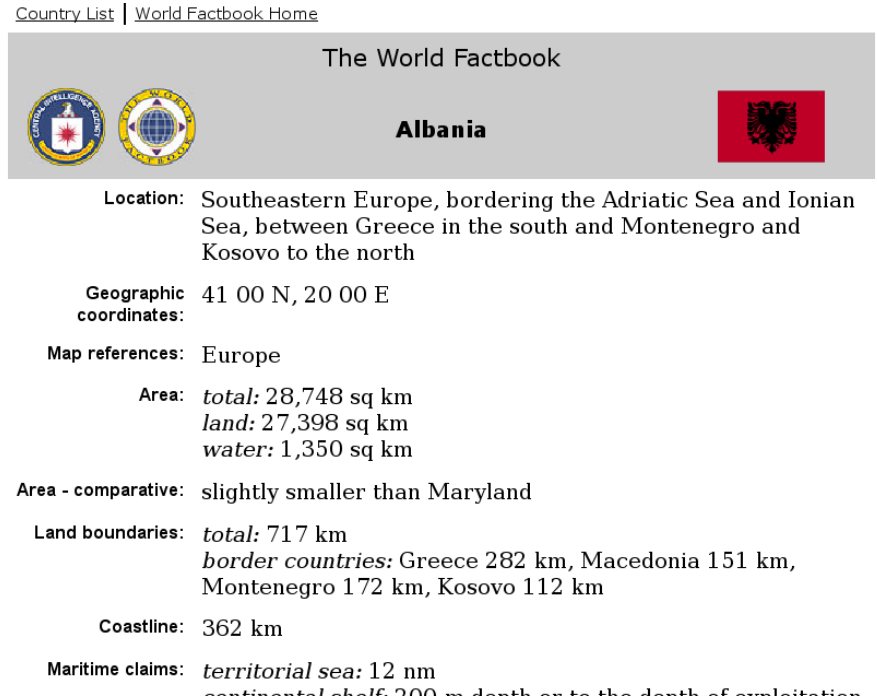
Linked Data on the Web

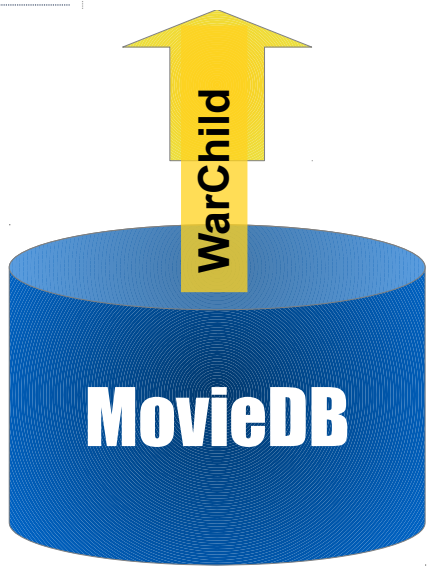
Olaf Hartig

David R. Cheriton School of Computer Science
University of Waterloo

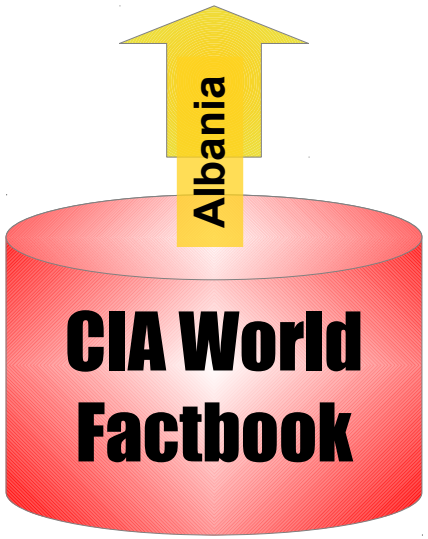
Guest Lecture in CS 743
Oct. 14, 2014

Data on the Traditional, Hypertext Web





Data exposed to the Web via HTML



Data on the Traditional, Hypertext Web

So what is the problem?

- Web content is only loosely structured
- Difficult for applications to do smart things

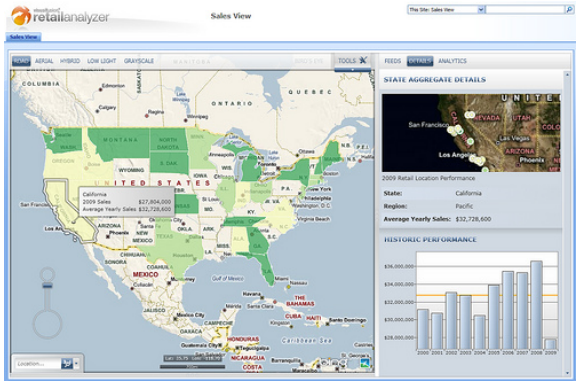
Solution:

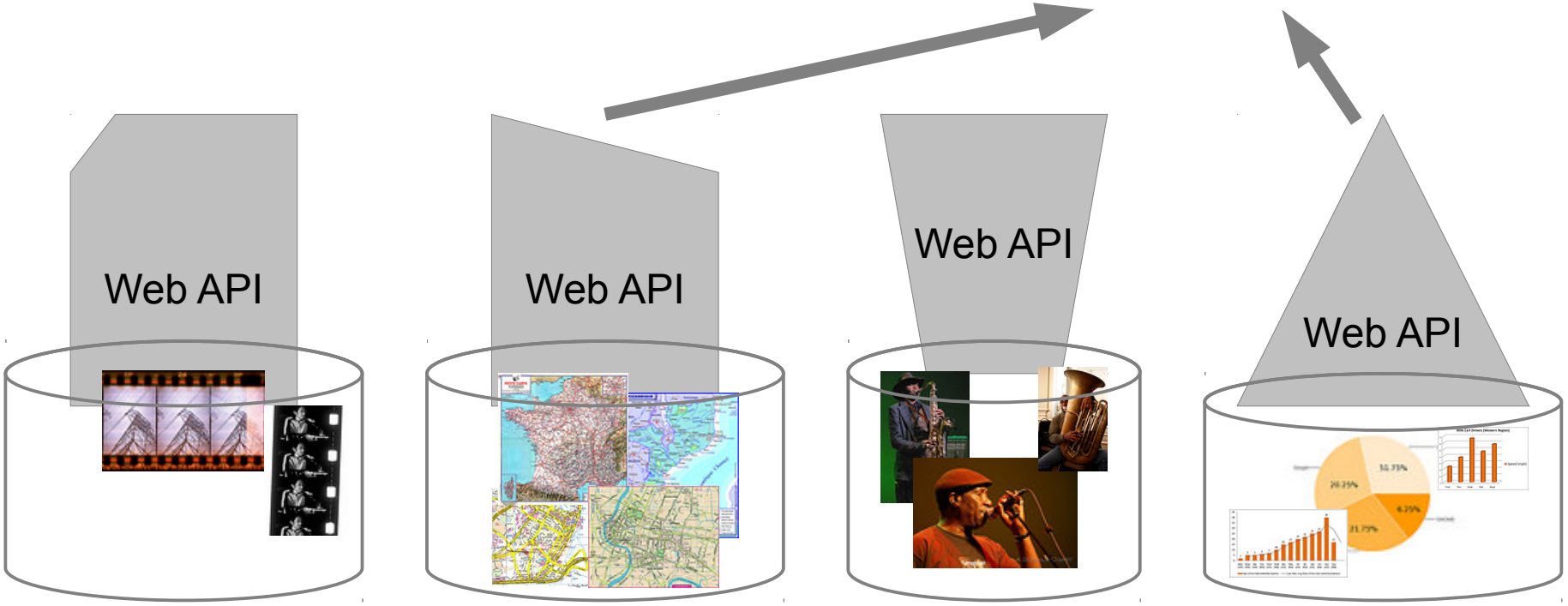
- Increase the structure of Web content
- Publish data in an machine-friendly way

But wait…
don't we do that already?

Web APIs

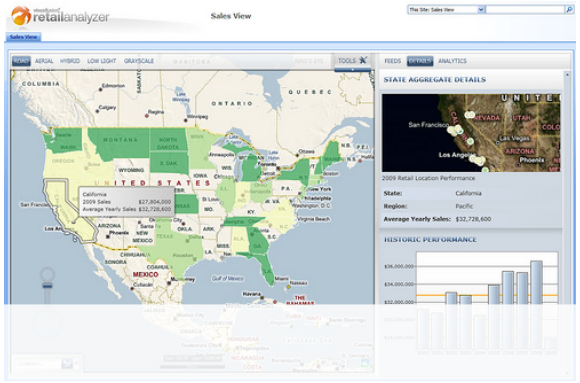
- **Content providers offer access via Web APIs**
- **Mashups combine this data**





Web APIs

- **Content providers offer access via Web APIs**
- **Mashups combine this data**



Shortcomings:

- APIs are proprietary
- Mashups are based on a fixed set of data sources
- You can not set hyperlinks between data object

Towards a Web of Linked Data

By using the following, well-established Web technologies the WWW evolves into a Web of Linked Data.

- Access mechanism: Hypertext Transfer Protocol (HTTP)
- Data model: The Resource Description Framework (RDF)
- Globally unique identifiers: Uniform Resource Identifier (URI)

Outline

(1) The Foundation: Linked Data on the Web

- The RDF Data Model and URIs
- The SPARQL Query Language
- The Linked Data Publishing Principles

(2) Querying Linked Data

The Resource Description Framework

- **A resource may basically be everything**
  - e.g. persons, places, Web documents, abstract concepts
- **Descriptions of resources**
  - Attributes
  - Relations
- **The framework contains:**
  - A data model, and
  - Languages and syntaxes

RDF Data Model

- **Data comes as a set of triples (subject, predicate, object)**

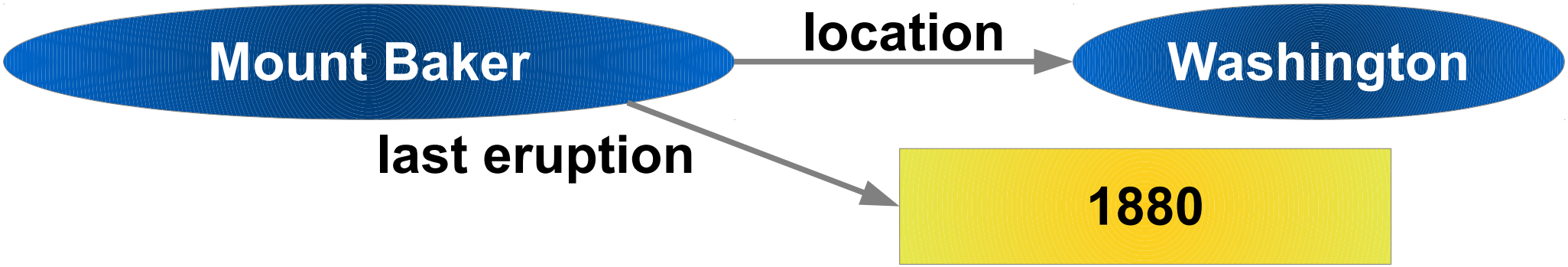
- **Subject: resources**
- **Predicate: properties**
- **Object: literals or resources**

- **Examples:**
  - (Mount Baker , last eruption , 1880)
  - (Mount Baker , location , Washington)

RDF Data Model (cont'd)

- **RDF based data may be understood as a graph:**
  - Triples as directed edges
  - Subjects and objects as vertices
  - Edges labeled by predicate

- **Example:**
  - (Mount Baker , last eruption , 1880)
  - (Mount Baker , location , Washington)



Uniform Resource Identifier (URI)

- **URIs extend the concept of URLs**
  - Globally unique identifier for resources
  - URL of a Web document usually used as its URI
  - Attention: URIs identify not only Web documents
- **Example:**
  - Me:

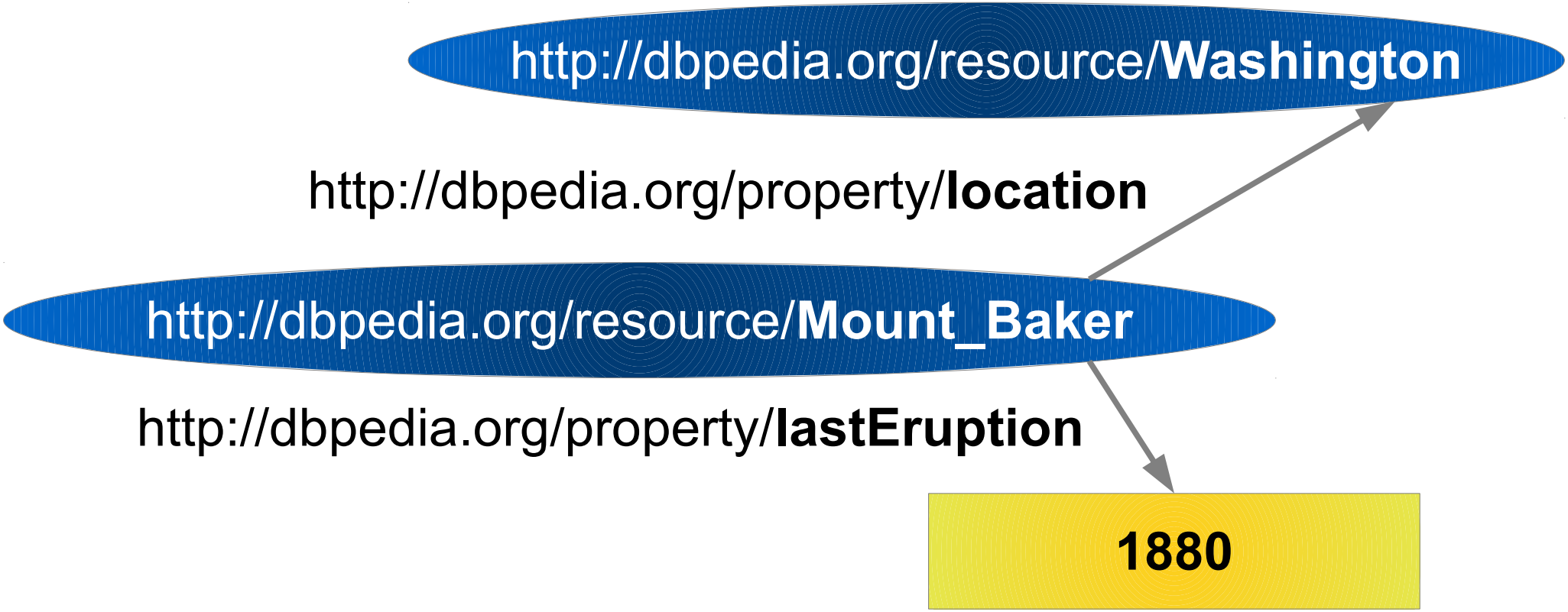
    http://olafhartig.de/foaf.rdf#olaf
  - RDF document about me:

    http://olafhartig.de/foaf.rdf
  - HTML document about me:

    http://olafhartig.de/index.html

Example (revisited)

- (http://dbpedia.org/resource/**Mount_Baker**, http://dbpedia.org/property/**lastEruption**, **1880**)
- (http://dbpedia.org/resource/**Mount_Baker**, http://dbpedia.org/property/**location**, http://dbpedia.org/resource/**Washington**)



Outline

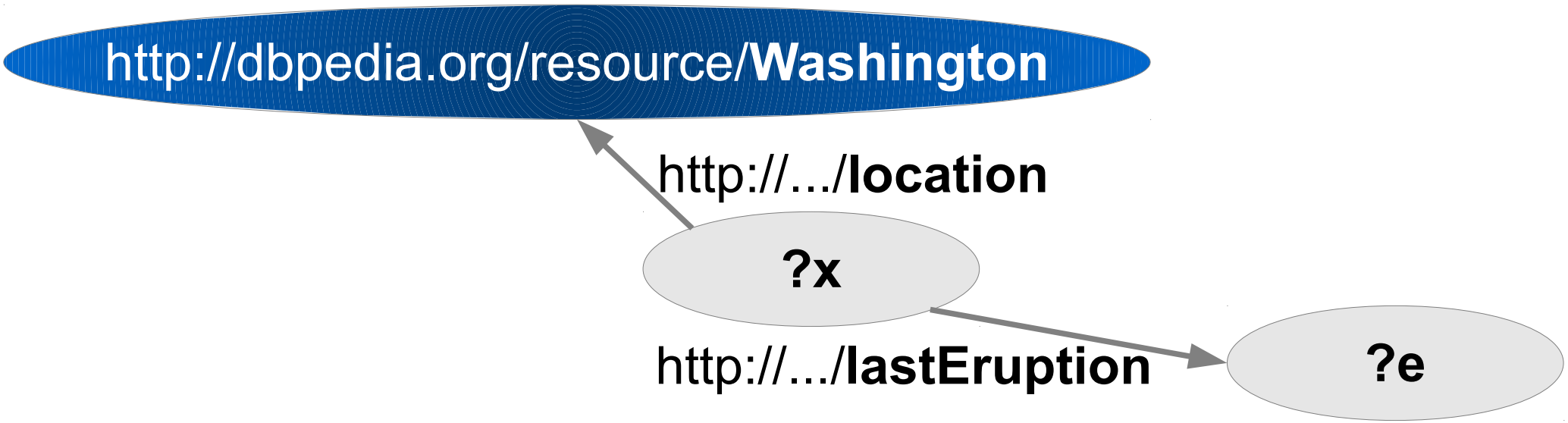
(1) The Foundation: Linked Data on the Web

- The RDF Data Model and URIs √
- The SPARQL Query Language
- The Linked Data Publishing Principles

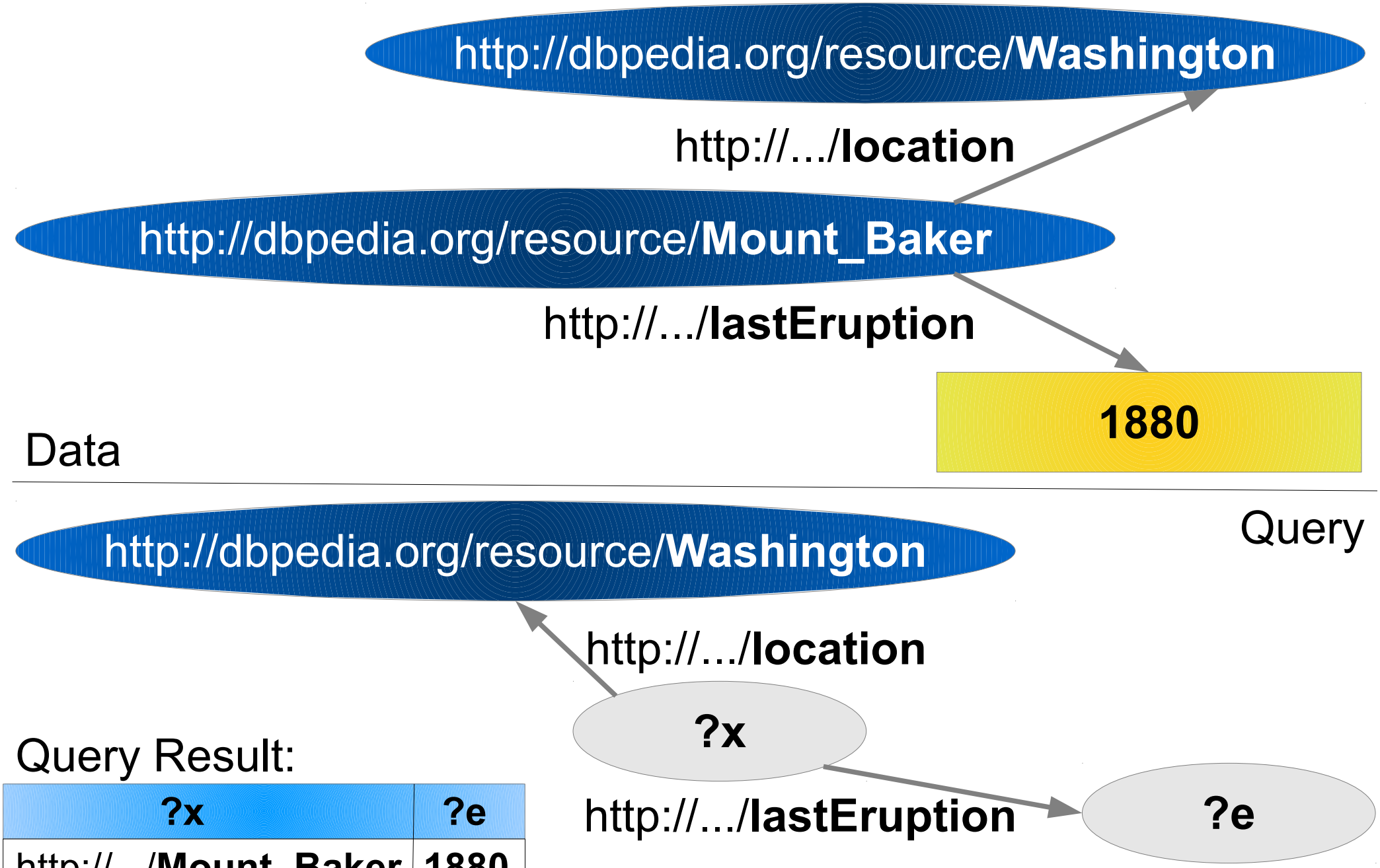
(2) Querying Linked Data

Querying RDF Data

- **SPARQL: Declarative query language for RDF data**
- **Main idea: pattern matching**
  - Describe subgraphs of the queried RDF graph
  - Subgraphs that match your description yield a result
  - Mean: graph patterns (i.e., RDF graphs with variables)



SPARQL Pattern Matching



Query Result:

?x	?e
http://.../Mount_Baker	1880

Components of a SPARQL Query

```
SELECT ?e ?name
FROM <http://example.org/myGeoData>
WHERE {
  ?x <http://.../location> <http://.../Washington> .
  ?x <http://.../lastEruption> ?e .
  OPTIONAL { ?x <http://.../name> ?name . }
}
ORDER BY ?e
```

Components of a SPARQL Query

```
SELECT ?e ?name
FROM <http://example.org/myGeoData>
WHERE {
  ?x <http://.../location> <http://.../Washington> .
  ?x <http://.../lastEruption> ?e .
  OPTIONAL { ?x <http://.../name> ?name . }
}
ORDER BY ?e
```

- **Result form specification:**
  - SELECT for projection
    (similar to projection in relational algebra)
  - Other forms: DESCRIBE, CONSTRUCT, and ASK

Components of a SPARQL Query

```
SELECT ?e ?name
FROM <http://example.org/myGeoData>
WHERE {
  ?x <http://.../location> <http://.../Washington> .
  ?x <http://.../lastEruption> ?e .
  OPTIONAL { ?x <http://.../name> ?name . }
}
ORDER BY ?e
```

- **Dataset specification:**
  - Specify the RDF dataset to be queried (use URIs that identify particular RDF graphs in your RDF database)

Components of a SPARQL Query

```
SELECT ?e ?name
FROM <http://example.org/myGeoData>
WHERE {
  ?x <http://.../location> <http://.../Washington> .
  ?x <http://.../lastEruption> ?e .
  OPTIONAL { ?x <http://.../name> ?name . }
}
ORDER BY ?e
```

- **Query Pattern:**
  - WHERE clause specifies the graph pattern to be matched
  - Expressive power equivalent to relational algebra
  - SPARQL 1.1 goes beyond (e.g., aggregation, property paths)

Components of a SPARQL Query

```
SELECT ?e ?name
FROM <http://example.org/myGeoData>
WHERE {
  ?x <http://.../location> <http://.../Washington> .
  ?x <http://.../lastEruption> ?e .
  OPTIONAL { ?x <http://.../name> ?name . }
}
ORDER BY ?e
```

- **Solution modifiers:**
  - Only for SELECT queries
  - Modify the result set as a whole (not single solutions)
  - Keywords: DISTINCT, ORDER BY, LIMIT, and OFFSET

Outline

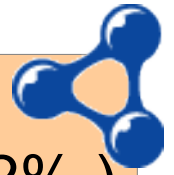
(1) The Foundation: Linked Data on the Web

- The RDF Data Model and URIs √
- The SPARQL Query Language √
- The Linked Data Publishing Principles

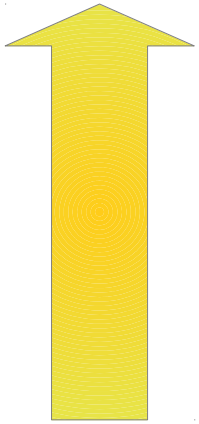
(2) Querying Linked Data

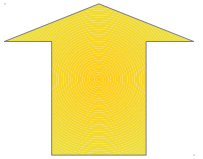
The Linked Data Principles

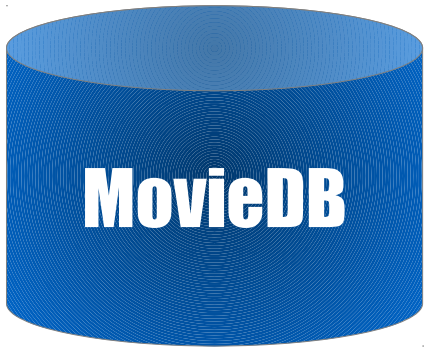
(http://...imdb.../WarChild , release date , 12 July 1999)
(http://...imdb.../WarChild , filming location , http://cia.../Albania)
(http://...imdb.../MichaelDavie , directed , http://...imdb.../WarChild)
:



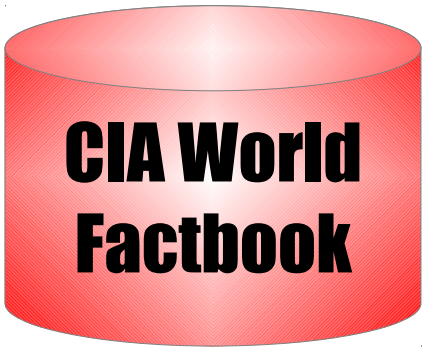
(http://cia.../Albania ,
unemployment rate , 13.2%)
:







Data model: RDF
Global identifier: URI
Access mechanism: HTTP
Connection: data links

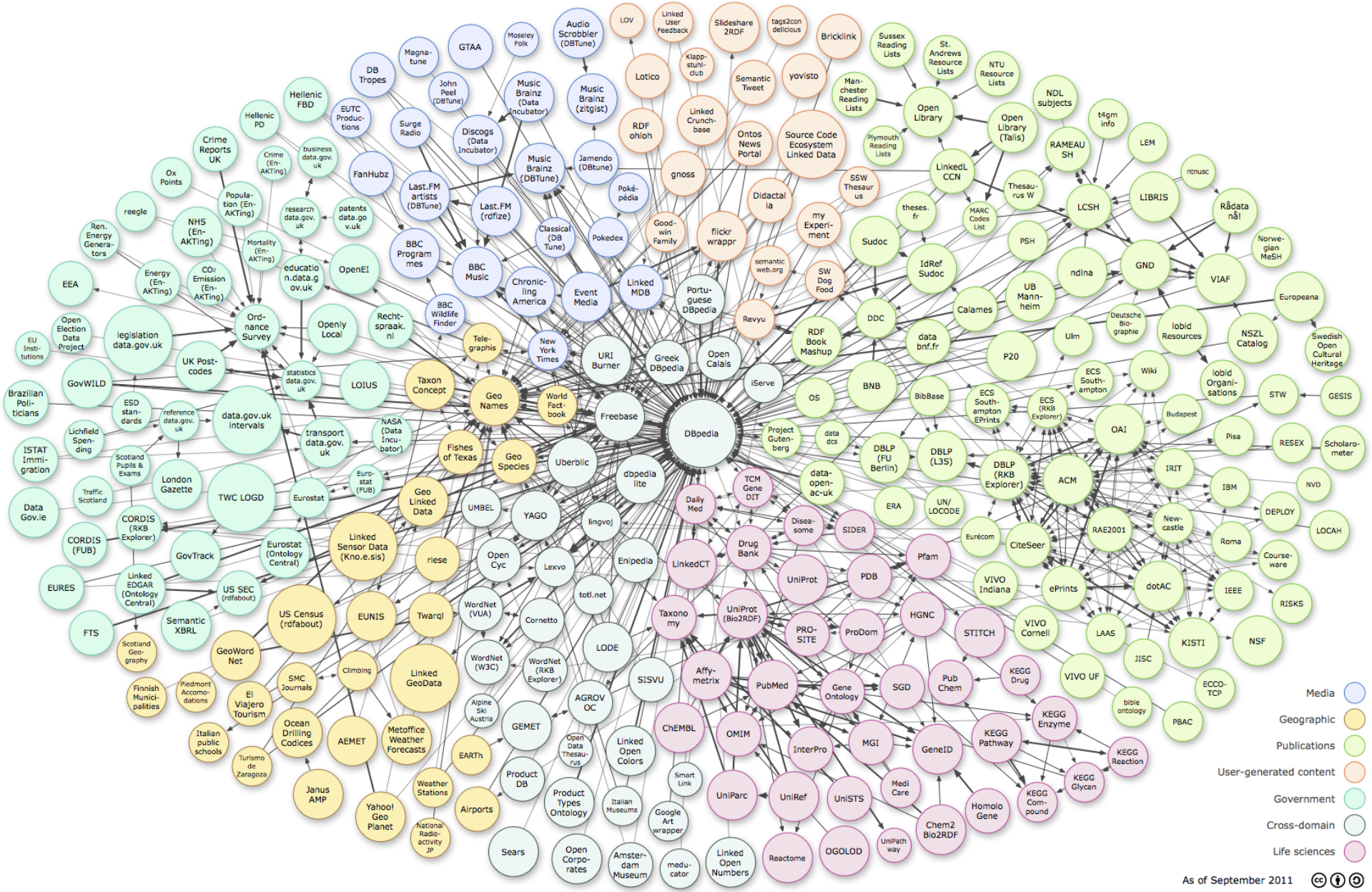


W3C Linking Open Data Project

- **Grassroots community effort**
- **Publish existing, open license datasets as Linked Data**
- **Interlink things between different data sources**

- **Prominent Linked Data publishers today:**
  - Governments (UK, US, I, etc.)
  - The Library of Congress
  - The New York Times
  - Thomson Reuters
  - Freebase (owned by Google)
  - Best Buy
  - Sears
  - CNET
  - BBC
  - etc.

W3C Linking Open Data Project



Outline

(1) The Foundation: Linked Data on the Web √

(2) Querying Linked Data

- Data Warehousing
- SPARQL Federation
- Linked Data Query Processing

Outline

(1) The Foundation: Linked Data on the Web √

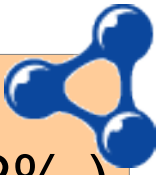
(2) Querying Linked Data

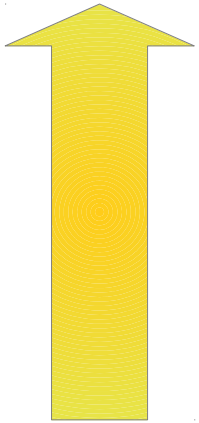
- Data Warehousing
- SPARQL Federation
- Linked Data Query Processing

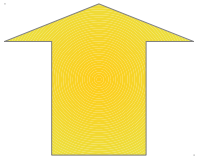
Remember ...

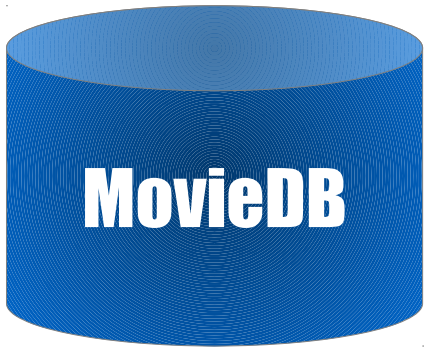
(http://...imdb.../WarChild , release date , 12 July 1999)
(http://...imdb.../WarChild , filming location , http://cia.../Albania)
(http://...imdb.../MichaelDavie , directed , http://...imdb.../WarChild)
:

(http://cia.../Albania ,
unemployment rate , 13.2%)
:

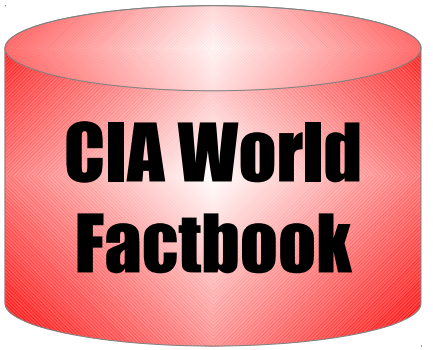






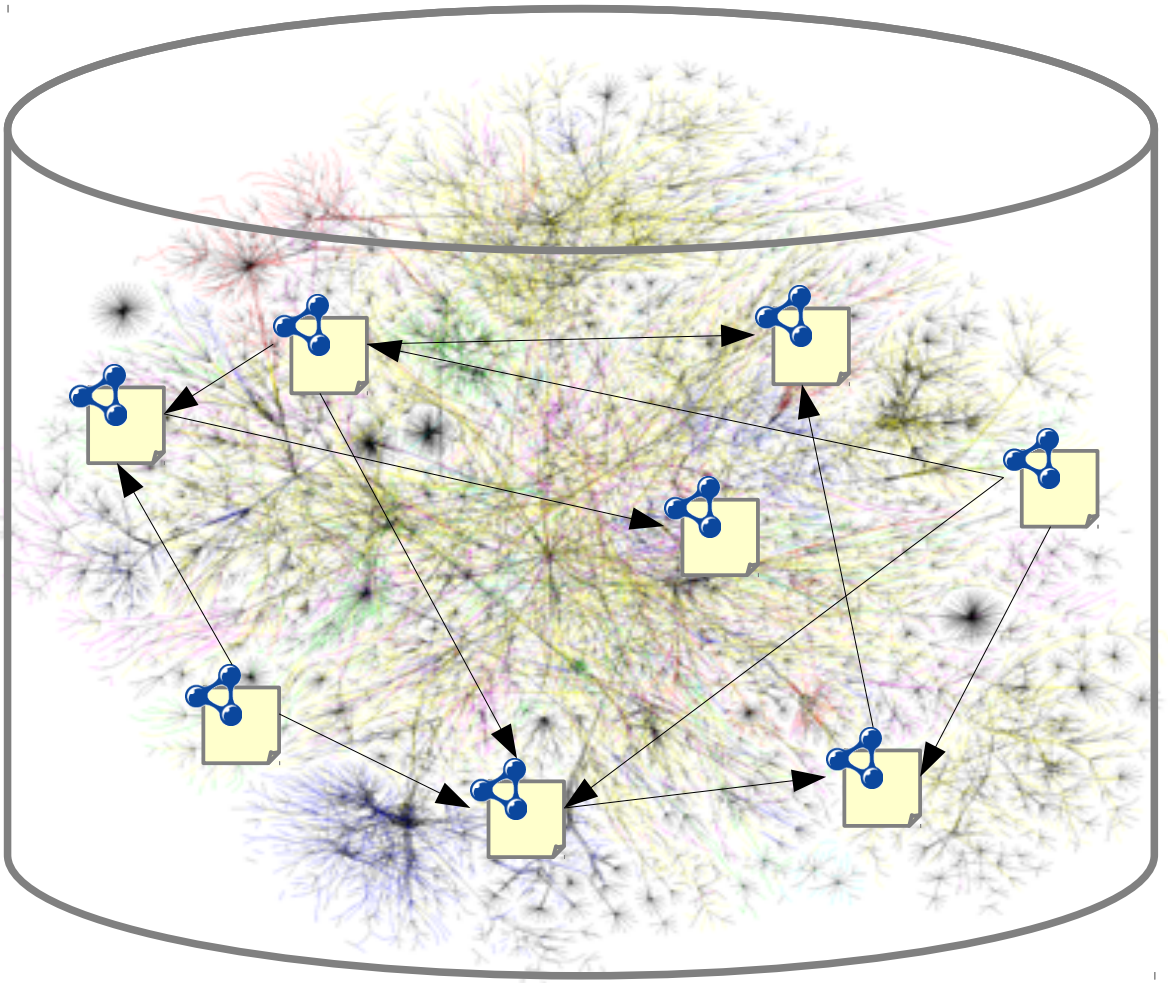


Data model: RDF
Global identifier: URI
Access mechanism: HTTP
Connection: data links



A Globally Distributed Network of Data

...which we understand as a huge distributed database



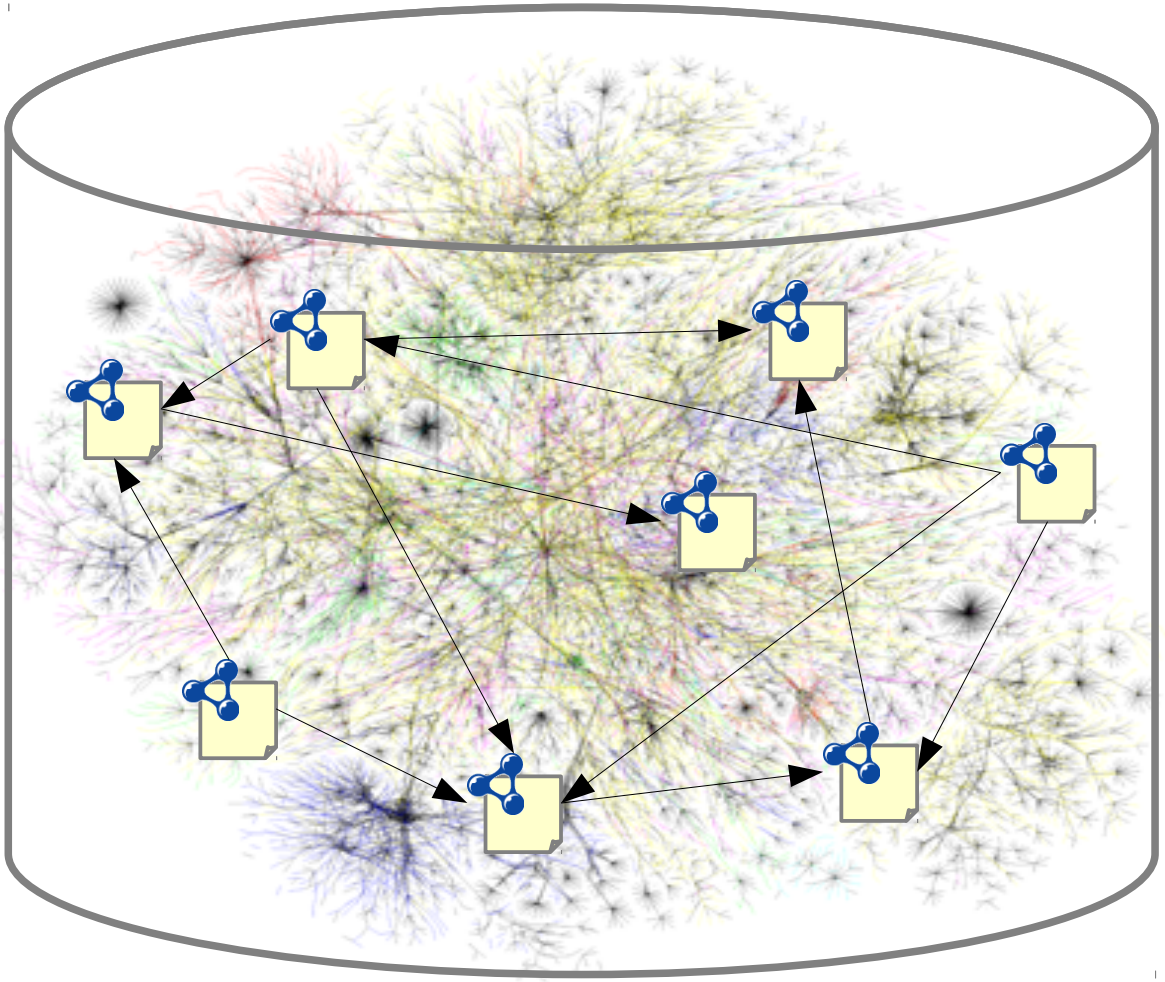
Unusual Characteristics of this DB

- **Data access capabilities of any system that aims to access this DB are inherently limited**
  - HTTP requests only
  - No queries (i.e., we cannot assume that data sources provide a query service)
- **Number of *potential* data sources is infinite**
- **It is impossible to have a DB catalog that is complete or up-to-date (or even both)**

Linked Data Query Processing

… is a new research field that focuses on querying this distributed DB

- **Criteria:**
  - On-line execution
  - Rely only on the Linked Data principles



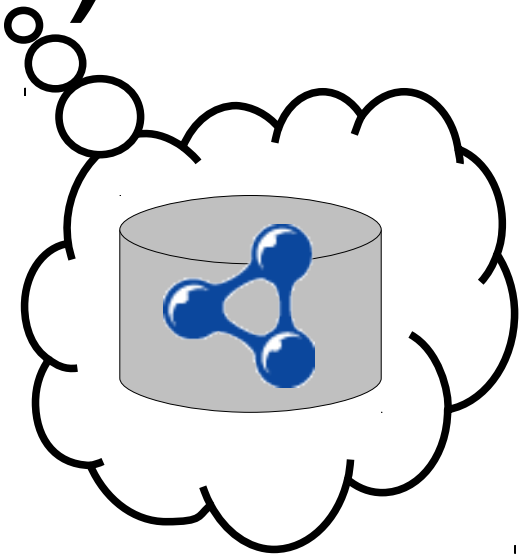
- **Use cases: live querying where freshness and discovery of results is more important than an almost instant answer**

Languages for Linked Data Queries

- **Navigational query languages**
  - Regular expressions to specify paths of data links
  - Query result: end nodes of matching paths
  - NautiLOD, LDPath

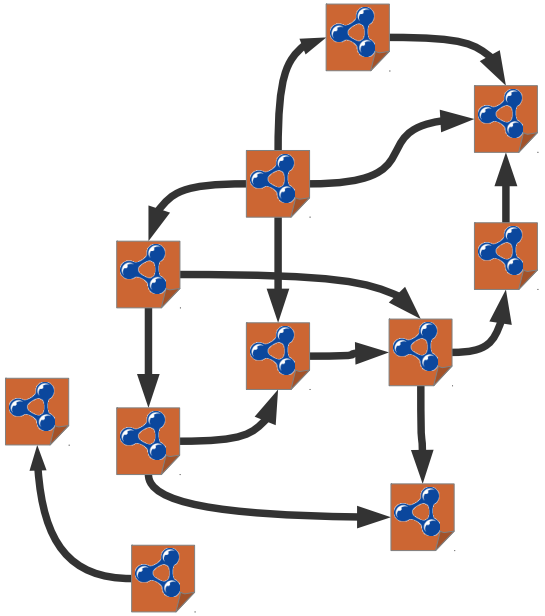
$$Q^{expr}(D) = ?$$

- **SPARQL**
  - Seems to be a natural choice
  - However, standard definition captures queries over a predefined dataset (e.g., stored in an RDF DBMS)



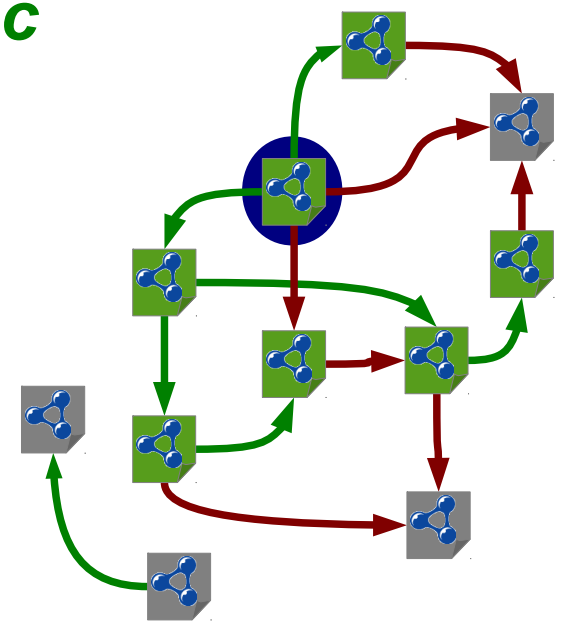
Semantics for SPARQL LD Queries

- **Full-Web query semantics**
  - Scope of evaluating a SPARQL expression is all Linked Data
  - Query result completeness cannot be guaranteed by any (terminating) execution



Semantics for SPARQL LD Queries

- **Full-Web query semantics**
  - Scope of evaluating a SPARQL expression is all Linked Data
  - Query result completeness cannot be guaranteed by any (terminating) execution
- **Reachability-based query semantics**
  - Query consists of a SPARQL expression, a set of seed URIs ***S***, and a reachability condition ***c***
  - Scope: all data along paths of data links that satisfy the condition
  - Computationally feasible



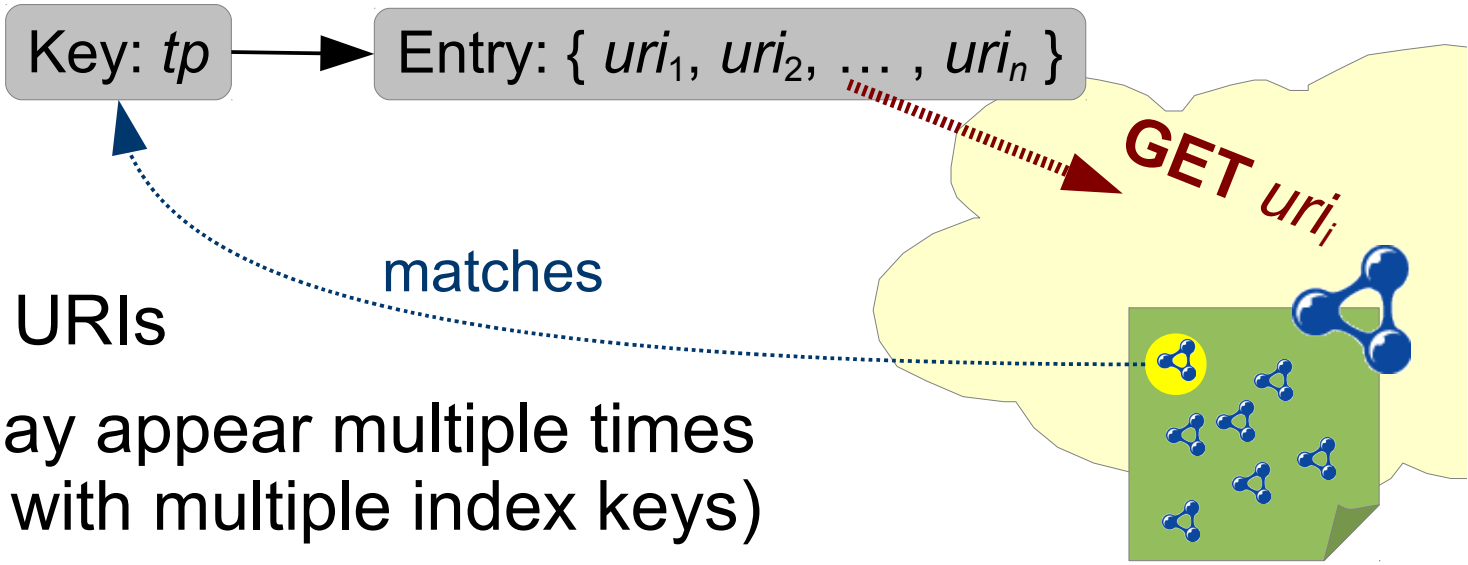
Query Execution

- **Two processes that may be intertwined:**
  - Fetching Linked Data by looking up URIs
  - Constructing the query result

- **Classes of approaches:**
  - Index-based approaches
  - Traversal-based approaches
  - Hybrids

Index-Based Source Selection

- **Idea:** Use pre-populated index to determine relevant URIs and to avoid as many irrelevant ones as possible)
- **Index keys:**
  - Different approaches possible
  - e.g., triple patterns



- **Index entries:**
  - Usually, a set of URIs
  - Indexed URIs may appear multiple times (i.e., associated with multiple index keys)
  - Each URI in such an entry may be paired with a cardinality (utilized for source ranking)

Traversal-Based Query Execution

- **Idea:** Discover relevant URIs recursively by traversing (specific) data links at query execution runtime
  - Natural support of reachability-based query semantics

- **Retrieved data serves two purposes:**

  (1) Discover further URIs

  (2) Construct query result

Traversal-Based – vs. – Index-Based

- **Possibilities for parallelized data retrieval are limited**
  - Data retrieval adds to query execution time significantly
- **Usable immediately**
  - Most suitable for “on-demand” querying scenario
- **Depends on the structure of the network of data links**

- **Data retrieval can be fully parallelized**
  - Reduces the impact of data retrieval on query exec. time
- **Usable only after initialization phase**
- **Depends on what has been selected for the index**
- **May miss new data sources**

None of both strategies is superior over the other w.r.t. result completeness (under full-Web query semantics).

- Both strategies may miss (different) solutions for a query

Summary

- **RDF**
  - Triple-based data model
- **SPARQL**
  - Declarative query language for RDF data
  - Main idea: pattern matching
- **Linked Data**
  - Structured, *interlinked* data on the Web
- **Querying Linked Data**
  - Data warehousing
  - SPARQL federation
  - Linked Data query processing (index-based, traversal-based)