/ontologies/atmonto/NAS#AirportInfrastructureComponent	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
6	Airport service vehicle	https://data.nasa.gov/ontologies/atmonto/NAS#AirportServiceVehicle	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
7	Continental US airport	https://data.nasa.gov/ontologies/atmonto/NAS#CONUSairport	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
8	Canadian airport	https://data.nasa.gov/ontologies/atmonto/NAS#CanadianAirport	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
9	International airport	https://data.nasa.gov/ontologies/atmonto/NAS#InternationalAirport	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
10	Non CONUS airport	https://data.nasa.gov/ontologies/atmonto/NAS#NonCONUSairport	archivo:data.nasa.gov/ontologies--atmonto--NAS/2020.06.10-215833/ontologies--atmonto--NAS_type=parsed.ttl
11	Airport Data	https://data.nasa.gov/ontologies/atmonto/data#AirportData	archivo:data.nasa.gov/ontologies--atmonto--data/2020.06.10-215840/ontologies--atmonto--data_type=parsed.ttl
12	"Airport"@en	http://www.ontotext.com/proton/protonext#Airport	archivo:ontotext.com/proton--protonext/2020.06.10-215116/proton--protonext_type=parsed.ttl

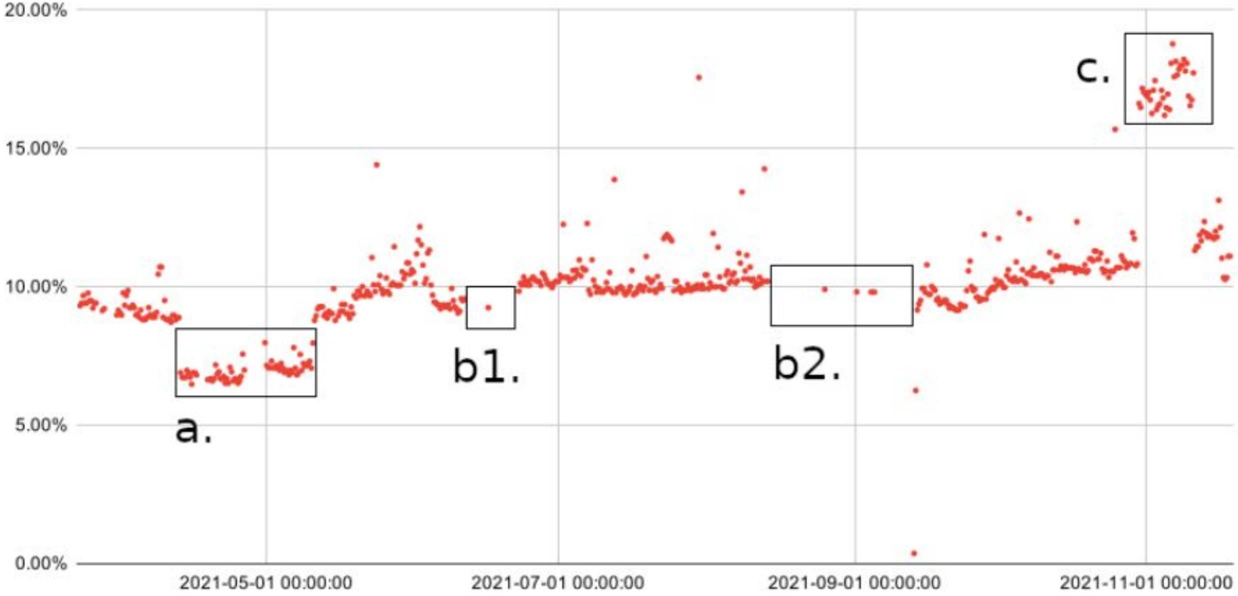
Showing 1 to 12 of 12 entries (in 0.043 seconds)

Accessing Ontologies



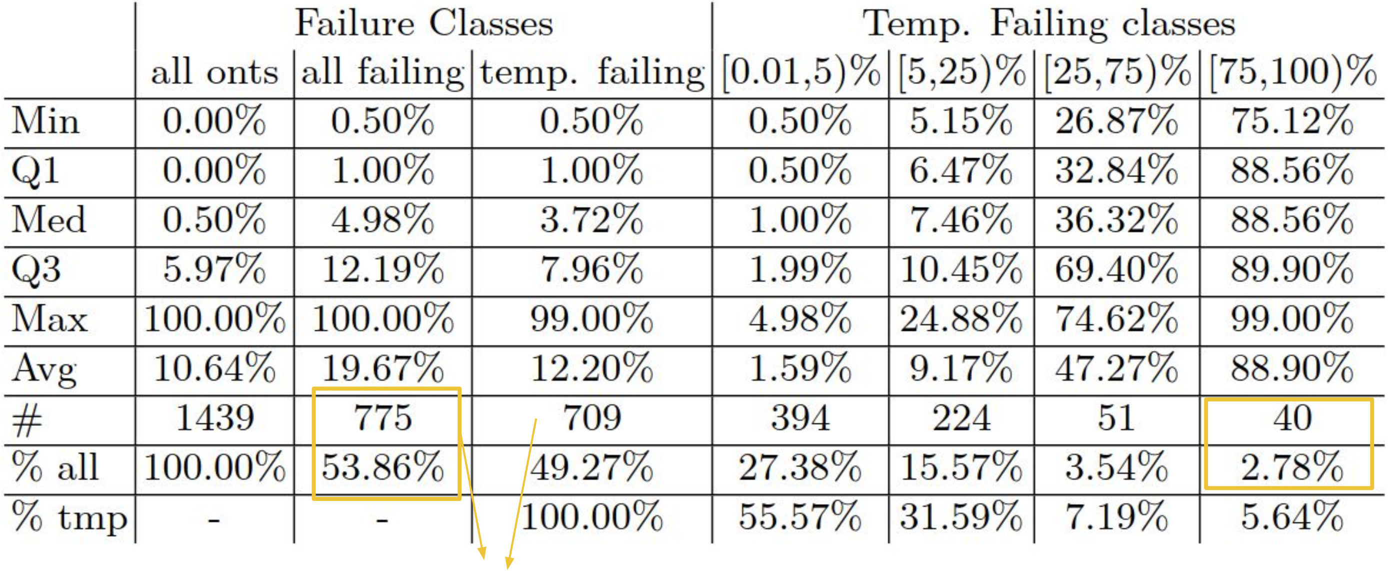
Ontology Accessibility Failures

Fraction of ontologies that failed in a crawling window normalized by the number of ontologies archived at that time



Acc. Failure Duration Classes

normalized no. of failed crawls by no. of all crawl attempts for that ontology

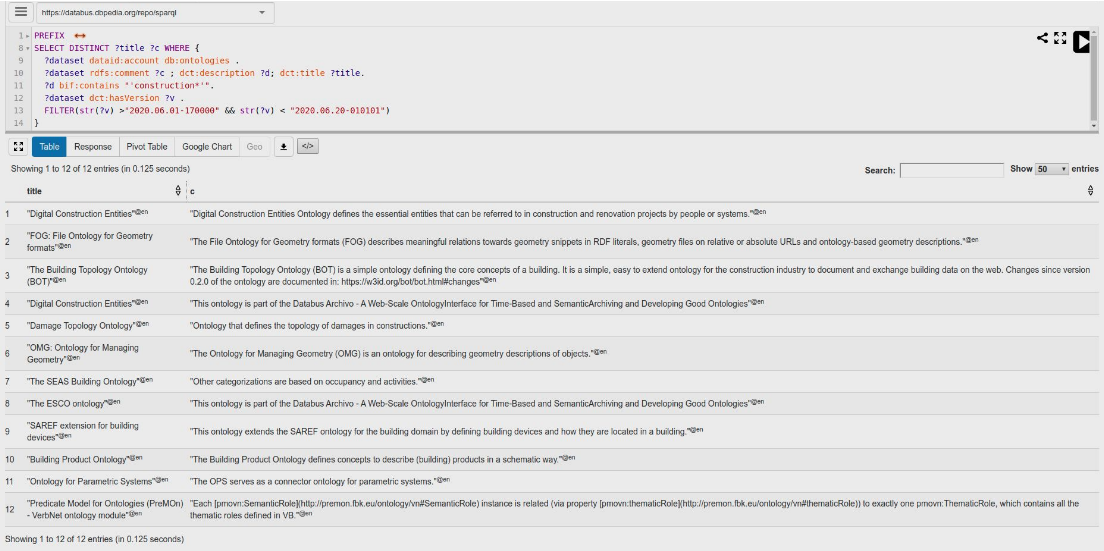


	Failure Classes			Temp. Failing classes			
	all onts	all failing	temp. failing	[0.01,5)%	[5,25)%	[25,75)%	[75,100)%
Min	0.00%	0.50%	0.50%	0.50%	5.15%	26.87%	75.12%
Q1	0.00%	1.00%	1.00%	0.50%	6.47%	32.84%	88.56%
Med	0.50%	4.98%	3.72%	1.00%	7.46%	36.32%	88.56%
Q3	5.97%	12.19%	7.96%	1.99%	10.45%	69.40%	89.90%
Max	100.00%	100.00%	99.00%	4.98%	24.88%	74.62%	99.00%
Avg	10.64%	19.67%	12.20%	1.59%	9.17%	47.27%	88.90%
#	1439	775	709	394	224	51	40
% all	100.00%	53.86%	49.27%	27.38%	15.57%	3.54%	2.78%
% tmp	-	-	100.00%	55.57%	31.59%	7.19%	5.64%

→ 66 (4.6%) perm. failing

Finding an Ontology

- search on Title or Ontology Metadata: via Databus SPARQL, or via Databus Databus Search



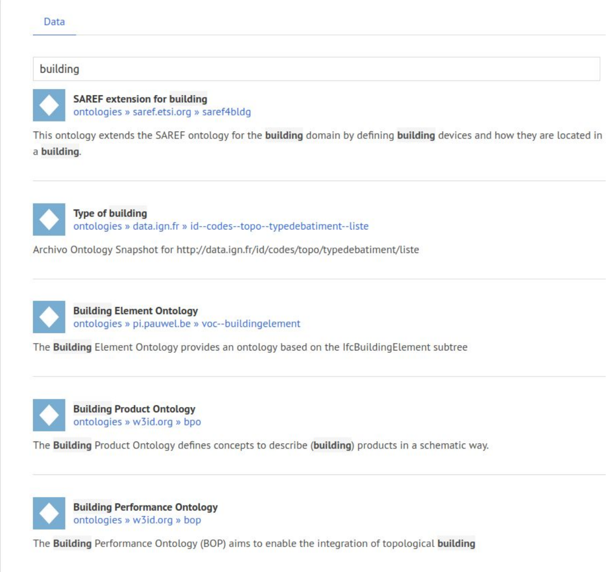
```
PREFIX  ↔
SELECT DISTINCT ?title ?c WHERE {
  ?dataset dataid:account db:ontologies .
  ?dataset rdfs:comment ?c ; dct:description ?d; dct:title ?title.
  ?d bif:contains "'construction*'".
  ?dataset dct:hasVersion ?v .
  FILTER(str(?v) >"2020.06.01-170000" && str(?v) < "2020.06.20-010101")
}
```

Showing 1 to 12 of 12 entries (in 0.125 seconds)

Search: Show 50 entries

	title	c
1	"Digital Construction Entities"@en	"Digital Construction Entities Ontology defines the essential entities that can be referred to in construction and renovation projects by people or systems."@en
2	"FOG: File Ontology for Geometry formats"@en	"The File Ontology for Geometry formats (FOG) describes meaningful relations towards geometry snippets in RDF literals, geometry files on relative or absolute URLs and ontology-based geometry descriptions."@en
3	"The Building Topology Ontology (BOT)"@en	"The Building Topology Ontology (BOT) is a simple ontology defining the core concepts of a building. It is a simple, easy to extend ontology for the construction industry to document and exchange building data on the web. Changes since version 0.2.0 of the ontology are documented in: https://w3id.org/bot/bot.html#changes"@en
4	"Digital Construction Entities"@en	"This ontology is part of the Databus Archivo - A Web-Scale OntologyInterface for Time-Based and SemanticArchiving and Developing Good Ontologies"@en
5	"Damage Topology Ontology"@en	"Ontology that defines the topology of damages in constructions."@en
6	"OMG: Ontology for Managing Geometry"@en	"The Ontology for Managing Geometry (OMG) is an ontology for describing geometry descriptions of objects."@en
7	"The SEAS Building Ontology"@en	"Other categorizations are based on occupancy and activities."@en
8	"The ESCO ontology"@en	"This ontology is part of the Databus Archivo - A Web-Scale OntologyInterface for Time-Based and SemanticArchiving and Developing Good Ontologies"@en
9	"SAREF extension for building devices"@en	"This ontology extends the SAREF ontology for the building domain by defining building devices and how they are located in a building."@en
10	"Building Product Ontology"@en	"The Building Product Ontology defines concepts to describe (building) products in a schematic way."@en
11	"Ontology for Parametric Systems"@en	"The OPS serves as a connector ontology for parametric systems."@en
12	"Predicate Model for Ontologies (PreMOn) - VerbNet ontology module"@en	"Each [pmovn:SemanticRole](http://premon.fbk.eu/ontology/vn#SemanticRole) instance is related (via property [pmovn:thematicRole](http://premon.fbk.eu/ontology/vn#thematicRole)) to exactly one pmovn:ThematicRole, which contains all the thematic roles defined in VB."@en

Showing 1 to 12 of 12 entries (in 0.125 seconds)



building

SAREF extension for building
ontologies » saref.etsi.org » saref4bldg

This ontology extends the SAREF ontology for the **building** domain by defining **building** devices and how they are located in a **building**.

Type of building
ontologies » data.ign.fr » id--codes--topo--typedebatiment--liste

Archivo Ontology Snapshot for http://data.ign.fr/id/codes/topo/typedebatiment/liste

Building Element Ontology
ontologies » pi.pauwel.be » voc--buildingelement

The **Building** Element Ontology provides an ontology based on the IfcBuildingElement subtree

Building Product Ontology
ontologies » w3id.org » bpo

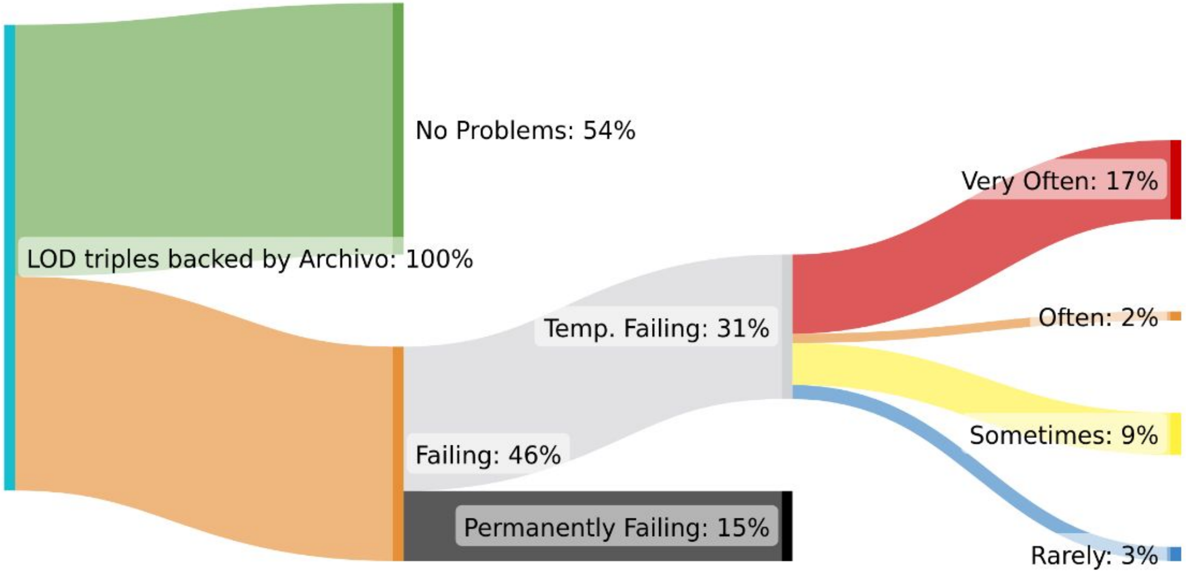
The **Building** Product Ontology defines concepts to describe (**building**) products in a schematic way.

Building Performance Ontology
ontologies » w3id.org » bop

The **Building** Performance Ontology (BOP) aims to enable the integration of topological **building**

Archivo Impact: Breakdown for backed triples

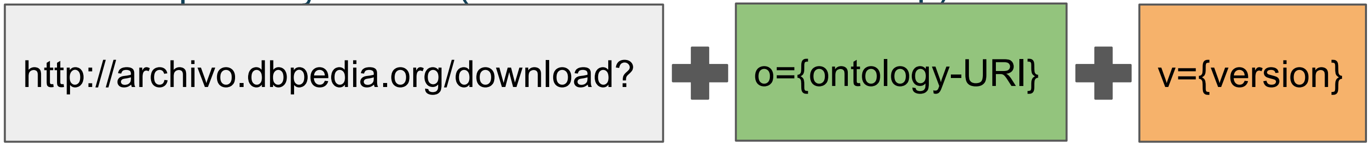
Fraction of LOD-a-lot triples covered by Archivo, categorized based on the accessibility class of the ontology that defines the term



Accessing 1 Ontology via Archivo API

- Access to all persisted snapshots of any ontology version regardless their current accessibility
- More robust and failure-tolerant
  - parsed versions based on recoverable best-effort crawling circumventing several deployment bugs

- One REST request:
  - requires Ontology NIR
  - Optionally version (defaults to latest timestamp)



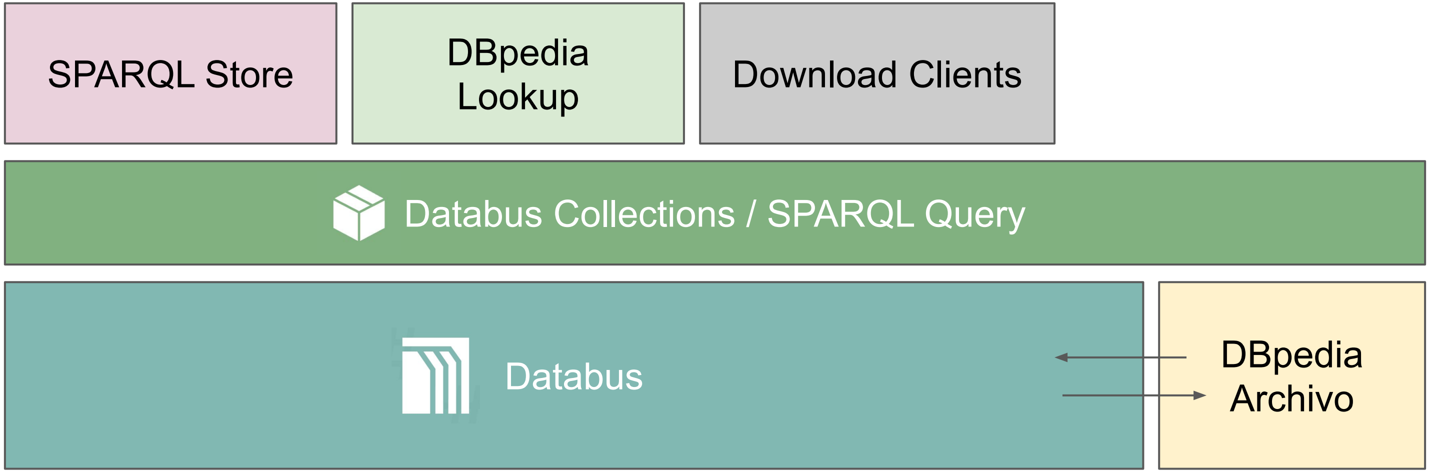
http://archivo.dbpedia.org/download? + o={ontology-URI} + v={version}

e.g. http://archivo.dbpedia.org/download?o=http://datashapes.org/dash&v=2020.07.16-115638

[1]more at https://archivo.dbpedia.org/api

Access via Databus Technology Stack

- Several self-hosted “one-click” deployment services of DBpedia Tech stack can be used with DBpedia Archivo
- Ontologies can be fed into application via Collection IDs (custom or official ones)



Tech Stack: Load Ont. into SPARQL endpoint

1. Create or select an existing Collection with Archivo ontologies
2. “One-click-load” the Collection in a local SPARQL endpoint

```
git clone https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart.git
cd virtuoso-sparql-endpoint-quickstart
COLLECTION_URI=https://databus.dbpedia.org/denis/collections/latest_ontologies_as
_nt_sample VIRTUOSO_ADMIN_PASSWD=secret docker-compose up
```

Useful collections:

- latest parsed ont. as turtle files: https://databus.dbpedia.org/jfrey/collections/archivo-latest-ontology-snapshots

Easy Download of Collections from Code

DBpedia

Accessing all (or variable subset of) Ontologies:

- Use or create collection (or custom SPARQL query)
- Execute query via HTTP
- Loop over Databus file IDs and fetch them

```
query=$(curl -H "Accept:text/sparql" https://databus.dbpedia.org/denis/collections/latest_ontologies_as_nt)
files=$(curl -H "Accept: text/csv" --data-urlencode "query=${query}" https://databus.dbpedia.org/sparql | tail -n+2 | sed 's/"//g')
while IFS= read -r file ; do wget $file; done <<< "$files"
```

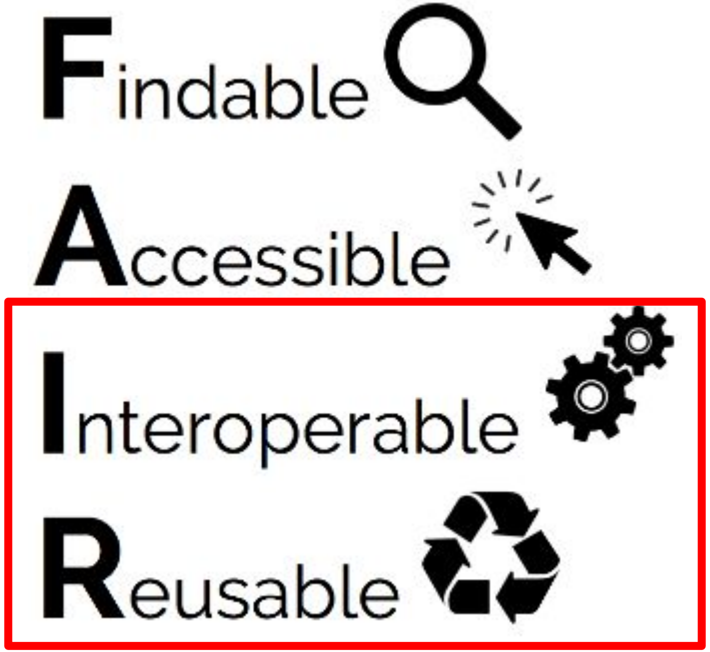
Future Work GSOC24 proposal

DBpedia

Goal: "Plug-and-play" Linked Data Resolution of Ontology (Terms) in a deterministic/controllable way based on ontology snapshots hosted in Archivo

- (transparent) ontology time machine proxy
  - time-based mode: serve versions archived for a certain point in time
  - dependent-lock based mode: serve specific versions based on a local manifest or manifest in (transitively) included ontologies
  - failover mode: redirect to the latest archived version in the event that an ontology is not available anymore
- realize "dependency package manager" with lockfile option
- vocabulary to specify dependencies for ontology publishers

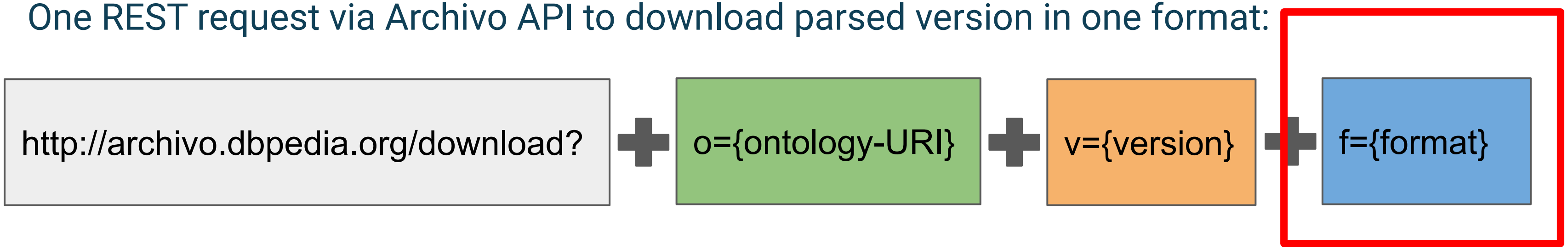
Interoperable & Reusable Ontologies



Ontology Format Interoperability

- Common parsed serialisations: RDF+XML, Turtle and N-Triples on Databus and API

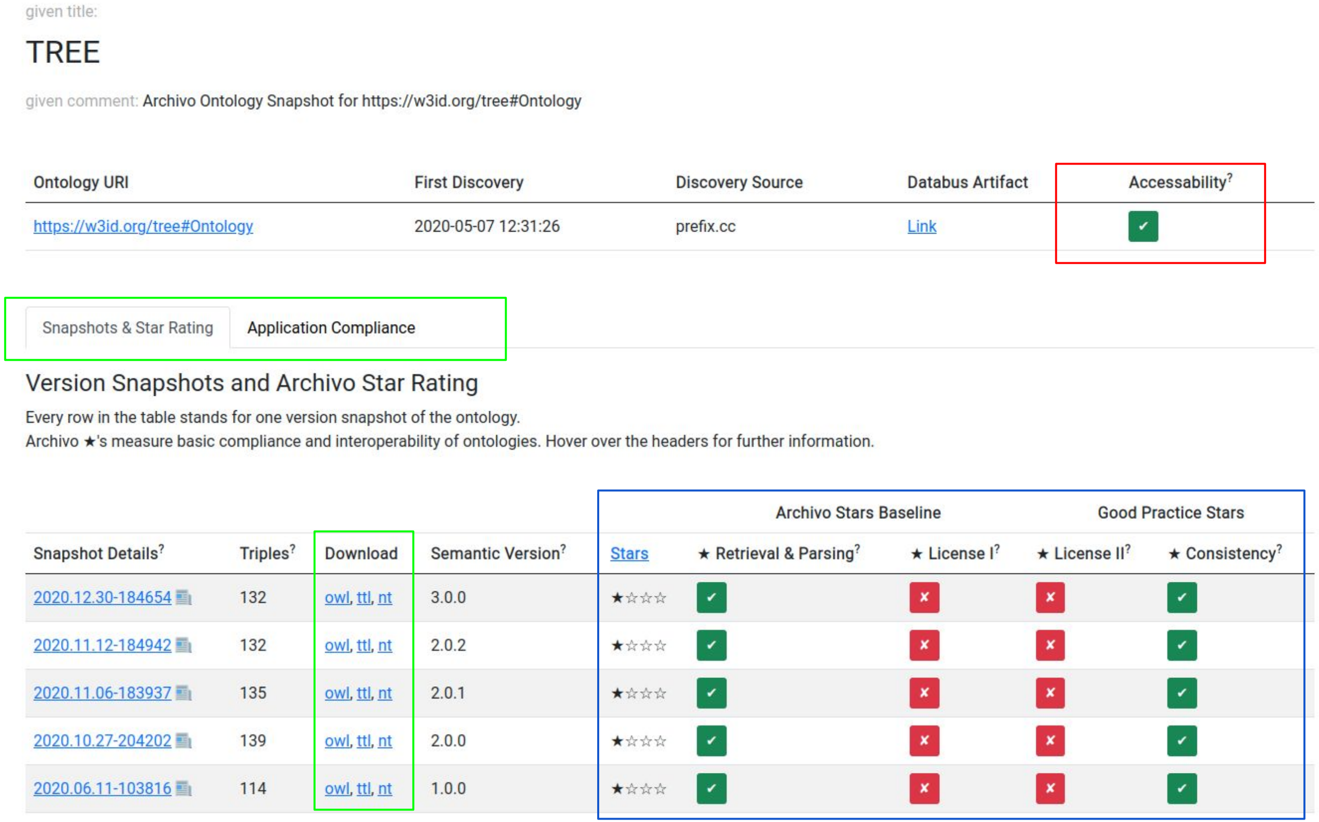
One REST request via Archivo API to download parsed version in one format:



http://archivo.dbpedia.org/download? + o={ontology-URI} + v={version} + f={format}

e.g. https://archivo.dbpedia.org/download?o=http%3A//www.georss.org/georss/&v=2020.08.10-110000&f=ttl

Measuring and Improving I+R via Archivo UI



given title:

TREE

given comment: Archivo Ontology Snapshot for https://w3id.org/tree#Ontology

Ontology URI	First Discovery	Discovery Source	Databus Artifact	Accessability?
https://w3id.org/tree#Ontology	2020-05-07 12:31:26	prefix.cc	Link	✓

Version Snapshots and Archivo Star Rating

Every row in the table stands for one version snapshot of the ontology.

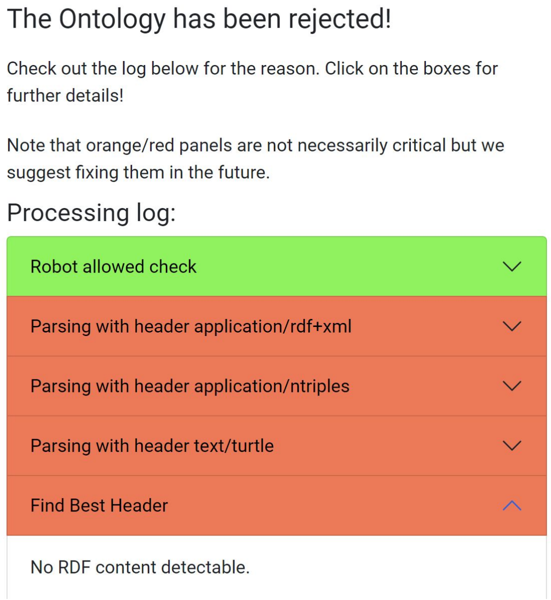
Archivo ★'s measure basic compliance and interoperability of ontologies. Hover over the headers for further information.

				Archivo Stars Baseline			Good Practice Stars	
Snapshot Details?	Triples?	Download	Semantic Version?	Stars	★ Retrieval & Parsing?	★ License I?	★ License II?	★ Consistency?
2020.12.30-184654	132	owl, ttl, nt	3.0.0	★☆☆☆	✓	✗	✗	✓
2020.11.12-184942	132	owl, ttl, nt	2.0.2	★☆☆☆	✓	✗	✗	✓
2020.11.06-183937	135	owl, ttl, nt	2.0.1	★☆☆☆	✓	✗	✗	✓
2020.10.27-204202	139	owl, ttl, nt	2.0.0	★☆☆☆	✓	✗	✗	✓
2020.06.11-103816	114	owl, ttl, nt	1.0.0	★☆☆☆	✓	✗	✗	✓

Debug common Ontology Pitfalls

Accessibility problems with correct ontology deployment:

- Testing / Reporting during the manual inclusion request of an ontology



The Ontology has been rejected!

Check out the log below for the reason. Click on the boxes for further details!

Note that orange/red panels are not necessarily critical but we suggest fixing them in the future.

Processing log:

🟩 Robot allowed check

🟥 Parsing with header application/rdf+xml

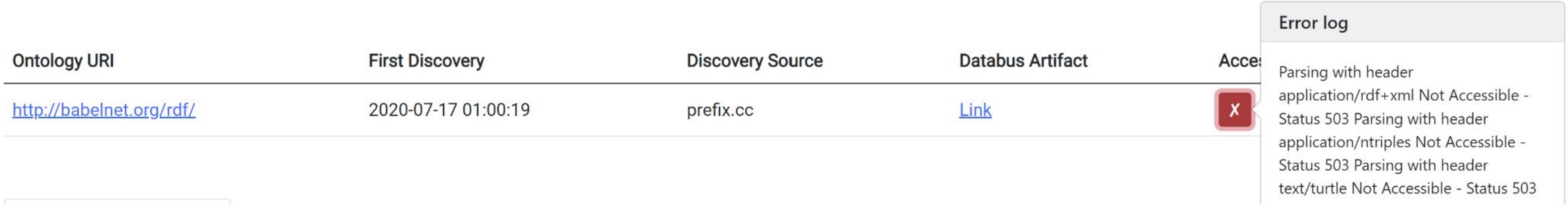
🟥 Parsing with header application/ntriples

🟥 Parsing with header text/turtle

🟥 Find Best Header

No RDF content detectable.

- or crawling update status



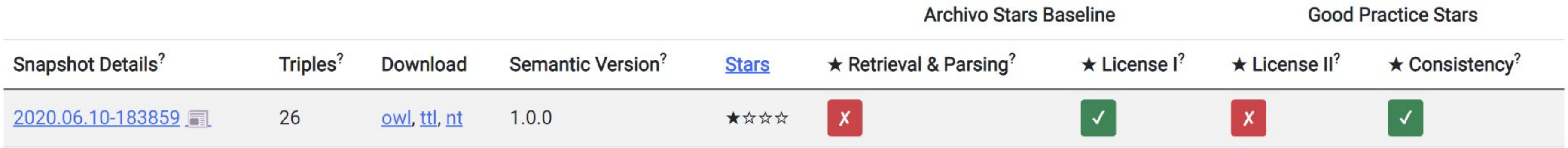
Ontology URI	First Discovery	Discovery Source	Databus Artifact	Acce
http://babelnet.org/rdf/	2020-07-17 01:00:19	prefix.cc	Link	✗

Error log

Parsing with header application/rdf+xml Not Accessible - Status 503 Parsing with header application/ntriples Not Accessible - Status 503 Parsing with header text/turtle Not Accessible - Status 503

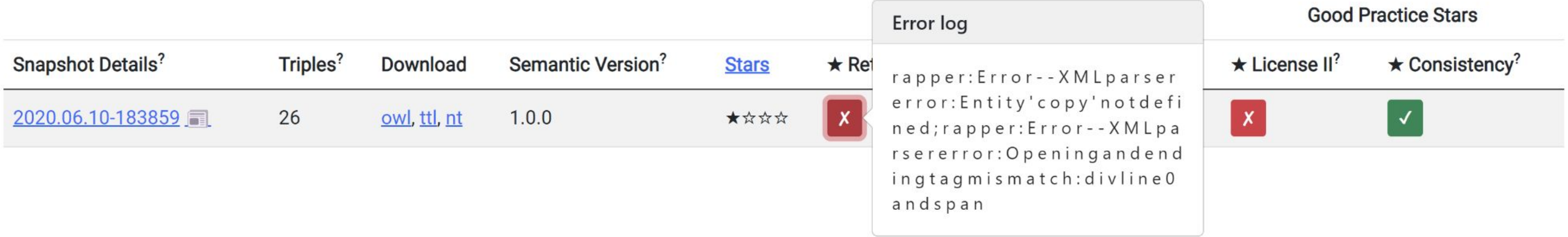
Debug common Ontology Pitfalls

- ontology star rating for (automated) (re)usability
  - testing parsing, license information and logical consistency



					Archivo Stars Baseline		Good Practice Stars	
Snapshot Details?	Triples?	Download	Semantic Version?	Stars	★ Retrieval & Parsing?	★ License I?	★ License II?	★ Consistency?
2020.06.10-183859	26	owl, ttl, nt	1.0.0	★☆☆☆	✗	✓	✗	✓

- Detailed error message on hover of error



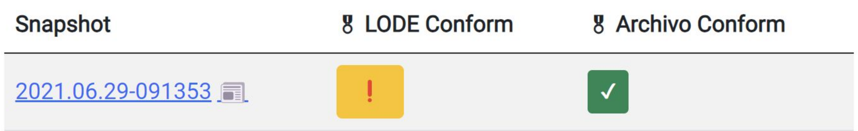
						Good Practice Stars	
Snapshot Details?	Triples?	Download	Semantic Version?	Stars	★ Ret	★ License II?	★ Consistency?
2020.06.10-183859	26	owl, ttl, nt	1.0.0	★☆☆☆	✗	✗	✓

Error log

```
rapper:Error--XMLparser
error:Entity'copy'notdefi
ned;rapper:Error--XMLpa
rsererror:Openingandend
ingtagmismatch:divline0
andspan
```

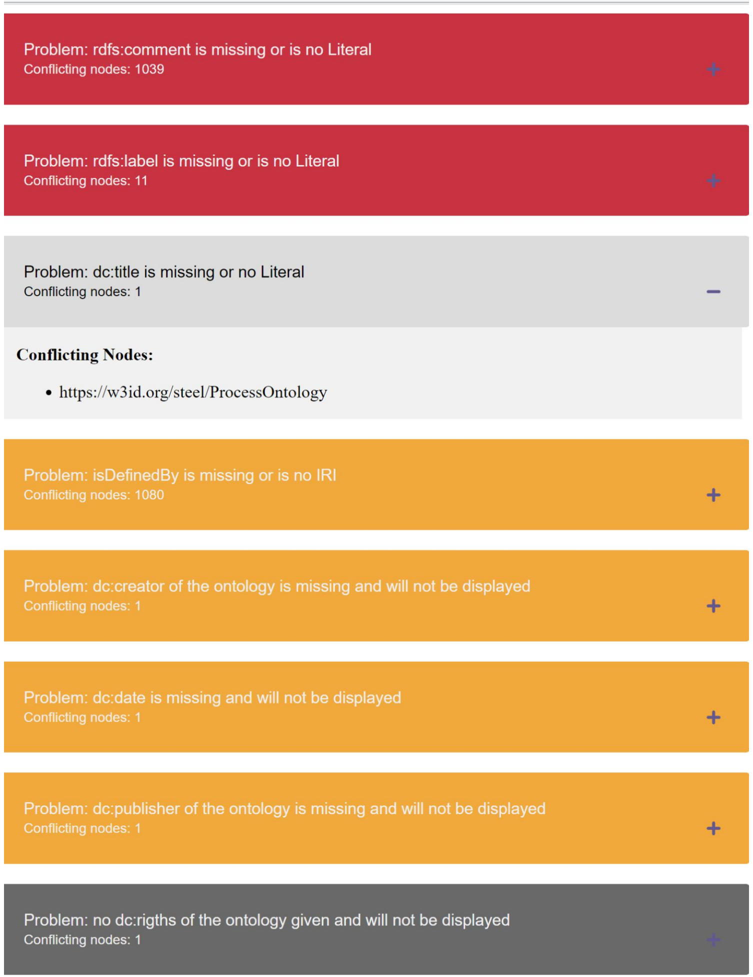
Analyze (Re)usability

- extensible SHACL library testing application compliance (fitness for use)
- Currently 2 badges
  - Metadata fitness for automatic LODE documentation
  - Metadata compliance to Archivo itself



Snapshot	🎖 LODE Conform	🎖 Archivo Conform
2021.06.29-091353	!	✓

- Detailed (machine readable!) reports of issues for each badge
- Categorized by issue classes
  - Severe / required info missing
  - optional but recommended / useful
  - Optional rather informative



Problem: rdfs:comment is missing or is no Literal
Conflicting nodes: 1039

Problem: rdfs:label is missing or is no Literal
Conflicting nodes: 11

Problem: dc:title is missing or no Literal
Conflicting nodes: 1

Conflicting Nodes:

- https://w3id.org/steel/ProcessOntology

Problem: isDefinedBy is missing or is no IRI
Conflicting nodes: 1080

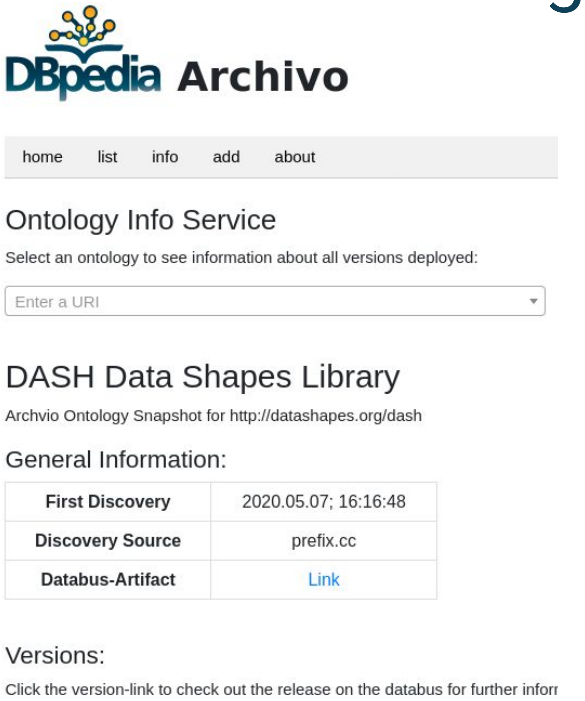
Problem: dc:creator of the ontology is missing and will not be displayed
Conflicting nodes: 1

Problem: dc:date is missing and will not be displayed
Conflicting nodes: 1

Problem: dc:publisher of the ontology is missing and will not be displayed
Conflicting nodes: 1

Problem: no dc:rigths of the ontology given and will not be displayed
Conflicting nodes: 1

Archivo: Ontology FAIRness Testing



DBpedia Archivo

home list info add about

Ontology Info Service

Select an ontology to see information about all versions deployed:

Enter a URI

DASH Data Shapes Library

Archvio Ontology Snapshot for http://datashapes.org/dash

General Information:

First Discovery	2020.05.07; 16:16:48
Discovery Source	prefix.cc
Databus-Artifact	Link

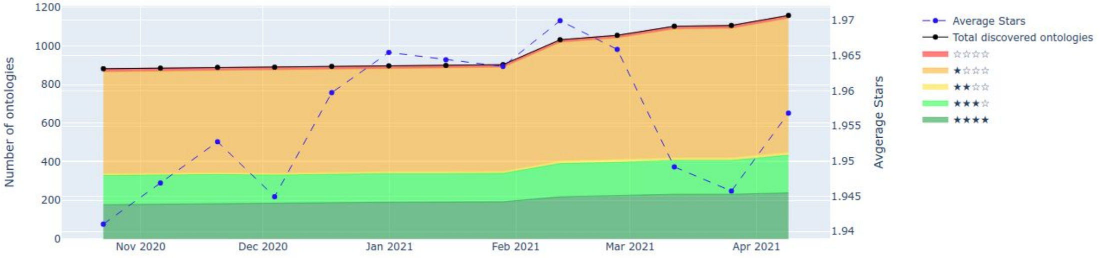
Versions:

Click the version-link to check out the release on the databus for further inforı

- Consumer-oriented star rating measuring aspects of FAIRness and fitness for automatic use
  - Linked Data retrieval & RDF parsing test
  - OWL API consistency test
  - extendible SHACL-based test library (e.g. checks for license statement, and basic relevant metadata)

Archivo Stars Distribution

At this moment Archivo contains **1205** Ontologies. See all available Ontologies



Version	Triples	Stars	Semantic Version	★ Retrieval & Parsing	★ License I	★ License II	★ Consistency	Lode-Conform
2020-07-16 11:56:03	1572	★☆☆☆	2.1.0	✓	✘	✘	✓	✘
2020-07-11 22:42:01	1551	★☆☆☆	2.0.0	✓	✘	✘	✓	✘
2020-06-10 18:13:02	1279	★☆☆☆	1.0.0	✓	✘	✘	✓	✘

Archivo Stars

Baseline:

- ⭐ ontology is retrievable without errors and parses
- ⭐ some kind of license can be found in metadata

Fitness for use stars:

- ⭐ license is given with dct:license and is an IRI
- ⭐ ontology is logically consistent

0x⭐ Ontology

- not parseable
- no license provided
- (maybe) logically inconsistent

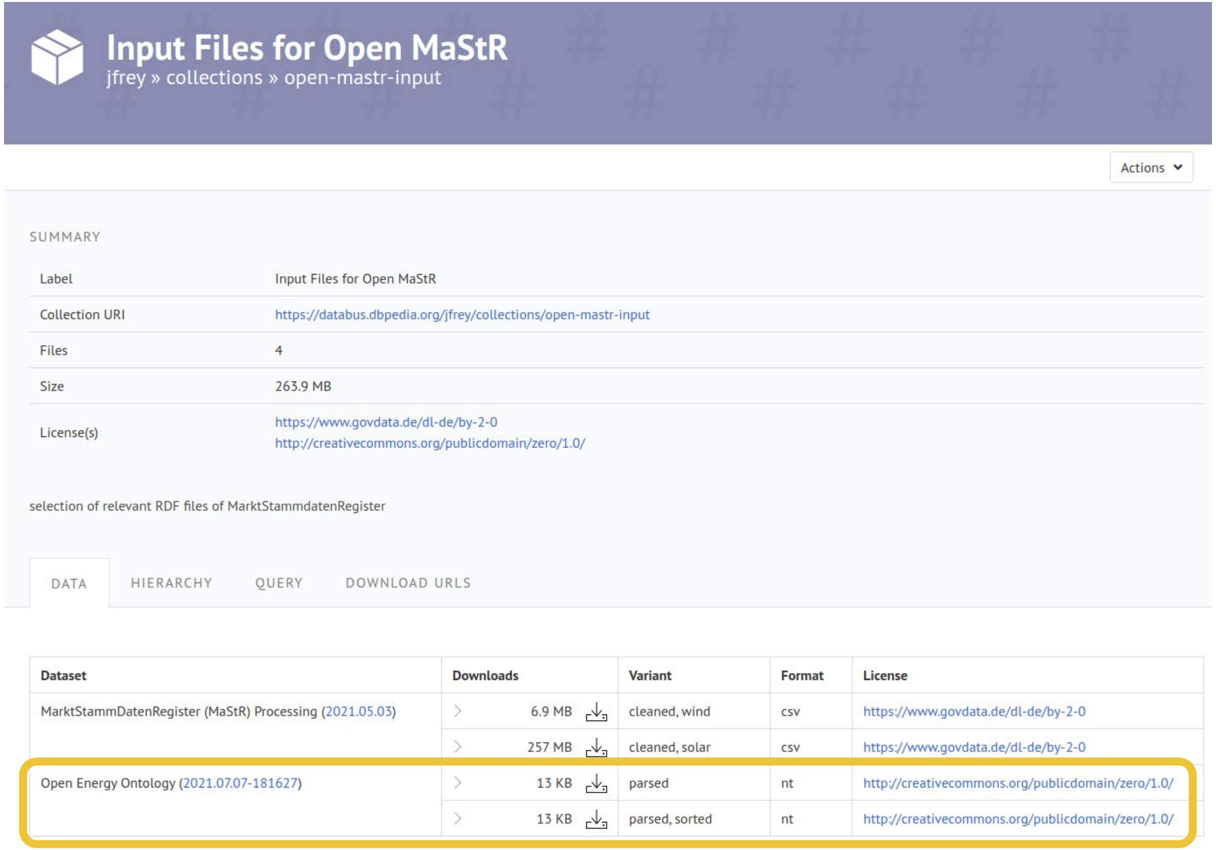
2x⭐ Ontology

- parseable & retrievable
- some license detected
- license only human readable or not unified
- logically inconsistent

4x⭐ Ontology

- parseable & retrievable
- unified license URI
- logically consistent

Archivo + Databus: Ontology Dependency/Citation



Input Files for Open MaStR

jfrey » collections » open-mastr-input

SUMMARY

Label	Input Files for Open MaStR
Collection URI	https://databus.dbpedia.org/jfrey/collections/open-mastr-input
Files	4
Size	263.9 MB
License(s)	https://www.govdata.de/dl-de/by-2-0 http://creativecommons.org/publicdomain/zero/1.0/

selection of relevant RDF files of MarktStammdatenRegister

Dataset	Downloads	Variant	Format	License
MarktStammDatenRegister (MaStR) Processing (2021.05.03)	6.9 MB	cleaned, wind	csv	https://www.govdata.de/dl-de/by-2-0
	257 MB	cleaned, solar	csv	https://www.govdata.de/dl-de/by-2-0
Open Energy Ontology (2021.07.07-181627)	13 KB	parsed	nt	http://creativecommons.org/publicdomain/zero/1.0/
	13 KB	parsed, sorted	nt	http://creativecommons.org/publicdomain/zero/1.0/

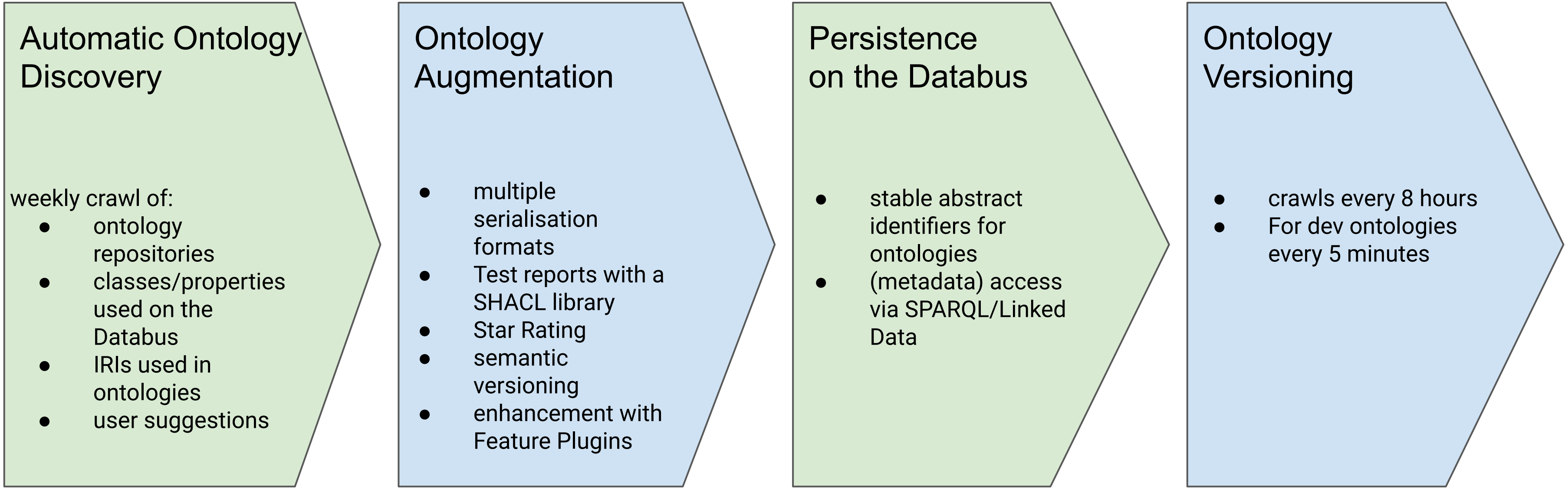
Ship datasets / apps with the **most recent or specific version of ontologies** using e.g. Databus collections

→ clear **provenance**

→ stable vs. updateable applications

→ **reproducible experiments**

Wrap Up: Archivo Workflow and Features



Automatic Ontology Discovery

weekly crawl of:

- ontology repositories
- classes/properties used on the Databus
- IRIs used in ontologies
- user suggestions

Ontology Augmentation

- multiple serialisation formats
- Test reports with a SHACL library
- Star Rating
- semantic versioning
- enhancement with Feature Plugins

Persistence on the Databus

- stable abstract identifiers for ontologies
- (metadata) access via SPARQL/Linked Data

Ontology Versioning

- crawls every 8 hours
- For dev ontologies every 5 minutes

Summary

Archivo …

- … is an exhaustive unified space for ontologies
- … provides **f**indable and easily **a**ccessible vocabularies
- … has a star rating and other tests measuring the **i**nteroperability and (**r**e)usabilty of ontologies
- … tries to encourage following community standards for ontology metadata

Contribute to Archivo and better Linked Open Data

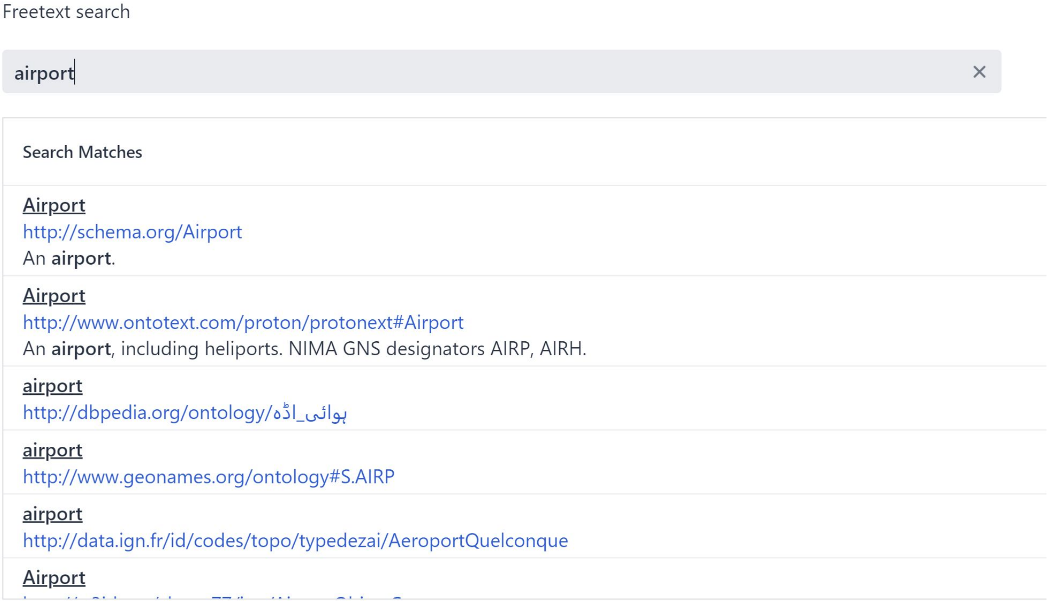
- add not-yet-discovered ontologies (esp. ones you use)
  - Check and report issues to maintainers of ontologies you use (point to Archivo issues)
- check your own ontology and their rating and improve them to get ★★★★
- add SHACL tests checking for the compliance to a certain service
- suggest new features/measurements/tests by creating an issue at the github repository

Terminology Server using Databus, Lookup and Archivo

by Johannes Frey

Searching for Terms from Ontologies (A)

Ranked Fuzzy Term Search via (Lucene) Index powered by DBpedia Lookup indexing configuration (public service alpha)



Freetext search

airport

Search Matches

Airport
http://schema.org/Airport
An **airport**.

Airport
http://www.ontotext.com/proton/protonext#Airport
An **airport**, including heliports. NIMA GNS designators AIRP, AIRH.

airport
http://dbpedia.org/ontology/ہوائی_اڈہ

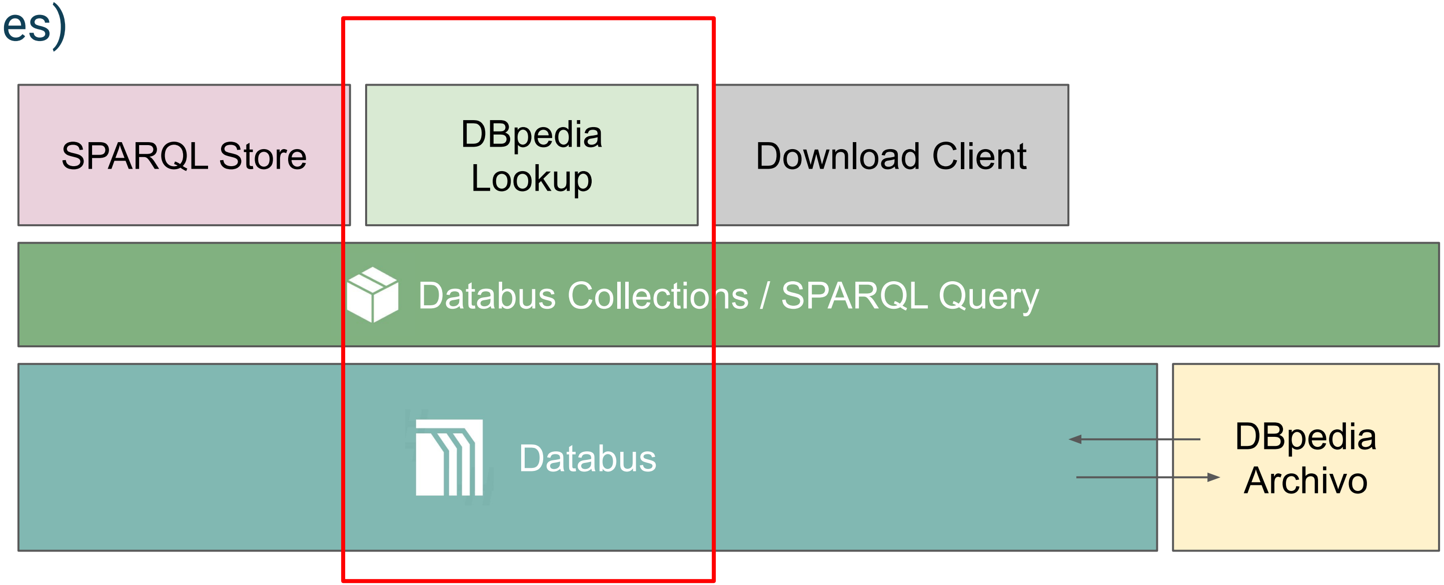
airport
http://www.geonames.org/ontology#S.AIRP

airport
http://data.ign.fr/id/codes/topo/typedezai/AeroportQuelconque

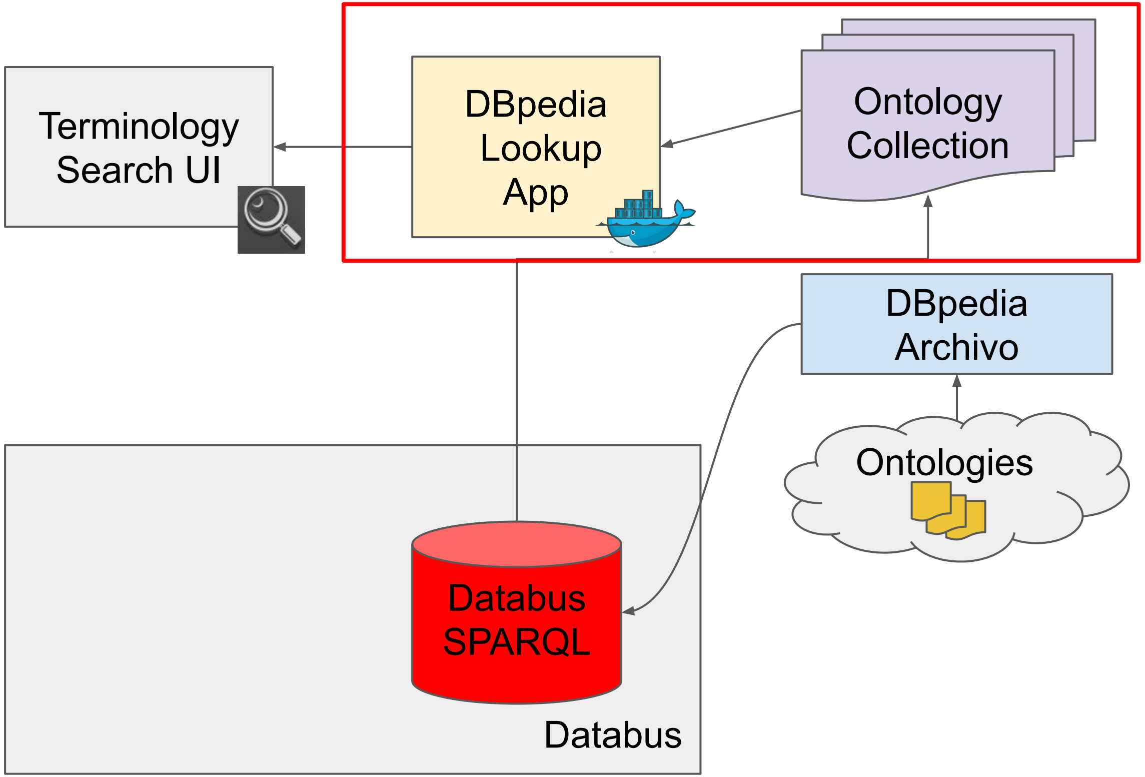
Airport

DBpedia Technology Stack

- Several “one-click” of deployment services of DBpedia Tech stack can be used with DBpedia Archivo
- Ontologies can be fed into application via Collection ID (custom or official ones)



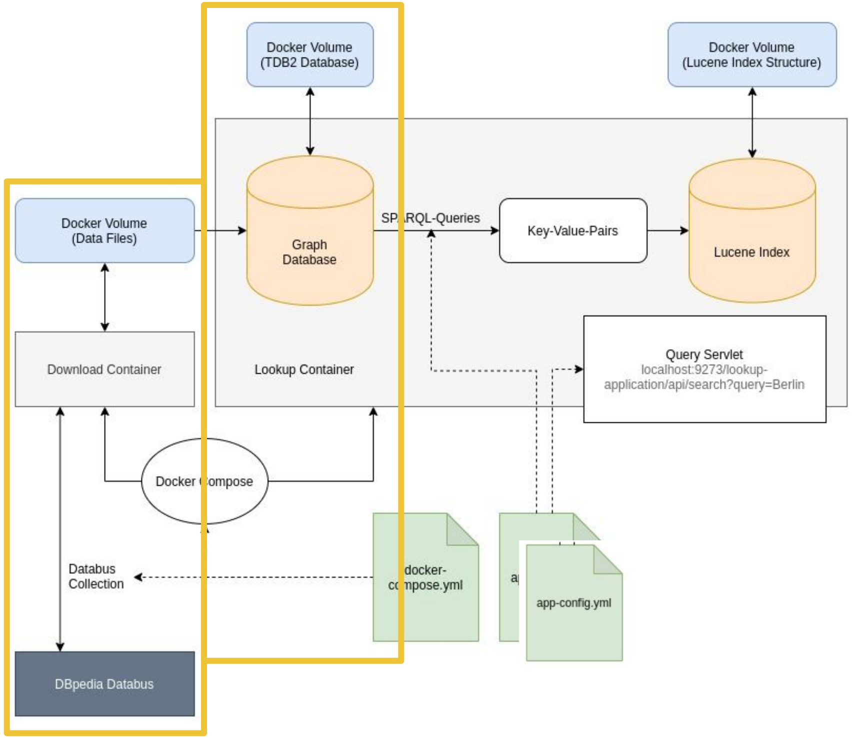
Terminology Server Workflow Overview



On-demand Lookup (Fetch Data)

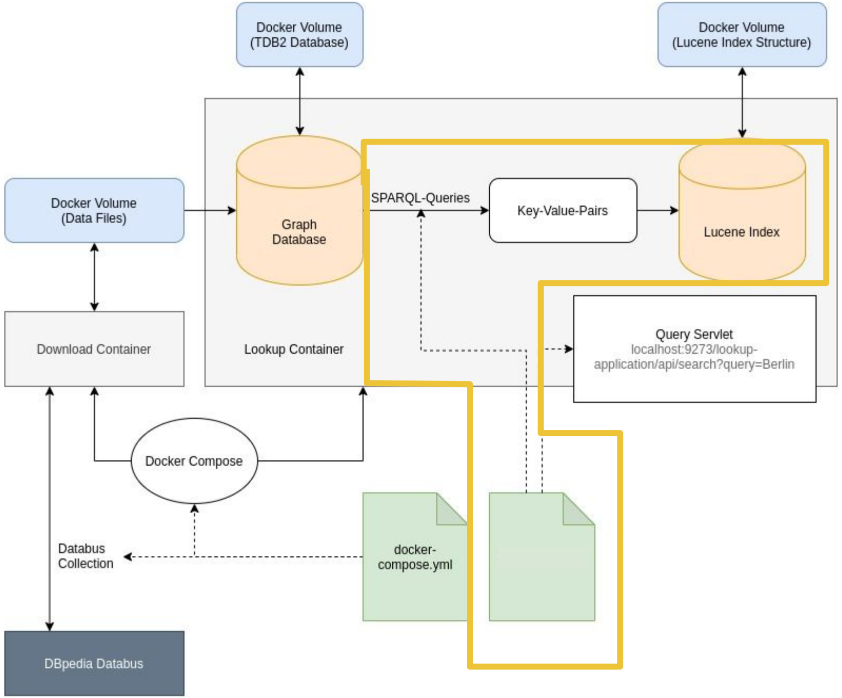
- Entity keyword/term search for datasets
- Composite of:
  - Download Container (optional - now builtin)
  - Lookup Container (Application Container)
- by default data to be indexed is loaded into an (on-disk) graph database via specifying a Databus Collection

https://github.com/dbpedia/dbpedia-lookup



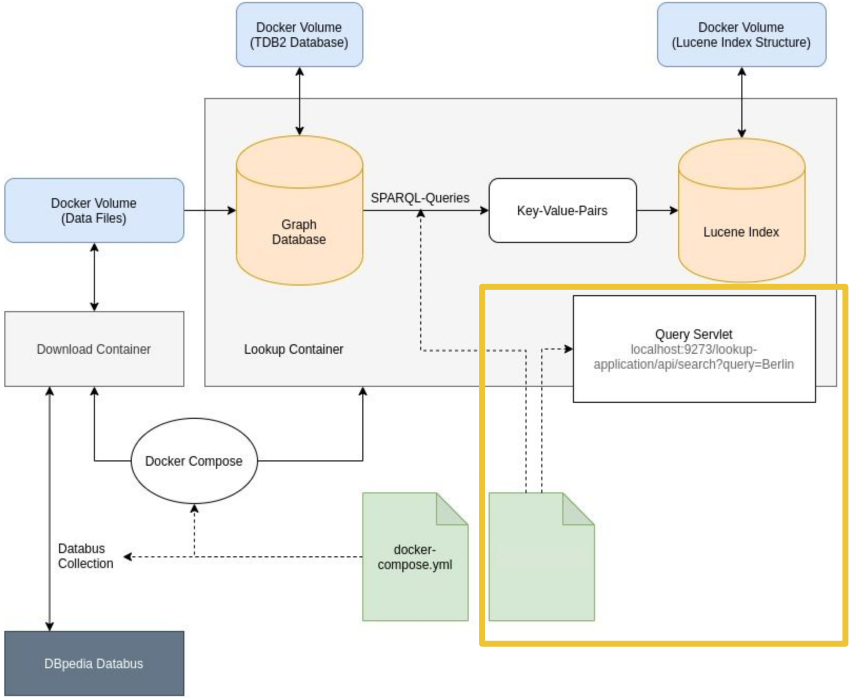
On-demand Lookup: (Indexing)

- Key-Value pairs are extracted via SPARQL queries to create a reverse index
- Customizable in YML file
- default configuration is provided that works out of the box for a plethora of RDF datasets (uses rdfs:label and rdfs:comment as keywords/”search fields”)
- prebuilt indexes for a selection of DBpedia datasets available on Databus



On-demand Lookup (Search)

- flexible search/query behaviour (based on Lucene - entities ~ documents)
    - e.g. the public DBpedia Lookup API “prefix search” and “autocompletion” services use same code and data but two different configurations https://lookup.dbpedia.org



Let`s create an Index on the DBpedia Ontology

```
git clone https://github.com/dbpedia/dbpedia-lookup
cd dbpedia-lookup/lookup
mvn package

EITHER   cd .. && docker compose up
OR       java -jar ./target/lookup-1.0-jar-with-dependencies.jar -c ../examples/config.yml

curl --request POST \
--url http://localhost:8082/api/index/run \
--header 'Content-Type: multipart/form-data' \
--form config=@examples/indexing/dbpedia-ontology-collection-indexer.yml

curl http://localhost:8082/api/search\?query\=Wind
```

Advantages and Usage Scenarios

- Search on your custom selection of ontologies / private of vocabulary
- Customize the search how you like to improve search results at ease
  - Prefix search
  - Fuzzy search
  - Indexed content and fields
  - …
- Quick and automated deployment

- We use it for assistance in interfacing KGs with LLMs (e.g. GraphRAG)

DBpedia

Q&A

Closing Session

by Milan Dojchinovski

What have you learned

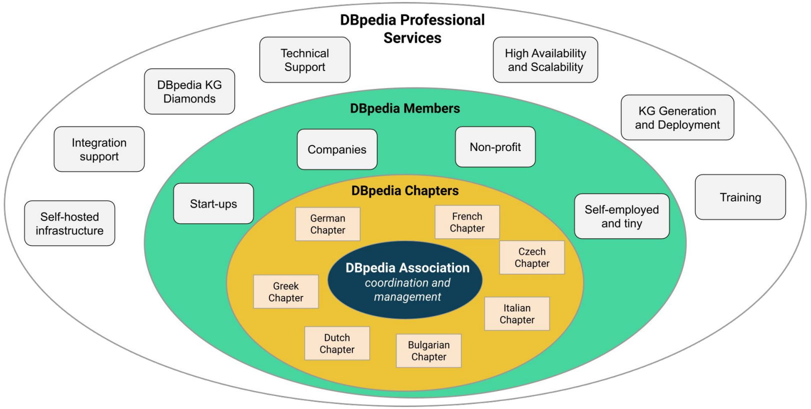
- What is DBpedia
  - history of DBpedia, community, DBpedia KG release process
  - how a DBpedia triple is born
  - ontology, endpoints
- The DBpedia technology stack
  - DBpedia Databus and collections
  - DBpedia Spotlight
  - DBpedia Lookup
- best practices via several practical showcases
  - Semantic Indexing and Search using Databus and DBpedia Spotlight
  - CI and Databus publishing using Jenkins
  - Databus Metadata Overlay Search System
  - Terminology Server using DBpedia Lookup

Useful pointers

- Please find more information about DBpedia and the community here: https://www.dbpedia.org/
- Join the DBpedia slack: https://dbpedia-slack.herokuapp.com/
- Join the DBpedia Forum: https://forum.dbpedia.org/
- Sign up for the fabulous DBpedia newsletter: http://eepurl.com/blg3qf

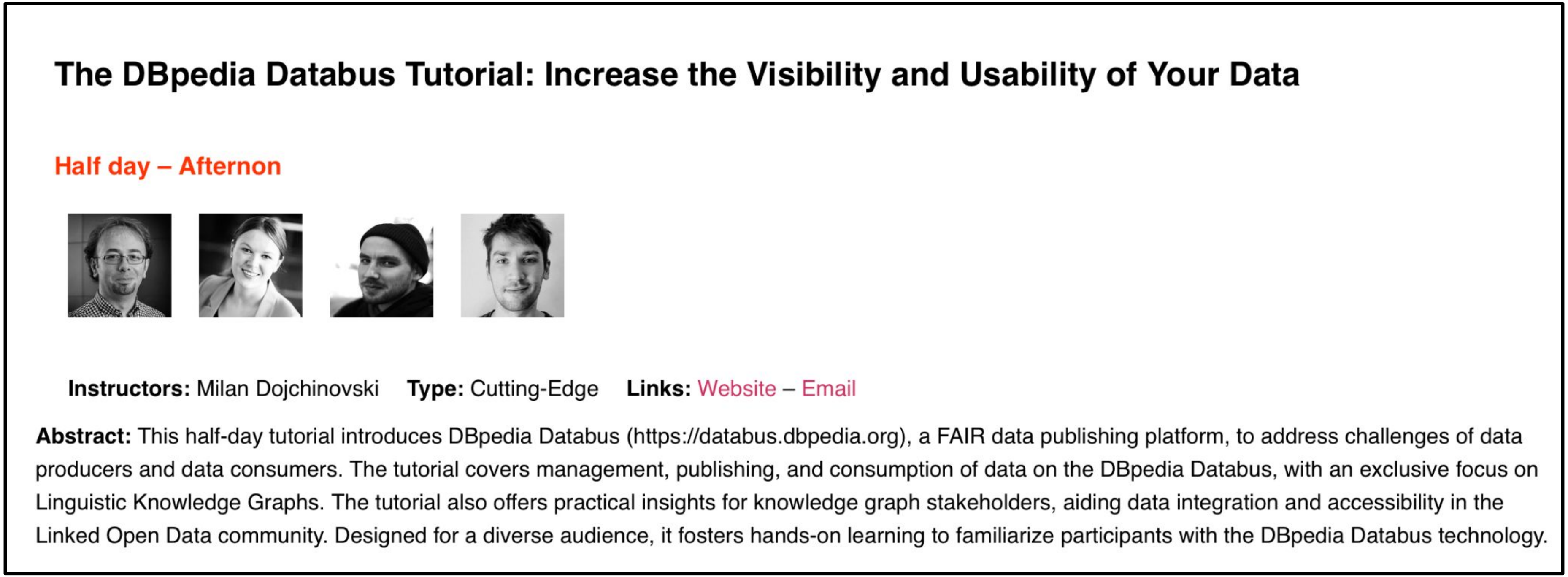
Join DBpedia

- Establish DBpedia chapter
  - https://www.dbpedia.org/members/chapter-overview/
- Become a member
  - https://www.dbpedia.org/members/membership/
  - request material via dbpedia@infai.org
- Get DBpedia professional services
  - training
  - consulting on your use cases
  - self-hosting DBpedia
  - technical support
  - request material via dbpedia@infai.org



Next events

DBpedia Tutorial at LREC-COOLING 2024, *May 20, 2024, Torino, Italy*



The DBpedia Databus Tutorial: Increase the Visibility and Usability of Your Data

Half day – Afternon

Instructors: Milan Dojchinovski **Type:** Cutting-Edge **Links:** Website – Email

Abstract: This half-day tutorial introduces DBpedia Databus (https://databus.dbpedia.org), a FAIR data publishing platform, to address challenges of data producers and data consumers. The tutorial covers management, publishing, and consumption of data on the DBpedia Databus, with an exclusive focus on Linguistic Knowledge Graphs. The tutorial also offers practical insights for knowledge graph stakeholders, aiding data integration and accessibility in the Linked Open Data community. Designed for a diverse audience, it fosters hands-on learning to familiarize participants with the DBpedia Databus technology.

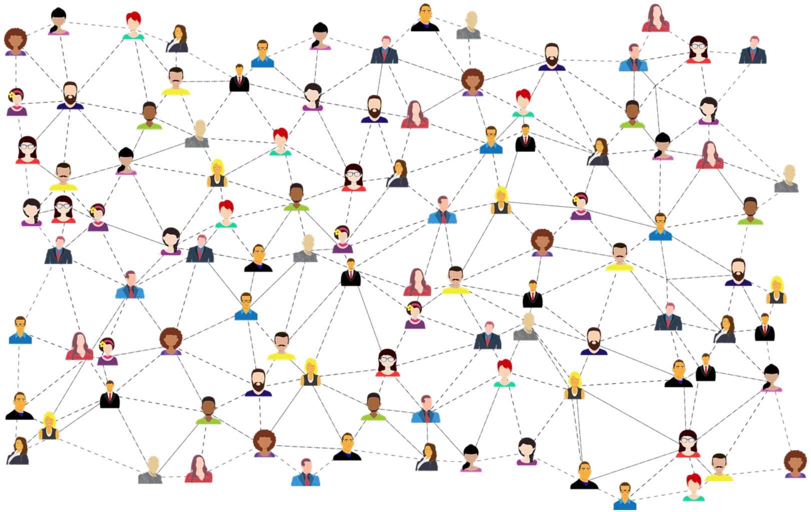
DBpedia for Saxony Digital Prize

sächsischer
digitalpreis

- DBpedia nominated in the Open Source category
- Voting period: 15-30 April, 2024 at 10:00 a.m.
- Please vote via the participation portal of the Free State of Saxony
  - Link to the Saxon Digital Award 2024 public voting:
  - https://mitdenken.sachsen.de/1040556
- Short films and further information on the nominees
  - the Saxon Digital Prize website https://www.digitales.sachsen.de/saechsischer-digitalpreis-2024-5634.html

Thank you! Q&A

… final thoughts or questions?



Leftovers
everything after this slide will be removed !!!

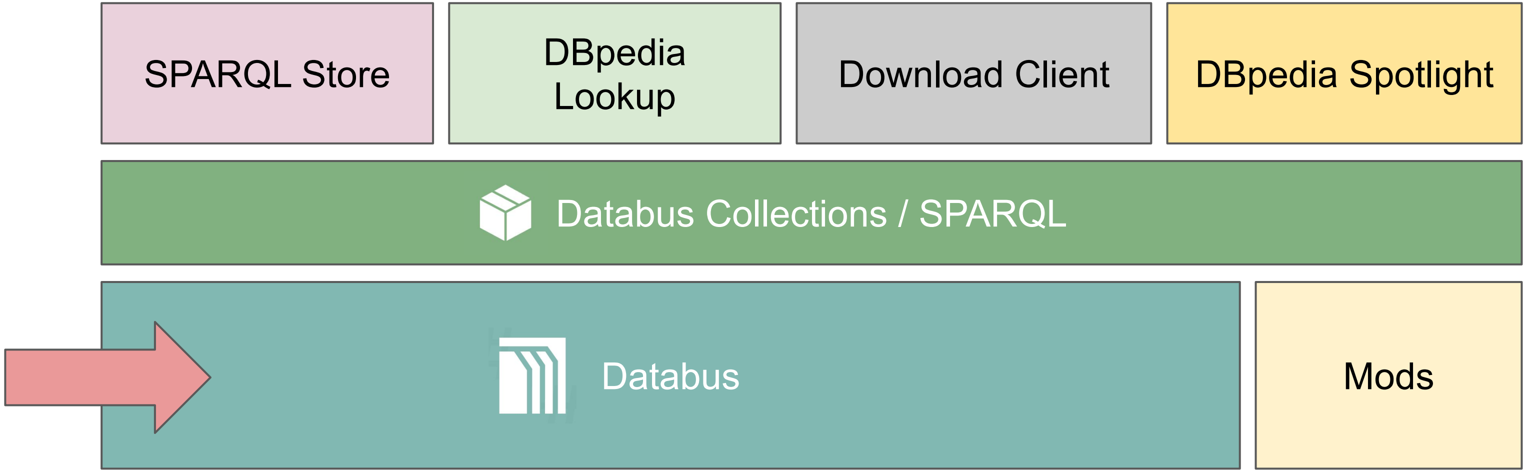
Part 2: DBpedia Technology Stack Overview and Demo case

by Jan Forberg

Overview

- DBpedia Technology Stack
- DBpedia Databus
  - Core Concepts
  - Collections
  - Publishing to the Databus
- Demo case using the Stack

The DBpedia Technology Stack



SPARQL Store

DBpedia Lookup

Download Client

DBpedia Spotlight

Databus Collections / SPARQL

Databus

Mods

DBpedia Databus

- RDF-based metadata registry
- Holds metadata about files
    - Format
    - Compression
    - Size
    - Checksums
    - Download URLs
    - Content-variant information
    - … and more
- Data-retrieval can be done via SPARQL-queries
- Federated SPARQL queries (SPARQL queries over multiple triple-stores) allow inter-Databus aggregation
- High focus on automatization, interoperability and extensibility

Databus Core Concepts

Databus (Data dependencies) is inspired by Maven (Software dependencies)

- **Artifact**

  gical Dataset (e.g. “All Wikipedia Labels”, “Data about Water Turbines”). May have multiple sions and files in different formats, languages, etc.

oup

tiple Artifacts grouped together.

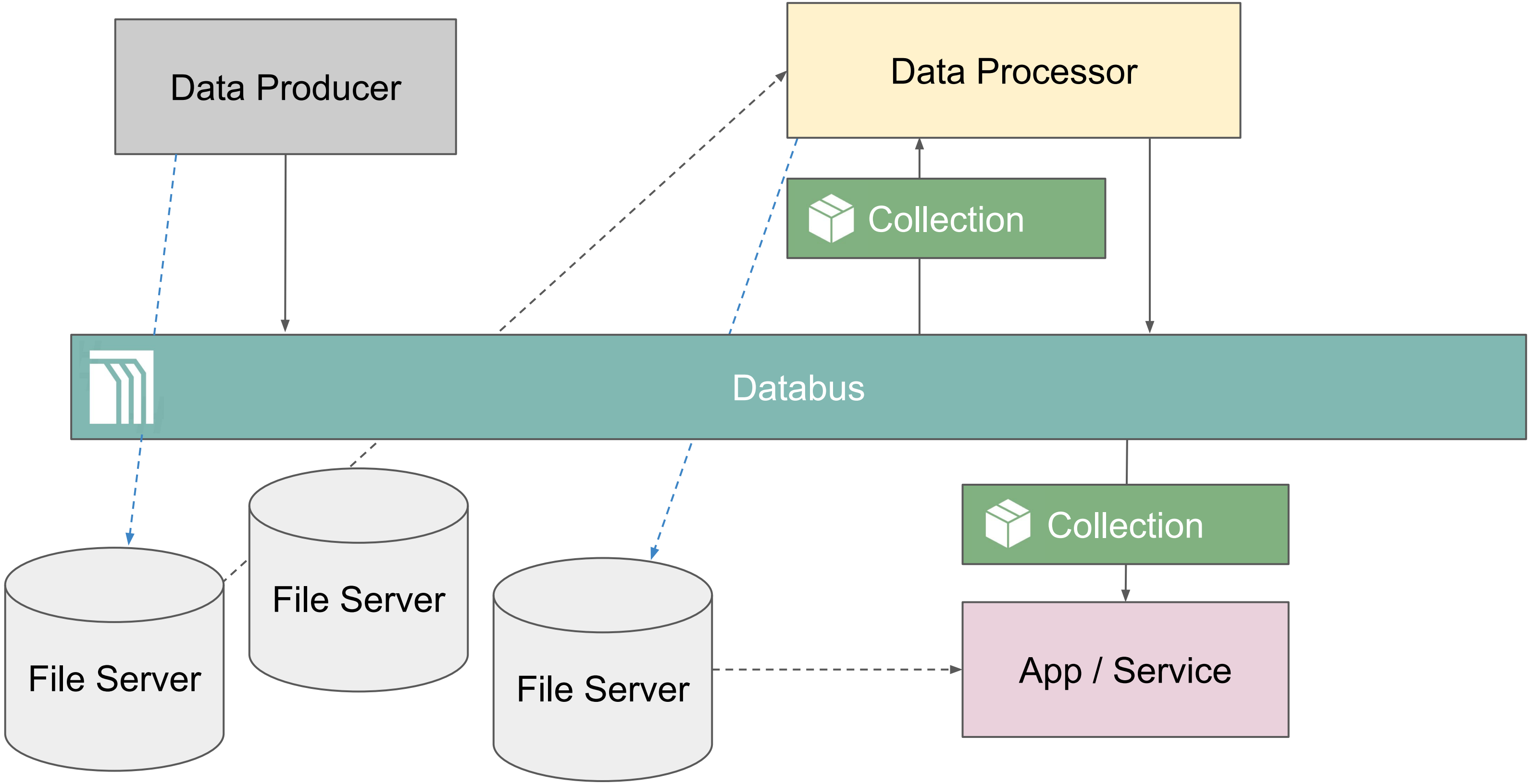
- **Version**

  sion of an Artifact. (e.g. “2016-10 release of All Wikipedia Labels”)

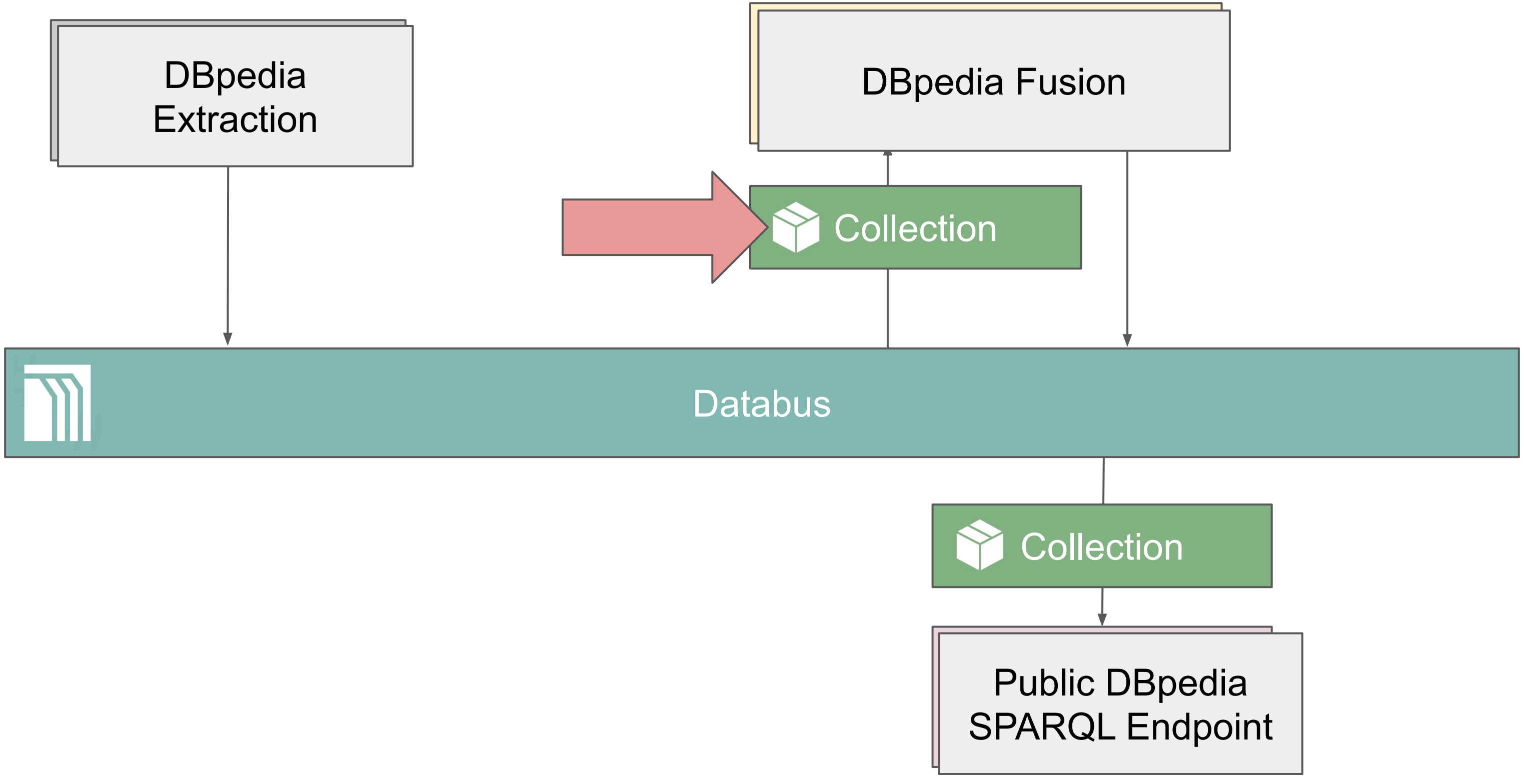
tald

Metadata document associated with exactly one Group, Artifact and Version

Databus for Data Automatization



Databus for Data Automatization



DBpedia Extraction

DBpedia Fusion

Collection

Databus

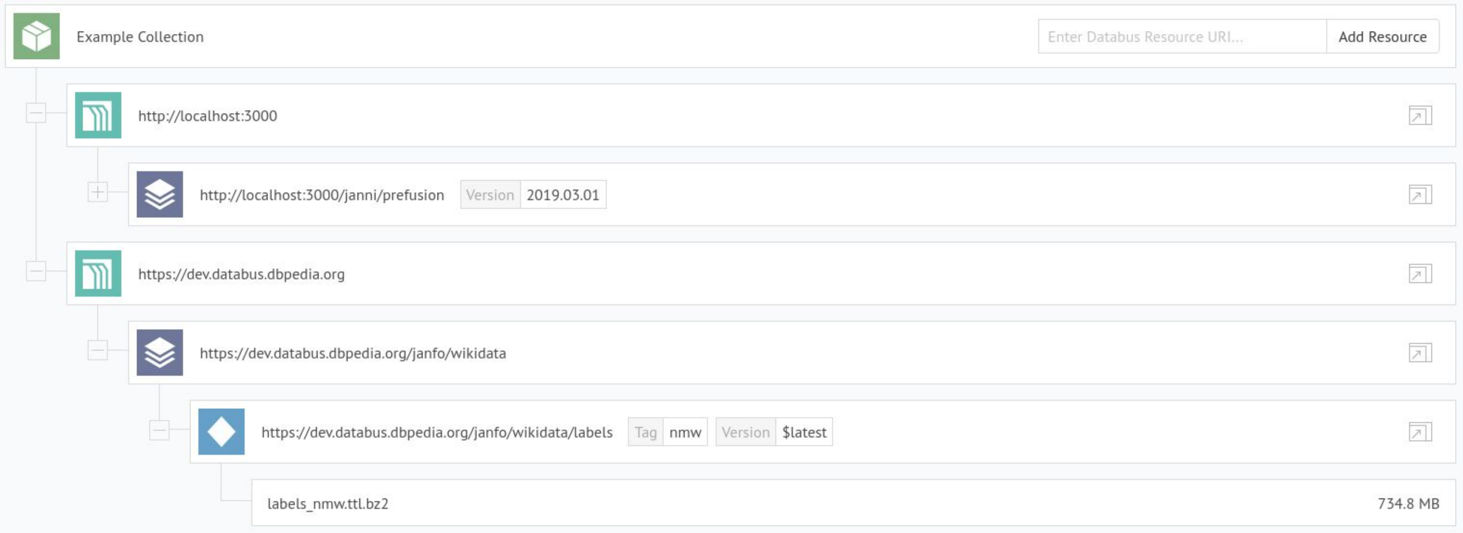
Collection

Public DBpedia SPARQL Endpoint

Databus Collections

- The core aggregation and retrieval mechanism of a Databus (abstraction layer for SPARQL)
- Shopping cart for data
- Editor provided with the web-interface

DBpedia Databus Collection Editor



Example Collection

http://localhost:3000

http://localhost:3000/janni/prefusion Version 2019.03.01

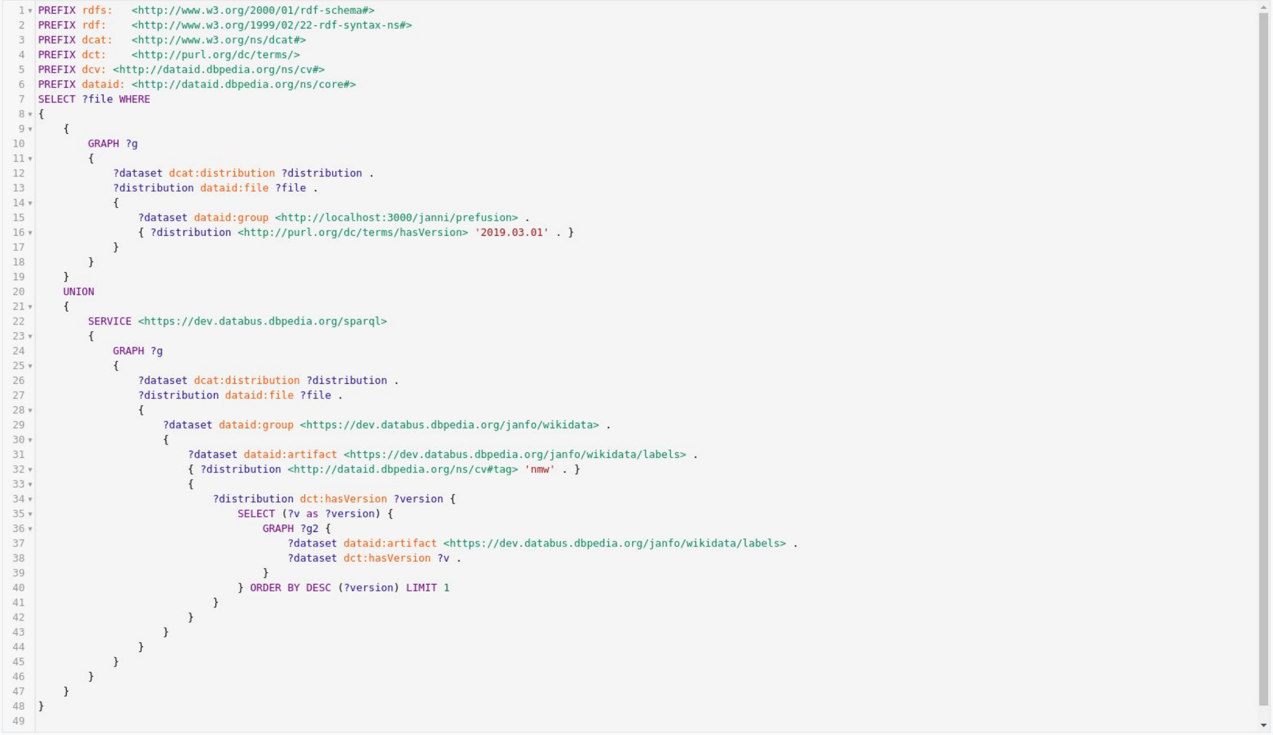
https://dev.databus.dbpedia.org

https://dev.databus.dbpedia.org/janfo/wikidata

https://dev.databus.dbpedia.org/janfo/wikidata/labels Tag nmw Version $latest

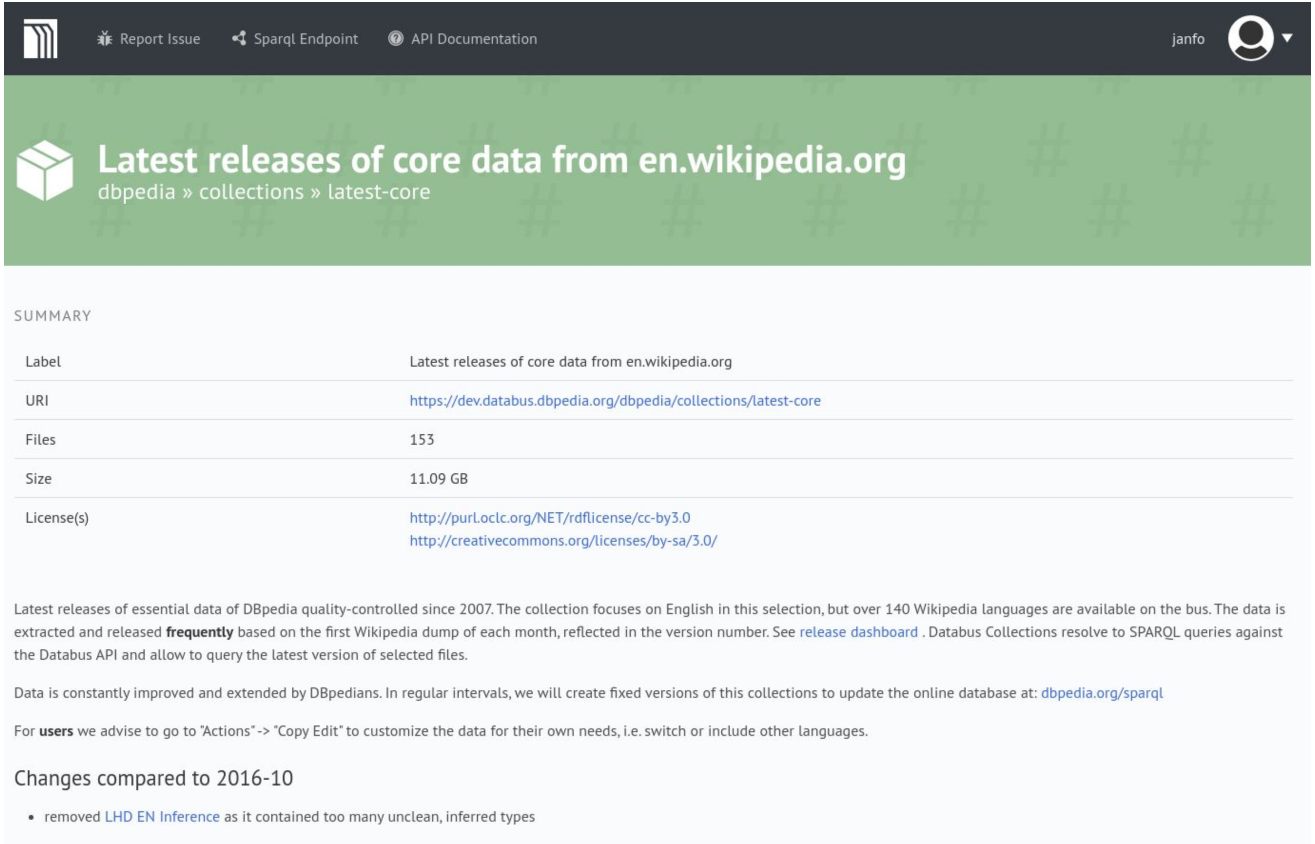
labels_nmw.ttl.bz2 734.8 MB

DBpedia Databus Collection Editor



```
PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcat:   <http://www.w3.org/ns/dcat#>
PREFIX dct:    <http://purl.org/dc/terms/>
PREFIX dcv: <http://dataid.dbpedia.org/ns/cv#>
PREFIX dataid: <http://dataid.dbpedia.org/ns/core#>
SELECT ?file WHERE
{
    {
        GRAPH ?g
        {
            ?dataset dcat:distribution ?distribution .
            ?distribution dataid:file ?file .
            {
                ?dataset dataid:group <http://localhost:3000/janni/prefusion> .
                { ?distribution <http://purl.org/dc/terms/hasVersion> '2019.03.01' . }
            }
        }
    }
    UNION
    {
        SERVICE <https://dev.databus.dbpedia.org/sparql>
        {
            GRAPH ?g
            {
                ?dataset dcat:distribution ?distribution .
                ?distribution dataid:file ?file .
                {
                    ?dataset dataid:group <https://dev.databus.dbpedia.org/janfo/wikidata> .
                    {
                        ?dataset dataid:artifact <https://dev.databus.dbpedia.org/janfo/wikidata/labels> .
                        { ?distribution <http://dataid.dbpedia.org/ns/cv#tag> 'nmw' . }
                        {
                            ?distribution dct:hasVersion ?version {
                                SELECT (?v as ?version) {
                                    GRAPH ?g2 {
                                        ?dataset dataid:artifact <https://dev.databus.dbpedia.org/janfo/wikidata/labels> .
                                        ?dataset dct:hasVersion ?v .
                                    }
                                } ORDER BY DESC (?version) LIMIT 1
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Databus Collections HTML View



Latest releases of core data from en.wikipedia.org

dbpedia » collections » latest-core

SUMMARY

Label	Latest releases of core data from en.wikipedia.org
URI	https://dev.databus.dbpedia.org/dbpedia/collections/latest-core
Files	153
Size	11.09 GB
License(s)	http://purl.oclc.org/NET/rdflicense/cc-by3.0 http://creativecommons.org/licenses/by-sa/3.0/

Latest releases of essential data of DBpedia quality-controlled since 2007. The collection focuses on English in this selection, but over 140 Wikipedia languages are available on the bus. The data is extracted and released **frequently** based on the first Wikipedia dump of each month, reflected in the version number. See release dashboard . Databus Collections resolve to SPARQL queries against the Databus API and allow to query the latest version of selected files.

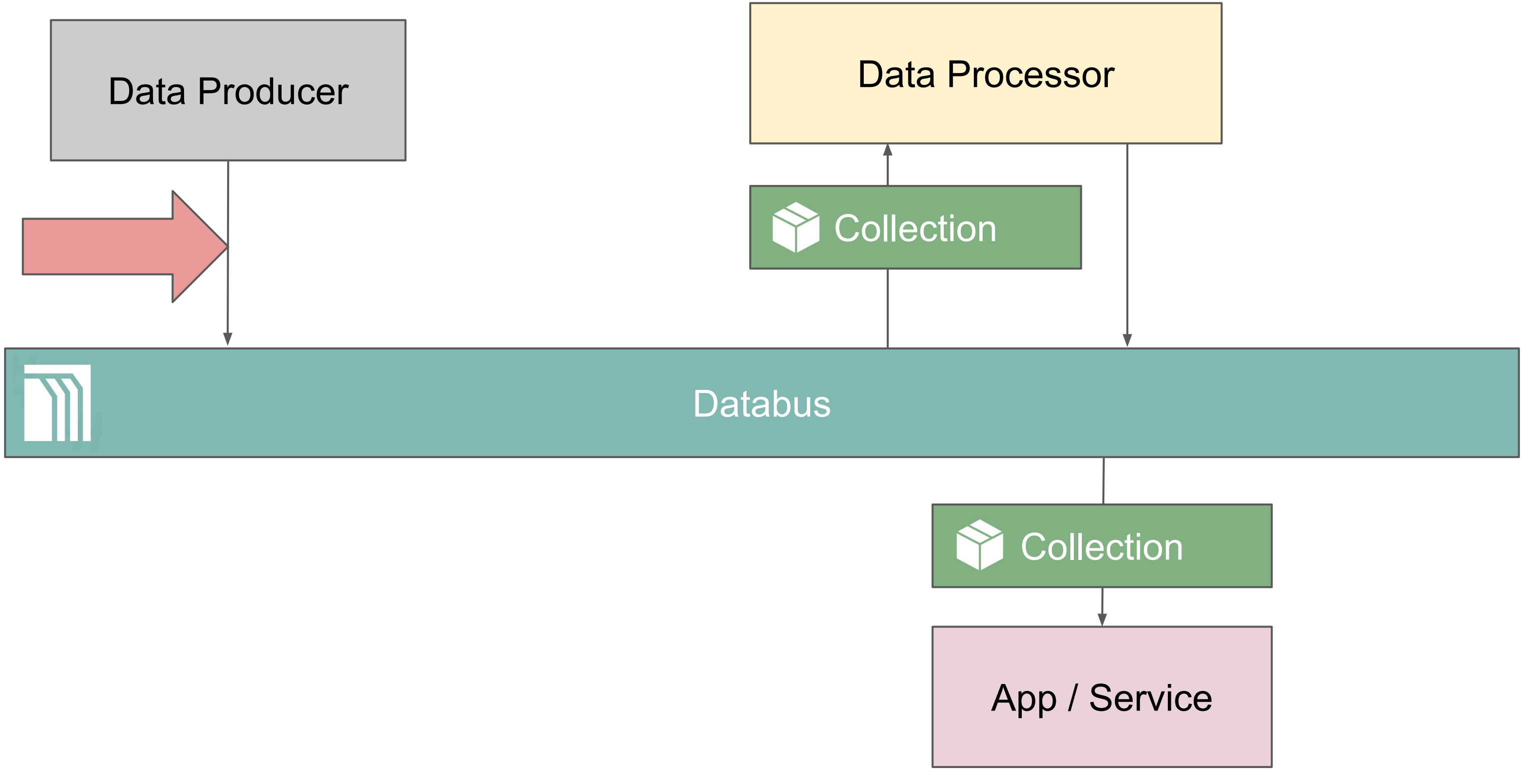
Data is constantly improved and extended by DBpedians. In regular intervals, we will create fixed versions of this collections to update the online database at: dbpedia.org/sparql

For **users** we advise to go to "Actions" -> "Copy Edit" to customize the data for their own needs, i.e. switch or include other languages.

Changes compared to 2016-10

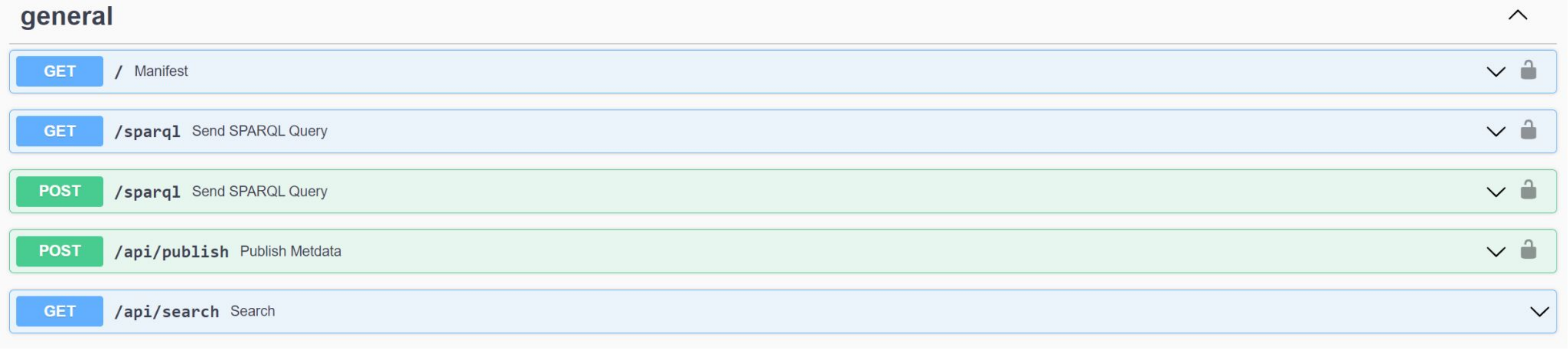
- removed LHD EN Inference as it contained too many unclean, inferred types

Databus for Data Automatization



Publishing to the Databus

- Extensive API for all Databus interactions
- Inputs based on JSON-LD
- Web UI helps with first steps
- Simple API-key authentication



general

GET / Manifest

GET /sparql Send SPARQL Query

POST /sparql Send SPARQL Query

POST /api/publish Publish Metdata

GET /api/search Search

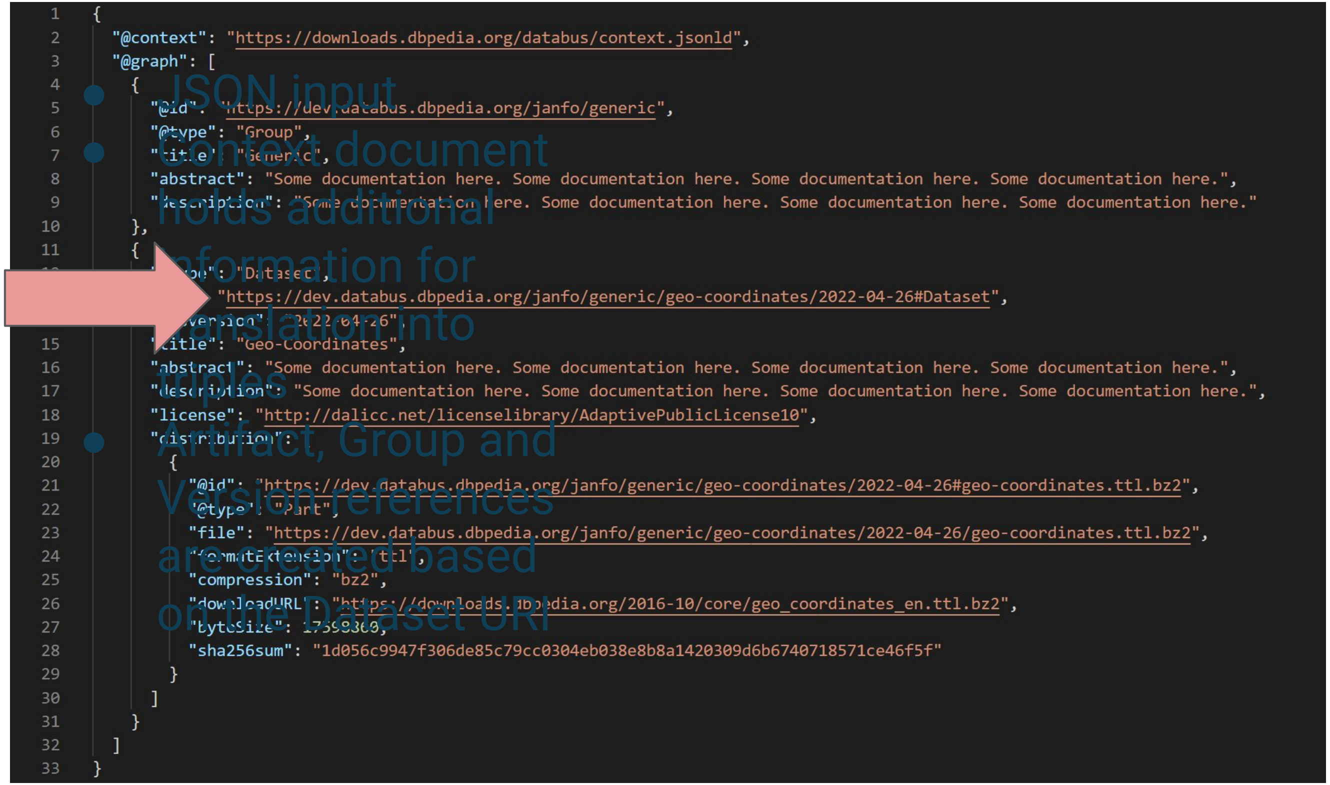
Creating a Databus Version

- Core piece: Dataset
- Dataset has
  - Title, Abstract, etc.
  - List of Parts
- Each Part describes a file with
  - Format
  - Compression
  - Download URL
  - .. and more
- Dataset is associated with
  - Account
  - Group
  - Artifact
  - Version

Server-side Auto-completion

- Reduces the amount of redundant information in the input
- Does tedious tasks such as generating checksums
- Creates entries for Groups, Artifacts and Versions

Example JSON Input

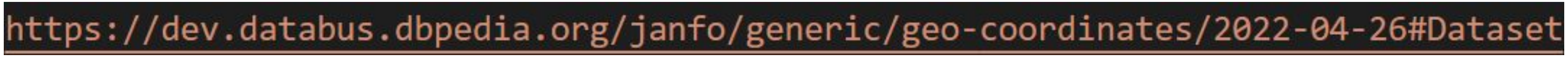


```
{
  "@context": "https://downloads.dbpedia.org/databus/context.jsonld",
  "@graph": [
    {
      "@id": "https://dev.databus.dbpedia.org/janfo/generic",
      "@type": "Group",
      "title": "Generic",
      "abstract": "Some documentation here. Some documentation here. Some documentation here. Some documentation here.",
      "description": "Some documentation here. Some documentation here. Some documentation here. Some documentation here."
    },
    {
      "@type": "Dataset",
            "https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates/2022-04-26#Dataset",
      "version": "2022-04-26",
      "title": "Geo-Coordinates",
      "abstract": "Some documentation here. Some documentation here. Some documentation here. Some documentation here.",
      "description": "Some documentation here. Some documentation here. Some documentation here. Some documentation here.",
      "license": "http://dalicc.net/licenselibrary/AdaptivePublicLicense10",
      "distribution": [
        {
          "@id": "https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates/2022-04-26#geo-coordinates.ttl.bz2",
          "@type": "Part",
          "file": "https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates/2022-04-26/geo-coordinates.ttl.bz2",
          "formatExtension": "ttl",
          "compression": "bz2",
          "downloadURL": "https://downloads.dbpedia.org/2016-10/core/geo_coordinates_en.ttl.bz2",
          "byteSize": 17598369,
          "sha256sum": "1d056c9947f306de85c79cc0304eb038e8b8a1420309d6b6740718571ce46f5f"
        }
      ]
    }
  ]
}
```

- JSON input context document holds additional information for translation into triples
- Artifact, Group and Version references are created based on the Dataset URI

Example JSON Input

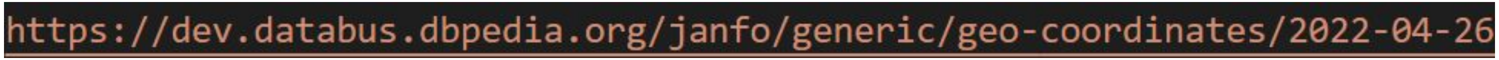
Dataset URI



```
https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates/2022-04-26#Dataset
```

Example JSON Input

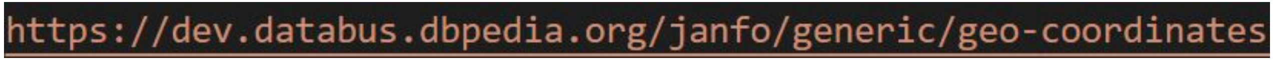
Version URI



```
https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates/2022-04-26
```

Example JSON Input

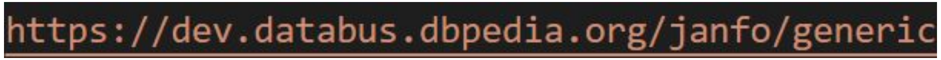
Artifact URI



```
https://dev.databus.dbpedia.org/janfo/generic/geo-coordinates
```

Example JSON Input

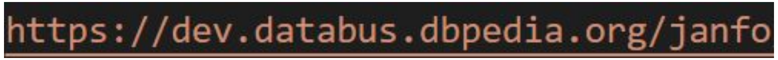
Group URI



```
https://dev.databus.dbpedia.org/janfo/generic
```

Example JSON Input

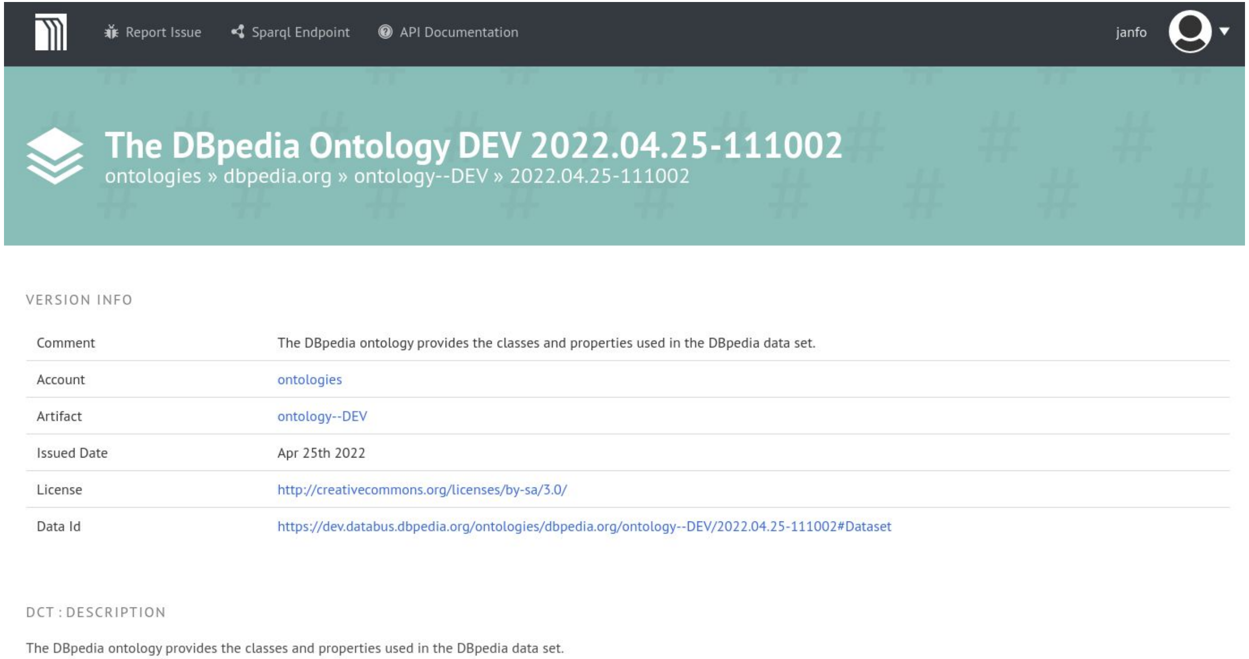
Account URI



```
https://dev.databus.dbpedia.org/janfo
```

Example Result

https://dev.databus.dbpedia.org/ontologies/dbpedia.org/ontology--DEV/2022.04.25-111002



The DBpedia Ontology DEV 2022.04.25-111002

ontologies » dbpedia.org » ontology--DEV » 2022.04.25-111002

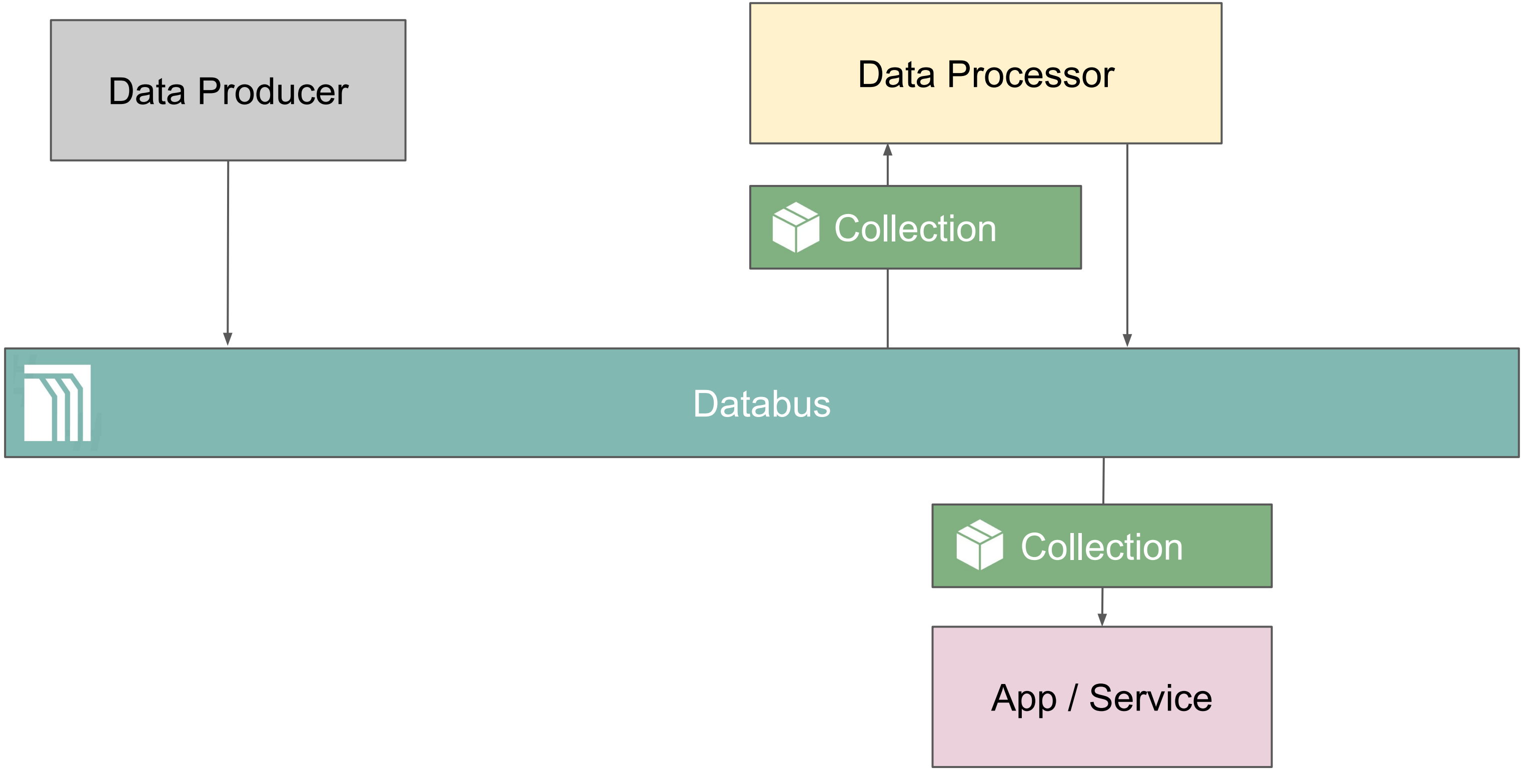
VERSION INFO

Comment	The DBpedia ontology provides the classes and properties used in the DBpedia data set.
Account	ontologies
Artifact	ontology--DEV
Issued Date	Apr 25th 2022
License	http://creativecommons.org/licenses/by-sa/3.0/
Data Id	https://dev.databus.dbpedia.org/ontologies/dbpedia.org/ontology--DEV/2022.04.25-111002#Dataset

DCT : DESCRIPTION

The DBpedia ontology provides the classes and properties used in the DBpedia data set.

Databus for Data Automatization



Demo Case: Semantic Indexing and Search using the DBpedia technology stack

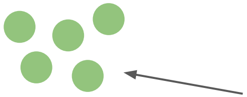
Part 3:
Use DBpedia on your local infrastructure

by Johannes Frey

Overview

- Option to access / download DBpedia KG
  - Navigating the DBpedia KG on the Databus
- Services deployable on your own infrastructure
  - Databus
  - Lookup
  - Spotlight

How to access the DBpedia KG?

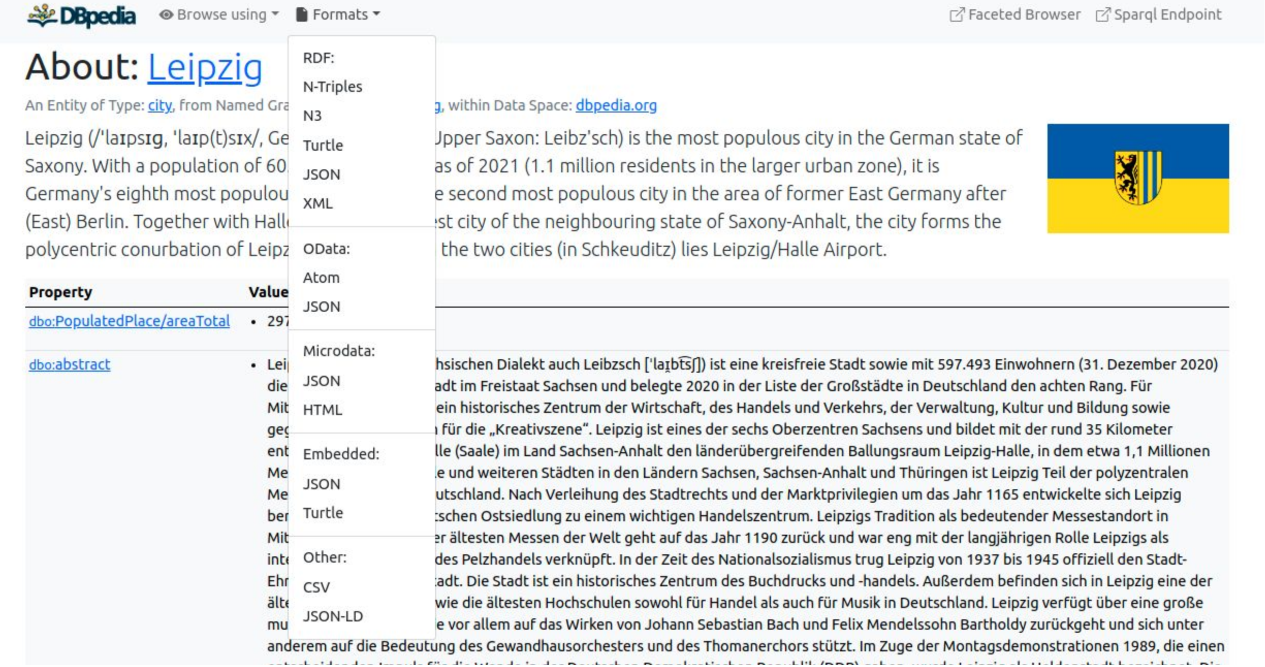


DBpedia KG

A set of files containing RDF data

How to access the DBpedia KG?

Option 0: Linked Data interface via https://dbpedia.org/resource/Leipzig



About: Leipzig

An Entity of Type: city, from Named Gra g, within Data Space: dbpedia.org

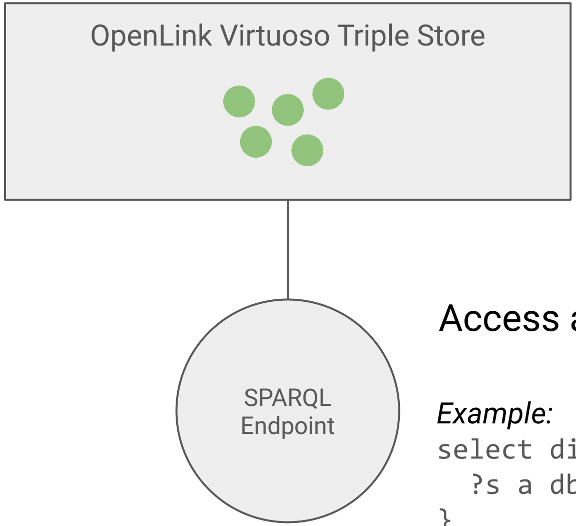
Leipzig (/ˈlaɪpsɪɡ, ˈlaɪp(t)sɪx/, Ge Upper Saxon: Leibz'sch) is the most populous city in the German state of
Saxony. With a population of 60 as of 2021 (1.1 million residents in the larger urban zone), it is
Germany's eighth most populou e second most populous city in the area of former East Germany after
(East) Berlin. Together with Hall st city of the neighbouring state of Saxony-Anhalt, the city forms the
polycentric conurbation of Leipz the two cities (in Schkeuditz) lies Leipzig/Halle Airport.

Property	Value
dbo:PopulatedPlace/areaTotal	• 297
dbo:abstract	• Lei hsischen Dialekt auch Leibzsch [ˈlai̯bt͡sʃ]) ist eine kreisfreie Stadt sowie mit 597.493 Einwohnern (31. Dezember 2020) die adt im Freistaat Sachsen und belegte 2020 in der Liste der Großstädte in Deutschland den achten Rang. Für Mit ein historisches Zentrum der Wirtschaft, des Handels und Verkehrs, der Verwaltung, Kultur und Bildung sowie ge für die „Kreativszene“. Leipzig ist eines der sechs Oberzentren Sachsens und bildet mit der rund 35 Kilometer ent lle (Saale) im Land Sachsen-Anhalt den länderübergreifenden Ballungsraum Leipzig-Halle, in dem etwa 1,1 Millionen Me e und weiteren Städten in den Ländern Sachsen, Sachsen-Anhalt und Thüringen ist Leipzig Teil der polyzentralen Me utschland. Nach Verleihung des Stadtrechts und der Marktprivilegien um das Jahr 1165 entwickelte sich Leipzig ber tschen Ostsiedlung zu einem wichtigen Handelszentrum. Leipzigs Tradition als bedeutender Messestandort in Mit er ältesten Messen der Welt geht auf das Jahr 1190 zurück und war eng mit der langjährigen Rolle Leipzigs als int des Pelzhandels verknüpft. In der Zeit des Nationalsozialismus trug Leipzig von 1937 bis 1945 offiziell den Stadt- Ehr tadt. Die Stadt ist ein historisches Zentrum des Buchdrucks und -handels. Außerdem befinden sich in Leipzig eine der ält wie die ältesten Hochschulen sowohl für Handel als auch für Musik in Deutschland. Leipzig verfügt über eine große mu e vor allem auf das Wirken von Johann Sebastian Bach und Felix Mendelssohn Bartholdy zurückgeht und sich unter anderem auf die Bedeutung des Gewandhausorchesters und des Thomanerchors stützt. Im Zuge der Montagsdemonstrationen 1989, die einen

```
curl -Lk -H "Accept: application/n-triples" https://dbpedia.org/resource/Leipzig | vim -
```

How to access the DBpedia KG?

Option A: The official DBpedia KG SPARQL-Endpoint
(or endpoints of national DBpedia Chapters or DBpedia Live SPARQL)



Access at https://dbpedia.org/sparql

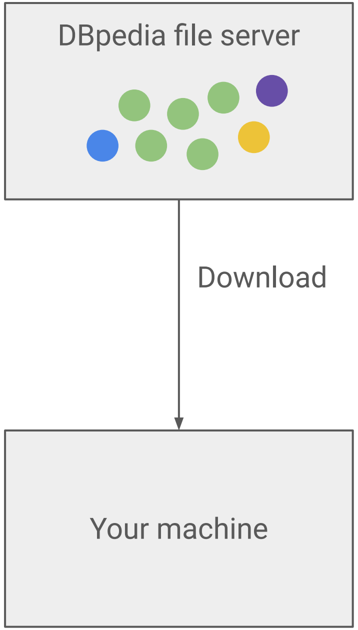
Example:

```
select distinct ?s where {
  ?s a dbo:Organisation .
}
```

YASGUI: https://yasgui.triply.cc/

How to access the DBpedia KG?

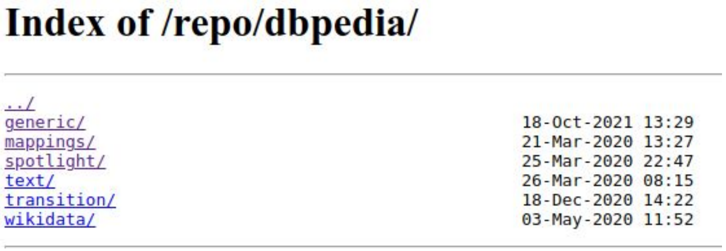
Option B: Old-school File Directory Download



Access at https://downloads.dbpedia.org

Monthly Modular Extractions (since 2020)
https://downloads.dbpedia.org/repo/dbpedia/<module>/<dataset>/<version>

Legacy releases (until 2017)
https://downloads.dbpedia.org/<version>/

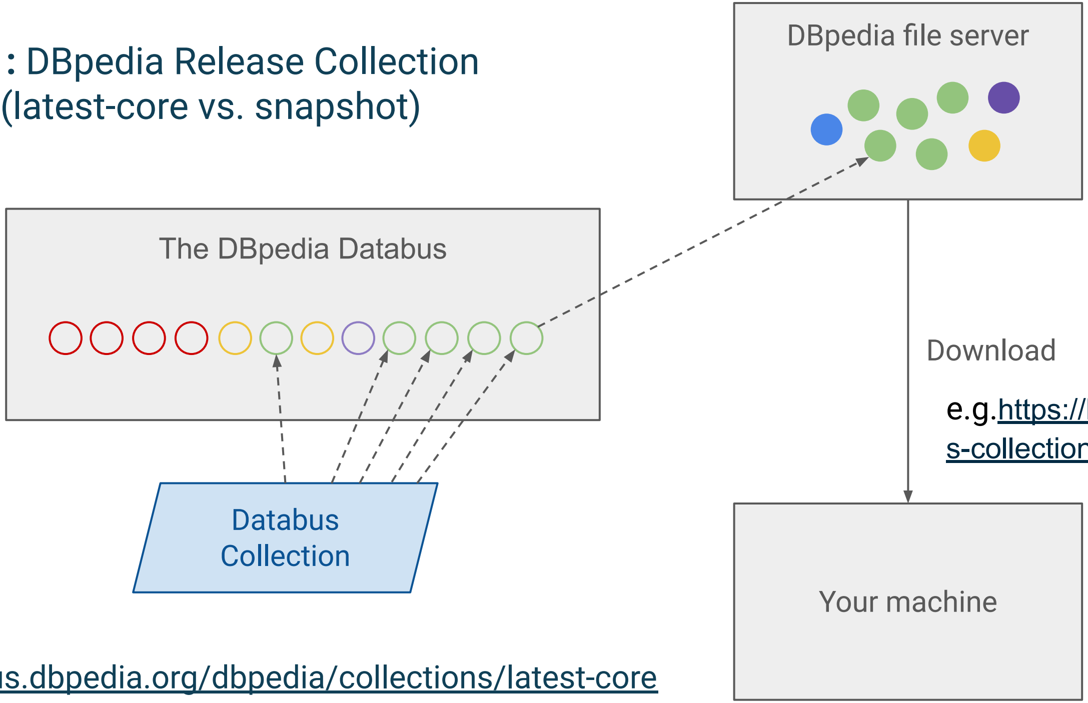


Index of /repo/dbpedia/

```
../
generic/                                           18-Oct-2021 13:29
mappings/                                          21-Mar-2020 13:27
spotlight/                                         25-Mar-2020 22:47
text/                                              26-Mar-2020 08:15
transition/                                        18-Dec-2020 14:22
wikidata/                                          03-May-2020 11:52
```

How to access the DBpedia KG?

Option C1: DBpedia Release Collection
(latest-core vs. snapshot)



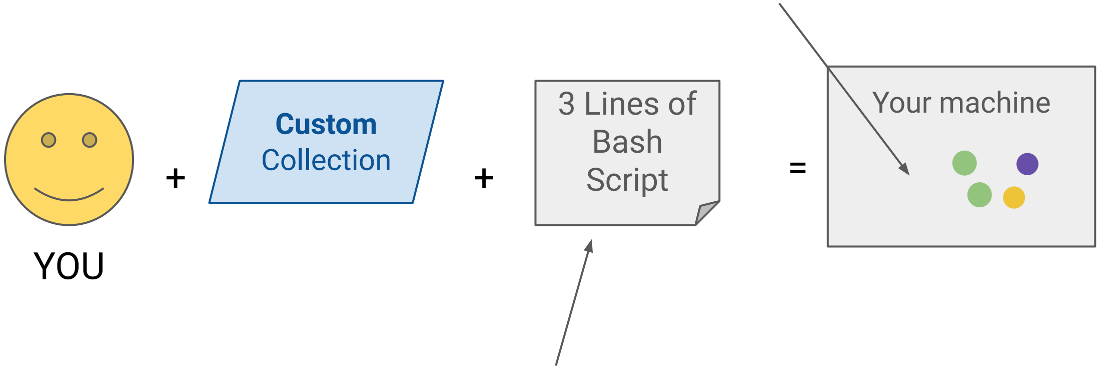
e.g.https://hub.docker.com/r/dbpedia/dbpedia-databus-collection-downloader

Core Release:
https://databus.dbpedia.org/dbpedia/collections/latest-core
Snapshot Release:
https://databus.dbpedia.org/dbpedia/collections/dbpedia-snapshot-2021-09/

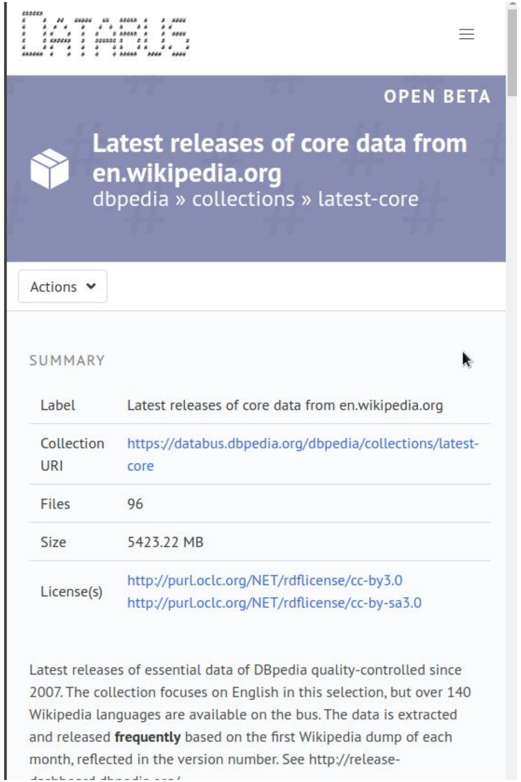
How to access the DBpedia KG?

Option C2: Copy and modify a DBpedia Release Collection

Your custom subset of DBpedia KG you need for the task at hand



or a comparable amount of LOC in any programming language



Latest releases of core data from en.wikipedia.org
dbpedia » collections » latest-core

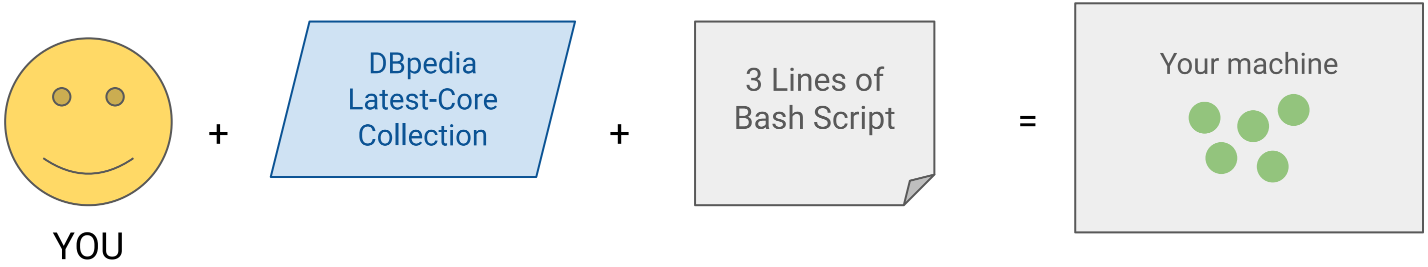
SUMMARY

Label	Latest releases of core data from en.wikipedia.org
Collection URI	https://databus.dbpedia.org/dbpedia/collections/latest-core
Files	96
Size	5423.22 MB
License(s)	http://purl.oclc.org/NET/rdflicense/cc-by3.0 http://purl.oclc.org/NET/rdflicense/cc-by-sa3.0

Latest releases of essential data of DBpedia quality-controlled since 2007. The collection focuses on English in this selection, but over 140 Wikipedia languages are available on the bus. The data is extracted and released **frequently** based on the first Wikipedia dump of each month, reflected in the version number. See http://release-

How to access the DBpedia KG?

Option CX: Download Collection: via command line



YOU + DBpedia Latest-Core Collection + 3 Lines of Bash Script = Your machine

Resolve the Collection ID (retrieve SPARQL query) via *curl*

```
query=$(curl -kH "Accept:text/sparql" https://databus.dbpedia.org/dbpedia/collections/dbpedia-snapshot-2021-09)
```

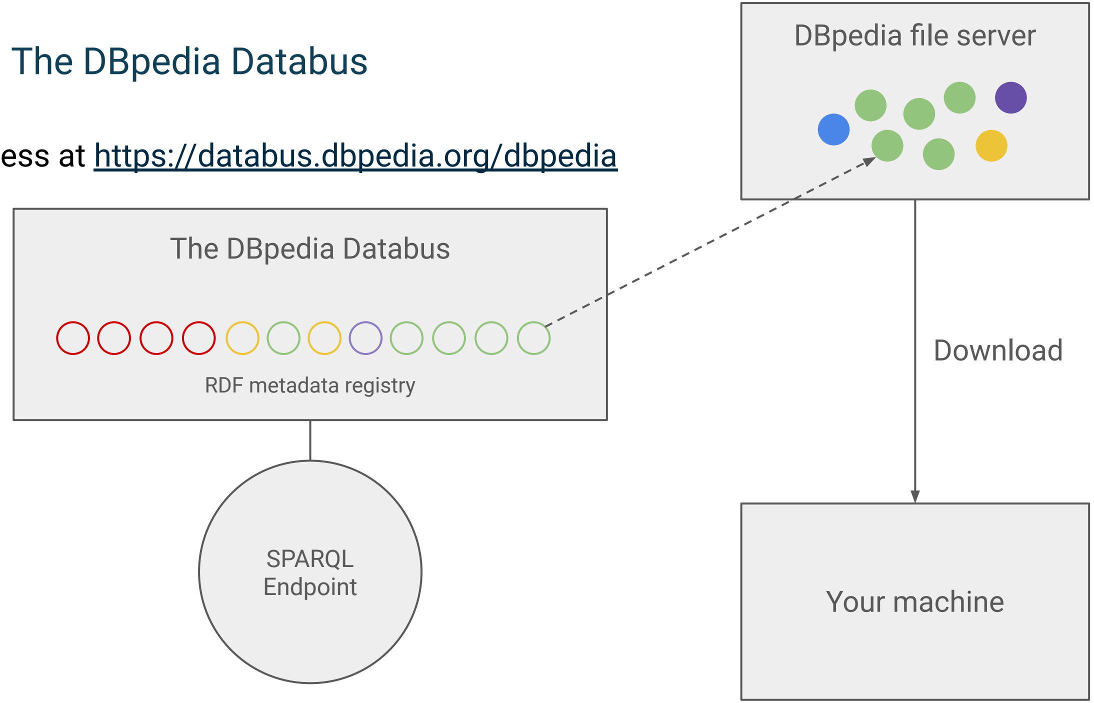
Download the files

```
files=$(curl -kH "Accept: text/csv" --data-urlencode "query=${query}" https://databus.dbpedia.org/repo/sparql | tail -n+2 | sed 's/"//g')

while IFS= read -r file ; do echo wget $file; done <<< "$files"
```

How to access the DBpedia KG?

Option D: The DBpedia Databus

Access at https://databus.dbpedia.org/dbpedia



Access at https://databus.dbpedia.org/repo/sparql

a)

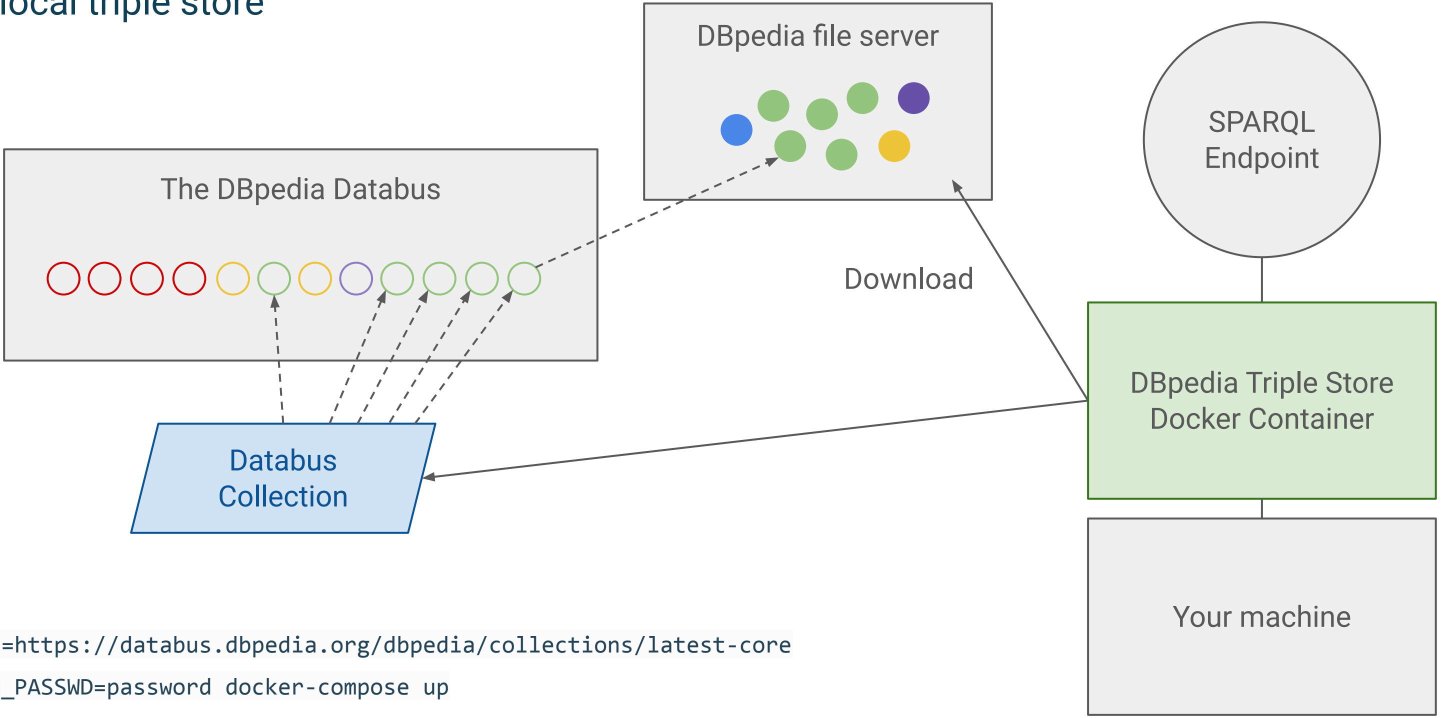
- Browse the Databus UI
- collect the Databus File IDs

b)

- Use the SPARQL endpoint with fine-grained SPARQL query and retrieve the Databus File IDs

How to access the DBpedia KG?

Option E: local triple store

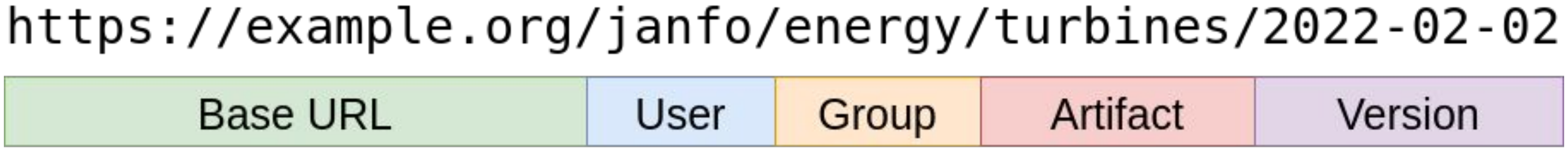


```
COLLECTION_URI=https://databus.dbpedia.org/dbpedia/collections/latest-core
VIRTUOSO_ADMIN_PASSWD=password docker-compose up
```

https://hub.docker.com/r/dbpedia/virtuoso-sparql-endpoint-quickstart

DBpedia Databus Identifiers

- The Databus offers a clean identifier structure:



`https://example.org/janfo/energy/turbines/2022-02-02`

- Enables queries such as:
  - “Give me all the versions in a group”
  - “Give me the latest version on an artifact”
- Version metadata enables even more fine-grained retrieval

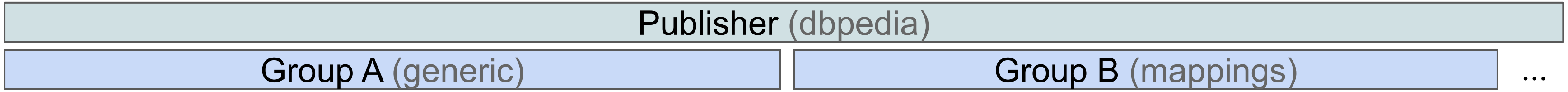
Navigating the DBpedia KG dump structure

Databus assets are structured hierarchically (similar to maven repositories).

DBpedia KG is published by ***dbpedia** Databus Account*

Groups are derived from respective Extraction module:

- **generic:** Generic Extraction
- **mappings:** Mapping-based Extraction
- **text:** Text Extraction
- **wikidata:** Wikidata Extraction



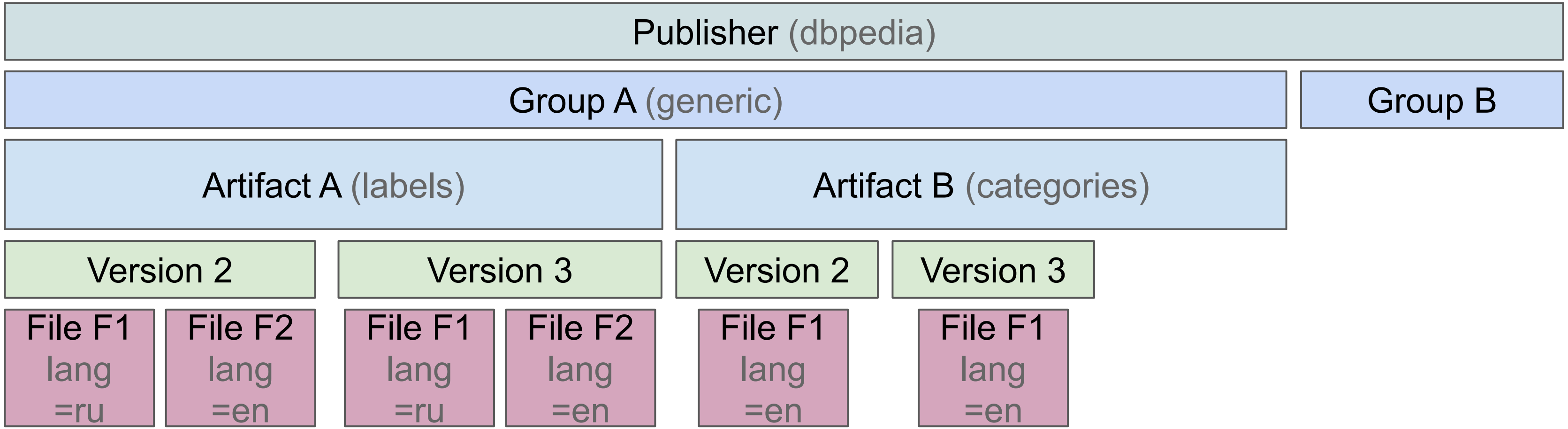
Publisher (dbpedia)

Group A (generic) | Group B (mappings) | ...

Navigating the DBpedia KG dump structure

A *Databus Group* bundles multiple *Artifacts* (abstract dataset)

Can contain files for different *versions* in different *formats*, *compressions*, and content *variants*



Databus Identifier Structure

- Databus URL: https://databus.dbpedia.org
- DBpedia Account URI https://databus.dbpedia.org/dbpedia
- Group URI: https://databus.dbpedia.org/dbpedia/generic
- Artifact URI https://databus.dbpedia.org/dbpedia/generic/labels
- Version URI https://databus.dbpedia.org/dbpedia/generic/labels/2022.04.01

Prominent DBpedia KG Artifacts

- Labels
- Geo-Coordinates
- Instance Types
- Mapping-based Objects / Mapping-based Literals
- DBpedia Ontology
- SameAs Links
- Global IDs

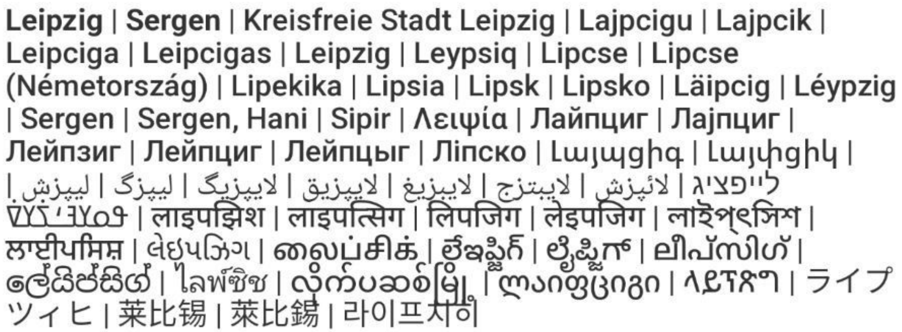
Extraction Artifact: Labels

ID: https://databus.dbpedia.org/dbpedia/generic/labels/

content variant dimensions: language

Example applications:

- Human readable names in more than 130 language versions
- Label-based indexing for search
- Creation of semantic dictionaries



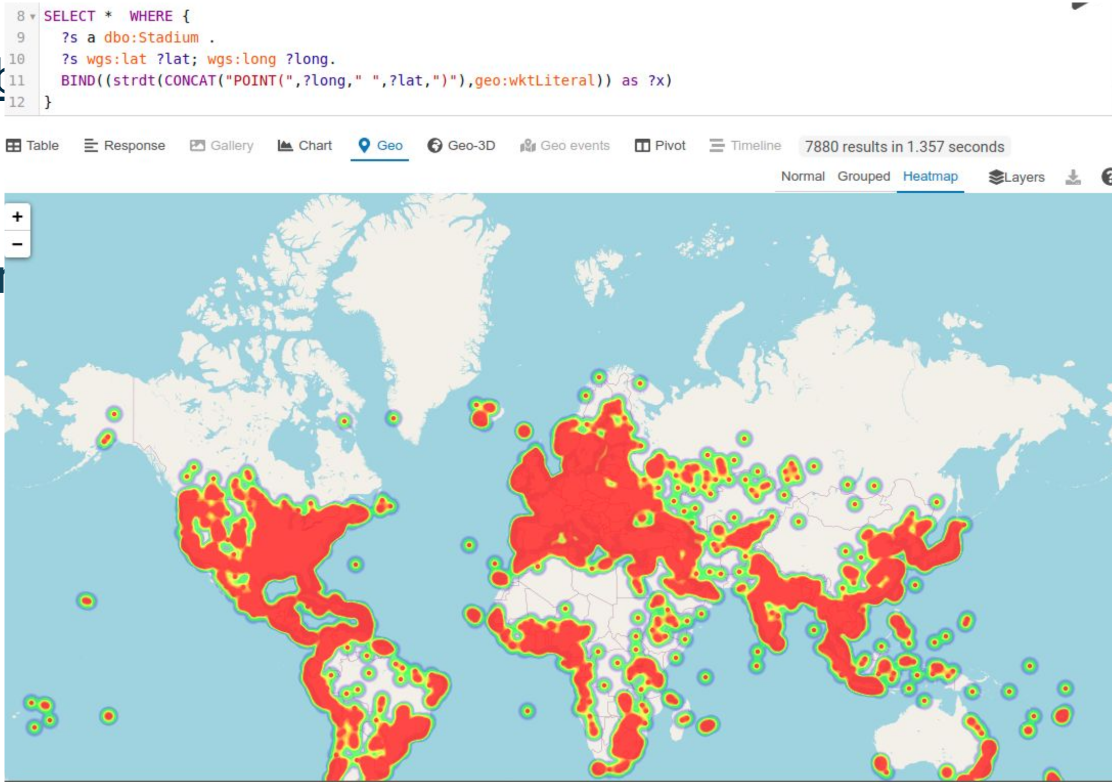
Leipzig | **Sergen** | **Kreisfreie Stadt Leipzig** | **Lajpcigu** | **Lajpcik** | **Leipciga** | **Leipcigas** | **Leipzig** | **Leypsiq** | **Lipcse** | **Lipcse (Németország)** | **Lipekika** | **Lipsia** | **Lipsk** | **Lipsko** | **Läipcig** | **Léypzig** | **Sergen** | **Sergen, Hani** | **Sipir** | **Λειψία** | **Лайпциг** | **Лајпциг** | **Лейпзиг** | **Лейпциг** | **Лейпцыг** | **Ліпско** | Լայպցիգ | Լայփցիկ |
לייפציג | لائپزش | لايبتزج | لايبزيغ | لايپزيق | لايپزيگ | ليپزگ | ليپزش |
ލައިޕްޒިގް | लाइपझिश | लाइपत्सिग | लिपजिग | लेइपजिग | লাইপ্‌ৎসিশ |
ਲਾਈਪਜ਼ਿਸ਼ | લેઇપઝિગ | லைப்சிக் | లైప్జిగ్ | ಲೈಪ್ಜಿಗ್ | ලීප්සිග් |
ലെയ്പ്സിഗ് | ไลพ์ซิช | လိုက်ပဇစ်မြို့ | ლაიფციგი | ላይፕሲግ | ライプツィヒ | 莱比锡 | 萊比錫 | 라이프치히

Extraction Artifact: Geo-Coordinates

ID: https://databus.dbpedia.org/db

Example applications:

- Area or neighborhood-based r
- Map visualizations



```
SELECT *  WHERE {
  ?s a dbo:Stadium .
  ?s wgs:lat ?lat; wgs:long ?long.
  BIND((strdt(CONCAT("POINT(",?long," ",?lat,")"),geo:wktLiteral)) as ?x)
}
```

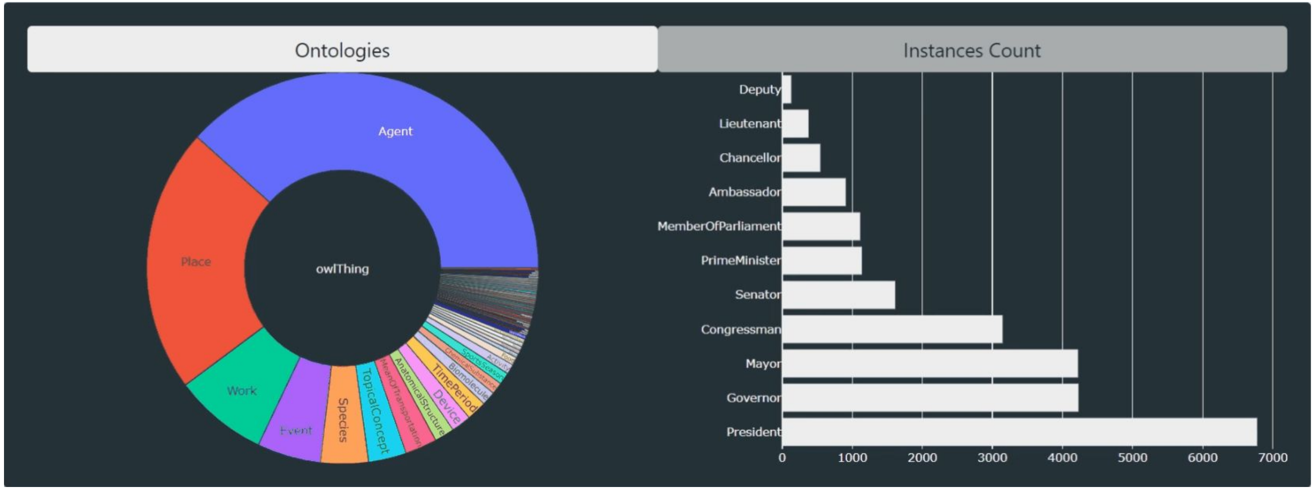
7880 results in 1.357 seconds

Extraction Artifact: Instance Types

ID: https://databus.dbpedia.org/dbpedia/mappings/instance-types/

special content variants: *specific* and *transitive*

Example applications: Filter entities by Type/Class using [DBpedia Ontology]()



Extraction Artifact: Mapping-based Objects/Literals

ID: https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects/

special content variants: *disjointDomain* and *disjointRange*

https://databus.dbpedia.org/dbpedia/mappings/mappingbased-literals/

“fact base” of the DBpedia KG containing object property / datatype property statements (numeric measurements are normalized to SI base units)

Example applications:

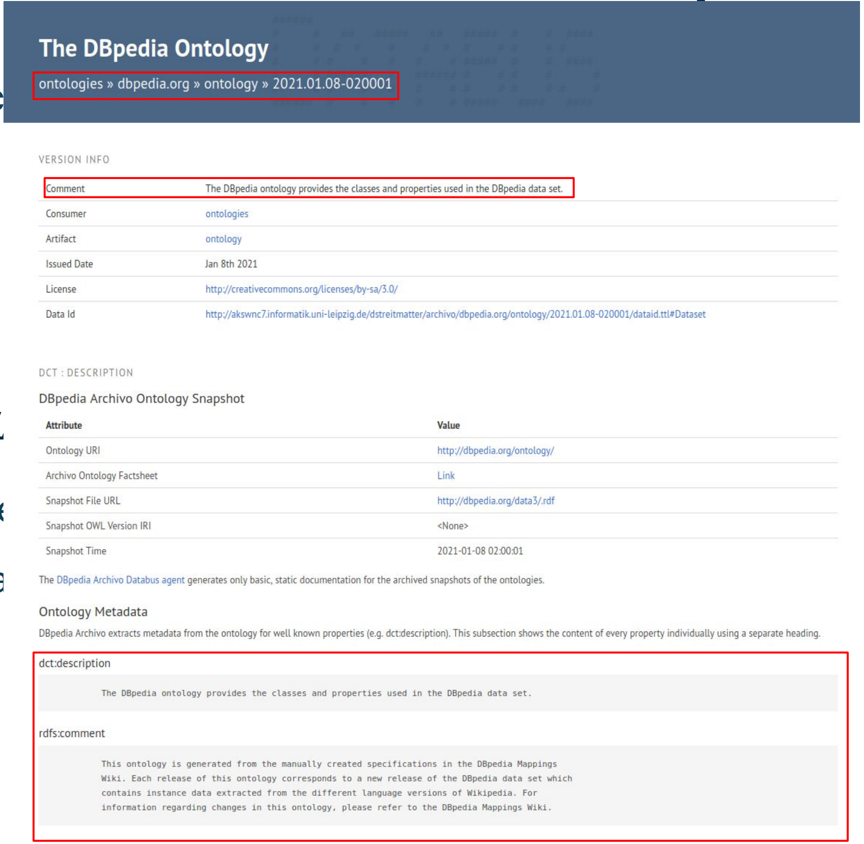
- unified querying for facts and entity relation across language versions using DBpedia ontology

DBpedia Ontology Artifact

- DBpedia Archivo uses the Databus for persistent arc
- dedicated **ontologies** publisher on the Databus
- Uses the ontology IRI for identification:
  - publisher → *ontologies*
  - group → domain of the ontology
  - artifact → path of IRI (suffix --DEV for dev ont.)
  - version → timestamp of discovery/update

https://databus.dbpedia.org/ontologies/dbpedia.org/ontology--DEV

- persistent, unified versioning & archiving of ontologie
- access archived metadata via SPARQL / Linked Data



The DBpedia Ontology

ontologies » dbpedia.org » ontology » 2021.01.08-020001

VERSION INFO

Comment	The DBpedia ontology provides the classes and properties used in the DBpedia data set.
Consumer	ontologies
Artifact	ontology
Issued Date	Jan 8th 2021
License	http://creativecommons.org/licenses/by-sa/3.0/
Data Id	http://akswnc7.informatik.uni-leipzig.de/dstreitmatter/archivo/dbpedia.org/ontology/2021.01.08-020001/dataid.ttl#Dataset

DCT : DESCRIPTION

DBpedia Archivo Ontology Snapshot

Attribute	Value
Ontology URI	http://dbpedia.org/ontology/
Archivo Ontology Factsheet	Link
Snapshot File URL	http://dbpedia.org/data3/.rdf
Snapshot OWL Version IRI	<None>
Snapshot Time	2021-01-08 02:00:01

The DBpedia Archivo Databus agent generates only basic, static documentation for the archived snapshots of the ontologies.

Ontology Metadata

DBpedia Archivo extracts metadata from the ontology for well known properties (e.g. dct:description). This subsection shows the content of every property individually using a separate heading.

dct:description

```
The DBpedia ontology provides the classes and properties used in the DBpedia data set.
```

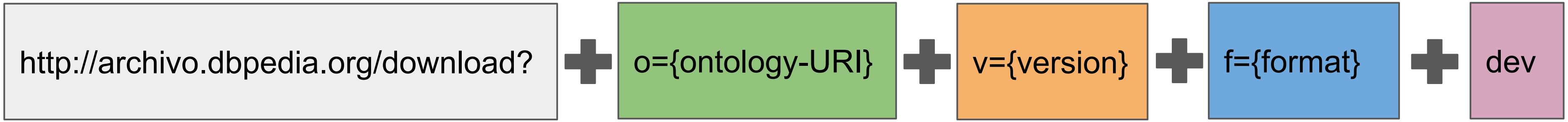
rdfs:comment

```
This ontology is generated from the manually created specifications in the DBpedia Mappings
Wiki. Each release of this ontology corresponds to a new release of the DBpedia data set which
contains instance data extracted from the different language versions of Wikipedia. For
information regarding changes in this ontology, please refer to the DBpedia Mappings Wiki.
```

DBpedia Ontology: Archivo download API

- abstract identifiers for ontologies
- various parsed serialisations: RDF+XML, Turtle and N-Triples
- persistent snapshots of any ontology version

One REST request to fetch version of an ontology:



http://archivo.dbpedia.org/download? + o={ontology-URI} + v={version} + f={format} + dev

Fetch the latest DBpedia DEV ontology
http://archivo.dbpedia.org/download?o=http://dbpedia.org/ontology/&f=ttl&dev →

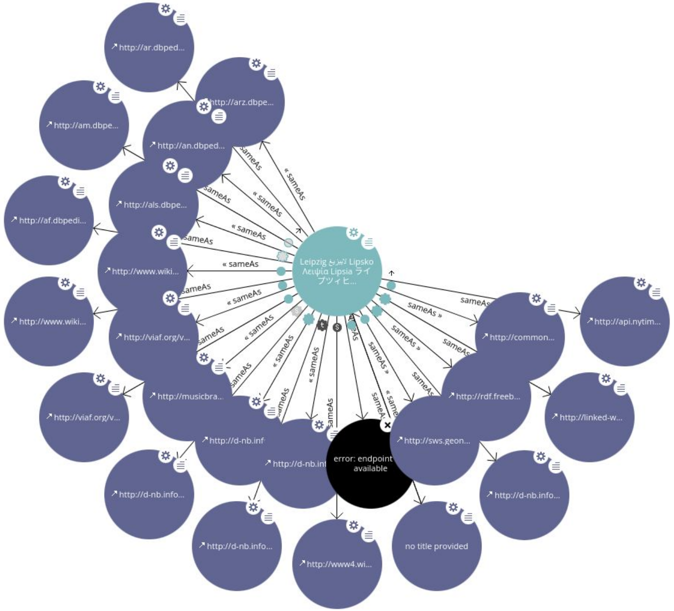
Extraction Artifact: SameAs links

ID: https://databus.dbpedia.org/dbpedia/wikidata/sameas-all-wikis/

https://databus.dbpedia.org/dbpedia/wikidata/sameas-external/

Wikidata → DBpedia links

Wikidata → external links



Example applications:

- federated querying of entities
- knowledge fusion
- popularity indicator

DBpedia Global Identity Management

ID:https://databus.dbpedia.org/jj-author/id-management/global-ids/

- Entity clustering information (connected components) based on owl:sameAs links
- Assigns Global Identifiers for a cluster based on lowest cluster member ID
- allow to discover known references to other datasets for same thing
- microservice which looks up cluster information for whitelisted known namespaces (including all DBpedia + Wikidata, amongst others)

```
global: "https://global.dbpedia.org/id/2gtFW",
 locals:
 [
                   "http://www.wikidata.org/entity/Q29032648 ",
                   "http://nl.dbpedia.org/resource/FAIR_data ",
                   "http://dbpedia.org/resource/FAIR_data ",
                   "http://fr.dbpedia.org/resource/Fair_data ",
                   "http://pt.dbpedia.org/resource/Dados_FAIR ",
                   "http://ca.dbpedia.org/resource/FAIR ",
 ],
 cluster:
 [
                   "2gtFW",
                   "6rGeQ",
                   "6xyfG",
                   "7TyKK",
                   "9M7Pp",
                   "9NpGc",
]
```

How to access the DBpedia KG?

Option F: Access through (public/self-deployed) DBpedia Services

- DBpedia Spotlight
  - Access at https://demo.dbpedia-spotlight.org/
  - Allows entity annotation of text

- DBpedia Lookup
  - Access at https://lookup.dbpedia.org
  - Keyword search and entity retrieval on DBpedia data
  - Dedicated prefix search for fast auto-complete in apps
  - Example queries:
    - https://lookup.dbpedia.org/api/search?query=Leipzig&maxResults=1
    - https://lookup.dbpedia.org/api/search?query=Leipzig&typeName=Organisation&maxResults=1

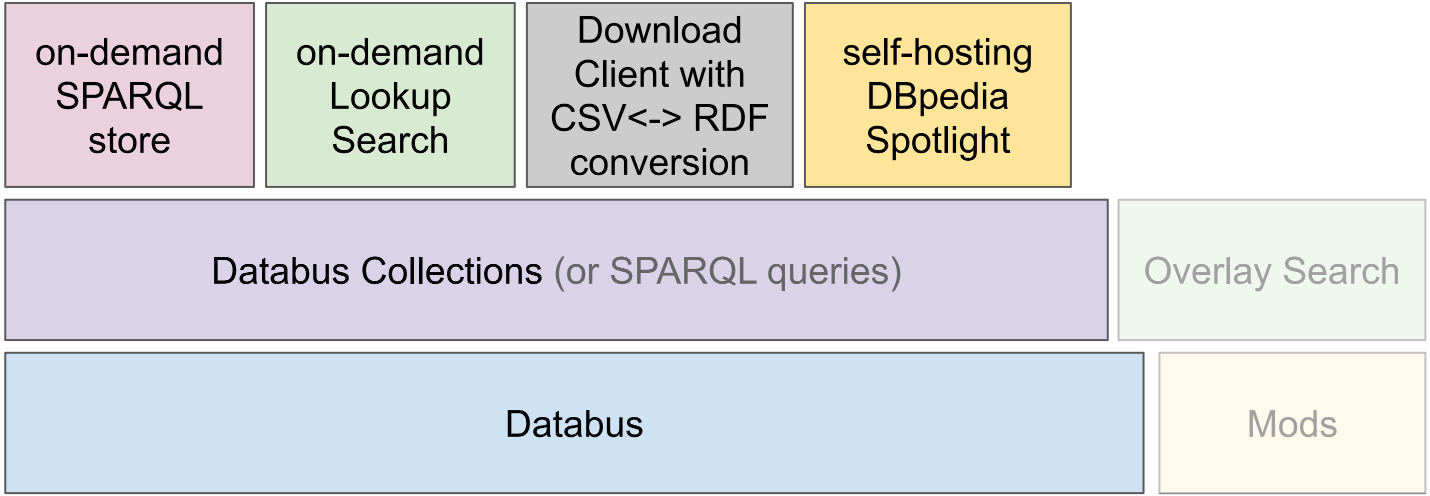
Demo

- Browse Latest-Core collection and create a custom copy

Self-deploy Services using Docker-Compose

The Databus Technology Stack

- a range of applications and services based on the Databus
- allows you to
  - deploy DBpedia services on your own infrastructure (no quota, full control)
  - on-demand services work with ***any Collection of RDF data*** out of the box in a generic + can be easily configured to work with other datasets



on-demand SPARQL store

on-demand Lookup Search

Download Client with CSV<-> RDF conversion

self-hosting DBpedia Spotlight

Databus Collections (or SPARQL queries)

Overlay Search

Databus

Mods

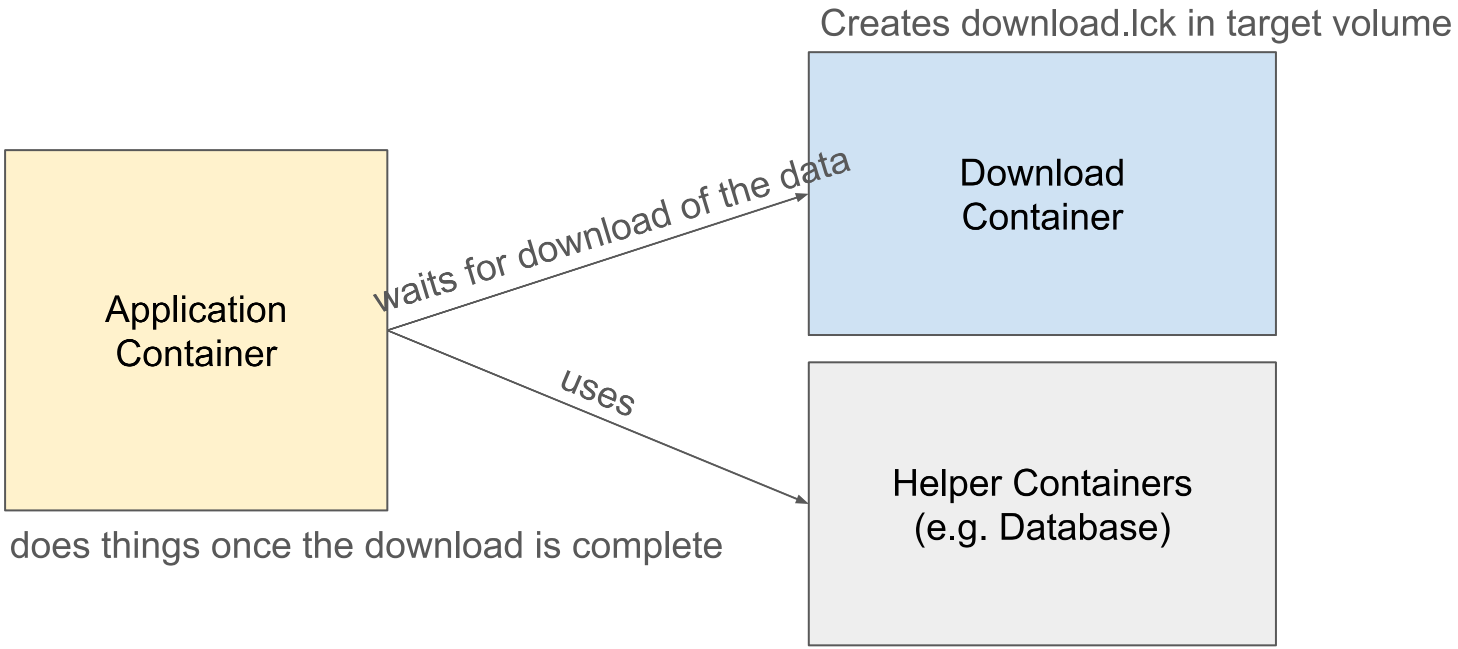
Databus Deploy Demo

- configuration
    - OIDC provider - Identity provider (e.g. Keycloak, Azure Active Directory, Auth0)
    - HTTPS proxy and (DNS) namespace

- consists of 3 docker services/containers with several components:
    - SPARQL graph store (currently supported Virtuoso / Jena)
    - G-store (git versioning of metadata, + graph store sync for most recent version)
    - Databus frontend and middleware
        - Web Interface
        - User management and logic
        - Lookup for Index on metadata
        - Swagger API
    -

Dockerized Databus Applications

Apps are run using docker-compose. Containers share data via volumes.



Creates download.lck in target volume

does things once the download is complete

Application Container
waits for download of the data
uses
Download Container
Helper Containers (e.g. Database)

On-demand SPARQL store

Visit https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart

Composite of:

- Download Container
- Openlink Virtuoso Container (Helper Container)
- DBpedia Virtuoso Loader Container (Application Container)

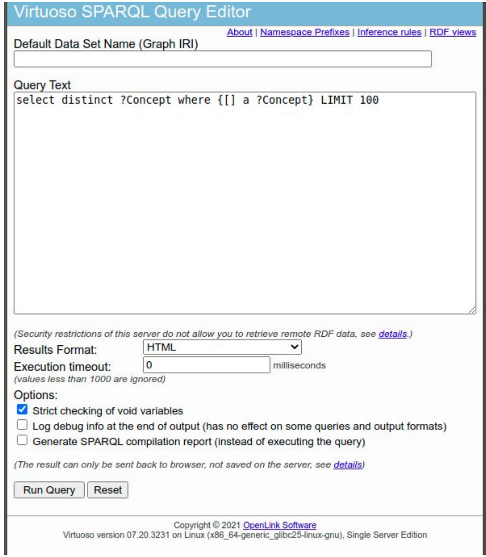
On-demand SPARQL store

Load **any** Databus Collection automatically into a Virtuoso SPARQL Store

```
git clone https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart.git
cd virtuoso-sparql-endpoint-quickstart

COLLECTION_URI=https://databus.dbpedia.org/jan/collections/more-cities/
VIRTUOSO_ADMIN_PASSWD=YourSecretPassword docker-compose up
```

Use one of the DBpedia Release Collections or your custom fork to deploy the DBpedia KG locally

collection can be passed via environment variable or in compose setup



Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see details.)

Results Format: HTML

Execution timeout: 0 milliseconds

(values less than 1000 are ignored)

Options:

☑ Strict checking of void variables

□ Log debug info at the end of output (has no effect on some queries and output formats)

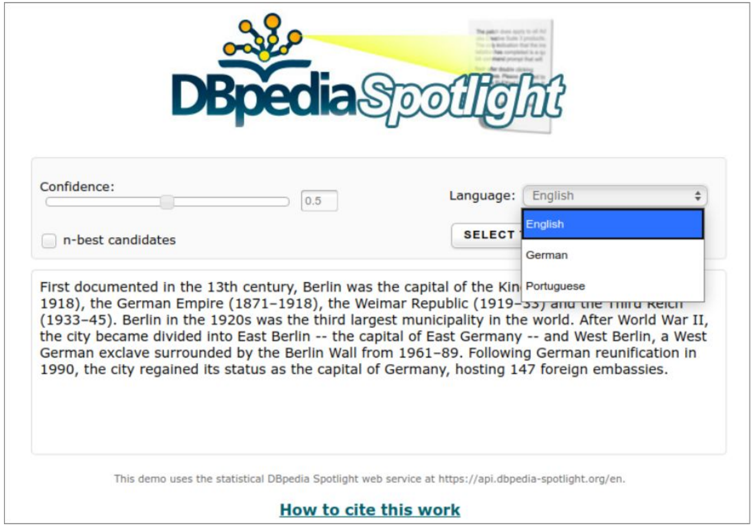
□ Generate SPARQL compilation report (instead of executing the query)

(The result can only be sent back to browser, not saved on the server, see details)

Copyright © 2021 OpenLink Software

Virtuoso version 07.20.3231 on Linux (x86_64-generic_glibc25-linux-gnu), Single Server Edition

DBpedia Spotlight



Confidence: 0.5

□ n-best candidates

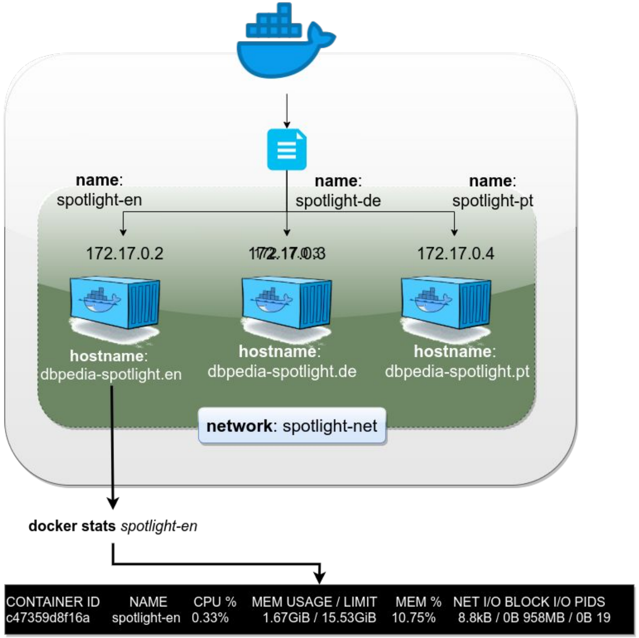
Language: English

First documented in the 13th century, Berlin was the capital of the Kin 1918), the German Empire (1871–1918), the Weimar Republic (1919–33) and the Third Reich (1933–45). Berlin in the 1920s was the third largest municipality in the world. After World War II, the city became divided into East Berlin -- the capital of East Germany -- and West Berlin, a West German exclave surrounded by the Berlin Wall from 1961–89. Following German reunification in 1990, the city regained its status as the capital of Germany, hosting 147 foreign embassies.

This demo uses the statistical DBpedia Spotlight web service at https://api.dbpedia-spotlight.org/en.

How to cite this work

a) DBpedia Spotlight web application to annotate text for three different languages: English, German and Portuguese



name: spotlight-en
name: spotlight-de
name: spotlight-pt
172.17.0.2
172.17.0.3
172.17.0.4
hostname: dbpedia-spotlight.en
hostname: dbpedia-spotlight.de
hostname: dbpedia-spotlight.pt
network: spotlight-net
docker stats spotlight-en

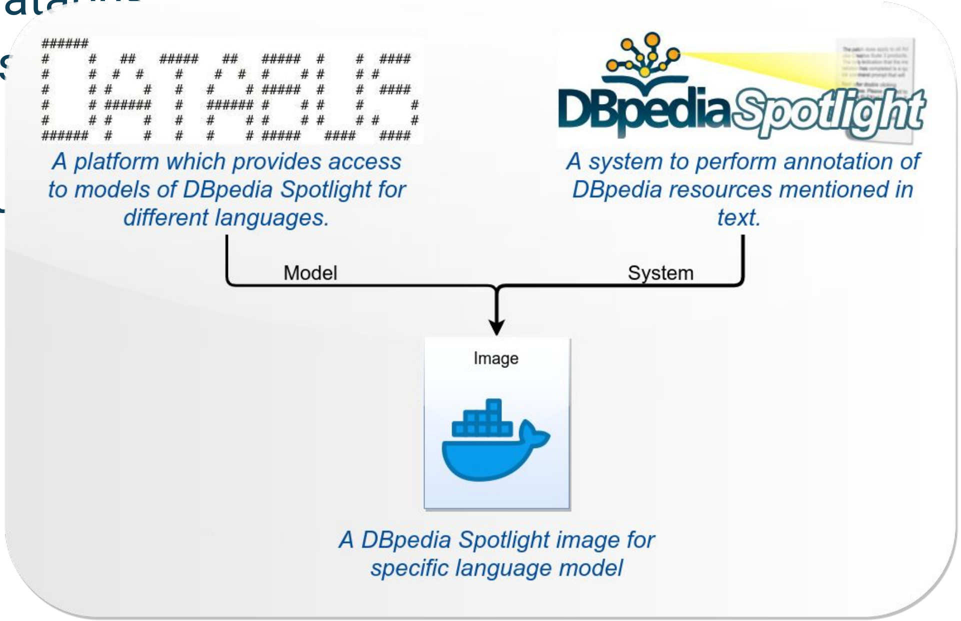
```
CONTAINER ID     NAME        CPU %    MEM USAGE / LIMIT    MEM %    NET I/O BLOCK I/O PIDS
c47359d8f16a   spotlight-en  0.33%    1.67GiB / 15.53GiB   10.75%    8.8kB / 0B 958MB / 0B 19
```

b) Three Docker containers running the DBpedia Spotlight service for the English, German and Portuguese languages.

https://demo.dbpedia-spotlight.org/

DBpedia Spotlight

- language models are managed on Databus
- but no generic/on-demand setup bas
  models are trained on data features
- more complex Docker compose setu



A platform which provides access to models of DBpedia Spotlight for different languages.

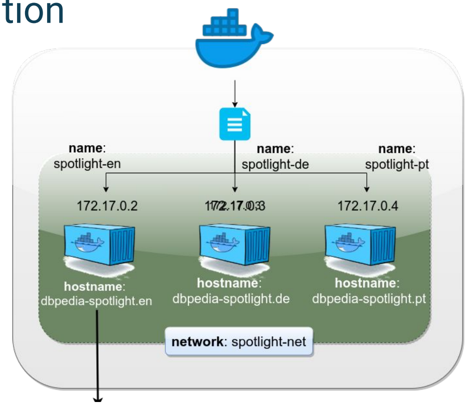
A system to perform annotation of DBpedia resources mentioned in text.

A DBpedia Spotlight image for specific language model

https://databus.dbpedia.org/dbpedia/spotlight/

DBpedia Spotlight

- Setup DBpedia Spotlight Multilingual
  - Create a volume:
    - `docker volume create spotlight-models`
  - Create a Docker network:
    - `docker network create spotlight-net`
  - Download the example docker compose and sites.xml files
    - The *docker compose* file contains the setup of the DBpedia Spotlight services and the web application.
    - The *sites.xml* file defines the models available on the web application
  - Run docker compose file
    - `docker-compose -f spotlight-compose.yml up -d`
  - Spotlight running at: http://localhost:2222/

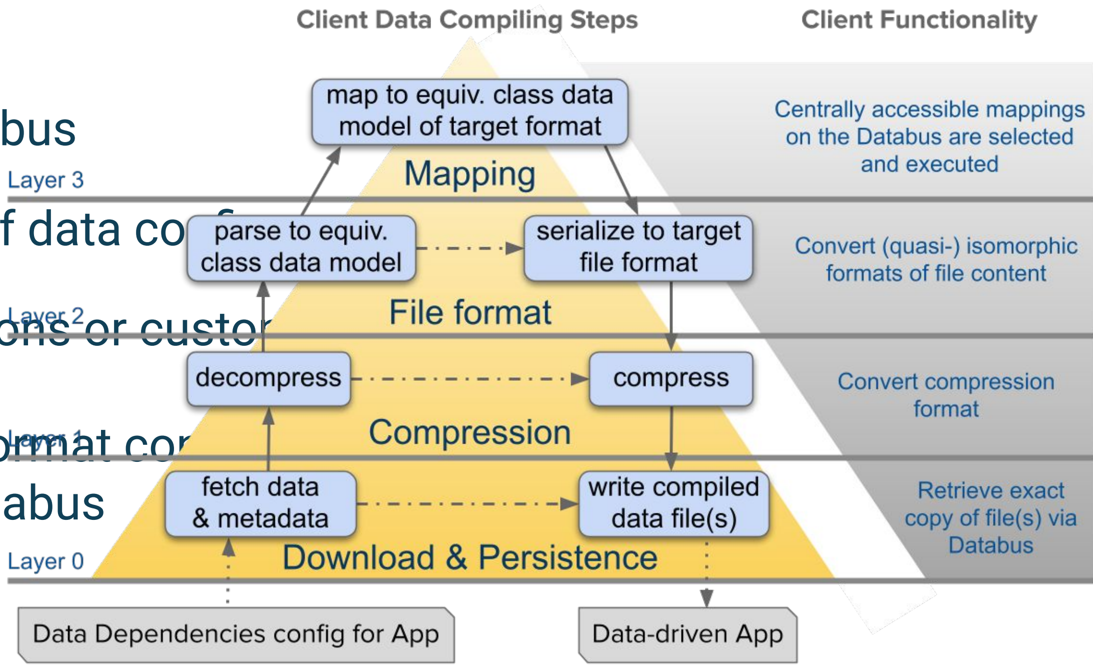


Databus Client

Enhanced consumer-oriented “download” client for DBpedia Databus

Vision: local “compile and install” of data co

- Download via Databus collections or custo
- Provenance tracking
- Download, compression and format co
- Mapping matchmaking via Databus



Client Data Compiling Steps
Client Functionality
map to equiv. class data model of target format
Mapping
Centrally accessible mappings on the Databus are selected and executed
Layer 3
parse to equiv. class data model
serialize to target file format
File format
Convert (quasi-) isomorphic formats of file content
Layer 2
decompress
compress
Compression
Convert compression format
Layer 1
fetch data & metadata
write compiled data file(s)
Download & Persistence
Retrieve exact copy of file(s) via Databus
Layer 0
Data Dependencies config for App
Data-driven App

```
bin/DatabusClient -f ttl -c gz -s <collection-ID>
```

DBpedia vs. Wikidata

Complementary but still different projects

- Wikidata not adopted to Wikipedia infoboxes
  - lost of workspace (47k editors vs 13k in Wikidata)
- Is Wikidata up-to-date?
  - some corona related values we found were/are over 1 year old
  - … likely only for stable values such as birth dates, but not for recent data
- Wikidata is growing
  - … but this would require a lot more editors to cover all that and keep it updated
  - similar problem with Freebase
- Live Updates via DBpedia Live
  - whenever something happens, in 30 min in Wikipedia, and then also in DBpedia Live
  - 2 Wikipedia edits every second!
- **DBpedia Global “beyond” Wikipedia**
  - link to recent and authoritative sources

Summary

- DBpedia Databus Platform
  - Upload data on the Databus
  - Work with Databus collections
- Consuming DBpedia via Databus
  - Dockerized DBpedia
  - Dockerized applications for the DBpedia Stack (Lookup, Spotlight)

DBpedia

Databus Mods & Overlay system