UNIVERSITY OF JYVÄSKYLÄ

Lecture 6: Semantic Annotation and Linked Data

TIES452 Practical Introduction to Semantic Technologies
Autumn 2014

University of Jyväskylä

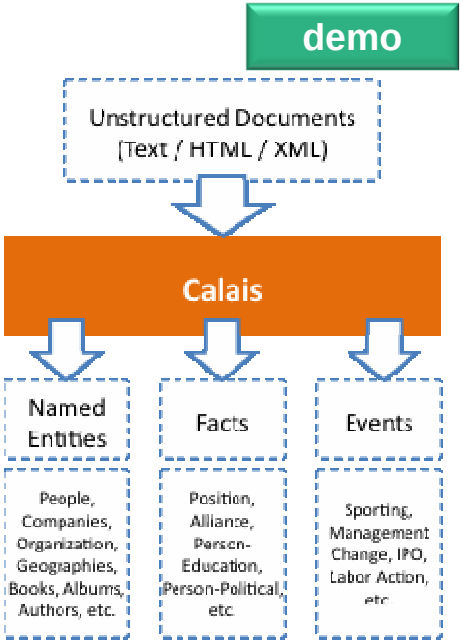
Khriyenko Oleksiy

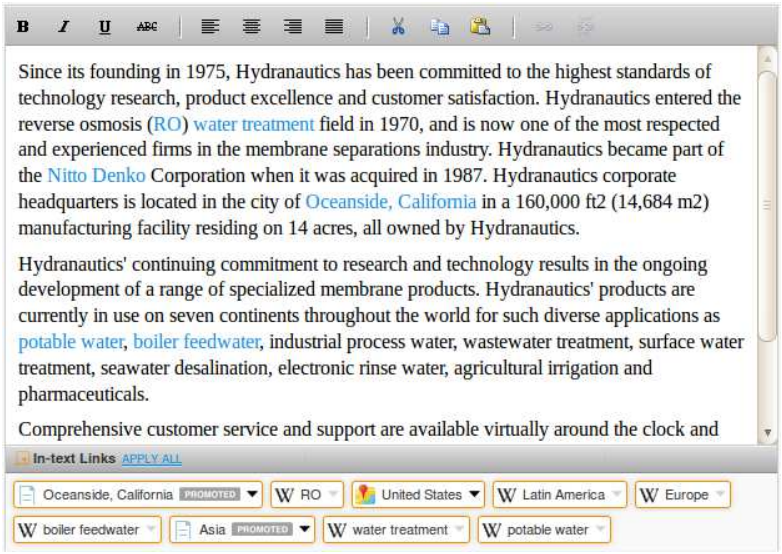
Part 1

Semantic Annotation

OpenCalais and Zemanta

- ***OpenCalais Web Service*** automatically creates rich semantic metadata for the content you submit. Using natural language processing (NLP), machine learning and other methods, Calais analyzes your document and finds the entities and returns the facts and events hidden within your text.
- But, *OpenCalais* is difficult to customize and has variable domain-specific coverage.
- Link: http://www.opencalais.com
- Demo: http://viewer.opencalais.com





- ***Zemanta*** is another general-purpose semantic annotation tool. Zemanta is used by bloggers and other content publishers to find links to relevant articles and media.
- Link: http://www.zemanta.com

DBpedia Spotlight

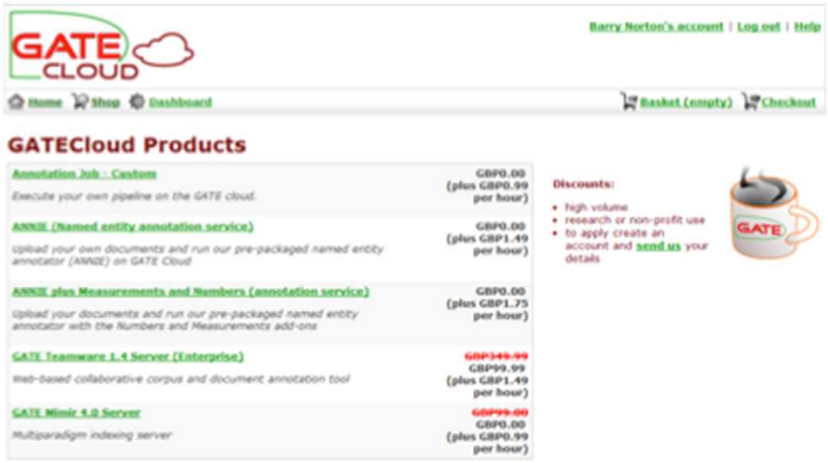
- ***DBpedia Spotlight*** is a tool for automatically annotating mentions of *DBpedia* resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through *DBpedia*. (http://spotlight.dbpedia.org)
- Try out DBpedia Spotlight through Web Application or Web Service endpoints:
  - The *Web Application* is a user interface that allows you to enter text in a form and generates an HTML annotated version of the text with links to DBpedia.
  - The *Web Service endpoints* provide programmatic access to the demo, allowing you to retrieve data also in XML or JSON.
- Demo: *http://dbpedia-spotlight.github.io/demo/*

GATE

- ***GATE*** *(General Architecture for Text Engineering)* is an open-source framework for text engineering. Started in 1996, *GATE* has a large developer community and can be more readily customized for text annotation in different domains and for different purposes. *GATE* is used worldwide to build bespoke solutions by organizations including the Press Association and National Archive. Information extraction is supported in many languages. (https://gate.ac.uk)

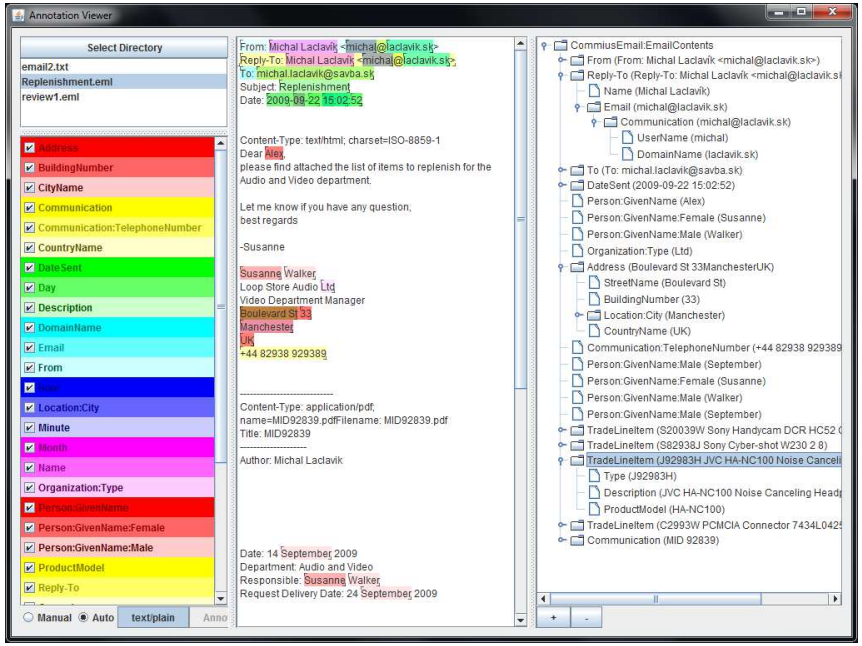
- There is a possibility to build a *GATE processing pipeline* specifically for your domain. For this we take an RDF dataset and use this to produce what is called a *GATE Gazetteer*, which is a list of entities in a domain and associated text labels used to refer to those entities. We can produce a gazetteer using the RDF data from chosen domain.

- A *GATE pipeline* can be run locally or uploaded to the *GATE cloud*. Once set up, text can be submitted and then annotated using the domain specific data. The annotated text can then be output in a format such as RDFa. (*https://gatecloud.net/*).



OnTeA

- ***OnTeA*** *(Ontology based Text Annotation)* is a Pattern based Semantic Annotation Platform. OnTeA search or create semantic meta data from text or documents using pattern based approaches. The Platform contains also graphical user interface, which shows identified objects in the text of email message or text file. The Platform also analyses HTML, PDF and Word email attachments.
- OnTeA uses two main techniques for information extraction:
  - patterns based on regular expressions
  - gazetteers: place names, locations, days of the weeks, etc. (now working with GATE or OntoText gazetteers).
- Link: http://ontea.sourceforge.net/



demo

OnTeA

- Motivation: *to extract semantic meta data from texts or documents…*
- Pattern based approach:
  - Unstructured content is also contains some patterns to be recognized…
  - Patterns are used to extract various objects that can be transformed to ontology class individuals and their properties…

Text	Patterns – regular expressions	Key - value
Nokia, Oy	*Company*: ([A-Za-z0-9]+)[,]+(Inc|Ltd|Oy)	*Company*: Nokia
info@dna.fi	*Email:* [-_.a-z0-9]+@[-_.a-zA-Z0-9]+\.[a-z]{2,8}	*Email*: info@dna.fi
Dr. Oleksiy Khriyenko	*Person:* (Mr.|Mrs.|Dr.) ([A-Z][a-z]+ [A-Z][a-z]+)	*Person*: Oleksiy Khriyenko
…	…	…

Example:

Text - ***Helsinki is the capital of Finland. Finland is in Europe.***

Patterns:

Location: *(in\|by) + (the)? *([A-Z][a-z]+)*

Country: *(capital of) + *([A-Z][a-z]+)*

City: *([A-Z][a-z]+) + (is) + (the)? + (capital of)*

Continent: <Country> *+ (is in) + (the)? *([A-Z][a-z]+)*

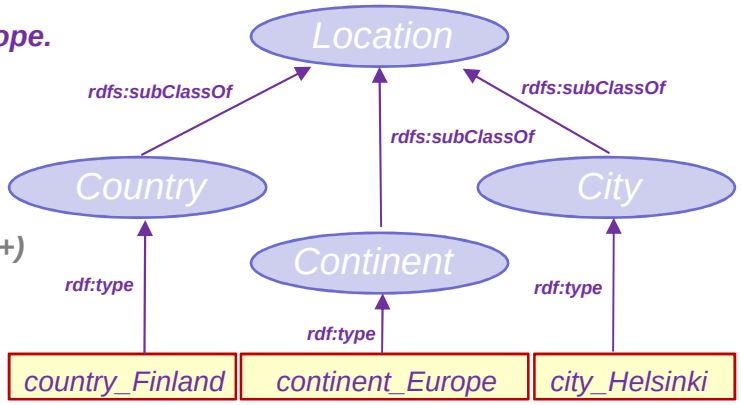
OnTeA key – value pairs:

Location: *Europe*

Country: *Finland*

City: *Helsinki*

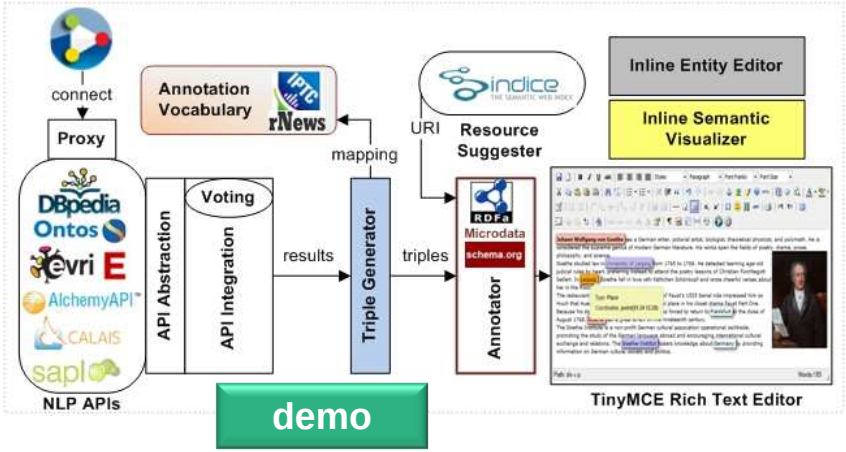
Continent: *Europe*



RDFaCE

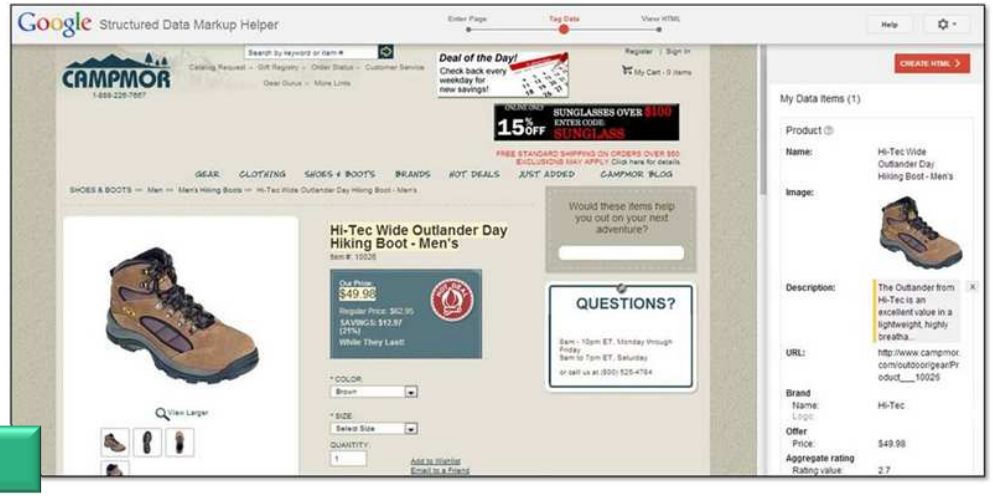
- ***RDFaCE*** *(RDFa Content Editor)* is a Semantic content editor based on TinyMCE WYSIWYG editor. *RDFaCE* is an implementation for WYSIWYM (What You See Is What You Mean) concept (***http://youtu.be/wxtlAol4HB0***). WYSIWYM aims to enable end-users to easily annotate their content using *RDFa* and *Microdata* markups based on *Schema.org* vocabularies. (***http://rdface.aksw.org***) (***http://youtu.be/W5CdPq0C1GU***)
- *RDFaCE* supports automatic content annotation employing *Sindice*, *Swoogle* and *Prefix.cc* APIs for resource suggestion (providing appropriate URIs for subjects, properties and namespaces) as well as using external *NLP APIs* (*Alchemy, Extractiv, Open Calais, Ontos, Evri, Saplo, Lupedia* and *DBpedia spotlight*).
- Available as a plugin for *WordPress* blogging platform (http://wordpress.org/plugins/rdface/).
- *RDFaCE –lite* is based on lite-weight version of RDFa. It supports RDFa and Microdata on *rNews schema* (limited to news-specific metadata: person, location, organization entities, etc.)
- Demos:

http://rdface.aksw.org/demo/
http://rdface.aksw.org/lite/test/tinymce/examples/rdface_lite.html
http://rdface.aksw.org/test/tinymce/examples/rdfaDemo.html



Structured Data Markup Helper

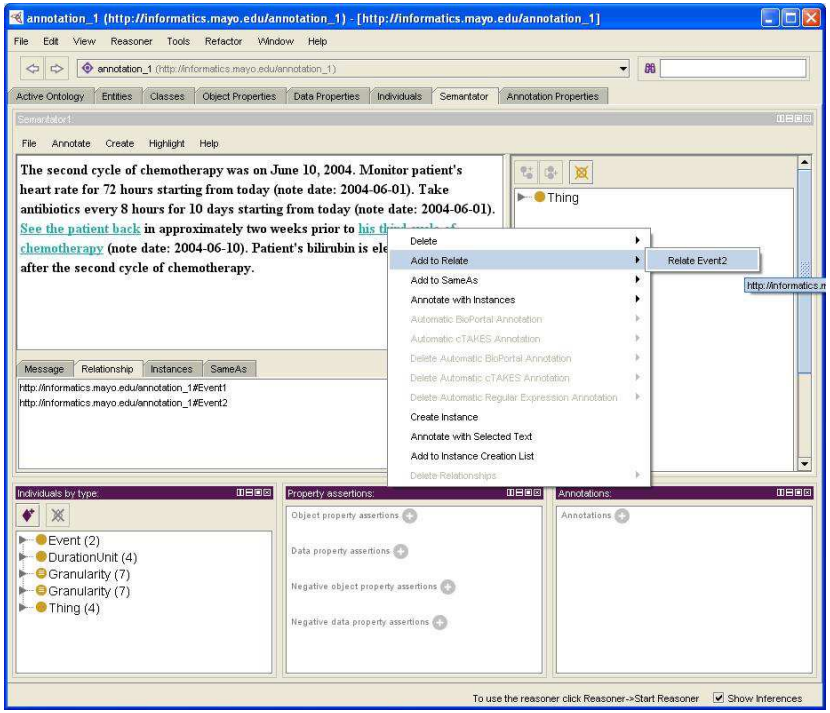
- ***Structured Data Markup Helper*** helps to update a site with on-page markup that enables search engines (e.g. Google, Bing, Yahoo!, Yandex) and other products to understand the information on web pages and provide richer search results in order to make it easier for users to find relevant information on the web. (https://www.google.com/webmasters/markup-helper/)
- Markup Helper uses *microdata* and *JSON-LD* formats with the *schema.org* vocabulary (a collaboration by Google, Microsoft, and Yahoo! to improve data description and interoperability on the web).
- Helpful links:
  - About Markup Helper: https://support.google.com/webmasters/answer/3069489?hl=en
  - Microdata: http://www.w3.org/TR/microdata/
  - JSON-LD: http://json-ld.org/
  - Schema.org: http://schema.org/



demo

Semantator

- ***Semantator*** *(Semantic Annotator)* is a tool developed in Mayo Clinic for users to semantically annotate data of interest with respect of domain ontologies in plain text. (http://informatics.mayo.edu/CNTRO/index.php/Semantator)
- *Semantator* is implemented as a Protege plug-in that allows users to view the ontology used for annotation, and the annotation results in the same environment. *Semantator* provides two modes:
  - *Manual annotation.* Expert can choose a document to be annotated and a domain ontology, highlight different pieces of information from the original text, and then mark which ontology concepts the information belongs to, link the instances together using the properties defined in the domain ontology.
  - *Semi-automatic annotation*. Users can choose to use different automatic annotation tools such as the National Center for Biomedical Ontologies (NCBO) annotator and Mayo Clinic's Clinical Text Analysis and Knowledge Extraction System (cTAKES), which are well-acknowledged tools for annotating biomedical and clinical text. Annotation results can be reviewed and modify as needed.

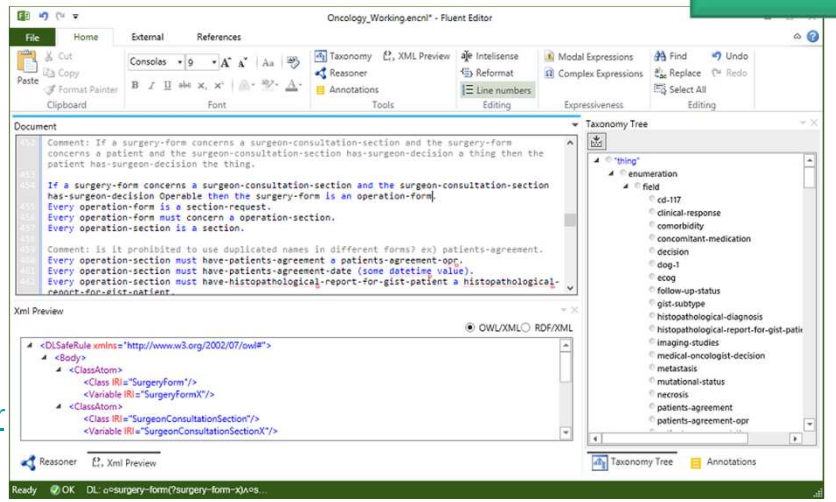


- Useful readings:
  - http://www.sciencedirect.com/science/article/pii/S1532046413001020#
  - http://swat.cse.lehigh.edu/pubs/song12a.pdf

Fluent Editor

- ***Fluet Editor*** is an comprehensive tool for editing and manipulating complex ontologies that uses *Controlled Natural Language (CNL):*
  - allows natural-language creation and editing of ontologies;
  - provides a more user-friendly alternative to XML-based OWL editors;
  - uses of *Controlled English* as a *knowledge modeling language*;
  - supported via *Predictive Editor*, it stops the user from entering any sentence that is grammatically or morphologically incorrect and actively helps the user during sentence writing;
  - can be integrated with any other 3rd party tools compliant with W3C standards.

demo



- Links: http://www.cognitum.eu/semantics/FluentEditor
  http://semanticweb.org/wiki/Fluent_Editor

Fluent Editor

- ***Controlled Natural Language***
  - Machine (web) processable;
  - Understandable to humans;
  - Restricted grammar and vocabulary;
  - Simplifies editing and structure;
  - Reduces ambiguity and errors;
  - Full OWL2, SWRL and SPARQL compatibility.
- ***Embedded reasoner***
  - Constantly checks knowledge base for consistency;
  - Generates XML automatically;
  - Corrects on the fly, preventing errors and speeding work pace;
  - Generates OWL files automatically.
- ***Predictive Editor***
  - Assists user with editing ontologies in real time;
  - Ensures perfect logic quality of ontologies;
  - Significantly increases editing speed;
  - Provides type-ahead hints and auto correct.
- ***With Fluent Editor™ business users can***
  - Import Existing ontologies from OWL files;
  - Export custom ontologies to OWL format;
  - Export custom rules to SWRL format;
  - Integrate custom software with CNL API;
  - Integrate with distributed semantic knowledge databases.

Fluent Editor

The same SWRL is verbalized in ***pure OWL/XML*** and ***Fluent Editor™***

```
<Body>
   <ClassAtom>
      <Class IRI="#Consent" />
      <Variable IRI="#Consent_0" />
   </ClassAtom>
   <ClassAtom>
      <Class IRI="#Patient" />
      <Variable IRI="#Patient_0" />
   </ClassAtom>
   <ClassAtom>
      <Class IRI="#Therapy" />
      <Variable IRI="#Therapy_0" />
   </ClassAtom>
   <ObjectPropertyAtom>
      <ObjectProperty IRI="#isRecommendedTo" />
      <Variable IRI="#Therapy_0" />
      <Variable IRI="#Patient_0" />
   </ObjectPropertyAtom>
   <ObjectPropertyAtom>
      <ObjectProperty IRI="#signs" />
      <Variable IRI="#Patient_0" />
      <Variable IRI="#Consent_0" />
   </ObjectPropertyAtom>
</Body>
<Head>
   <ObjectPropertyAtom>
      <ObjectProperty IRI="#isAppliedTo" />
      <Variable IRI="#Therapy_0" />
      <Variable IRI="#Patient_0" />
   </ObjectPropertyAtom>
</Head>
```

SWRL rule in Fluent Editor™:

```
If a patient signs a consent and a therapy is-recommended-to the patient then the therapy is-applied-to the patient.
```

Asking questions in Fluent Editor™:

```
Who-Or-What is a city that belongs-to Texas-State and has-latitude greater-or-equal-to 0?
```

or

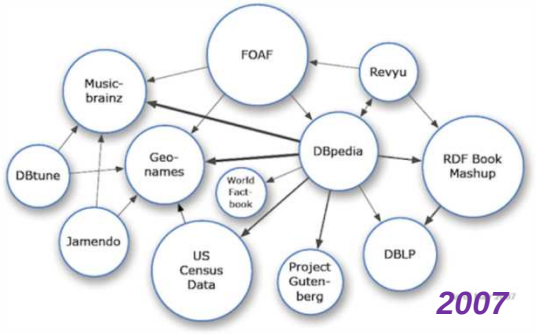
```
Who-Or-What is a customer that lives-in a city that belongs-to California-State and has-firstname equal-to 'John'?
```

Part 2

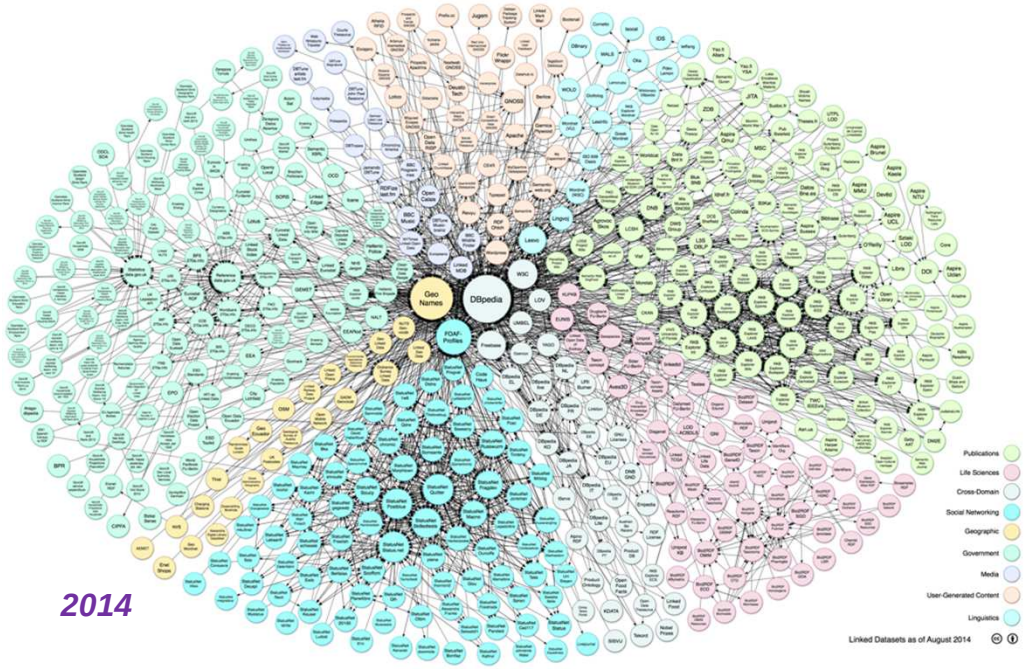
Linked Data

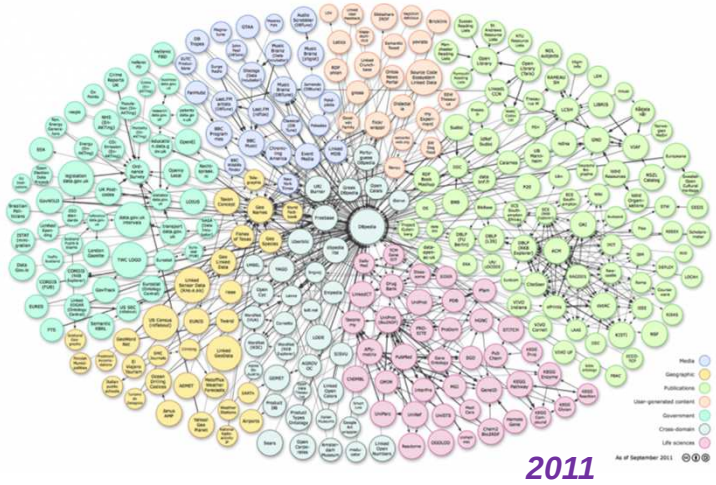
Linked Data

- In 2006, Tim Berners-Lee set out four simple principles for publishing data on the web. (http://linkeddata.org)
  - Use URIs to identify things.
  - Use HTTP URIs so that people can look up those names.
  - When someone looks up a URI, provide useful information, using the standards (RDF, RDFS, SPARQL).
  - Include links to other URIs, so that they can discover more things.



volume of data has grown from around 2 billion triples in 2007 to over 30 billion in 2011…





In 2014, altogether, the diagram contains 570 datasets and 2909 linkage relationships between the datasets...

Cool URIs – what's the problem?

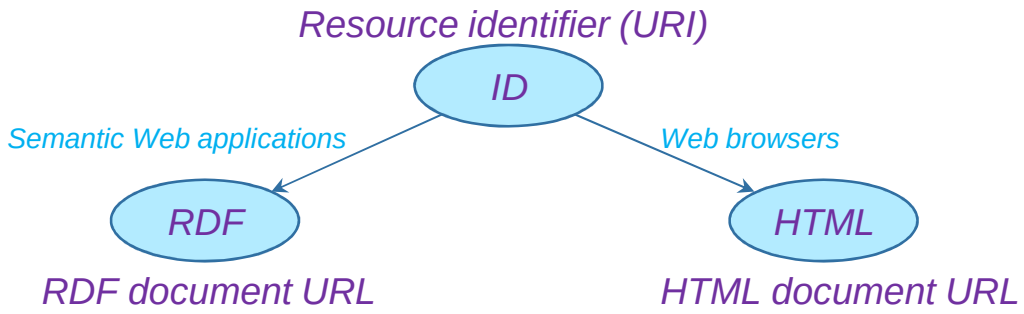
- W3C note from 2008 (http://www.w3.org/TR/cooluris/)
- URIs identify concepts *(real-world objects)/(non-information resource)*
- At the same time, web documents have always been addressed with URIs
- What URIs should we use in our RDF documents?
- Problem:
  - Alice is a real person that has a web page
  - What URI should represent Alice as an individual?
    - Her web page URL?
    - Her email address?

```
<URI-of-alice> a foaf:Person;
    foaf:name "Alice";
    foaf:mbox <mailto:alice@example.com>;
    foaf:homepage <http://example.com/people/alice> .
```

- Crucial concept: HTTP content negotiation

Cool URIs rules

- Be on the Web
  - Given only a URI of a resource, both machines and people will get the description of the resource
  - People will get human-readable HTML page
  - Machines will get RDF data
- Be unambiguous
  - No confusion between identifiers for Web documents and identifiers for other resources
  - One URI can't stand for both a Web document and real-world object (RWO)
- So which URI for Alice???



Cool URIs: good practice

- The URIs related to a single real-world object (non-information resource):
  - *resource identifier*
  - *HTML document URL*
  - *RDF document URL*

- Several ideas for choosing related URIs:

`http://smith-family.com/resource/alice`	- Identifier for Alice, the person
`http://smith-family.com/page/alice`	- Alice's homepage
`http://smith-family.com/data/alice`	- RDF document with description of Alice
`http://id.smith-family.com/alice`	- Identifier for Alice, the person
`http://pages.smith-family.com/alice`	- Alice's homepage
`http://data.smith-family.com/alice`	- RDF document with description of Alice
`http://smith-family.com/alice`	- Identifier for Alice, the person
`http://smith-family.com/alice.html`	- Alice's homepage
`http://smith-family.com/alice.rdf`	- RDF document with description of Alice

Solution: 303 URIs

- Use HTTP redirect status code ***303 See Other***
  - to distinguish non-document resources from regular web documents
  - to point to the proper human-readable document

HTTP request:

```
GET /page/alice HTTP/1.1
Host: www.acme.com
Accept: text/html
Accept-Language: en, de
```

HTTP response (web document):

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Language: en
```

HTTP request:

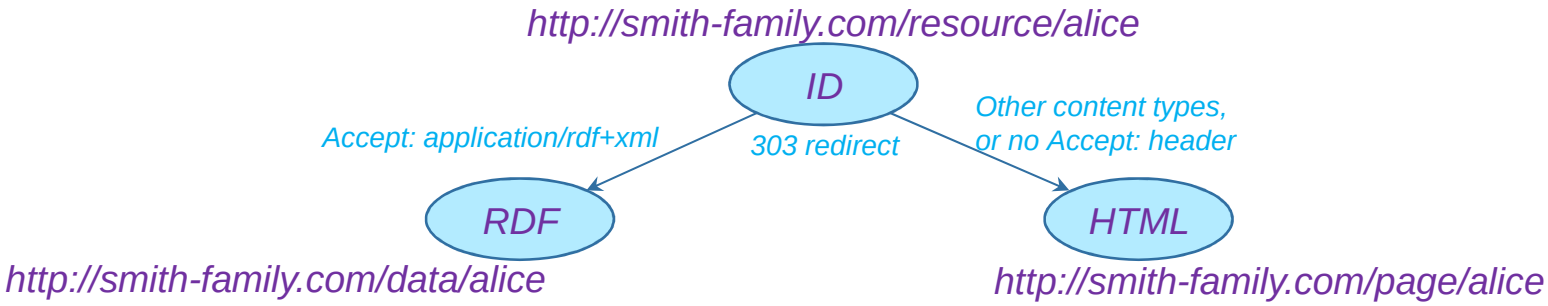
```
GET /resource/alice HTTP/1.1
Host: www.acme.com
Accept: application/rdf+xml
```

HTTP response (content negotiation):

```
HTTP/1.1 303 See Other
Location: http://www.acme.com/data/alice.rdf
Vary: Accept
```

Solution: 303 URIs

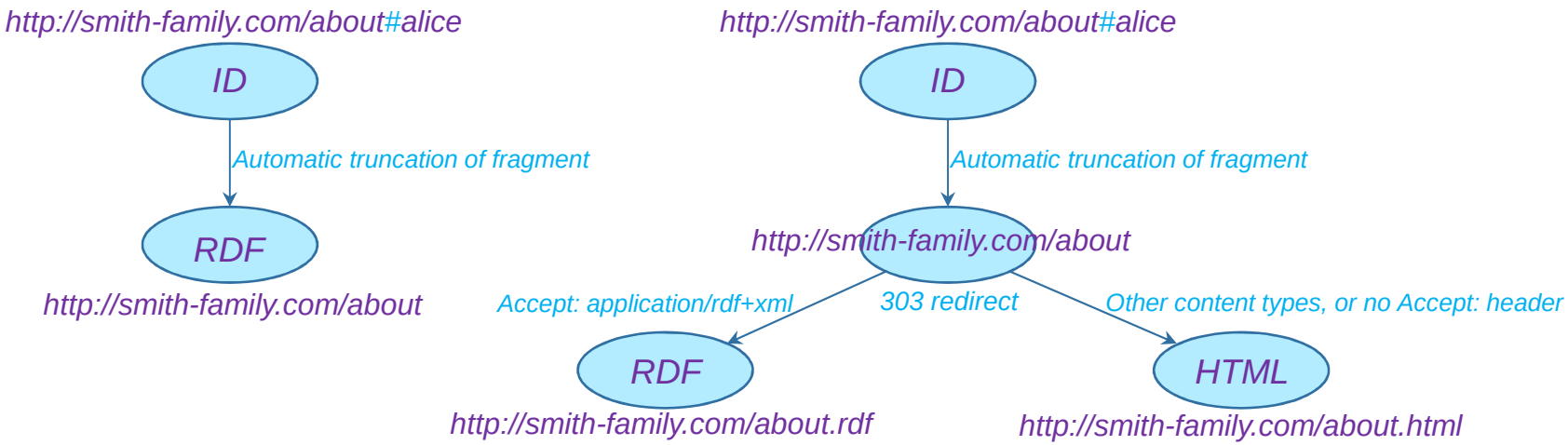
- Alice, the person (RWO)
  - Link: `http://smith-family.com/resource/alice`
  - Machine access -> redirect to an RDF file http://smith-family.com/data/alice
  - Human access -> redirect to address http://smith-family.com/page/alice



- Alice, the document describing her (Web document)
  - Link: `http://smith-family.com/page/alice`
  - Human access -> returns HTML page
  - Machine access -> returns RDF description of the *web page* or an error (page not found)
    - This could be a URI for the web page about Alice
    - This is not URI for Alice as a person

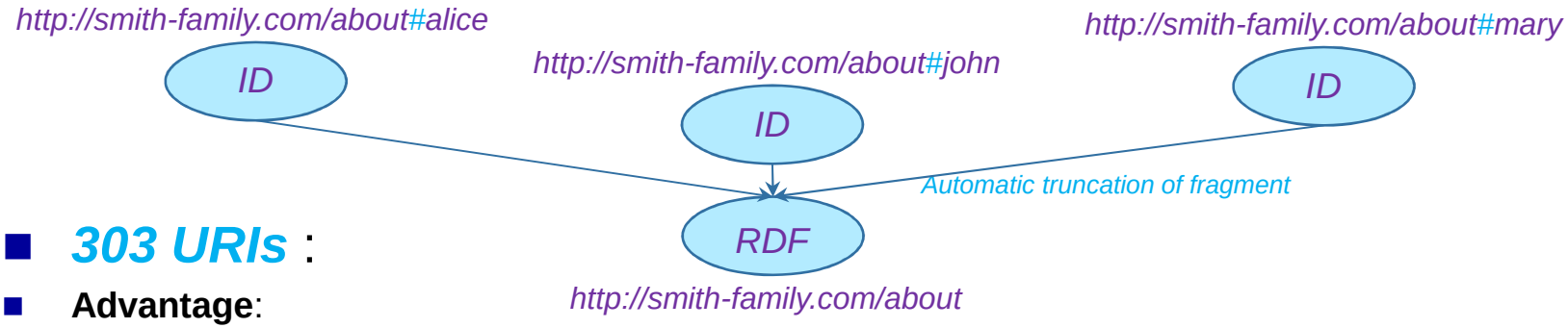
Solution: Hash URIs

- Use URIs with fragments (#) for non-information resources
  - URI with a hash cannot be retrieved directly (it is required to strip off the fragment part) and therefore cannot identify a web document
  - we can use them to identify other, non-information resources
- Alice, the person (RWO)
  - Link: `http://smith-family.com/about#alice`
  - Machine access -> returns http://smith-family.com/about as RDF file (which contains info about Alice)
  - Human access -> returns http://smith-family.com/about as HTML file



Hash URIs vs. 303 URIs

- ***Hash URIs*** :
- **Advantage**:
  - reduced number of necessary HTTP requests
  - a family of URIs can share the same non-hash part
- **Disadvantage:**
  - it loads other unrequested data



- ***303 URIs*** :
- **Advantage**:
  - redirection target can be configured separately for each resource
  - there could be one describing document for each resource, or one large document for all of them, or any combination in between.
- **Disadvantage:**
  - the large number of redirects may cause higher latency (waiting time)

Cool URIs: good practice

- All the URIs related to a single real-world object – *resource identifier*, *RDF document URL*, *HTML document URL* – should be explicitly linked with each other to help information consumers understand their relation.

`http://smith-family.com/resource/alice`	- Identifier for Alice, the person
`http://smith-family.com/page/alice`	- Alice's homepage
`http://smith-family.com/data/alice`	- RDF document with description of Alice

RDF file from *http://www.smith-family.com/data/alice*

```
...
<http://smith-family.com/resource/alice>
                    foaf:page <http://smith-family.com/page/alice>;
                    rdfs:isDefinedBy <http://smith-family.com/data/alice>;
                    a foaf:Person;
                    foaf:name "Alice";
                    foaf:mbox <mailto:alice@acme.com>;
...
```

HTML file from *http://www.smith-family.com/people/alice*

```
<html lang="en">
   <head>
      <title>Alice's Homepage</title>
      <link rel="alternate" type="application/rdf+xml"
         title="RDF Version"
         href="http://smith-family.com/data/alice" />
   </head> ...
```

Linked Open Data

- In 2010, Tim Berners-Lee suggested a ***5 star*** deployment scheme for ***Open Data*** to encourage people (especially government data owners) to improve linked data.
- Linked *Open* Data (LOD) is Linked Data which is released under an open license, which does not impede its reuse for free. *LOD2* - http://lod2.eu/Welcome.html

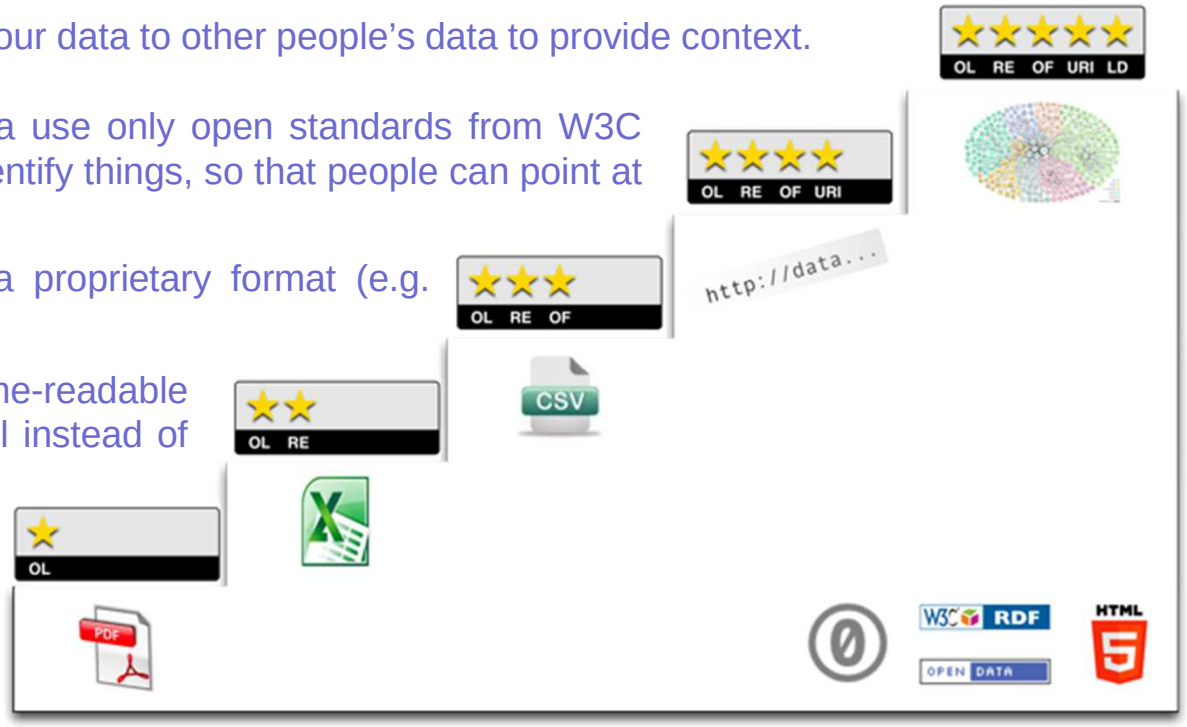
All the before, plus: Link your data to other people's data to provide context.

All the previous plus, data use only open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff.

The data does not use a proprietary format (e.g. CSV instead of excel).

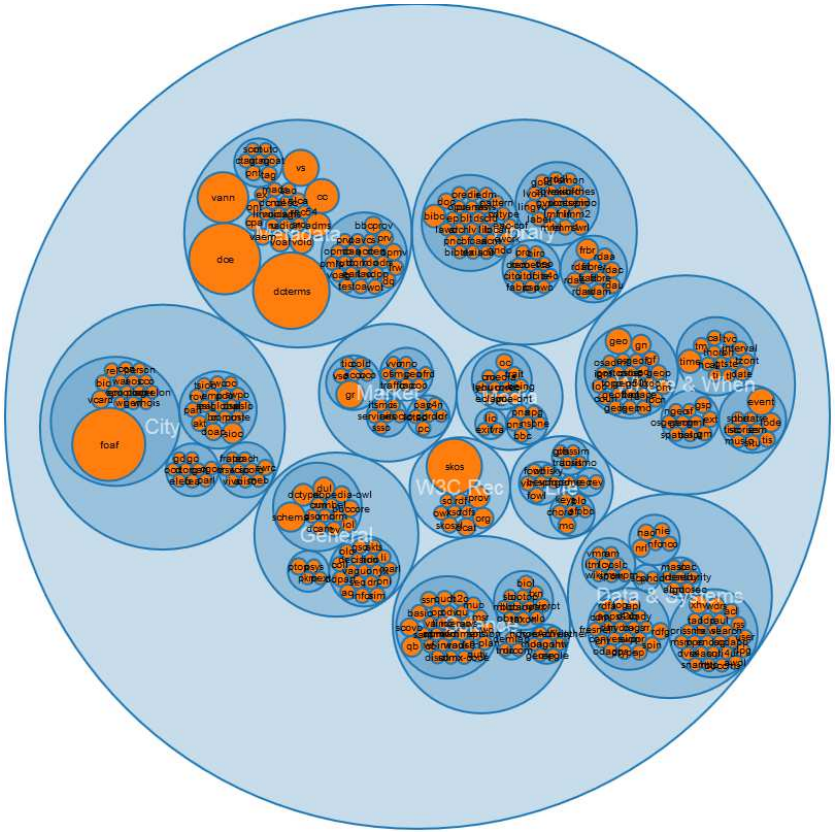
Available as machine-readable structured data (e.g. excel instead of image scan of a table).

Available on the web (whatever format) *but with an open license, to be Open Data*.



Linked Data: good practice

- *Reusing existing well-known vocabularies*. In order to make it possible for client applications to process your data, you should *reuse terms from well-known vocabularies* wherever possible. You should only define new terms yourself if you can not find required terms in existing vocabularies. It is common practice to *mix terms* from different vocabularies.
  - ***Friend-of-a-Friend (FOAF)*** provides terms for describing people and their social network
  - ***SIOC*** Semantically-Interlinked Online Communities
  - ***DOAP*** Description of a Project
  - ***Dublin Core*** Defines general metadata attributes.
  - ***SKOS*** Simple Knowledge Organization System
  - ***SKOS DataZone*** list of vocabularies available in SKOS schema
  - ***Review Vocabulary*** provides terms for representing reviews.
  - ***GoodRelations*** provides terms for describing products and business entities.
  - ***Music Ontology*** provides terms for describing artists, albums, tracks, but also performances, arrangements, etc.
  - ***Organization Ontology*** for describing the structure of organizations.
  - ***Linking Open Description of Events (LODE)*** provides terms for describing the basic properties of an event and contains a list of axioms expressing mapping relationships with other ontologies such as DOLCE, CYC, CIDOC-CRM, Event Ontology, F, and SEM.
  - Google, Yahoo and Microsoft have agreed on vocabularies for publishing structured data on the Web. Their shared 'ontology' is maintained on ***schema.org***.
  - ***MarineTLO (core) Ontology*** is a top-level ontology for the marine domain (also applicable to the terrestrial domain) and ***MarineTLO (imarine) Ontology*** is an extension and operational version of the MarineTLO core.
  - etc.

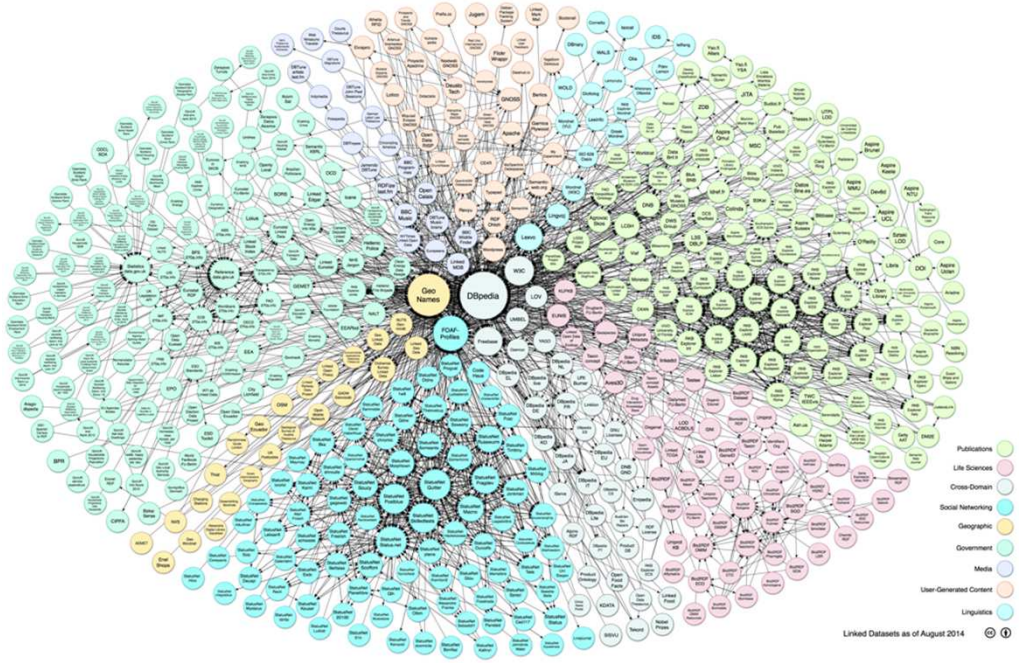


Linked Open Vocabularies: http://lov.okfn.org/dataset/lov/

Well-known vocabularies: http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies

Linked Data: good practice

- *Reusing existing URIs*. If you need URI references for *geographic places, research areas, general topics, artists, books or CDs*, you should consider using URIs from existing data sources (for instance *Geonames*, *DBpedia*, *Musicbrainz*, *dbtune*, *RDF Book Mashup*, etc.). The two main benefits of using URIs from such data sources are:
  - The URIs are dereferenceable, meaning that a description of the concept can be retrieved from the Web.
  - The URIs are already linked to URIs from other data sources.



Well-known Data Sets: http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets
Linked Data Sets available as RDF Dumps: http://www.w3.org/wiki/DataSetRDFDumps
SparqlEndpoints list: http://www.w3.org/wiki/SparqlEndpoints

Linked Data: good practice

Guidance for *own term definition*:

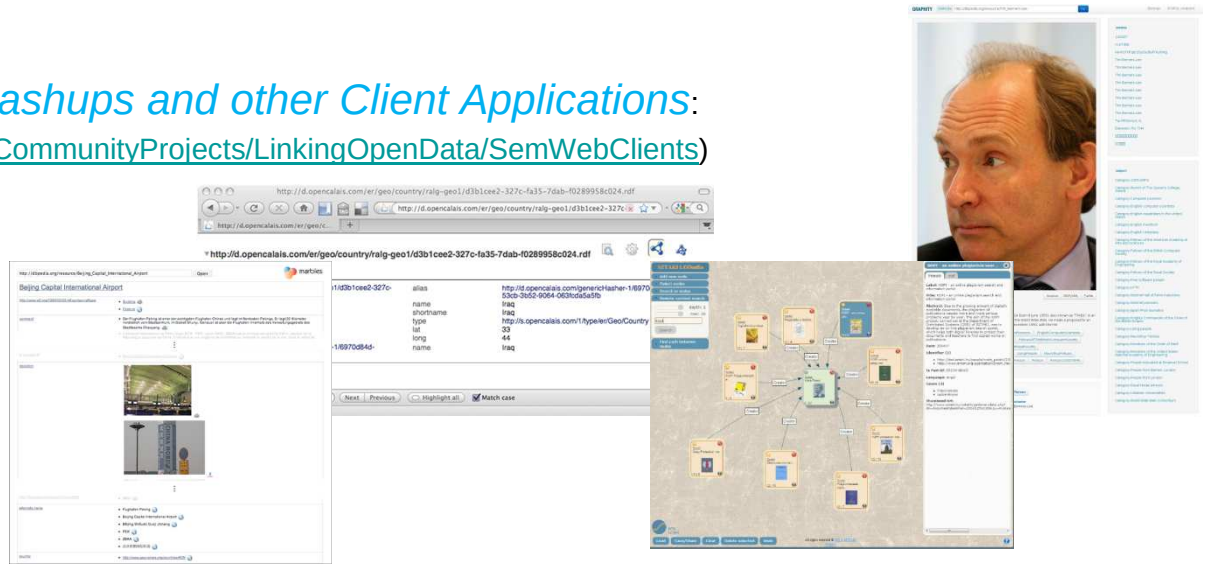
- *Do not define new vocabularies from scratch*, but complement existing vocabularies with additional terms (in your own namespace) to represent your data as required.
- *Provide for both humans and machines.* At this stage in the development of the Web of Data, more people will be coming across your code than machines, even though the Web of Data is meant for machines in the first instance. Don't forget to add prose, e.g. *rdfs:comment* for each term invented. Always provide a label for each term using the *rdfs:label* property.
- *Make term URIs dereferenceable.* It is essential that term URIs are dereferenceable so that clients can look up the definition of a term. Therefore you should make term URIs dereferenceable following the W3C Best Practice Recipes for Publishing RDF Vocabularies (http://www.w3.org/TR/swbp-vocab-pub/).
- *Make use of other people's terms.* Using other people's terms, or providing mappings to them, helps to promote the level of data interchange on the Web of Data, in the same way that hypertext links built the traditional document Web. Common properties for providing such mappings are *rdfs:subClassOf* or *rdfs:subPropertyOf*.
- *State all important information explicitly.* For example, state all ranges and domains explicitly. Remember: humans can often do guesswork, but machines can't. Don't leave important information out!
- *Do not create over-constrained, brittle models; leave some flexibility for growth.* For instance, if you use full-featured OWL to define your vocabulary, you might state things that lead to unintended consequences and inconsistencies when somebody else references your term in a different vocabulary definition. Therefore, unless you know exactly what you are doing, use RDF-Schema to define vocabularies.

Web of Data Tools

Linked Data Browsers, Mashups and other Client Applications:

(http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients)

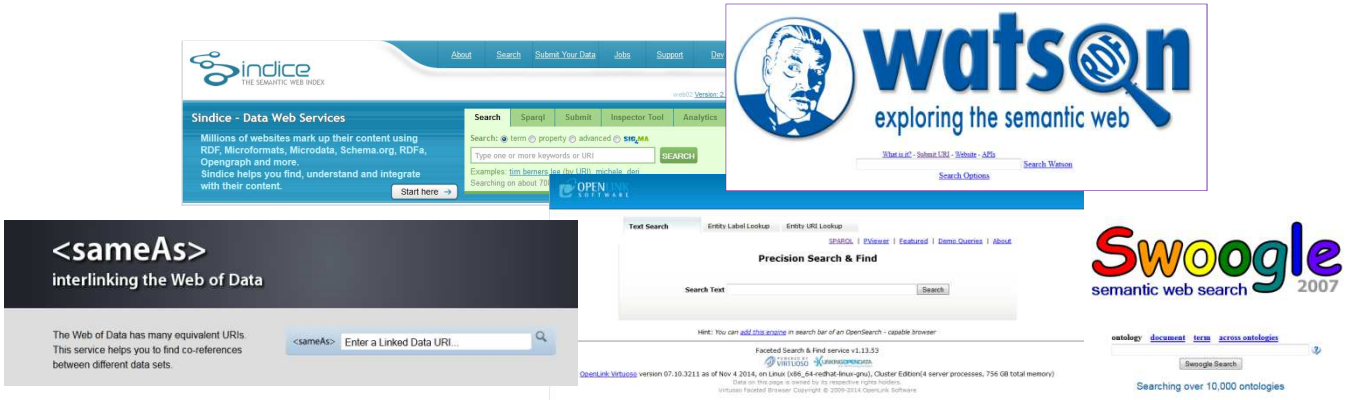
- *Tabulator*
- *OpenLink Data Explorer*
- *DBpedia Mobile*
- *Marbles*
- *Graphity Browser*
- *Quick & Dirty RDF Browser*
- *LODmilla*
- *Etc.*



Semantic Web Search Engines:

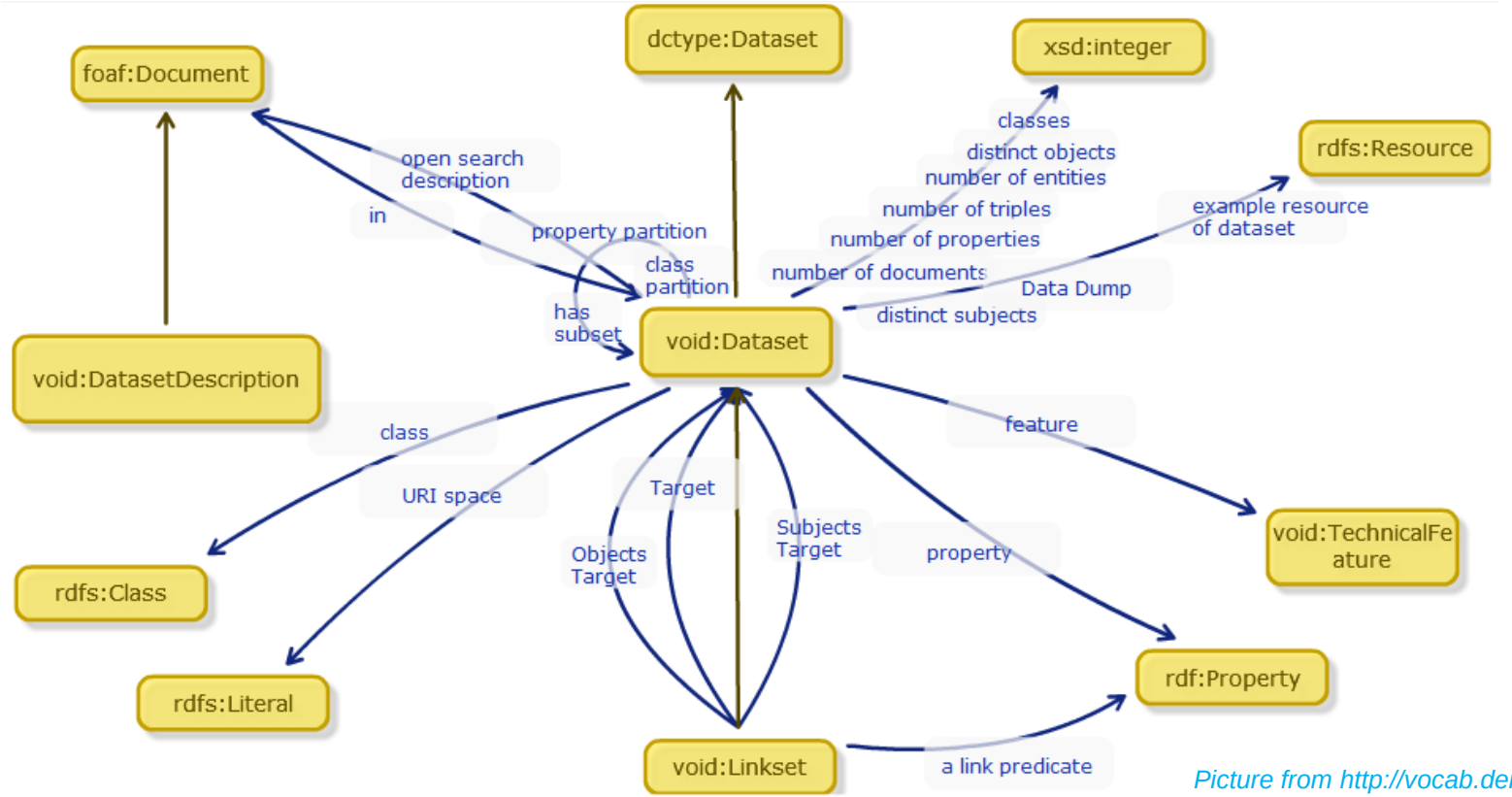
(http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/SemanticWebSearchEngines)

- *<sameAs.org>*
- *VisiNav*
- *Falcons*
- *Sindice*
- *Watson*
- *Swoogle*
- *Etc.*



VoID

- ***VoID*** - ***Vocabulary of Interlinked Datasets*** - is an RDF Schema vocabulary for expressing metadata about linked datasets.
- Documentation: http://www.w3.org/TR/void/
- Vocabulary: http://vocab.deri.ie/void



Picture from http://vocab.deri.ie/void

VoID: Datasets

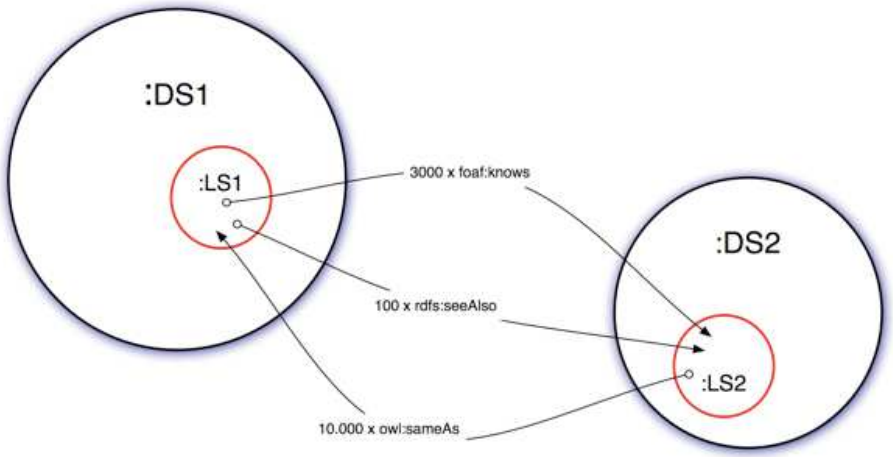
- Definition: ***void:Dataset***
- *Dataset is a collection of data which is:*
  - *published and maintained by a single provider*
  - *available as RDF*
  - *accessible, for example, through HTTP URIs or a SPARQL endpoint.*

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix void: <http://rdfs.org/ns/void#> .

:DBpedia rdf:type void:Dataset ;
         foaf:homepage <http://dbpedia.org/> .
:DBLP rdf:type void:Dataset ;
      foaf:homepage <http://www4.wiwiss.fu-berlin.de/dblp/all> ;
      dcterms:subject <http://dbpedia.org/resource/Computer_science> ;
      dcterms:subject <http://dbpedia.org/resource/Journal> ;
      dcterms:subject <http://dbpedia.org/resource/Proceedings> .
```

VoID: Linksets

- Definition: ***void:Linkset***
  - *Is a subclass of void:Dataset , used for storing triples to express the interlinking relationship (e.g. owl:sameAs or foaf:knows) between datasets*
  - *In each interlinking triple, the subject is a resource hosted in one dataset and the object is a resource hosted in another dataset*

```
:DBpedia void:subset :DBpedia2DBLP  .

:DBpedia2DBLP rdf:type void:Linkset ;
              void:linkPredicate owl:sameAs ;
              void:target :DBpedia ;
              void:target :DBLP .
```



Picture from http://semanticweb.org/wiki/VoID

VoID: SPARQL endpoints

- SPARQL endpoints: ***void:sparqlEndpoint***

```
@prefix void: <http://rdfs.org/ns/void#> .

:DBpedia a void:Dataset;
    void:sparqlEndpoint <http://dbpedia.org/sparql> .
```

- SPARQL query for available *SPARQL endpoints*:

```
PREFIX void: <http://rdfs.org/ns/void#>

SELECT DISTINCT ?endpoint
WHERE {
         ?ds a void:Dataset .
         ?ds void:sparqlEndpoint ?endpoint
      }
```

- See: http://void.rkbexplorer.com/sparql/

VoID: URI lookup endpoints

- URI lookup endpoints: ***void:uriLookupEndpoint***

```
@prefix void: <http://rdfs.org/ns/void#> .

:Sindice a void:Dataset;
    void:uriLookupEndpoint <http://api.sindice.com/v2/search?qt=term&q=>.
```

- *Endpoint Lookup Service* allows a URI(s) to be submitted, and returns SPARQL endpoint(s) which may serve information about the requested resource.
  - See: http://void.rkbexplorer.com/endpoint-search/

VoID: Technical features

- ***void:feature*** property can be used for expressing certain technical features of a dataset (e.g. supported RDF serialization formats). The domain of the property is *void:Dataset* and its range is *void:TechnicalFeature*.

```
:DBpedia a void:Dataset;
    void:feature <http://www.w3.org/ns/formats/RDF_XML> .
```

W3C URIs for formats are instances of class *http://www.w3.org/ns/formats/vocab-data/Format*, which is a sub-class of ***void:TechnicalFeature***.

- Customized definition of technical feature, e.g. HTTP features such as content negotiation or ETag headers…

```
:HTTPCachingETags a void:TechnicalFeature;
    rdfs:label "HTTP ETag support";
    rdfs:comment "the dataset supports HTTP caching using ETags";
    rdfs:seeAlso <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#> .
```

VoID: Distributed location

- If an RDF dump of the dataset is available, then its location can be announced using ***void:dataDump***. If the dataset is split into multiple dumps, then several values of this property can be provided.

```
:NYTimes a void:Dataset;
    void:dataDump <http://data.nytimes.com/people.rdf>;
    void:dataDump <http://data.nytimes.com/organizations.rdf>;
    void:dataDump <http://data.nytimes.com/locations.rdf>;
    void:dataDump <http://data.nytimes.com/descriptors.rdf> .
```

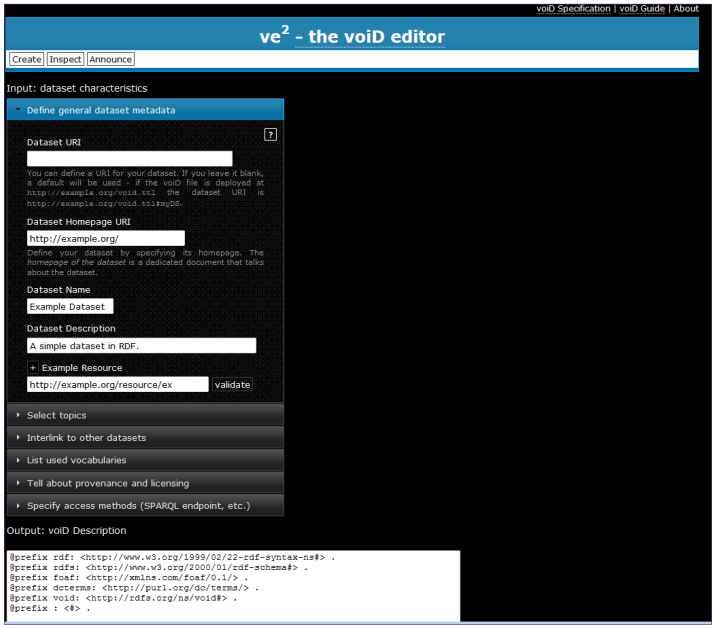
- ***void:subset*** property can be used to provide descriptions of parts of a dataset. A part of a dataset is itself a *void:Dataset*.

```
:DBpedia a void:Dataset;
    void:subset :DBpedia_shortabstracts;
    void:subset :DBpedia_infoboxes .
:DBpedia_shortabstracts a void:Dataset;
    dcterms:title "DBpedia Short Abstracts";
    dcterms:description "Short Abstracts of Wikipedia Articles";
    void:dataDump <http://downloads.dbpedia.org/3.3/en/shortabstract_en.nt.bz2> .
:DBpedia_infoboxes a void:Dataset;
    dcterms:title "DBpedia Infoboxes";
    dcterms:description "Information that has been extracted from Wikipedia infoboxes.";
    void:dataDump <http://downloads.dbpedia.org/3.3/en/infobox_en.nt.bz2> .
```

VoID: voiD Store

- ***voiD Store*** (http://void.rkbexplorer.com)
  - simply gathers a number of voiD documents and stores them in a repository
  - makes it easy for clients and applications to query these descriptions in order to identify which datasets may be of relevance for a particular need or request
  - makes it possible to find endpoints which may contain a given URI

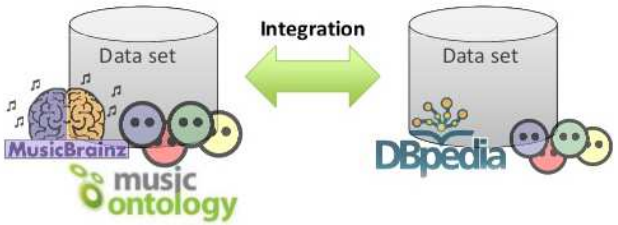
- Service contains:
  - voiD vocabulary (*http://vocab.deri.ie/void*)
  - URI to endpoint lookup
  - SPARQL query engine
  - voiD Editor – ***ve2***
  - etc.
- ***ve2***(*http://lab.linkeddata.deri.ie/ve2/*) allows to:
  - generate a voiD file in RDF Turtle format and define the characteristics of your linked dataset (categories, interlinking, technical features, licensing, etc.)
  - announce it to the wide world



Linked Data Integration

Linked Data applications that want to consume data from the global data space face following challenges:

- Data sources use a wide range of different RDF vocabularies to represent data about the same type of entity;
- The same real-world entity, for instance a person or a place, is identified with different URIs within different data sources;
- Data about the same real-world entity coming from different sources may contain conflicting value.



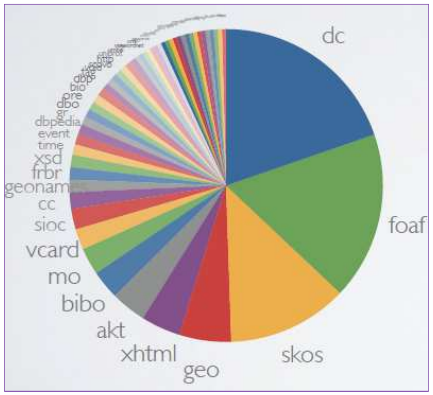
4 steps to Linked Data Integration:

Step#1: *Access linked data*: on-the-fly dereferencing, Query federation, Crawling and Caching.

Step#2: *Normalize vocabularies.* Schema Mapping can be performed based on rules or SPARQL queries.

Step#3: *Resolve identifiers.* Most LOD sources only provide *owl:sameAs* links to one other data source. Identity resolution can be done by manual merging or rule-based approaches (e.g. SILK, LIMES)

Step#4: *Filter Data.* Due to the different knowledge levels, views and intents of data sources as well as wrong, inconsistent or outdated information, data can be stored and queried separately using named graphs based structure of a storage.



Linked Data supporting Tools

- ***LDIF*** (Linked Data Integration Framework) integrates Linked Data from multiple sources into a clean, local target representation while keeping track of data provenance (http://ldif.wbsg.de/)

The *LDIF integration pipeline* consists of the following steps:

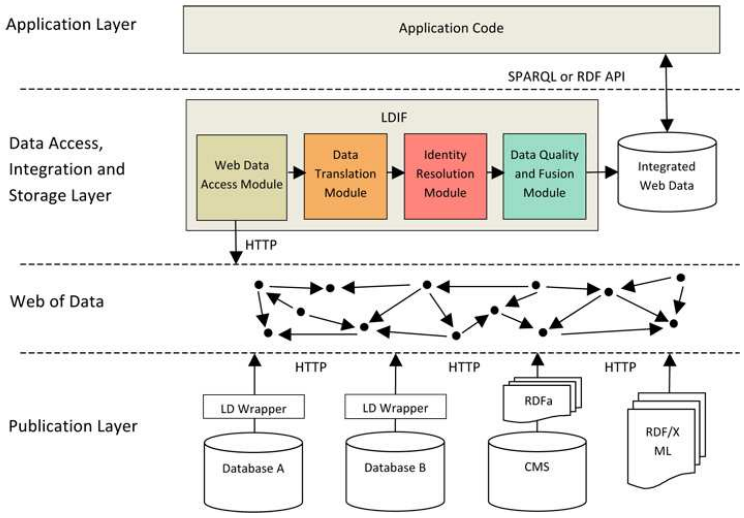
Step#1: *Collect data.* Import modules locally replicate data sets via file download, crawling or SPARQL. Supported data sources: RDF dumps (all common formats), SPARQL Endpoints, Crawling Linked Data via HTTP.

Step#2: *Translate data (map to Schema).* An expressive mapping language allows for translating data from the various vocabularies that are used on the Web into a consistent, local target vocabulary. LDIF supports simple mappings using OWL/RDFS statements (x rdfs:subClassOf y), complex mappings with SPARQL expressivity, built-in transformation function library (XPath) as well as *R2R Framework* (http://wifo5-03.informatik.uni-mannheim.de/bizer/r2r/).

Step#3: *Resolve identifiers.* An identity resolution component discovers URI aliases in the input data and replaces them with a single target URI based on user-provided matching heuristics. LDIF uses automated link creation based on SILK Link Specifications as well as supports various comparators and transformations (string similarity, basic arithmetics, time, geographical distance).

Step#4: *Cleanse data.* A data cleansing component filters data according to different quality assessment policies (assign quality scores to Named Graphs by time, by source preference, thresholds, etc.) and provides data fusion according to different conflict resolution methods (resolve conflicting property values according to quality scores, frequency, averages, etc.). LDIF employs *Sieve* (http://sieve.wbsg.de/).

Step#5: *Output.* LDIF outputs the integrated data in N-Quads, N-Triples or SPARQL Update Stream. For provenance tracking, LDIF employs the Named Graphs data model.



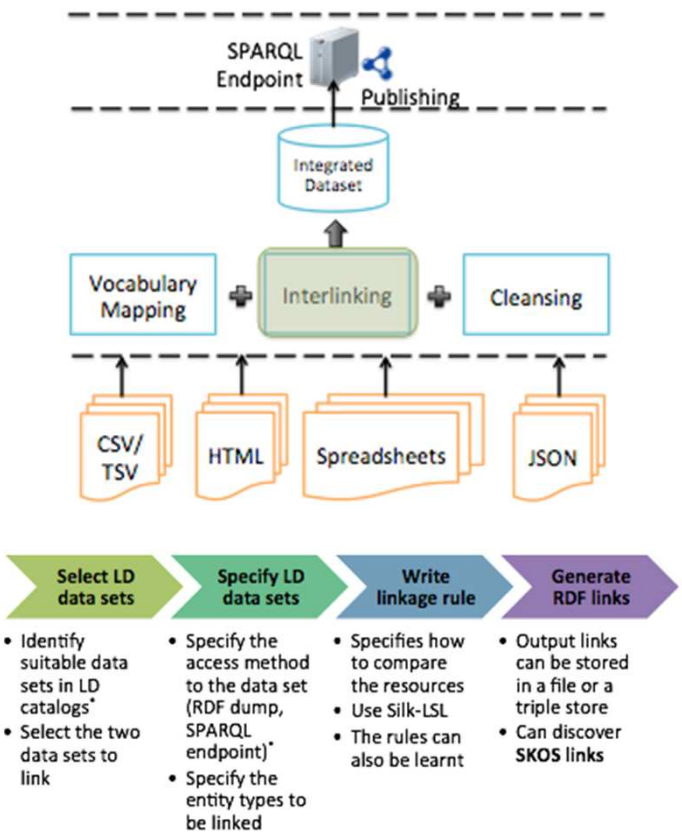
Linked Data supporting Tools

- ***SILK*** – a Link Discovery Framework for the Web of Data. SIKL is an open source tool for discovering RDF links between data items within different Linked Data sources. (http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/)
- ***Silk Link Specification Language*** *(Silk-LSL)* is used to define rules for linking entities from two different datasets. For example, a rule may express that if two entities belong to specified classes and have matching labels then they should be linked by a certain property. This property could be *owl:sameAs* or some other property such as *skos:closeMatch*.
- *SILKS* can run in different variations:

  Silk Single Machine:
  - Generate links on a single machine
  - Local or remote data set

  Silk MapReduce:
  - Generate RDF links using a cluster of multiple machines
  - Based on Hadoop (Can be run on Amazon Elastic MapReduce)

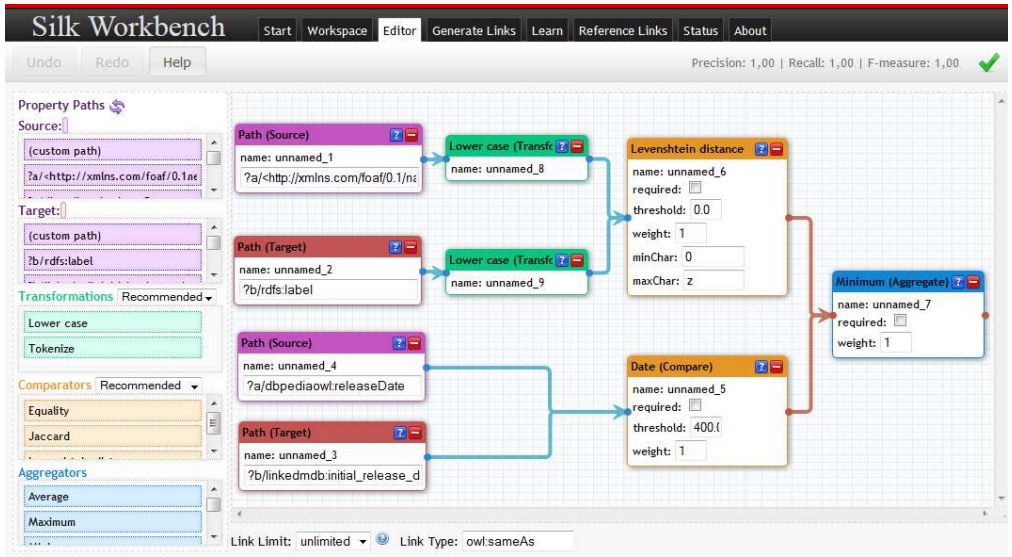
  Silk Server :
  - Provides an HTTP API for matching instances for an incoming stream of RDF data while keeping track of known entities
  - Can be used as an identity resolution component within applications that consume Linked Data from the Web



Linked Data supporting Tools

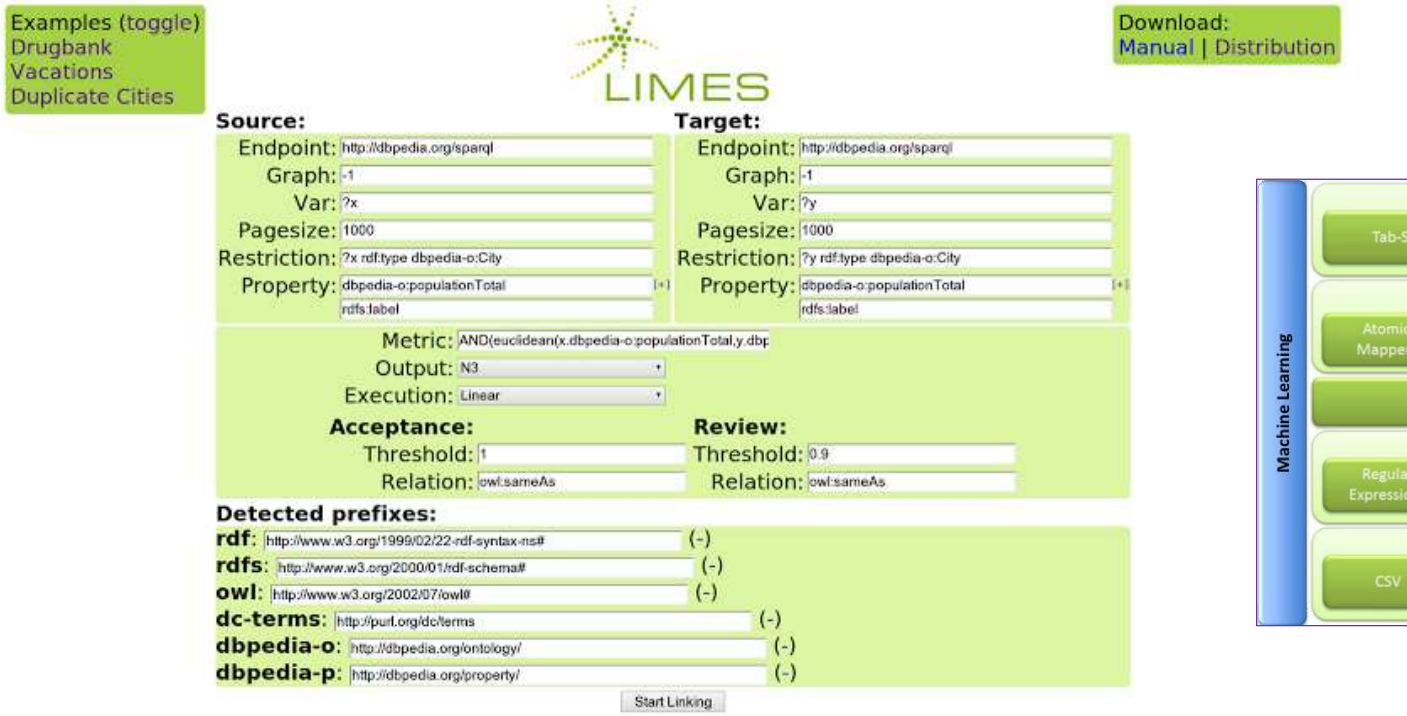
- ***SILK Workbench*** is a web application built on top of *SILK* that can be used to create projects and manage the creation of links between two RDF datasets. (https://www.assembla.com/spaces/silk/wiki/Silk_Workbench)

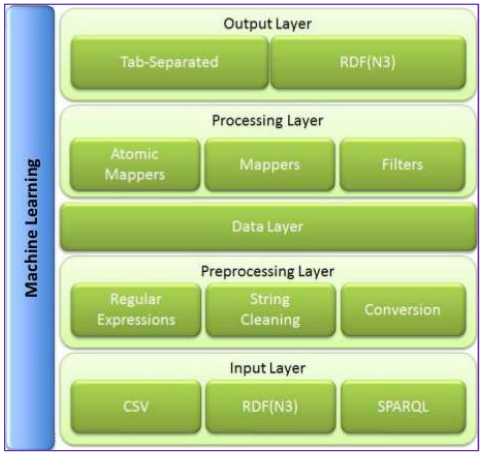
- The *SILK Workbench* has a graphical editor that can be used to create linkage rules. Support is also provided for the automatic learning of linkage rules.
- The *SILK Workbench* also provides an interface for examining automatically learned rules. These suggested rules can then be added to the set of linkage rules or rejected.



Linked Data supporting Tools

- ***LIMES*** (LInk discovery framework for MEtric Spaces) is a link discovery framework for the Web of Data. It implements time-efficient approaches for large-scale link discovery based on the characteristics of metric spaces. LIMES applies different approximation techniques to compute estimates of the similarity between instances. It is easily configurable via a web interface as well as can be user as standalone tool locally. (http://aksw.org/Projects/LIMES.html)

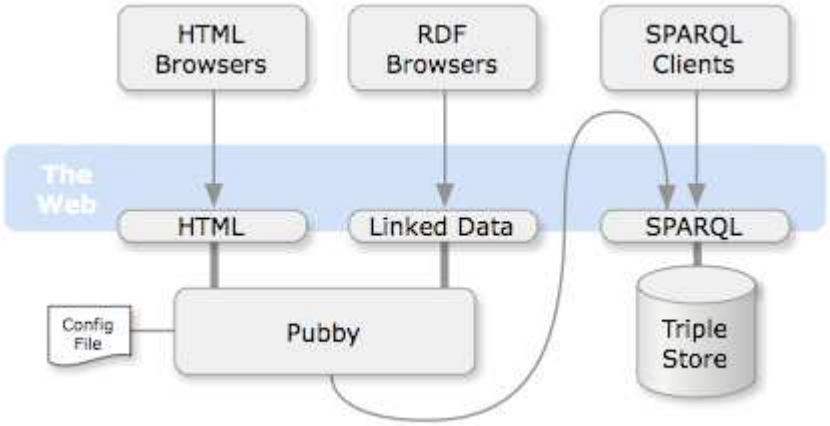




Linked Data supporting Tools

Much Semantic Web data lives inside triple stores and can be accessed only by sending SPARQL queries to a SPARQL endpoint. Triple stores and other SPARQL endpoints can be accessed only by SPARQL client applications that use the SPARQL protocol. It cannot be accessed by the growing variety of Linked Data clients.

- ***Pubby*** is a Linked Data Frontend for SPARQL Endpoints. *Pubby* makes it easy to turn a SPARQL endpoint into a Linked Data server providing a Linked Data interface to those RDF data sources. (http://wifo5-03.informatik.uni-mannheim.de/pubby/)



Homework

1. Play with Semantic Annotation tools. Try Automatic annotation mode as well as manual annotation.
2. Play around with a voiD Store:
   - Create voiD document to announce your dataset
3. Familiarize yourself with Linked Data Tools.