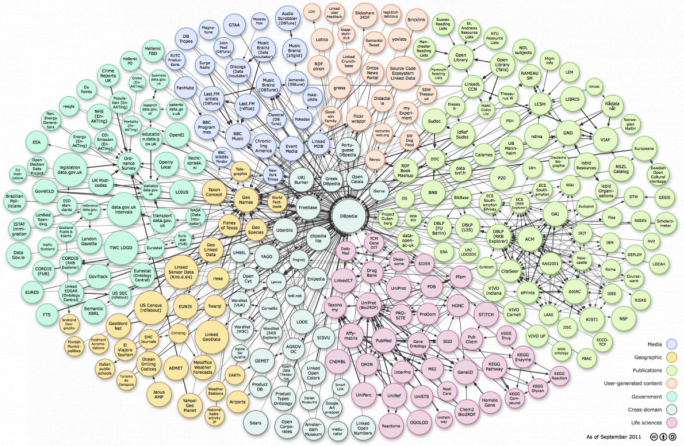
Linked Data


Media
Geographic
Publications
User-generated content
Government
Cross-domain
Life sciences
As of September 2011

Acknowledge

Some slides from:

Linked Data: Evolving the Web into a Global Data Space

and

How to Publish Linked Data on the Web

Tom Heath , Talis / KMI, The Open University

Christian Bizer , Freie Universität Berlin

Richard Cyganiak, DERI Galway, Ireland

Motivation

- Key questions raised
  - How best to provide access to data so it can be most easily reused
  - How to enable the discovery of relevant data within the multitude of available data sets?
  - How to enable applications to integrate data from large numbers of formerly unknown data sources

The Web of Linked Documents

- Many companies and organisations make their data available to third parties via a Web API
- Some Web APIs provide results in structured data formats such as XML and JSON, but there are limitations:

HTML

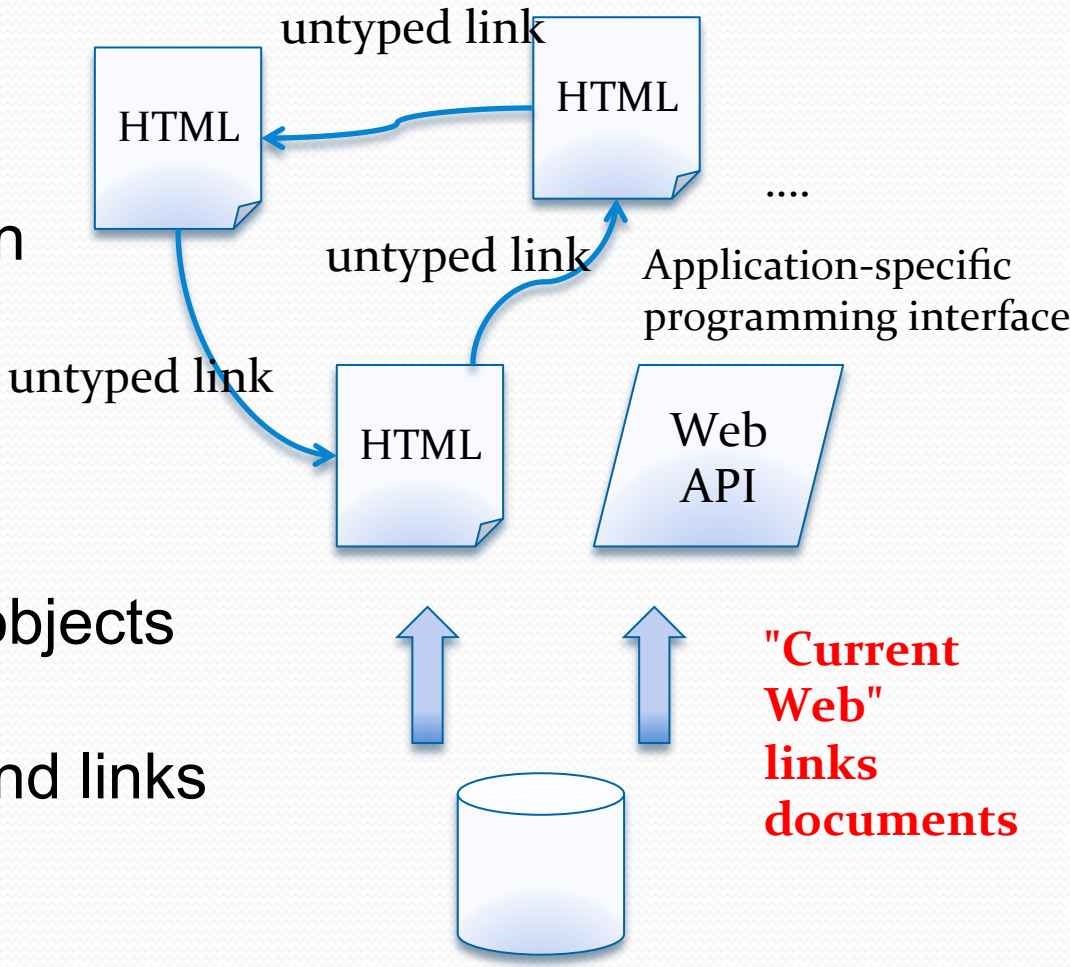
- HTML tags/attributes such as href indicate an outgoing link from the current document.
- A human user or search engine crawlers can follow or to traverse the link to retrieve the referenced document
- Connectivity between documents, supported by a standard syntax
- But links are untyped and only human readable

Web API

- The data returned does not have the equivalent of the HTML href attribute, to indicate links that should be followed to find related data
- Identifiers only have local scope e.g. Identifier such as PID001 returned is meaningless when taken out of the context of that specific API.
- Dose NOT make the data linkable and discoverable

The Web of Linked Documents

- Analogy
  - a global **filesystem**
- Designed for
  - **human** consumption
- Primary objects
  - **documents**
- Links between
  - **documents**
- Degree of structure in objects
  - fairly **low**
- Semantics of content and links
  - **implicit**

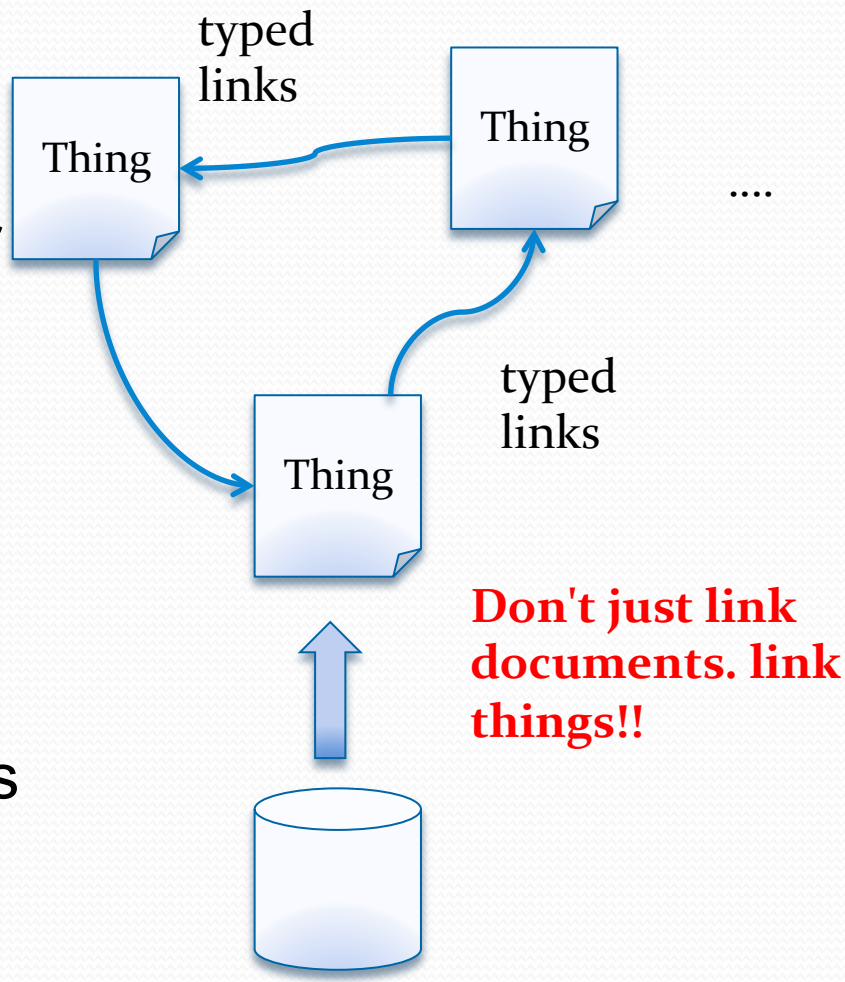


The Web of Linked Data

- To create Web of Data
  - It is important to have the huge amount of data on the Web available in a standard format
  - Reachable and manageable by Semantic Web tools.
  - The *relationships among data* should be made available.

The Web of Linked Data

- Analogy
  - a global **database**
- Designed for
  - **machine** first, human later
- Primary objects
  - **things**
- Links between
  - **things**
- Degree of structure in objects
  - **high**
- Semantics of content and links
  - **explicit**



What is Linked Data?

"Like the web of hypertext, the web of data is constructed with documents on the web. However, unlike the web of hypertext, where links are relationships anchors in hypertext documents written in HTML, for data they links between arbitrary things described by RDF"

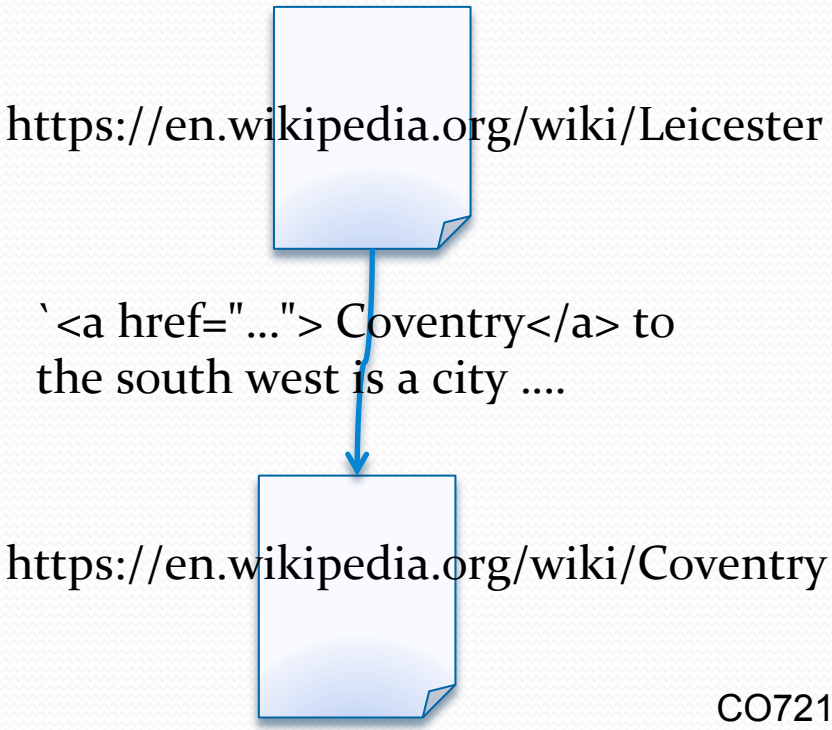
-- Tim Berners-Lee, Linked Data

Linked Data vs HTML

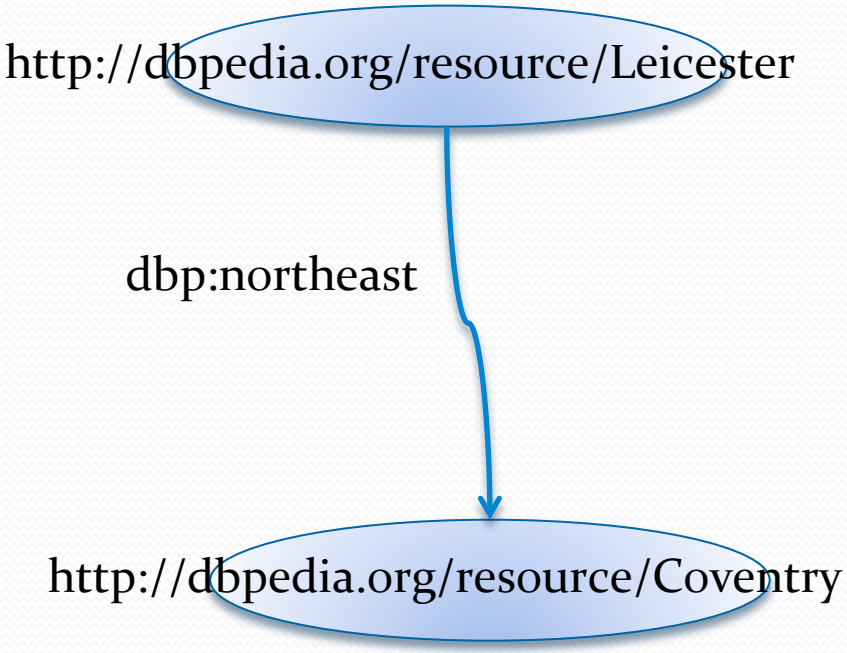
(1) RDF links things, not just documents

"Leicester is located to the northeast of Coventry"

Wikipedia (HTML)



Dbpedia (RDF)

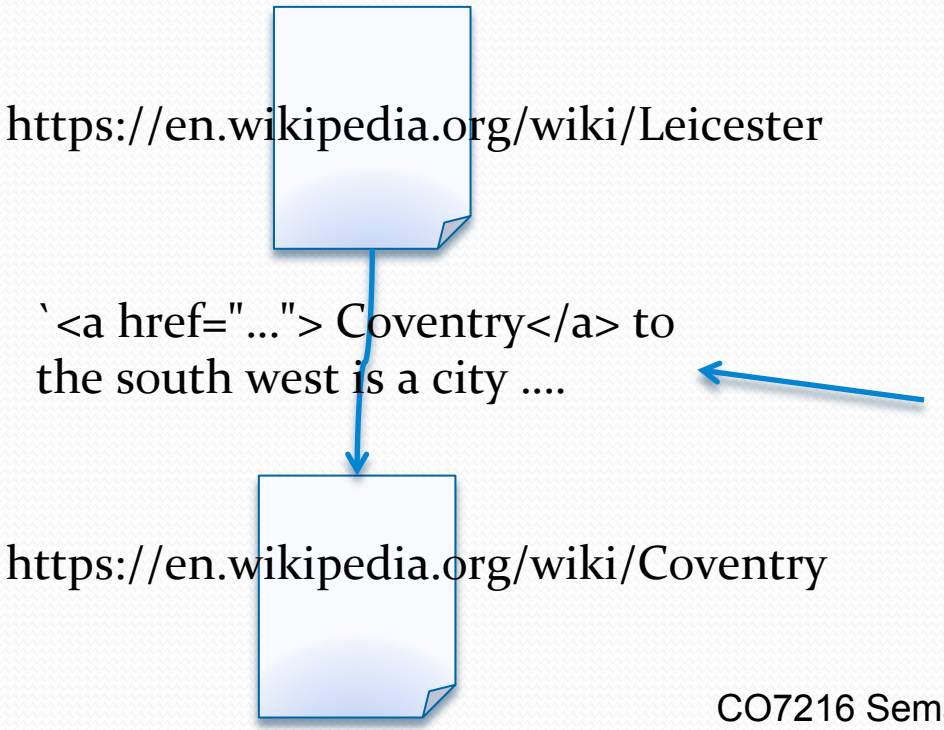


Linked Data vs HTML (2)

(2) RDF links are typed

"Leicester is located to the northeast of Coventry"

Wikipedia (HTML)



HTML hyperlinks typically indicate that two documents are related in some way, but mostly leave the user to infer the nature of the relationship.

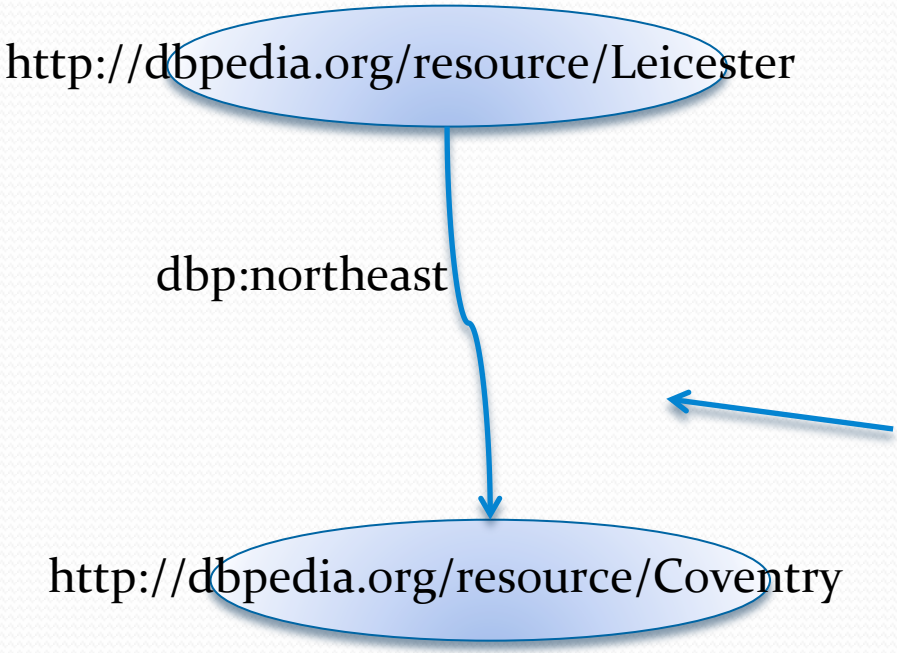
e.g. user needs to read the surrounding text to put the hyperlink into context.

Linked Data vs HTML (2) cont.

(2) RDF links are typed

"Leicester is located to the northeast of Coventry"

Dbpedia (RDF)



RDF property is used to indicate the relationship between two resources.

e.g. in contrast to untyped hyperlink, "*dbp:northeast*" is a machine-readable RDF property, its semantics is defined.

Linked Data is ..

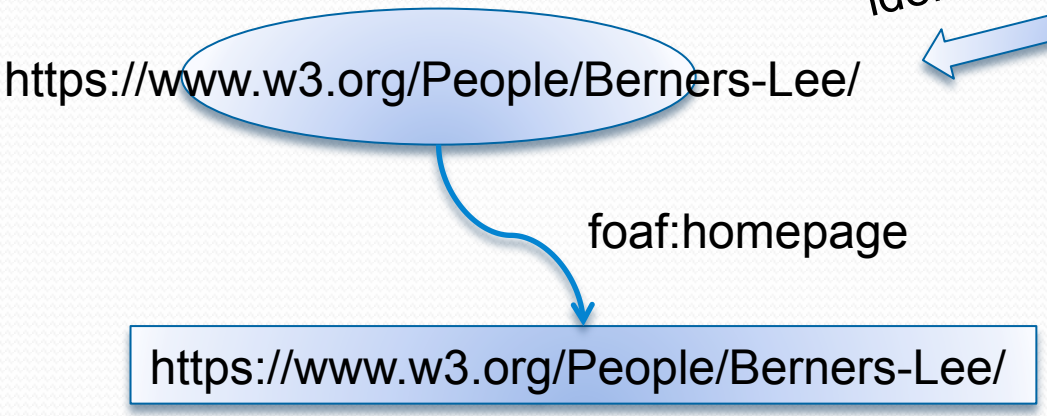
- ...a way of publishing data on the Web that:
  - encourages reuse
  - reduces redundancy
  - maximises its (real and potential) inter-connectedness
  - enables network effects to add value to data

Principles of Linked Data

- Use URIs as names for things.
  - anything, not just documents
  - you are not your homepage
  - information resources and non-information resources
- Use HTTP URIs, so that people can look up those names.
  - globally unique names, distributed ownership
  - allows people to look up those names
- Provide useful information,
  - When someone looks up a URI,
  - using the standards (RDF, SPARQL).
- Include links to other URIs
  - to enable discovery of related information

Use URIs as names for things.

- Use URI references to identify real world objects and abstract concepts, which could be
  - tangible things: a place, city , person etc.
  - abstract concepts or relationships: topics, subTopicOf, isLocatedIn, or a document on the web: https://www.w3.org/People/Berners-Lee/



- create globally unique names
- as a means of accessing information

Making URIs Defererenceable.

- Any HTTP URI should be dereferenceable, meaning that HTTP clients can look up the URI using the HTTP protocol and retrieve a description of the resource that is identified by the URI.
- Where URIs identify real-world objects, it is essential to not confuse the objects themselves with the Web documents that describe them.
- It is, therefore, common practice to use different URIs to identify the real-world object and the document that describes it, in order to be unambiguous.

Content Negotiation

- The Web is intended to be an information space that may be used by humans as well as by machines.
- Both should be able to retrieve representations of resources in a form that meets their needs, such as HTML for humans and RDF for machines.
- This can be achieved using an HTTP mechanism called content negotiation.

Content Negotiation (2)

- The basic idea of content negotiation is that HTTP clients send HTTP headers with each request to indicate what kinds of documents they prefer.
- Servers can inspect these headers and select an appropriate response.
- If the headers indicate that the client prefers HTML, then the server will respond by sending an HTML document. If the client prefers RDF, then the server will send the client an RDF document.

Content Negotiation (3)

- There are two different strategies to make URIs that identify real-world objects dereferenceable.
- Both strategies ensure that objects and the documents that describe them are not confused, and that humans as well as machines can retrieve appropriate representations.
- The strategies are called **303 URIs** and **hash URIs**.

303 URIs

- Real-world objects, like houses or people, can not be transmitted over the wire using the HTTP protocol. Thus, it is also not possible to directly dereference URIs that identify real-world objects.

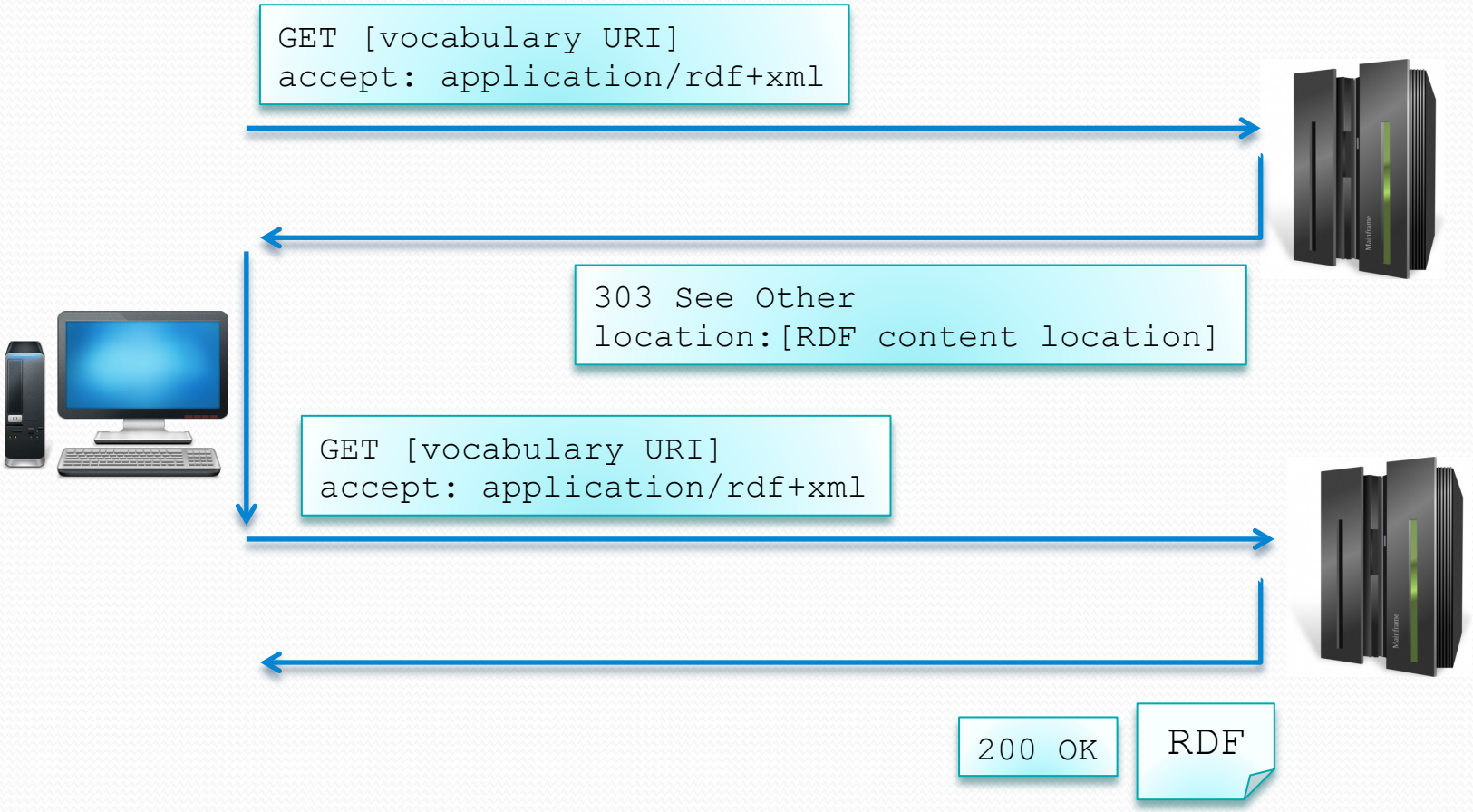
- the server responds to the client with the HTTP response code 303 See Other and the URI of a Web document which describes the real-world object. This is called a 303 redirect.

- In a second step, the client dereferences this new URI and gets a Web document describing the real-world object

Dereferencing a HTTP URI via 303 redirect

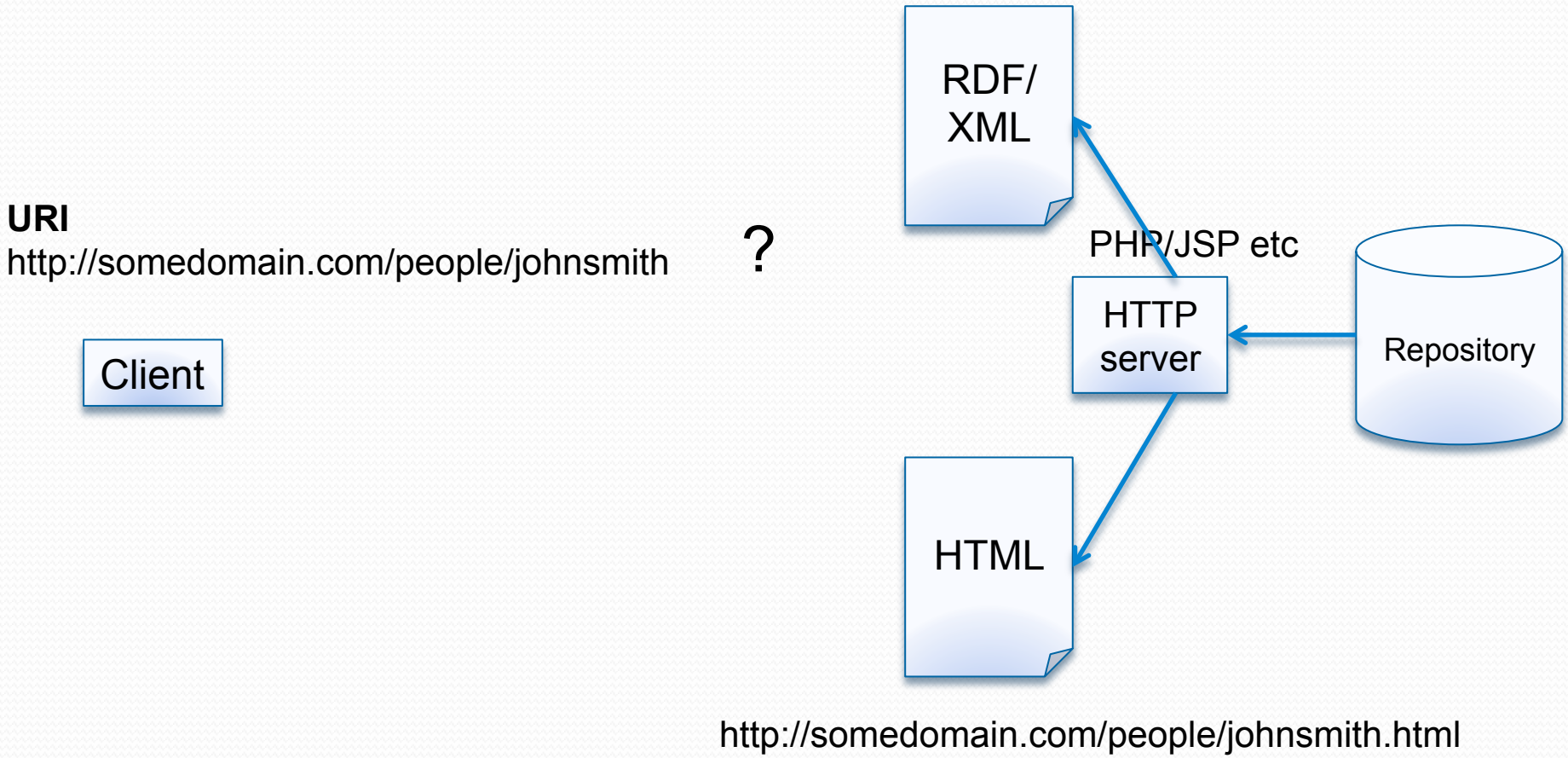
- 4 steps
  - Step 1: the client performs a HTTP GET request on a URI identifying a real-world object or abstract concept. Different clients send different header along with the request.
  - Step 2: The server recognises that the URI identifies a real-world object or abstract concept. If the server does not have a representation of this resource, it answers using the HTTP 303 See Other response code and sends the client another URI with the requested format.
  - Step 3: The client performs an HTTP GET request on this URI returned by the server.
  - Step 4: Answers with a HTTP response code 200 OK and sends the client the requested document

Dereferencing a HTTP URI via 303 redirect



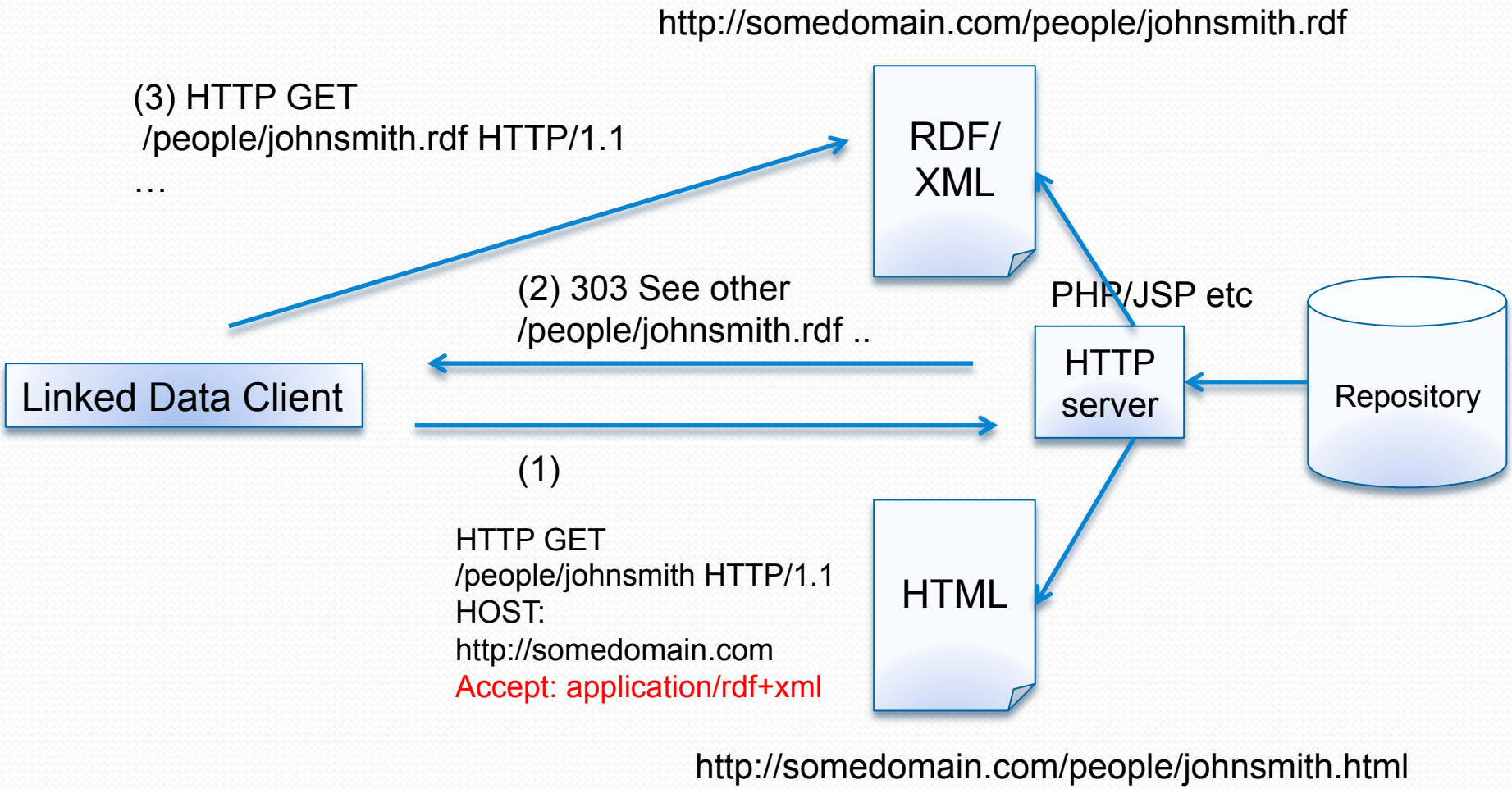
Dereferencing a HTTP URI via 303 redirect

http://somedomain.com/people/johnsmith.rdf

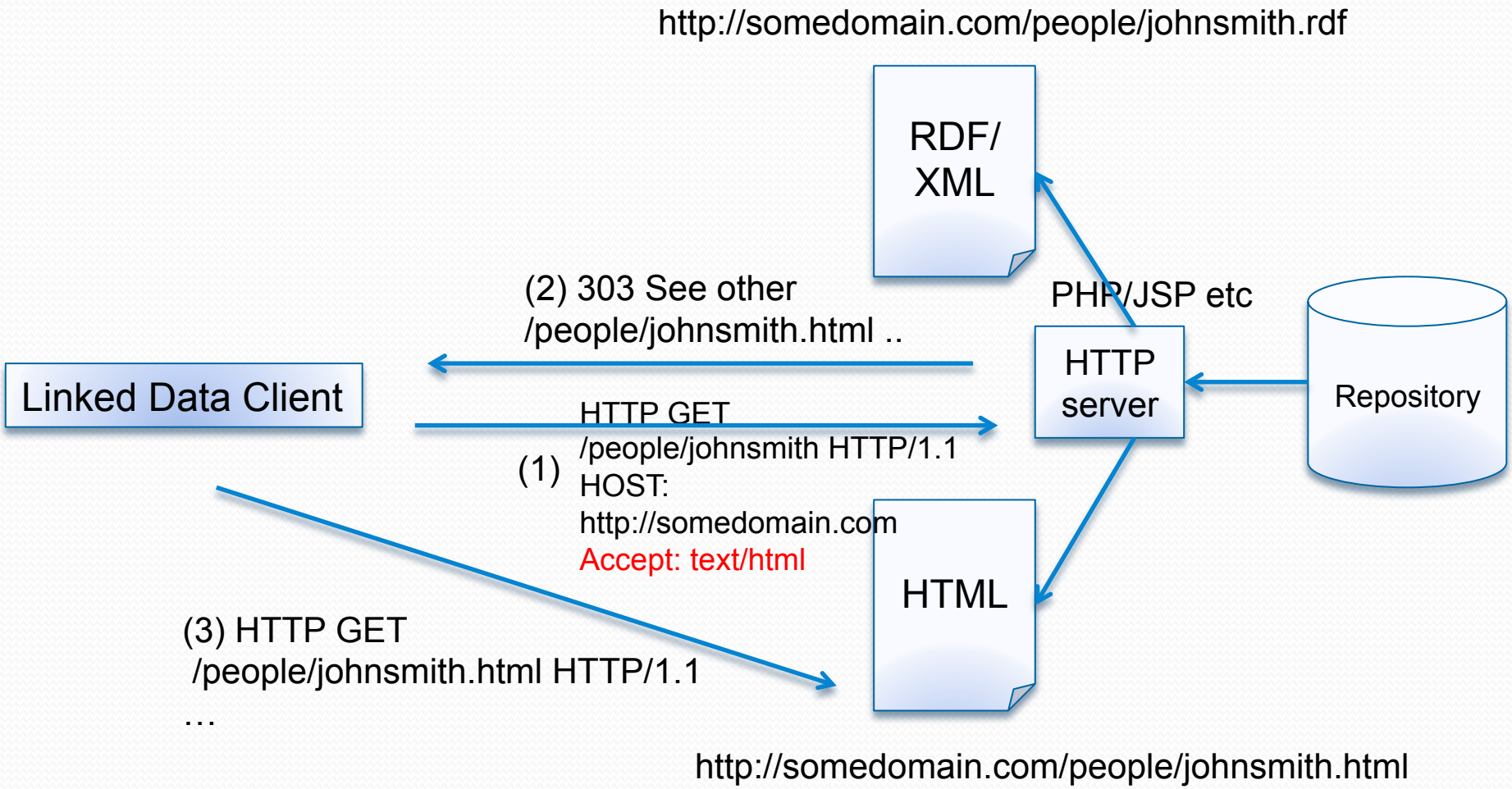


http://somedomain.com/people/johnsmith.html

Dereferencing a HTTP URI via 303 redirect



Dereferencing a HTTP URI via 303 redirect



Hash URIs

- A widespread criticism of the 303 URI strategy is that it requires two HTTP requests to retrieve a single description of a real-world object.
- The hash URI strategy builds on the characteristic that URIs may contain a special part that is separated from the base part of the URI by a hash symbol (#). This special part is 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.
- This means a URI that includes a hash cannot be retrieved directly and therefore does not necessarily identify a Web document. This enables such URIs to be used to identify real-world objects and abstract concepts, without creating ambiguity.

Hash URIs (2)

http://somedomain.com/vocab#johnsmith

The returned document not only a description of http://somedomain.com/vocab#johnsmith but also other triples

```
1 HTTP/1.1 200 OK
2 Content-Type: application/rdf+xml;charset=utf-8
3
4
5 <?xml version="1.0"?>
6 <rdf:RDF
7 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
8 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
9
10 <rdf:Description rdf:about="http://somedomain.com/vocab#johnsmith">
11 <rdf:type rdf:resource="http://somedomain.com/vocab#Person" />
12 </rdf:Description>

13 <rdf:Description rdf:about="http://somedomain.com/vocab#janesmith">
14 <rdf:type rdf:resource="http://somedomain.com/vocab#Person" />
15 </rdf:Description>
16 ...
```

If the triples returned are not related to the request then they will be discard by the Linked Data-aware client

303 vs Hash URIs

- 303 URIs
  - *Pros:* very flexible, the redirection target can be configured separately for each resource
  - *Cons:* multiple HTTP requests can cause latency
  - Often used to serve resource descriptions that are part of very large data sets (e.g. DBpedia)
- Hash URIs
  - *Pros*: reduces access latency by reducing the number of necessary HTTP round-trips
  - *Cons*: a large amount of data being unnecessarily transmitted
  - Often used to identify terms within RDF vocabularies as the definitions of RDF vocabularies are usually rather small (e.g. FOAF vocabularies)

Content Negotiation Experiment: DBpedia

Dbpedia

- likes "RDF-version" of Wikipedia, consisting of 3.6 million concepts which are described by over 380 million triples
- uses 303 redirection

HTTP GET
/resource/Leicester
HOST:
http://dbpedia.org
Accept: application/rdf+xml

http://dbpedia.org/resource/Leicester

Windws / Linux:

wget –header="Accept: application/rdf+xml" -L http://dbpedia.org/resource/Leicester

****For Mac OS X, use :***

curl --header "Accept: application/rdf+xml" -L http://dbpedia.org/resource/Leicester

Alternatively you could use Linked Data browsers:
https://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients

Experiment: DBpedia

wget –header="Accept: application/rdf+xml" -L http://dbpedia.org/resource/Leicester

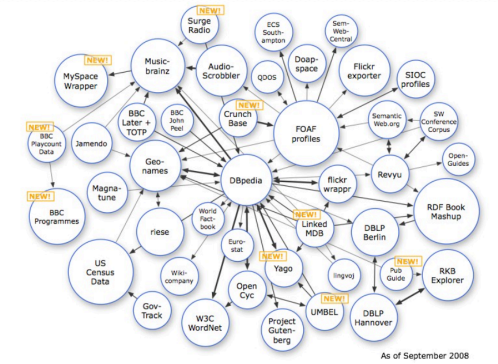
```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:georss="http://www.georss.org/georss/"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:dbp="http://dbpedia.org/property/"
        xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
        xmlns:ns7="http://www.w3.org/ns/prov#"
        xmlns:ns8="http://dbpedia.org/ontology/PopulatedPlace/"
        xmlns:dbo="http://dbpedia.org/ontology/"
        xmlns:dct="http://purl.org/dc/terms/" >
 <rdf:Description rdf:about="http://dbpedia.org/resource/C._P._Snow">
  <dbo:birthPlace rdf:resource="http://dbpedia.org/resource/Leicester" />
  <dbp:birthPlace rdf:resource="http://dbpedia.org/resource/Leicester" />
  <dbp:placeOfBirth rdf:resource="http://dbpedia.org/resource/Leicester" />
 </rdf:Description>
 <rdf:Description rdf:about="http://dbpedia.org/resource/Fullhurst_Community_College">
  <dbo:city rdf:resource="http://dbpedia.org/resource/Leicester" />
  <dbo:localAuthority rdf:resource="http://dbpedia.org/resource/Leicester" />
  <dbp:city rdf:resource="http://dbpedia.org/resource/Leicester" />
  <dbp:lea rdf:resource="http://dbpedia.org/resource/Leicester" />
 </rdf:Description>
.....
```

Experiment: Dbpedia (cont.)

wget –header="Accept: text/html" -L http://dbpedia.org/resource/Leicester

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://
www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:dbpprop="http://dbpedia.org/property/"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    version="XHTML+RDFa 1.0" xml:lang="en">


<!-- header -->
<head profile="http://www.w3.org/1999/xhtml/vocab">
    <title>About: Leicester</title>
    <link rel="alternate" type="application/rdf+xml" href="http://dbpedia.org/
data/Leicester.rdf" title="Structured Descriptor Document (RDF/XML format)" /
>
    <link rel="alternate" type="text/rdf+n3" href="http://dbpedia.org/data/
Leicester.n3" title="Structured Descriptor Document (N3/Turtle format)" />
…..
```



How to publish Linked Data

Steps to Publishing Linked Data

- Understand the Principles *(You already did!)*
- Understand your Data
- Choose URIs for Things in your Data
- Setup Your Infrastructure
- Link to other Data Sets

Understand Your Data

- Support we want to publish linked data for the King Richard III of England, what are the key things present in your data?
  - People?
  - Era?
  - Marriage and Family Relationship?
  - Places?
  - Historical Events
  - Location?
  - Discovery?
  - Books?
  - ….

Understand Your Data

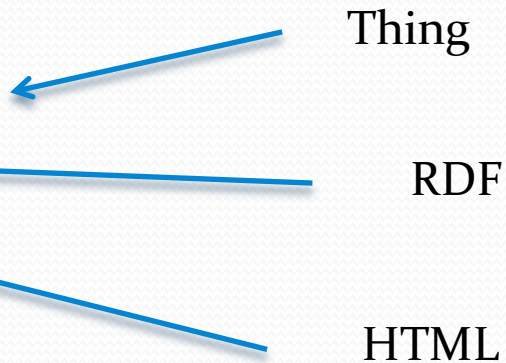
- What vocabularies can be used
  - *Principles*
    - *Reuse, don't reinvent*
    - *Mix liberally*
  - *Potential Ontologies/Vocabularies*
    - *CIDOC-CRM ontology*
    - *Geo W3C Geospatial Ontologies*
    - *Family Ontology http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl)*
    - *Dublin Core*
    - …

Choose URIs for Things in your Data

- Use HTTP URIs
- Keep out of other peoples' namespaces
  - http://www.imdb.com/title/tt0441773/
  - http://www.imdb.com/title/tt0441773/thing
  - http://myfilms.com/tt0441773
  - http://myfilms.com/tt0441773/html
- Abstract away from implementation details
  - http://dbpedia.org/resource/Berlin
  - http://www4.wiwiss.fu-berlin.de:2020/demos/dbpedia/cgi- bin/ resources.php?id=Berlin
- Hash or Slash
  - http://mydomain.com/foaf.rdf#me
  - http://mydomain.com/id/me

Choosing URIs: Common Patterns

- • *http://dbpedia.org/resource/New_York_City*
- • *http://dbpedia.org/data/New_York_City*
- • *http://dbpedia.org/page/New_York_City*

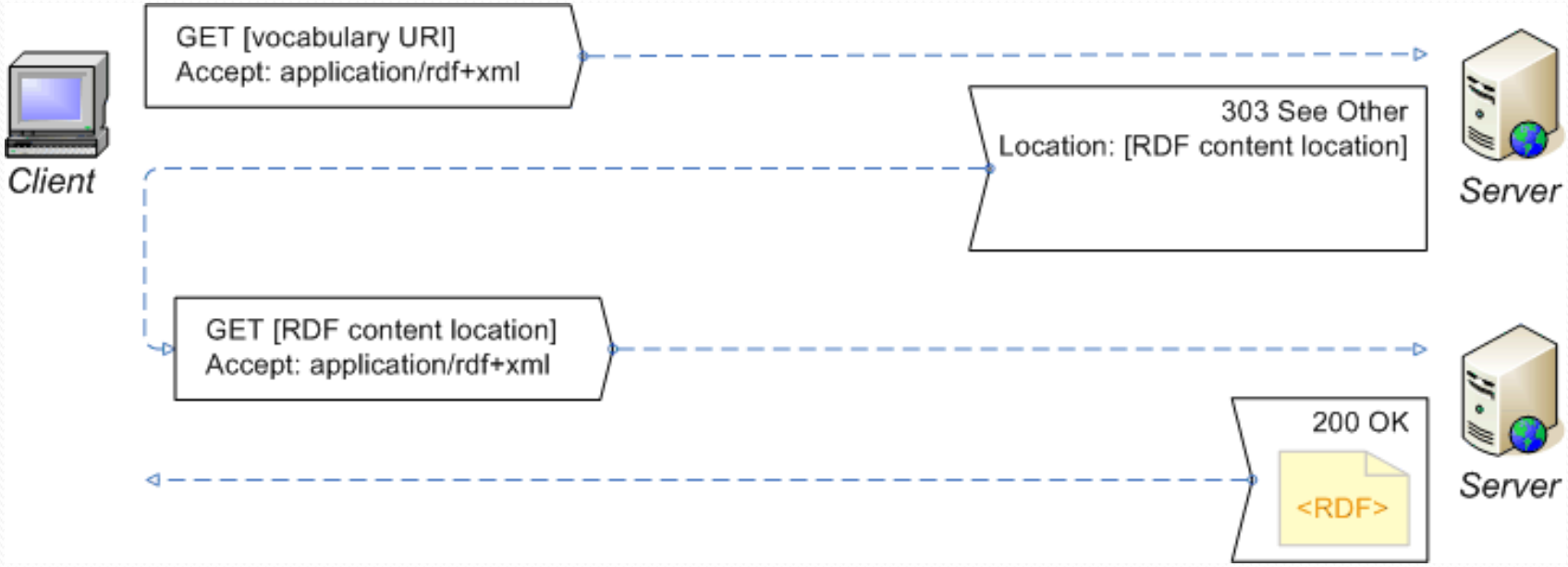


- • *http://revyu.com/people/tom*
- • *http://revyu.com/people/tom/about/rdf*
- • *http://revyu.com/people/tom/about/html*

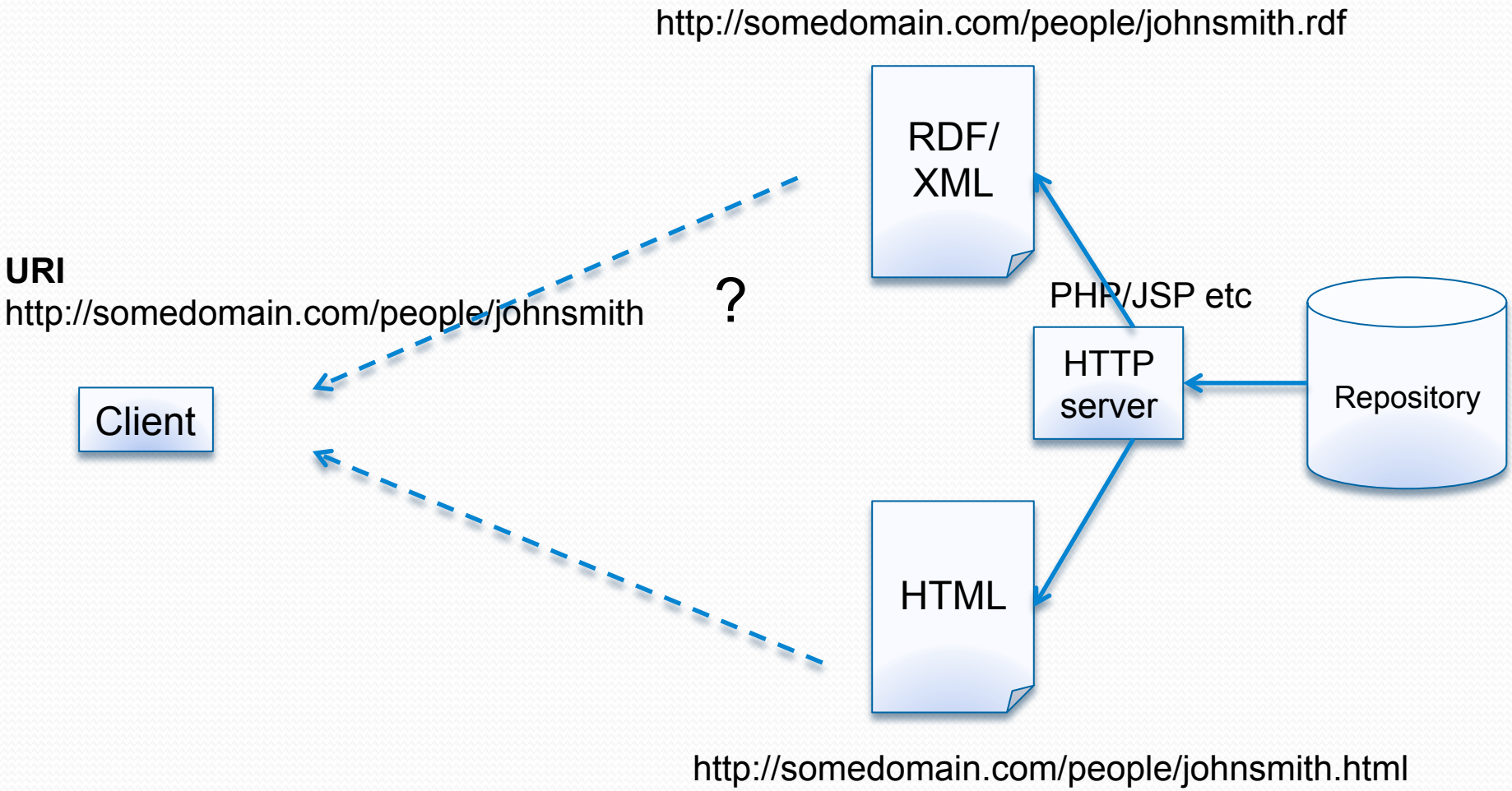
- • *http://kmi.open.ac.uk/people/tom/*
- • *http://kmi.open.ac.uk/people/tom/rdf*
- • *http://kmi.open.ac.uk/people/tom/html*

Setup Your Infrastructure

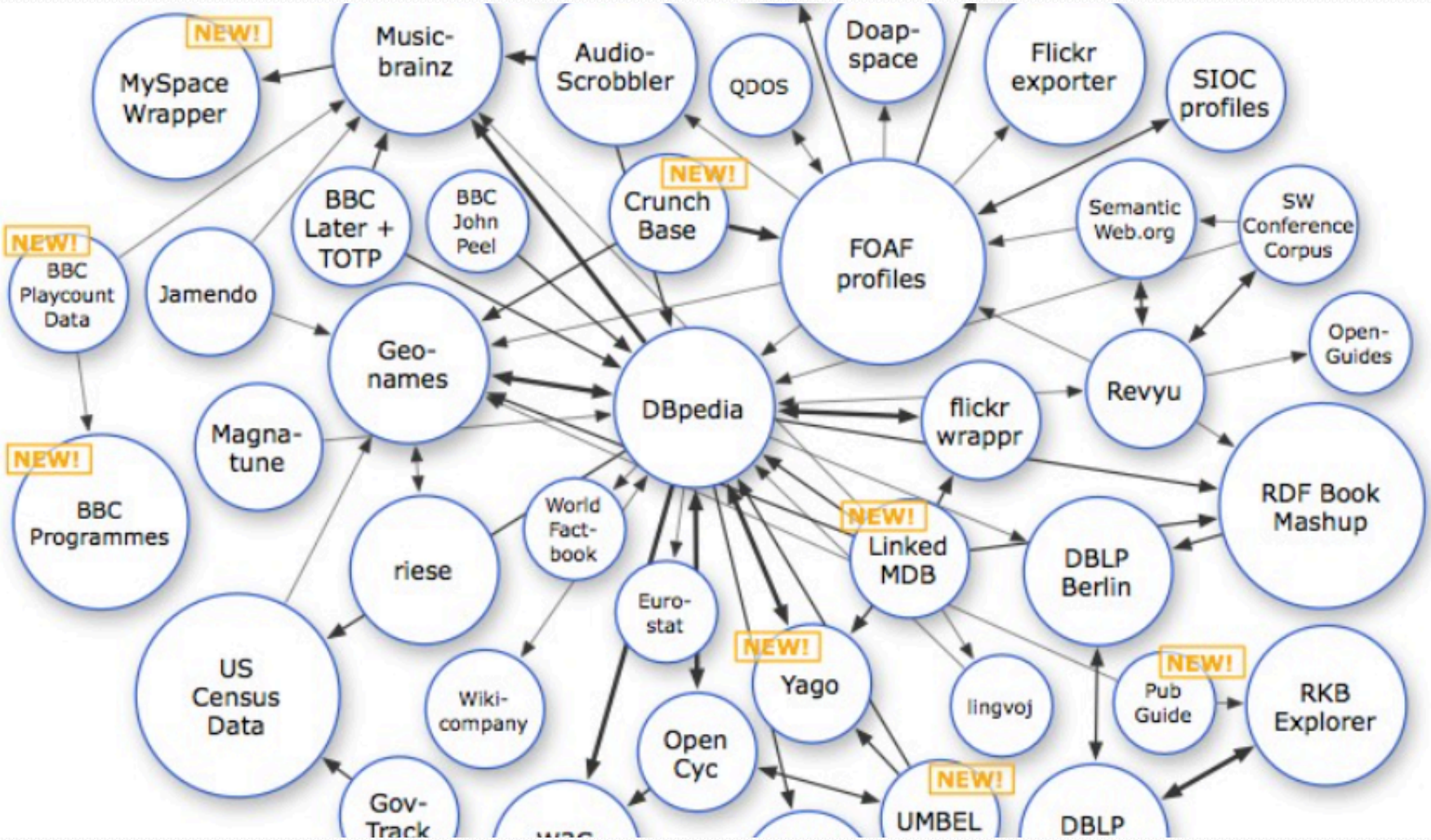
- Content Negotiation (303 Redirect)



Setup Your Infrastructure



Link to Other Data Sets



Link to Other Data Sets

- Popular Predicates for Linking
  - owl:sameAs
  - foaf:homepage
  - foaf:topic
  - foaf:based_near
  - foaf:maker/foaf:made
  - foaf:depiction
  - foaf:page
  - foaf:primaryTopic
  - rdfs:seeAlso

Link to other Data Sets

- Linking Algorithms
  - String Matching
    - e.g. Lexical Distance between labels
  - Common Key Matching
    - e.g. ISBN, Musicbrainz IDs
  - Property-based Matching
    - Do these two things have the same label, type and coordinates
  - Aim for reciprocal links

Summary – How To Publish Linked Data

- Understand the Principles
- Understand your Data
- Choose URIs for Things in your Data
- Setup Your Infrastructure
- Link to other Data Sets

Essential Reading

- Free online book, Linked Data: Evolving the Web into a Global Data Space.
  - http://linkeddatabook.com/editions/1.0/ Chapters 1-3.