Semantic Web Technologies

Topic: Providing Access to RDF Data

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Overview

- Files and Dumps
- SPARQL Endpoints
- Linked Data
- Write Access

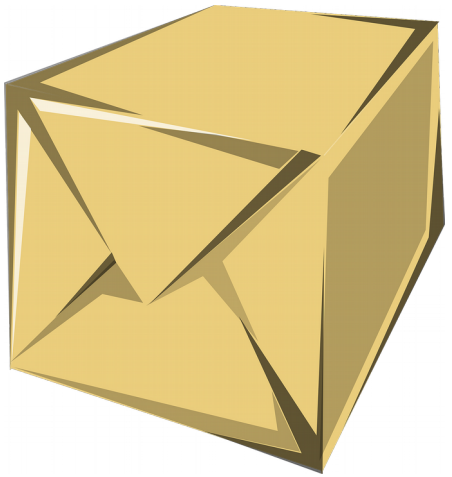
image source: http://maxpixel.freegreatpicture.com/Watch-Binoculars-To-Find-Overview-Search-See-1026425

Files and Dataset Dumps

The Most Simple Option

- Create files in any RDF serialization format and put them on your Web server

- File that contains a whole dataset is called *dump file*
  - Usually not plain RDF file but compressed (e.g., ZIP)



Picture sources: https://pixabay.com/en/package-box-delivery-container-24546/
https://pixabay.com/en/documents-paper-files-148079/

HDT

header dictionary triples

- Compression format for RDF that supports search operations without decompression

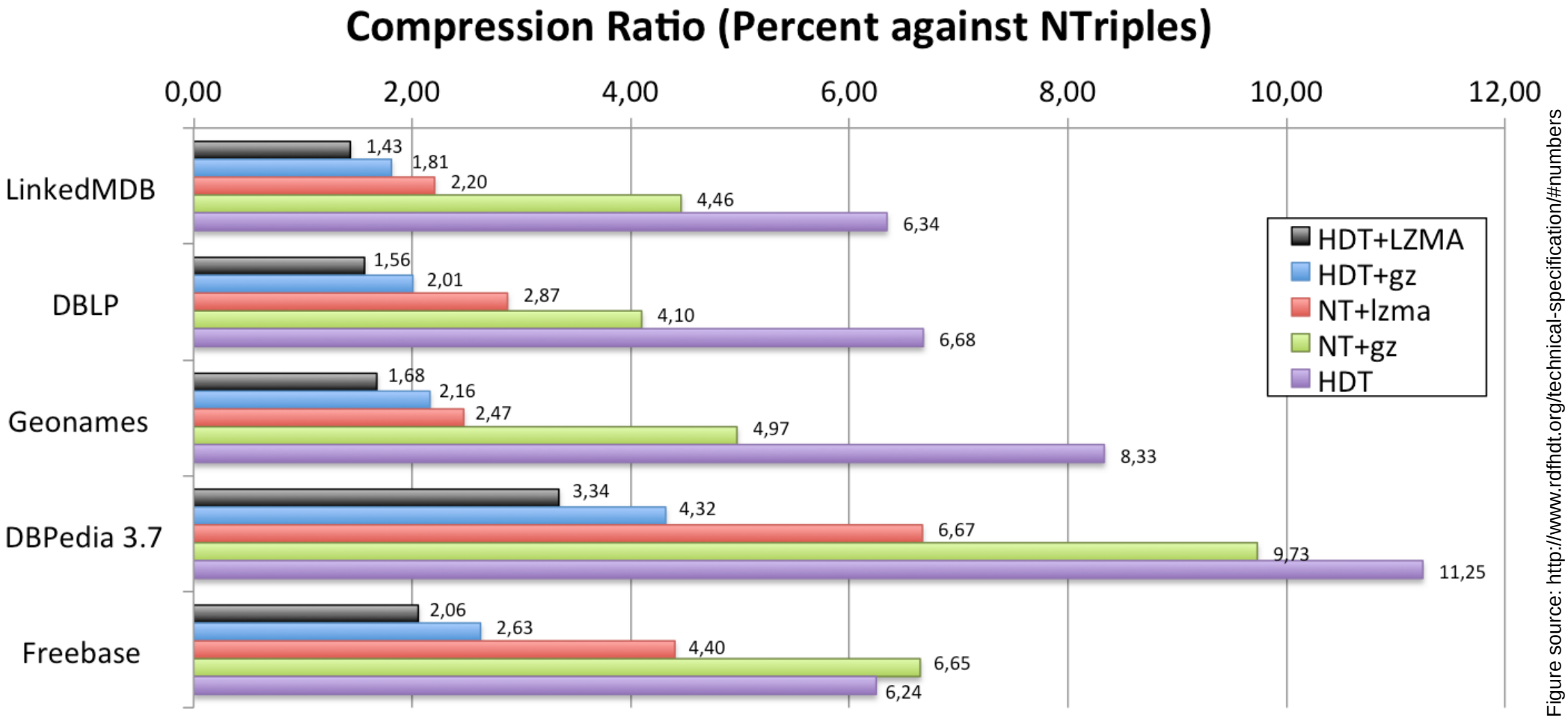


Figure source: http://www.rdfhdt.org/technical-specification/#numbers

HDT

header dictionary triples

- Compression format for RDF that supports search operations without decompression

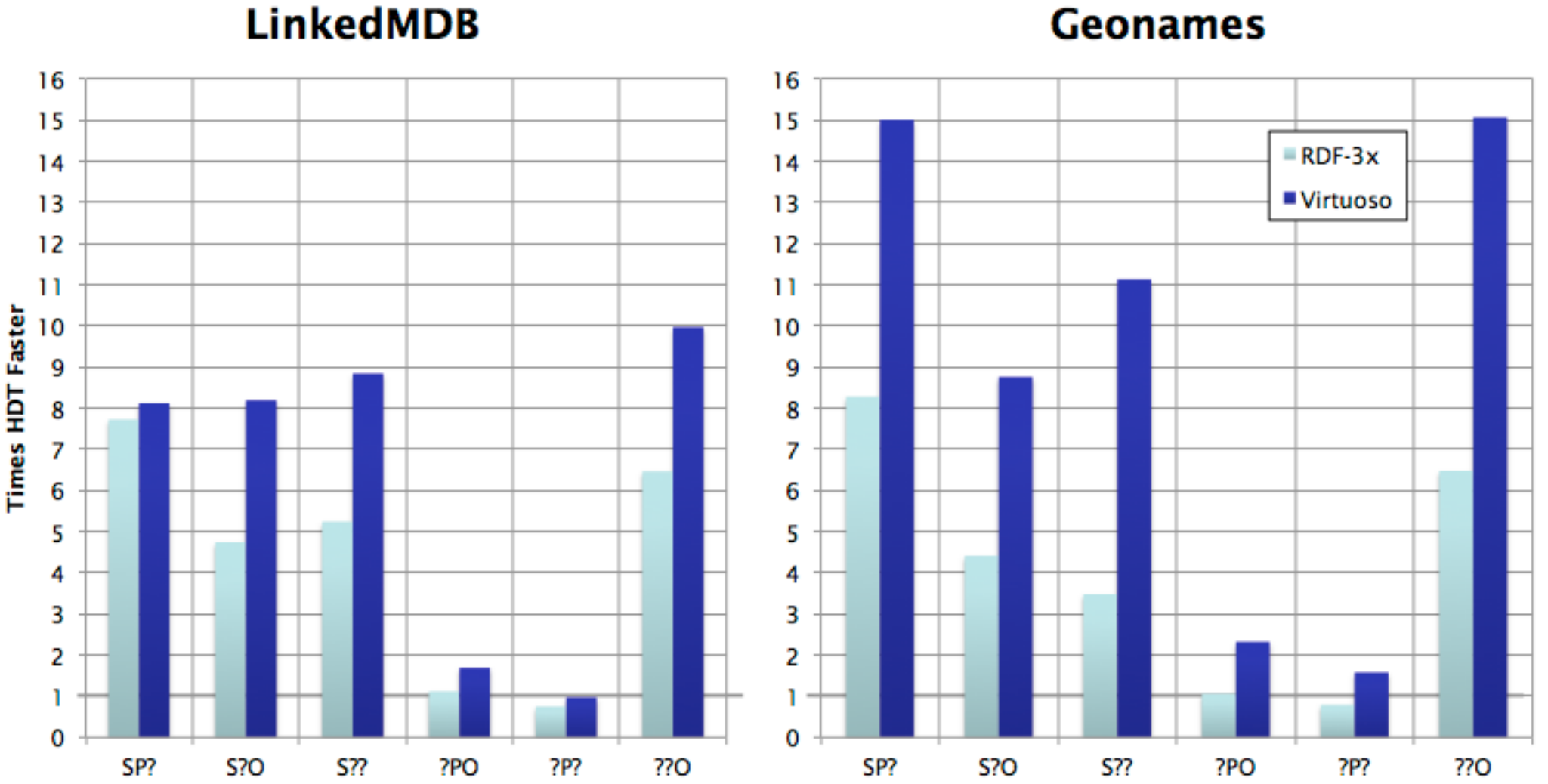
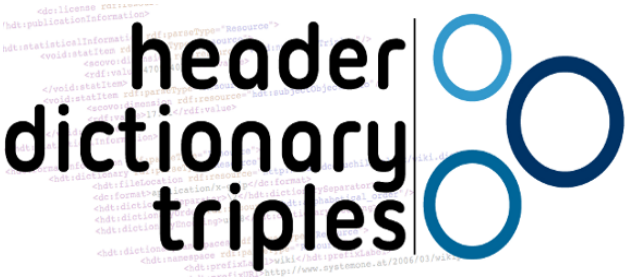


Figure source: http://www.rdfhdt.org/technical-specification/#numbers

HDT



- Compression format for RDF that supports search operations without decompression
  - Fast triple pattern queries (1.5x faster than Virtuoso)
  - Queries with joins at the same scale as triple stores
- Application scenarios
  - Publication of RDF datasets in a ready-to-use form
  - Consumption with limited resources (e.g., smartphones, standard laptops)
  - Scalable storage of large datasets
  - Fast, low-cost SPARQL engine
  - Lightweight back-end for query-based data access interfaces (e.g., SPARQL endpoints, TPF, brTPF)

Picture source: http://www.rdfhdt.org/wp-content/uploads/2013/01/hdt-front-web.png

SPARQL Endpoints

What is a SPARQL Endpoint?

- Web service that provides SPARQL-based access to a server-side RDF dataset
- Send your SPARQL query → receive the query result
- Supports the SPARQL protocol (which is part of the family of W3C recommendations for SPARQL)

Accessing a SPARQL Endpoint

- Issuing a query is as simple as sending an HTTP GET request with parameter **`query`** to the address of the endpoint
- Example (using the SPARQL endpoint of DBpedia whose address is **`http://dbpedia.org/sparql`**):

```
GET /sparql?query=PREFIX+rd... HTTP/1.1
Host: dbpedia.org
User-agent: my-sparql-client/0.1
```

URL-encoded string with the SPARQL query

Query Result Formats

- SPARQL usually support different result formats:
  - XML, JSON, CSV, plain text
    (for SELECT and ASK queries)
  - Turtle, RDF/XML, N-Triples
    (for CONSTRUCT and DESCRIBE queries)

Requesting a Specific Result Format

- Use the `ACCEPT` header in your GET request

```
GET /sparql?query=PREFIX+rd... HTTP/1.1
Host: dbpedia.org
User-agent: my-sparql-client/0.1
Accept: application/sparql-results+json
```

- Examples of other possible mime types:
  - `application/sparql-results+xml`
  - `text/csv`
  - `text/tab-separated-values`
  - `text/turtle`
  - `application/rdf+xml`

Requesting a Specific Result Format (cont'd)

- As another option, several SPARQL endpoint implementations support an additional parameter, typically called `out`

```
GET /sparql?out=json&query=PREFI... HTTP/1.1
Host: dbpedia.org
User-agent: my-sparql-client/0.1
```

Some Software Libraries for SPARQL Clients

- JavaScript
  - https://www.w3.org/community/rdfjs/wiki/Comparison_of_RDFJS_libraries
- PHP
  - sparqllib.php http://graphite.ecs.soton.ac.uk/sparqllib/
  - RAP http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/
- Java
  - Apache Jena https://jena.apache.org/
  - Eclipse RDF4J (formerly Sesame) http://rdf4j.org/
- Python
  - sparql-client https://pypi.python.org/pypi/sparql-client
  - sparqlwrapper https://rdflib.github.io/sparqlwrapper/

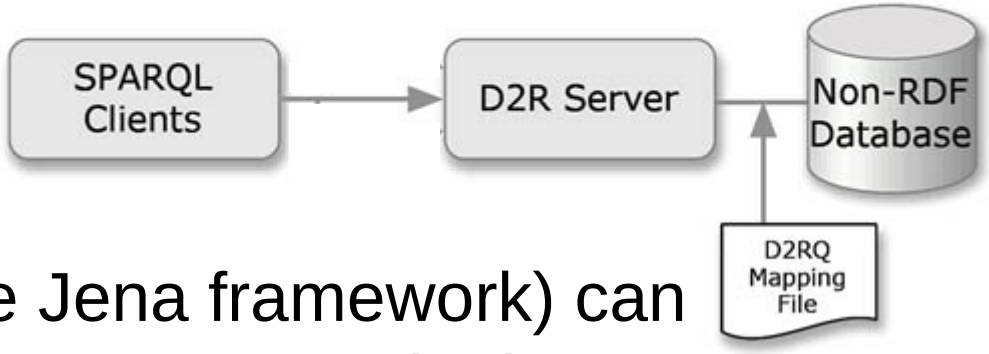
Apache Jena Example

```
import org.apache.jena.query.*;

String service = "..."; // address of the SPARQL endpoint
String query = "SELECT ..."; // your SPARQL query
QueryExecution e = QueryExecutionFactory.sparqlService( service,
                                                        query );
ResultSet results = e.execSelect();
while ( results.hasNext() ) {
    QuerySolution s = results.nextSolution();
    // …
}
e.close();
```

Providing a SPARQL Endpoint

- Many triple stores can provide a SPARQL endpoint to access their datasets (e.g., Blazegraph)
- RDB2RDF wrappers provide SPARQL access to a relational DB by rewriting SPARQL queries to SQL
  - D2R Server http://d2rq.org/d2r-server
  - Sparqlify http://aksw.org/Projects/Sparqlify.html



- Fuseki (part of the Apache Jena framework) can be used to easily set up a SPARQL endpoint over RDF data loaded from a file

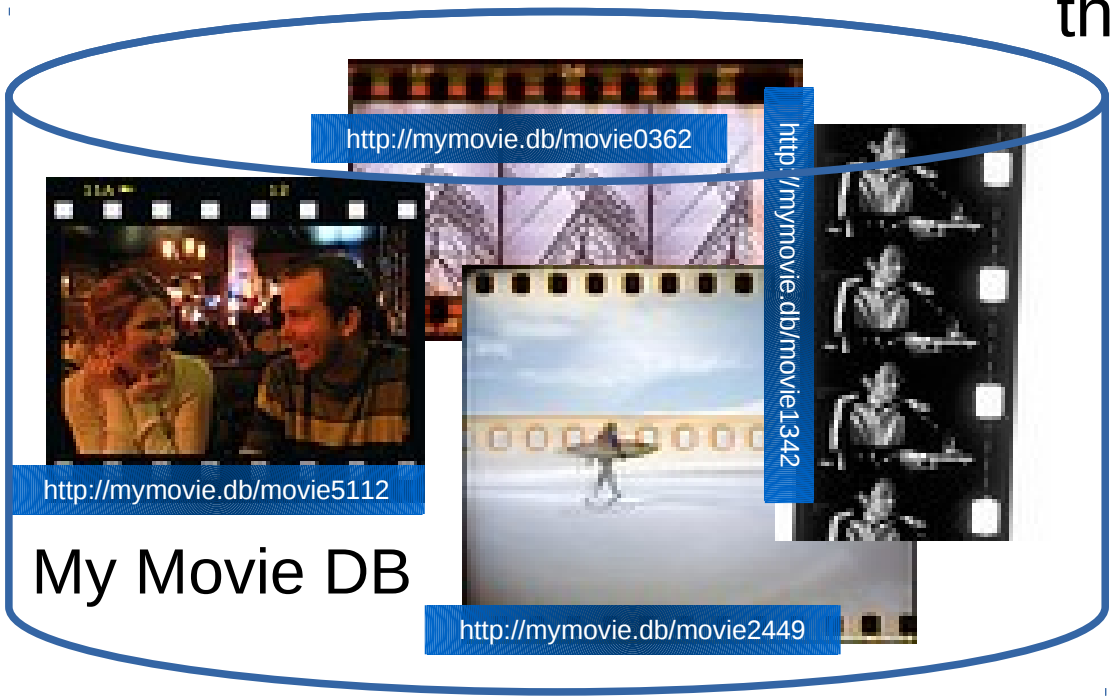
Picture source: http://linkeddatabook.com/editions/1.0/#htoc69

Linked Data

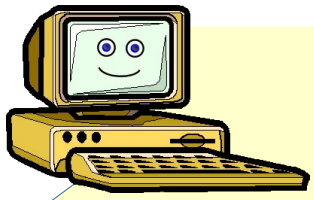
The Linked Data Principles

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

Tim Berners-Lee, July 2006



The Linked Data Principles

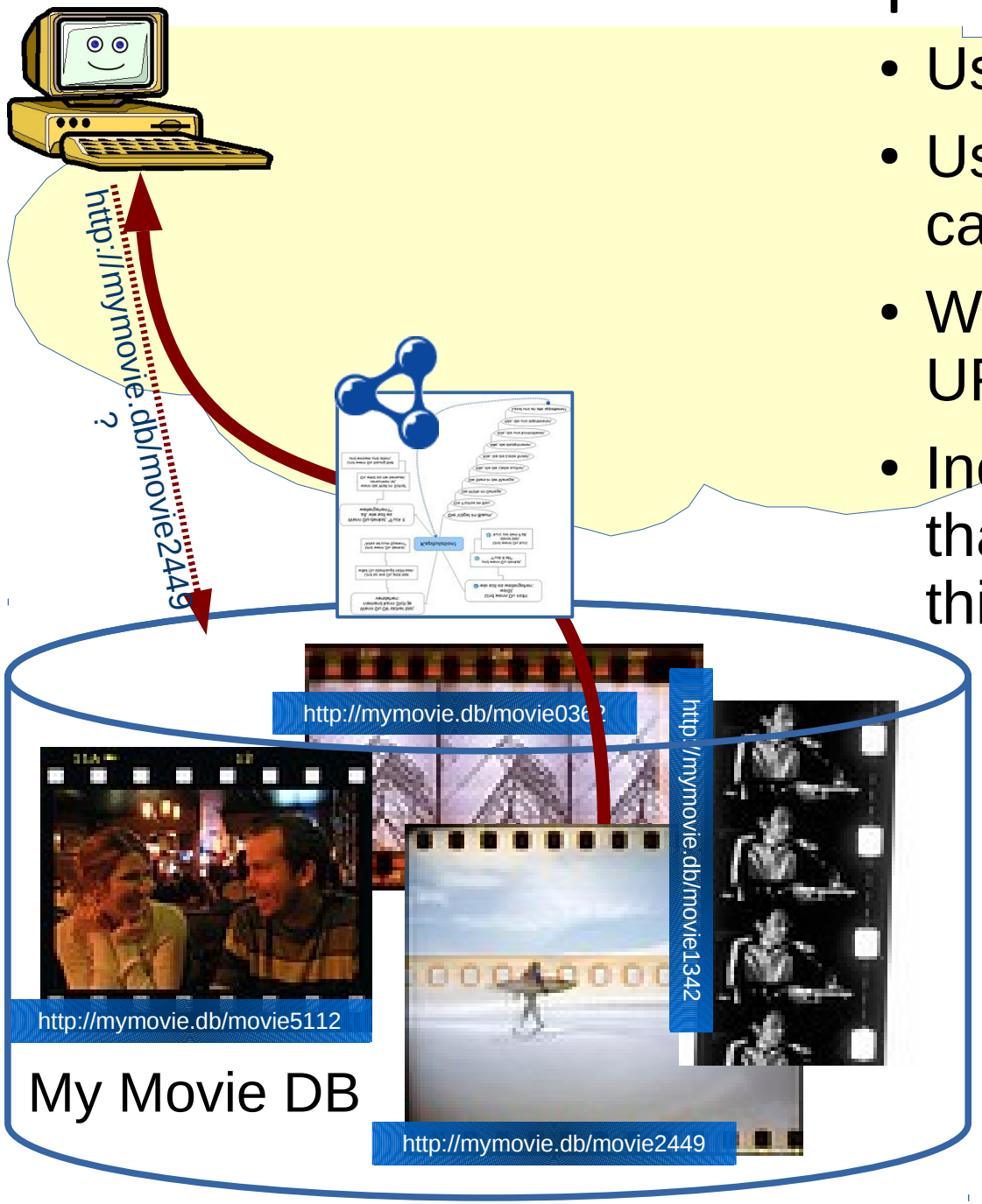


http://mymovie.db/movie2449
?

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

Tim Berners-Lee, July 2006

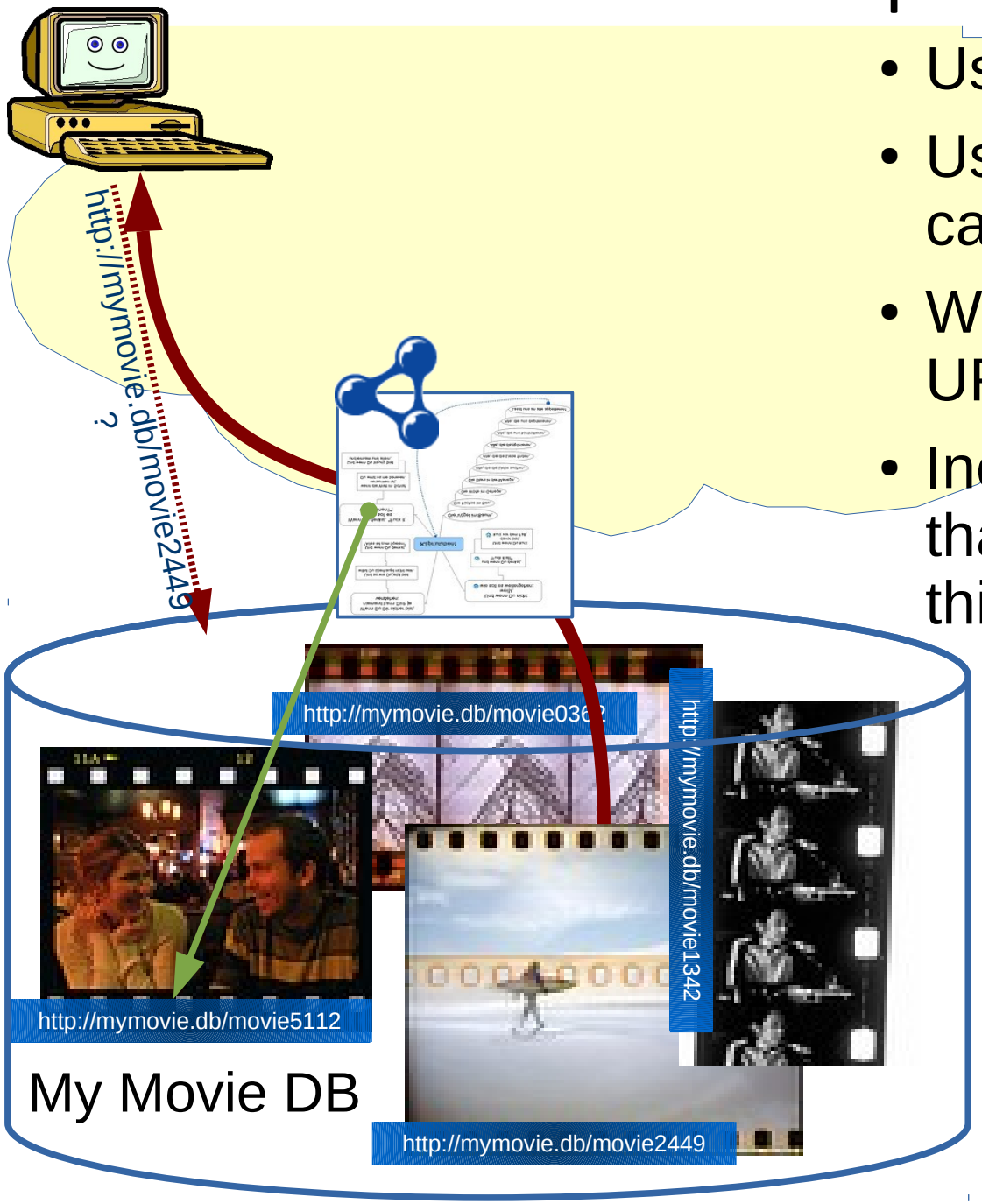
The Linked Data Principles



- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

Tim Berners-Lee, July 2006

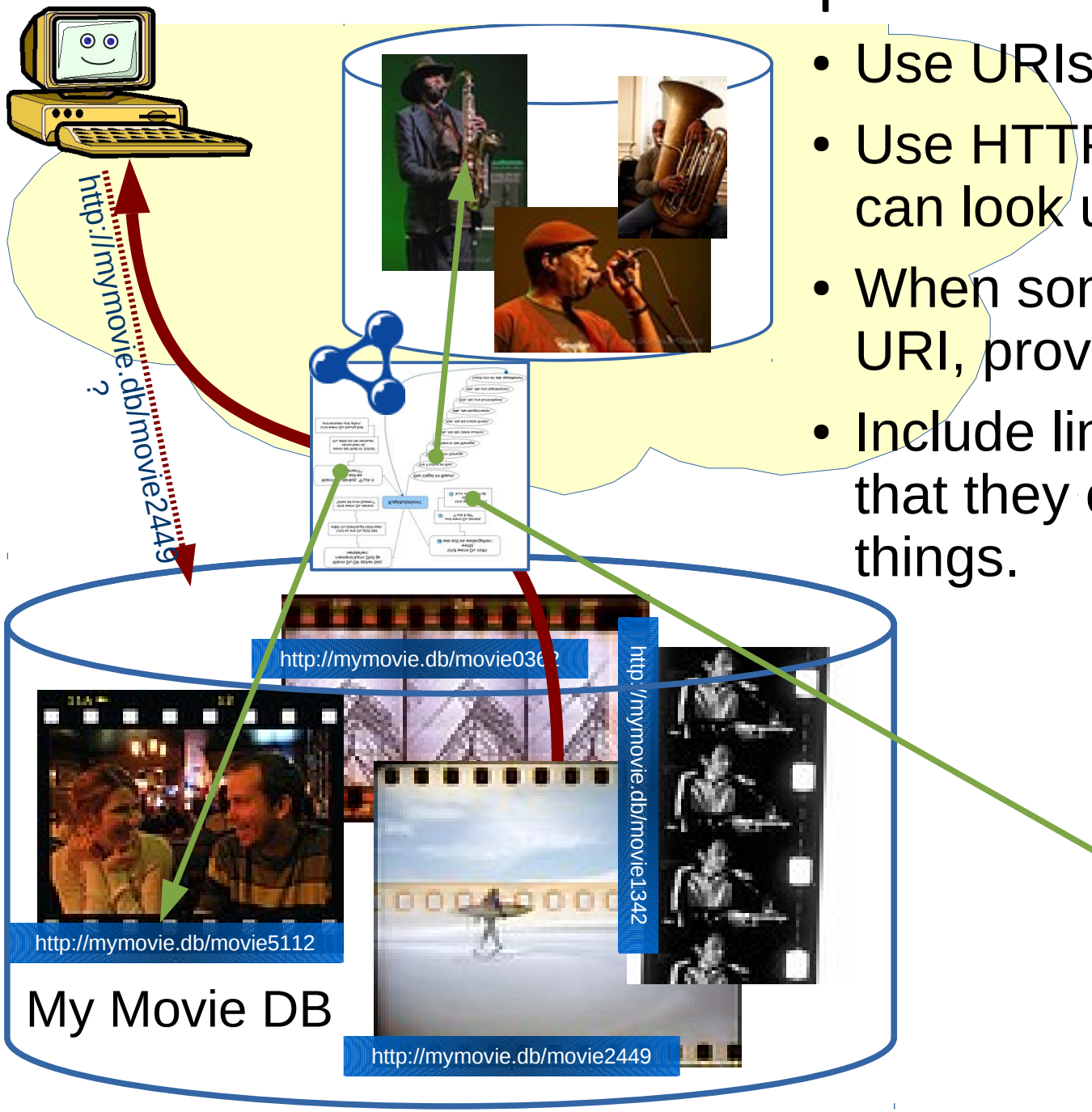
The Linked Data Principles



- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

Tim Berners-Lee, July 2006

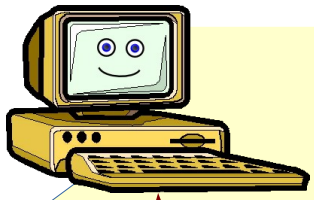
The Linked Data Principles



- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

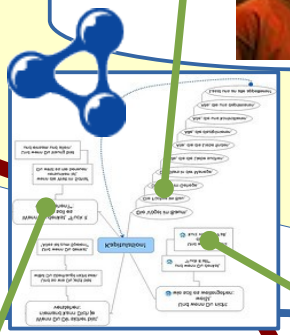
Tim Berners-Lee, July 2006

The Linked Data Principles



http://mymovie.db/movie2449

?



- Use URIs as names for things
- Use HTTP [illegible] people can look up
- When someone looks up a URI, provide useful information.
- Include links to other URIs so that they can discover more things.

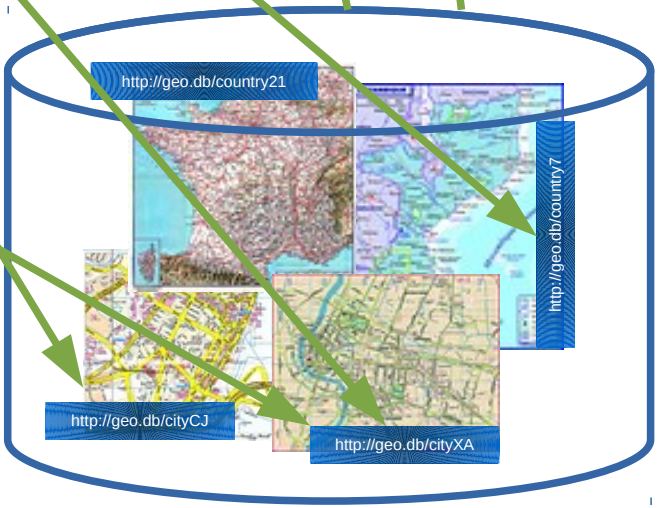
Tim Berners-Lee, July 2006

http://mymovie.db/movie5112

http://mymovie.db/movie1342

http://mymovie.db/movie2449

My Movie DB



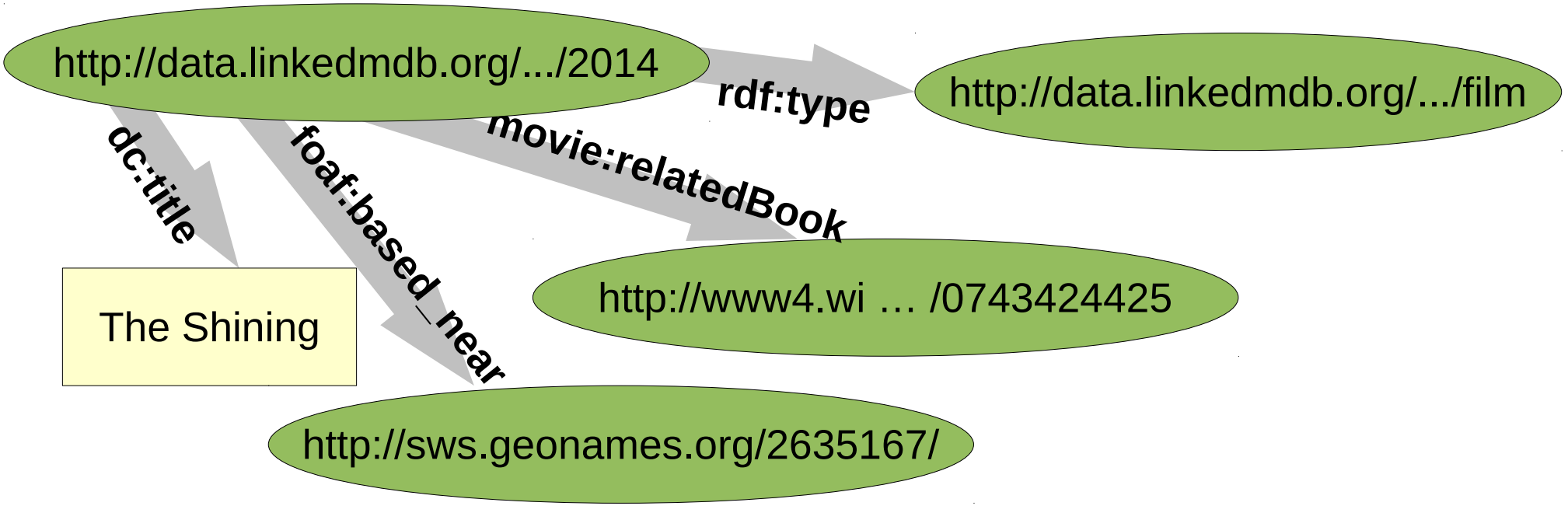
http://geo.db/country21

http://geo.db/country7

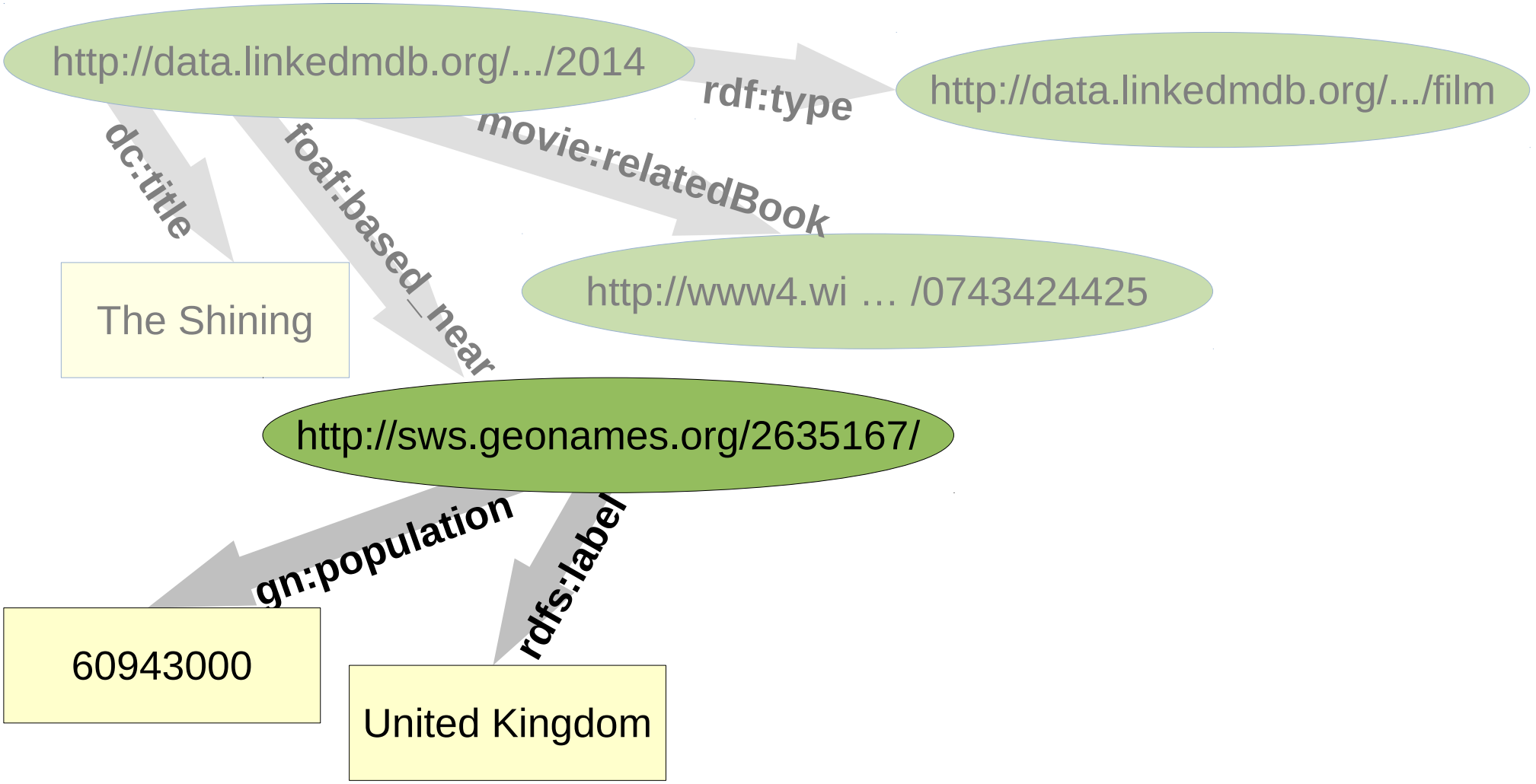
http://geo.db/cityCJ

http://geo.db/cityXA

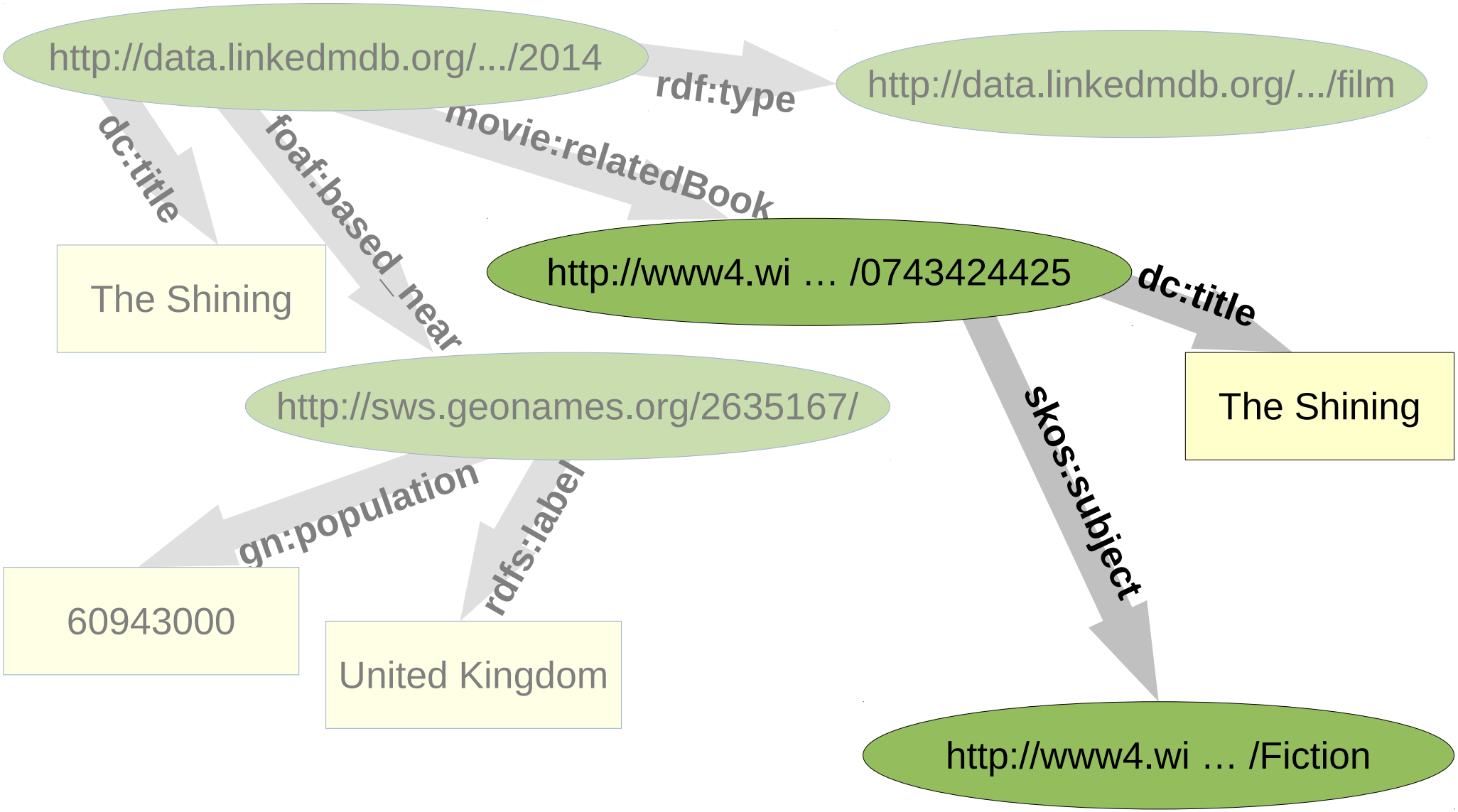
Example


http://data.linkedmdb.org/.../2014
rdf:type
http://data.linkedmdb.org/.../film
dc:title
foaf:based_near
movie:relatedBook
The Shining
http://www4.wi … /0743424425
http://sws.geonames.org/2635167/

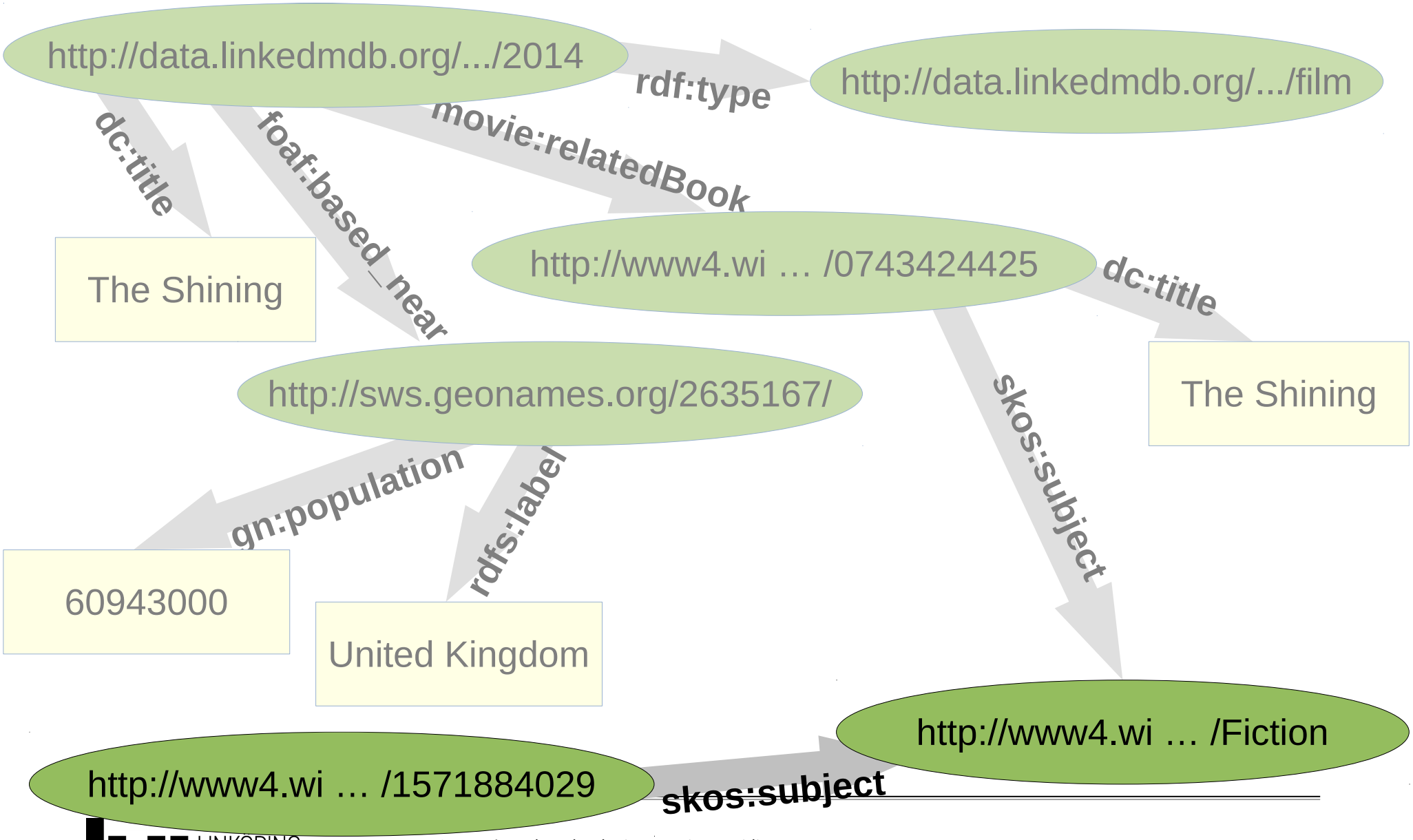
Example


http://data.linkedmdb.org/.../2014
rdf:type
http://data.linkedmdb.org/.../film
dc:title
foaf:based_near
movie:relatedBook
The Shining
http://www4.wi … /0743424425
http://sws.geonames.org/2635167/
gn:population
rdfs:label
60943000
United Kingdom

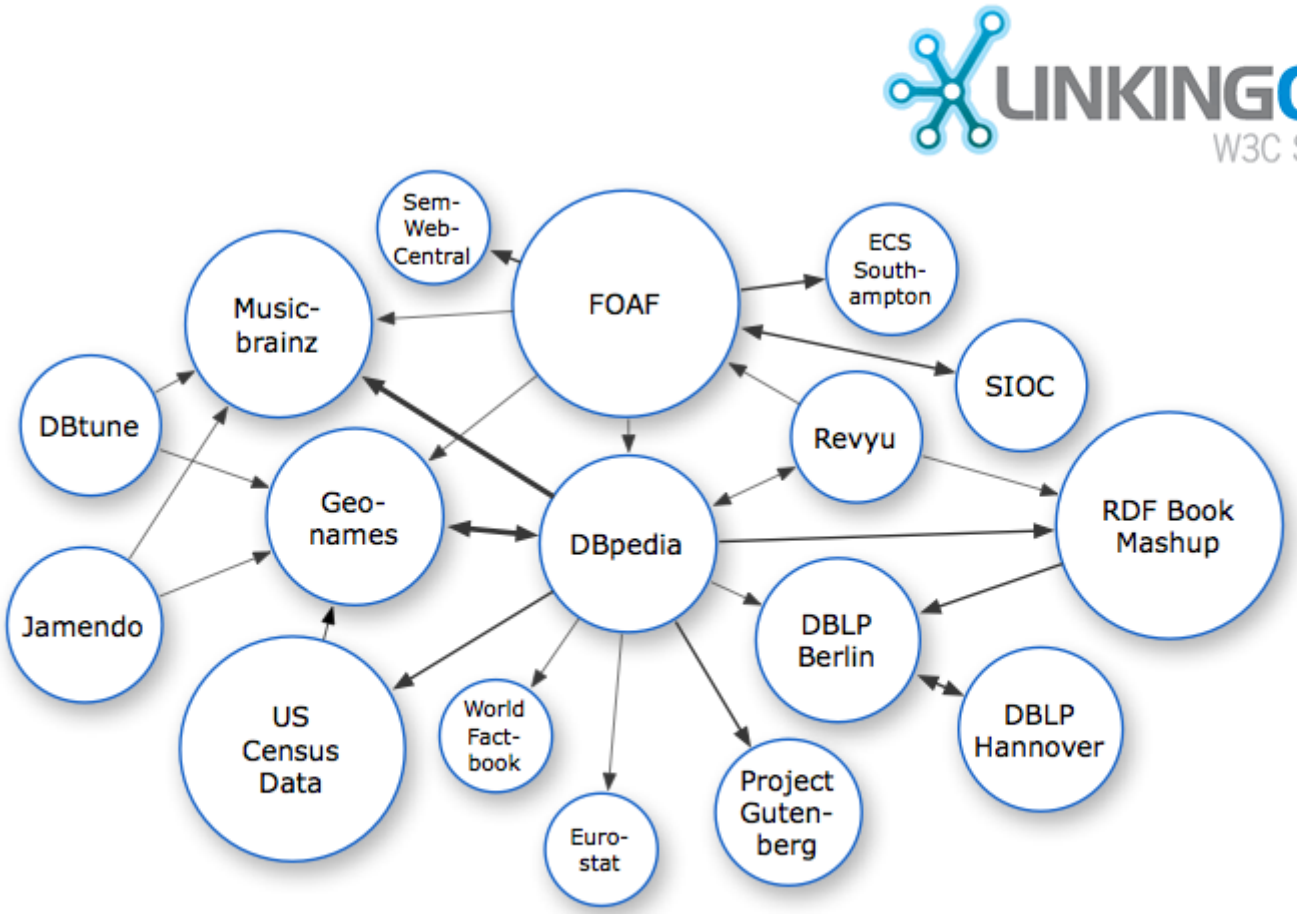
Example


http://data.linkedmdb.org/.../2014
rdf:type
http://data.linkedmdb.org/.../film
dc:title
foaf:based_near
movie:relatedBook
The Shining
http://www4.wi … /0743424425
dc:title
The Shining
http://sws.geonames.org/2635167/
skos:subject
gn:population
rdfs:label
60943000
United Kingdom
http://www4.wi … /Fiction

Example

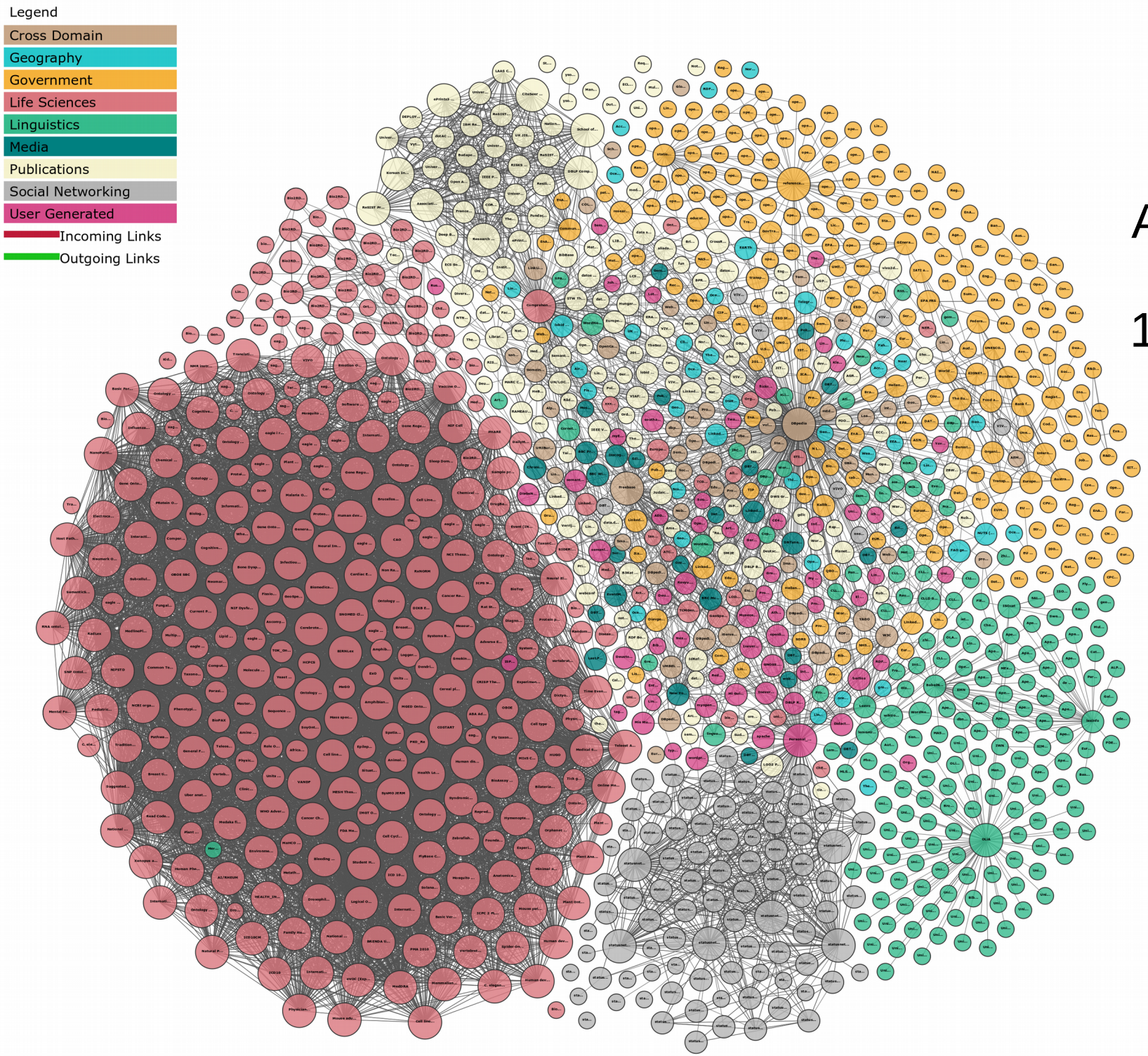

http://data.linkedmdb.org/.../2014
rdf:type
http://data.linkedmdb.org/.../film
dc:title
foaf:based_near
movie:relatedBook
The Shining
http://www4.wi … /0743424425
dc:title
The Shining
http://sws.geonames.org/2635167/
skos:subject
gn:population
rdfs:label
60943000
United Kingdom
http://www4.wi … /Fiction
http://www4.wi … /1571884029
skos:subject

Initial LOD Cloud



> 500M triples ca. 120,000 links

Picture source: http://lod-cloud.net/



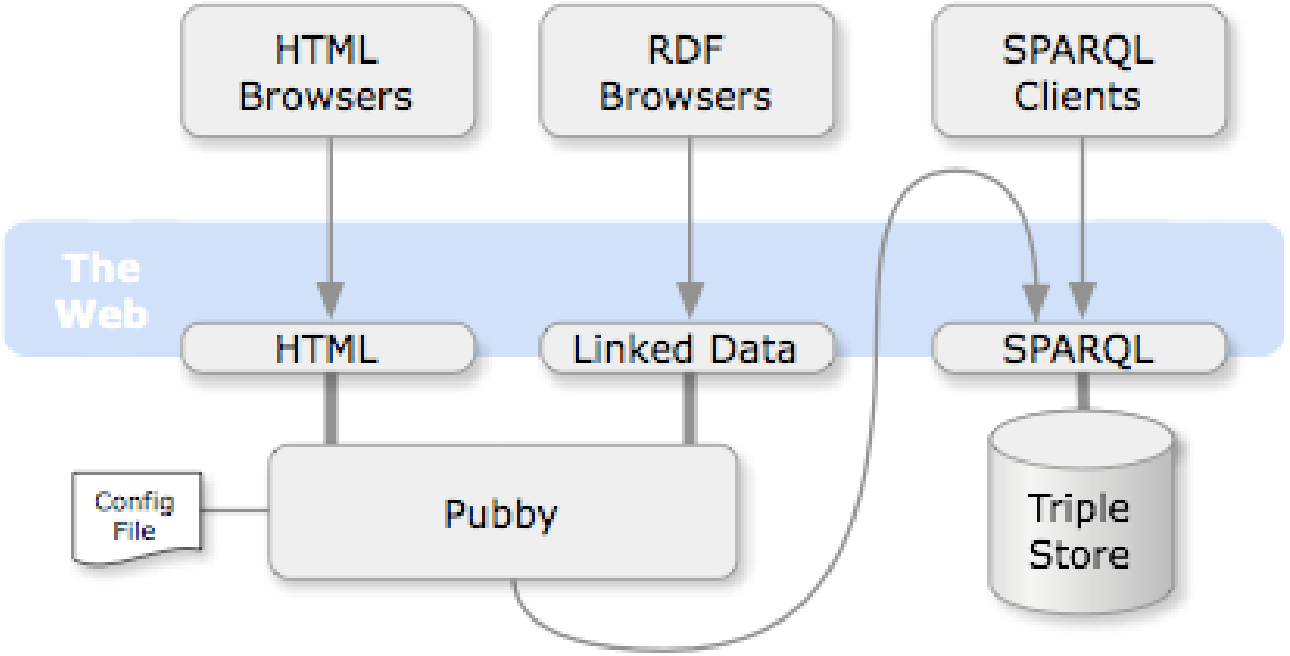
Aug. 22, 2017

1163 datasets

Picture source: http://lod-cloud.net/

Providing a Linked Data Interface

- Some triple stores provide a Linked Data interface to access their datasets (e.g., Virtuoso)
- Implementations on top of SPARQL endpoints
  - Pubby http://wifo5-03.informatik.uni-mannheim.de/pubby/
  - Elda http://epimorphics.github.io/elda/

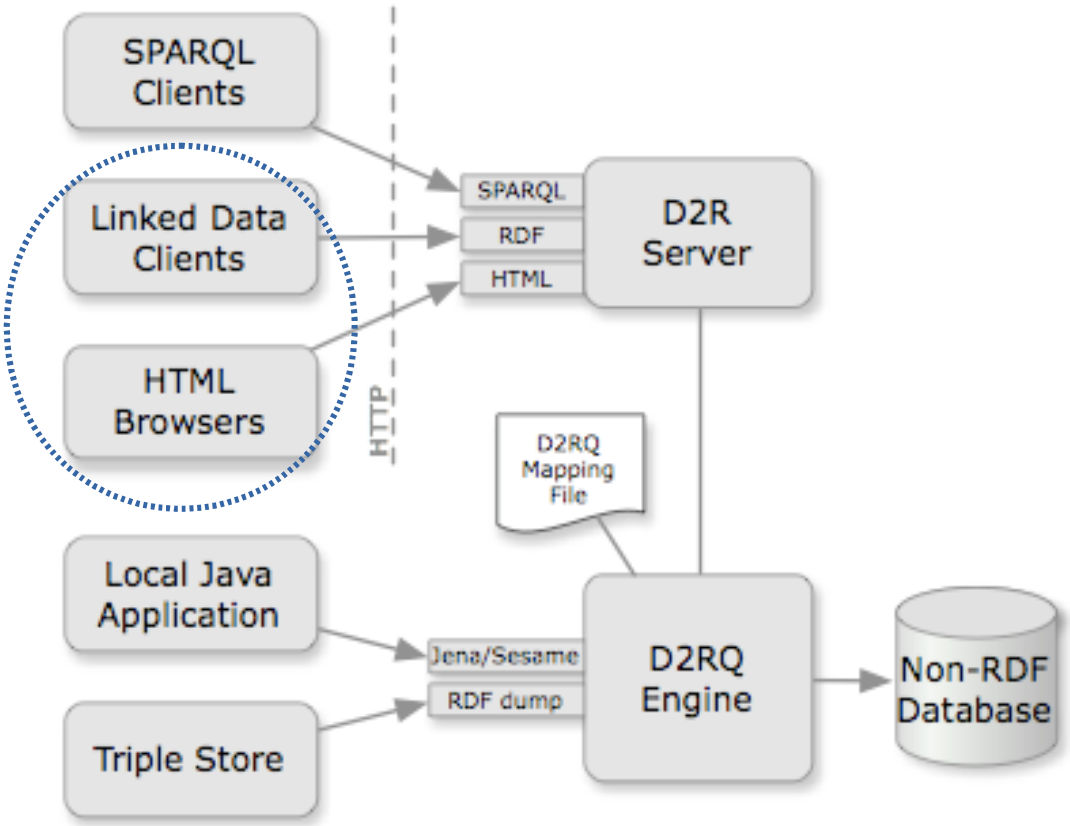


Picture source: http://wifo5-03.informatik.uni-mannheim.de/pubby/

Providing a Linked Data Interface

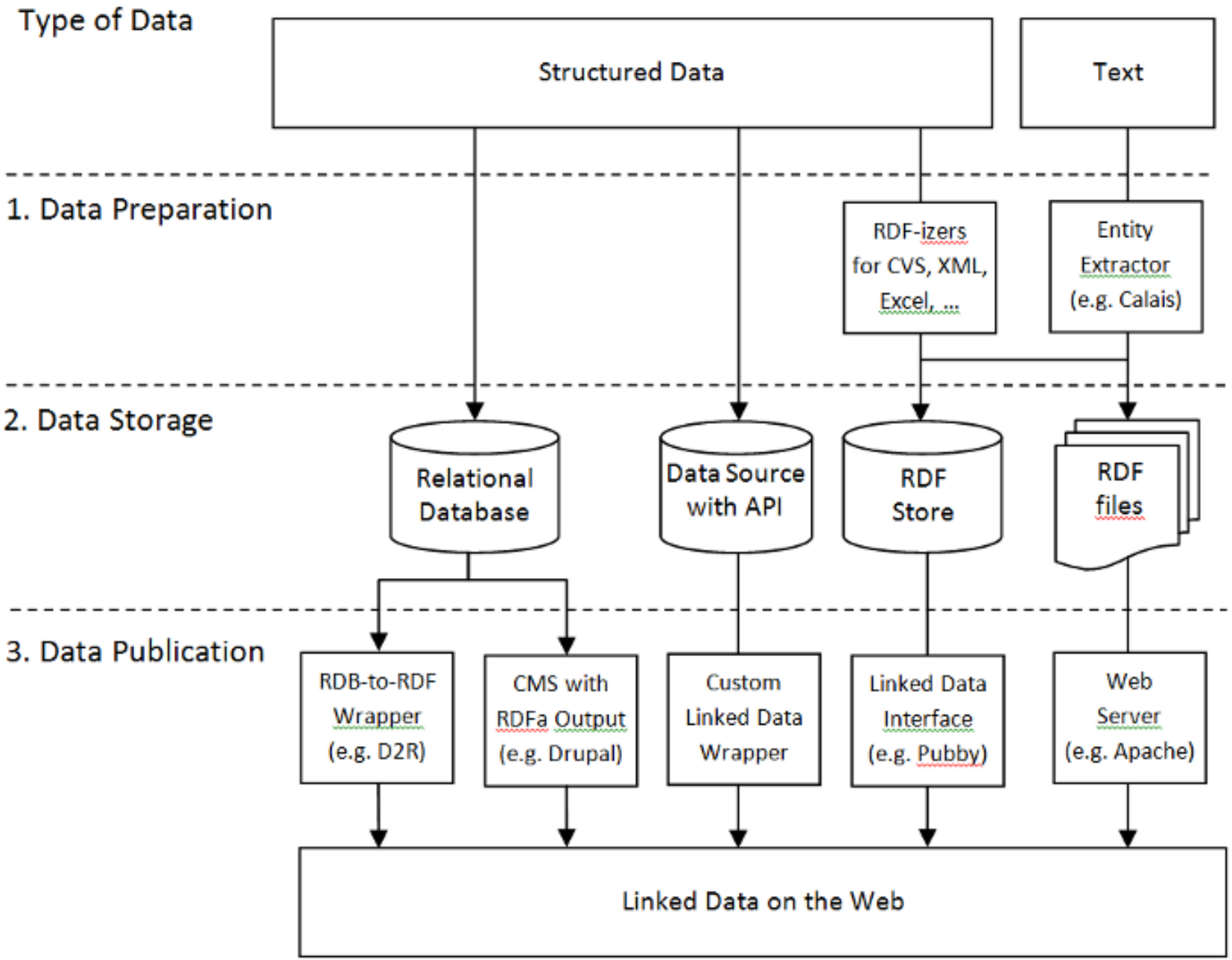
- Some triple stores provide a Linked Data interface to access their datasets (e.g., Virtuoso)
- Implementations on top of SPARQL endpoints
  - Pubby
  - Elda
- RDB2RDF wrappers
  - D2R Server

    http://d2rq.org/d2r-server



Picture source: http://d2rq.org/

Providing a Linked Data Interface



Picture source: http://linkeddatabook.com/editions/1.0/#htoc62

Linked Data URI Look-Ups in Detail

- We distinguish two types of URIs
  - "Slash URIs" e.g., http://data.linkedmdb.org/resource/film/2014
  - "Hash URIs" e.g., http://olafhartig.de/foaf.rdf#olaf
- In both cases, URI look-ups are HTTP GET requests

```
GET /resource/film/2014 HTTP/1.1
Host: data.linkedmdb.org
User-agent: curl/7.19.6 (i686-pc-linux-gnu)
```

  - For hash URIs, the URI without the hash part is requested and the response has to be searched for the full URI

```
GET /foaf.rdf HTTP/1.1
Host: olafhartig.de
User-agent: curl/7.19.6 (i686-pc-linux-gnu)
```

Linked Data URI Look-Ups in Detail (cont'd)

- Request a specific format by using HTTP content negotiation; i.e., `Accept` field in request header specifies the desired media type

```
GET /resource/film/2014 HTTP/1.1
Host: data.linkedmdb.org
User-agent: curl/7.19.6 (i686-pc-linux-gnu)
Accept: application/rdf+xml
```

- Some possible media types:
  - `application/rdf+xml`
  - `application/ld+json`
  - `application/n-triples`
  - `text/turtle`
  - `text/n3`
  - `text/html`

Linked Data URI Look-Ups in Detail (cont'd)

- "303 Redirection": depending on the requested media type, the client may be redirected to different resources

```
GET /resource/film/2014 HTTP/1.1
Host: data.linkedmdb.org
User-agent: curl/7.19.6 (i686-pc-linux-gnu)
Accept: application/rdf+xml
```

  - Possible response:

```
HTTP/1.1 303 See Other
Date: Thu, 11 Mar 2010 08:15:50 GMT
Server: Jetty(6.1.4)
Location: http://data.linkedmdb.org/data/film/2014
Content-Length: 0
Via: 1.1 data.linkedmdb.org
Content-Type: text/plain
```

Linked Data URI Look-Ups in Detail (cont'd)

- "303 Redirection": depending on the requested media type, the client may be redirected to different resources

```
GET /resource/film/2014 HTTP/1.1
Host: data.linkedmdb.org
User-agent: curl/7.19.6 (i686-pc-linux-gnu)
Accept: text/html
```

  - Possible response:

```
HTTP/1.1 303 See Other
Date: Thu, 11 Mar 2010 08:15:50 GMT
Server: Jetty(6.1.4)
Location: http://data.linkedmdb.org/page/film/2014
Content-Length: 0
Via: 1.1 data.linkedmdb.org
Content-Type: text/plain
```

Write Access

W3C Recommendations

- SPARQL Update
  - Graph management operations (CREATE, DROP, COPY, MOVE, ADD) and graph update operations (INSERT DATA, DELETE DATA, LOAD, CLEAR) for SPARQL endpoints
  - See: http://www.w3.org/TR/sparql11-update/
- SPARQL Graph Store HTTP Protocol
  - How to use HTTP operations (GET, PUT, POST, DELETE, HEAD, PATCH) for managing a collection of RDF graphs
  - See: http://www.w3.org/TR/sparql11-http-rdf-update/
- Linked Data Platform (LDP)
  - Introduces notions of LDP resource and LDP container
  - Defines rules for HTTP operations to access, create, update, and delete such resources and containers
  - See: http://www.w3.org/2012/ldp

www.liu.se

LINKÖPING UNIVERSITY