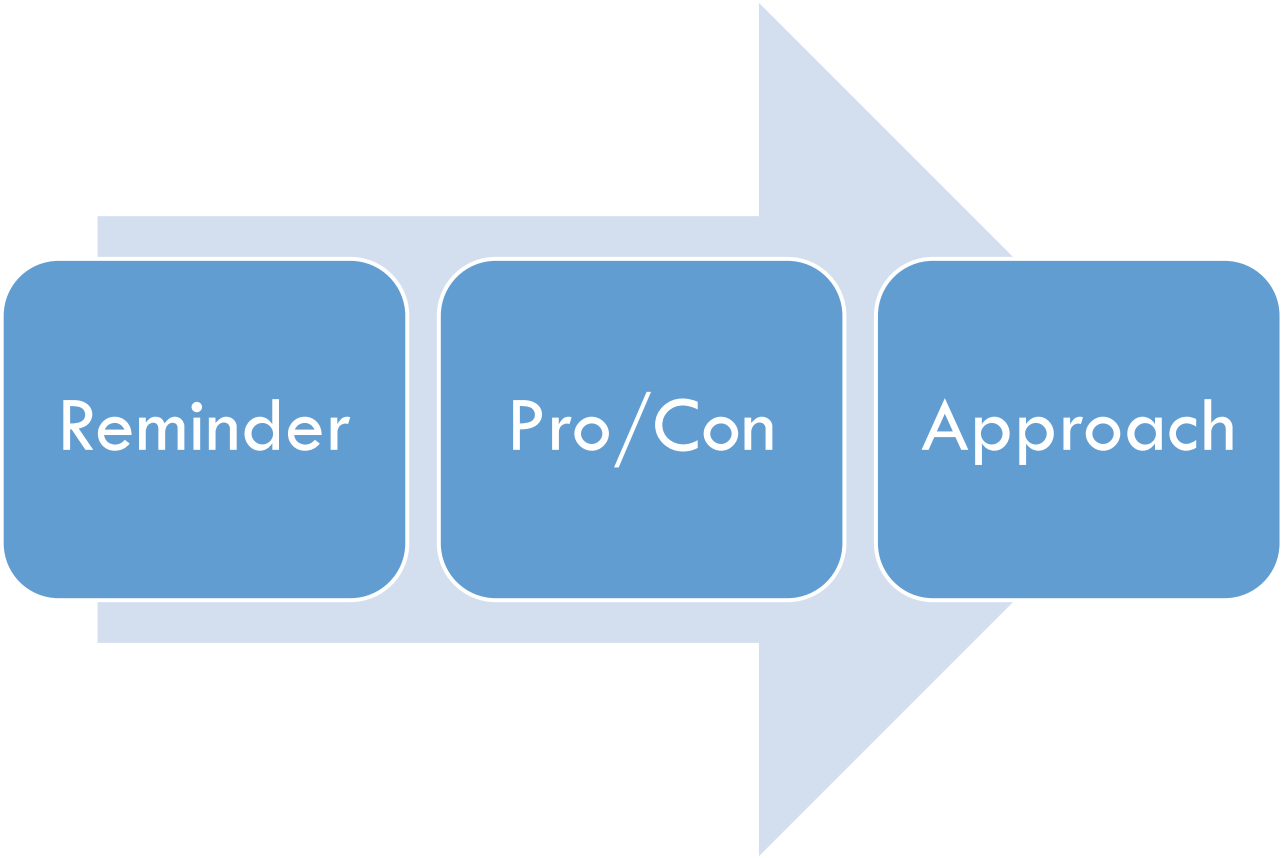
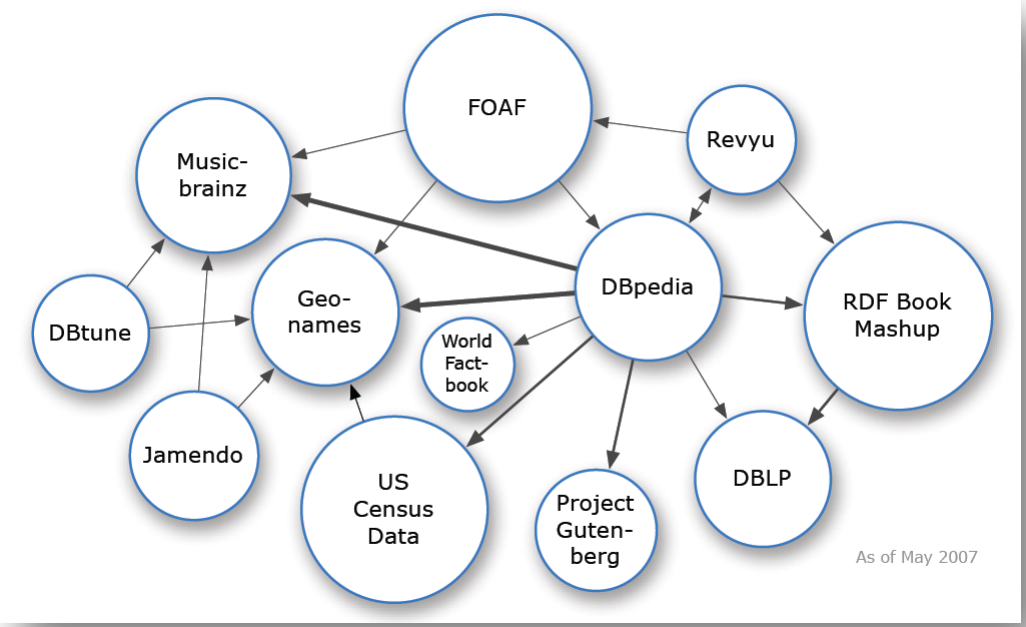
SPARQL FOR LINKED DATA

Agenda



Linked Data - Reminder

- *"a recommended best practice for **exposing, sharing, and connecting** pieces of **data, information, and knowledge** on the Semantic Web using URIs and RDF."*



Linked Data - Reminder

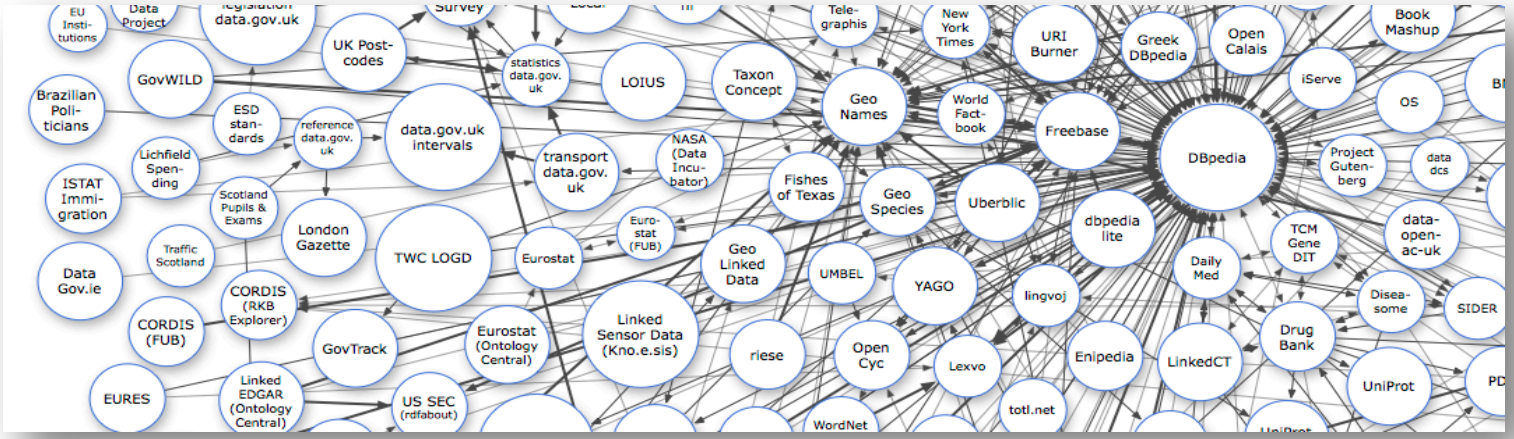
- **Principles**
  - use URIs as names for things
  - use HTTP URIs so that people can look up those names
  - when someone looks up a URI, provide useful RDF information
  - include RDF statements that link to other URIs so that they can discover related things

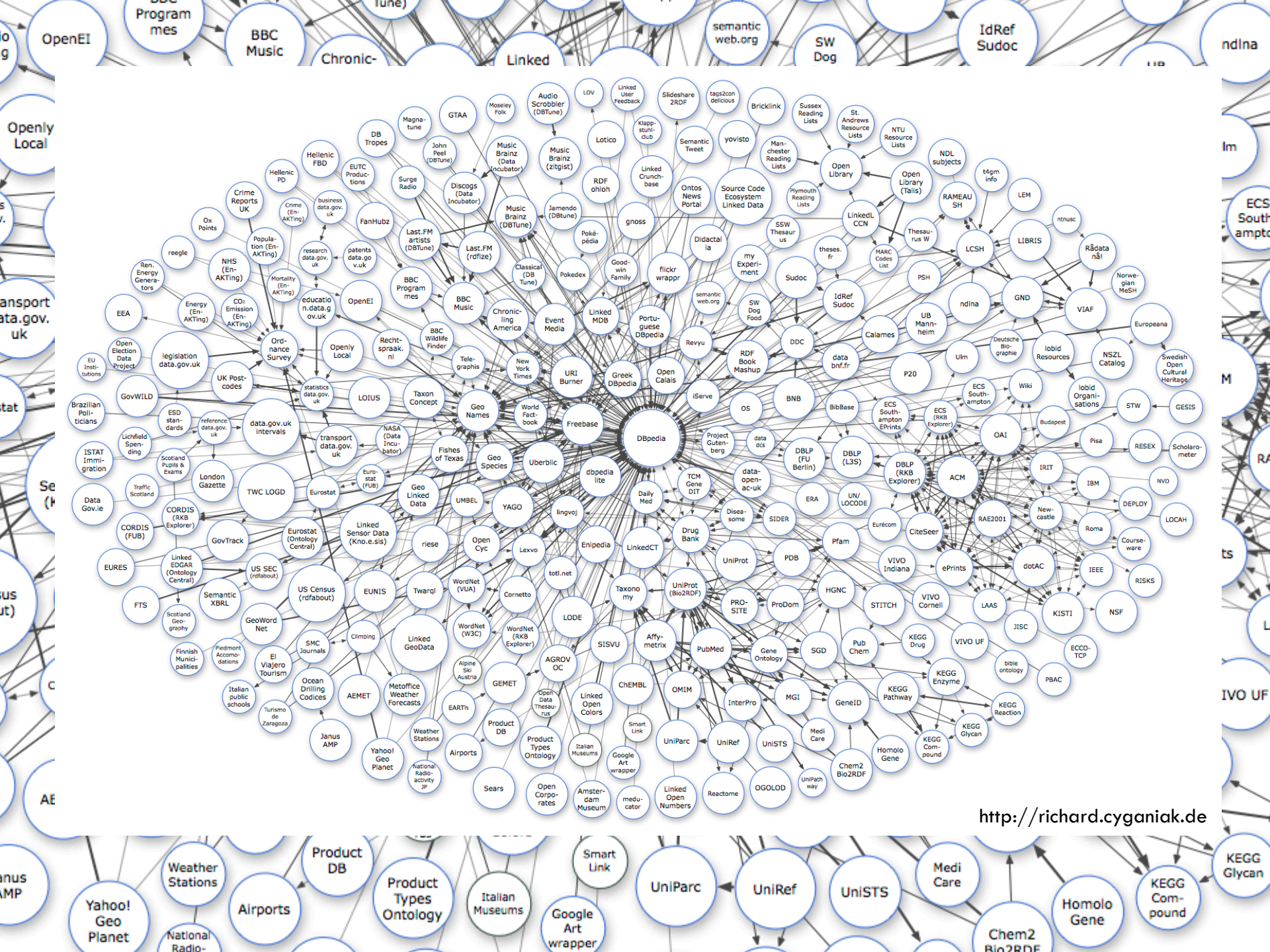
Advantages

- Openness
  - virtually unbound set of data sources
  - data from different data sources can be aggregated
  - fragmentary information from multiple sources can be integrated to achieve a more complete view.

Challenges

- not possible to know all data sources that might be relevant for answering





http://richard.cyganiak.de

Iterator-based Query Execution

- **Iterator**
  - Each iterator is responsible for a single triple pattern
- Query Execution uses a chain of iterators


```
SELECT ?c ?u WHERE {
 <http://mymovie.db/movie2449> mov:filming_location ?c .
 ?c geo:statistics ?cStats .
 ?cStats stat:unempRate ?u . }
```

l_2 - Get the statistics

tp_2 = (?c geo:statistics ?cStats)
μ_{cur} = { ?p -> http://geo.. , ?c -> http://geo..}

1. Substitute $tp_{cur} = \mu_{cur}[tp_2]$

   tp_{cur} = (http://geo.. geo:statistics ?cStats)

2. Find matching triples $match(tp_{cur})$ in queried data set

   (http://geo.. geo:statistics http://1..), (http://geo.. geo:statistics http://2..)

3. Create solution μ' for each t in $match(tp_{cur})$

   μ' = { ?cStats -> http://1..}

4. Return each $\mu_{cur} \cup \mu'$ as a result

   μ_{cur} = { ?p -> http://geo.. , ?c -> http://geo.., ?cStats -> http://1..}

Automated Link Traversal

- retrieve further data by looking-up relevant URIs
  - queried data set grows

- **Look-up Requirement**
  - Don‘t evaluate tp_{cur} until queried data set contains all data that can be retrieved from all URIs in tp_{cur}

Automated Link Traversal

1. Substitute $tp_{cur} = \mu_{cur}[tp_i]$
2. Ensure look-up requirement for tp_{cur}
3. Find matching triples $match(tp_{cur})$ in queried data set
4. Create solution μ' for each t in $match(tp_{cur})$
5. Return each $\mu_{cur} \cup \mu'$ as a result

Waiting for the URI look-ups blocks the query execution!

URI Prefetching – a solution?

- When a URI is bound to a variable initiate look-up in the background

URI Prefetching – a solution?

1. Substitute $tp_{cur} = \mu_{cur}[tp_i]$
2. Ensure look-up requirement for tp_{cur}
3. Find matching triples $match(tp_{cur})$ in queried data set
4. Create solution μ' for each t in $match(tp_{cur})$
5. Initiate parallel look-up for each new URI in μ'
6. Return each $\mu_{cur} \cup \mu'$ as a result

URI Prefetching – a solution?

- **No! Since it can also block the query execution**

Extension of the Iterator paradigm

- **POSTPONE**
  - take most recently provided result back
  - allows to temporarily reject input solution μ_{cur}

- Rejected results are kept for later requests and are dismissed if they haven‘t been rejected again

Extension of the Iterator paradigm

1. Substitute $tp_{cur} = \mu_{cur}[tp_i]$
2. POSTPONE μ_{cur} if look-up requirement fails for tp_{cur}
3. Find matching triples $match(tp_{cur})$ in queried data set
4. Create solution μ' for each t in $match(tp_{cur})$
5. Initiate parallel look-up for each new URI in μ'
6. Return each $\mu_{cur} \cup \mu'$ as a result

References

- **[Hartig, Bizer ,Freytag 2009]** Olaf Hartig, Christian Bizer, Johann Christoph Freytag: **Executing SPARQL Queries over the Web of Linked Data.** International Semantic Web Conference 2009: 293-309

- **[Hartig 2012]** Olaf Hartig: **SPARQL for a Web of Linked Data: Semantics and Computability**. ESWC 2012: 8-23

- **[Correndo et al. 2010]: SPARQL query rewriting for implementing data integration over linked data.** Proceedings of the 2010 EDBT/ICDT Workshops

- **[Bizer, Heath, Berners-Lee]** Bizer, C., Heath, T., Berners-Lee, T.: **Linked data - the story so far.** Journal on Semantic Web and Information Systems 2009