



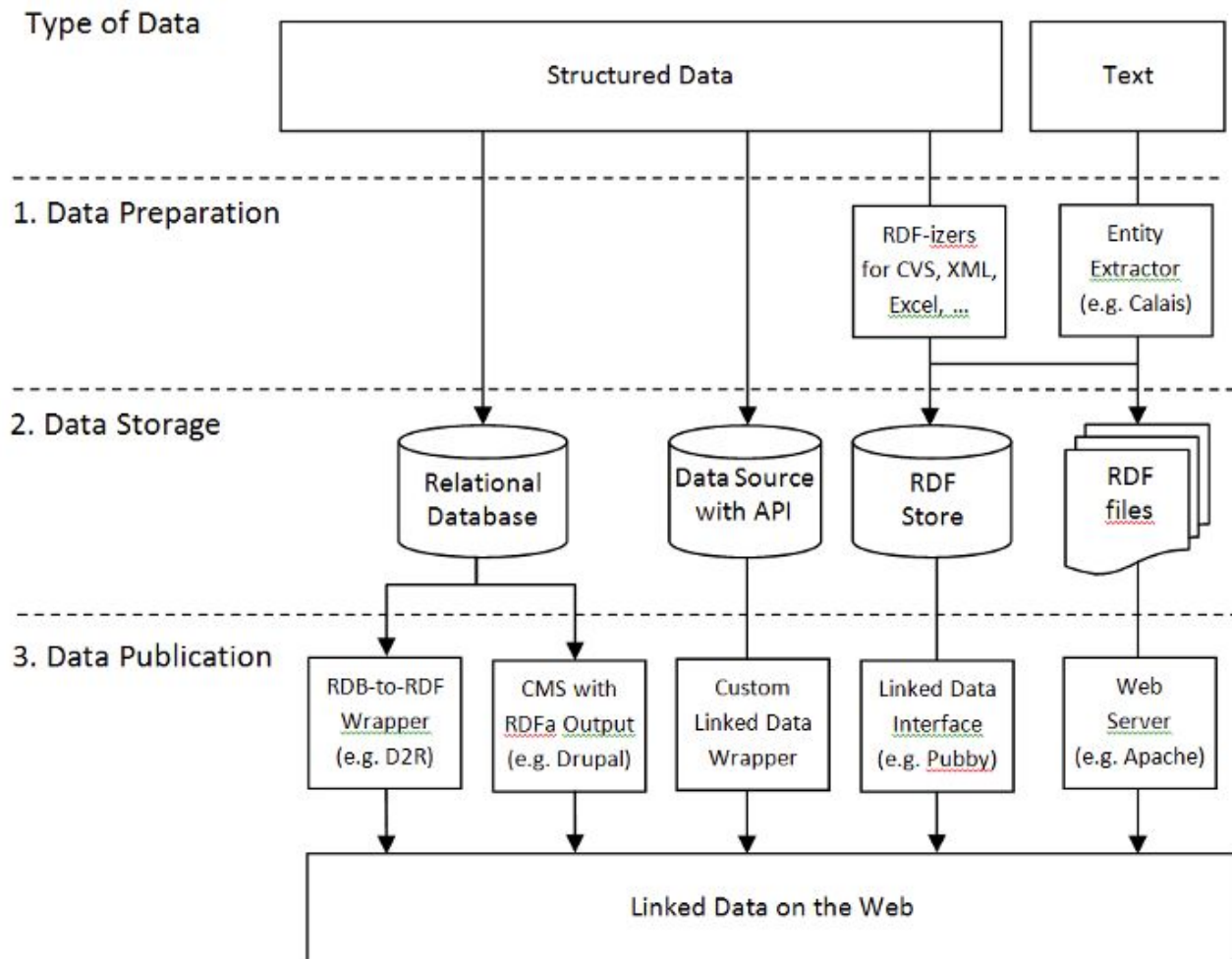
Setup of a Linked Data Node

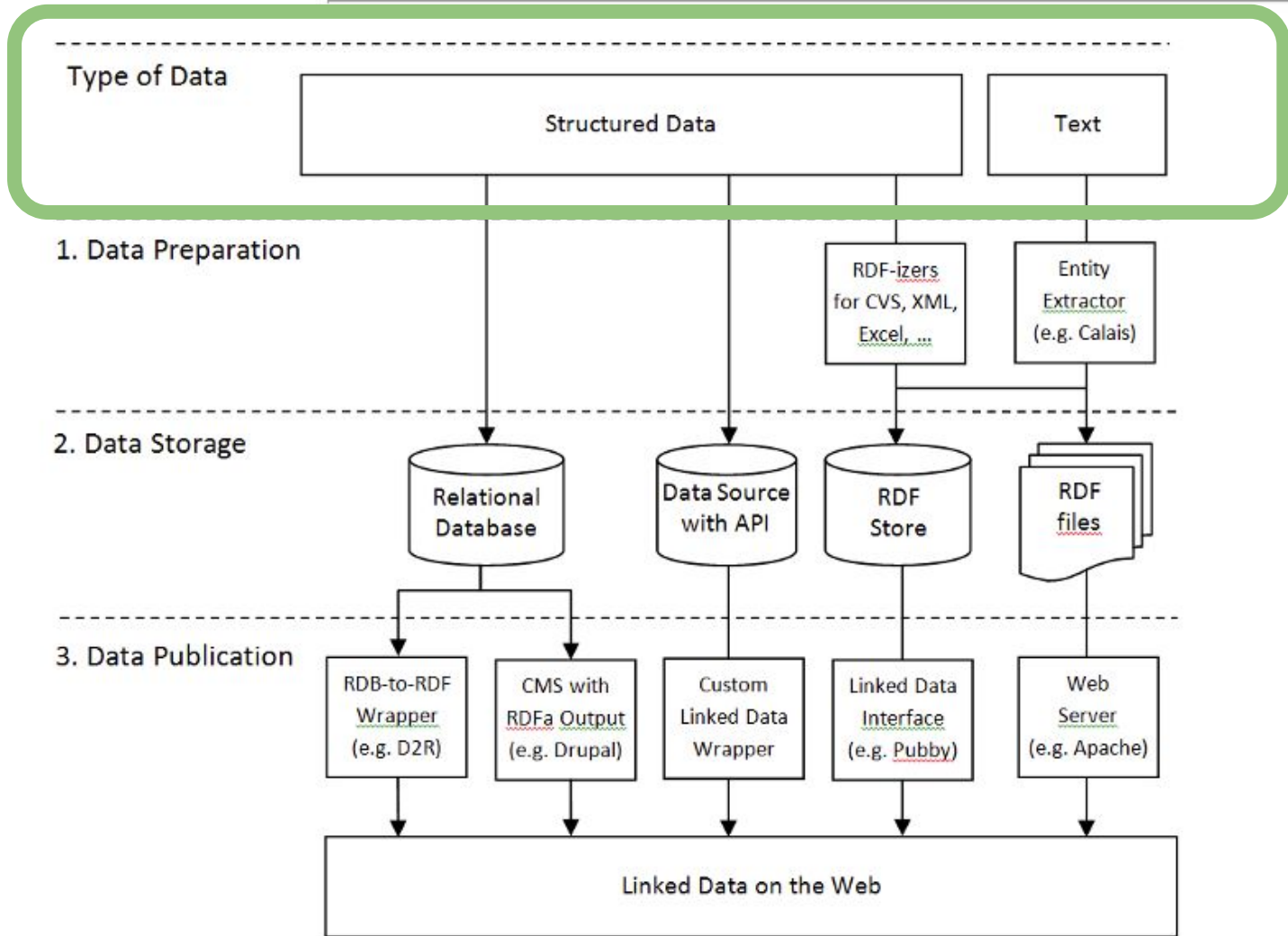
Angelica Lo Duca

IIT-CNR

angelica.loduca@iit.cnr.it

**Linked Open Data:
a paradigm for the
Semantic Web**





Type of Data

Structured Data

Text

1. Data Preparation

RDF-izers
for CSV, XML,
Excel, ...

Entity
Extractor
(e.g. Calais)

2. Data Storage

Relational
Database

Data Source
with API

RDF
Store

RDF
files

3. Data Publication

RDB-to-RDF
Wrapper
(e.g. D2R)

CMS with
RDFa Output
(e.g. Drupal)

Custom
Linked Data
Wrapper

Linked Data
Interface
(e.g. Pubby)

Web
Server
(e.g. Apache)

Linked Data on the Web

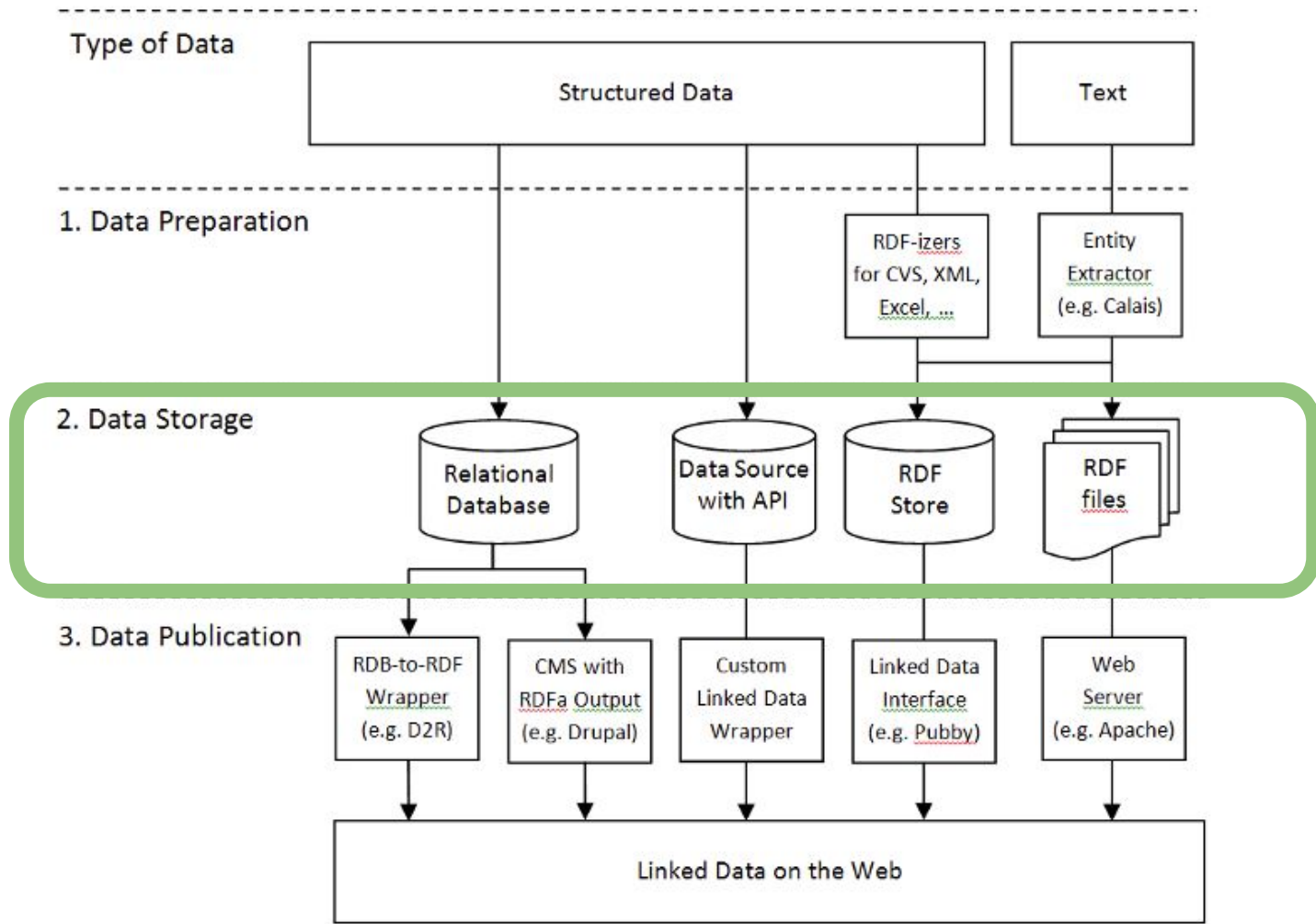
Data Preparation

- [Protégé](#)
 - tool to build simple and complex ontology-based architectures

Example - Scientists

Name	Surname	Birthday	Death Day
Albert	Einstein	1879-14-03	1955-04-18
Galileo	Galilei	1564-02-15	1642-01-08

- Use FOAF vocabulary to represent data
 - <http://xmlns.com/foaf/spec/>
- Add property deathDay to FOAF



Data Storage

- Small datasets (few triples)
 - RDF static file published on the web or stored locally
 - Examples: *.rdf, *.nt, *.ttl, *.n3, etc.
- Large datasets (thousands to millions of triples)
 - Usually in form of RDF storage
- Legacy data
 - Keep in original form
 - Provide mapping to RDF

Native RDF stores

- RDF stores that implement their own database engine without reusing the storage and retrieval functionalities of other database management systems:
 - AllegroGraph (commercial)
 - GraphDB (GNU LGPL license and commercial)
 - Stardog
 - RDFox

DBMS-backed stores

- RDF Stores that use the storage and retrieval functionality provided by another database management system:
 - Apache Jena SDB (open-source)
 - Oracle Spatial and Graph:
 - Semantics Platform
 - ARC2
 - RDFLib

Non-RDF DB support

- Store data into another format (e.g. SQL) and export them as RDF through a mapping
 - D2RQ Platform

A more complete list of triple stores can be found here

https://www.w3.org/2001/sw/wiki/Category:Triple_Store

Analysed Stores

- Graph DB
- Apache Jena (Fuseki)
- D2RQ

Graph DB

Introduction

- Graph DB is a semantic graph database
 - <http://ontotext.com/products/graphdb/>
- Three editions
 - Graph DB free
 - billions of RDF statements with at most two concurrent queries
 - Graph DB SE
 - billions of RDF statements with unlimited concurrent queries
 - Graph DB EE
 - enterprise level resilience

Setup & Run (Free Edition)

- Download the package and install it
- Run the application
 - start at <http://localhost:7200>

Jena Fuseki

Introduction

- Apache Jena is a Java framework which permits the construction of Linked Data applications
 - <http://jena.apache.org>
- Jena Fuseki provides to SPARQL endpoint to Jena

Setup & Run

- http://jena.apache.org/documentation/serving_data/
- ```
./fuseki-server --update --loc=DB /ds
```

  - `--update` sets (true) the possibility to update stored data. The default value is false (db is read-only)
  - `--loc=DB` stores data in a TDB contained in the folder DB
  - `/ds` is the name associated to the dataset path within the server. It must start with / il nome del percorso del dataset.
- <http://localhost:3030>

# Populate the dataset

```
PREFIX rdfs: <http://www.w3.org/2000&01/rdf-schema#>
```

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>
```

```
PREFIX d: <http://learningsparql.com/ns/data#>
```

## **INSERT DATA**

```
{
d:i1 a ab:Person;
ab:firstName "Dan";
ab:lastName "Jonathan" .
}
```

This first method requires too effort



## Populate the dataset (2)

- Upload an external file

# Remove data

```
PREFIX rdfs: <http://www.w3.org/2000&01/rdf-schema#>
```

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>
```

```
PREFIX d: <http://learningsparql.com/ns/data#>
```

## **DELETE DATA**

```
{
d:i1 a ab:Person;
ab:firstName "Pinco";
ab:lastName "Pallino" .
}
```

# Update existing data

PREFIX rdfs: <http://www.w3.org/2000&01/rdf-schema#>

PREFIX ab: <http://learningsparql.com/ns/addressbook#>

PREFIX d: <<http://learningsparql.com/ns/data#>>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

**DELETE**{?s ab:firstName ?o }

**INSERT** { ?s foaf:firstName ?o }

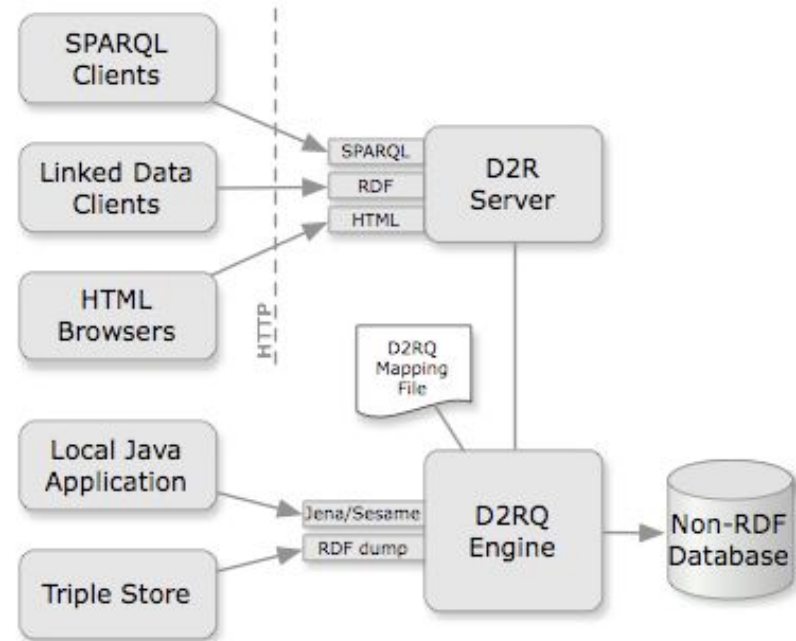
**WHERE** { ?s ab:firstName ?o }



**D2R**

# D2RQ

- **D2RQ Mapping Language**, a declarative mapping language for describing the relation between an ontology and an relational data model.
- **D2RQ Engine**, a plug-in for the Jena Semantic Web toolkit, which uses the mappings to rewrite Jena API calls to SQL queries against the database and passes query results up to the higher layers of the frameworks.
- **D2R Server**, an HTTP server that provides a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database.

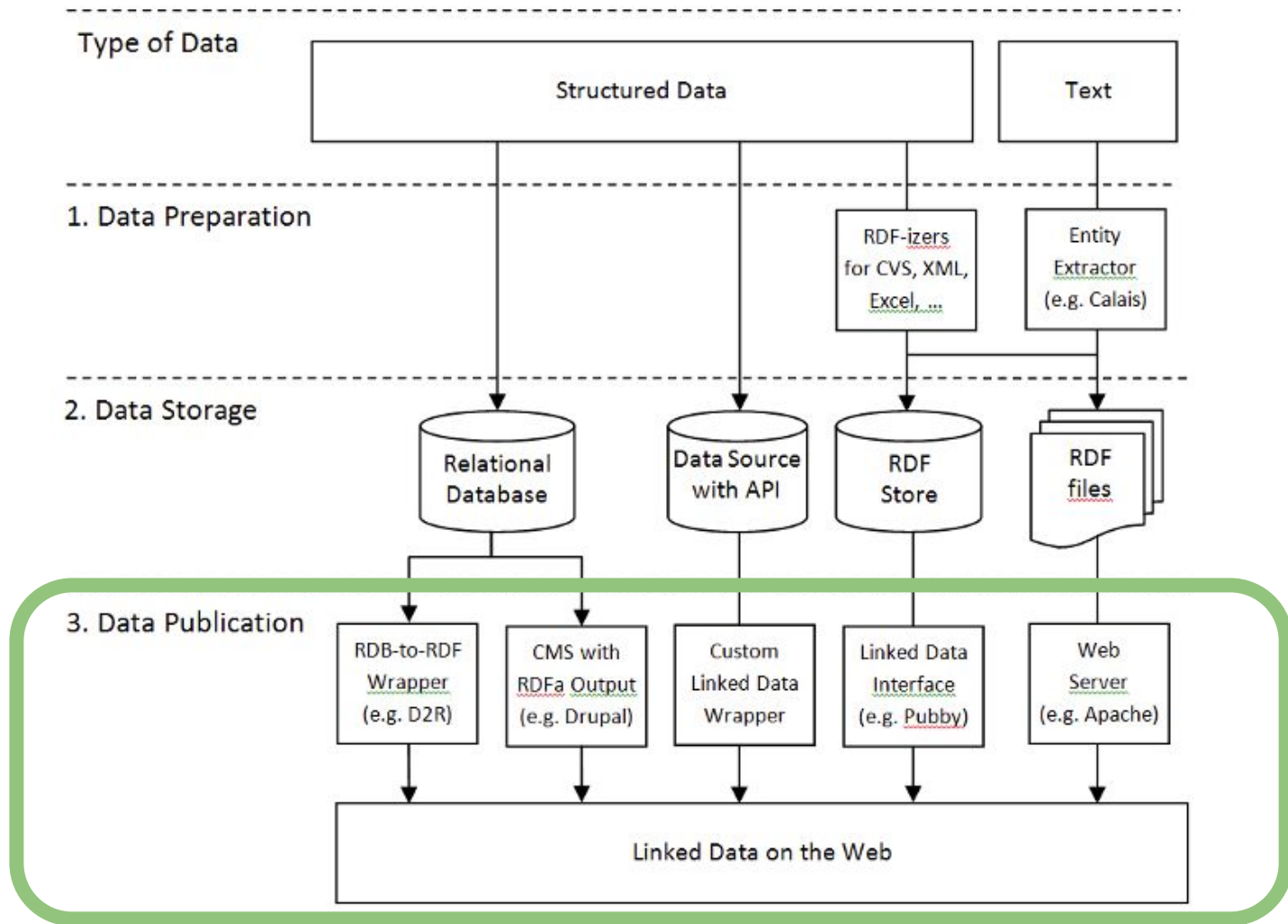


# Generate a new mapping file

- <http://d2rq.org/d2r-server>
- A basic structure is generated through the following command
  - `generate-mapping -u <user> -p <pass> -o <mapping.ttl> jdbc:mysql://localhost/<DB_name>`

## Setup & Run

- Run the server from command line
  - `d2r-server mapping-file.ttl`
- `mapping-file.ttl` maps data from D2RQ to SQL



Apache Jena and Graph DB provide only a SPARQL endpoint, but not a direct HTTP access to resources.

In order to publish a Linked Data node,  
you need an HTTP Server that dereferences HTTP URIs !!!

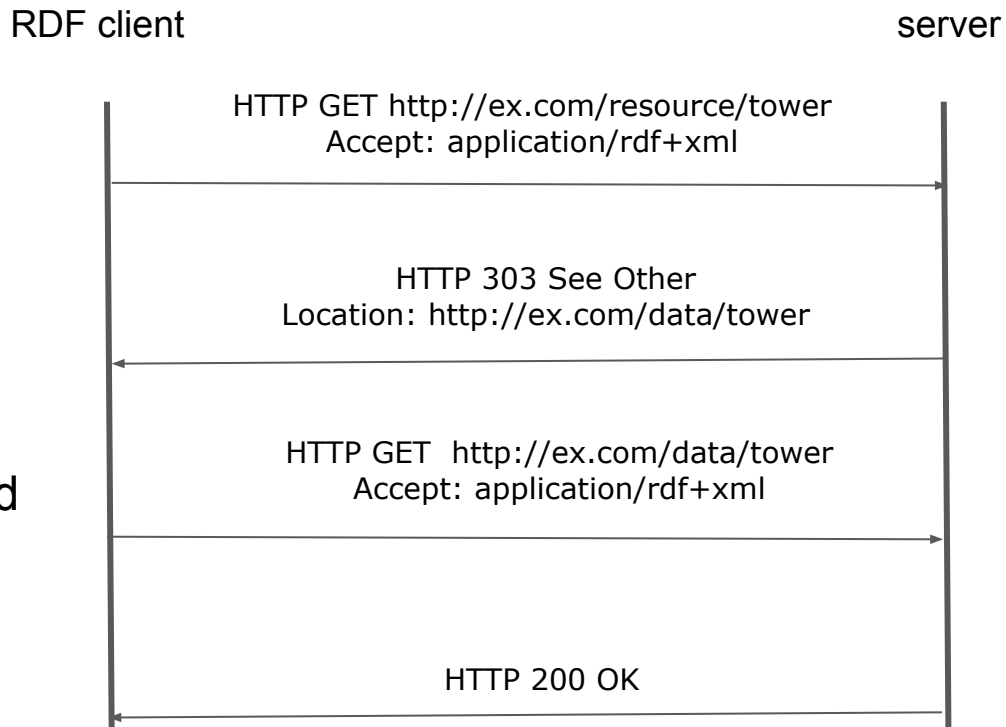
# How to build HTTP URIs

- 303 URIs
- Hash URIs



# 303 URIs

- In the *303 URIs* strategy, the Server maintains three versions of the same entity
  - one to access the entity (called generically resource)
  - one for RDF clients (called data)
  - one for HTML clients (called page).

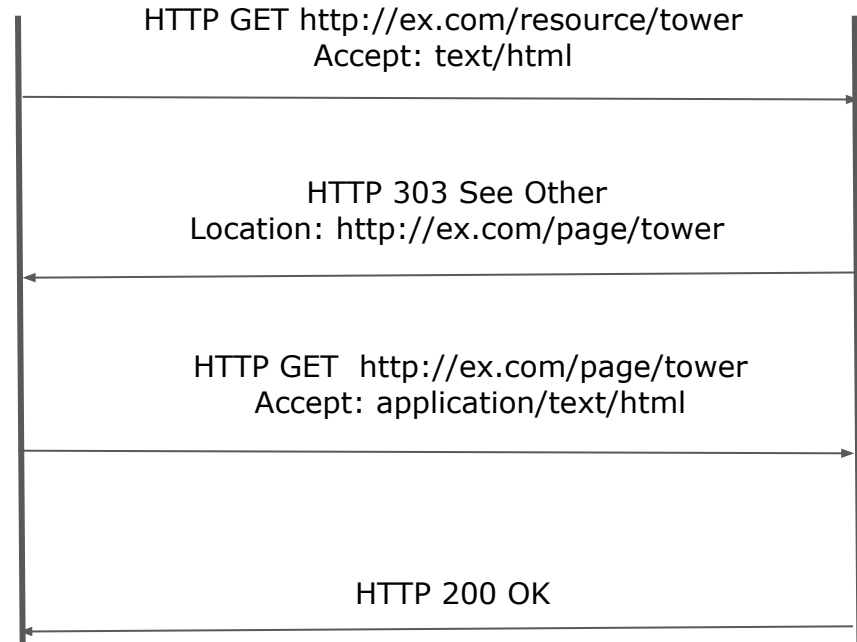


# 303 URIs

- In the *303 URIs* strategy, the server maintains three versions of the same entity
  - one to access the entity (called generically resource)
  - one for RDF clients (called data)
  - one for HTML clients (called page).

HTTP client

server



# Hash URIs

- URIs may contain a special part that is separated from the base part of the URI by a hash symbol (#), called the *fragment identifier*.
- When a client wants to retrieve a hash URI, the HTTP protocol requires the fragment part to be stripped off before requesting the URI from the server.
- The server responds with an HTTP 200 OK status and sends back to the client the whole document.

# Hash URIs vs 303 URIs

|           | Pros                                                   | Cons                                                                                                                   |
|-----------|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Hash URIs | need only one HTTP round-trip                          | the descriptions of all resources that share the same non-fragment URI part are always returned to the client together |
| 303 URIs  | only the requested resource is sent back to the client | need two HTTP round trips                                                                                              |

# Publish Linked Data

- Pubby
  - A Linked Data front-end for SPARQL endpoints
  - It is a servlet (runs under Tomcat)
  - <http://wifo5-03.informatik.uni-mannheim.de/pubby/>

# Setup & Run

- Download Tomcat
  - <http://tomcat.apache.org>
- Put the Pubby folder in the webapps folder of Tomcat and rename it as you want
- Modify the file config.ttl (located within the WEB-INF folder) according to your needs