O'REILLY®
Strata
CONFERENCE
Making Data Work
cloudera

1–2 OCTOBER 2012
LONDON, ENGLAND

#strataconf
strataconf.com/london

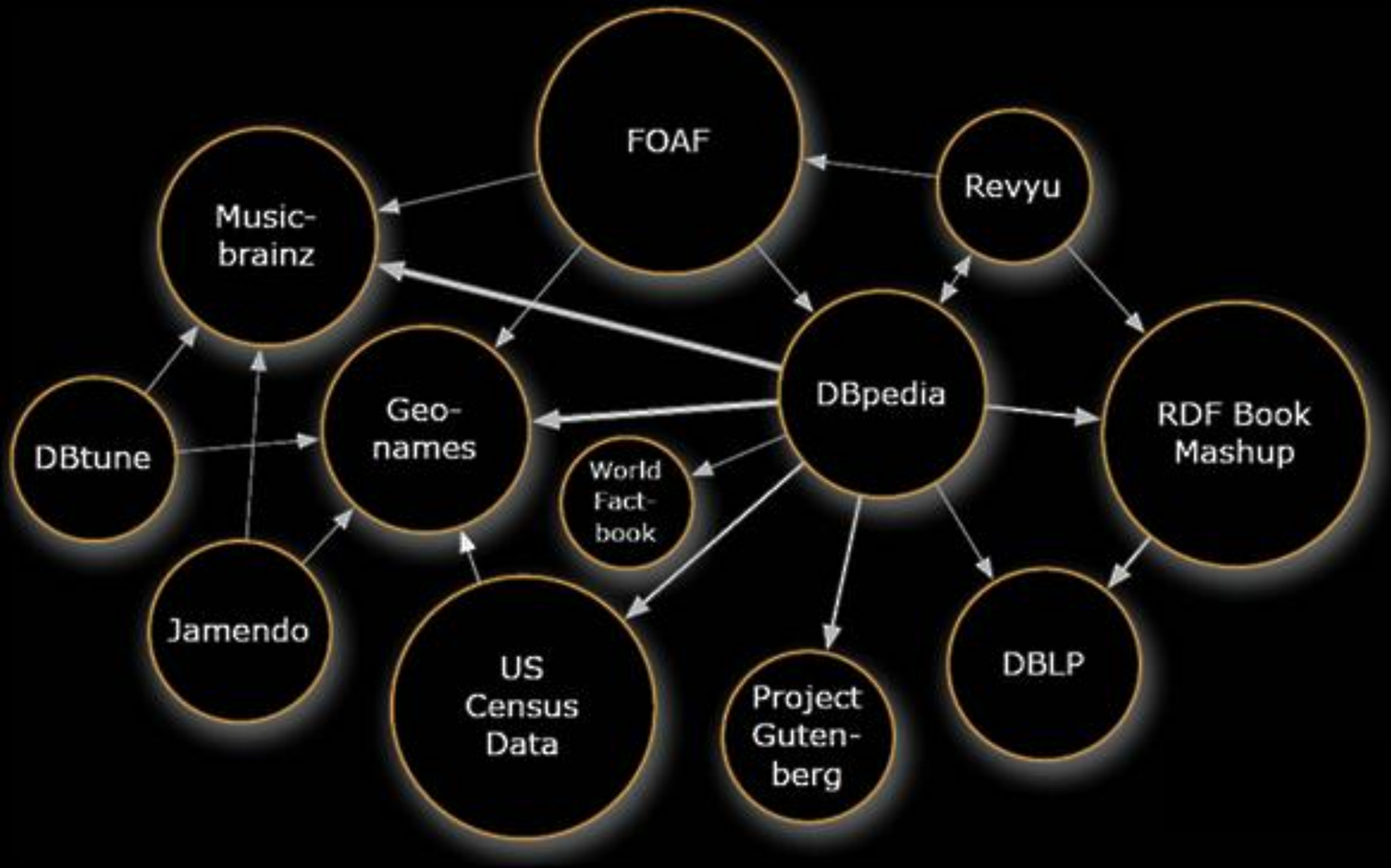
Handling RDF data with tools from the Hadoop ecosystem

Paolo Castagna
Solution Architect at Cloudera
@castagna

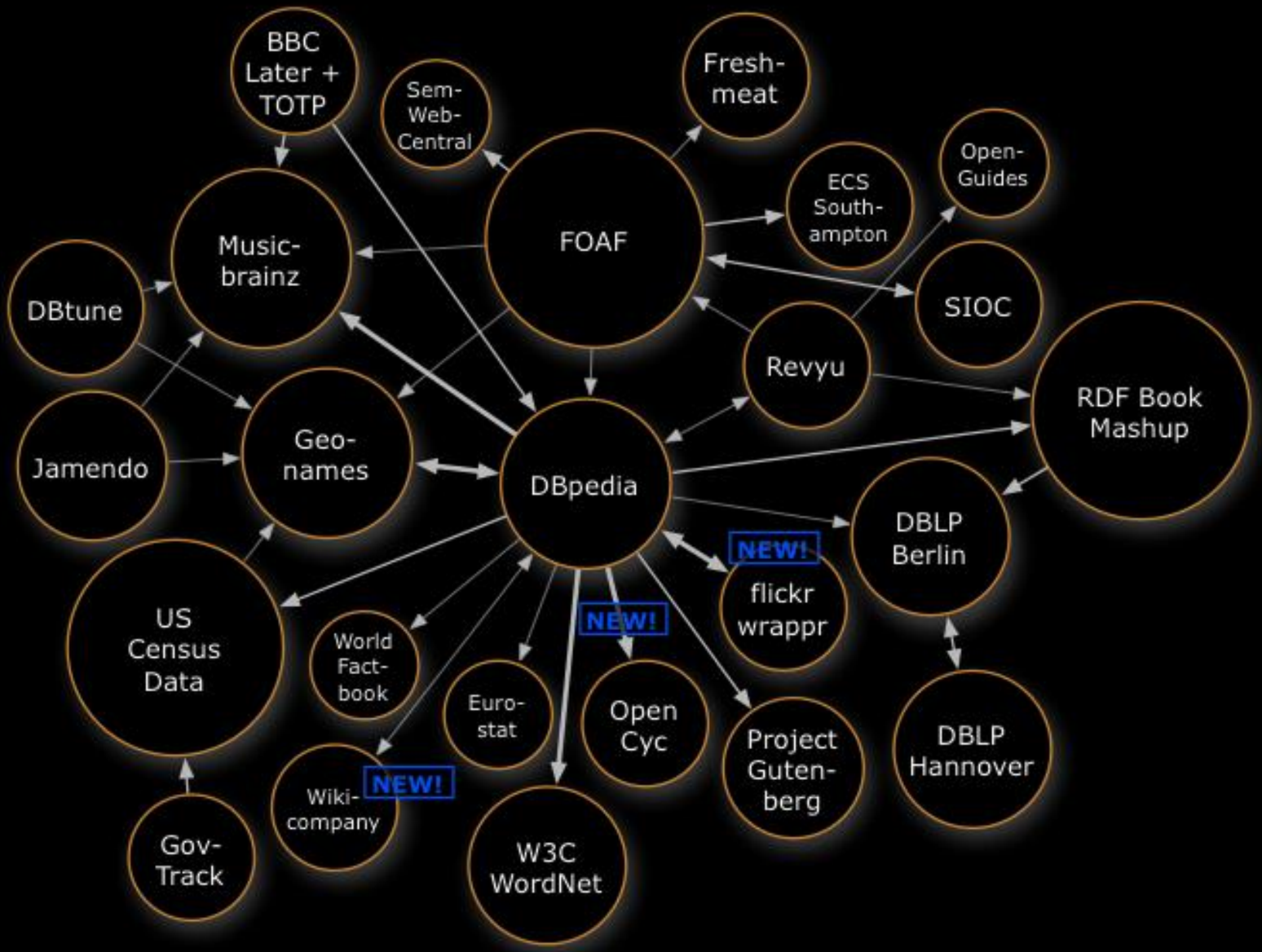
WHY RDF?

There might be data on the Web you are interested in.

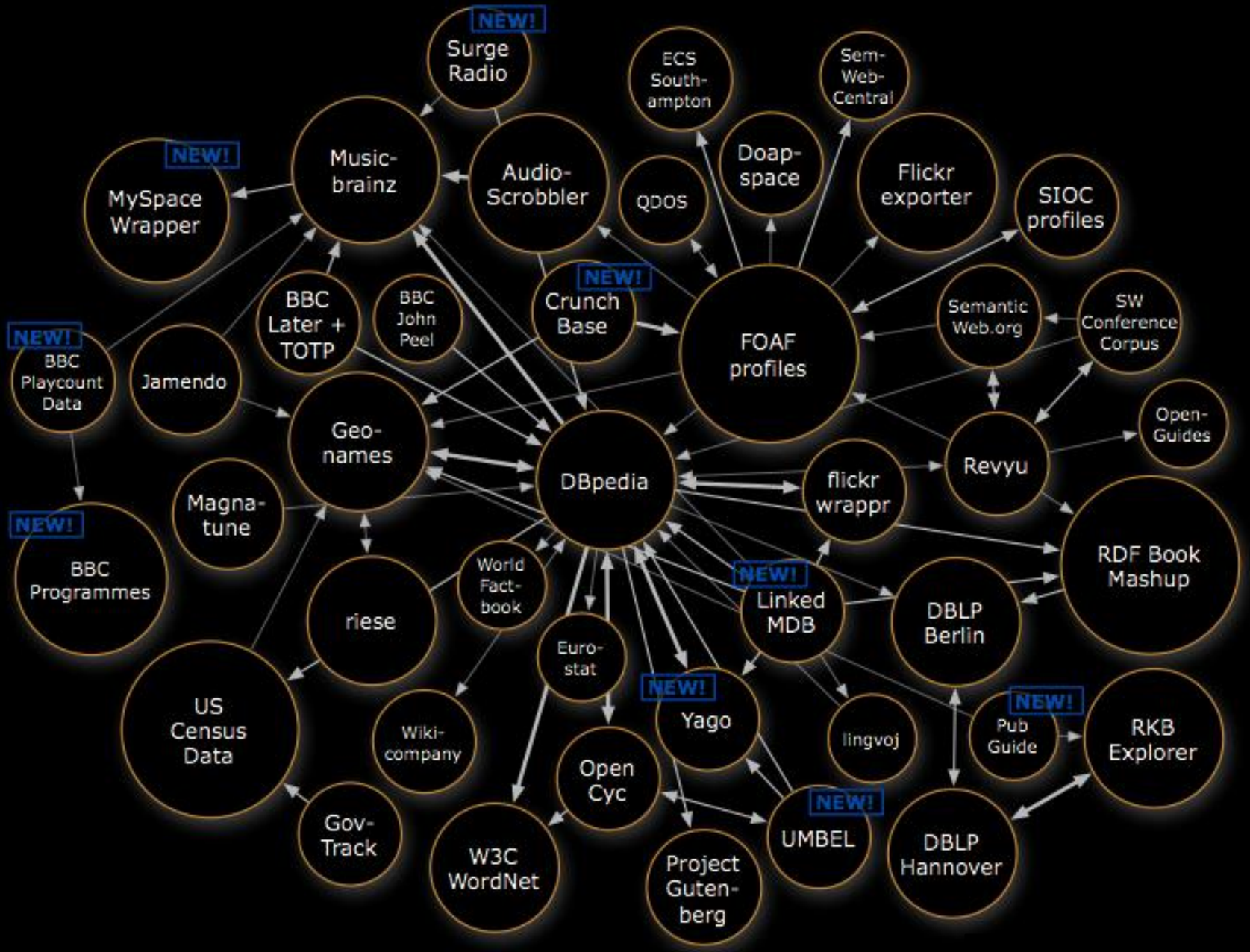
2007



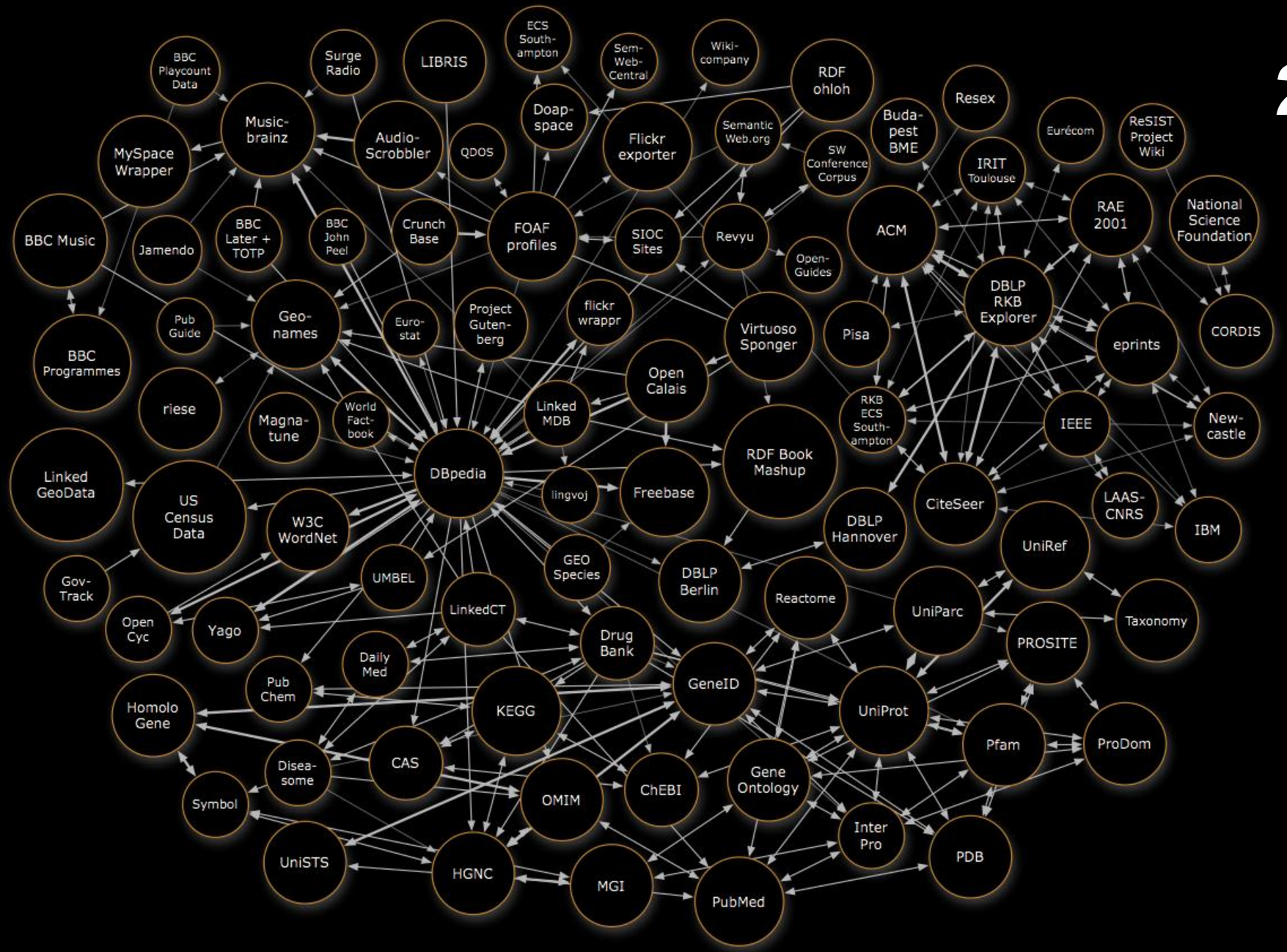
2007



2008

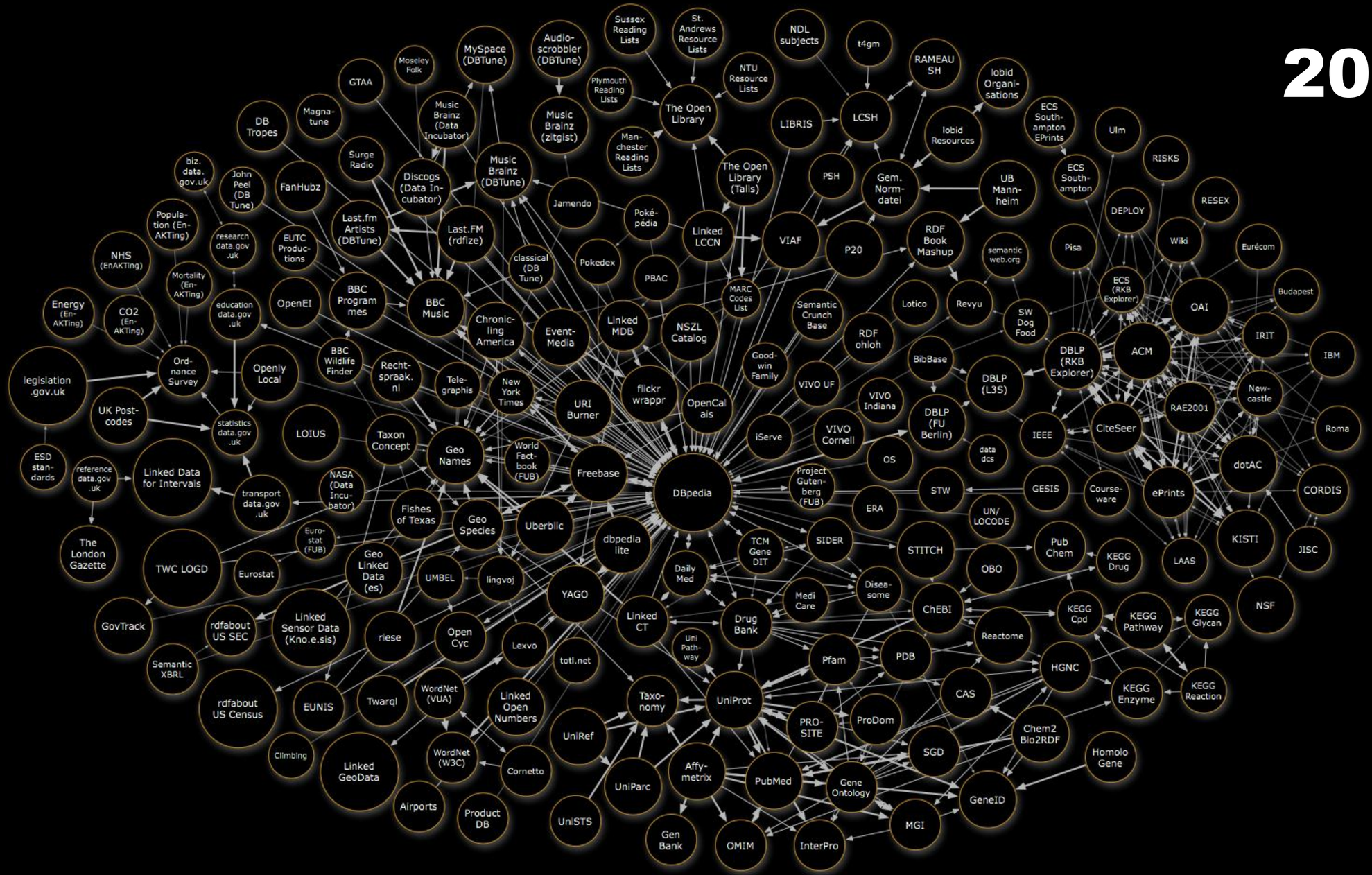


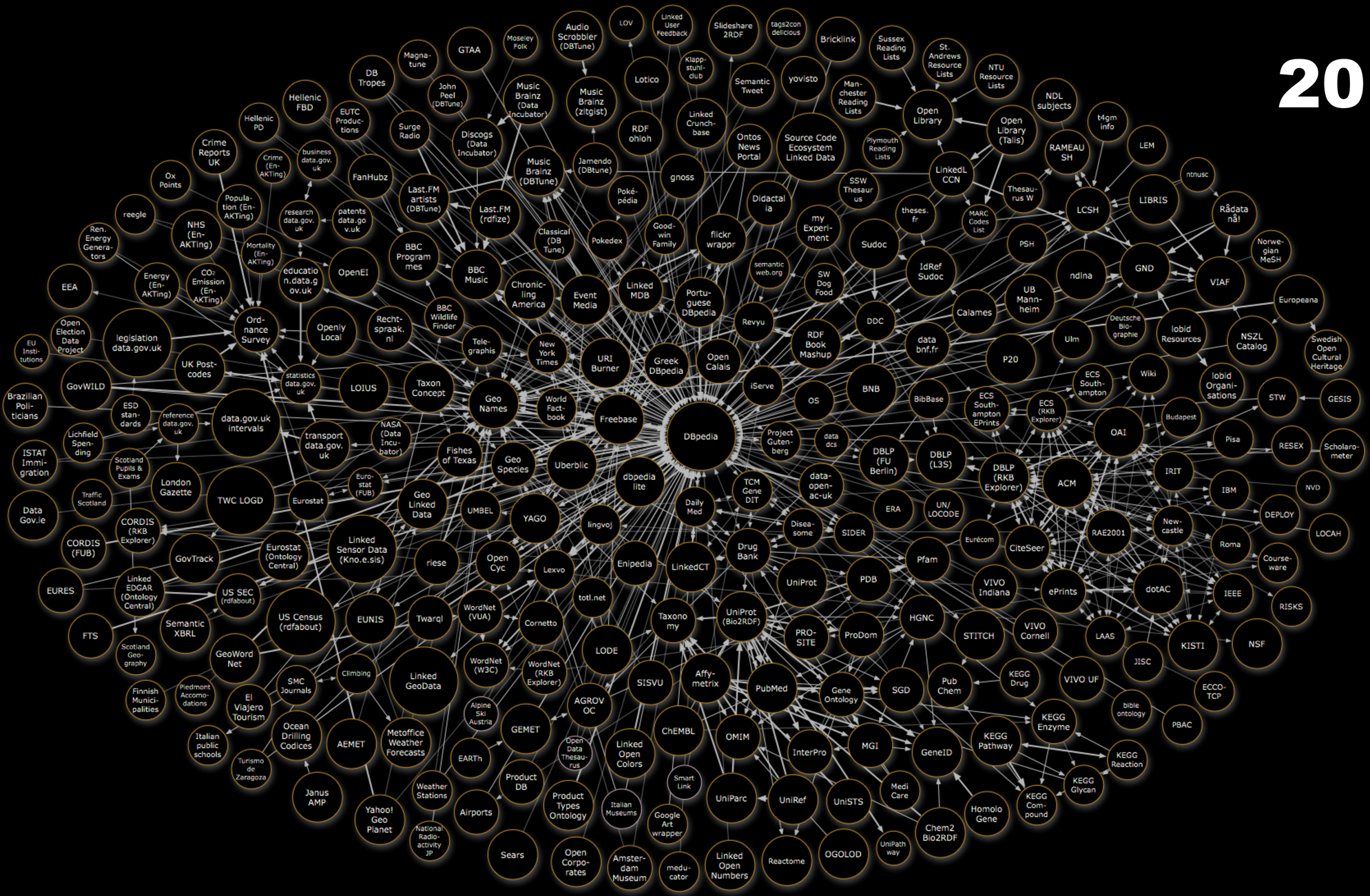
2009



Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

2010





cloudera

Strata CONFERENCE

WHAT'S RDF?

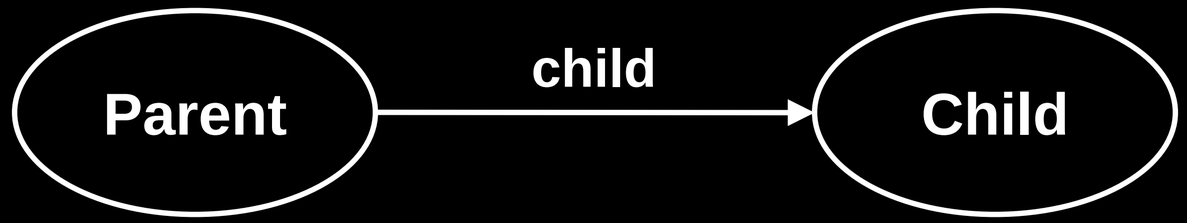
A data model to represent relationships between things or concepts.

Making relationships explicit is a way to capture knowledge, describe and understand things.

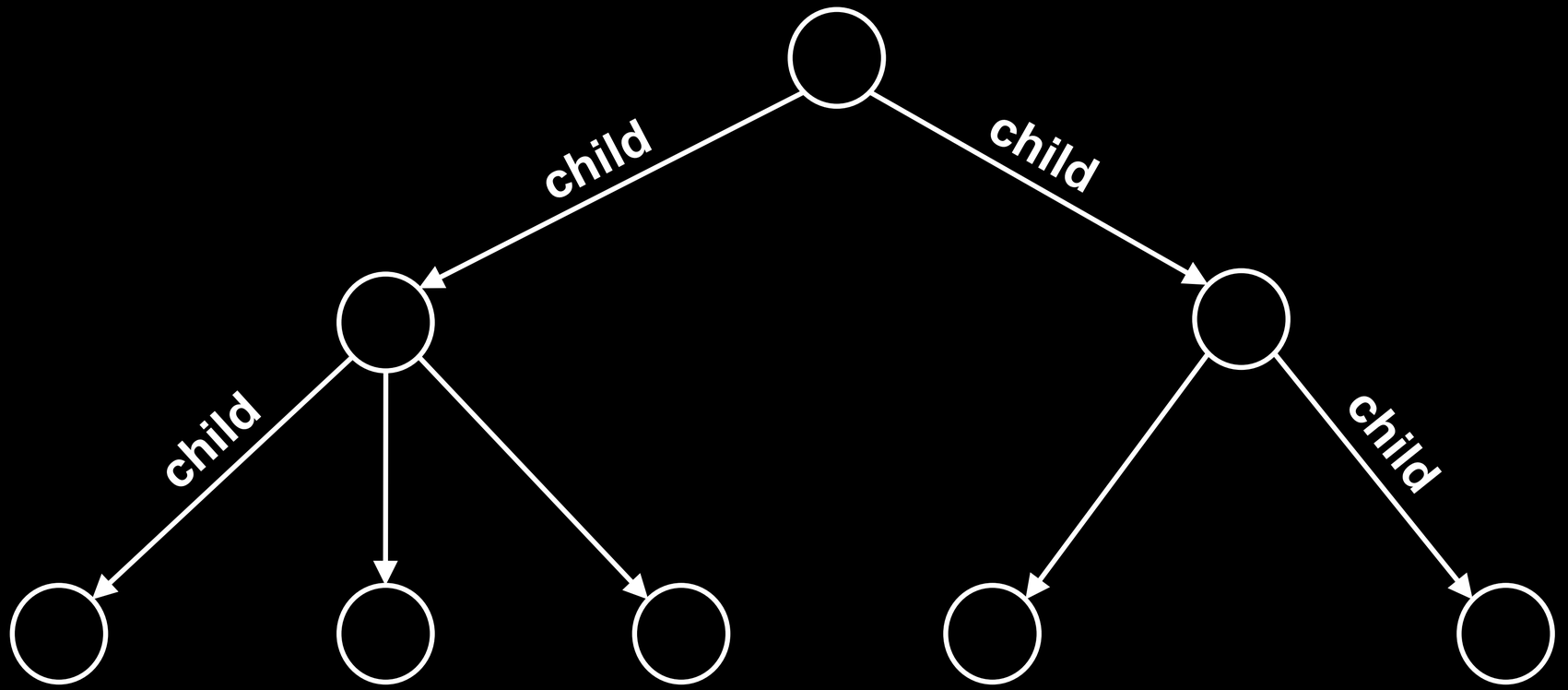
Data Models

Tables	Trees	Graphs
Tabular data in RDBMS	XML documents in XML databases	RDF datasets in triple stores
Records	<elements> and “attributes”	Triple subj –pred→obj
Fixed schema	Semi structured	Schema less
SQL	XPath, XSLT and XQuery	SPARQL and inference

XML

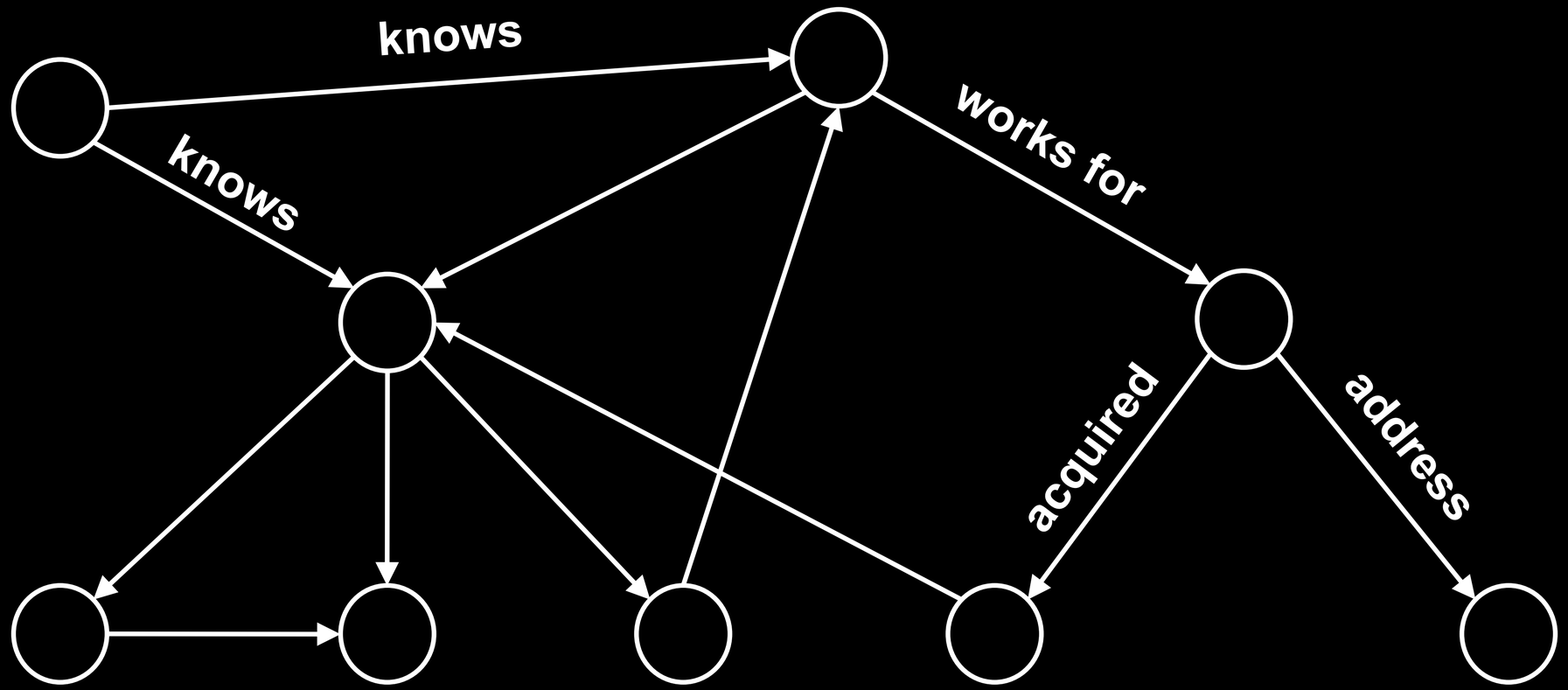


XML



RDF

RDF



HOW TO PROCESS RDF AT SCALE?

Use MapReduce and other tools from the Hadoop ecosystem!

Use N-Triples or N-Quads serialization formats

- One triple|quad per line: splittable and easy to parse
- Use MapReduce to sort or group triples|quads by graph or by subject
- Write your own **`NQuads{Input|Output}Format`** and **`QuadRecord{Reader|Writer}`**
- Parsing one line at the time not ideal, but robust to syntax errors (see also: **`NLineInputFormat`**)

N-Triples
Example

```
<http://example.org/alice>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://xmlns.com/foaf/0.1/Person>  .
<http://example.org/alice>  <http://xmlns.com/foaf/0.1/name>  "Alice"  .
<http://example.org/alice>  <http://xmlns.com/foaf/0.1/mbox>  <mailto:alice@example.org>  .
<http://example.org/alice>  <http://xmlns.com/foaf/0.1/knows>  <http://example.org/bob>  .
<http://example.org/alice>  <http://xmlns.com/foaf/0.1/knows>  <http://example.org/charlie>  .
<http://example.org/alice>  <http://xmlns.com/foaf/0.1/knows>  <http://example.org/snoopy>  .
<http://example.org/bob>  <http://xmlns.com/foaf/0.1/name>  "Bob"  .
<http://example.org/bob>  <http://xmlns.com/foaf/0.1/knows>  <http://example.org/charlie>  .
<http://example.org/charlie>  <http://xmlns.com/foaf/0.1/name>  "Charlie"  .
<http://example.org/charlie>  <http://xmlns.com/foaf/0.1/knows>  <http://example.org/alice>  .
```

Turtle
Example

```
@prefix :        <http://example.org/>   .
@prefix foaf:    <http://xmlns.com/foaf/0.1/>   .

:alice
     a           foaf:Person ;
     foaf:name   "Alice" ;
     foaf:mbox   <mailto:alice@example.org>  ;
     foaf:knows  :bob ;
     foaf:knows  :charlie ;
     foaf:knows  :snoopy ;
     .

:bob
     foaf:name   "Bob" ;
     foaf:knows  :charlie ;
     .

:charlie
     foaf:name   "Charlie" ;
     foaf:knows  :alice ;
```

RDF/XML

Example

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns="http://example.org/" >
  <rdf:Description rdf:about="http://example.org/alice">
    <foaf:knows rdf:resource="http://example.org/snoopy"/>
    <foaf:knows rdf:resource="http://example.org/charlie"/>
    <foaf:knows rdf:resource="http://example.org/bob"/>
    <foaf:mbox rdf:resource="mailto:alice@example.org"/>
    <foaf:name>Alice</foaf:name>
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/bob">
    <foaf:knows rdf:resource="http://example.org/charlie"/>
    <foaf:name>Bob</foaf:name>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/charlie">
    <foaf:knows rdf:resource="http://example.org/alice"/>
    <foaf:name>Charlie</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

Convert RDF/XML, Turtle, etc. To N-Triples

- RDF/XML or Turtle cannot be easily splitted
- Use **`WholeFileInputFormat`** from Tom's book to convert one file at the time
- Many small files can be combined using **`CombineFileInputFormat`**, however in case of RDF/XML or Turtle things get complicated

Validate your RDF data

- Validate each triple/quad separately
- Log a warning with line or offset in bytes of any syntax error, but continue processing
- Write a separate report on bad data: so problems with data can be fixed in one pass
- This can be done with a simple MapReduce job using N-Triples|N-Quads files

Turtle and adjacency lists

```
<http://example.org/alice> <http://xmlns.com/foaf/0.1/mbox> <mailto:alice@example.org>;
<http://xmlns.com/foaf/0.1/name> "Alice"; <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>; <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie>,
<http://example.org/bob>, <http://example.org/snoopy>; . <http://example.org/charlie> <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice> .
<http://example.org/bob> <http://xmlns.com/foaf/0.1/name> "Bob"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/charlie>; . <http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/bob> .
<http://example.org/charlie> <http://xmlns.com/foaf/0.1/name> "Charlie"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice>; . <http://example.org/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
<http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
```

Turtle and adjacency lists

```
<http://example.org/alice> <http://xmlns.com/foaf/0.1/mbox> <mailto:alice@example.org>;
<http://xmlns.com/foaf/0.1/name> "Alice"; <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>; <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie>,
<http://example.org/bob>, <http://example.org/snoopy>; . <http://example.org/charlie> <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice> .
<http://example.org/bob> <http://xmlns.com/foaf/0.1/name> "Bob"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/charlie>; . <http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/bob> .
<http://example.org/charlie> <http://xmlns.com/foaf/0.1/name> "Charlie"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice>; . <http://example.org/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
<http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
```

Turtle and adjacency lists

```
<http://example.org/alice> <http://xmlns.com/foaf/0.1/mbox> <mailto:alice@example.org>;
<http://xmlns.com/foaf/0.1/name> "Alice"; <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>; <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie>,
<http://example.org/bob>, <http://example.org/snoopy>; . <http://example.org/charlie> <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice> .
<http://example.org/bob> <http://xmlns.com/foaf/0.1/name> "Bob"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/charlie>; . <http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/bob> .
<http://example.org/charlie> <http://xmlns.com/foaf/0.1/name> "Charlie"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice>; . <http://example.org/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
<http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
```

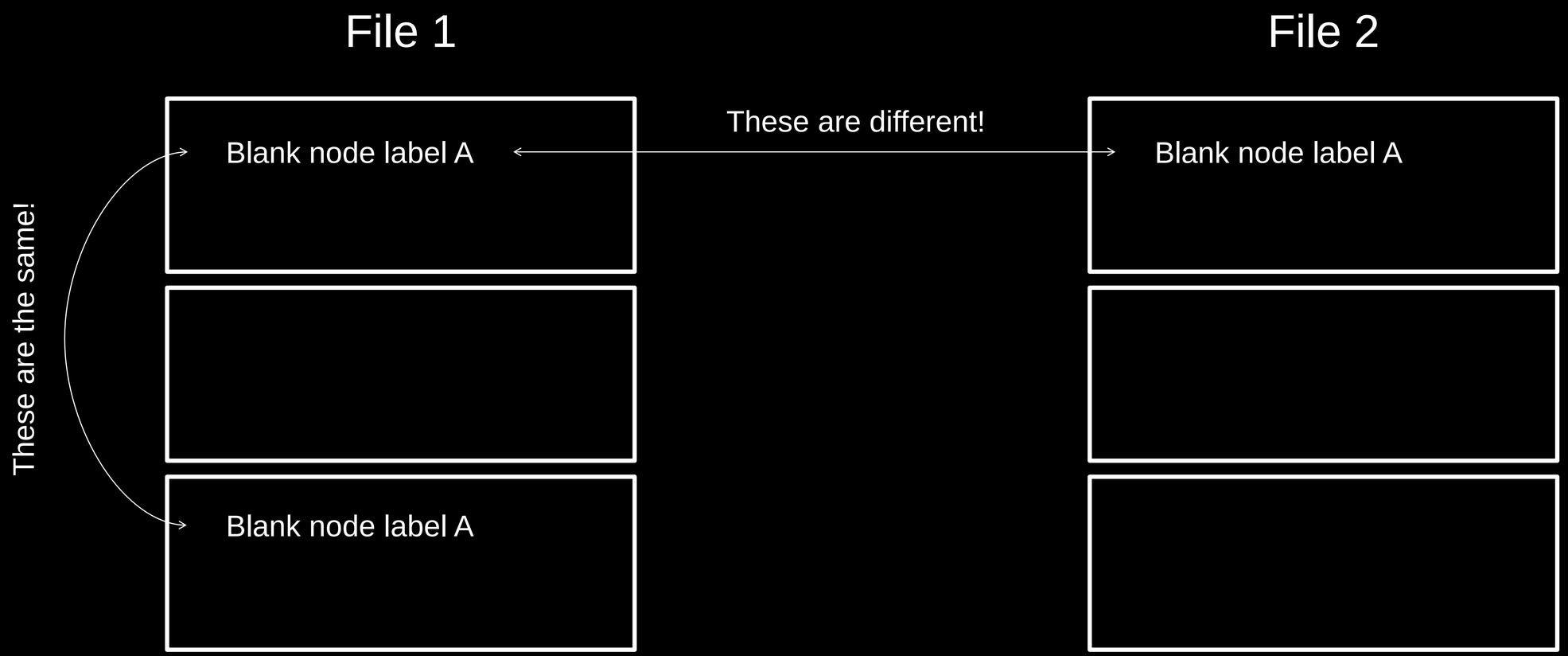
Turtle and adjacency lists

```
<http://example.org/alice> <http://xmlns.com/foaf/0.1/mbox> <mailto:alice@example.org>;
<http://xmlns.com/foaf/0.1/name> "Alice"; <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>; <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie>,
<http://example.org/bob>, <http://example.org/snoopy>; . <http://example.org/charlie> <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice> .
<http://example.org/bob> <http://xmlns.com/foaf/0.1/name> "Bob"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/charlie>; . <http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/bob> .
<http://example.org/charlie> <http://xmlns.com/foaf/0.1/name> "Charlie"; <http://xmlns.com/foaf/0.1/knows>
<http://example.org/alice>; . <http://example.org/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
<http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/charlie> .
```

Giraph

- Organize the subset of your RDF data you are interested in as adjacency lists (eventually using Turtle syntax)
- Apache Giraph is a good fit for shortest paths, PageRank and graph or iterative algorithms

Blank nodes


File 1
File 2
These are different!
Blank node label A
Blank node label A
These are the same!
Blank node label A

Blank nodes

```
public MapReduceAllocator (JobContext context, Path path) {
    this.runId = context.getConfiguration().get(Constants.RUN_ID);
    if ( this.runId == null ) {
           this.runId = String.valueOf(System.currentTimeMillis());
    }
    this.path = path;
}

@Override
public Node create(String label) {
    String strLabel = "mrbnode_" + runId.hashCode() + "_" + path.hashCode() + "_" + label;
    return Node.createAnon(new AnonId(strLabel)) ;
}
```

Inference

- For RDF Schema and subsets of OWL inference can be done via MapReduce:
  - use `DistributedCache` for vocabularies or ontologies
  - perform inference “as usual” in the map function
- WARNING: this does not work in general
- For RDFS and OWL ter Horst rule sets:
  - Urbani J., Kotoulas, S., …
    “WebPIE: a Web-scale Parallel Inference Engine”
    Submission to the SCALE competition at CCGrid 2010

Pig

- If you use Pig with Pig Latin scripts, create Pig input/output formats for N-Quads
- PigSPARQL, an interesting research effort:
  - Alexander Schätzle, Martin Przyjaciel-Zablocki, ...
    “PigSPARQL: Mapping SPARQL to Pig Latin”
    3th International Workshop on Semantic Web Information Management

Jena Grande

https://github.com/castagna/jena-grande

- A collection of utilities, experiments and examples on how to use MapReduce, Pig, HBase or Giraph to process data in RDF format
- Apache Jena is a Java library to parse, store and query RDF data
- Experimental and work in progress

QUESTIONS?

There is no such a thing as a stupid question!