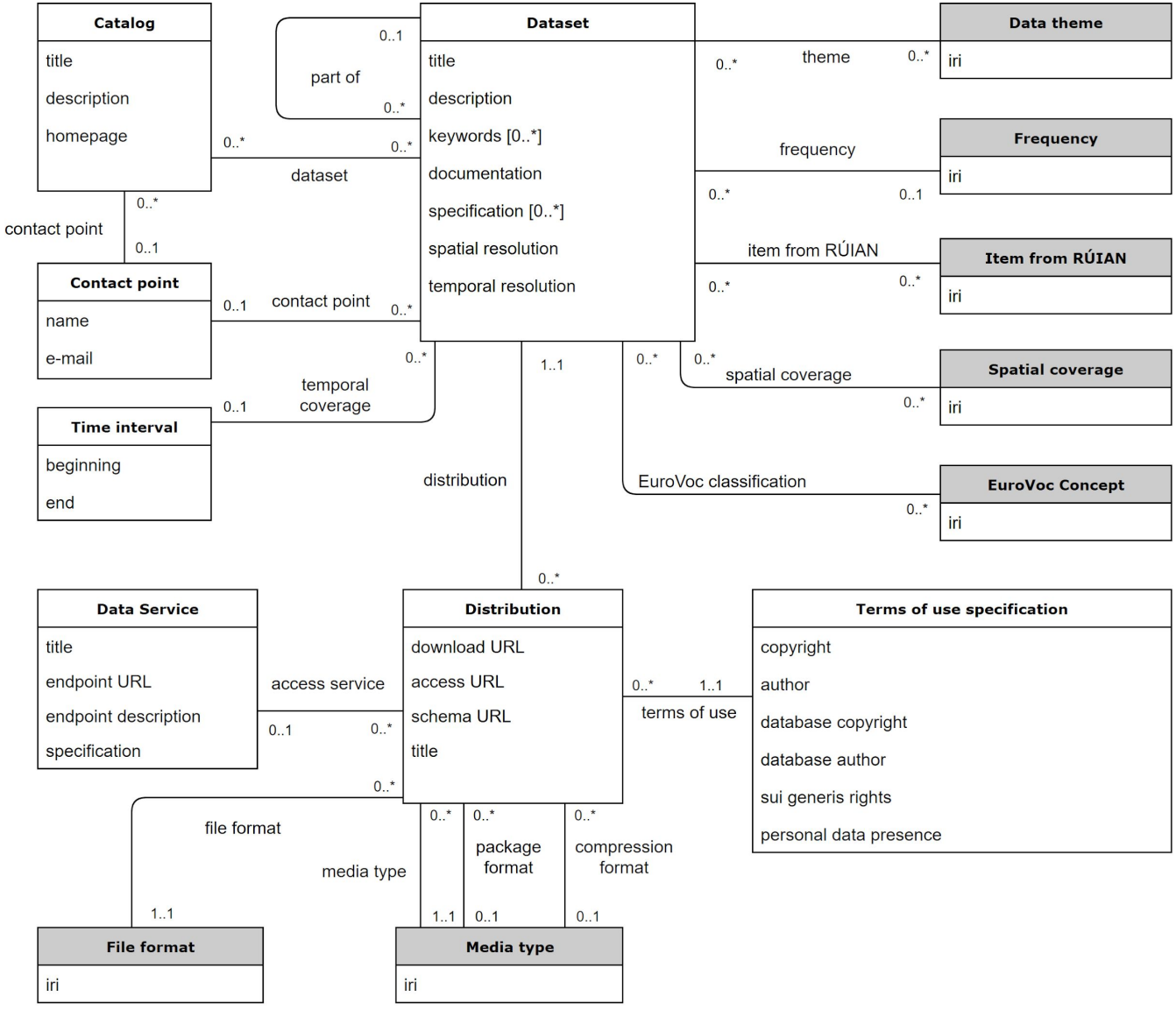
Graph data formats: RDF, RDFS, Linked Data

Jakub Klímek


Catalog
title
description
homepage
Dataset
title
description
keywords [0..*]
documentation
specification [0..*]
spatial resolution
temporal resolution
part of
0..1
0..*
dataset
contact point
Contact point
name
e-mail
Time interval
beginning
end
temporal coverage
distribution
Data theme
iri
theme
Frequency
iri
frequency
Item from RÚIAN
iri
item from RÚIAN
Spatial coverage
iri
spatial coverage
EuroVoc Concept
iri
EuroVoc classification
Data Service
title
endpoint URL
endpoint description
specification
access service
Distribution
download URL
access URL
schema URL
title
terms of use
Terms of use specification
copyright
author
database copyright
database author
sui generis rights
personal data presence
file format
media type
package format
compression format
File format
iri
Media type
iri

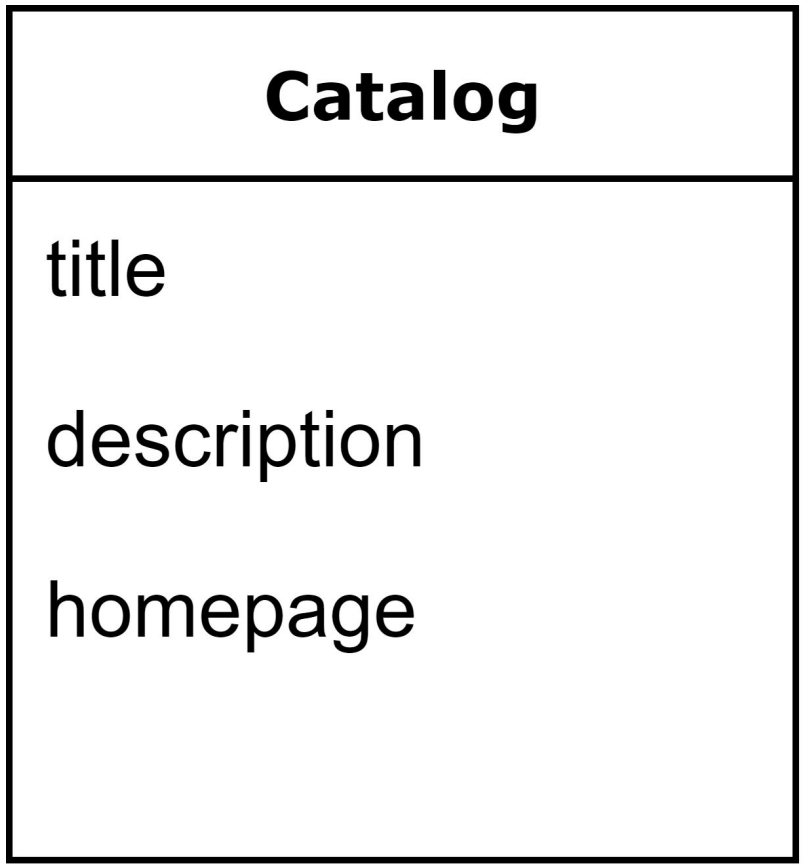
Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”



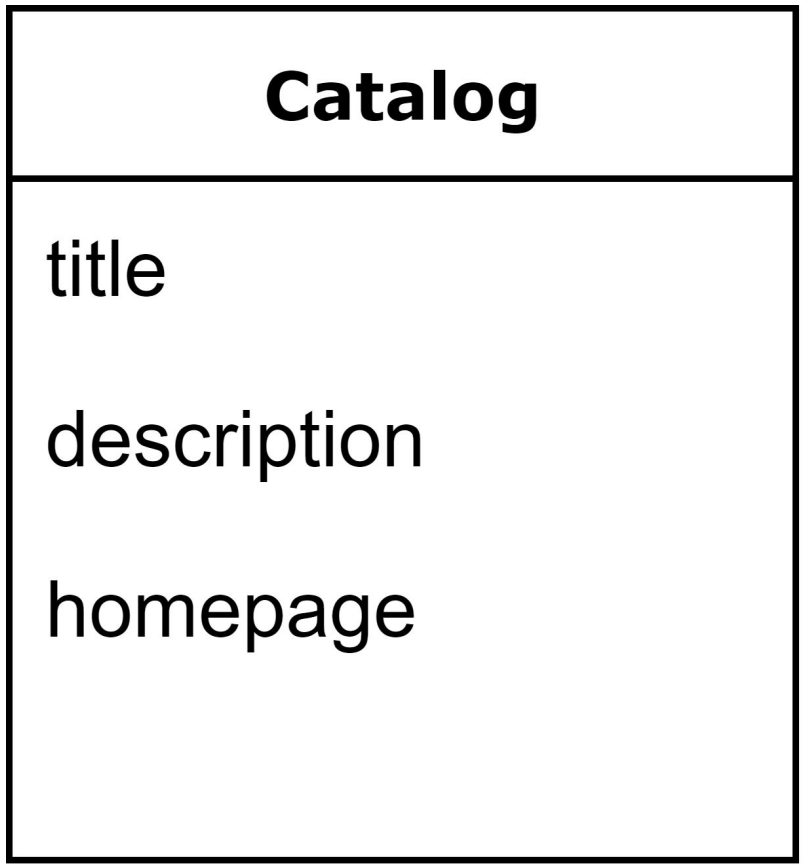
Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

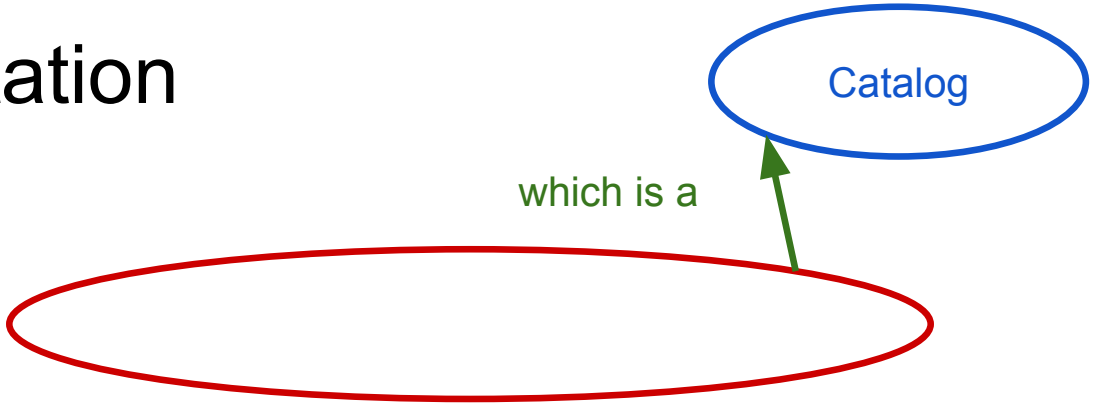
The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”



Graph data representation

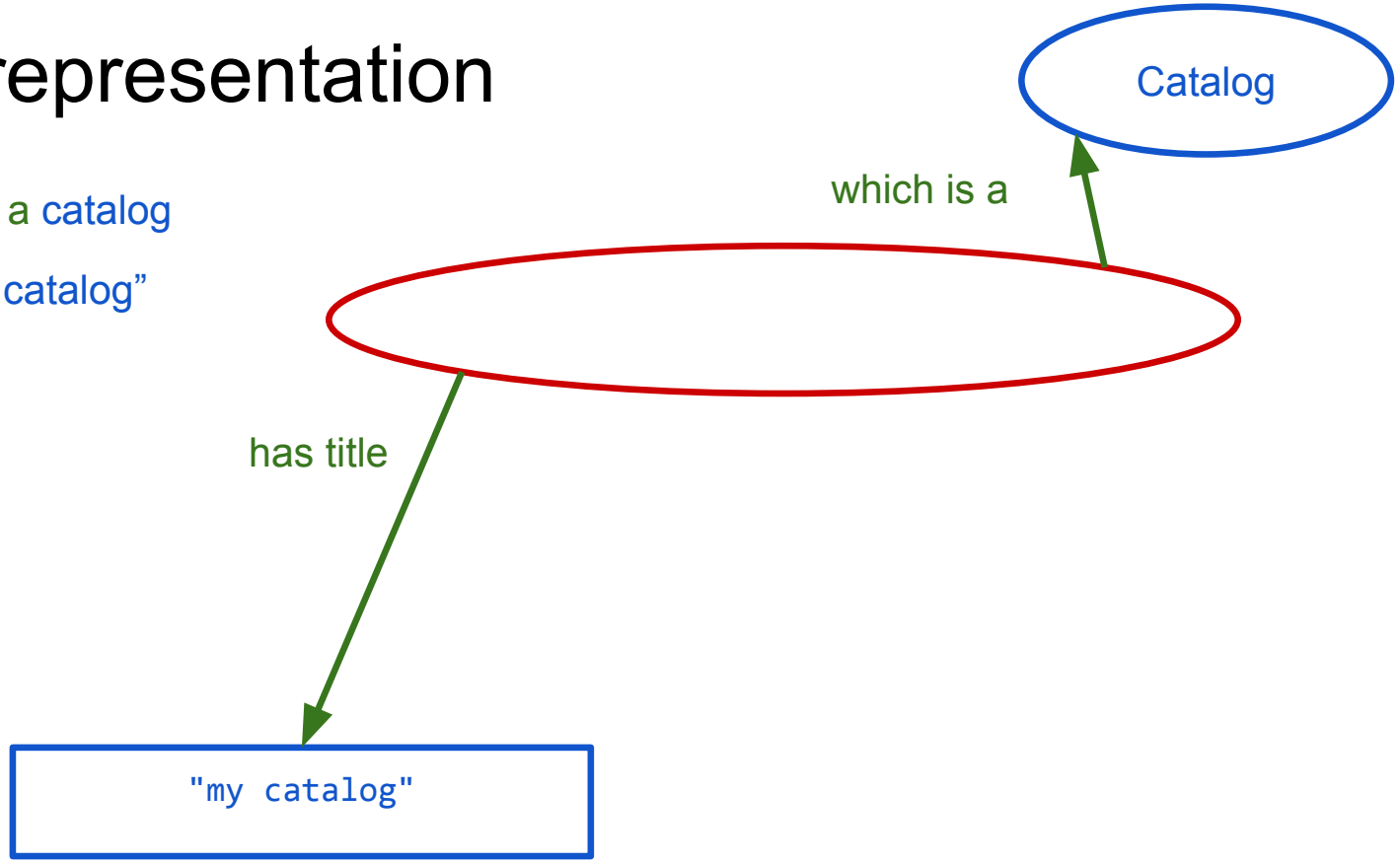
There is a thing, which is a catalog



Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

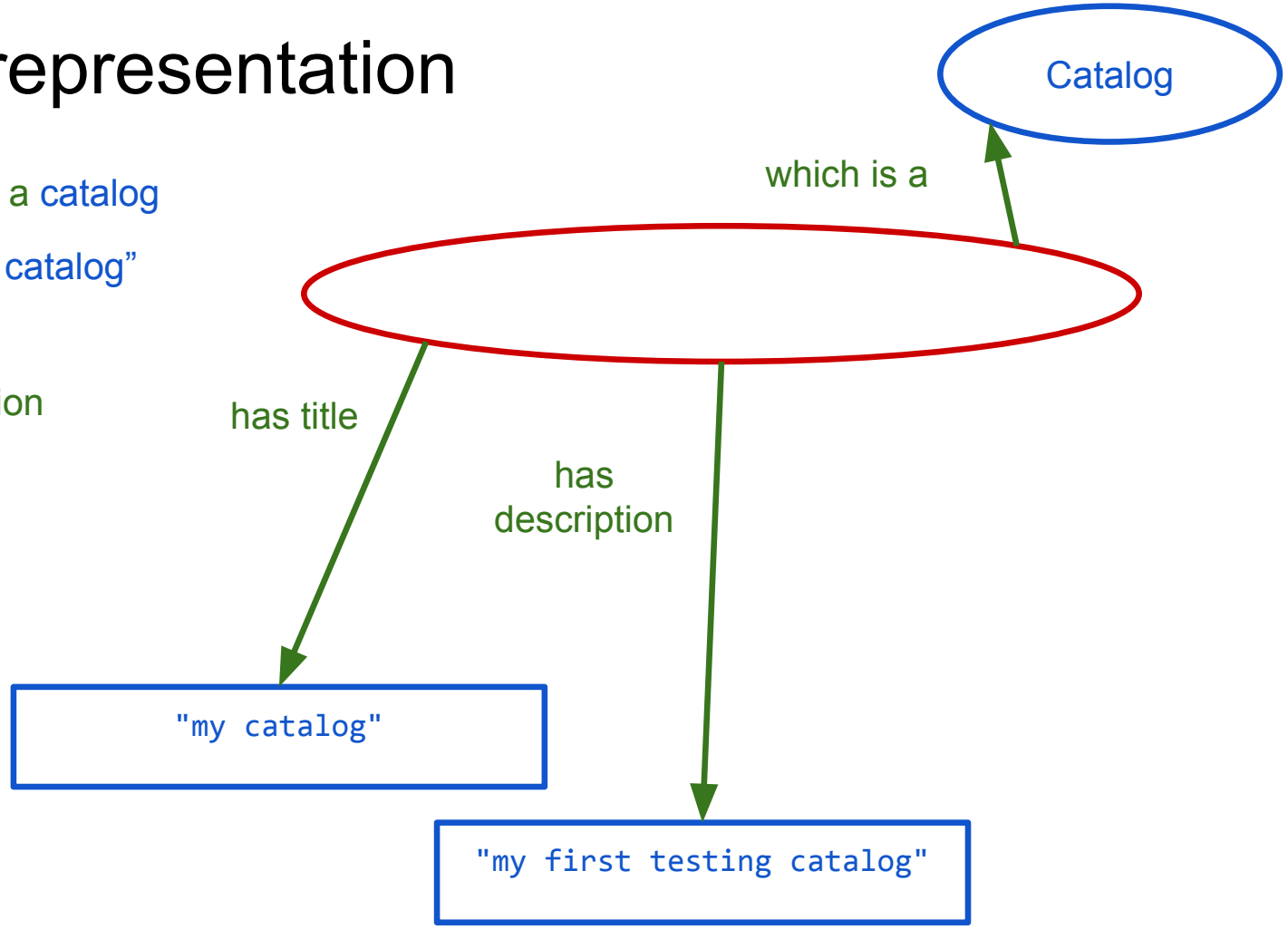


Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”



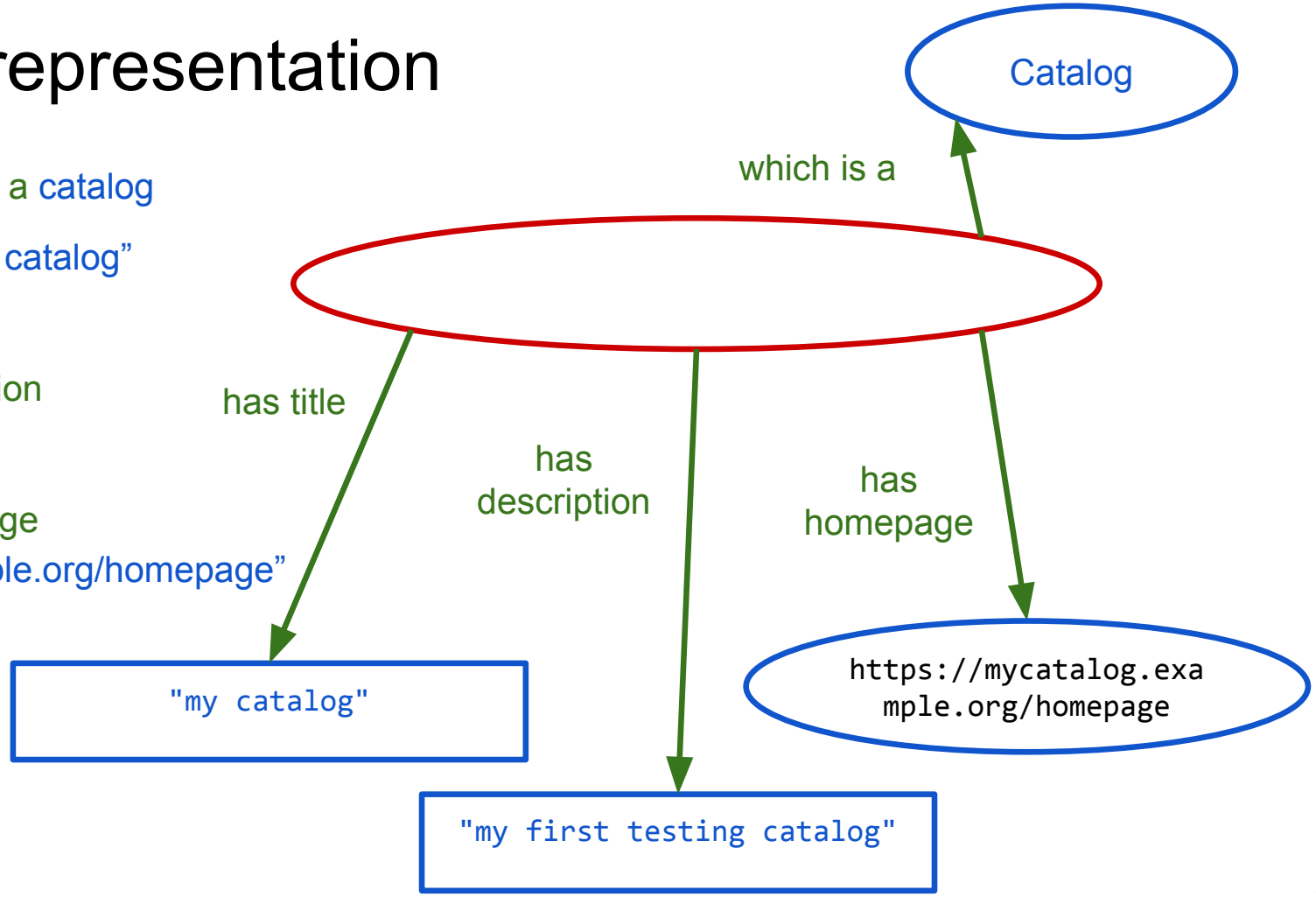
Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”



Graph data representation

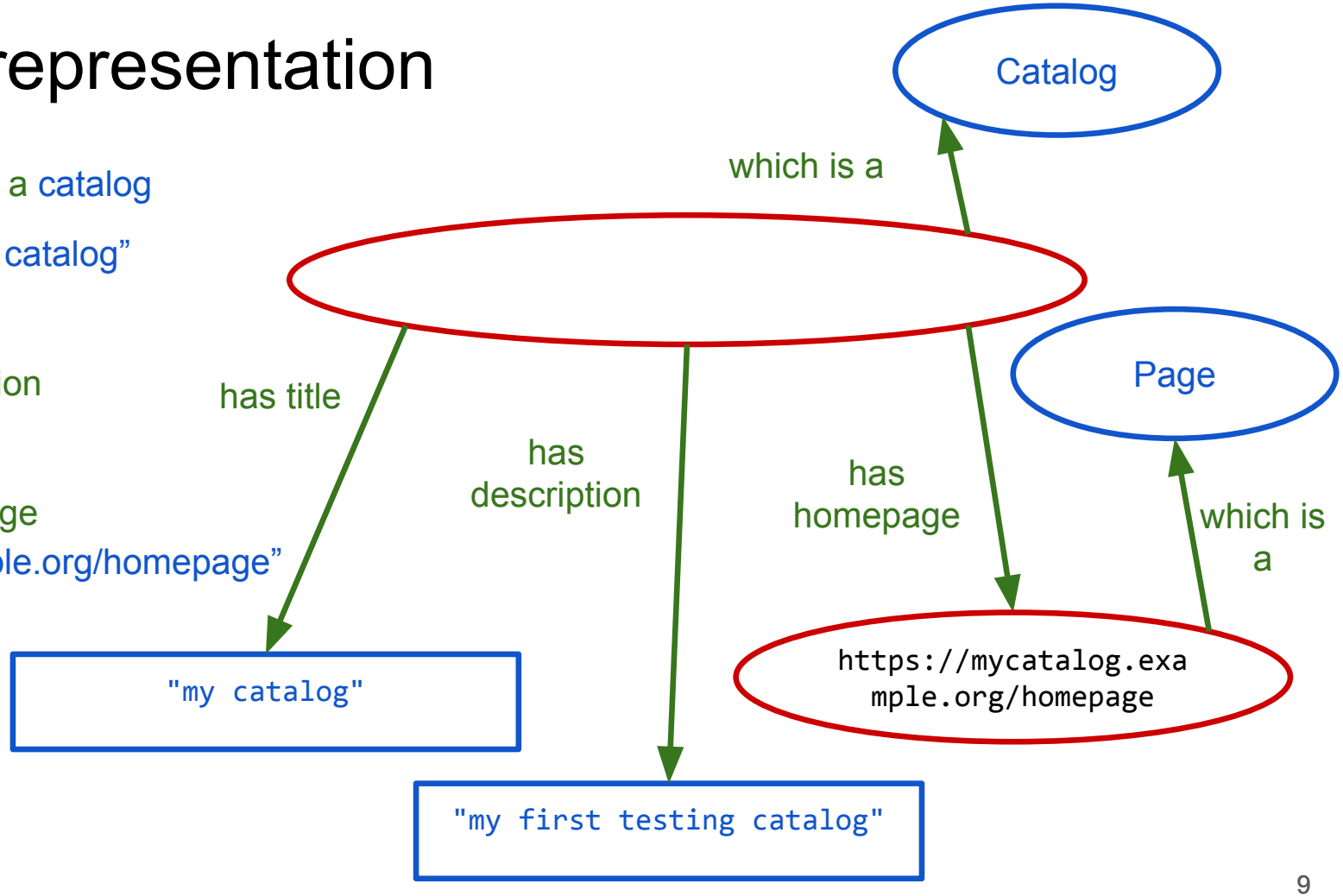
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

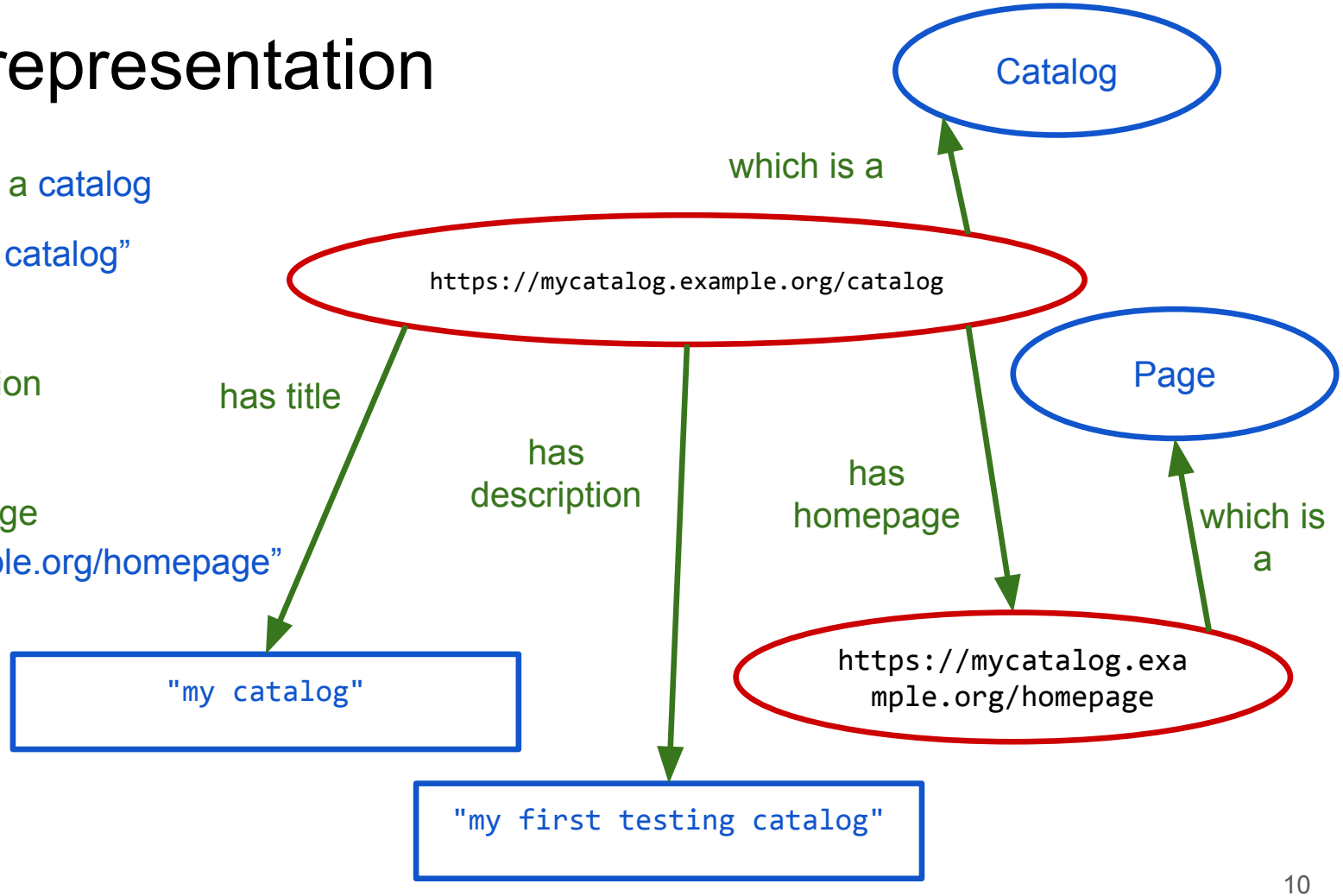
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

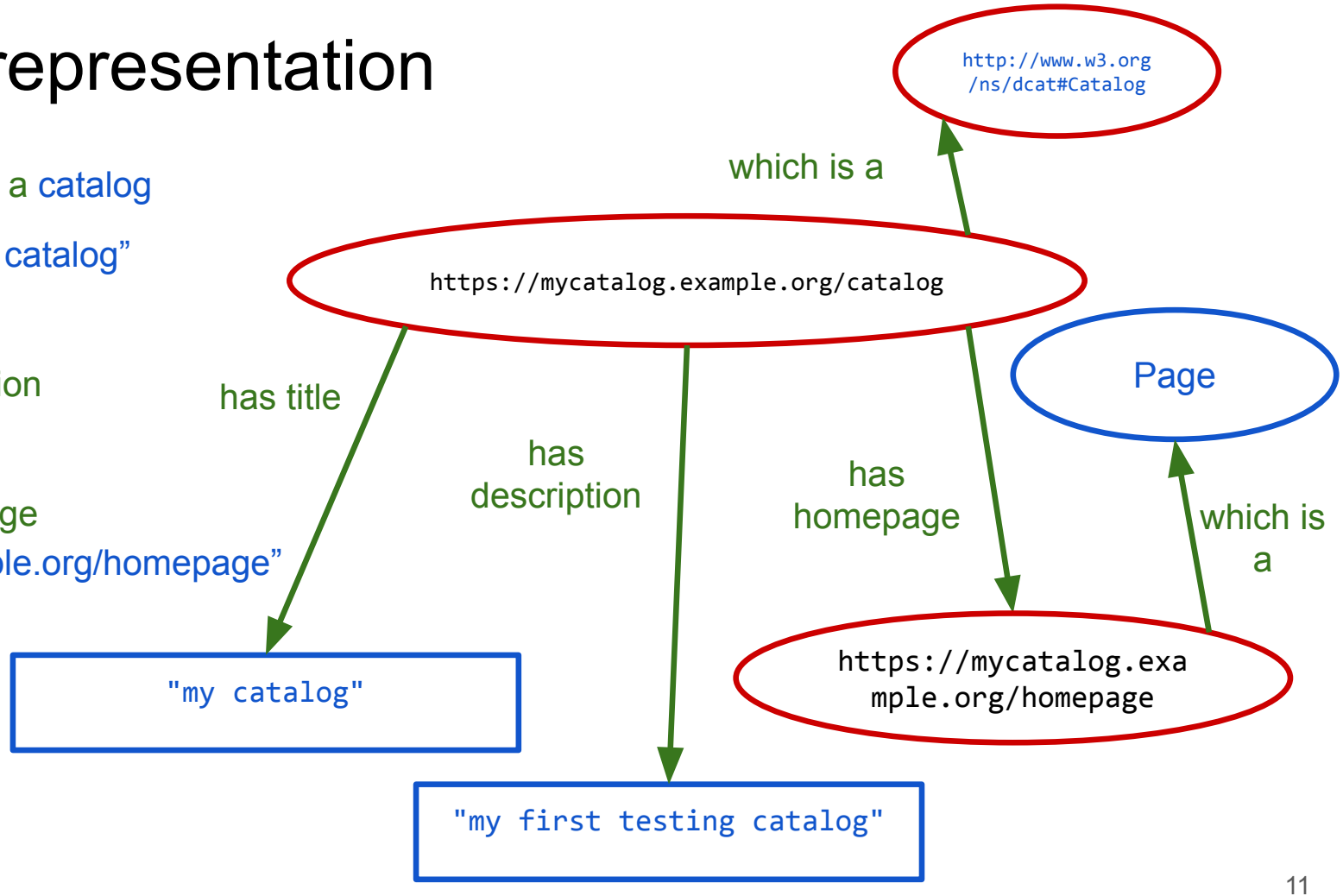
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

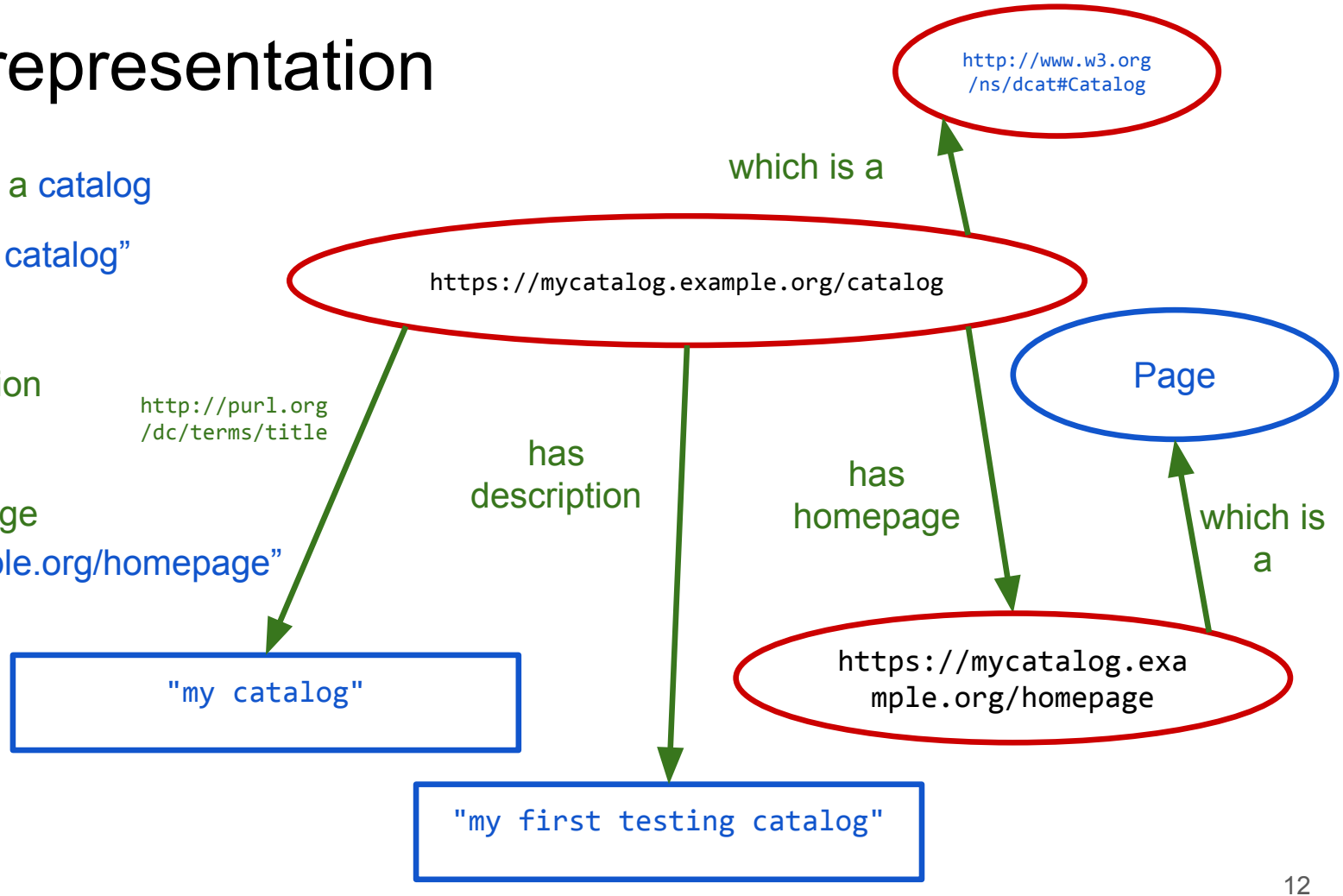
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

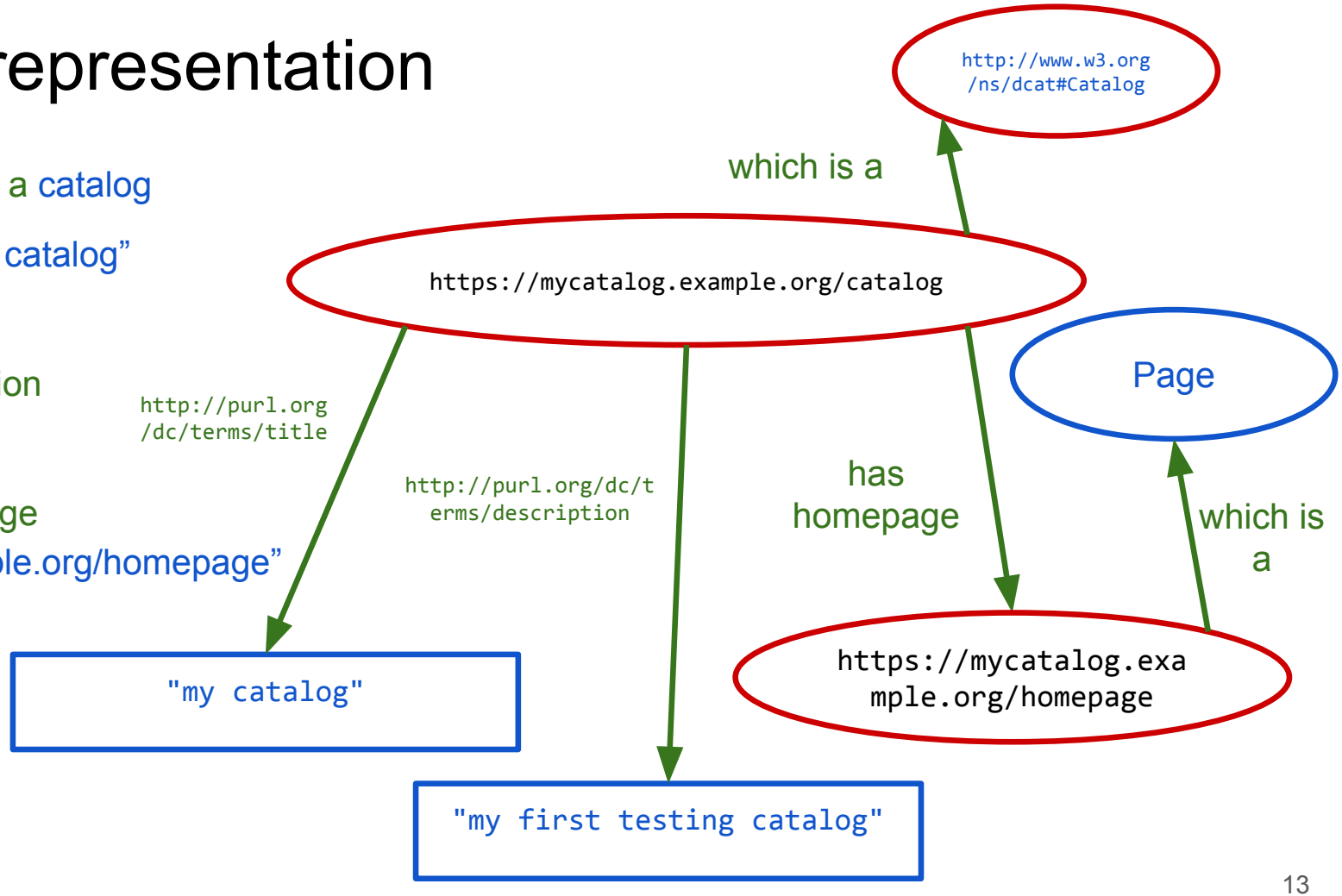
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description “my first testing catalog”

The catalog has homepage “https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

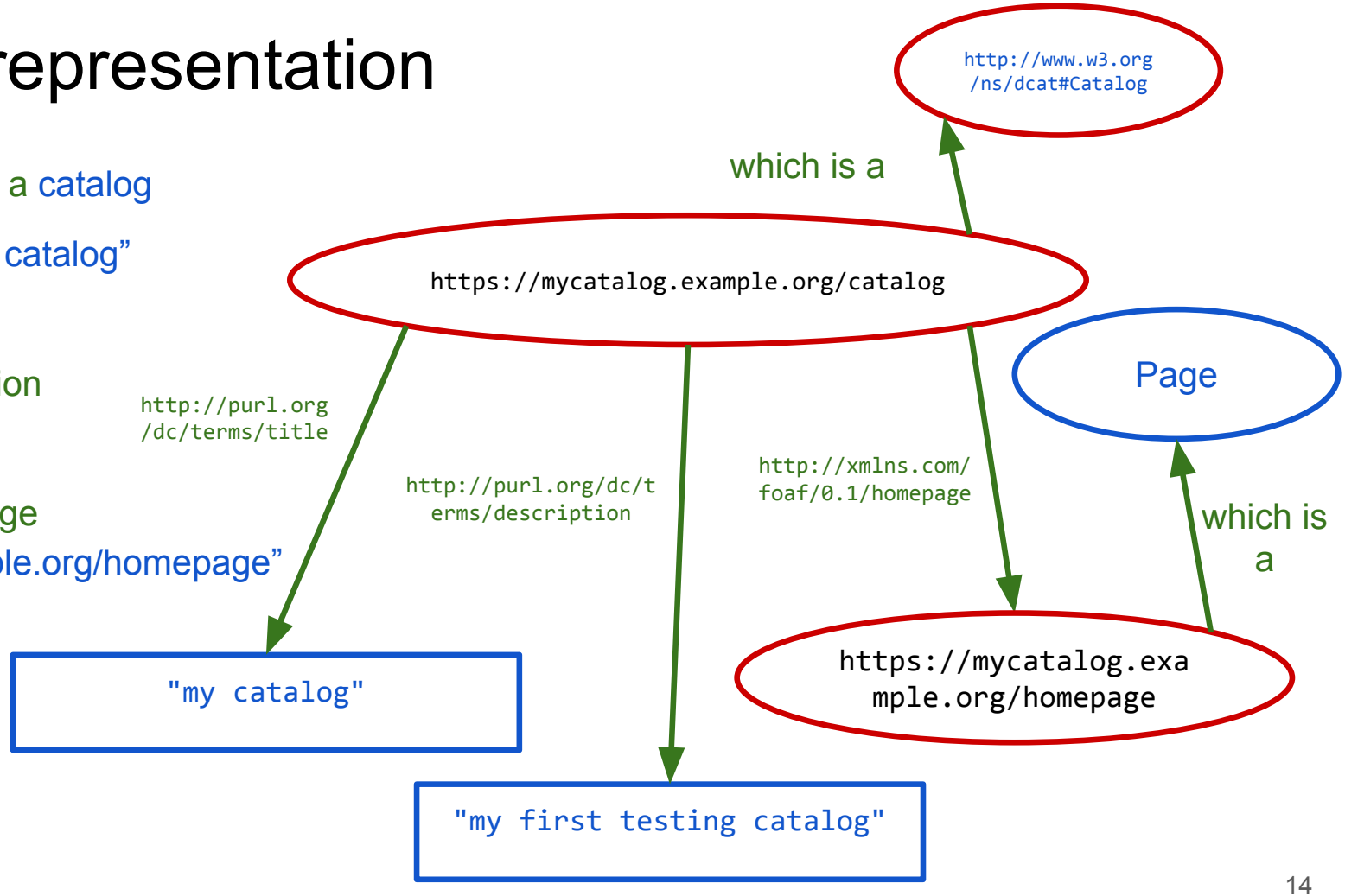
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description “my first testing catalog”

The catalog has homepage “https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

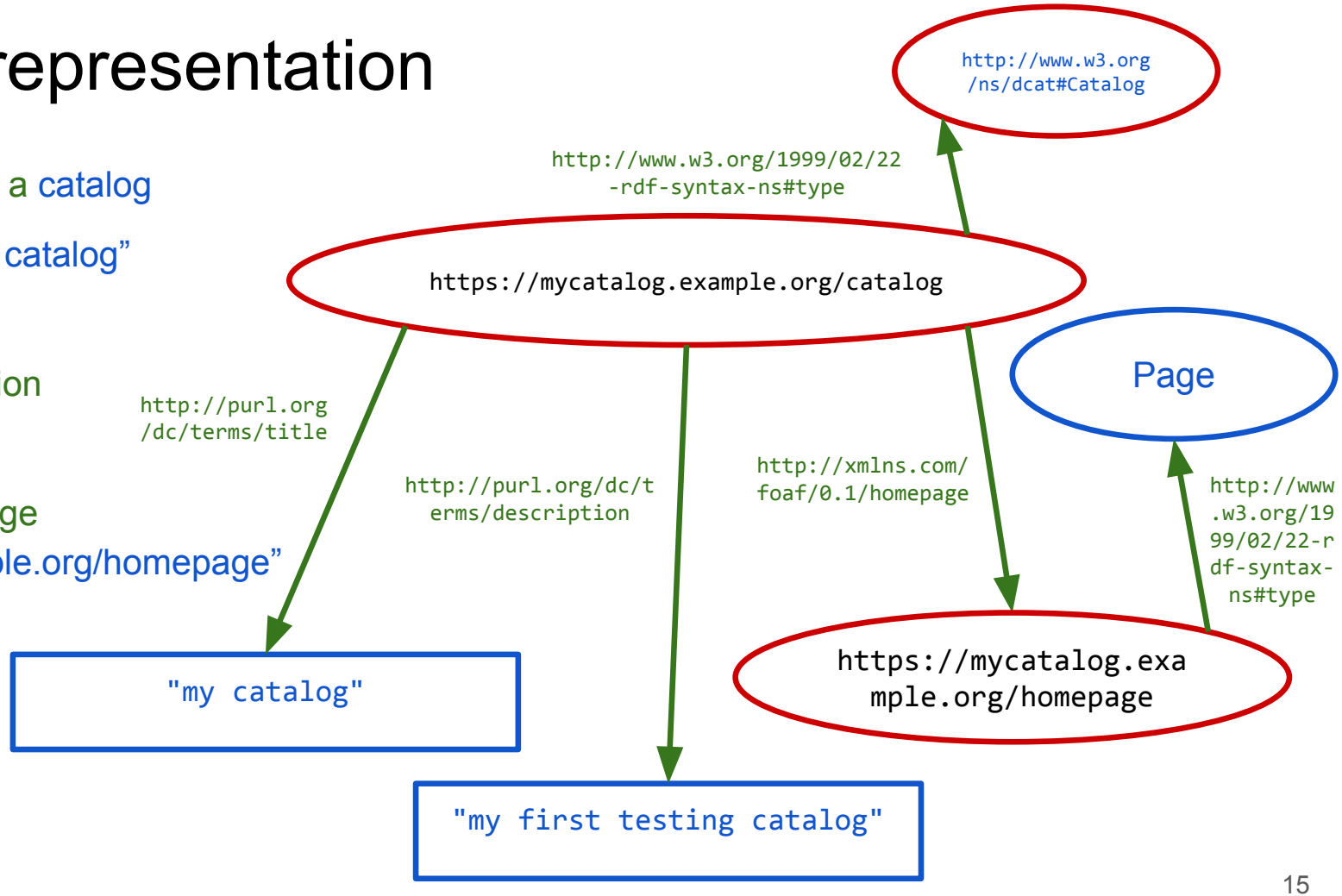
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description “my first testing catalog”

The catalog has homepage “https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

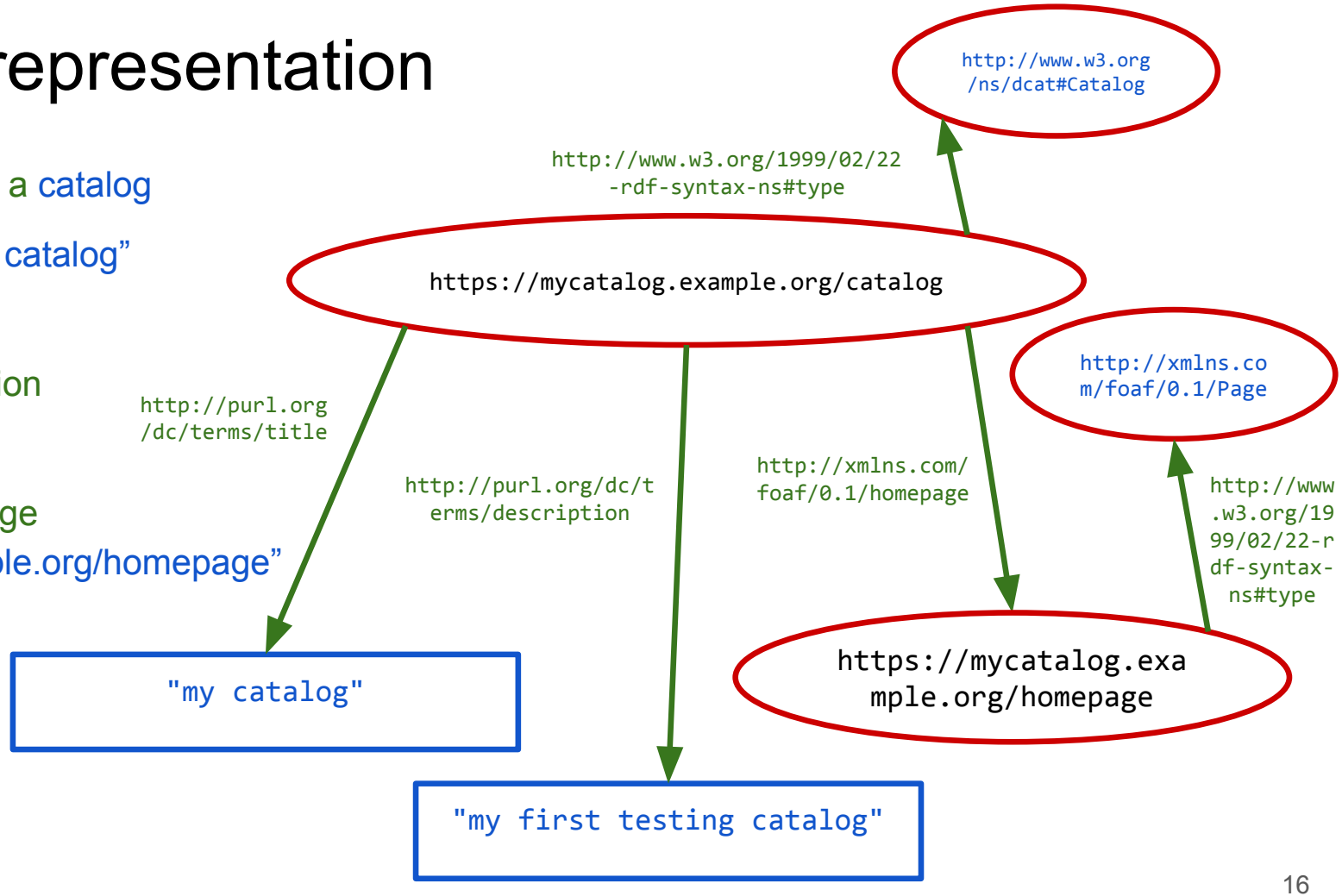
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description “my first testing catalog”

The catalog has homepage “https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

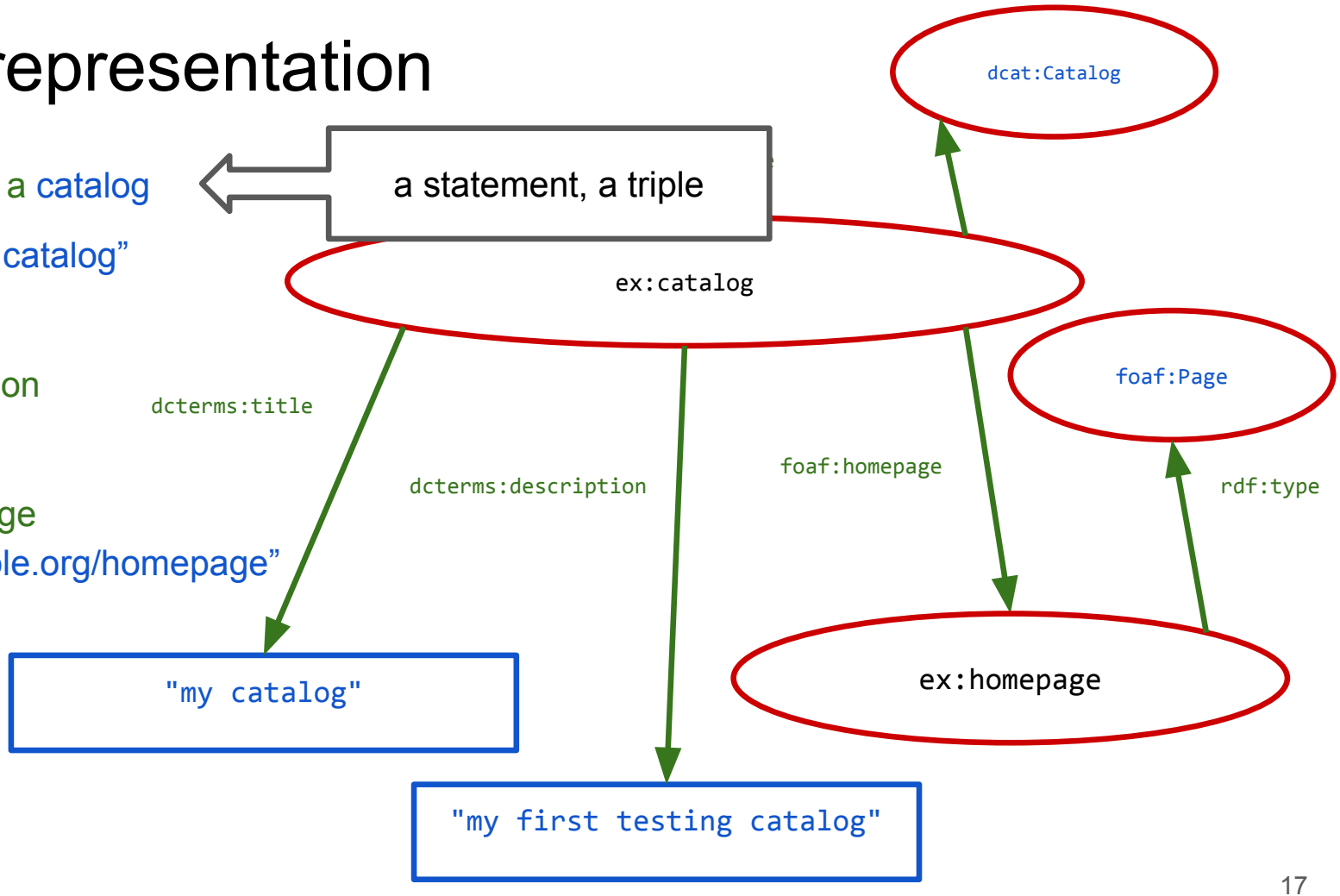
There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

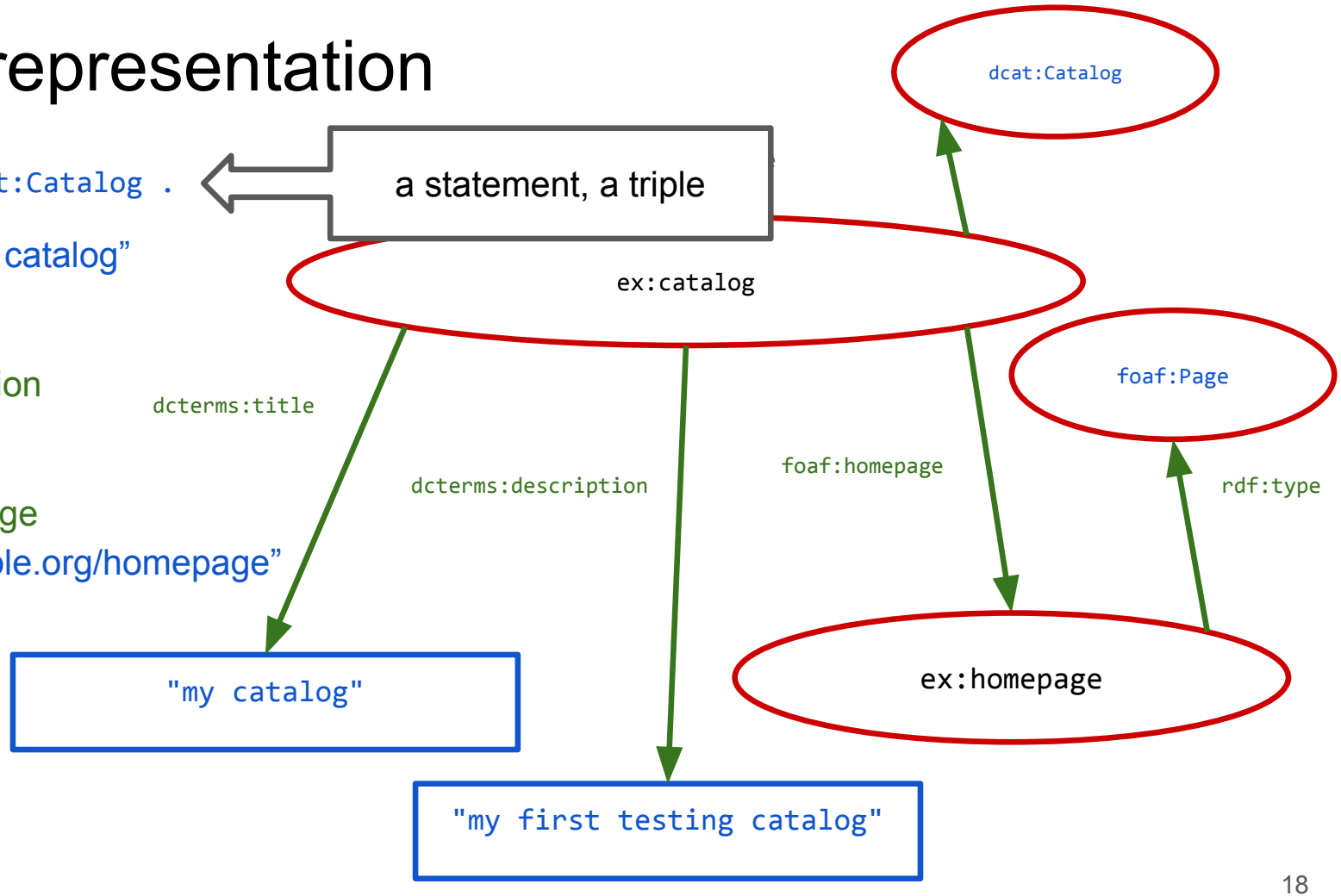
```
ex:catalog rdf:type dcat:Catalog .
```

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



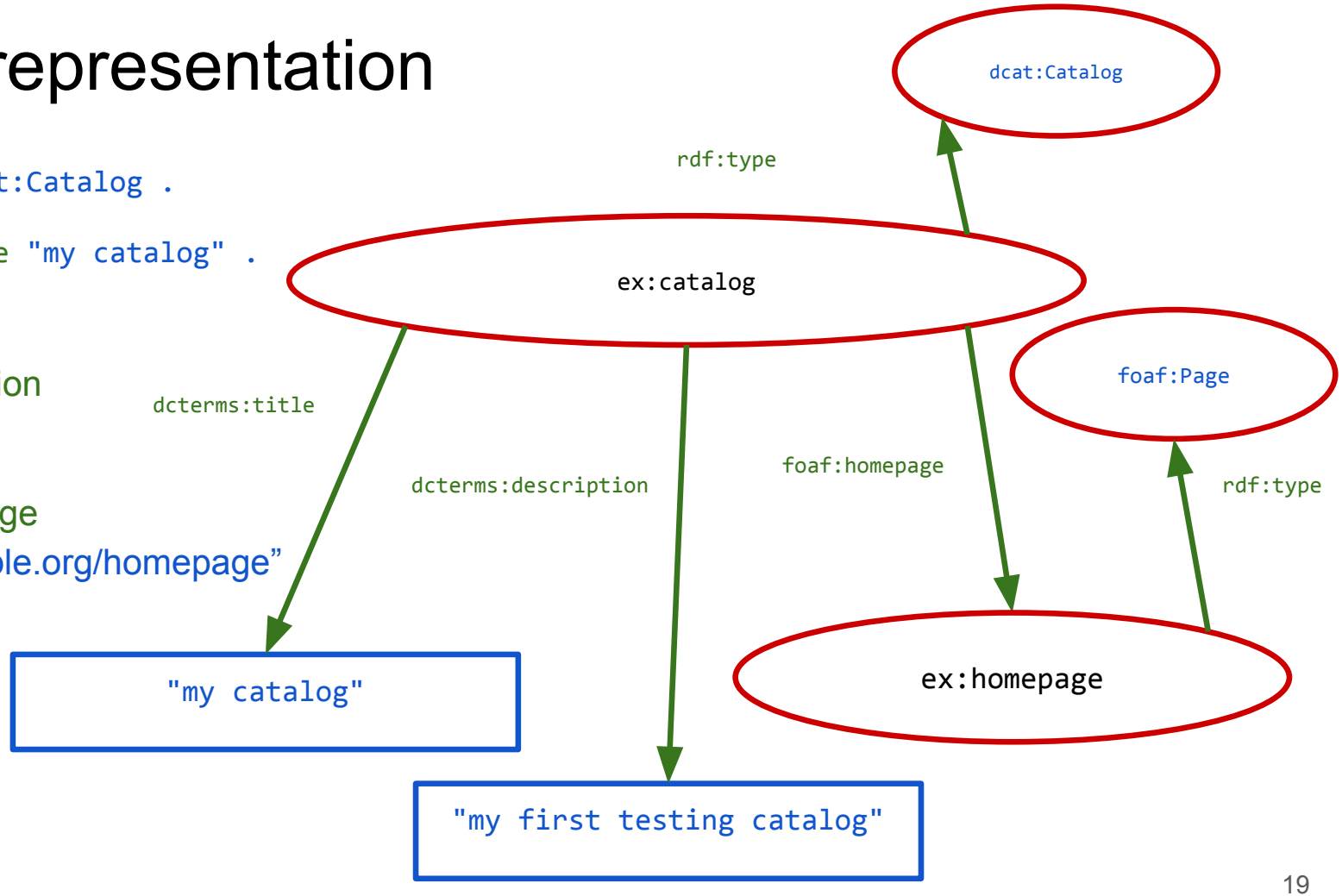
Graph data representation

```
ex:catalog rdf:type dcat:Catalog .

ex:catalog dcterms:title "my catalog" .
```

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page

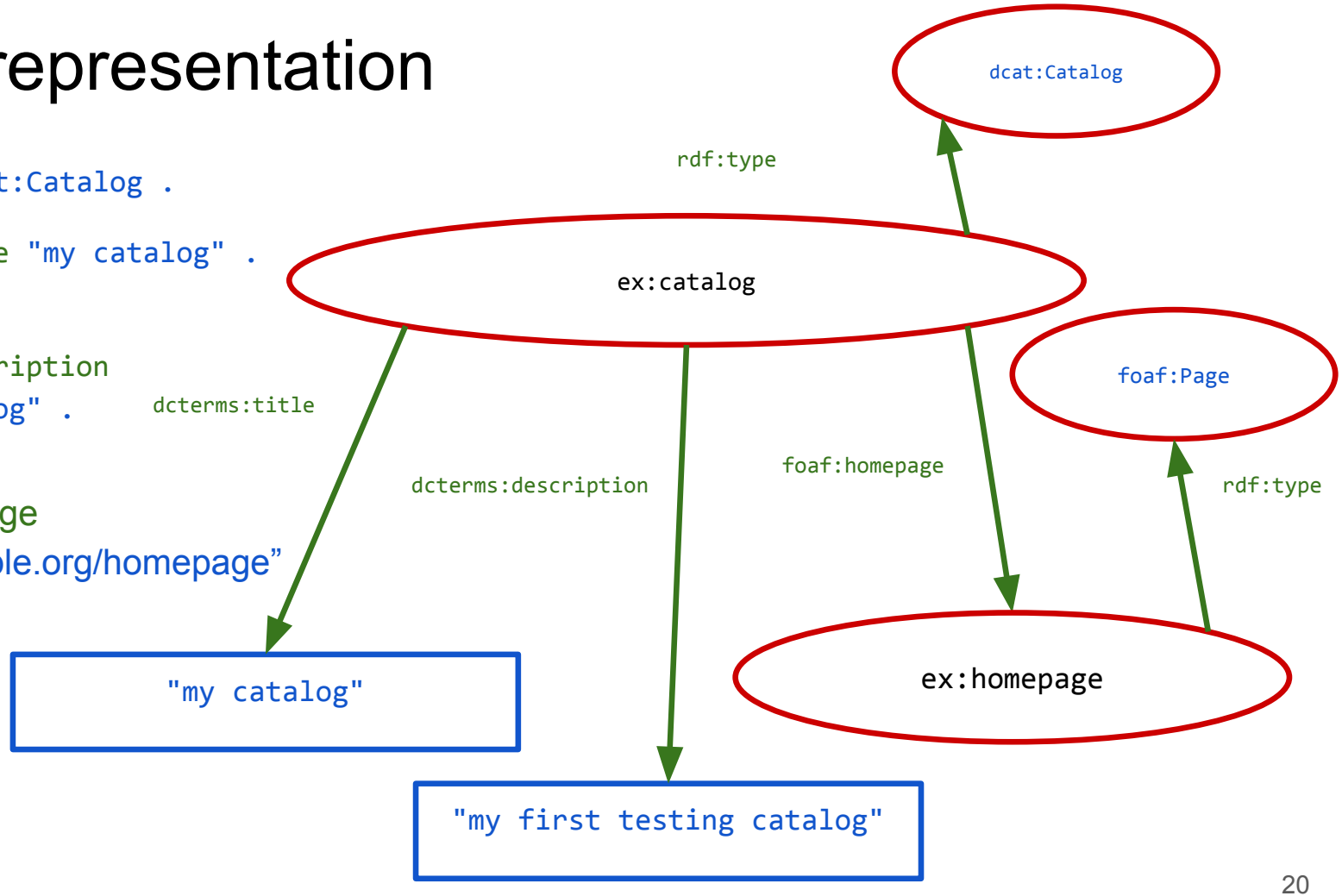


Graph data representation

```
ex:catalog rdf:type dcat:Catalog .

ex:catalog dcterms:title "my catalog" .

ex:catalog dcterms:description
"my first testing catalog" .
```

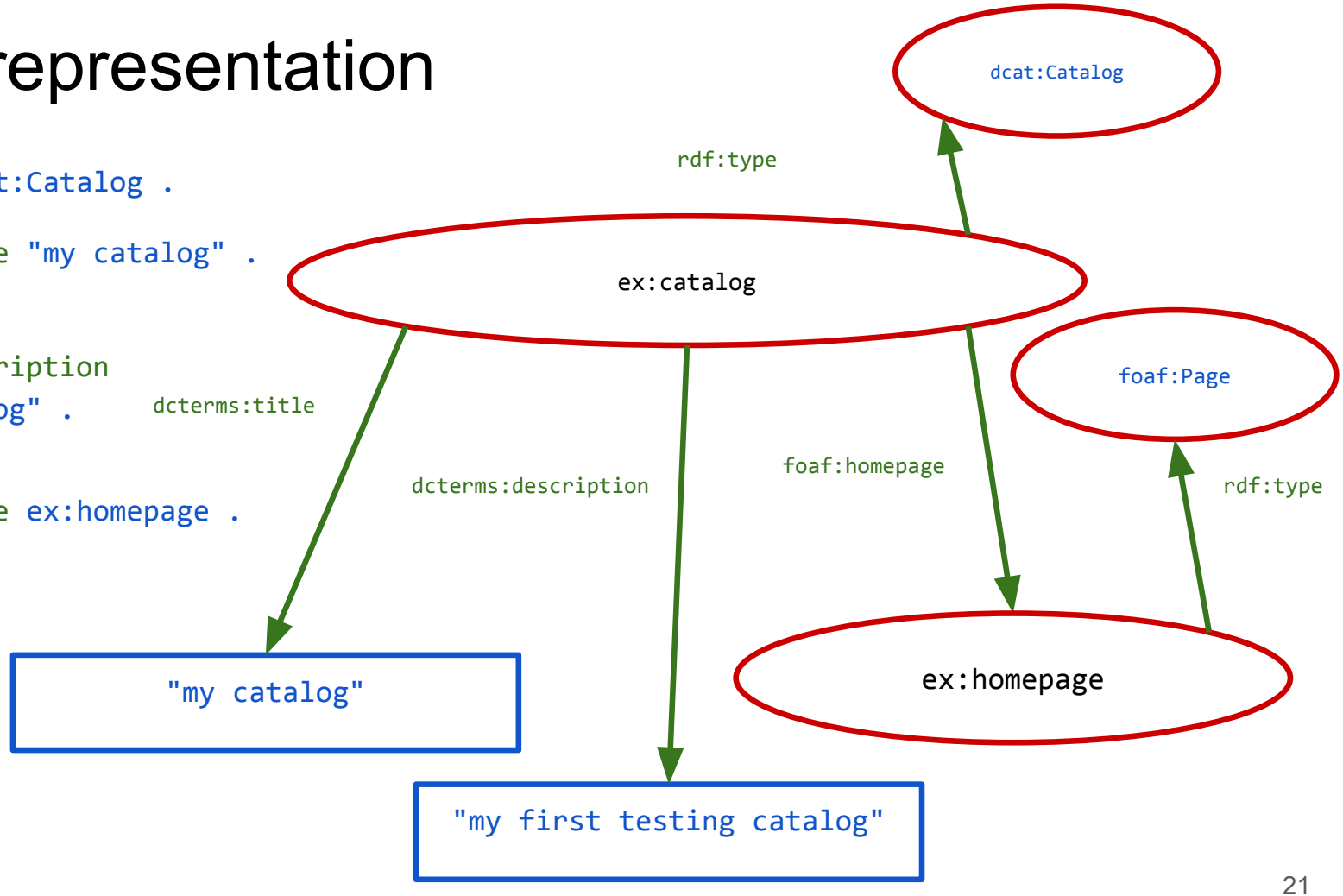
The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page



Graph data representation

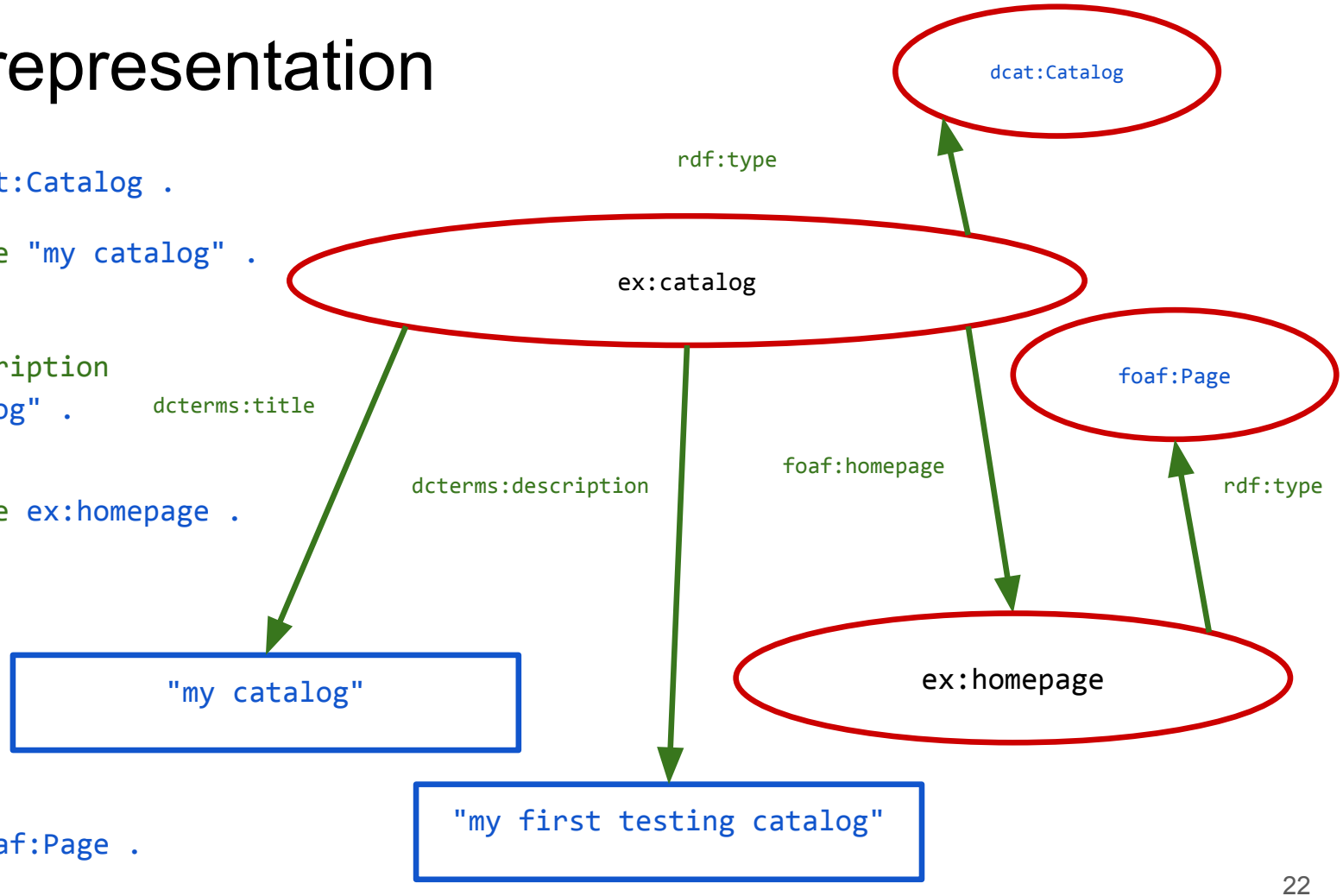
```
ex:catalog rdf:type dcat:Catalog .

ex:catalog dcterms:title "my catalog" .

ex:catalog dcterms:description
"my first testing catalog" .

ex:catalog foaf:homepage ex:homepage .
```



The homepage is a page

Graph data representation

```
ex:catalog rdf:type dcat:Catalog .

ex:catalog dcterms:title "my catalog" .

ex:catalog dcterms:description
"my first testing catalog" .

ex:catalog foaf:homepage ex:homepage .

ex:homepage rdf:type foaf:Page .
```



Graph data representation

There is a thing, which is a catalog

The catalog has title “my catalog”

The catalog has description
“my first testing catalog”

The catalog has homepage
“https://mycatalog.example.org/homepage”

The homepage is a page

```
ex:catalog rdf:type dcat:Catalog .
ex:catalog dcterms:title "my catalog" .
ex:catalog dcterms:description "my first testing catalog" .
ex:catalog foaf:homepage ex:homepage .
ex:homepage rdf:type foaf:Page .
```

RDF

RDF - Resource Description Framework - idea

RDF - graph based data model -
a **set** of **triples**

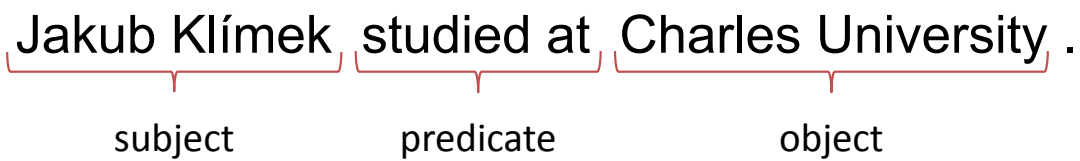
Triple describes a relation as:

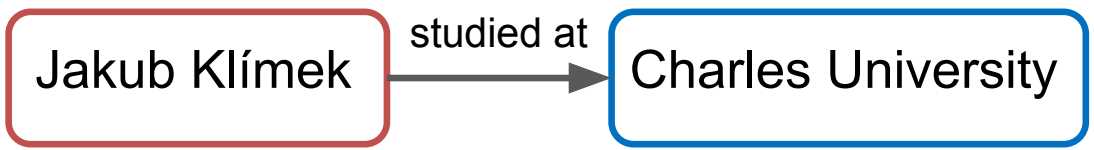
subject predicate object

2004 & 2014 W3C Recommendations

Triples are written in one of RDF notations / syntaxes / serializations:

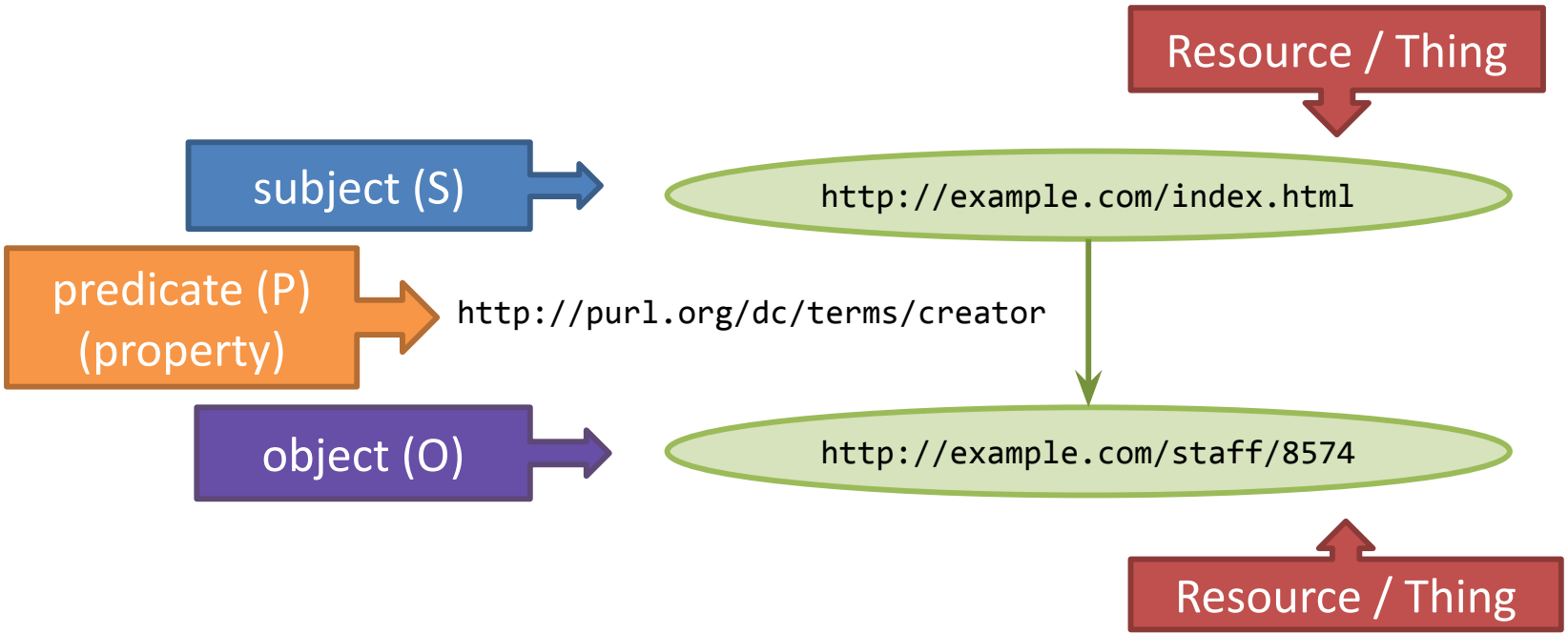
RDF/XML, RDFa, N-Triples, Turtle, JSON-LD, N-Quads, TriG





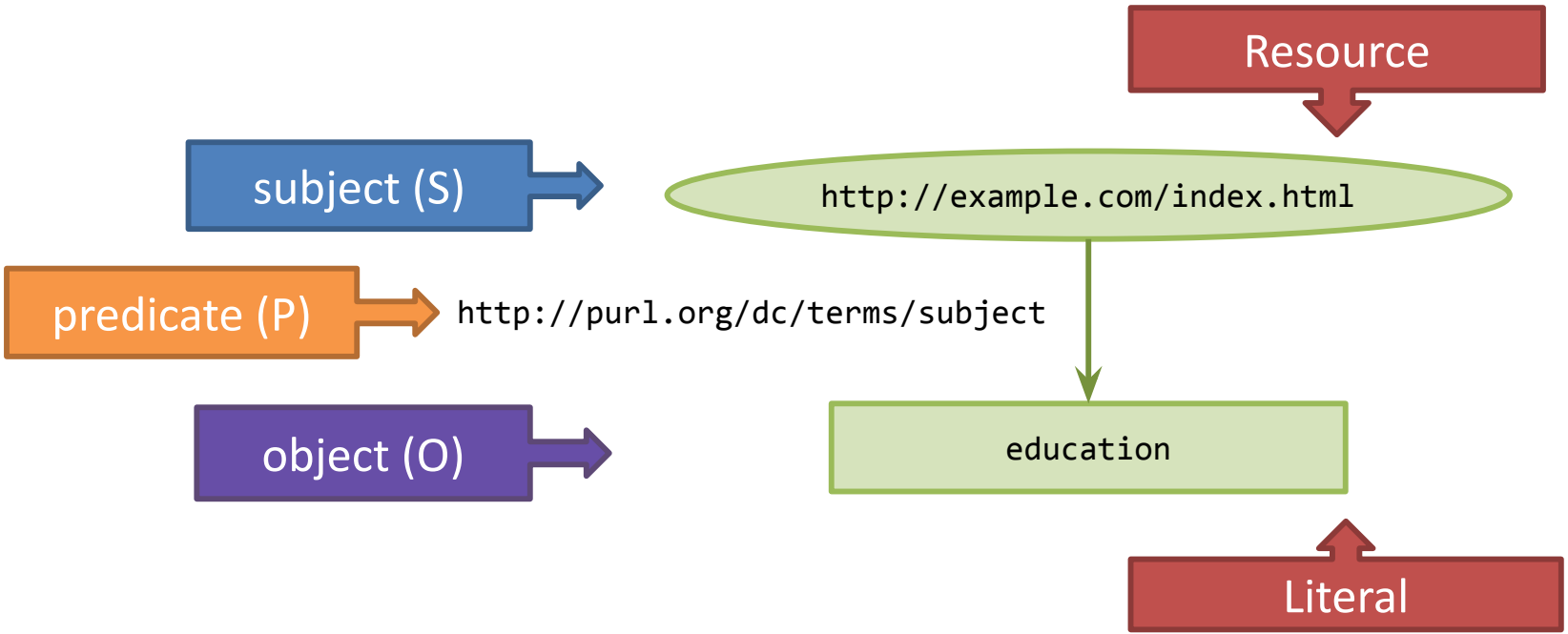
RDF model: a triple, a statement

```
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8574> .
```



RDF model: a triple with literal value

```
<http://example.com/index.html> <http://purl.org/dc/terms/subject> "education" .
```



RDF serializations: IRIs and IRI prefixes

```
<http://purl.org/dc/terms/creator>
```

=

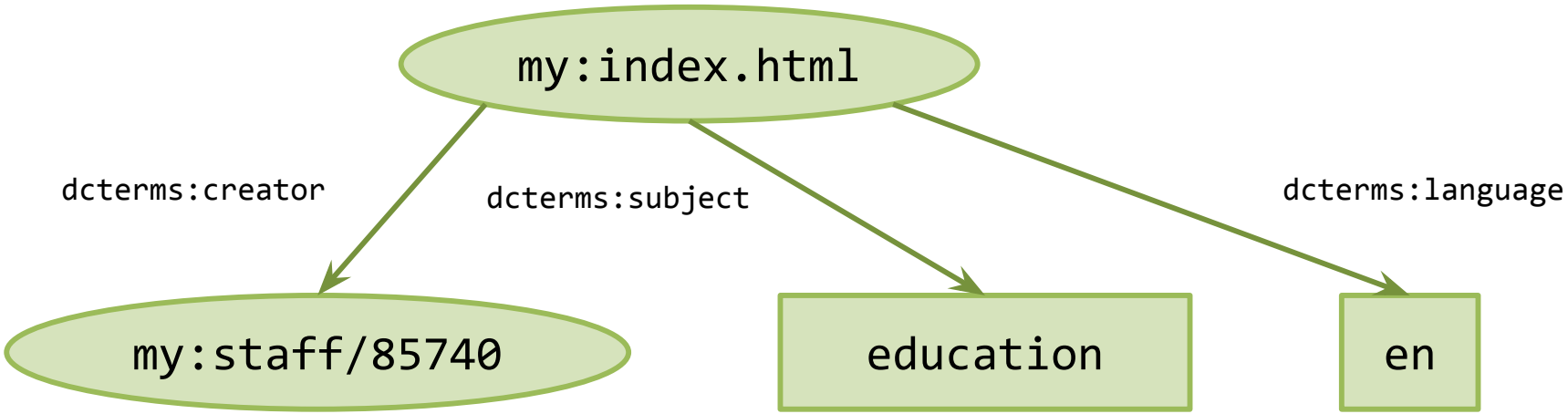
```
@prefix dcterms: <http://purl.org/dc/terms/> .
dcterms:creator
```

See http://prefix.cc/dcterms

RDF model: multiple properties

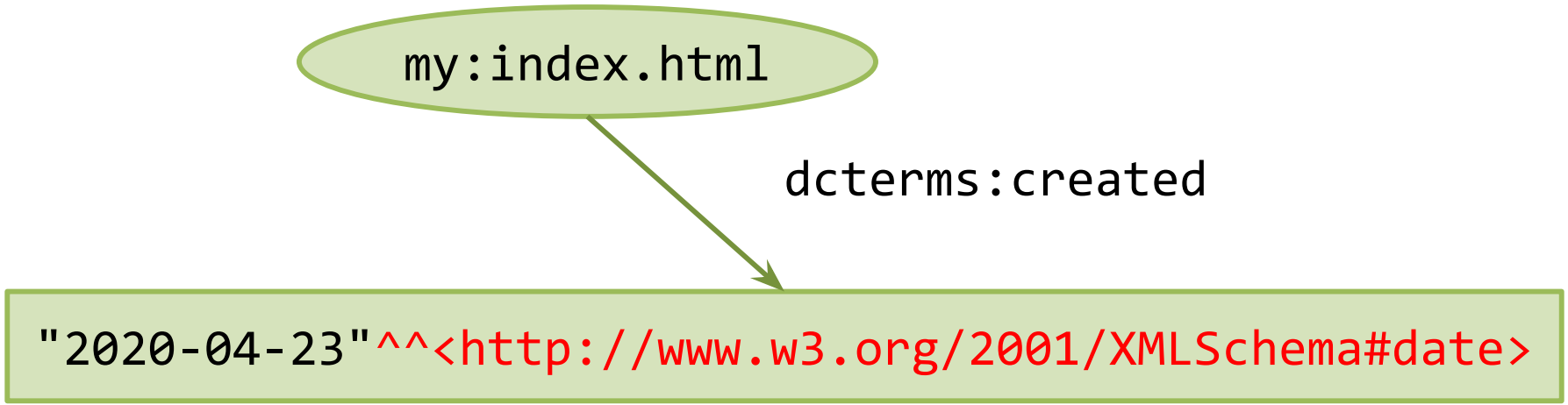
```
my:index.html dcterms:creator exstaff:85740 .
my:index.html dcterms:subject "education" .
my:index.html dcterms:language "en" .
```

a set
i.e. no ordering
among triples



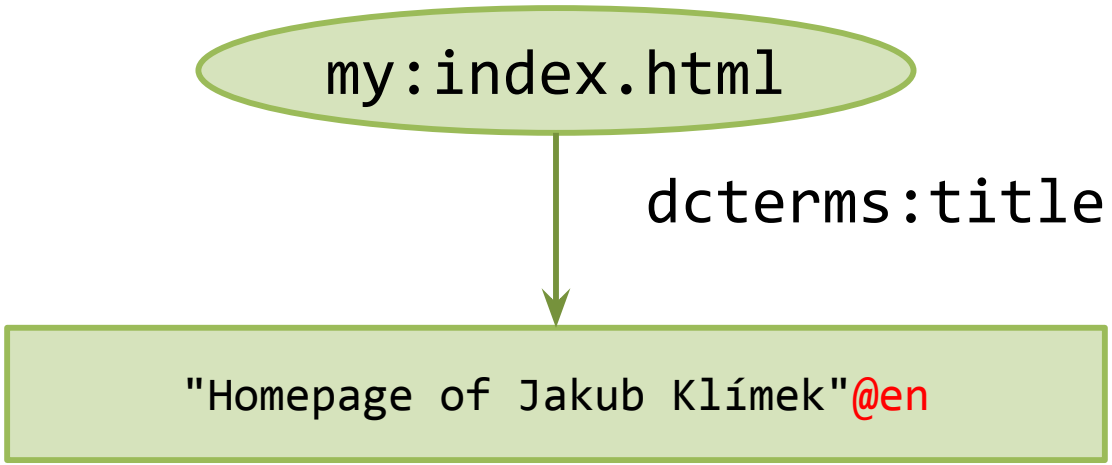
RDF model: typed literals

```
my:index.html dcterms:created "2020-04-23"^^xsd:date .
```

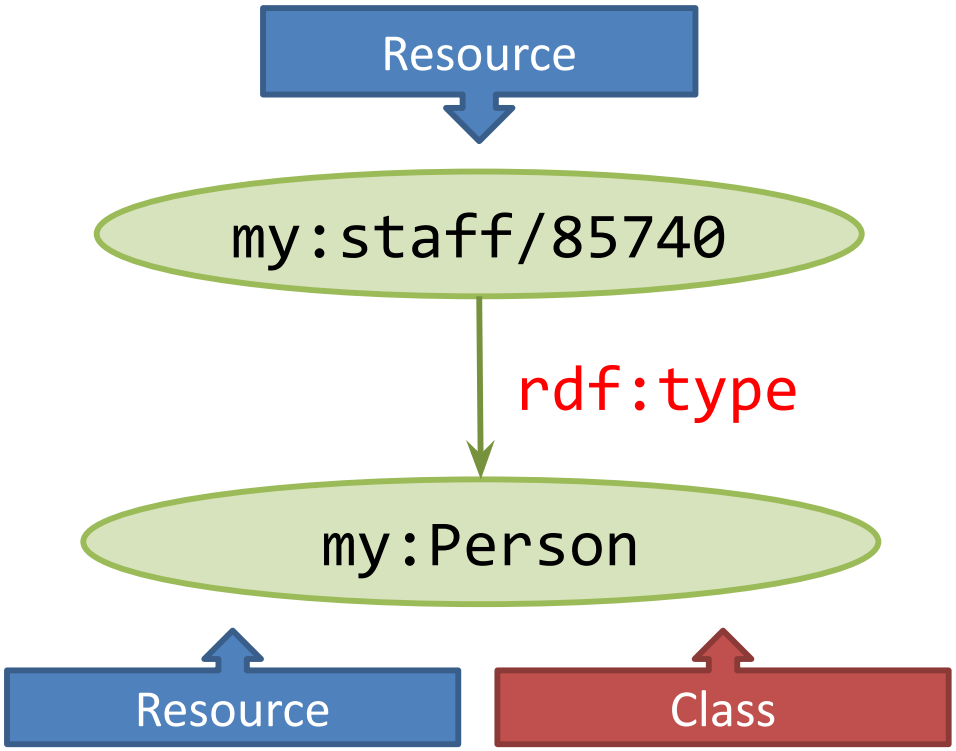


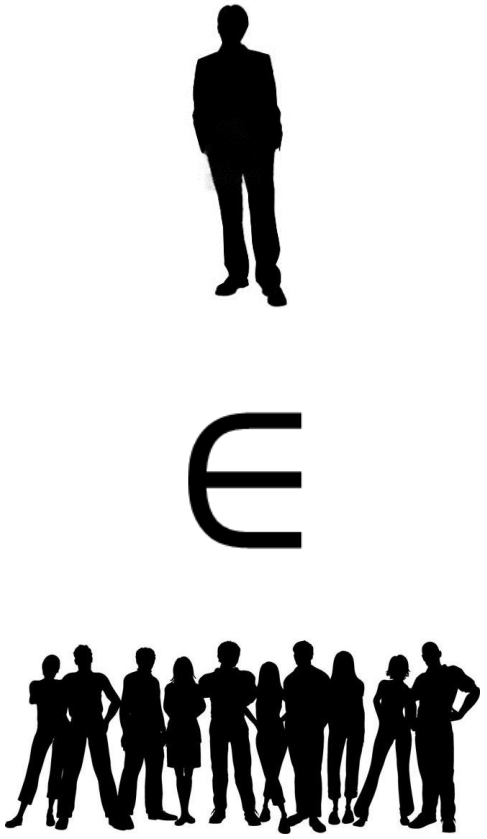
RDF model: text literals with a language tag

```
my:index.html dcterms:title "Homepage of Jakub Klímek"@en .
```



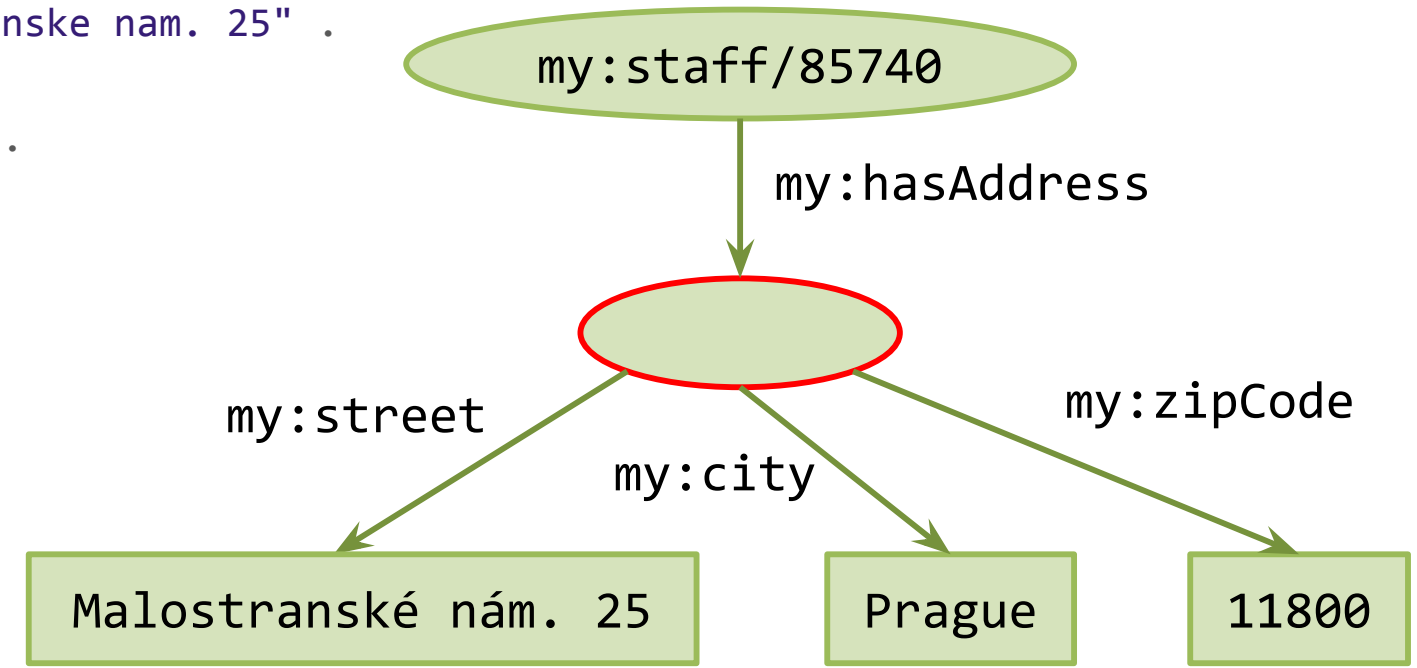
RDF model: classes





RDF model: blank nodes

```
my:staff/85740 my:hasAddress _:a1 .

_:a1 my:street "Malostranske nam. 25" .
_:a1 my:city "Prague" .
_:a1 my:zipCode "11800" .
```



RDF - Resource Description Framework

- 1.0: W3C Recommendation
  - 10 February 2004
- 1.1: W3C Recommendation
  - 25 February 2014
- 1.2: Working draft
  - 22 September 2023
- Graph data model
  - Directed labeled multigraph
    - Vertices for subjects and objects
    - Labeled edges for particular triples

RDF serializations

RDF 1.1 N-Triples

- W3C Recommendation
  - 25 February 2014

```
<http://one.example/subject1> <http://one.example/predicate1>
<http://one.example/object1> . # comments here

<http://example.org/show/218> <http://example.org/show/localName> "That
Seventies Show"@en . # literal with a language tag

<http://en.wikipedia.org/wiki/Helium>
<http://example.org/elements/atomicNumber>
"2"^^<http://www.w3.org/2001/XMLSchema#integer> . # xsd:integer
```

RDF 1.1 N-Triples

```
<http://example.com/index.html> <http://purl.org/dc/terms/created> "2020-04-23"^^<http://...#date>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8574>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8575>  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "Moje stránka"@cs  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "My page"@en  .
```

RDF 1.1 Turtle - Prefixes

```
<http://example.com/index.html> <http://purl.org/dc/terms/created> "2020-04-23"^^<http://...#date>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8574>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8575>  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "Moje stránka"@cs  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "My page"@en  .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix my: <http://example.com/> .
@prefix staff: <http://example.com/staff/> .

my:index.html dcterms:created "2020-04-23"^^xsd:date  .
my:index.html dcterms:creator staff:8574  .
my:index.html dcterms:creator staff:8575  .
my:index.html dcterms:title "Moje stránka"@cs  .
my:index.html dcterms:title "My page"@en  .
```

RDF 1.1 Turtle - Prefixes and ";"

```
<http://example.com/index.html> <http://purl.org/dc/terms/created> "2020-04-23"^^<http://...#date>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8574>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8575>  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "Moje stránka"@cs  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "My page"@en  .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix my: <http://example.com/> .
@prefix staff: <http://example.com/staff/> .

my:index.html dcterms:created "2020-04-23"^^xsd:date  ;
              dcterms:creator staff:8574  ;
              dcterms:creator staff:8575  ;
              dcterms:title "Moje stránka"@cs  ;
              dcterms:title "My page"@en  .
```

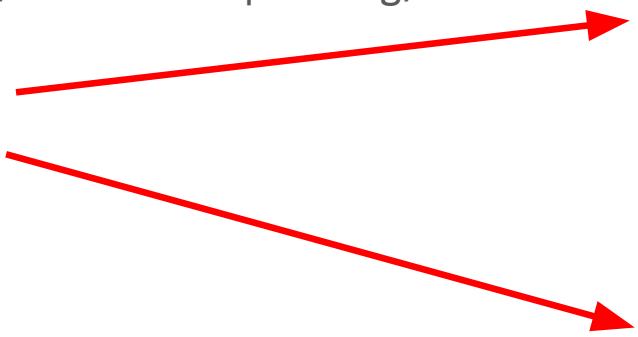
RDF 1.1 Turtle - Prefixes and ";" and ","

```
<http://example.com/index.html> <http://purl.org/dc/terms/created> "2020-04-23"^^<http://...#date>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8574>  .
<http://example.com/index.html> <http://purl.org/dc/terms/creator> <http://example.com/staff/8575>  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "Moje stránka"@cs  .
<http://example.com/index.html> <http://purl.org/dc/terms/title> "My page"@en  .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix my: <http://example.com/> .
@prefix staff: <http://example.com/staff/> .

my:index.html dcterms:created "2020-04-23"^^xsd:date  ;
              dcterms:creator staff:8574  ,
                              staff:8575  ;
              dcterms:title "Moje stránka"@cs  ,
                            "My page"@en  .
```

RDF 1.1 Turtle - More prefixes

```
@prefix foo: <http://example.org/ns#> .
@prefix : <http://other.example.org/ns#> .

foo:bar foo: : .
:bar : foo:bar .
```



```
<http://example.org/ns#bar>
<http://example.org/ns#>
<http://other.example.org/ns#> .

<http://other.example.org/ns#bar>
<http://other.example.org/ns#>
<http://example.org/ns#bar> .
```

RDF 1.1 Turtle - Relative IRIs

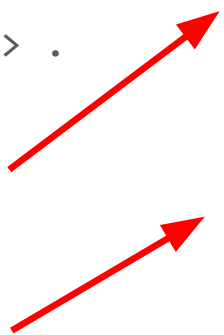
- </path>
- <#fragment>
- <>

Need to know, relative to WHAT the IRI is.

- Implicitly, a document URL (if known)
- Explicitly using @base

Assuming Document URL https://test.org/doc

```
@prefix foo: <http://example.org/ns#> .

<#document> foo: <https://jk.com> .

@base <http://newbase.com/> .
<#document> foo: <https://jk.com> .
```



```
<https://test.org/doc#document>
<http://example.org/ns#>
<https://jk.com> .

<http://newbase.com/#document>
<http://example.org/ns#>
<https://jk.com> .
```

RDF 1.1 Turtle - Multiline strings, escapes

```
"""a string
with newlines
"""
```

- \t (U+0009, tab)
- \n (U+000A, linefeed)
- \r (U+000D, carriage return)
- \" (U+0022, double quote - only allowed inside strings)
- \> (U+003E, greater than - only allowed inside URIs)
- \\ (U+005C, backslash)
- \uHHHH or \UHHHHHHHH for writing Unicode characters by hexadecimal codepoint where H is a single hexadecimal digit.

RDF 1.1 Turtle - Class assignment (`rdf:type`)

```
<http://example.com/index.html>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
foaf:Document .
```

=

```
<http://example.com/index.html> a foaf:Document .
```

RDF 1.1 Turtle - blank nodes

```
<http://example.com/> a v:VCard ;
   v:adr [
     a v:Work ;
      v:country-name "Australia" ;
      v:locality "WonderCity" ;
      v:postal-code "5555" ;
      v:street-address "33 Enterprise Drive"
   ] ;
  .
```

=

```
<http://example.com/> v:adr _:1234 .
_:1234 a v:Work ;
       v:locality "WonderCity" ;
       ...
```

RDF 1.1 Turtle - datatype shortcuts

```
ex:Car ex:numberOfWheels 4 ;

ex:Car ex:numberOfWheels +4 ;

ex:Car ex:numberOfWheels "4"^^xsd:integer ;
```

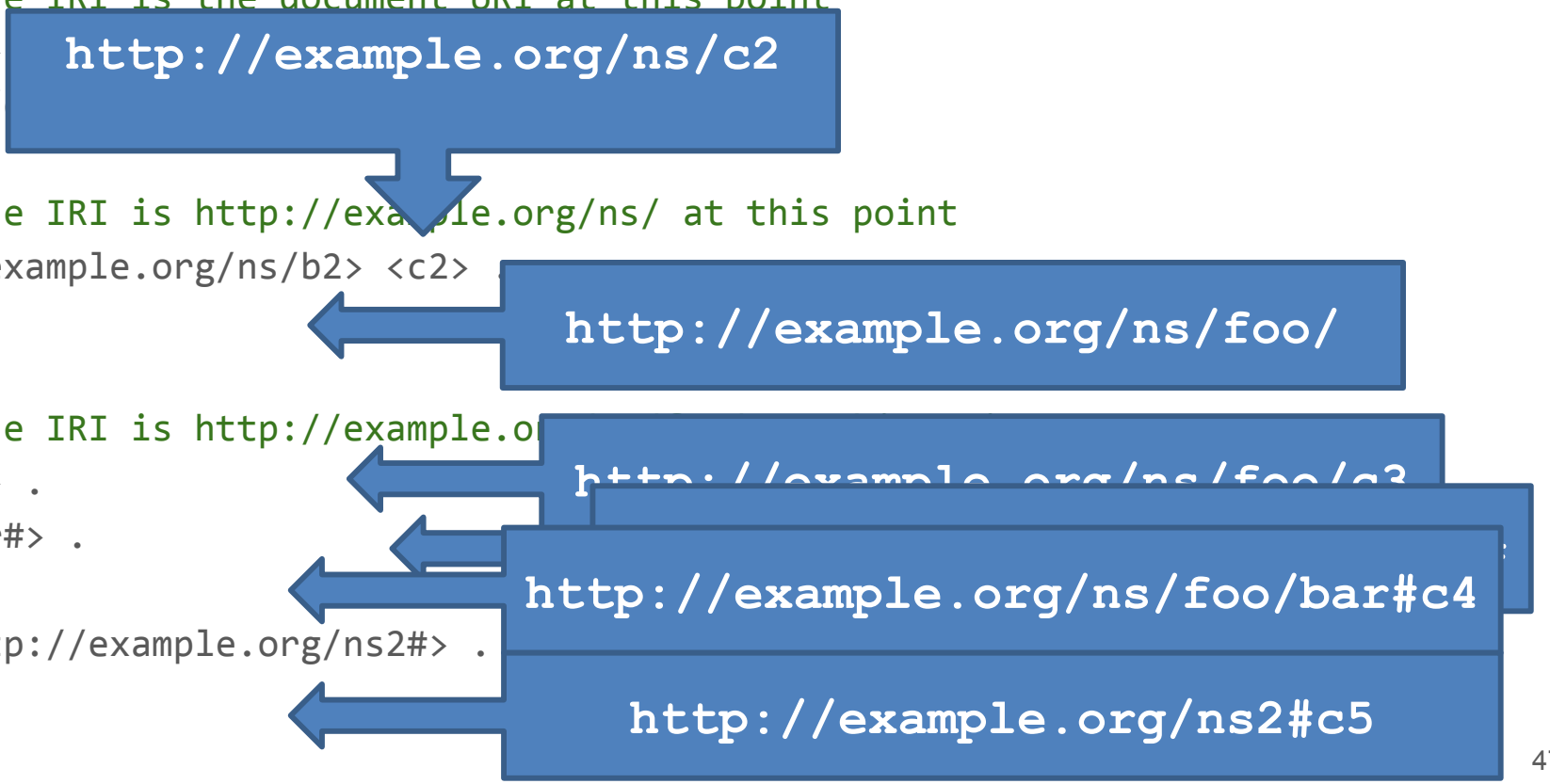
```
ex:Car ex:value 1.3e6 ;

ex:Car ex:value "1.3e6"^^xsd:double ;
```

```
ex:Car ex:value 1300000.0 ;

ex:Car ex:value "1300000.0"^^xsd:decimal ;
```

```
ex:Car ex:leftHandDrive true ;

ex:Car ex:leftHandDrive "true"^^xsd:boolean ;
```

RDF 1.1 Turtle - playing with prefixes and rel. IRIs

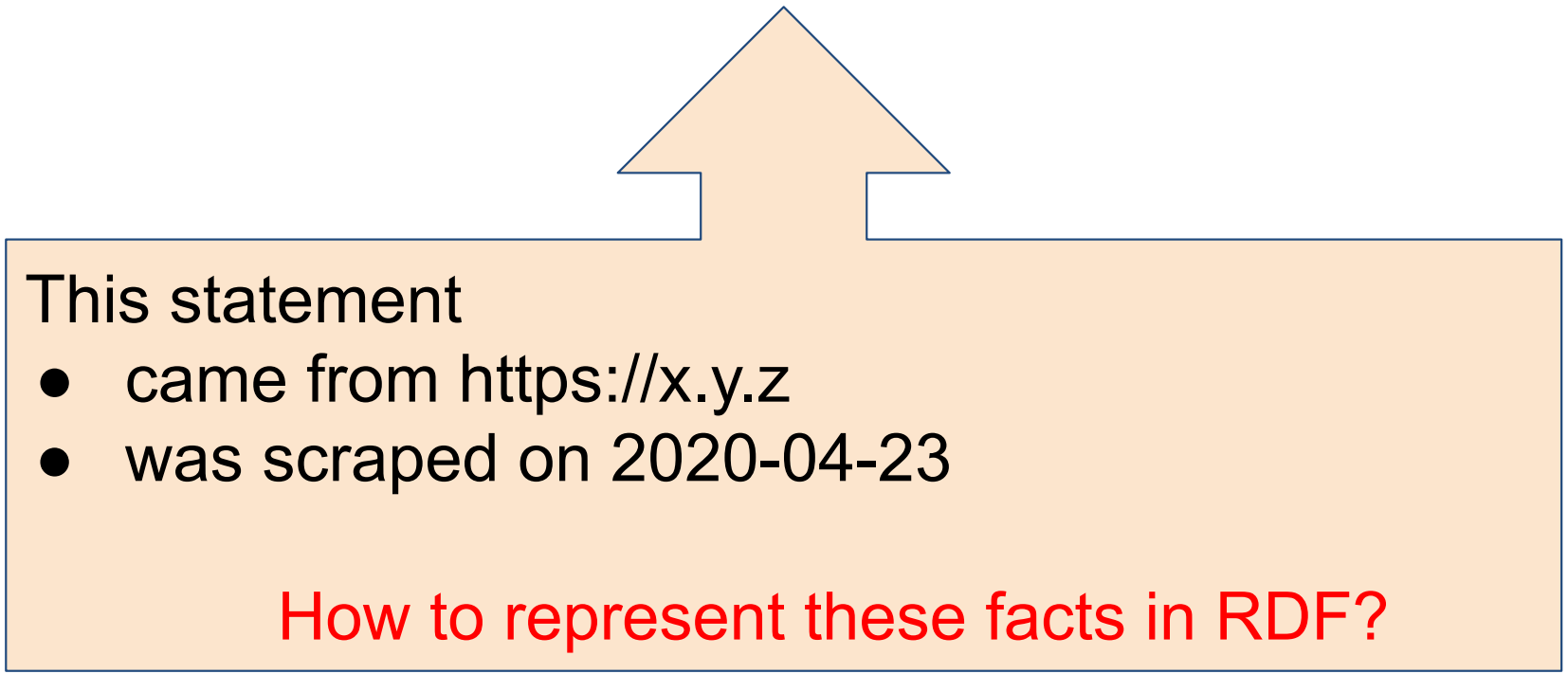
```
# In-scope base IRI is the document URI at this point
<a1> <b1> <c1>
@base <http://

# In-scope base IRI is http://example.org/ns/ at this point
<a2> <http://example.org/ns/b2> <c2>
@base <foo/> .

# In-scope base IRI is http://example.o
<a3> <b3> <c3> .
@prefix : <bar#> .
:a4 :b4 :c4 .
@prefix : <http://example.org/ns2#> .
:a5 :b5 :c5 .
```



Detour from RDF serializations

RDF model - statements about statements

```
my:index.html my:createdBy "Jakub Klímek" .
```



RDF model - reification

Direct approach to the problem:

Statement will become a resource.

- Assign IRI to the statement itself, or
- make it a blank node (e.g. `_:triple1`)

Original statement:

```
my:index.html my:createdBy "Jakub Klímek" .
```

Reified statement:

```
_:triple1 a rdf:Statement .

_:triple1 rdf:subject   my:index.html .

_:triple1 rdf:predicate my:createdBy  .

_:triple1 rdf:object    "Jakub Klímek".
```

Possibility of additional statements we need:

```
_:triple1 dcterms:created
"2020-04-23"^^xsd:date .
```

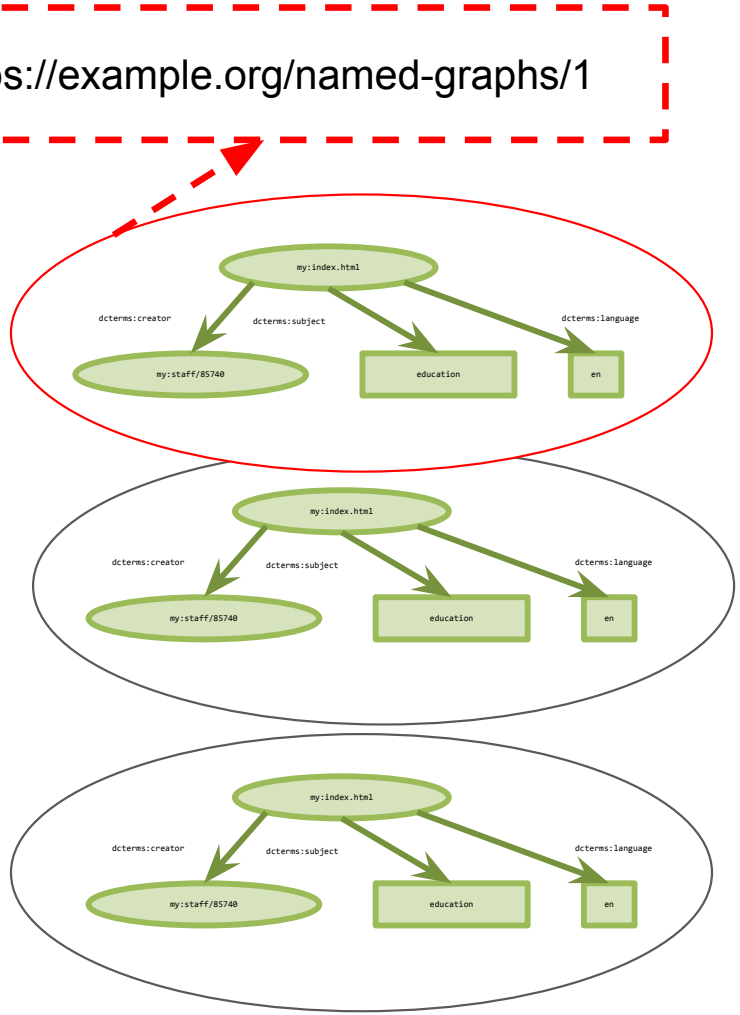
RDF model - named graphs

Alternative approach to the problem:

- Statements belong to “named graphs”
- Named graphs are resources
- We can state facts about resources

RDF Triples become RDF Quads

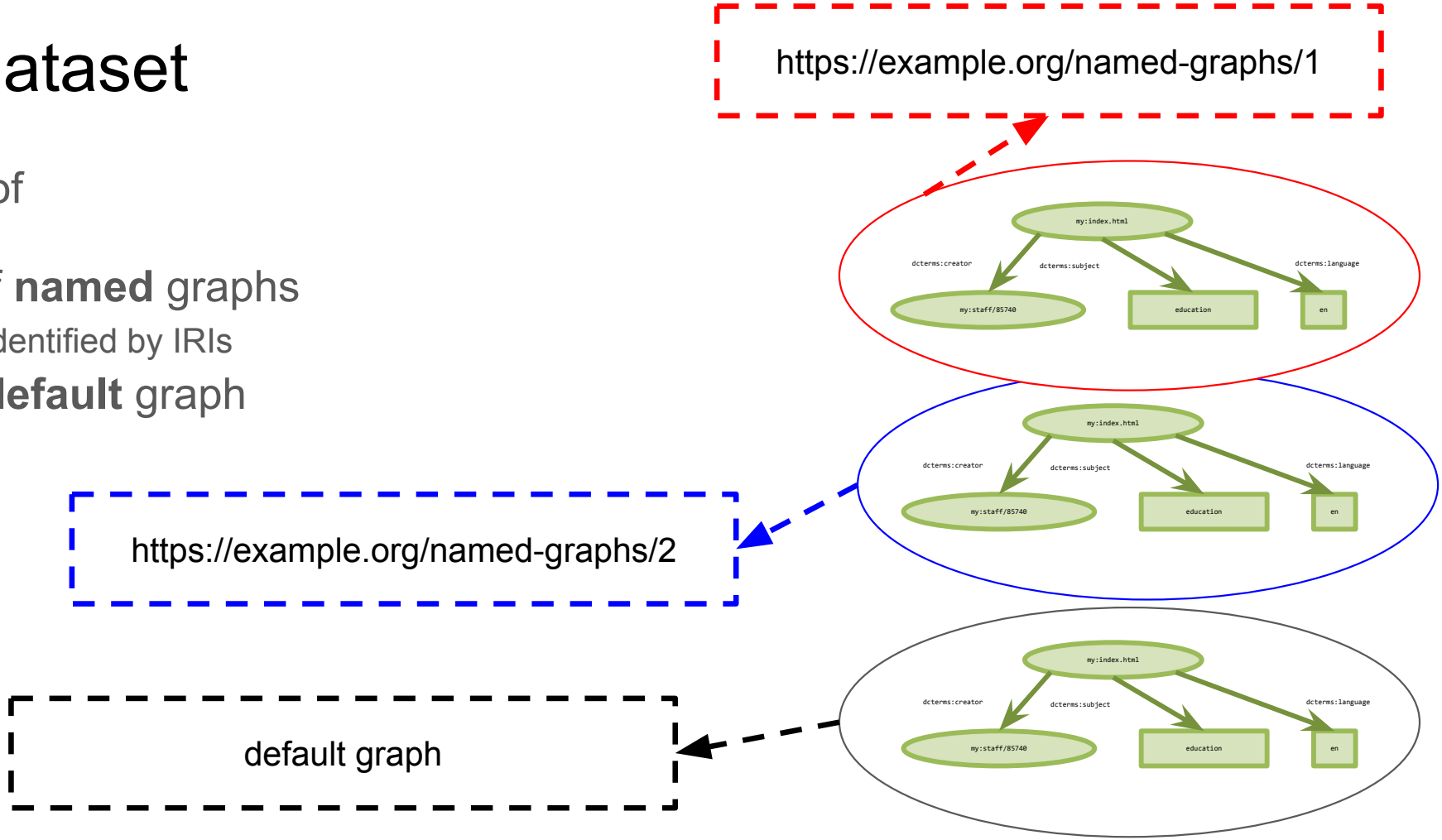
- S P O G
- G can be used as subject of another triple (quad)



RDF dataset

Consists of

- set of **named** graphs
  - identified by IRIs
- one **default** graph



Back to RDF serializations

RDF 1.1 N-Quads

- W3C Recommendation
  - 25 February 2014
- Based on N-Triples, adds support for named graphs

S P O G

```
<http://example.org/#spiderman>
<http://www.perceive.net/schemas/relationship/enemyOf>
<http://example.org/#green-goblin>
<http://example.org/graphs/spiderman> .
```

RDF 1.1 TriG

- W3C Recommendation
  - 25 February 2014
- Based on RDF Turtle, adds support for named graphs

RDF Dataset consists of

- 1 default graph
- N named graphs

```
@base <http://www.w3.org/People/> .
@prefix : <http://xmlns.com/foaf/0.1/> .

# default graph
{
     ericFoaf:ericP :givenName "Eric" .
}

# also default graph, no {}
ericFoaf:ericP :givenName "Eric" .

# graph highlight
GRAPH <Eric/ericP-foaf.rdf> {
      ericFoaf:ericP :givenName "Eric" .
}
```

RDFS: RDF Schema

RDFS - RDF Schema 1.1

W3C Recommendation

- 25 February 2014

Vocabulary for creation of other RDF vocabularies

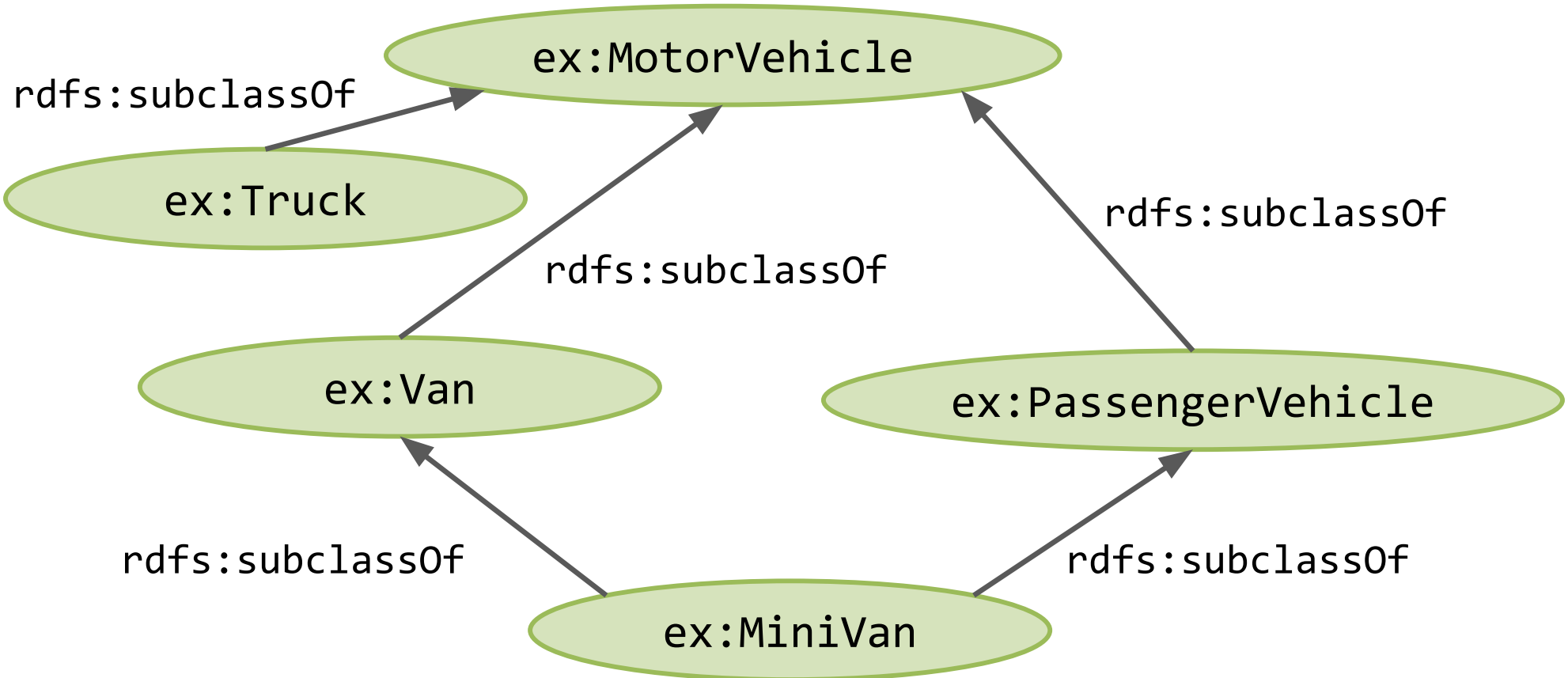
RDF Vocabulary

- collection of classes and properties, their IRIs and their definitions

RDFS - defining classes and class hierarchies

```
ex:MotorVehicle       rdf:type          rdfs:Class .
ex:PassengerVehicle   rdf:type          rdfs:Class .
ex:Van                rdf:type          rdfs:Class .
ex:Truck              rdf:type          rdfs:Class .
ex:MiniVan            rdf:type          rdfs:Class .

ex:PassengerVehicle   rdfs:subClassOf   ex:MotorVehicle .
ex:Van                rdfs:subClassOf   ex:MotorVehicle .
ex:Truck              rdfs:subClassOf   ex:MotorVehicle .

ex:MiniVan            rdfs:subClassOf   ex:Van .
ex:MiniVan            rdfs:subClassOf   ex:PassengerVehicle .
```

RDFS - defining classes and class hierarchies



RDFS - defining properties and property hierarchies

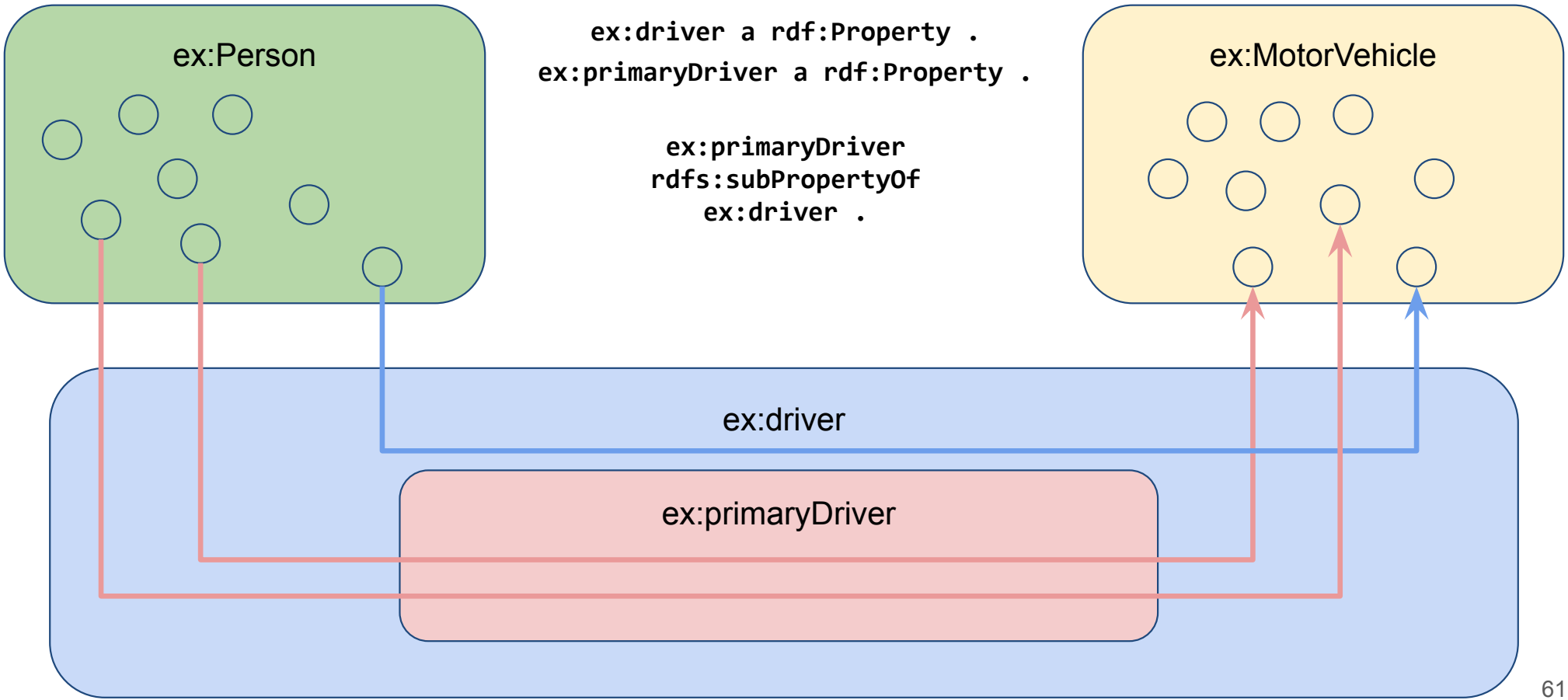
```
ex:Person    rdf:type      rdfs:Class .
ex:author    rdf:type      rdf:Property .
ex:author    rdfs:range    ex:Person .

ex:hasMother    rdf:type      rdf:Property .
ex:hasMother    rdfs:range    ex:Female .
ex:hasMother    rdfs:domain   ex:Person .

ex:age    rdf:type     rdf:Property .
ex:age    rdfs:range   xsd:integer .

exterms:weight    rdfs:domain    ex:Book .
exterms:weight    rdfs:domain    ex:MotorVehicle .

ex:driver           rdf:type             rdf:Property .
ex:primaryDriver    rdf:type             rdf:Property .
ex:primaryDriver    rdfs:subPropertyOf   ex:driver .
```

Big difference between RDFS and object-oriented programming (OOP): RDF properties are *first class citizens*.

They can exist on their own, independently of any class.

RDFS - property hierarchies

```
ex:driver a rdf:Property .
ex:primaryDriver a rdf:Property .

ex:primaryDriver
rdfs:subPropertyOf
ex:driver .
```

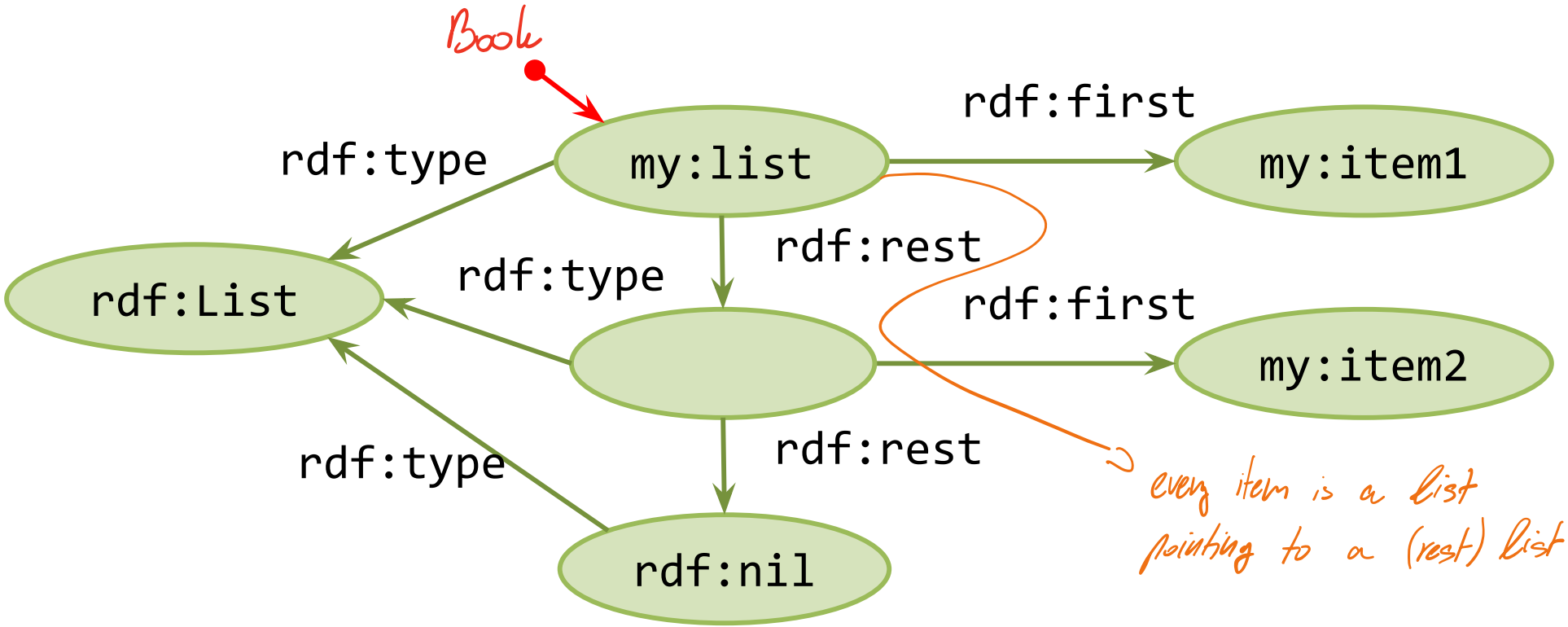


RDFS: label, comment, seeAlso

- `rdfs:label`
  - Human readable name of a resource
- `rdfs:comment`
  - Longer description of a resource
- `rdfs:seeAlso`
  - Points to a resource that might provide more information about the subject resource
- `rdfs:isDefinedBy`
  - In a sense not specified by RDF

Book of authors example:

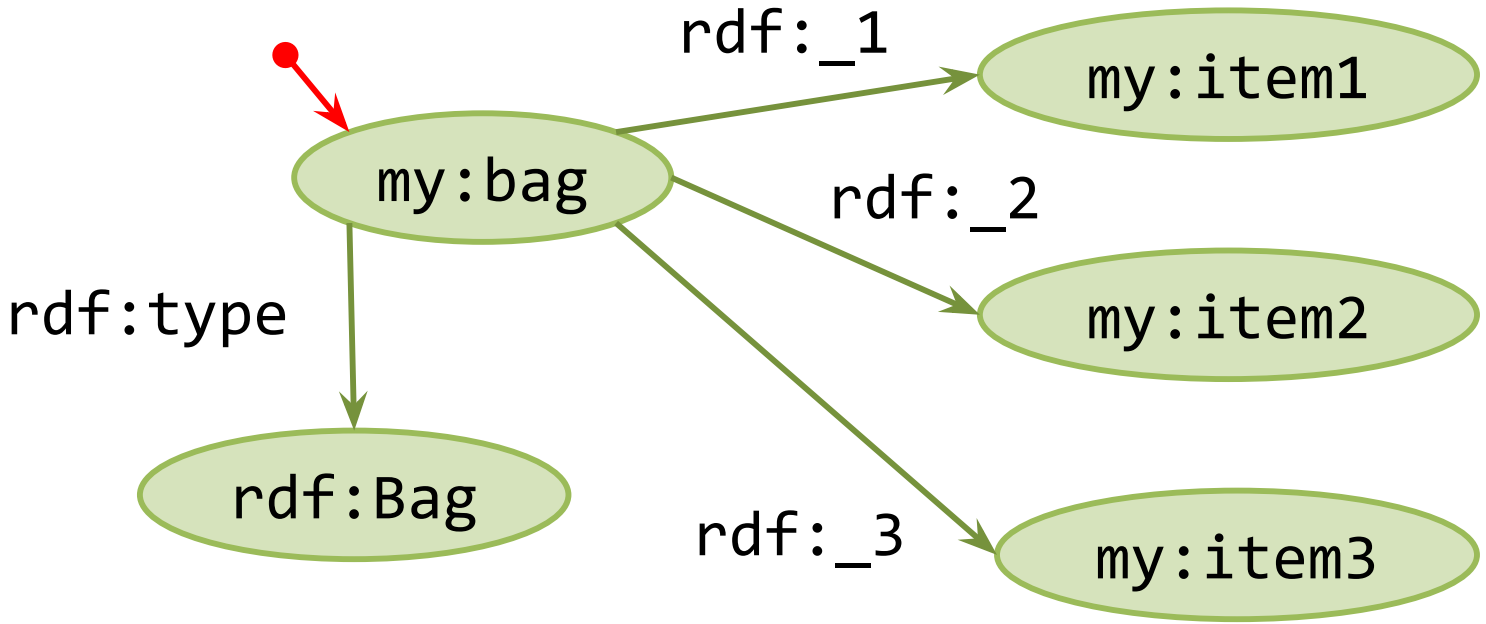
RDF model: `rdf:List` for closed collection



It's linked list representation

RDF model: containers for open collections

`rdf:Bag, rdf:Seq, rdf:Alt`



RDF 1.1 Turtle - `rdf:List` shortcut

```
# the value of this triple is the RDF collection blank node

:subject :predicate ( :a :b :c ) .


# an empty collection value - rdf:nil

:subject :predicate2 () .
```

RDFS - RDF Schema 1.1

Functionally different from other schema languages, e.g. XML Schema

RDF is *“schema-less”*

W3C Recommendation

- 25 February 2014

Vocabulary for creation of other RDF vocabularies

RDF Vocabulary

- collection of classes and properties and their definitions

Open World Assumption (OWA)

*"**open-world assumption** is the assumption that the truth value of a statement may be true irrespective of whether or not it is known to be true"*

-- Open-world assumption

```
Statement: "Mary" "is a citizen of" "France"
Question: Is Paul a citizen of France?

"Closed world" (for example SQL) answer: No.
"Open world" answer: Unknown.
```

Linked Data

Regular data ~ not linked

Hey, look at **1234**, he's cool!

Stars **in**, or stars **as**?

This 1234?

ID	Jméno	Hraje
1235	Joaquin Phoenix	Joker
1234	Robert De Niro	Joker
...		

Where and **how** do I get data about 1234?

Which Joker, the 5.6 or the 9.0 one?

Or this 1234?

```
[{
     "id": "1234",
     "id2": "az-11",
     "name": {
          "en": "Joker"
     },
     "rating": 9.0
}, {
     "id": "1235",
     "id2": "yt-18",
     "name": {
          "en": "Joker"
     },
     "rating": 5.6
}]
```

Issues with regular, non-linked data

ID	Name	Stars
1235	Joaquin Phoenix	Joker
1234	Robert De Niro	Joker
...		

i.e. CSV, JSON, XML, Excel files...

- Ambiguous identification of entities in data
  - Person with ID aaa1234 in a document located on my laptop in folder /data/temp/people.json
  - Another person with ID aaa1234 in the XML file on this CD

- Low findability and accessibility of data
  - Get data about person aaa1234 => Go to my laptop, open the folder, load/open the file, search/query
- No contextual information
  - Person aaa1234 lives in Prague. I want to know more about Prague.
    Where and how do I get the information?

Issues => Additional requirements on data

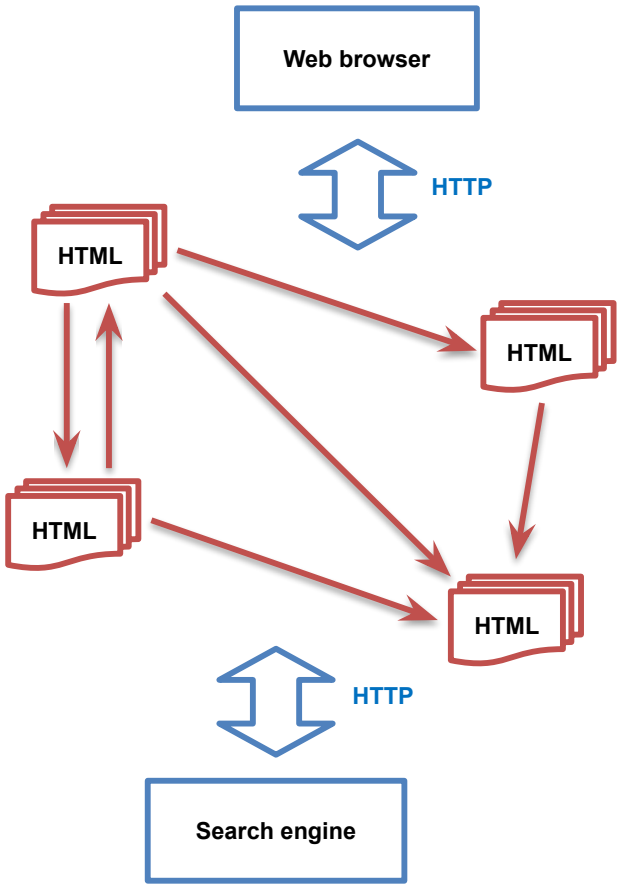
i.e. CSV, JSON, XML, ...

- Identification of entities in data
  - Global
  - Unique

- Findability and accessibility of data
  - Find data based on the identification
  - Access it in single format
- Contextual information
  - When I access information, I want to know where and how to find more

Is there such a system?

The World Wide Web



Shared global space of documents

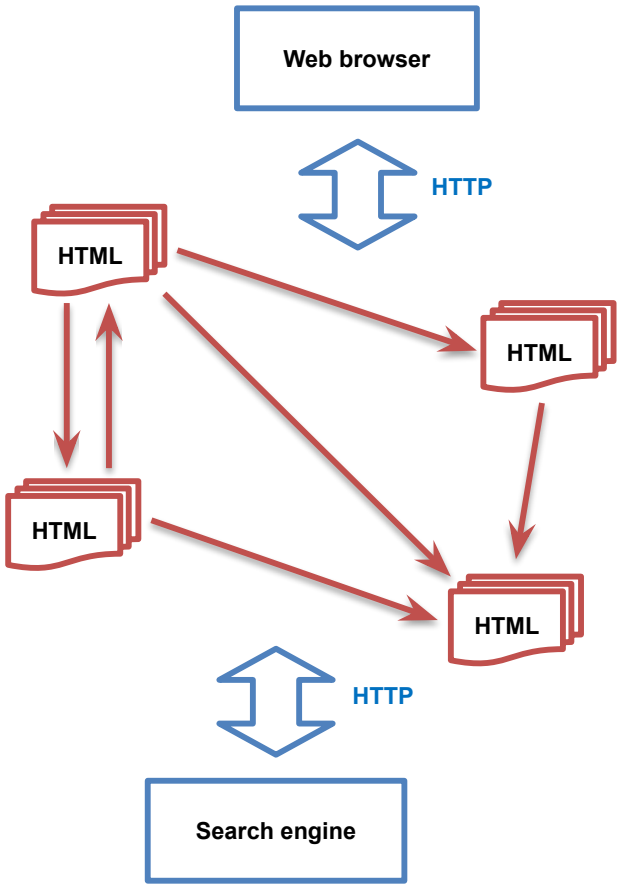
Built on top of several **simple** principles:

1. HTML as a format for publishing documents
2. URLs as unique global identifiers of documents
3. HTTP for localization and accessing documents by their URLs
4. hyperlinks between documents

There are two kinds of applications working in **this space of documents**:

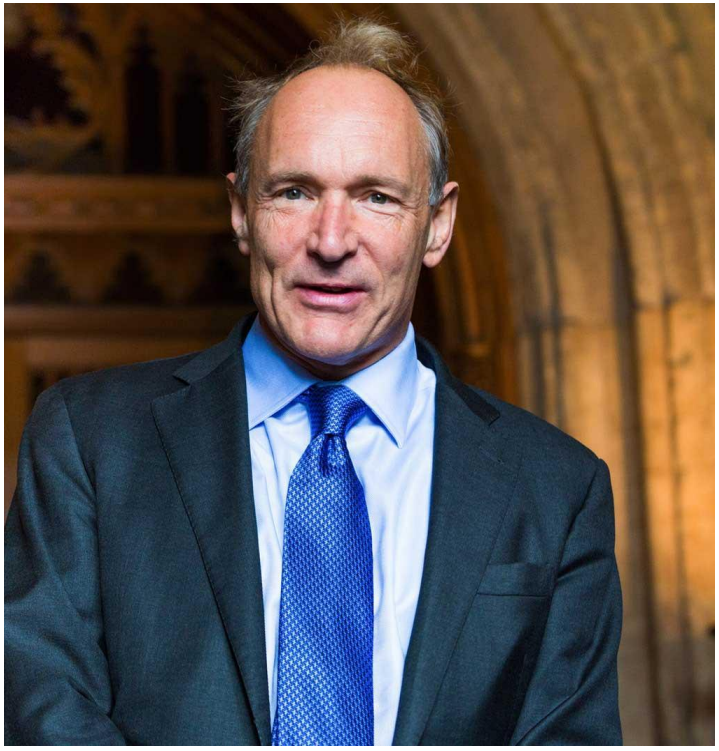
- **web browsers** (localizing and browsing documents through hyperlinks)
- **search engines** (indexing and full text searching of documents)

The World Wide Web - what can we do with it?



- Publish human-readable documents
- Everyone can view them in their browser
  - if they know the URL
- + links
  - To documents with yet unknown URL
  - From other documents
  - From catalogs
- Fulltext search, keyword search

The World Wide Web - 30 years, Sir Tim Berners-Lee

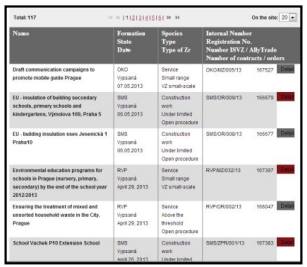


30th Anniversary of the World Wide Web @ CERN (March 2019)

From the Web to Linked Data

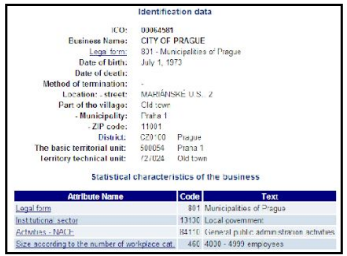
Web of Documents

Prague public contracts



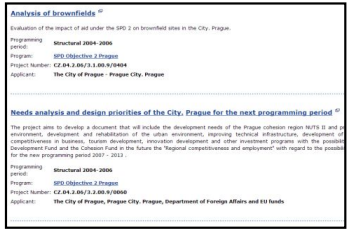
http://www.praha.eu

Basic info about Prague



http://registry.czso.cz

EU funded projects in Prague



http://www.risy.cz

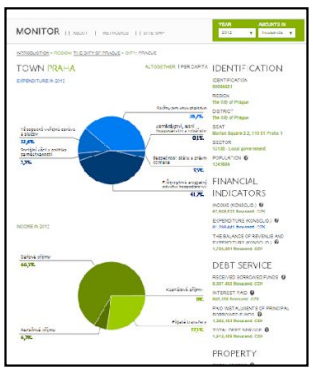
Demography of Prague

http://www.czso.cz

Lots of information about **Prague** in the Web of Documents. Problems:

- Encoded in documents distributed across the Web
- Documents intended for humans not computers
- Documents about Prague or related things not linked
- Computers not able to process data about Prague published on the Web

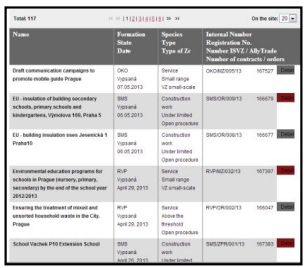
Prague budget



http://monitor.statnipokladna.cz

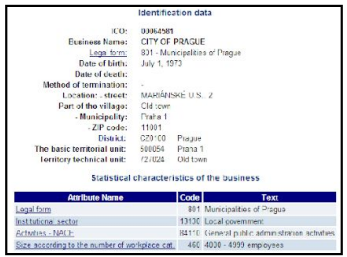
Web of Documents

Prague public contracts



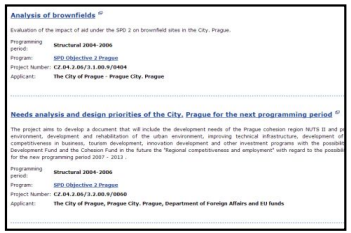
http://www.praha.eu

Basic info about Prague



http://registry.czso.cz

EU funded projects in Prague



http://www.risy.cz

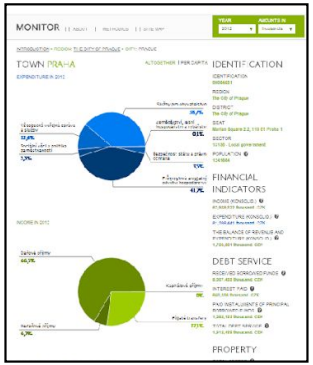
Demography of Prague

http://www.czso.cz

Queries, for which there is information:

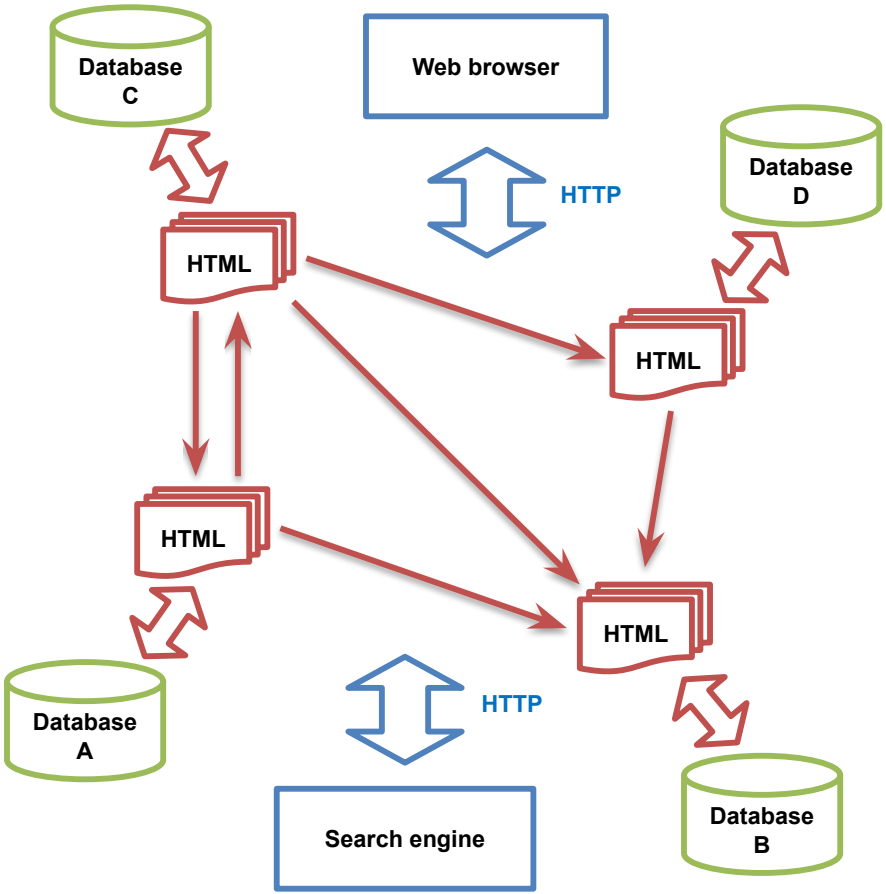
- Top 100 suppliers of Prague with headquarters outside of Prague region.
- Money spent in Prague for new children playgrounds in the last 5 years per one child.
- Organizations in Prague funded by EU structural funds and their top 100 suppliers.

Prague budget



http://monitor.statnipokladna.cz

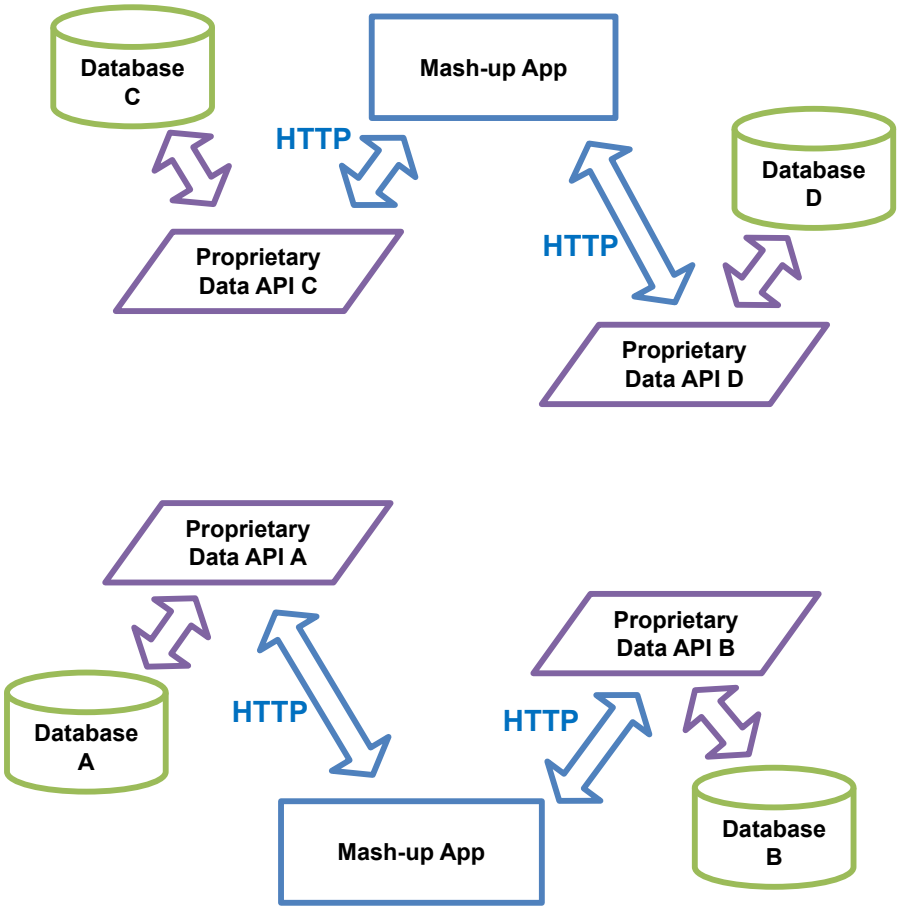
The World Wide Web



- Typically, there are underlying databases
- From which, human readable documents are generated

- ...and scraped by users who want to query it

The World Wide Web

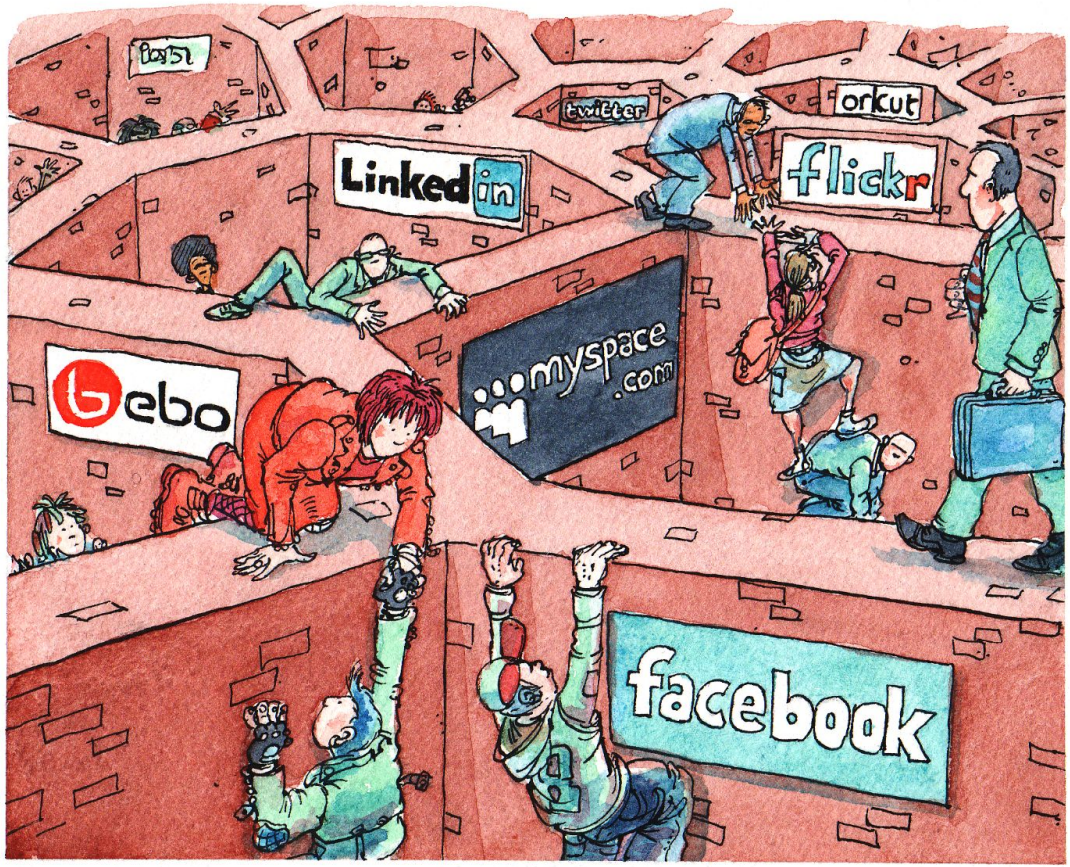


Different APIs provide machine readable data for further processing in so called mash-up applications.

Also built on several simple principles:

- XML/JSON as formats for publishing data
- HTTP protocol for transferring data between APIs and applications

Social network silos



https://www.w3.org/2010/Talks/0303-socialcloud-tbl/#(2)

Problems with data on the current Web

Web of Documents	Current Web IS NOT Web of Data!
URLs as unique global identifiers of documents	no unique global identifiers of things
HTML as a format for publishing documents	many formats for publishing data (XML, JSON, CSV, XLS, ...)
HTTP for localization and accessing documents by their URLs	HTTP for localization of APIs and accessing them (REST) [but not for localization of *things* and accessing their data]
hyperlinks between documents	none of current formats enables us to link related entities

Can we apply the principles of the Web to data?

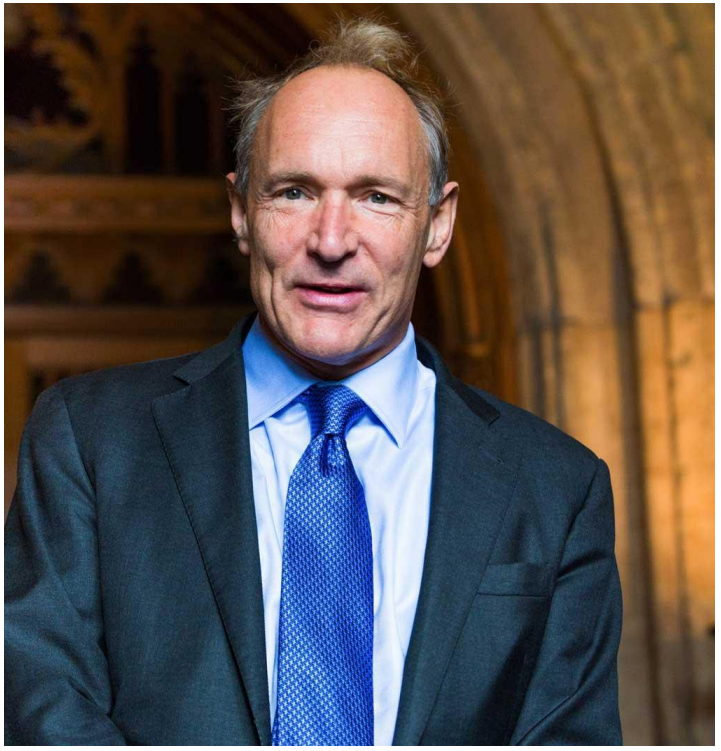
Linked Data ~ the Web of Data

Principles, “best practices” for publishing and linking data about entities on the Web.

- Application of the proven principles of Web of Documents to data
- 2 main goals
  - Machine readable and understandable data (based on the Semantic Web)
  - Providing context to data (via links to other data)

The principles of Linked Data

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



https://5stardata.info/

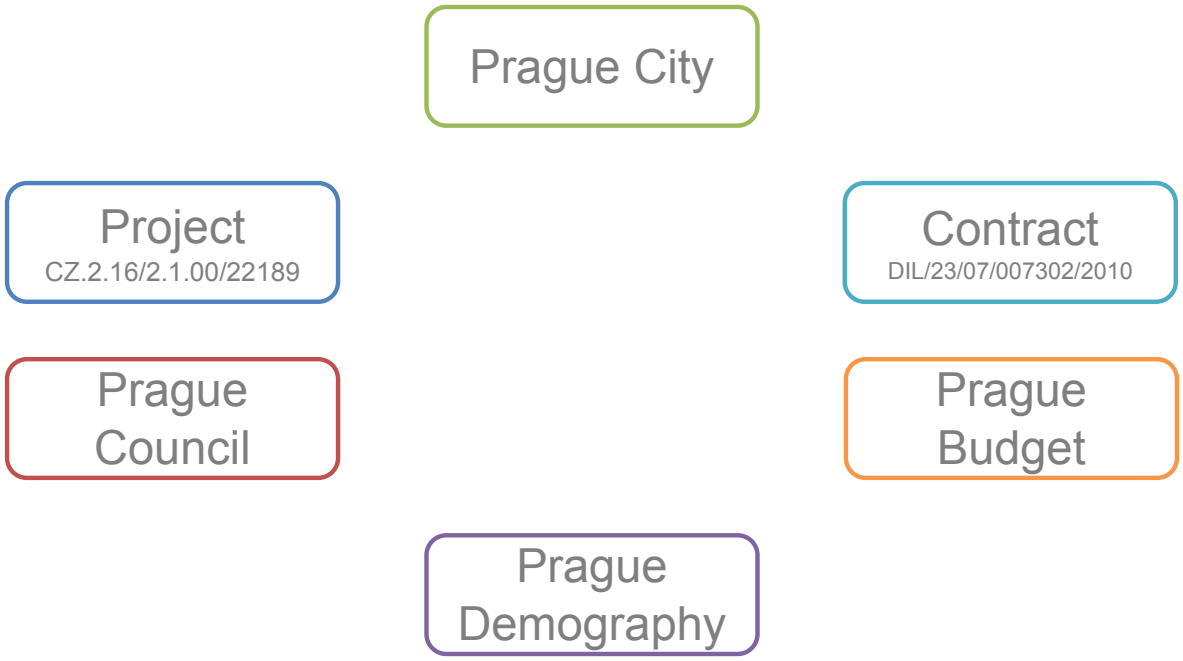
Web of Documents without the first two principles

Web pages without URLs

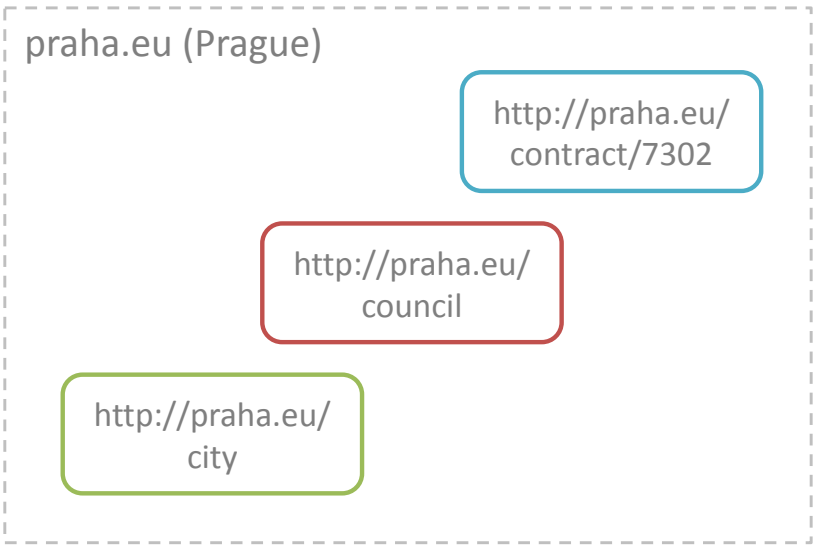
- What web page are you talking about?
- Where do I get the web page you are talking about?
- How do I get the web page you are talking about?
- ...

ID	Name	Stars
1235	Joaquin Phoenix	Joker
1234	Robert De Niro	Joker
...		

Things as first-class citizens

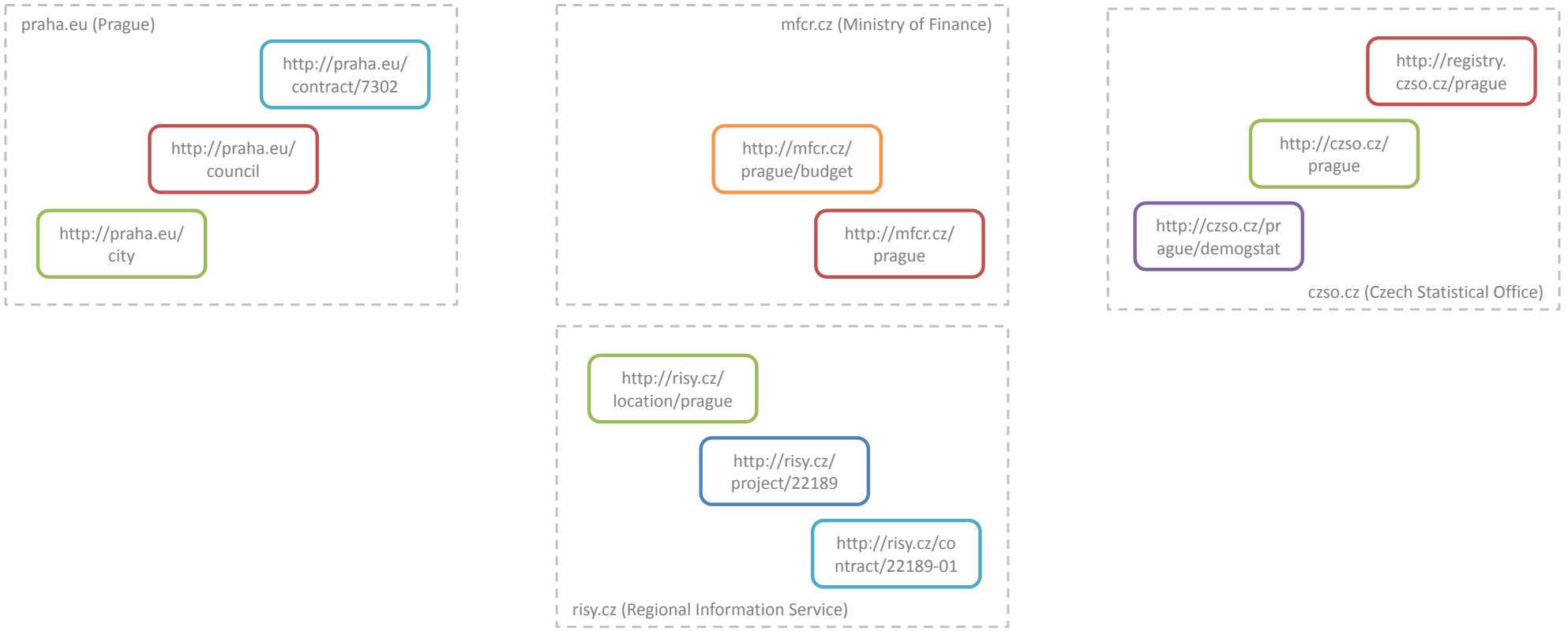


1. + 2. Use HTTP URIs as names for things



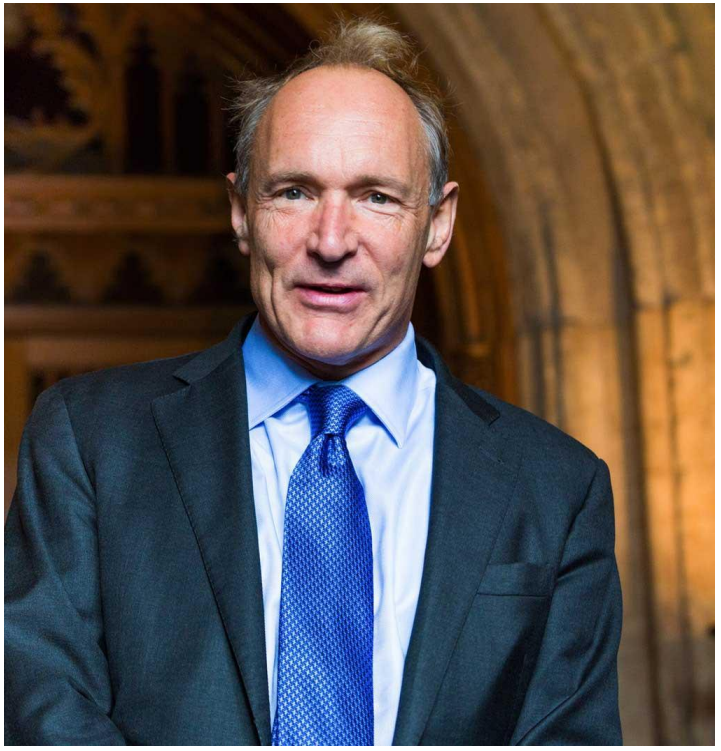
1. + 2. Use HTTP URIs as names for things

1. + 2. Use HTTP URIs as names for things



The principles of Linked Data

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



https://5stardata.info/

Web of Documents without the third principle

Web pages are in many formats, not only HTML

Thanks for the URI of your web page

- In what language/format is your page?
- Which software supports your language for pages?
- How many browsers do you have?
- …

… we all know this - how many times you click on a link and PDF/Word/Excel opens

Technical detour: HTTP Accept header and URIs

Web browser

Applications

HTTP
(HTML)

http://esfcr.cz/.../projekt/
CZ10421016300169

HTTP
(RDF)

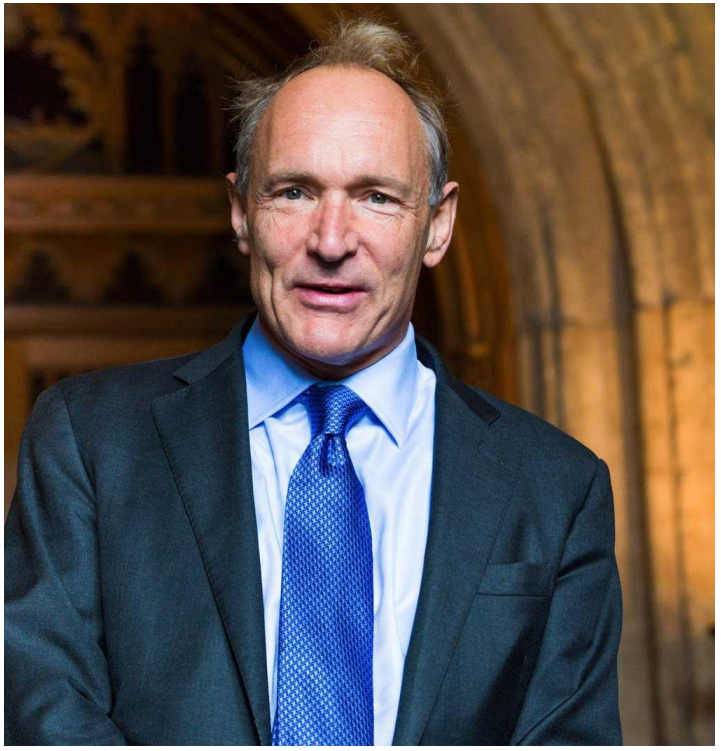
http://esfcr.cz/.../projekt/
CZ10421016300169

```
<http://esfcr.cz/data/projekt/CZ10421016300169>
    esf:nazev "INNOSTART" ;
    esf:registracni_cislo "CZ.1.04/2.1.01/63.00169" ;
    esf:castka "4711681" ;
    esf:realizace_od "2011-06-01" ;
    esf:realizace_do "2013-03-31" ;
    esf:realizator <http://esfcr.cz/.../25438352> ;
    esf:partner <http://esfcr.cz/.../25438352> ;
    ontaktni_osoba <http://esfcr.cz/.../8541274571>;
    egion <http://esfcr.cz/.../ustecky> .
```

esfcr.cz

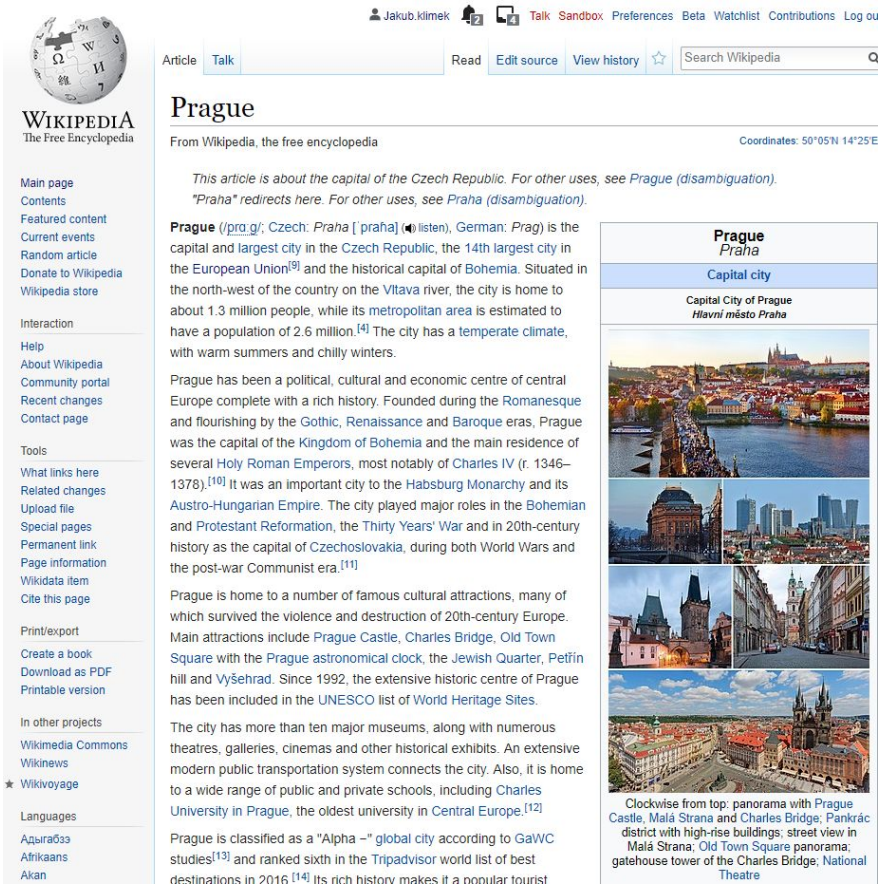
The principles of Linked Data

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



https://5stardata.info/

Web of Documents without the fourth principle



Jakub.klimek Talk Sandbox Preferences Beta Watchlist Contributions Log ou

Article Talk Read Edit source View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects
Wikimedia Commons
Wikinews
Wikivoyage

Languages
Адыгабзэ
Afrikaans
Akan

Prague

From Wikipedia, the free encyclopedia

Coordinates: 50°05′N 14°25′E

This article is about the capital of the Czech Republic. For other uses, see Prague (disambiguation).
"Praha" redirects here. For other uses, see Praha (disambiguation).

Prague (/prɑːɡ/; Czech: *Praha* [ˈpraɦa] listen), German: *Prag*) is the capital and largest city in the Czech Republic, the 14th largest city in the European Union[9] and the historical capital of Bohemia. Situated in the north-west of the country on the Vltava river, the city is home to about 1.3 million people, while its metropolitan area is estimated to have a population of 2.6 million.[4] The city has a temperate climate, with warm summers and chilly winters.

Prague has been a political, cultural and economic centre of central Europe complete with a rich history. Founded during the Romanesque and flourishing by the Gothic, Renaissance and Baroque eras, Prague was the capital of the Kingdom of Bohemia and the main residence of several Holy Roman Emperors, most notably of Charles IV (r. 1346–1378).[10] It was an important city to the Habsburg Monarchy and its Austro-Hungarian Empire. The city played major roles in the Bohemian and Protestant Reformation, the Thirty Years' War and in 20th-century history as the capital of Czechoslovakia, during both World Wars and the post-war Communist era.[11]

Prague is home to a number of famous cultural attractions, many of which survived the violence and destruction of 20th-century Europe. Main attractions include Prague Castle, Charles Bridge, Old Town Square with the Prague astronomical clock, the Jewish Quarter, Petřín hill and Vyšehrad. Since 1992, the extensive historic centre of Prague has been included in the UNESCO list of World Heritage Sites.

The city has more than ten major museums, along with numerous theatres, galleries, cinemas and other historical exhibits. An extensive modern public transportation system connects the city. Also, it is home to a wide range of public and private schools, including Charles University in Prague, the oldest university in Central Europe.[12]

Prague is classified as a "Alpha −" global city according to GaWC studies[13] and ranked sixth in the Tripadvisor world list of best destinations in 2016.[14] Its rich history makes it a popular tourist

Prague
Praha

Capital city

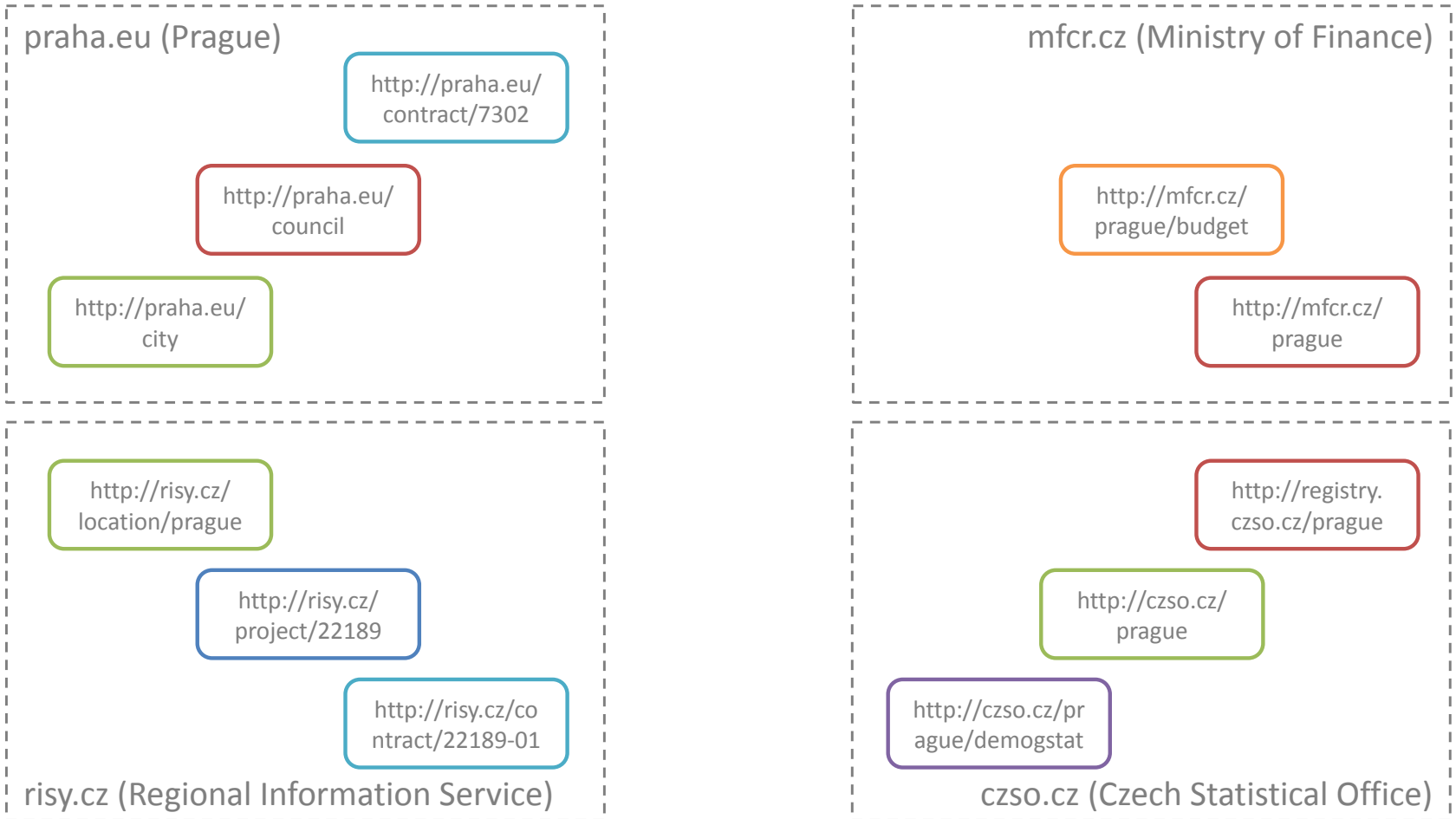
Capital City of Prague
Hlavní město Praha

Clockwise from top: panorama with Prague Castle, Malá Strana and Charles Bridge; Pankrác district with high-rise buildings; street view in Malá Strana; Old Town Square panorama; gatehouse tower of the Charles Bridge; National Theatre

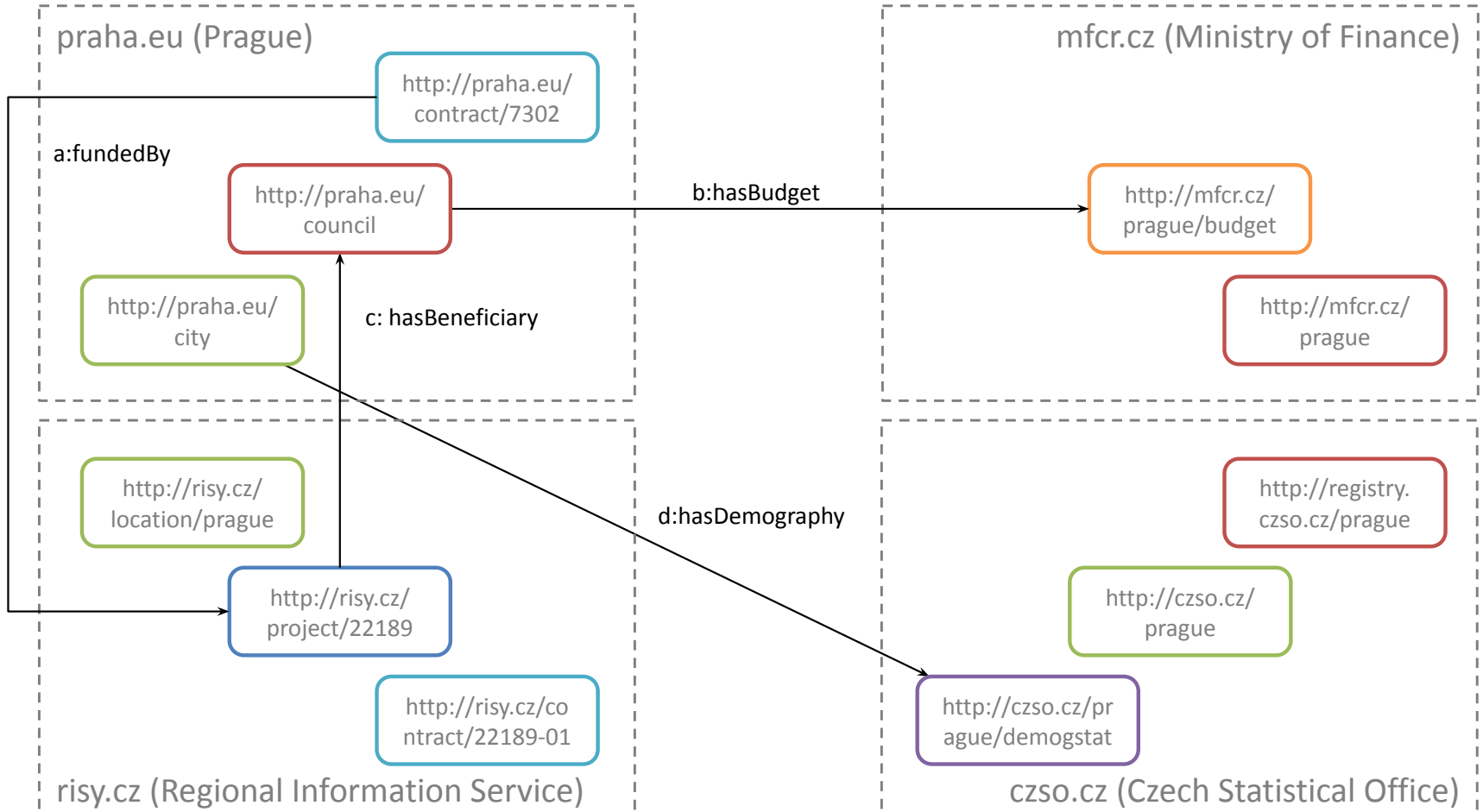
Web pages without links to other pages

- Imagine a Wiki page with no links
- Where do I get more information about the things mentioned in the article?
- Is the thing mentioned in the article really the one I think it is?

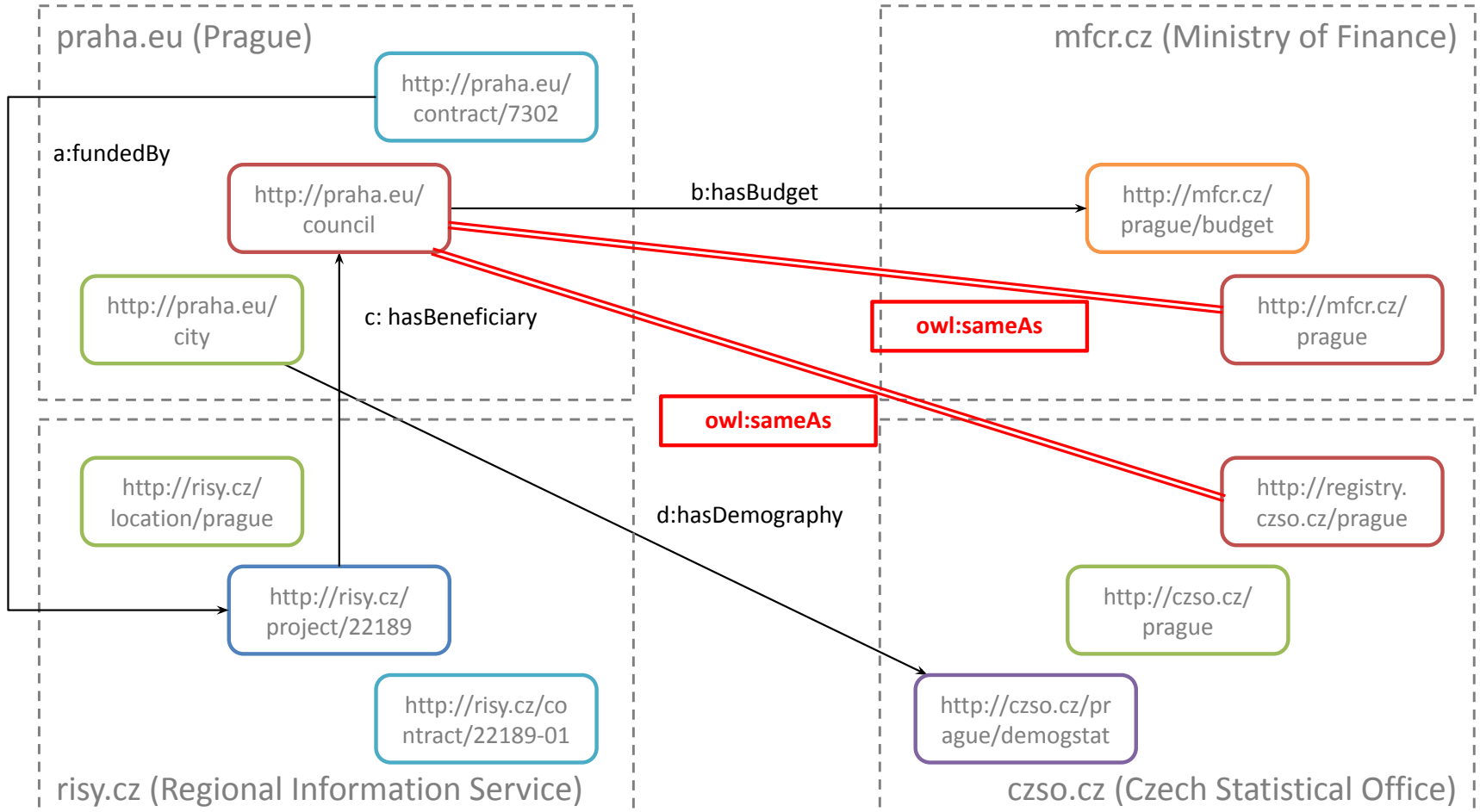
4. Include links to other URIs (provide context)



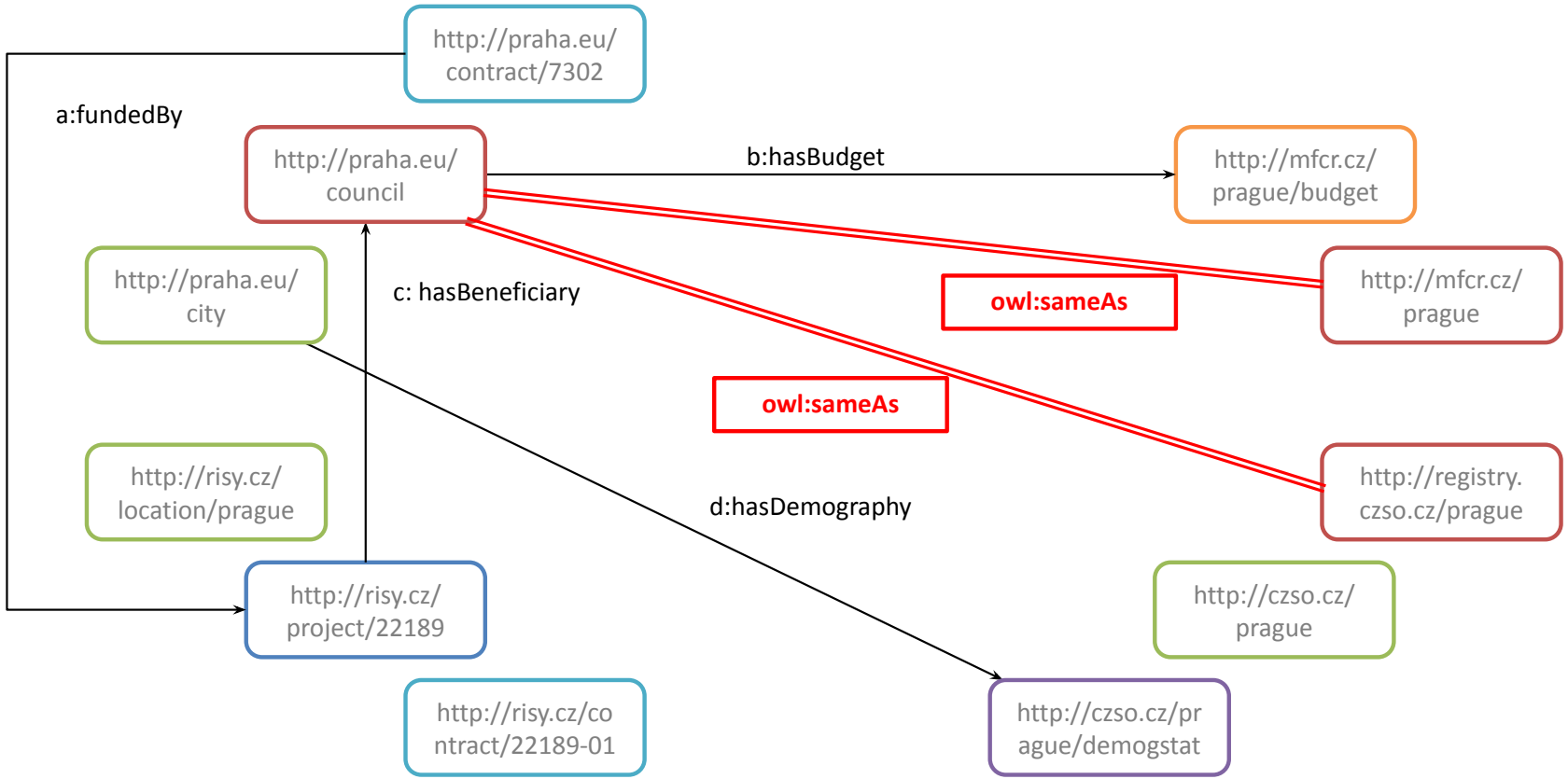
4. Include links to other URIs (provide context)



4. Include links to other URIs (provide context)



The Web of Data



Web of documents vs. Web of data (Linked Data)

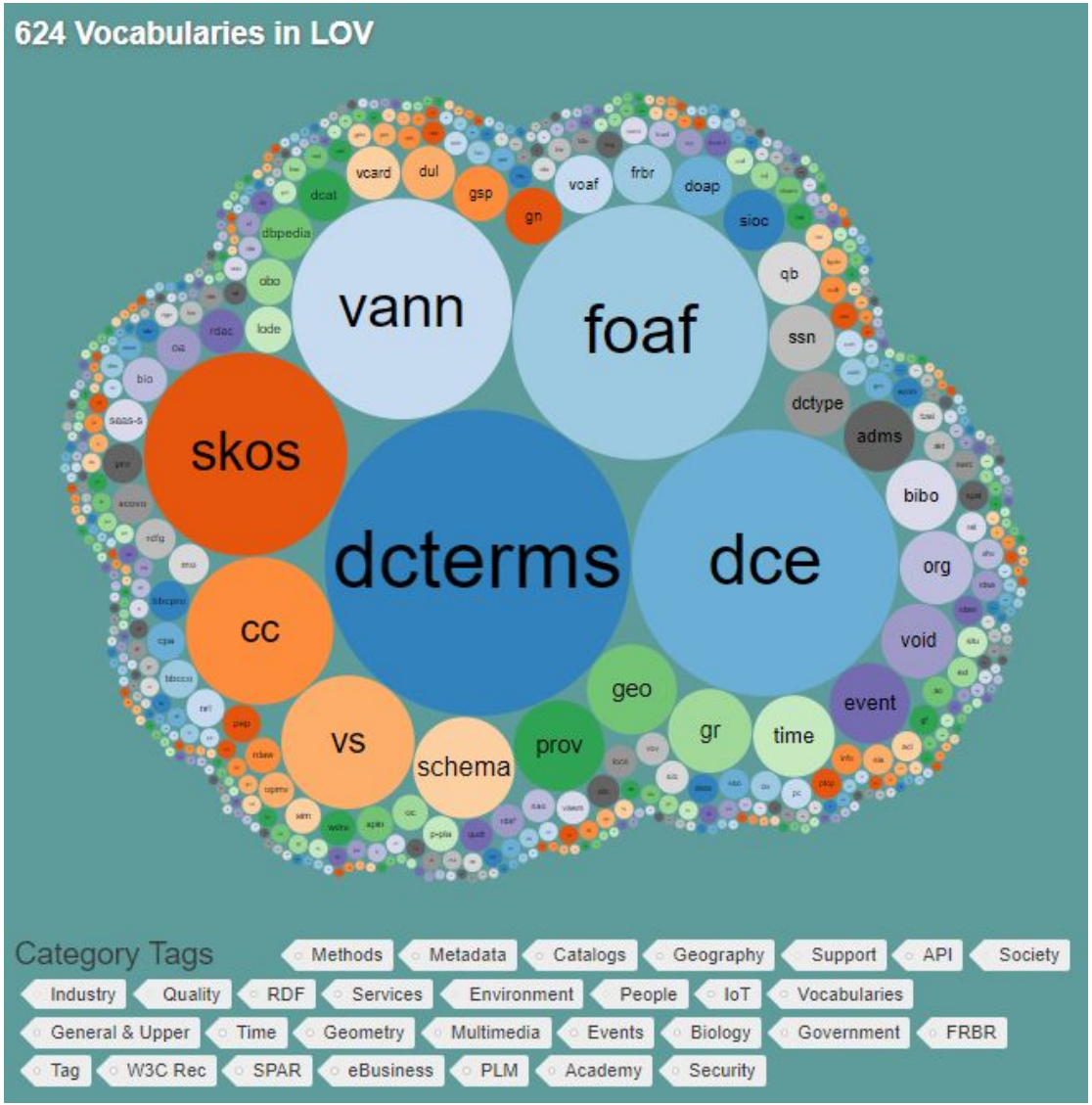
Web of documents	Linked Data
HTML as **document** publication format	RDF as a **data** publication format
URL as a unique global **document** identifier	URL as a unique global **entity** identifier
HTTP protocol for accessing **documents** using their URL	HTTP protocol for accessing **data about entities** using their URL
Links to other **documents**	Links to other **entities**
	vocabularies – standards for common data representation

Linked Open Vocabularies

Catalog of vocabularies used on the Web of Data

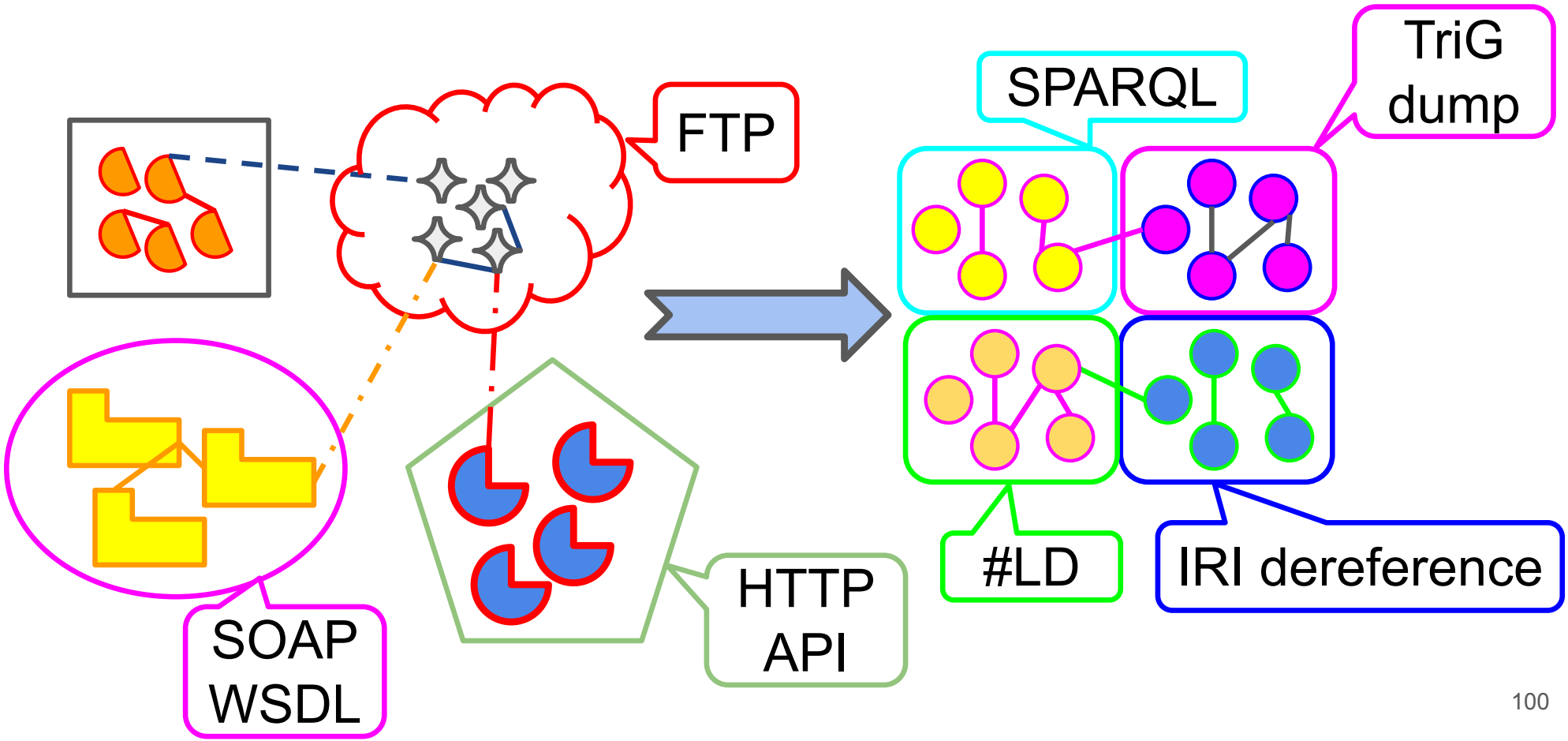
Basic rule - vocabulary reuse

- Schema.org
- Dublin Core Vocabulary
- Data Cube Vocabulary
- Simple Knowledge Organization System (SKOS)



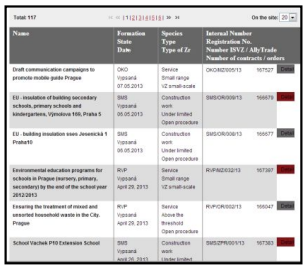
https://lov.linkeddata.es/dataset/lov/

Linked Data = Technical interoperability solution



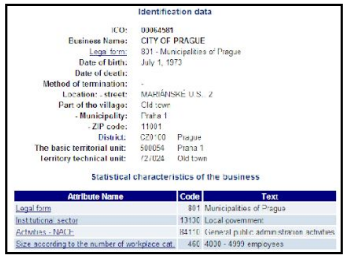
Web of Data

Prague public contracts



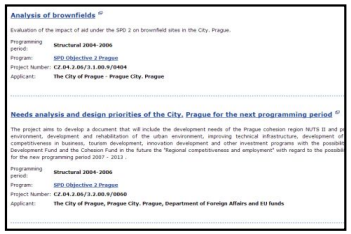
http://www.praha.eu

Basic info about Prague



http://registry.czso.cz

EU funded projects in Prague



http://www.risy.cz

Demography of Prague

http://www.czso.cz

If the data from those web sites was published as Linked Data, getting the answer to the queries, e.g.

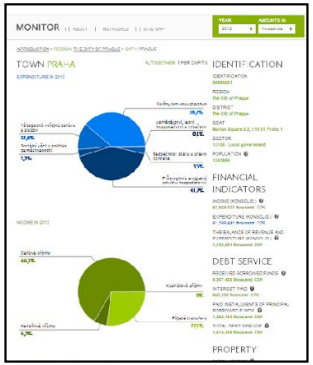
- e.g. Money spent in Prague for new children playgrounds in the last 5 years per one child.

would have only one step:

1. Write the query answering the question

(in the worst case there would be a preceding "download data" step if federated querying was not supported)

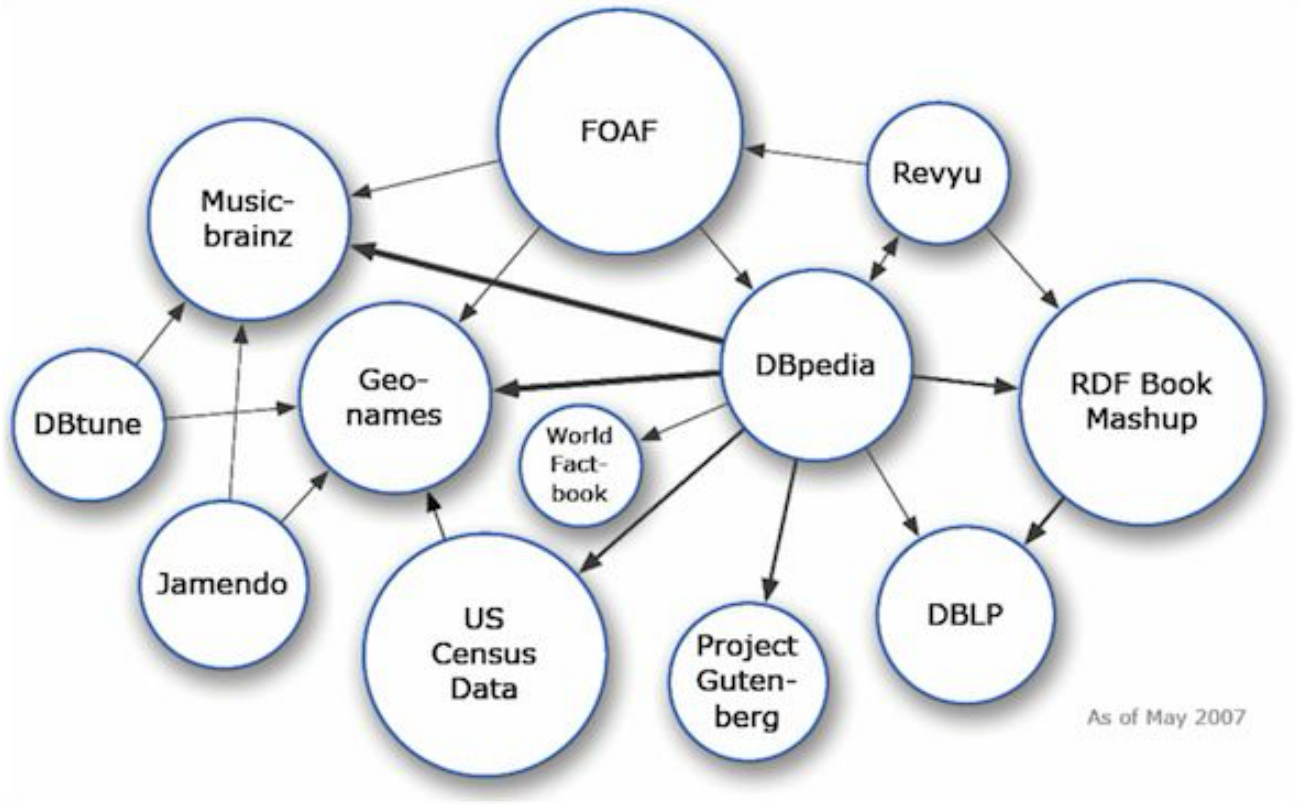
Prague budget



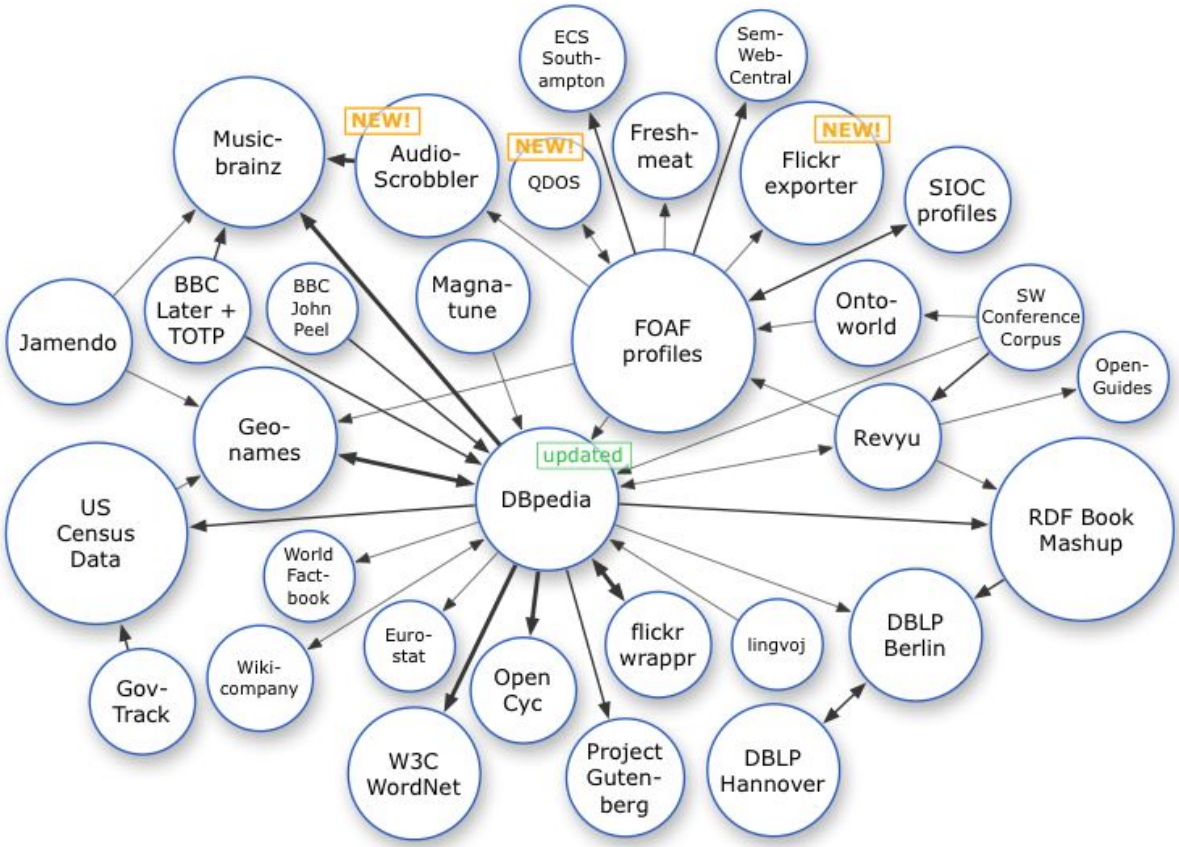
http://monitor.statnipokladna.cz

The Web of Data - will it be successful?

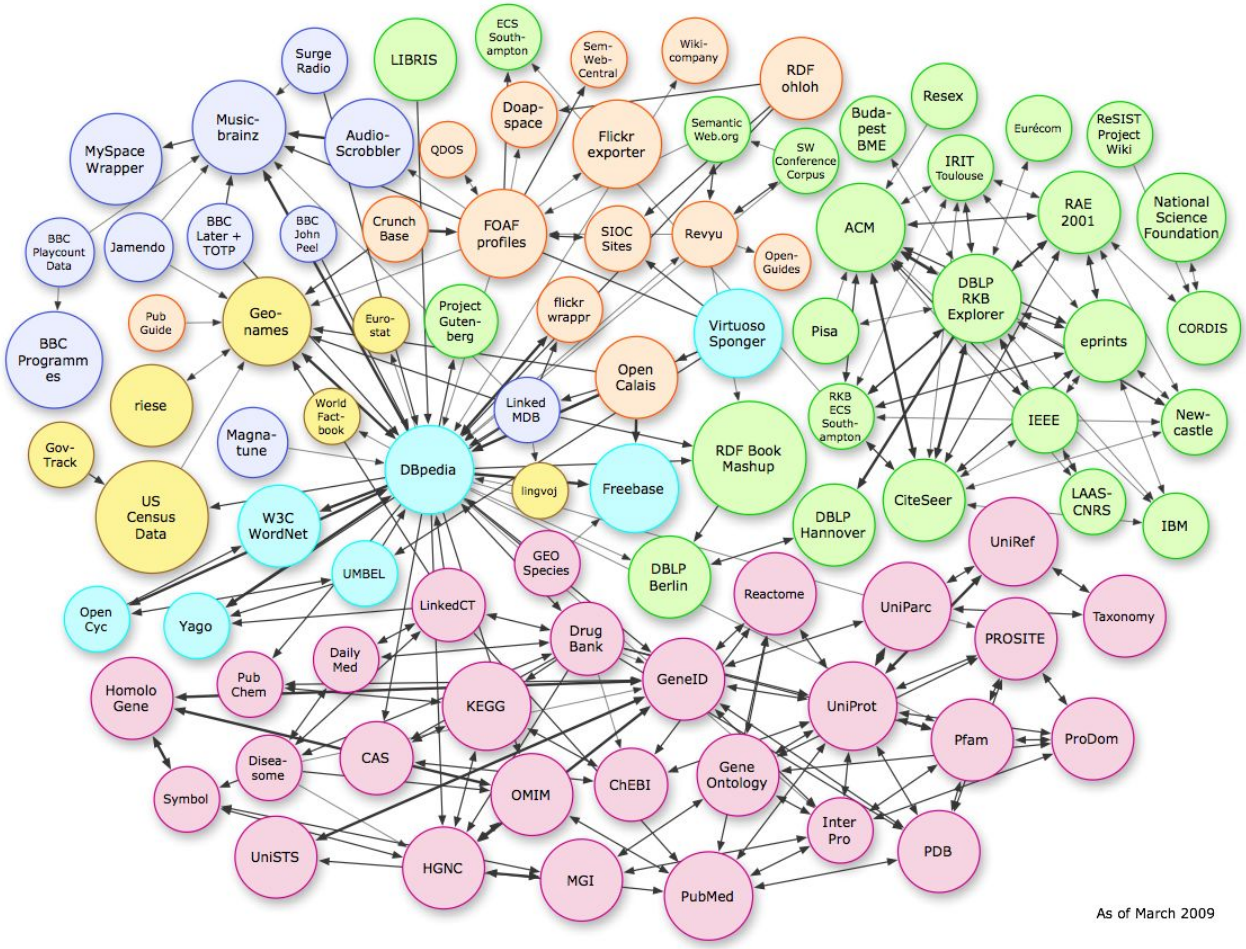
The LOD cloud - 2007 - 12 datasets



The LOD cloud - 2008 - 32 datasets

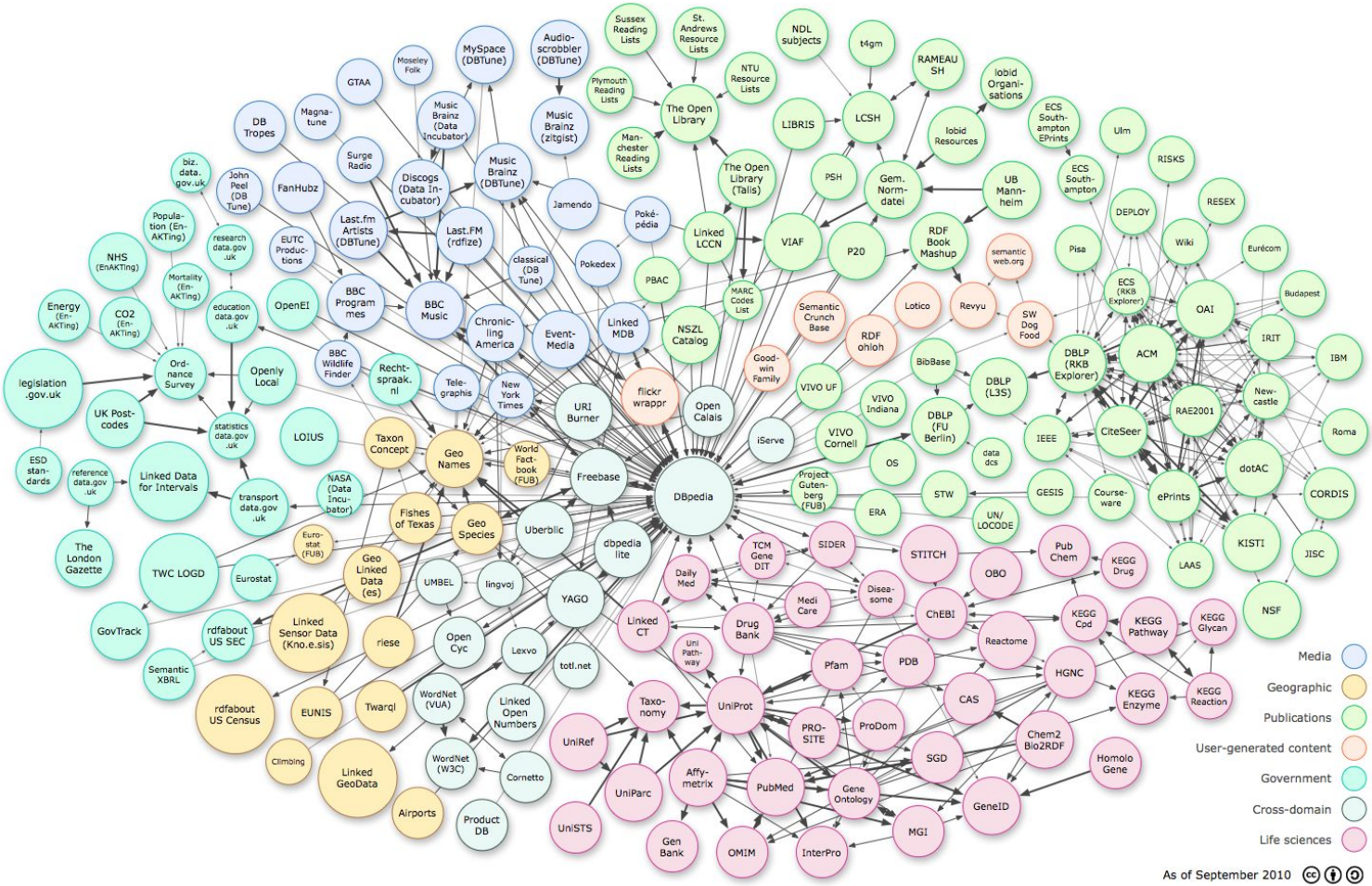


The LOD cloud - 2009 - 89 datasets

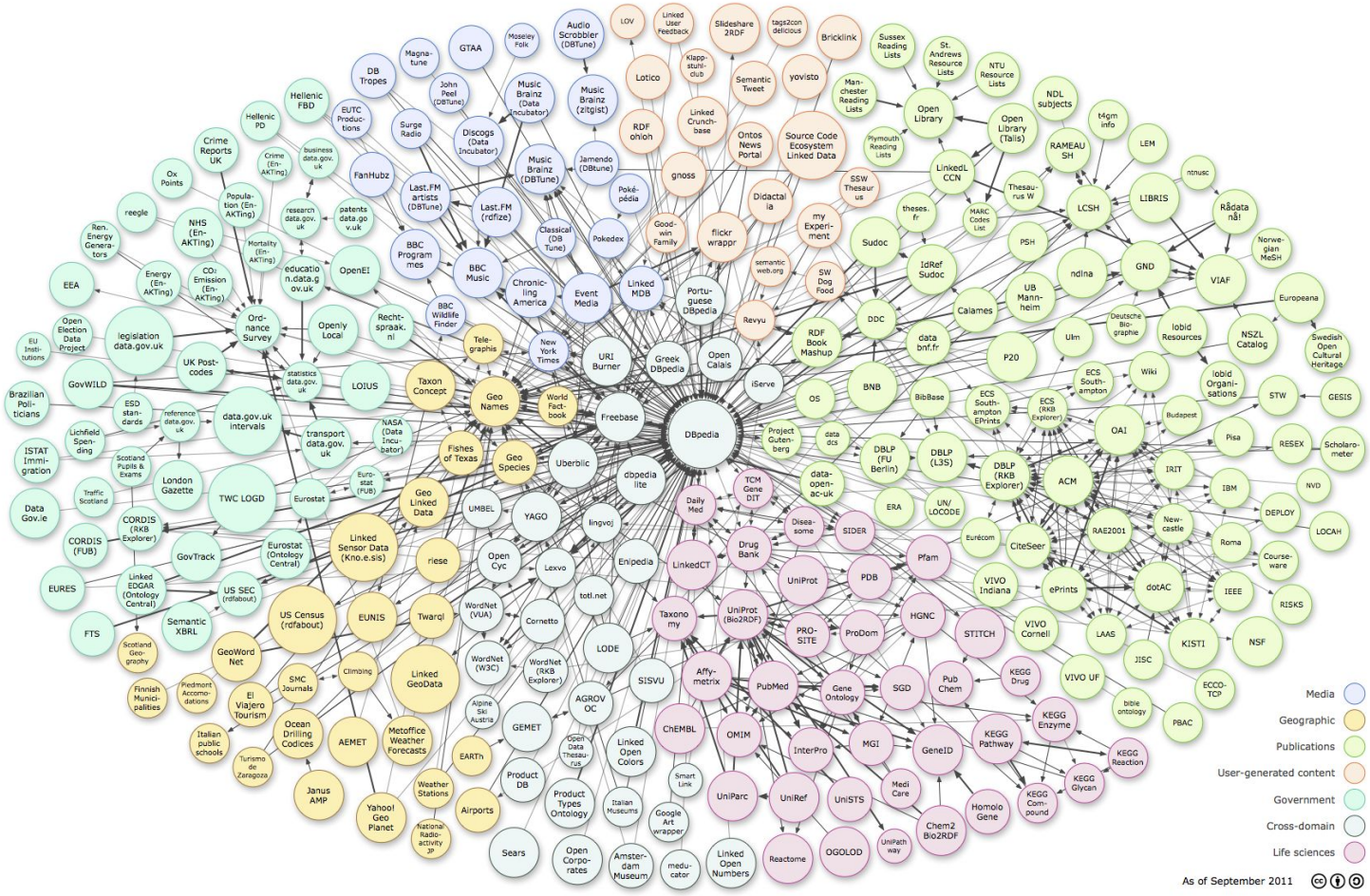


As of March 2009

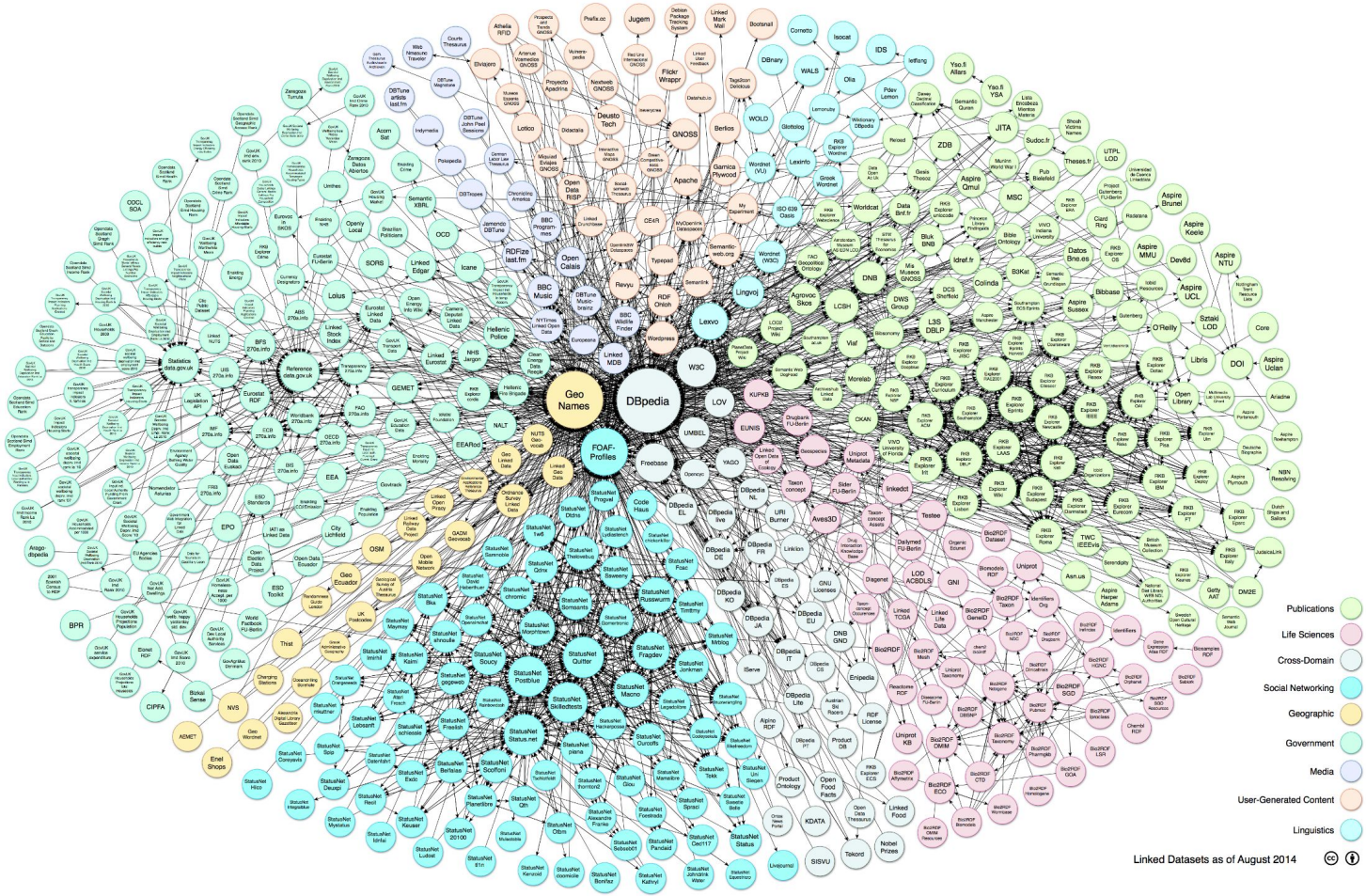
The LOD cloud - 2010 - 203 datasets



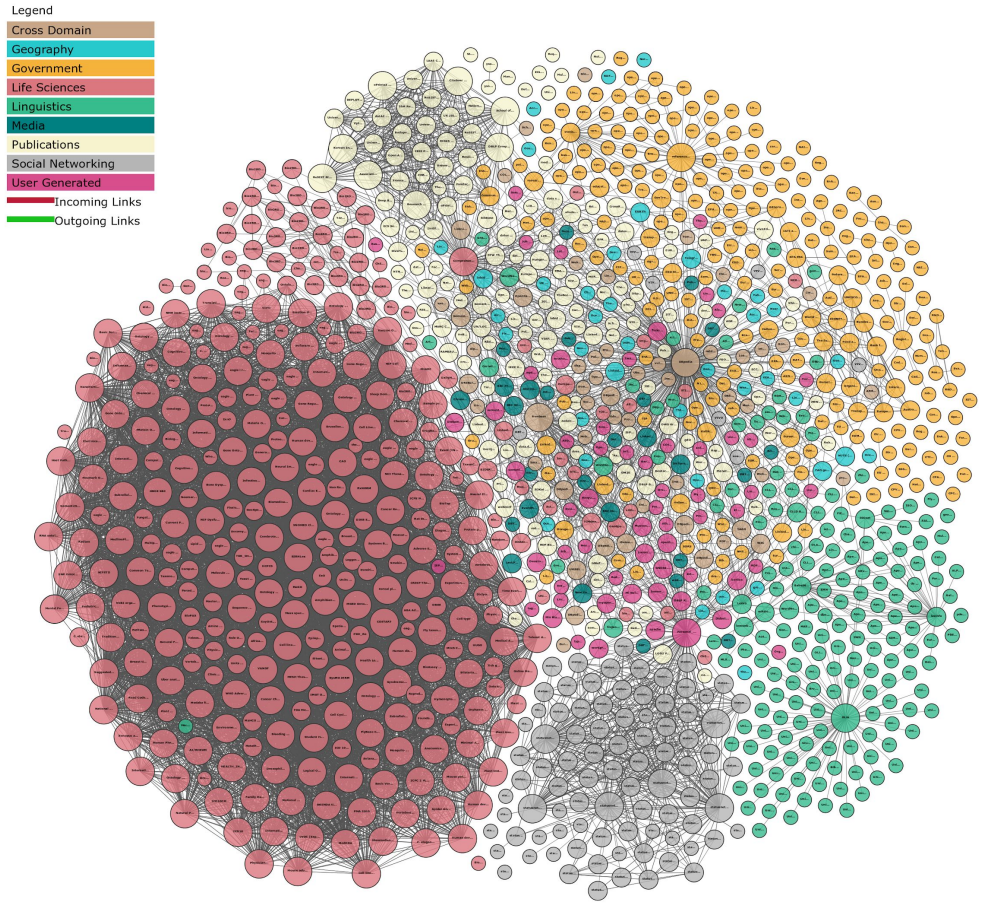
The LOD cloud - 2011 - 295 datasets



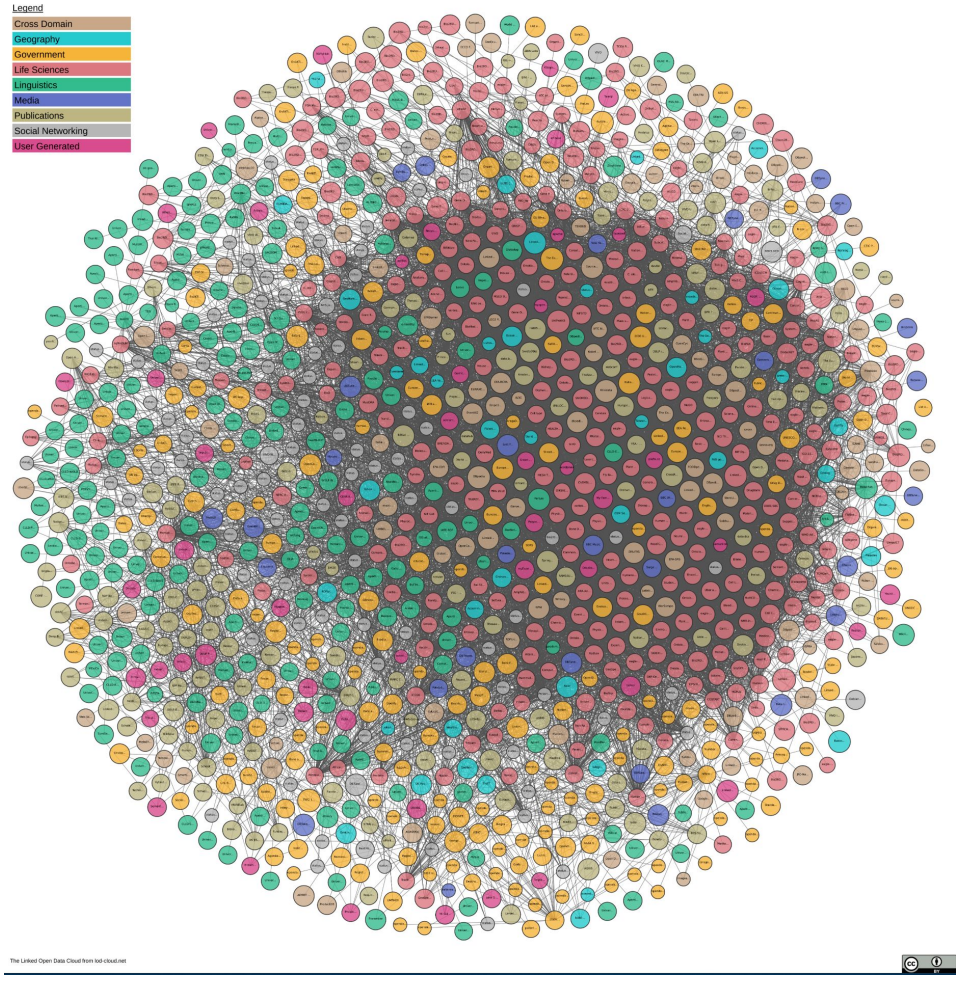
The LOD cloud - 2014 - 570 datasets



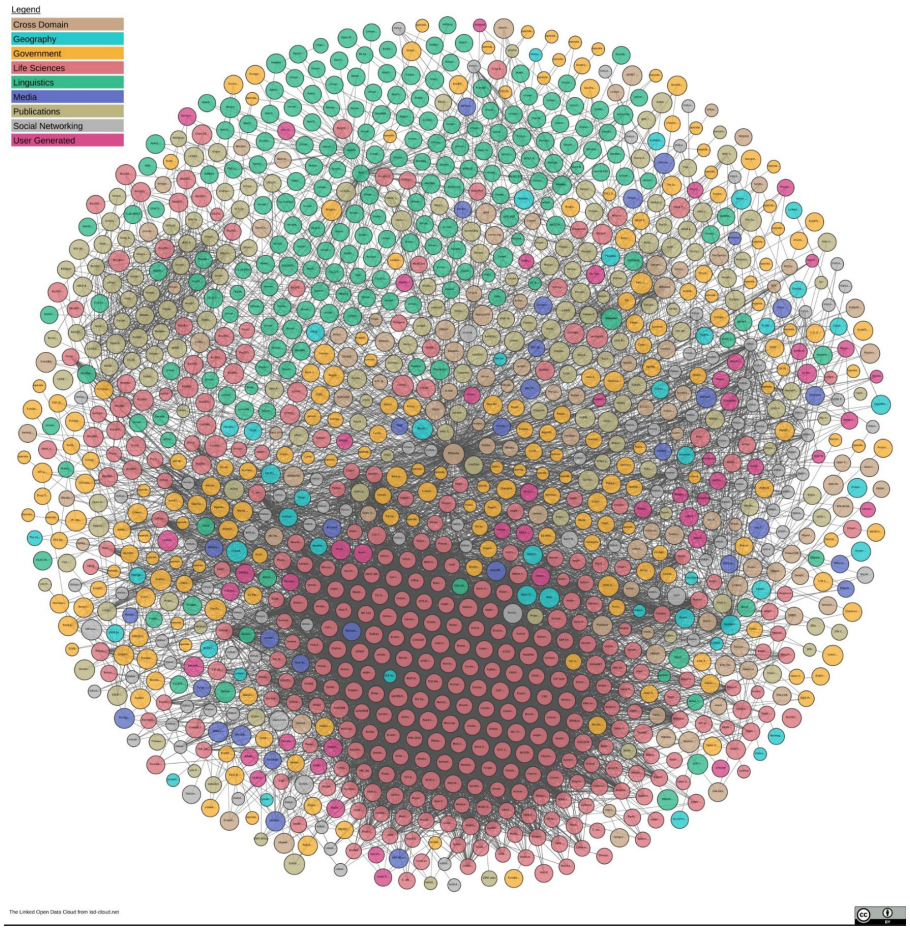
The LOD cloud - 2017 - 1146 datasets



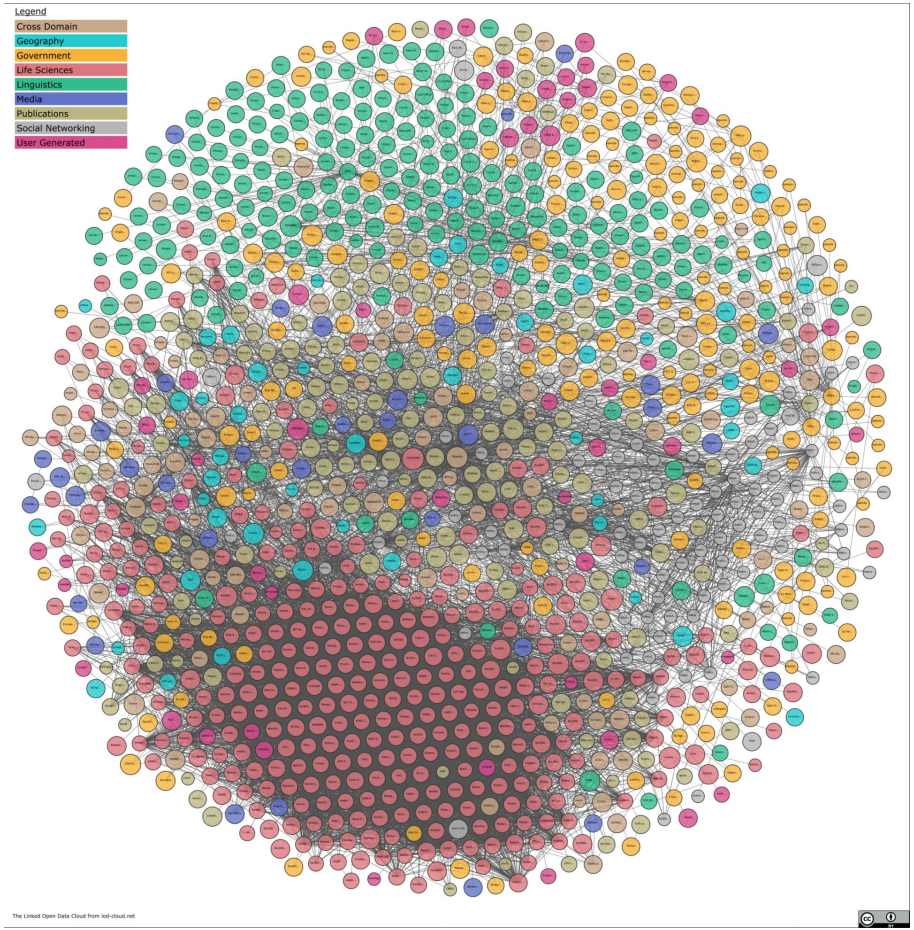
The LOD cloud - 2018 - 1229 datasets



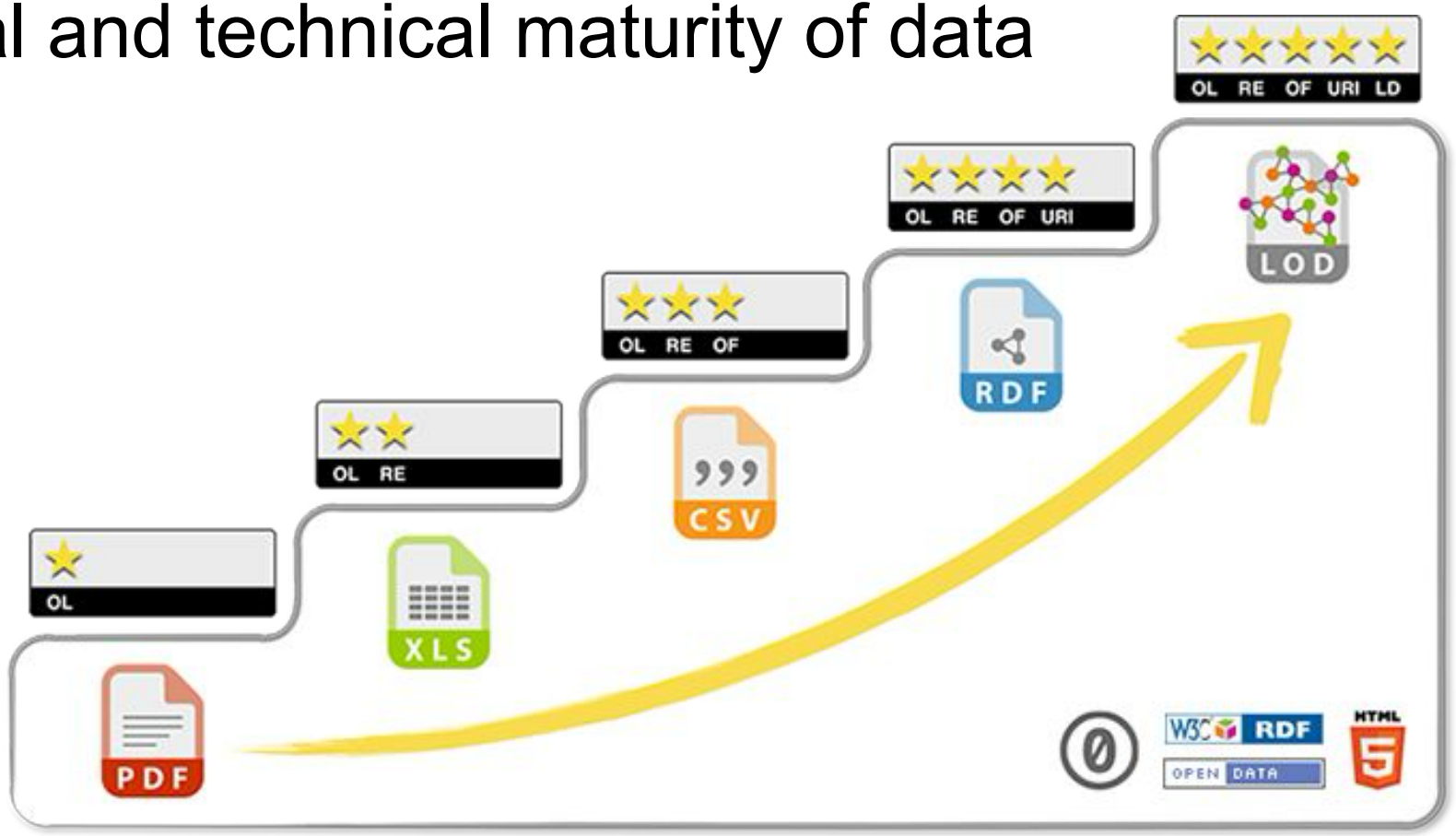
The LOD cloud - 2019 - 1239 datasets



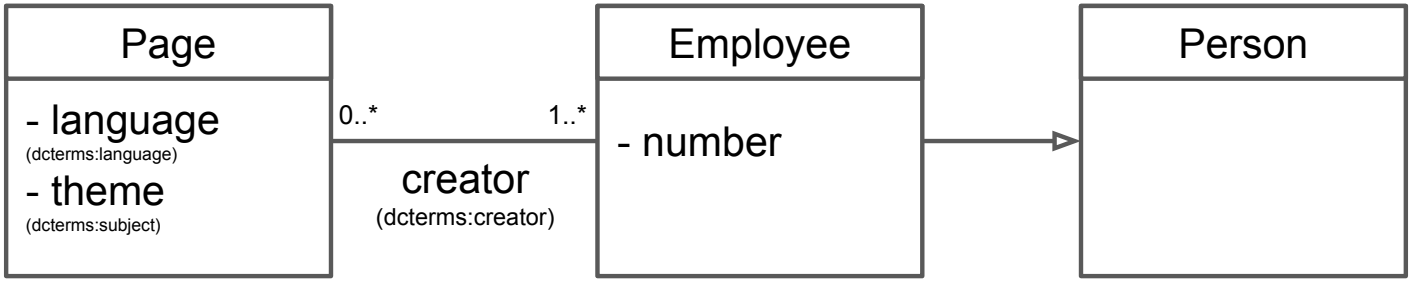
The LOD cloud - 2020 - 1255 datasets



Open data - 5 star classification
Legal and technical maturity of data



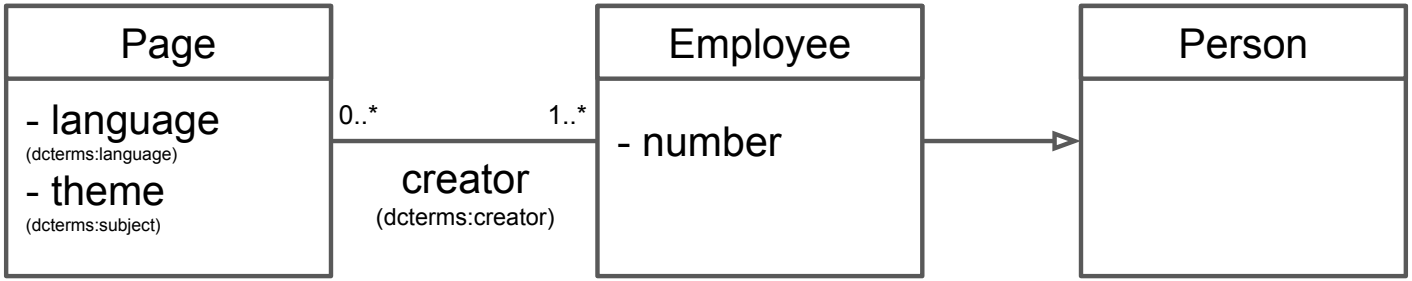
RDF - relation to conceptual model




```
ex:index.html a ex:Page .
ex:index.html dcterms:creator emp:85740 .
ex:index.html dcterms:subject "education" .
ex:index.html dcterms:language "en" .

emp:85740 a ex:Employee .
emp:85740 ex:number 42 .
```

```
ex:Page a rdfs:Class .
ex:Employee a rdfs:Class .
ex:Person a rdfs:Class .
ex:Employee rdfs:subClassOf ex:Person .

dcterms:creator a rdf:Property .
dcterms:subject a rdf:Property .
dcterms:language a rdf:Property .
ex:number a rdf:Property .
```

RDF - relation to conceptual model




```
ex:index.html a ex:Page .
ex:index.html dcterms:creator emp:85740 .
ex:index.html dcterms:subject "education" .
ex:index.html dcterms:language "en" .

emp:85740 a ex:Employee .
emp:85740 ex:number 42 .
```

```
ex:Page a rdfs:Class ;
        rdfs:label "Page"@en ;
        rdfs:comment "A web page"@en .

ex:number a rdf:Property ;
          rdfs:label "Employee number"@en ;
          rdfs:domain ex:Employee ;
          rdfs:range xsd:integer .
```