www.semantic-web-grundlagen.de

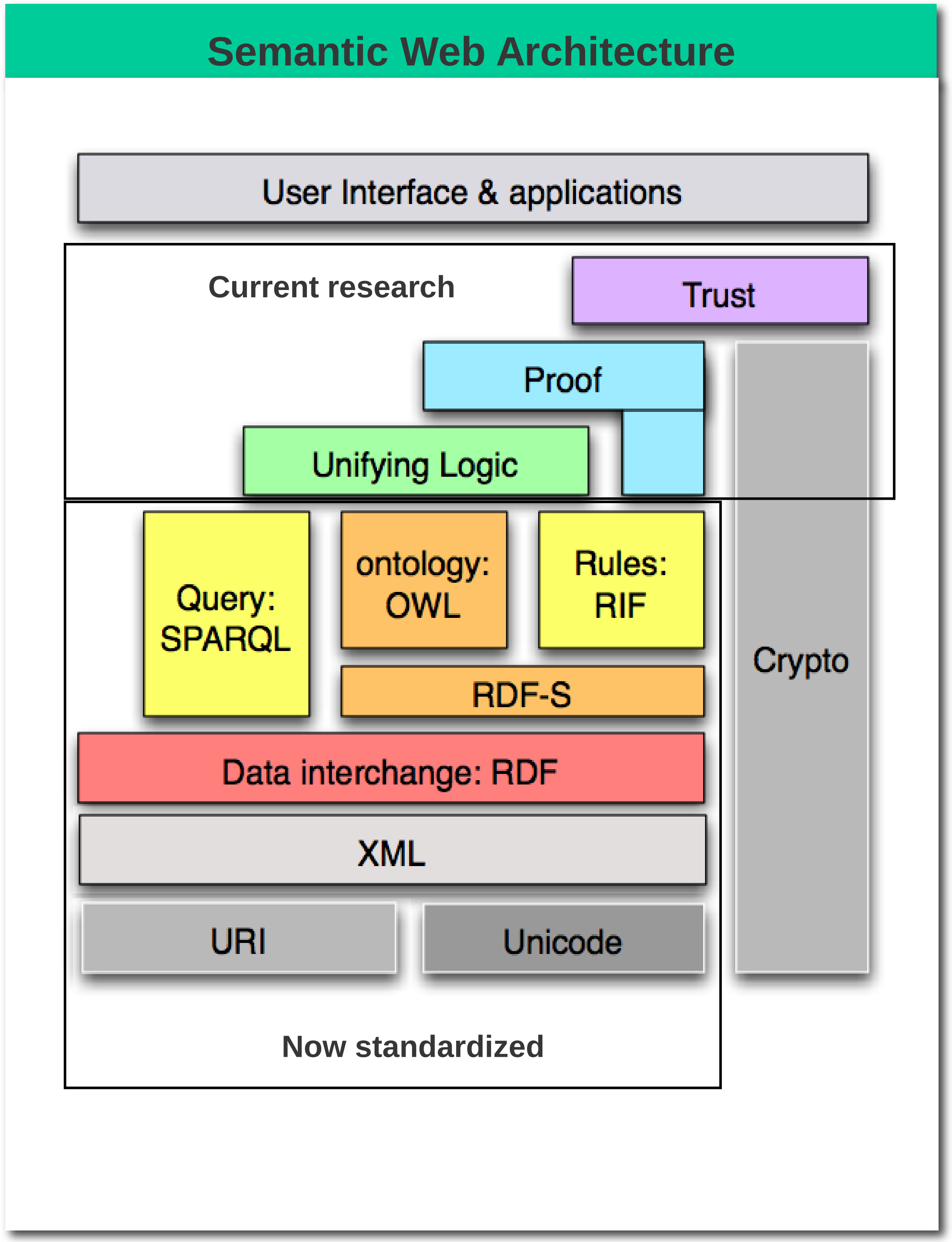
Semantic Web Technologies I

Lehrveranstaltung im WS12/13

PD Dr. Sebastian Rudolph
Dr. Duc Thanh Tran
M.Sc. Anees ul Mehdi

Linked Data

Dr. Duc Thanh Tran



Agenda

- Motivation
- Linked Data Principles
- Linked Data Access

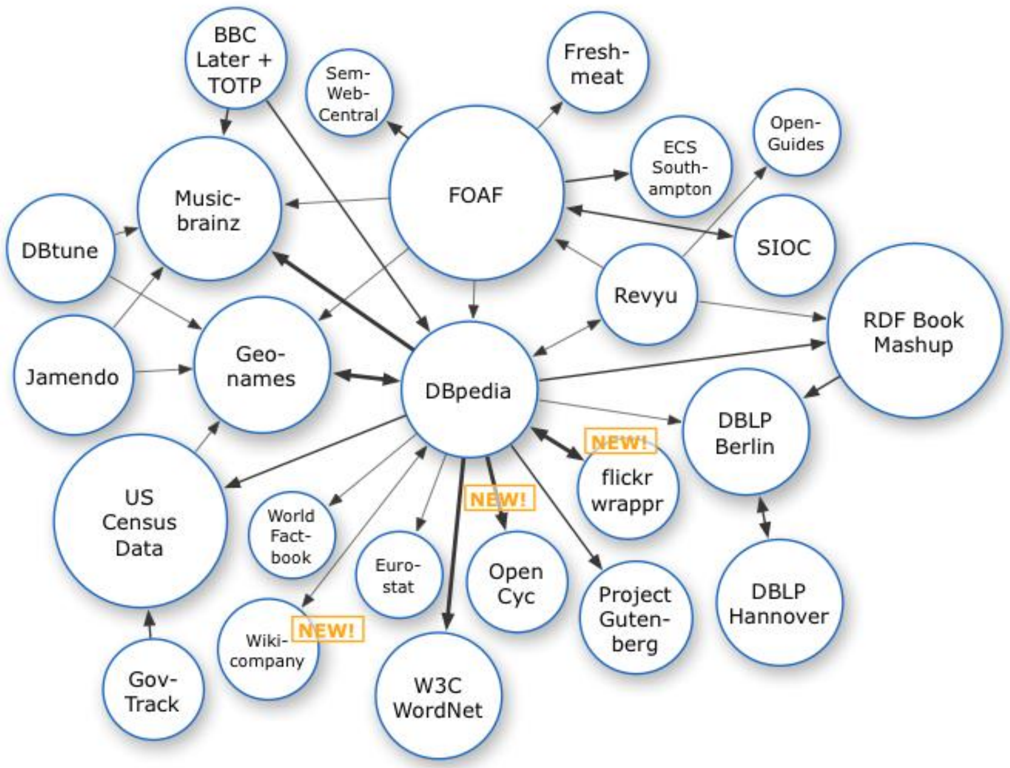
MOTIVATION

Motivation

- Data volumes explode
  - More and more **data available on the Web is represented in Semantic Web standards**
  - Linking Open Data (LOD) initiative
- Semantic Web standards & technologies facilitate
  - Representation, exchange
  - Integration & linking of data from multiple sources
  - Consumption

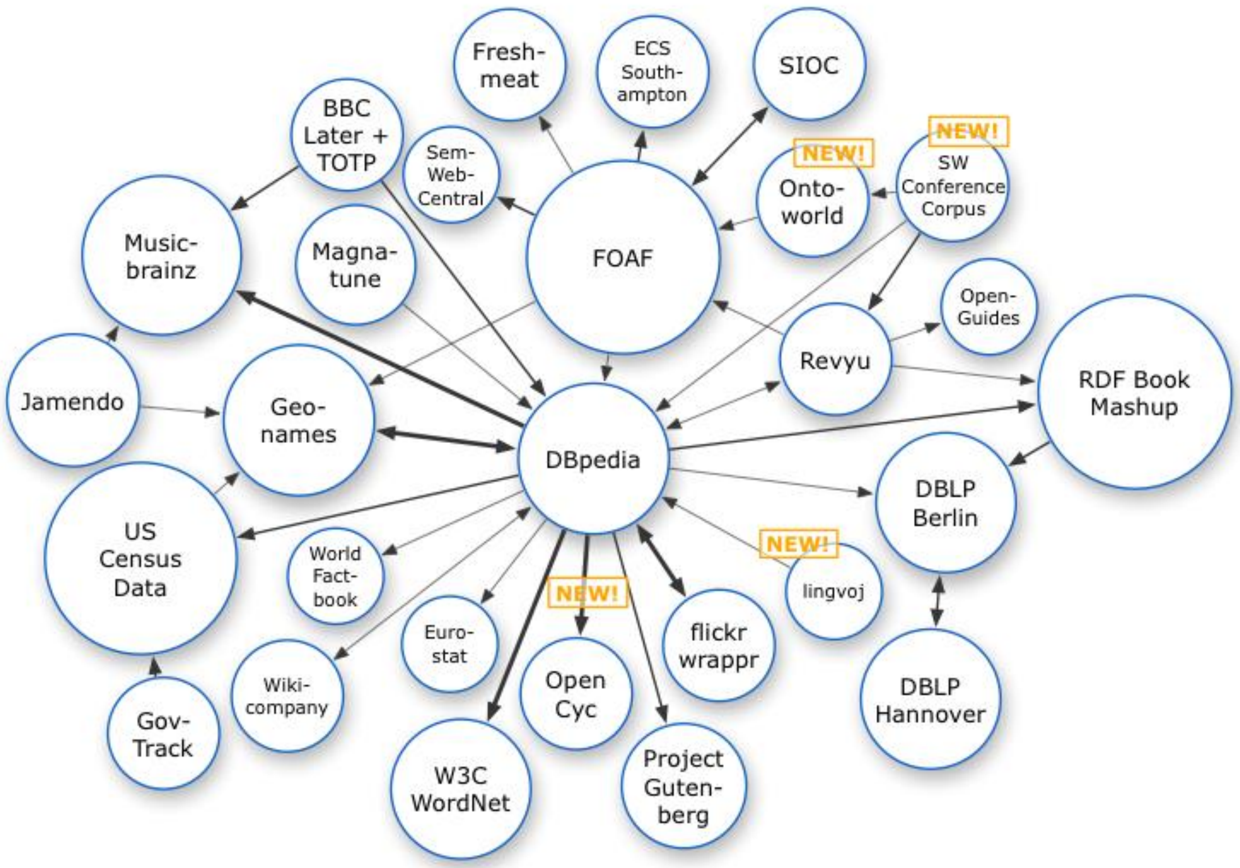
Linked Data on the Web

AIFB



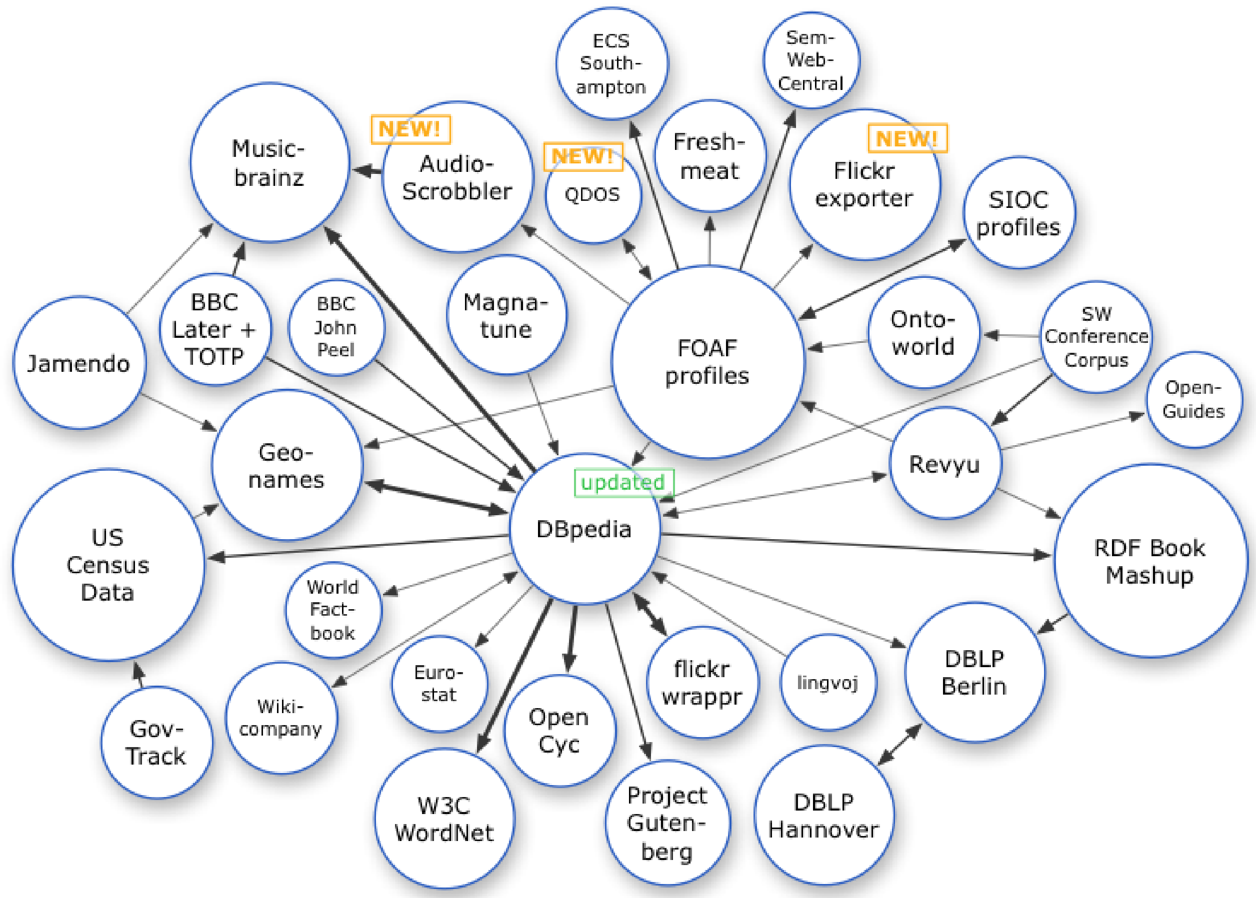
2007-10

Linked Data on the Web



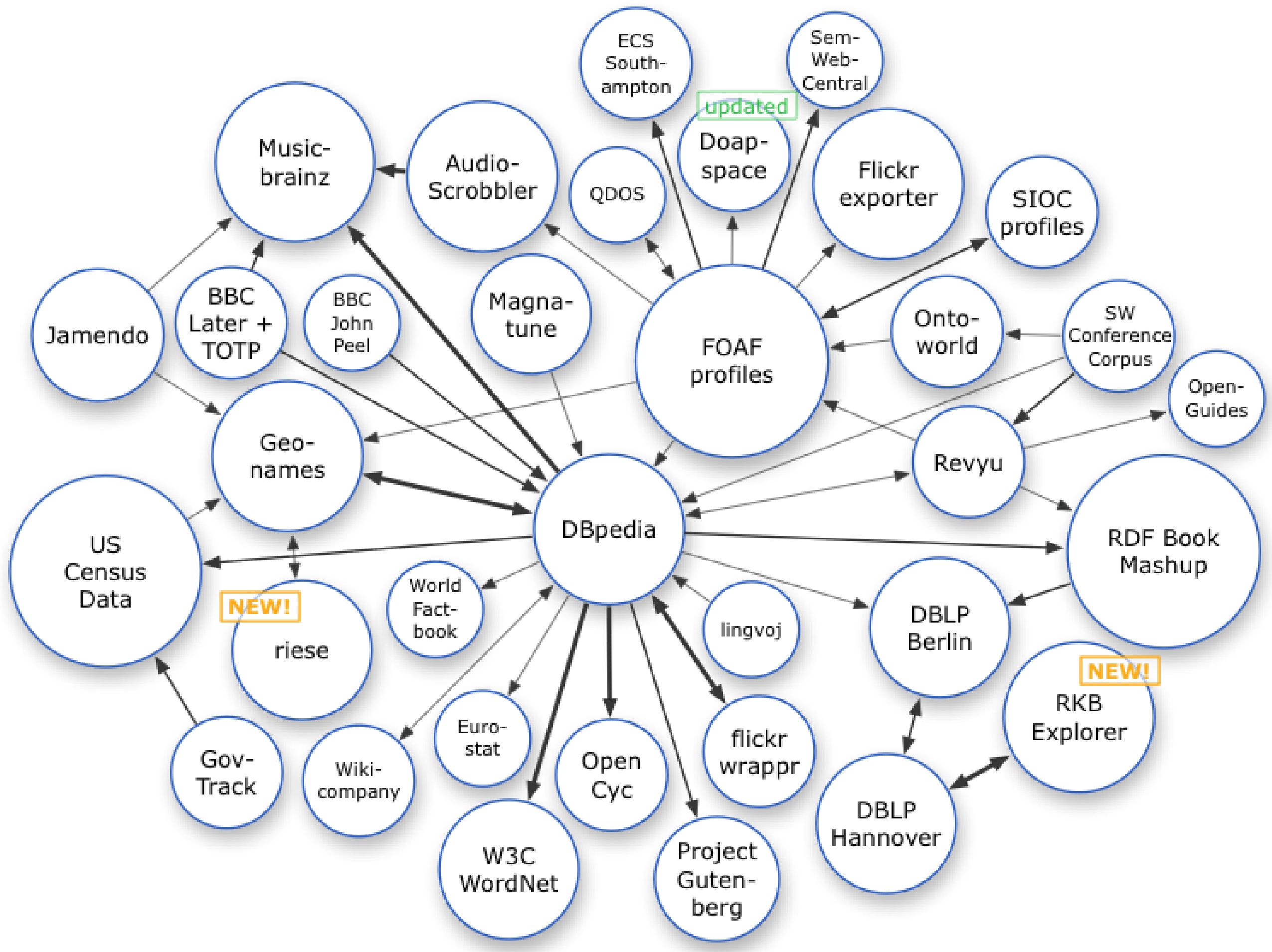
2007-11

Linked Data on the Web



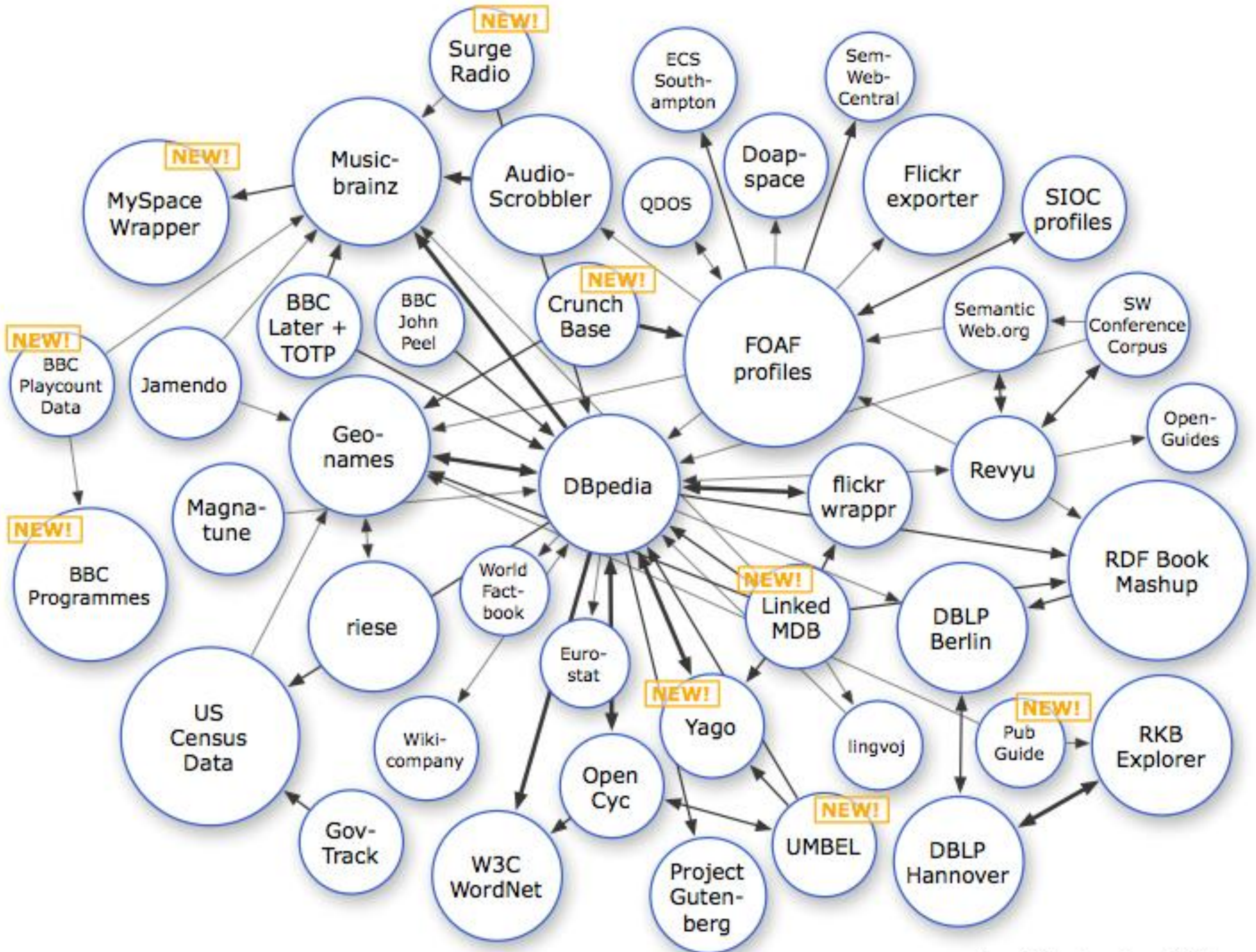
2008-02

Linked Data on the Web



2008-03

Linked Data on the Web

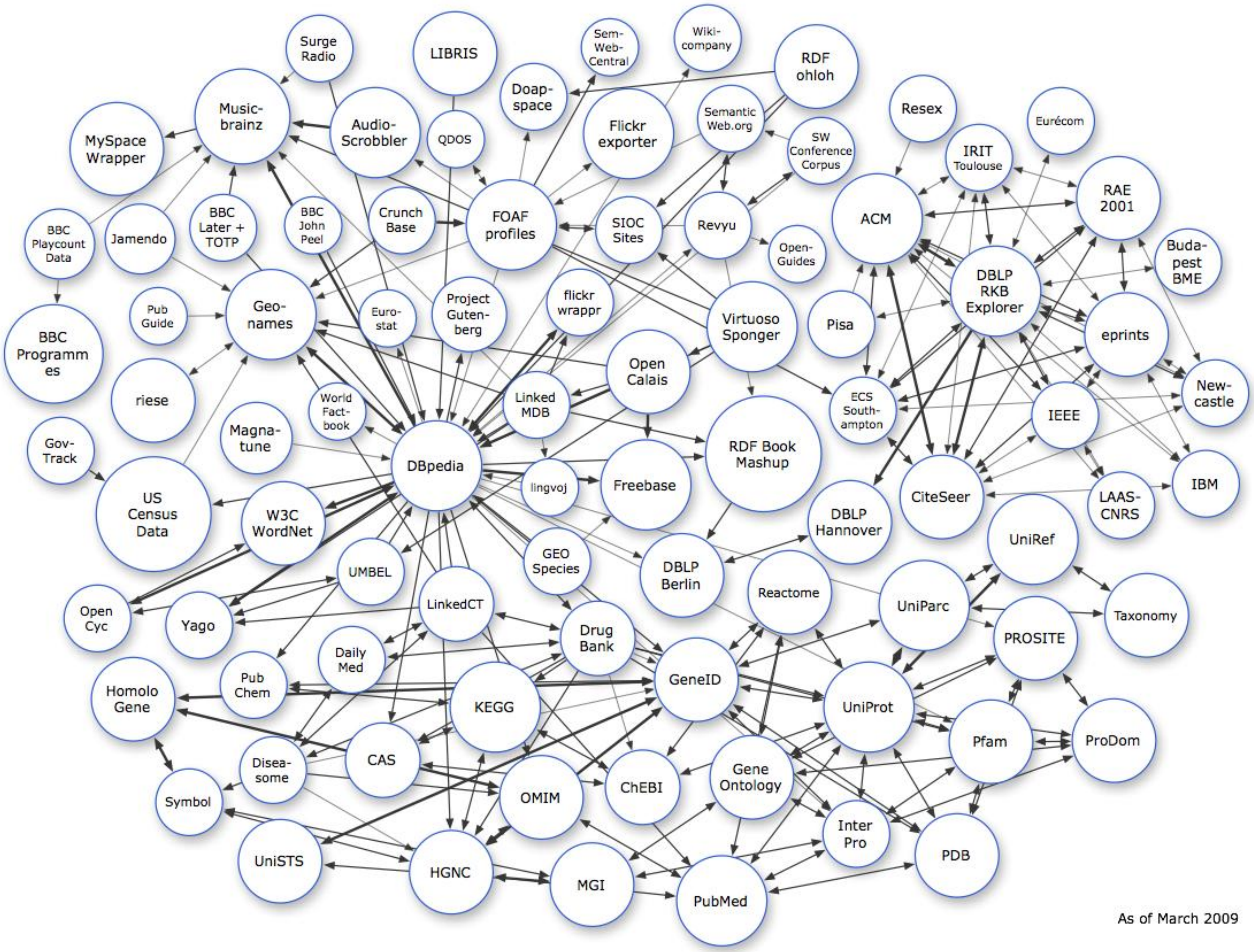


2008-09

Semantic Web TECHNOLOGIES

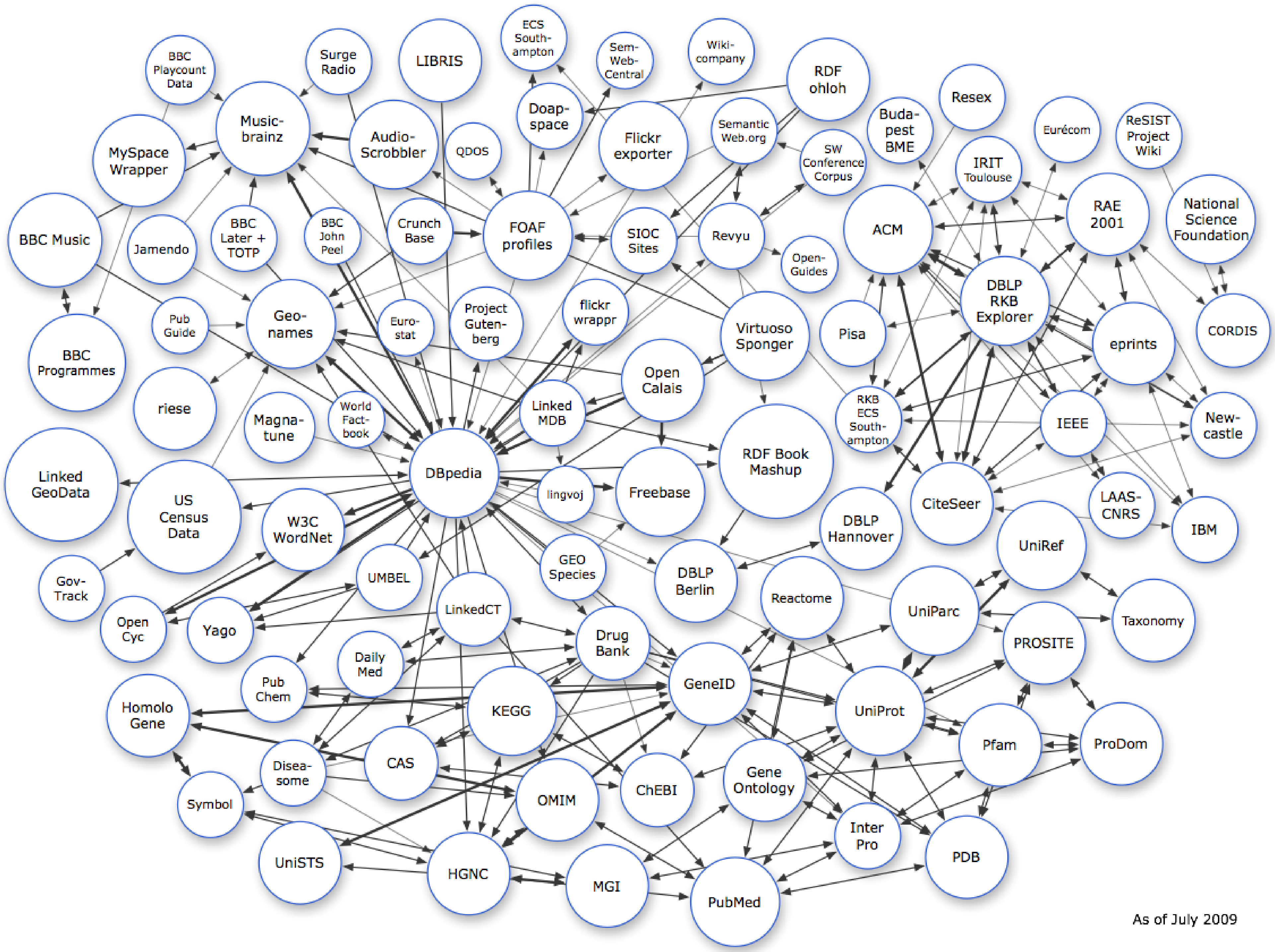
AIFB

Linked Data on the Web



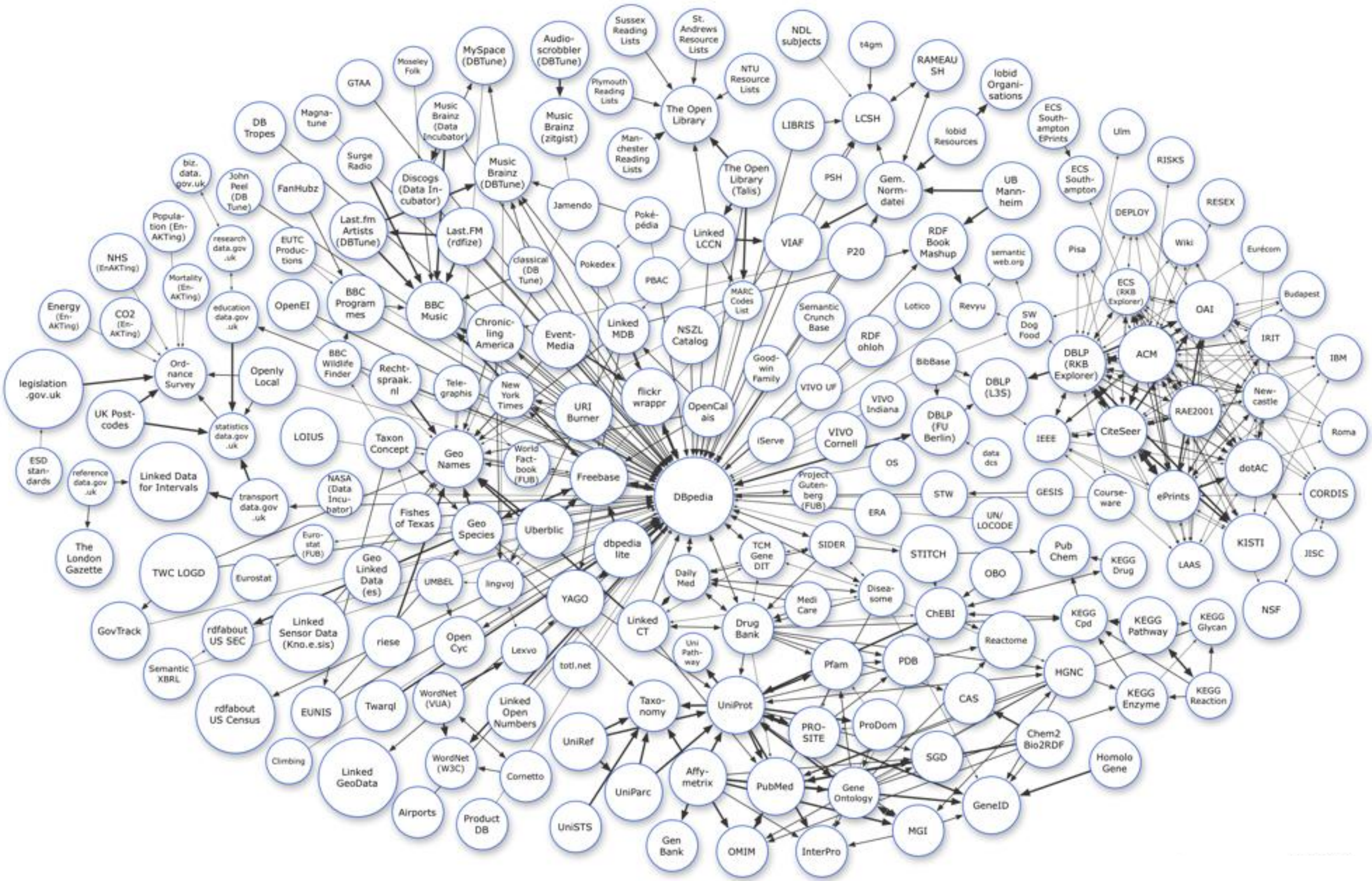
2009-03

Linked Data on the Web



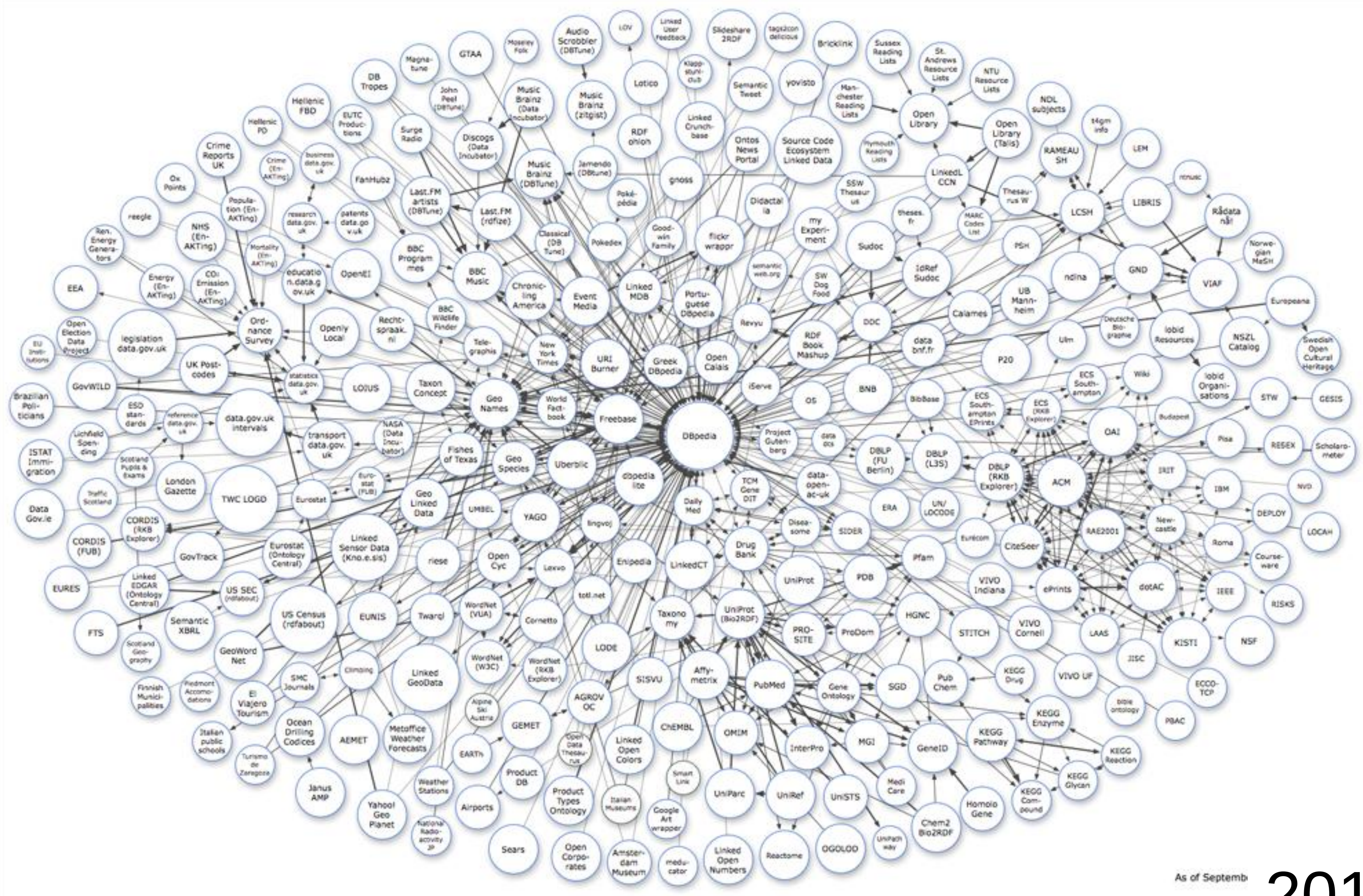
2009-07

Linked Data on the Web



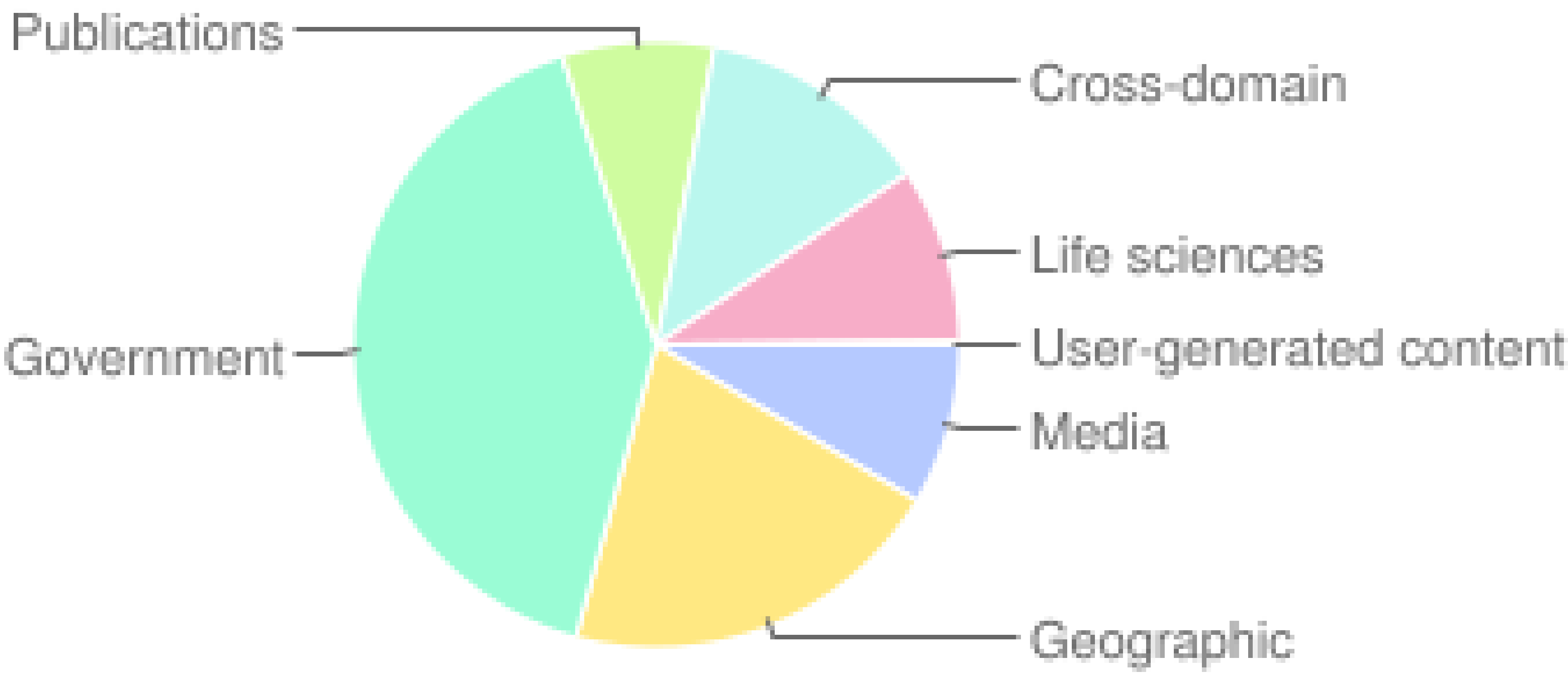
2010-09

Linked Data on the Web



2011-09

Types of Data in the Linking Open Data Cloud

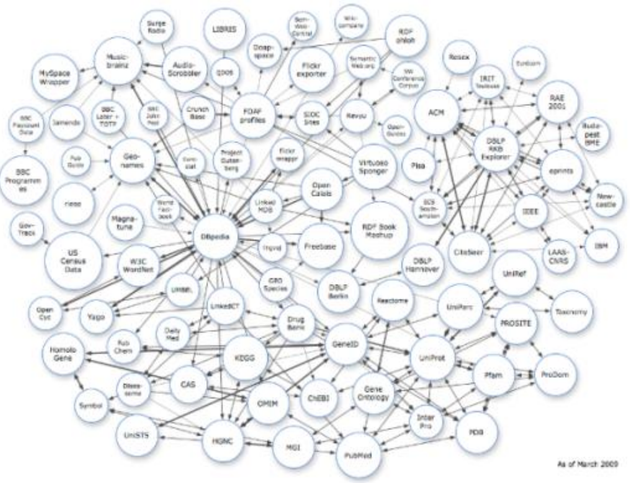


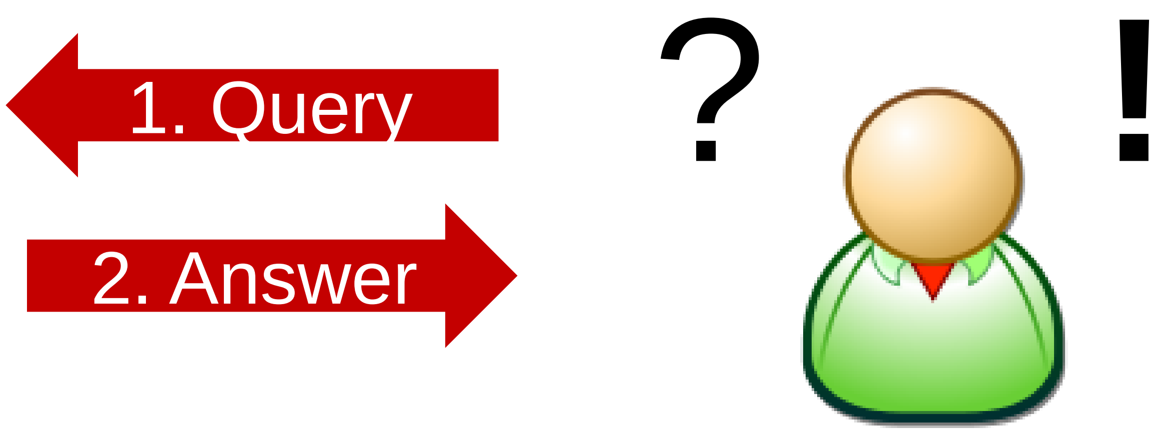
http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (Sept 2010)

Semantic Web TECHNOLOGIES

Consumption
A Search Scenario

AIFB



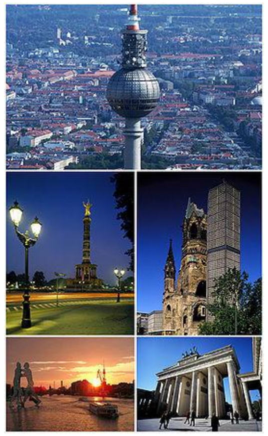


Semantic Technologies facilitate access to data

- Q: data about Berlin?
- Q: famous people that died in Berlin?
- Q: data about Hegel?
- Q: Hegel's publications?
- Q: data about Marlene Dietrich?
- Q: Dietrich's songs?

DBpedia

- Linked Data version of Wikipedia
- Scripts that extract data (text, links, infoboxes) from Wikipedia
- Published as Linked Data
- Interlinking hub in the Linked Data web



- Berlin
  - http://dbpedia.org/resource/Berlin
- Hegel
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- Marlene Dietrich
  - http://dbpedia.org/resource/Marlene_Dietrich

BBC Music

- Data about BBC (radio) programmes, artists, songs…
- Combination of BBC-internal data (playlists), MusicBrainz (artists, albums), Wikipedia (artists)
- Underpinning the BBC Music website
- Data published according to Linked Data principles

- Marlene Dietrich
  - http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5

Virtual International Authority File (VIAF)

- Joint project of national libraries and related organisations
  - 21 institutions, among them the Deutsche Nationalbibliothek
- Provide access to “authority files”
- Matching and interlinking collections from participating institutions

- Hegel
  - http://viaf.org/viaf/89774942

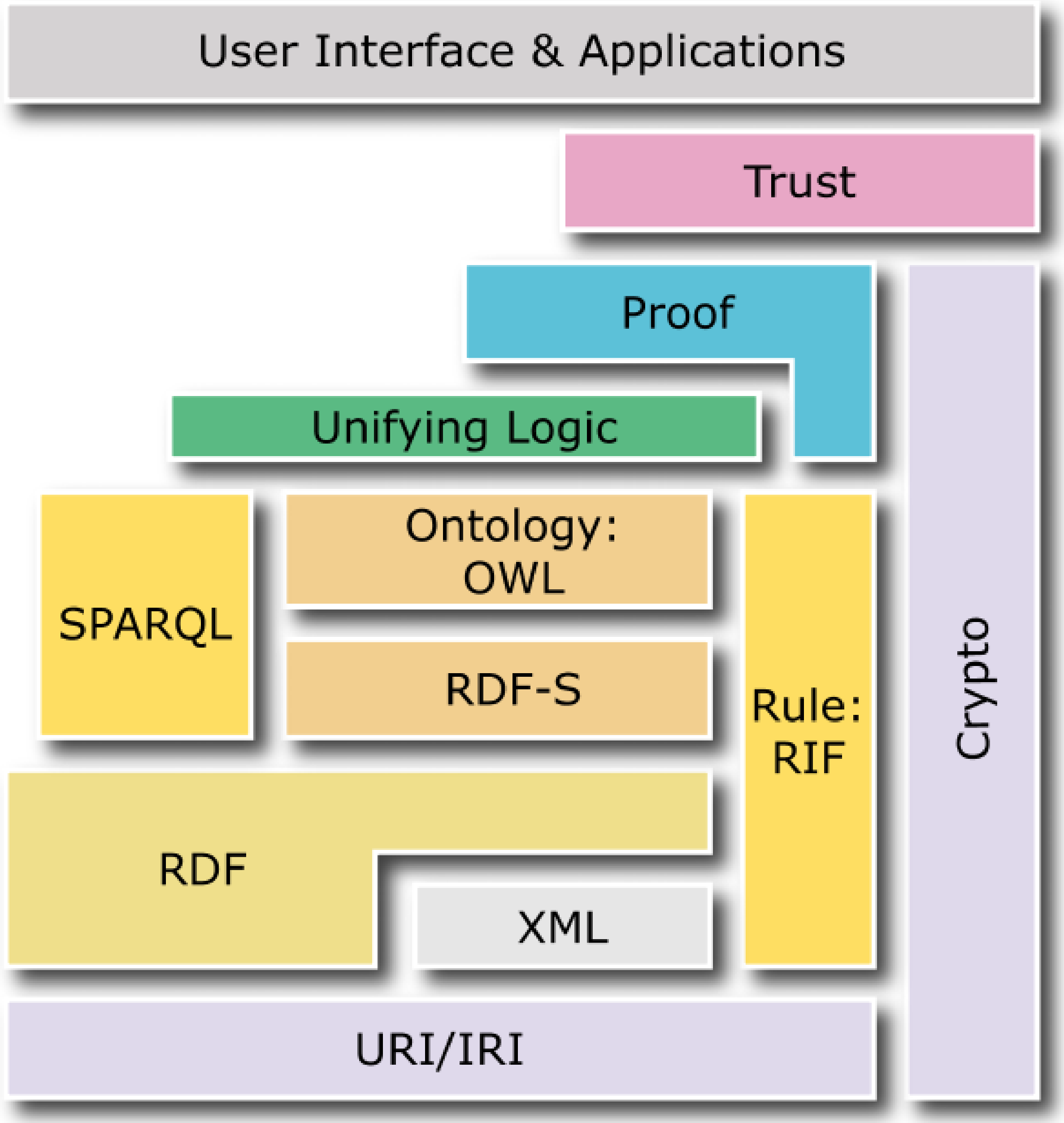
- Marlene Dietrich
  - http://viaf.org/viaf/97773925

LINKED DATA PRINCIPLES

Semantic Technologies

AIFB

- Semantic Web technologies, standardised by the W3C, are mature:
  - RDF recommendation in 1999, update in 2004
  - RDFa (RDF in HTML) note in 2008
  - RDFS recommendation in 2004
  - SPARQL recommendation in 2008
  - OWL recommendation in 2004, update in 2009
- Linked Data is a subset of the Semantic Web stack, including web architecture:
  - HTTP
  - IRI / URI
  - (RDF / XML)



Linked Data Principles

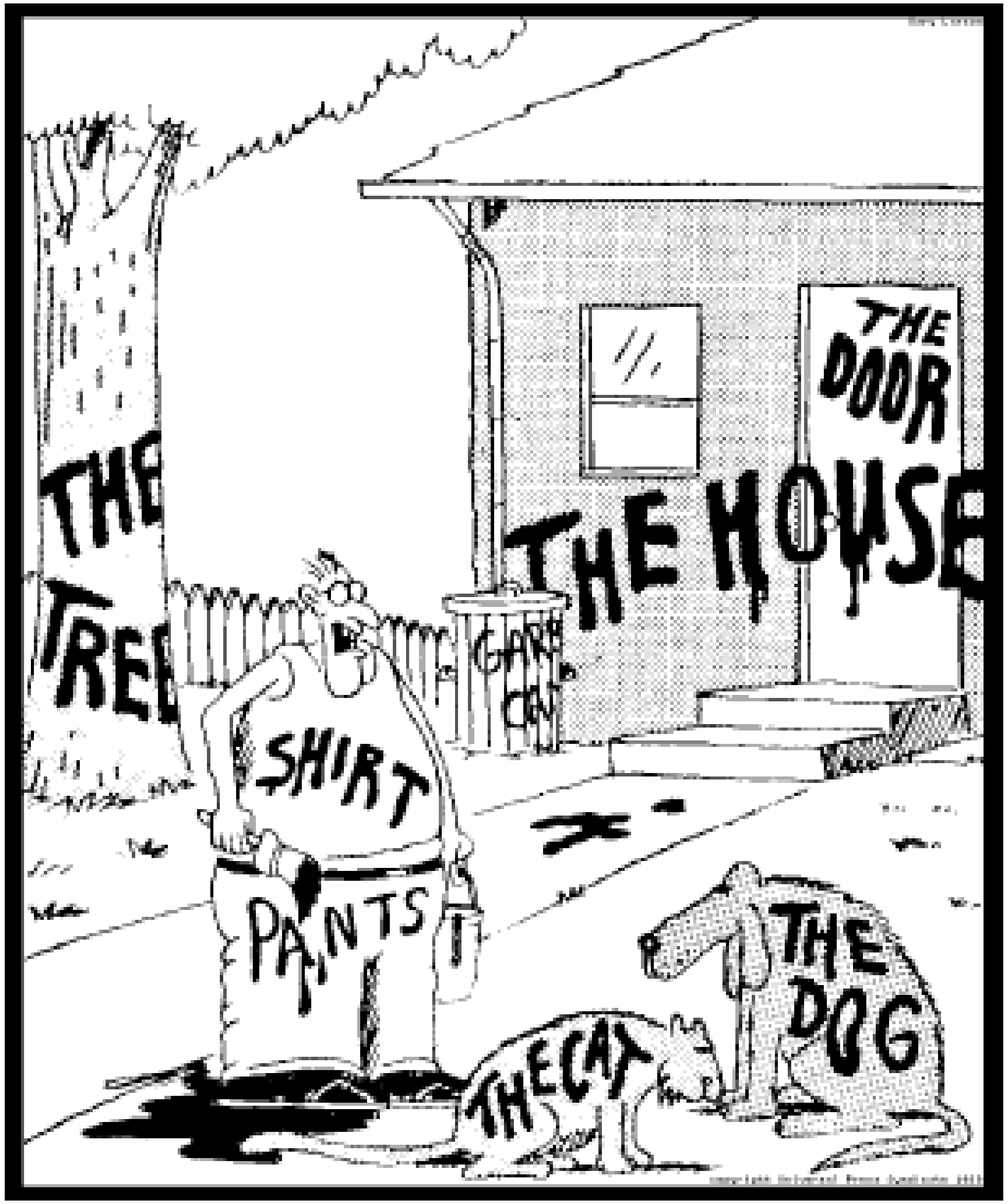
1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData

1. Use URIs as Names for Things

- Use a unique identifier (URIs) to denote things
- Hegel, Georg Wilhelm Friedrich
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
  - http://viaf.org/viaf/89774942
  - …

Names for Things



“Now! *That* should clear up a few things around here!”

2. Use HTTP URIs

- Enables “lookup” of URIs
- Via Hypertext Transfer Protocol (HTTP)
- Piggy-backs on hierarchical Domain Name System to guarantee uniqueness of identifiers
- Uses established HTTP infrastructure
- Connects logical level (thing) with physical level (source)
- Important: distinction between name/“thing URI” and location/“source URI” („other resource“/„non-information resource“ vs. „information resource“)

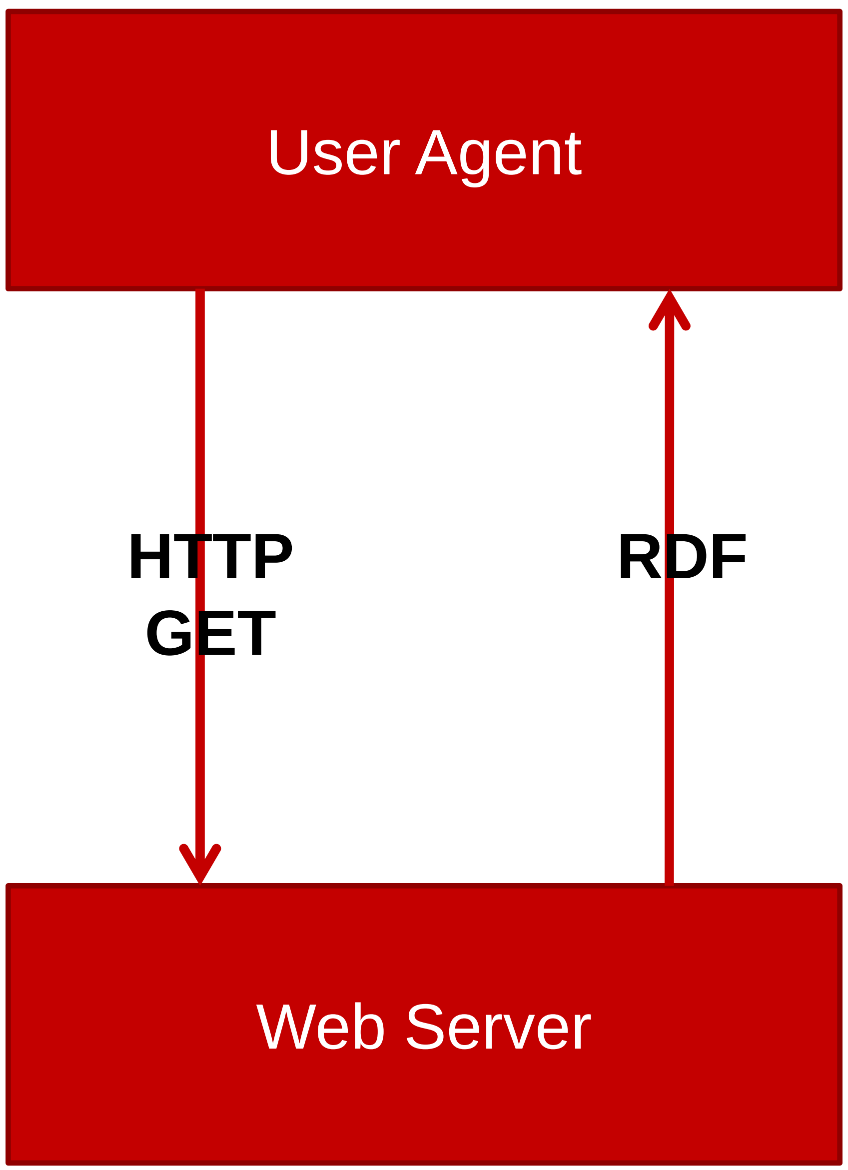
Information Resources vs. Other Resources

Marlene Dietrich, the person

File containing data about
Marlene Dietrich

Name?
Creator?
Birth date?
Last change date?
License?
Copyright?
…

Correspondence between thing-URI and source-URI



http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5#artist

RDF

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf

3. Provide Useful Information

- When somebody looks up a URI, return data using the standards (RDF*, SPARQL)
- Resource Description Framework, a format for encoding graph-structured data (with URIs to identify nodes/vertices and links/edges)

Resource Description Framework

- Directed, labeled graph
- `triple(subject, predicate, object)`
  - subject: URI (or blank node)
  - predicate: URI
  - object: URI (or blank node) or RDF literal (string, integer, date…)

- RDF/XML is the most widely deployed serialisation
- Other serialisations possible (N-Triples, Turtle, Notation3…)

4. Link to Other URIs

- Enable people (and machines) to jump from server to server
- External links vs. internal links (for any predicate)

- Using external vocabularies enables linking
- Vocabularies might be interlinked, too

- Special owl:sameAs links to denote equivalence of identifiers (useful for data merging)

Semantic Web TECHNOLOGIES

Equivalences via owl:sameAs

AIFB

http://viaf.org/viaf/89774942

- http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- http://www.idref.fr/026917467/id
- http://libris.kb.se/resource/auth/190350
- http://d-nb.info/gnd/118547739

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5#artist

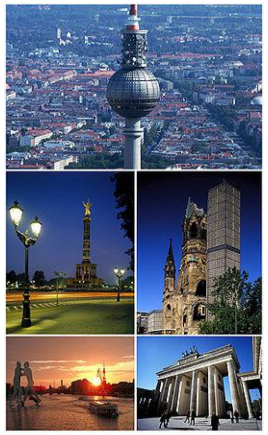
- http://dbpedia.org/resource/Marlene_Dietrich

http://viaf.org/viaf/97773925

- http://dbpedia.org/resource/Marlene_Dietrich
- http://d-nb.info/gnd/118525565
- http://libris.kb.se/resource/auth/238817
- http://www.idref.fr/027561844/id

http://dbpedia.org/resource/Berlin

- http://mpii.de/yago/resource/Berlin
- http://data.nytimes.com/N50987186835223032381 - Berlin (Germany)
- http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Berlin
- http://data.nytimes.com/16057429728088573361 - Gaspe Peninsula (Quebec) (?)



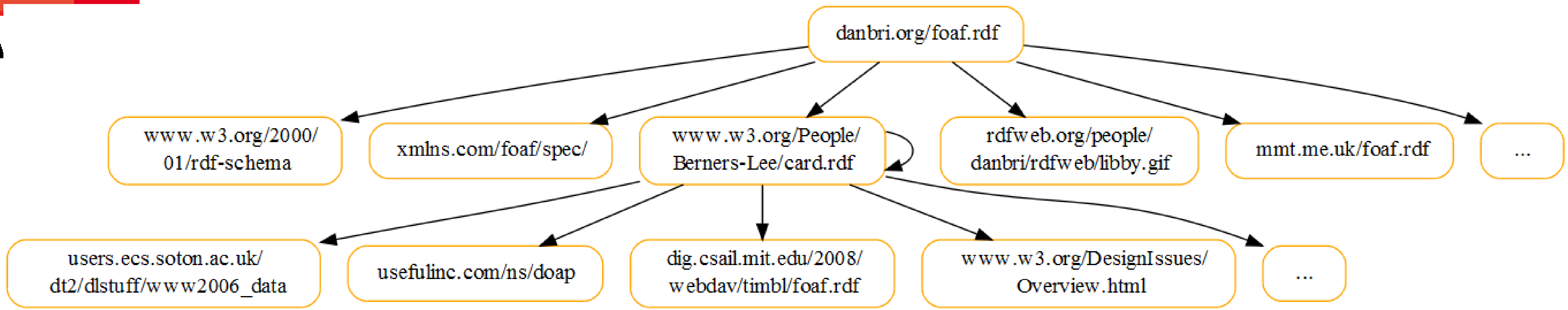
Benefits of Linked Data

- Explicit, simple data representation
  - Common data representation (Resource Description Framework, RDF) hides underlying technologies and systems
- Distributed System
  - Decentralised distributed ownership and control facilitates adoption and scalability
- Loose coupling with common language layer
  - Large scale systems require loose coupling, via HTTP as common access protocol
- Ease of publishing and consumption
  - Simple and easy-to-use systems and technologies to facilitate uptake
- Cross-referencing
  - Allows for linking and referencing of existing data, via reuse of URIs
- Incremental data integration
  - Start with merged RDF graphs and provide mappings as you go

Challenges

- Ramp-up cost for data conversion
  - May be alleviated by semi-automatic mappings and adequate tool support for manual conversion
- Integrated data may be messy at first
  - But can be refined as need arises
- Distributed creation and loose coordination may result in inconsistencies
  - Can be detected, diagnosed, and fixed with appropriate tools
- Not only quality but also volume is a problem

Resource Graph Explosion



Directed graph rooted in http://danbri.org/foaf.rdf

Level 0: 1

Level 1: 25^1 (avg), 105^1 (worst)

Level 2: 25^2 (avg), 105^2 (worst)

Level 3: 25^3 (avg), 105^3 (worst)

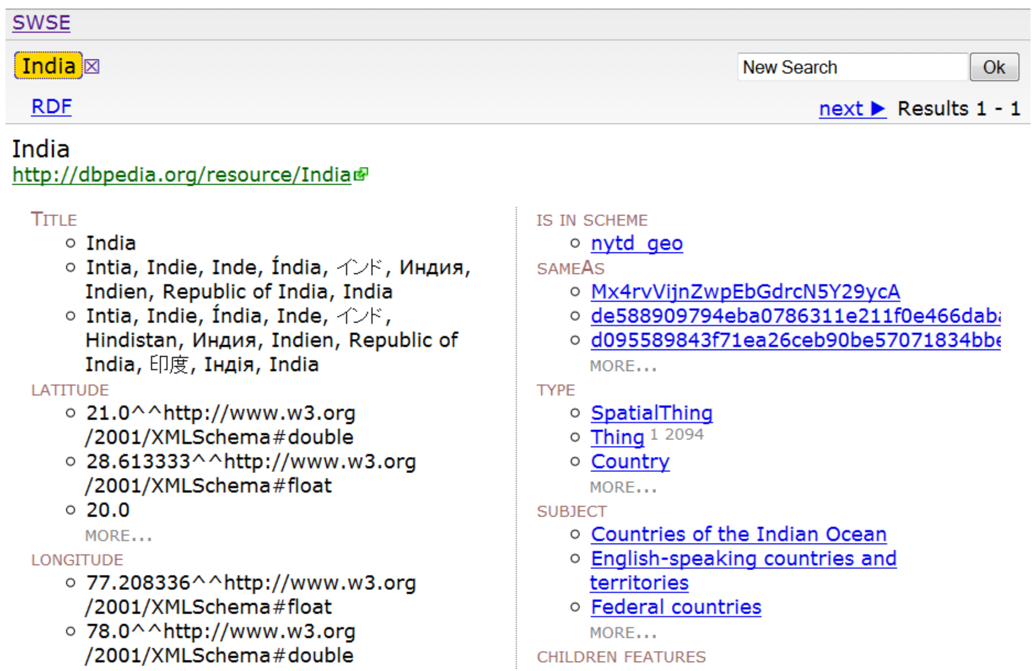
Level k: n^k

ACCESSING LINKED DATA

- APPLICATIONS
- CRAWLING
- INDEXING
- QUERY PROCESSING

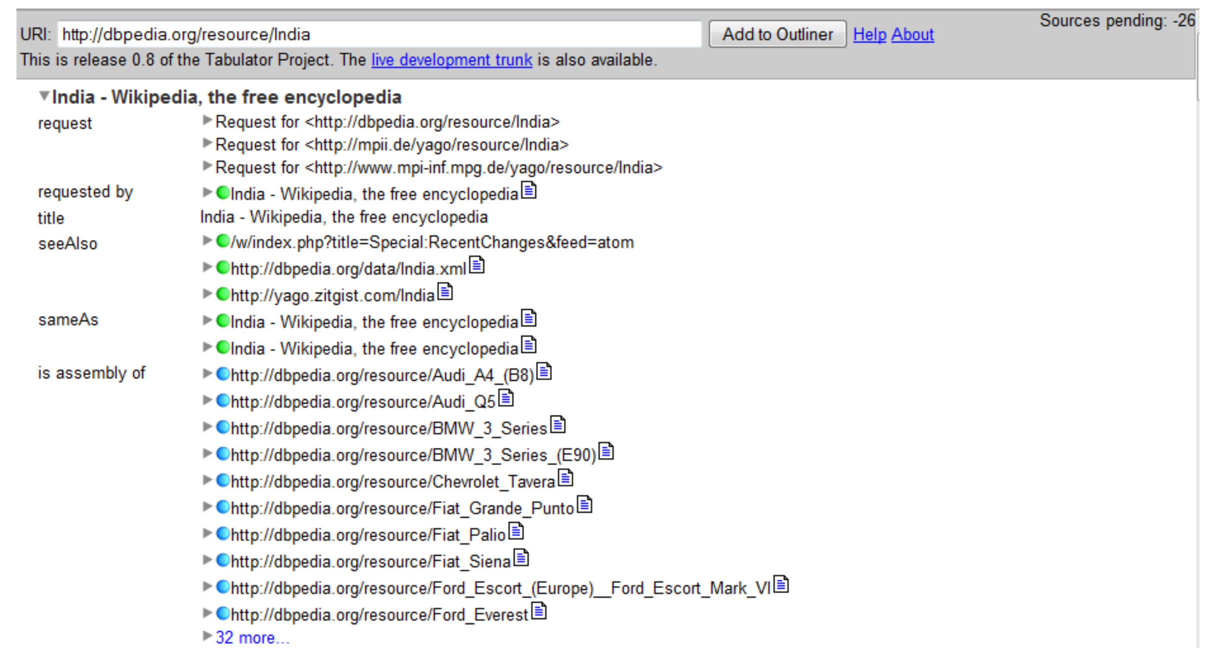
Basic Application: Entity Browsing

Warehousing/ Crawl-Index-Serve



SWSE, Falcons, Sindice, Watson, FactForge…

Virtual Integration/ Distributed Querying

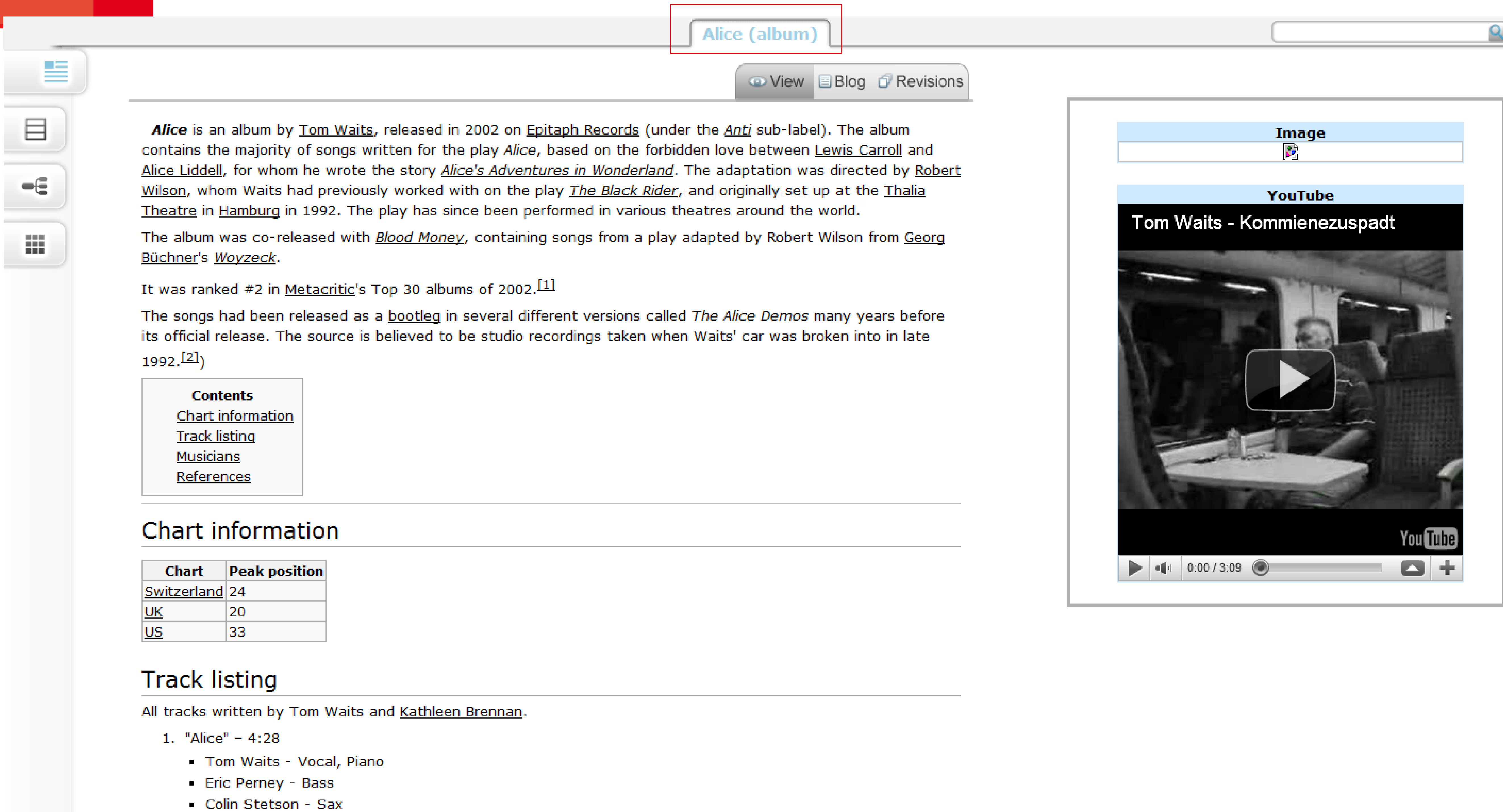


Tabulator, Disco, Zitgist…

Data Sources in SemanticSearch@AIFB Demo [SWC Finalist 2010]

- English Wikipedia
- Data from Linked Open Data
  - DBpedia
  - YAGO
  - Many more
- Live data from Data.gov (US Government)
  - E.g. live data about earthquakes

Wikipedia + DBpedia


Alice (album)

View Blog Revisions

Alice is an album by Tom Waits, released in 2002 on Epitaph Records (under the *Anti* sub-label). The album contains the majority of songs written for the play *Alice*, based on the forbidden love between Lewis Carroll and Alice Liddell, for whom he wrote the story *Alice's Adventures in Wonderland*. The adaptation was directed by Robert Wilson, whom Waits had previously worked with on the play *The Black Rider*, and originally set up at the Thalia Theatre in Hamburg in 1992. The play has since been performed in various theatres around the world.

The album was co-released with *Blood Money*, containing songs from a play adapted by Robert Wilson from Georg Büchner's *Woyzeck*.

It was ranked #2 in Metacritic's Top 30 albums of 2002.[1]

The songs had been released as a bootleg in several different versions called *The Alice Demos* many years before its official release. The source is believed to be studio recordings taken when Waits' car was broken into in late 1992.[2])

Contents

Chart information

Chart	Peak position
Switzerland	24
UK	20
US	33

Track listing

All tracks written by Tom Waits and Kathleen Brennan.

1. "Alice" – 4:28
   - Tom Waits - Vocal, Piano
   - Eric Perney - Bass
   - Colin Stetson - Sax

Image

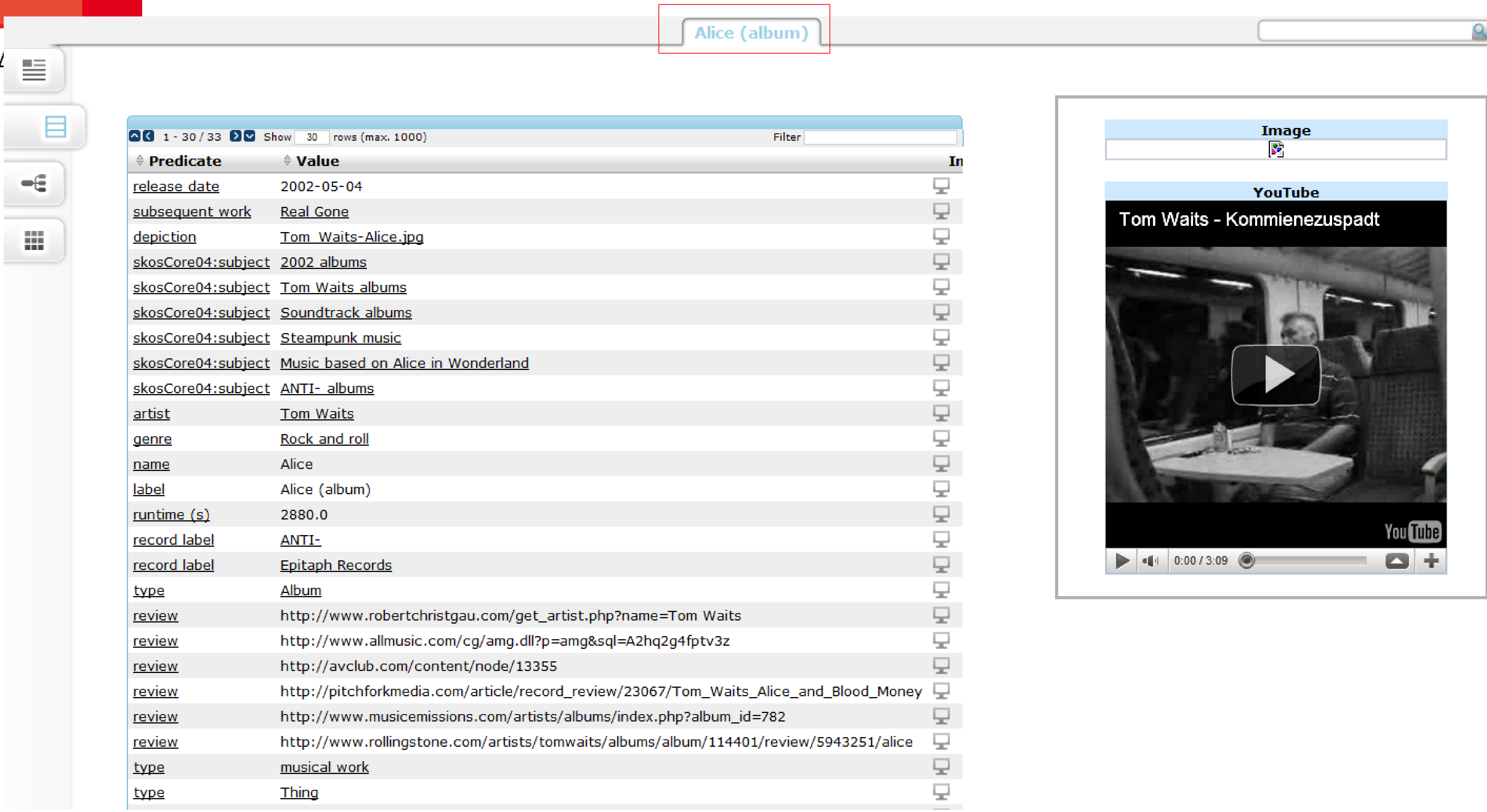
YouTube

Tom Waits - Kommienezuspadt

0:00 / 3:09

source: http://iwb.fluidops.com

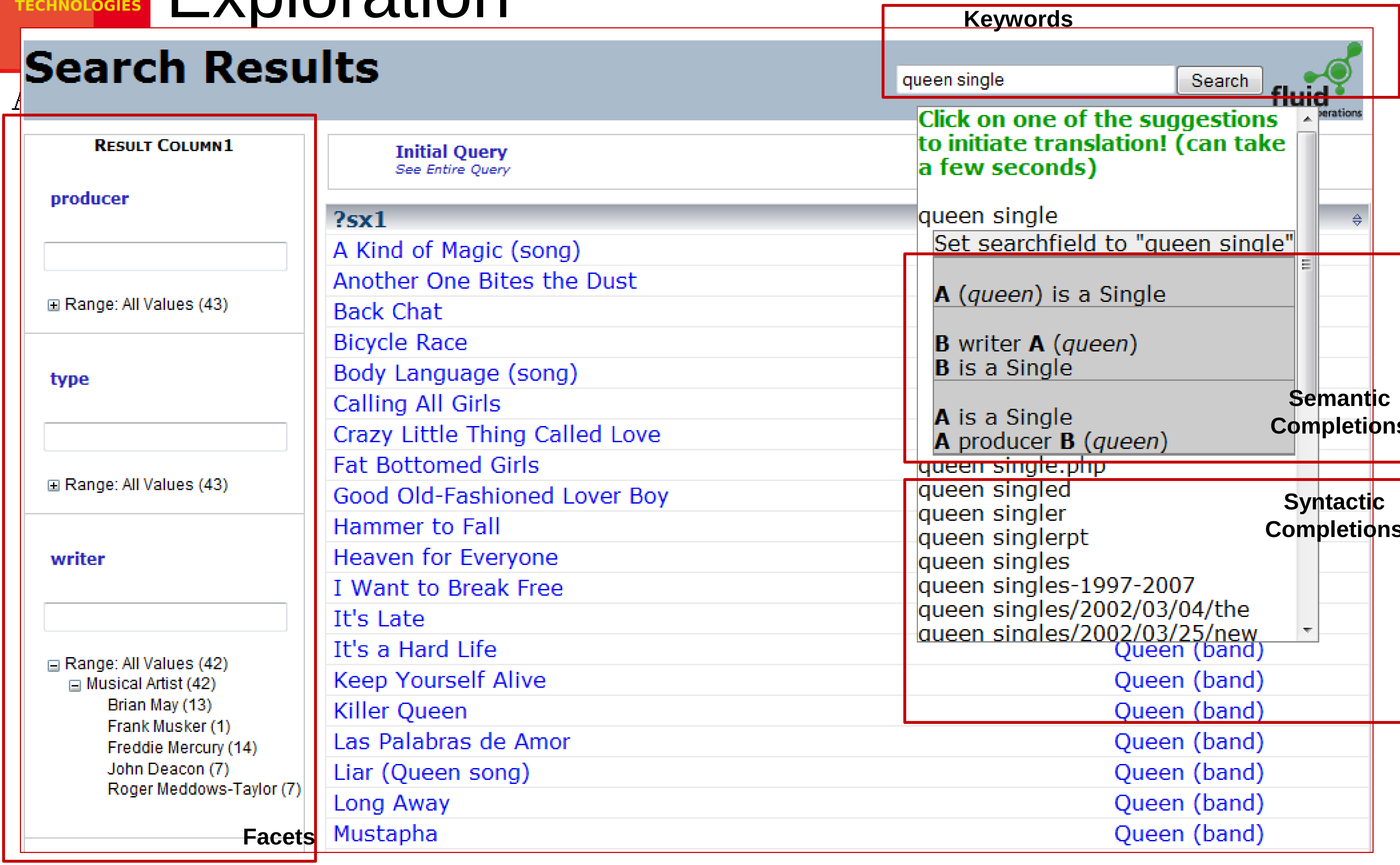
Wikipedia + DBpedia



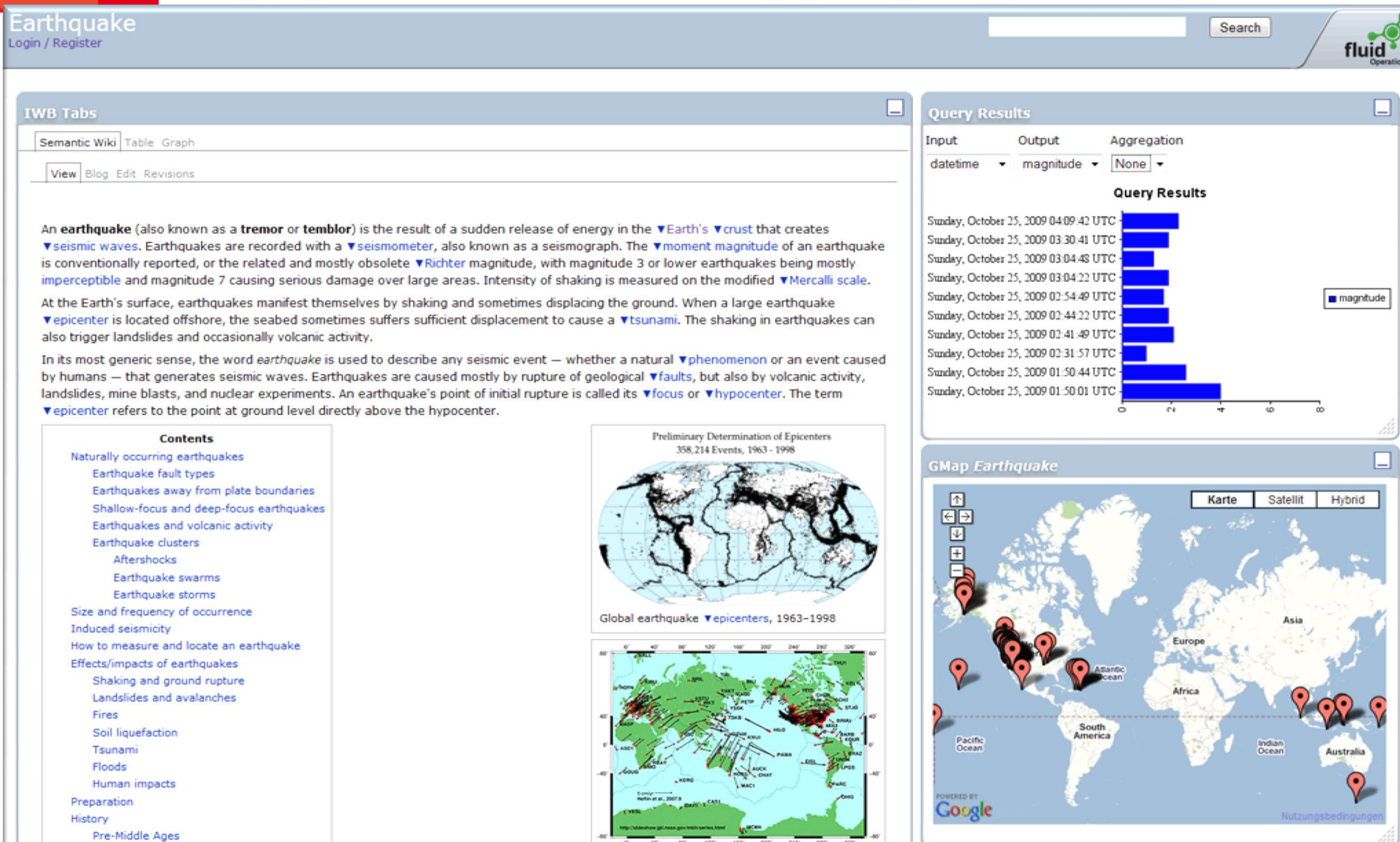
Predicate	Value
release date	2002-05-04
subsequent work	Real Gone
depiction	Tom_Waits-Alice.jpg
skosCore04:subject	2002 albums
skosCore04:subject	Tom Waits albums
skosCore04:subject	Soundtrack albums
skosCore04:subject	Steampunk music
skosCore04:subject	Music based on Alice in Wonderland
skosCore04:subject	ANTI- albums
artist	Tom Waits
genre	Rock and roll
name	Alice
label	Alice (album)
runtime (s)	2880.0
record label	ANTI-
record label	Epitaph Records
type	Album
review	http://www.robertchristgau.com/get_artist.php?name=Tom Waits
review	http://www.allmusic.com/cg/amg.dll?p=amg&sql=A2hq2g4fptv3z
review	http://avclub.com/content/node/13355
review	http://pitchforkmedia.com/article/record_review/23067/Tom_Waits_Alice_and_Blood_Money
review	http://www.musicemissions.com/artists/albums/index.php?album_id=782
review	http://www.rollingstone.com/artists/tomwaits/albums/album/114401/review/5943251/alice
type	musical work
type	Thing

source: http://iwb.fluidops.com

Query Interpretation, Refinement and Exploration



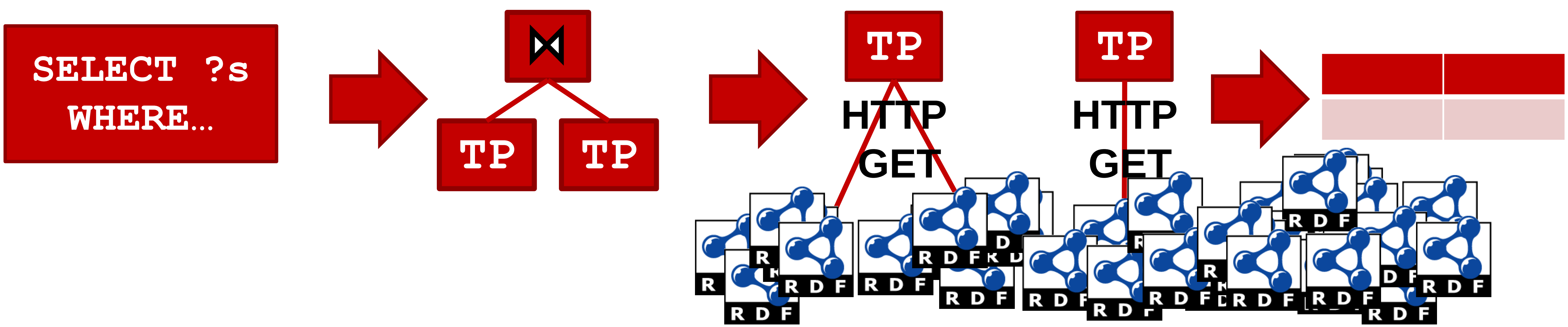
Result Inspection, Analysis and Browsing



Querying Data Across Sources

- Data warehousing or materialisation-based approaches (MAT)

- Distributed query processing approaches (DQP) over Linked Data sources



- DQP over RDF stores (not covered)

(Linked Data) Crawling

1. Get URI from a queue
2. Open connection and fetch content
3. Process and store content
4. Extract new links and put into queue
5. At defined intervals: schedule URIs in queue

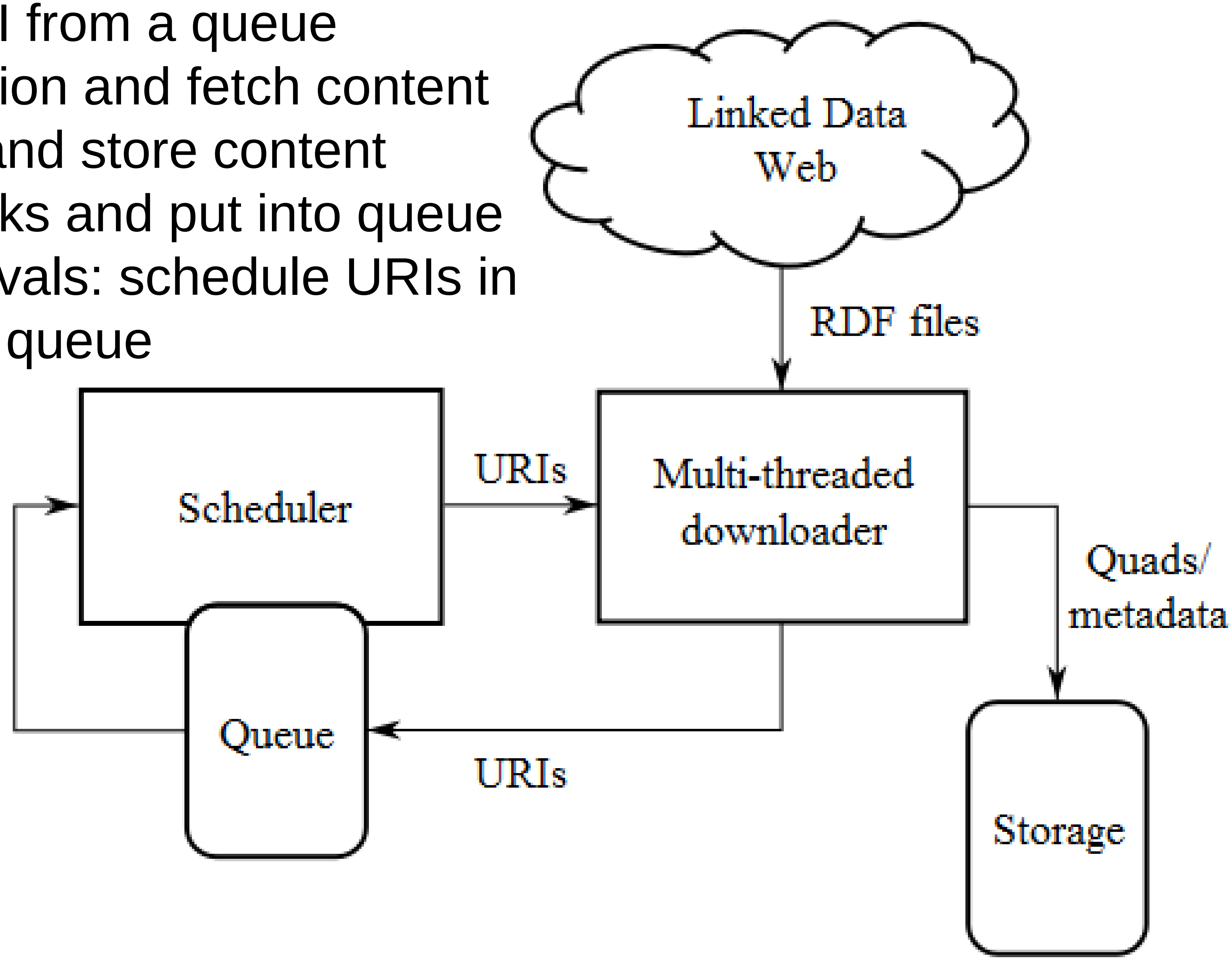


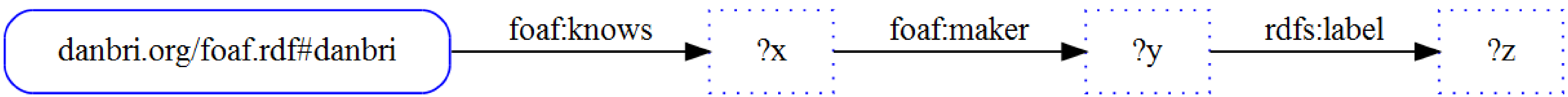
Image: Carlos Castillo: Effective Web Crawling, via Wikipedia

Conjunctive Queries / BGP

<http://danbri.org/foaf.rdf#danbri> foaf:knows ?x .

?x foaf:maker ?y .

?y rdfs:label ?z .



- Built on triple patterns containing variables (?, ?, ?), (s, ?, ?), (?, p, ?), (?, ?, o), (s, p, ?), (?, p, o), (s, p, o)
- Variables are bound during query evaluation
- Query evaluation results in a set of variable bindings

?x	?y	?z
http://www.w3.org/People/Berners-Lee/card#i	http://www.w3.org/DesignIssues/LinkedData.html	Linked Data - Design Issues - W3C

RDF Storage (1/4)
Horizontal table-per-class

Class: **Car**

plate	model	doors	owner	series	img	seeAlso
234HSJ	Toyota	5	Frank	Corolla	x.jpg	toyota.com

- Pros:
  - Fast for certain queries, esp. “star shaped” queries
  - Little redundancy in cells
- Cons:
  - Becomes very sparse for larger schema
    - Lots of nulls needed
  - Special handling needed for multi-valued attributes

Frank	234HSJ	France	1.92		a.svg	…
Jill	148YUI	Ireland	1.75			…
Jim	923HIJ	US & UK		black	b.svg	

RDF Storage (2/4)
Vertical triple table

subject	predicate	object
Frank	ownsCar	234HSJ
234HSJ	seeAlso	toyota.com
234HSJ	series	Corolla

- Pros:
  - No more nulls needed
  - Flexible for updates (even to data-schema)
  - Multi-valued attributes no problem
- Cons:
  - Lot’s of self-joins
  - Lot’s of redundancy in the cells

RDF Storage (3/4) Vertical table per prop.

Property: **model**

subject	object
234HSJ	Toyota
923HIJ	Ford
242HFI	Fiat
541PJH	Porsche
148YUI	Jeep

Property: **ownsCar**

subject	object
Frank	234HSJ
Jill	148YUI
Jim	923HIJ
Joan	541PJH
Mary	242HFI

Property: **type**

subject	object
234HSJ	Car
923HIJ	Car
242HFI	Car
541PJH	Car
148YUI	Car

- Pros:
  - Less redundancy
- Cons:
  - Potentially many tables
  - New property = new table
  - Assumes predicate always known

RDF Storage (4/4) Hybrid

Class: **Car**

plate	model	doors	owner	series
234HSJ	Toyota	5	Frank	Corolla
923HIJ	Ford	3	Jim	Ka
242HFI	Fiat	3	Mary	Tempura
541PJH	Pors			
148YUI	Jeep			

Property: **seeAlso**

subject	object
234HSJ	toyota.com
923HIJ	ford.com
242HFI	fiat.com
	porsche.com
	jeep.com
	joan.com

- Pros:
  - ~Depends
- Cons:
  - Likely to be more costly to manage

name	ownsCa			
Frank	234HSJ	France	1.92	
Jill	148YUI	Ireland	1.75	
Jim	923HIJ	US & UK		black
Joan	541PJH	India	1.75	
Mary	242HFI	Scotland		

Property: **img**

subject	object
234HSJ	x.jpg
148YUI	i.png
Frank	a.svg
Jim	b.svg

Triple stores vs. Quad stores

1. Triple stores
   - Only service simple RDF triple patterns
     - ?s rdf:type foaf:Person .
     - aidan ?p galway .
     - ?s ?p ?o .
     - …
2. Quad stores/SPARQL engines
   - Also service patterns involving named graphs
   - Typical for indexing data from multiple sources
   - Needed for SPARQL querying!!
     - GRAPH ?g { ?s rdf:type foaf:Person }
     - GRAPH foaf.rdf { aidan ?p galway }
     - FROM graph1.rdf … WHERE { ?s ?p ?o . }

Building a full Quad index

- (subject, predicate, object, graph)
  - graph sometimes called context
- 2^4 = 16 patterns to service!

No	Access pattern	No	Access pattern
1	(?:?:?:?)	9	(s:?:o:c)
2	(s:?:?:?)	10	(?:?:o:c)
3	(s:p:?:?)	11	(?:?:o:?)
4	(s:p:o:?)	12	(?:?:?:c)
5	(s:p:o:c)	13	(s:?:?:c)
6	(?:p:?:?)	14	(s:p:?:c)
7	(?:p:o:?)	15	(?:p:?:c)
8	(?:p:o:c)	16	(s:?:o:?)

Index Patterns
Six prefix-indexes for quads

- Requires six indexes to service all 16 quad patterns
  - assuming prefix lookups
    - e.g., can lookup SP in SPOC index... (~indexes should be sorted)

GRAPH ?c { :ted?p ?o }
pattern: (s:?:?:?)
index: spoc

GRAPH :dblp { :ted ?p ?o }
pattern: (s:?:?:c)
index: csp

GRAPH ?c { ?s rdf:type foaf:Person }
pattern: (?:p:o:?)
index: poc

spoc	poc	ocs
(?:?:?:?)	(?:p:?:?)	(?:?:o:?)
(s:?:?:?)	(?:p:o:?)	(?:?:o:c)
(s:p:?:?)	(?:p:o:c)	(s:?:o:c)
(s:p:o:?)		
(s:p:o:c)		

csp	cp	os
(?:?:?:c)	(?:p:?:c)	(s:?:o:?)
(s:?:?:c)		
(s:p:?:c)		

Index Implementation
Blocked, Sorted File

- Sorted, c
- Combinat
- Read opti
- Bulk-load

- Pros:
  - !Fast! sequential reads (no OIDs)
  - Fast load
  - Good for simple queries, lots of results
  - Configurable block sizes
- Cons:
  - Does not support updates!
  - Does not support updates!!!

```
...
http://deri.org/ dc:tit
http://jfdk_id=47 foaf:
http://www.deri.org/ dc:title
...
```

```
~aharth/foaf.rdf
eri.org/.../foaf.rdf

of.fds...foaf.rdf
fds...foaf.rdf
ri.org..foaf.rdf
http://jfdk_id=59 foaf:name "Dieter Fensel" http://deri.at/...foaf.rdf
...
```

```
http://www.deri.org/ dc:title "DERI" http://www.deri.ie/index.rdf
...
```

```
...
```

sparse index in memory

quadruples stored in blocks on disk, sorted according to natural ordering

Dictionary Encoding / Object IDs

Data Table

subject	predicate	object
1	2	3
1		
1		
1		
1		
1		
1		
…		

Dictionary

OID	String
1	Frank

- Pros:
  - Can load more data in memory
  - Faster to compute joins
  - Better RISC-style processing [RDF-3X]
  - Smaller on-disk footprint
- Cons:
  - Maintain a potentially massive dictionary
  - Slower to externalise streaming results
  - FILTERs optimisations are trickier

Query Processing (1/4)

```
?person foaf:name ?name .
?person foaf:based_near dbpedia:Karlsruhe .
:aidan foaf:knows ?person .
```

name?

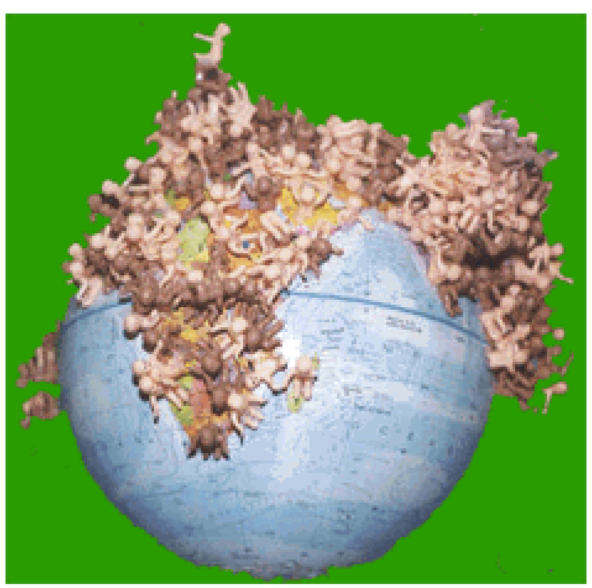
based in Karlsruhe?

get aidans friends

(3)

(2)

(1)



7 billion

0.3 million

2

Query Processing (2/4)
Push FILTER expressions

```
?person foaf:name ?name .
?person foaf:age ?age .
?person foaf:based_near dbpedia:Karlsruhe .
FILTER ( ?age > 110 )
```

- Filter intermediate results as soon as possible

get name

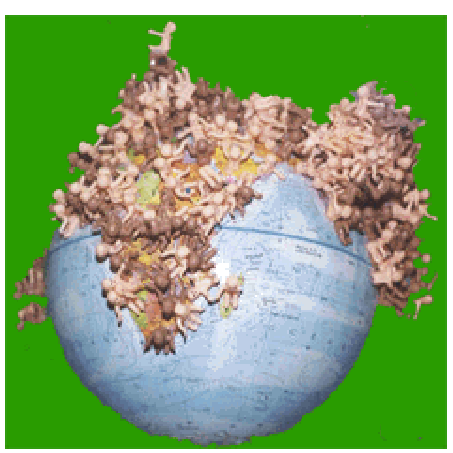
get age: 110+?

get Karlsruhe folks

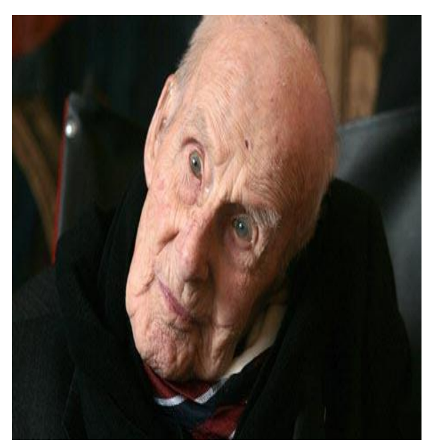
(3)

(2)

(1)



7 billion



400

0.5 million

Query Processing (3/4)
Use Index, Push FILTER

```
?person foaf:name ?name .
?person foaf:age ?age .
?person foaf:based_near dbpedia:Karlsruhe .
FILTER ( ?age > 110 )
```

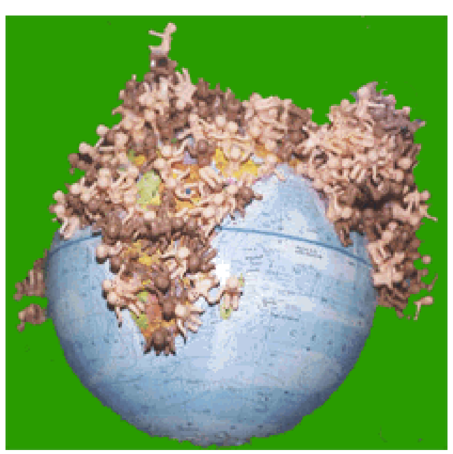
- If you have the right index (sorted by `PO` or [maybe] `OP` prefix with numeric order on O), seek directly to 110+ for age

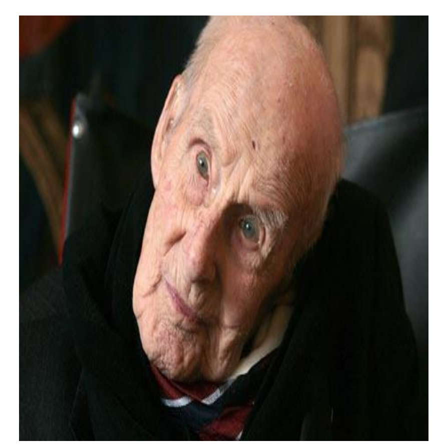
get name

in Karlsruhe?

jump directly to age > 110

(3) (2) (1)





7 billion **0.5 million** **400**

Query Processing (4/4)
No keyword search support in SPARQL

...give me resources mentioning "`Bonn ISWC`"

- **Can use `FILTER`/`REGEX` in SPARQL...**

```
?s ?p ?o , ?p2 ?o2 .
FILTER ( REGEX(?o, "Bonn")  && REGEX(?o2, "ISWC") )
```

  - ***...but that sucks and is expensive***
  - ***...need an IR-style inverted index, not*** `FILTER`

- **Most SPARQL engines support custom keyword index/syntax**
  - **Against literals or concatenation of literals**
  - **Apache Lucene! ...perfect for the job!**

Plan

Attribution

- Slides erstellt von Andreas Harth, Aidan Hogan, Spyros Kotoulas, Jacopo Urbani für das Tutorial "**Scalable Integration and Processing of Linked Data**"
  - siehe http://sild.cs.vu.nl/