First things first…

- Assignment of slots for final presentations
- Q&A – I expect you to resend me corrected assignments, taking my feedback into account
  - e.g. for **Assignment 1**: make sure that your FOAF file validates in an RDF validator
    for **Assignment 2**: send me only parseable Turtle
    for **Assignment 3**: send me only **running** SPARQL queries, which you have tested.
    don't forget **Assignment 4** (just published)

- **Grades:**
  - No exam necessary.
  - But no "**Sehr Gut**" unless you have been excellent in the assignments and in your presentation.
  - I will send you some suggested grade after the presentation.
  - You can improve in an oral exam, if you want – by appointment.

Unit 7:
Querying and Exchanging Data on the Web

Overview

- Linked Data – The idea
- Why is it interesting for companies?
- Which challenges are lying ahead?
- XSPARQL: An approach to query and combine several Web Data Formats at once.

Linked Data – The idea

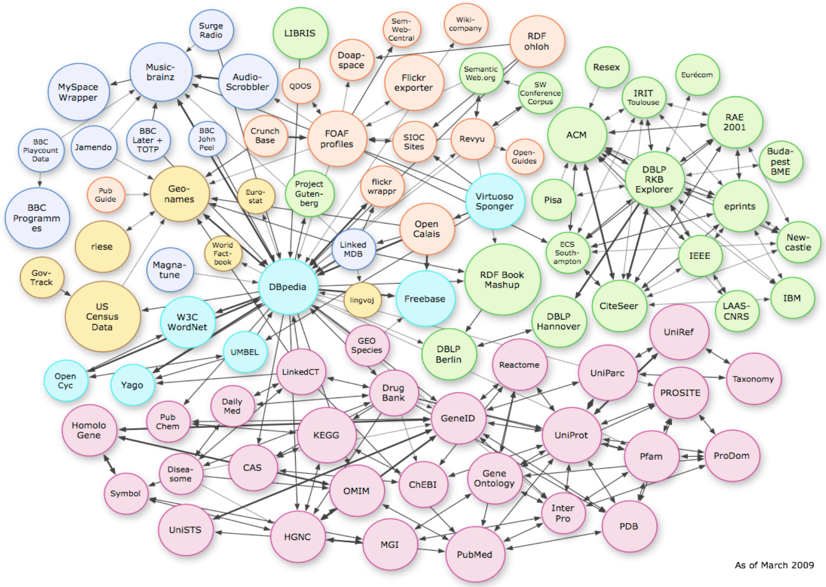
1. Everything gets a URI (conferences, people, talks, …)
2. These URIs are linked via RDF describing relations
3. Relations are URIs again (e.g. :name)
4. When I dereference the URIs, I should find more information about them

dblp.uni-trier.de
Computer Science Bibliography

DATA.GOV
EMPOWERING PEOPLE

data.gov.uk BETA

The New York Times



Freebase

WIKIPEDIA

BEST BUY

facebook

Linked Data – The idea

Let Tim Berners-Lee explain it:

http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html

(around 5:40)

http://www.ted.com/talks/tim_berners_lee_the_year_open_data_went_worldwide.html

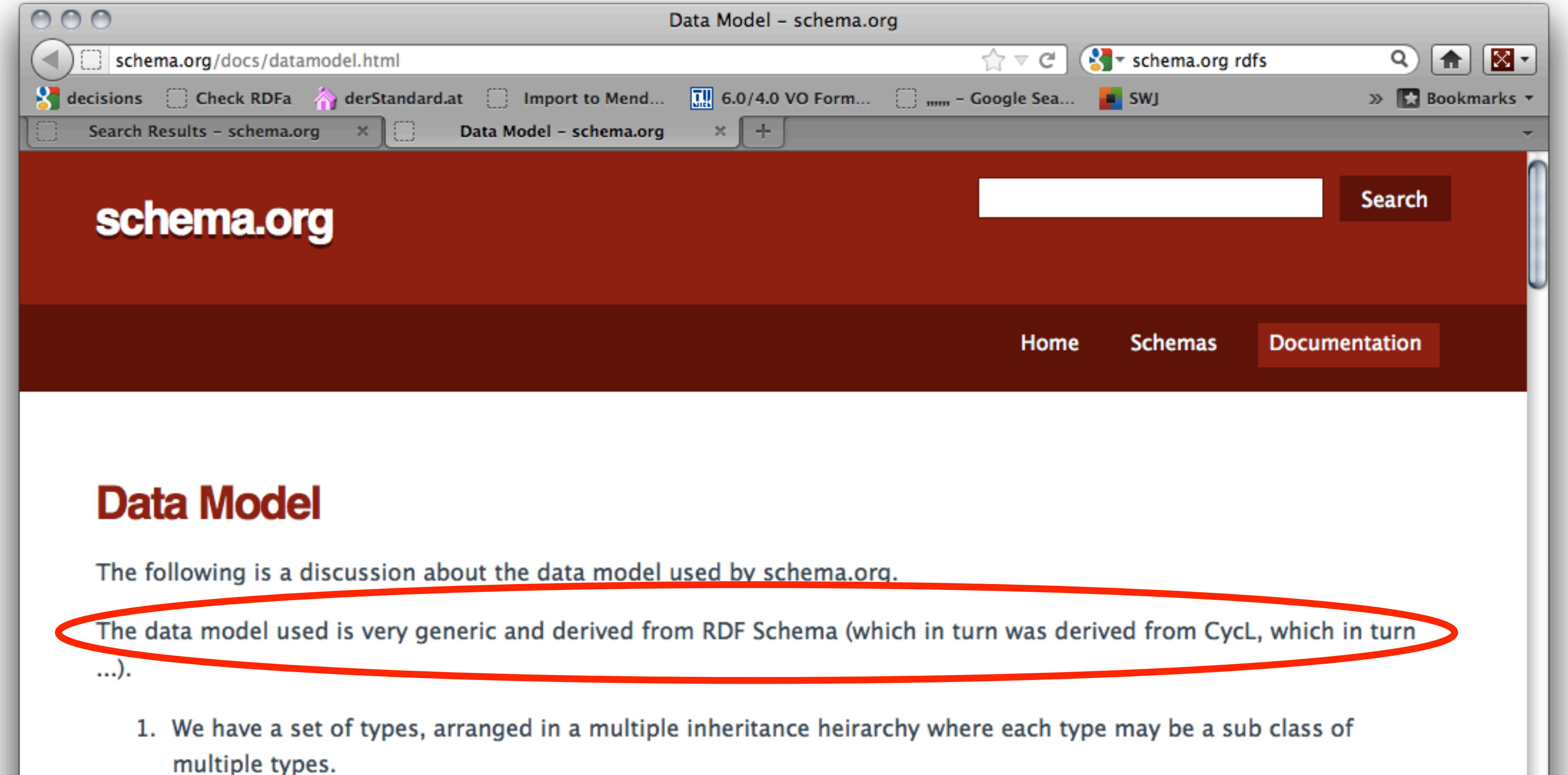
Why is this all interesting for companies?

SIEMENS

Why is this interesting for companies?

Linked Data and Open Data (apart from Linked Open Data) are both emerging paradigms:

- **Linked Data apart from the "LOD cloud":**
  - Enterprise Linked Data (for Knowledge Management within the Enterprise
  - Online companies (eCommerce, Search) start to leverage and support Linked Data



Data Model

The following is a discussion about the data model used by schema.org.

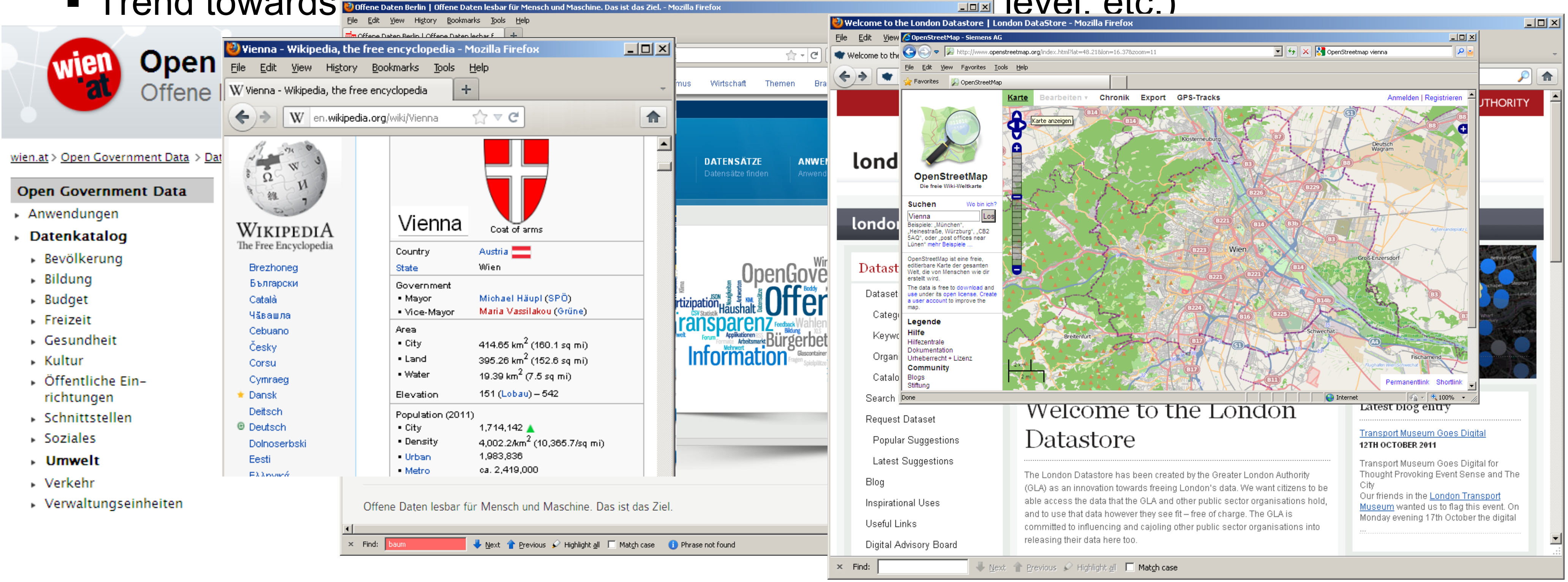
The data model used is very generic and derived from RDF Schema (which in turn was derived from CycL, which in turn ...).

1. We have a set of types, arranged in a multiple inheritance heirarchy where each type may be a sub class of multiple types.

Why is this interesting for companies?

Linked Data and Open Data (apart from Linked Open Data) are both emerging paradigms:

- **Open Data:**
  - Open Data is a trend towards transparency for Governments
  - More Publically available Data leverages new Business Models (not only for SMEs!)
  - Many Governments realize that Opening Data brings more revenue than selling it
  - (EU) regulations force Cities and Governments to publish Data
  - Trend towards harmonization (nationally, at European level, etc.)



wien .at
Open
Offene
Offene Daten Berlin | Offene Daten lesbar für Mensch und Maschine. Das ist das Ziel. - Mozilla Firefox
Vienna - Wikipedia, the free encyclopedia - Mozilla Firefox
File Edit View History Bookmarks Tools Help
en.wikipedia.org/wiki/Vienna
WIKIPEDIA
The Free Encyclopedia
Welcome to the London Datastore | London DataStore - Mozilla Firefox
OpenStreetMap - Siemens AG
OpenStreetMap
Die freie Wiki-Weltkarte
Karte Bearbeiten Chronik Export GPS-Tracks
Anmelden | Registrieren
Wien
DATENSÄTZE
Datensätze finden
Wirtschaft
Themen
OpenGove
Offen
Transparenz
Haushalt
Bürgerbet
Information
Wahlen
AUTHORITY
Find:
Next
Previous
Highlight all
Match case
Phrase not found

wien.at > Open Government Data > Dat

Open Government Data

- Anwendungen
- **Datenkatalog**
  - Bevölkerung
  - Bildung
  - Budget
  - Freizeit
  - Gesundheit
  - Kultur
  - Öffentliche Einrichtungen
  - Schnittstellen
  - Soziales
  - **Umwelt**
  - Verkehr
  - Verwaltungseinheiten

Brezhoneg
Български
Català
Чăвашла
Cebuano
Česky
Corsu
Cymraeg
Dansk
Deitsch
Deutsch
Dolnoserbski
Eesti

Vienna

Coat of arms

Country	Austria
State	Wien
Government	
• Mayor	Michael Häupl (SPÖ)
• Vice-Mayor	Maria Vassilakou (Grüne)
Area	
• City	414.65 km^2 (160.1 sq mi)
• Land	395.26 km^2 (152.6 sq mi)
• Water	19.39 km^2 (7.5 sq mi)
Elevation	151 (Lobau) – 542
Population (2011)	
• City	1,714,142
• Density	4,002.2/km^2 (10,365.7/sq mi)
• Urban	1,983,836
• Metro	ca. 2,419,000

Suchen

Vienna

OpenStreetMap ist eine freie, editierbare Karte der gesamten Welt, die von Menschen wie dir erstellt wird.

The data is free to download and use under its open license. Create a user account to improve the map.

Legende

Hilfe

Offene Daten lesbar für Mensch und Maschine. Das ist das Ziel.

Dataset
Categ
Keywo
Organ
Catalo
Search
Request Dataset
Popular Suggestions
Latest Suggestions
Blog
Inspirational Uses
Useful Links
Digital Advisory Board

Welcome to the London Datastore

The London Datastore has been created by the Greater London Authority (GLA) as an innovation towards freeing London's data. We want citizens to be able access the data that the GLA and other public sector organisations hold, and to use that data however they see fit – free of charge. The GLA is committed to influencing and cajoling other public sector organisations into releasing their data here too.

Latest blog entry

Transport Museum Goes Digital

12TH OCTOBER 2011

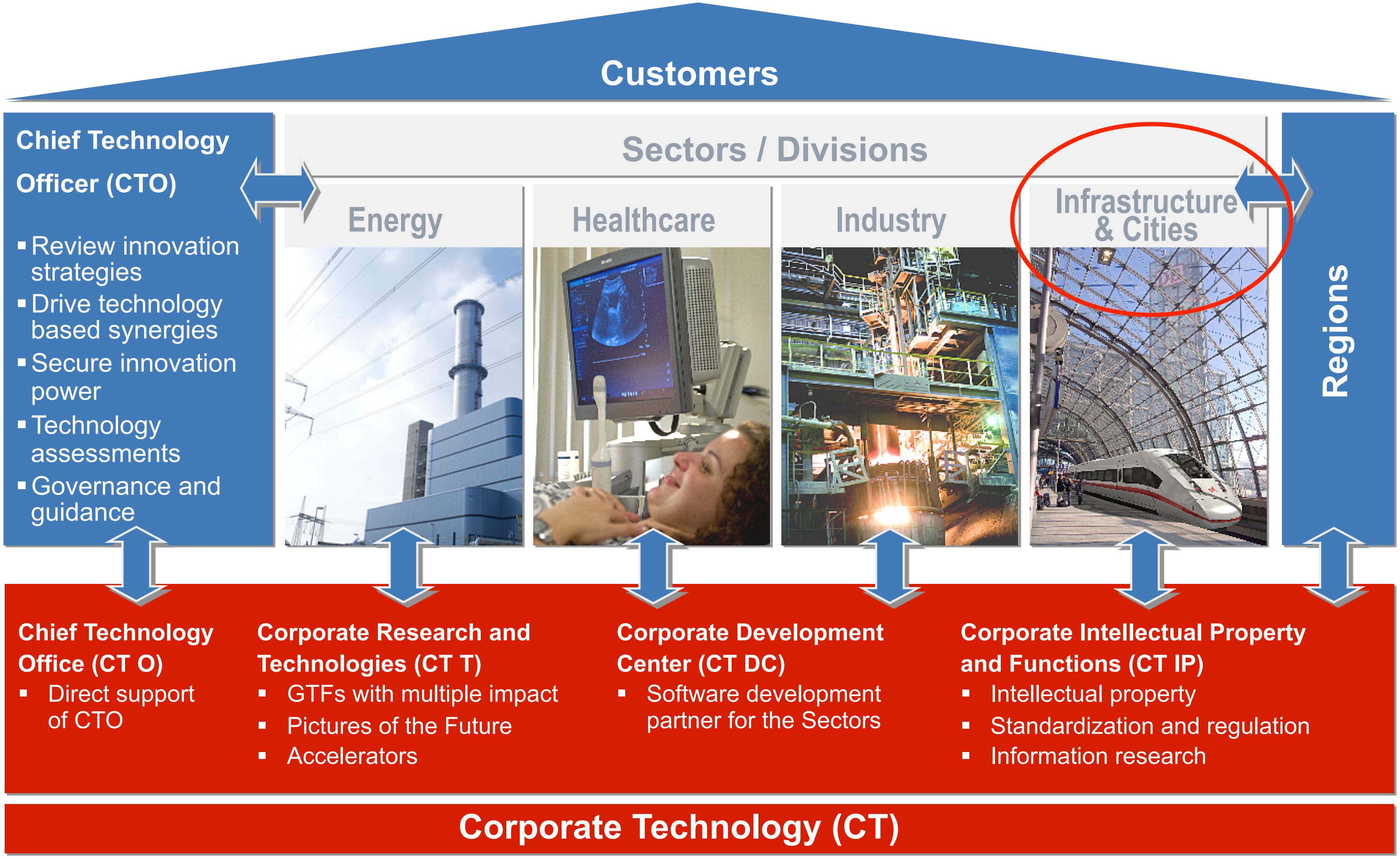
Transport Museum Goes Digital for Thought Provoking Event Sense and The City

Our friends in the London Transport Museum wanted us to flag this event. On Monday evening 17th October the digital

Siemens Corporate Technology (CT)

Networking the integrated technology company

SIEMENS



Customers

Chief Technology Officer (CTO)

- Review innovation strategies
- Drive technology based synergies
- Secure innovation power
- Technology assessments
- Governance and guidance

Sectors / Divisions

Energy

Healthcare

Industry

Infrastructure & Cities

Regions

Chief Technology Office (CT O)

- Direct support of CTO

Corporate Research and Technologies (CT T)

- GTFs with multiple impact
- Pictures of the Future
- Accelerators

Corporate Development Center (CT DC)

- Software development partner for the Sectors

Corporate Intellectual Property and Functions (CT IP)

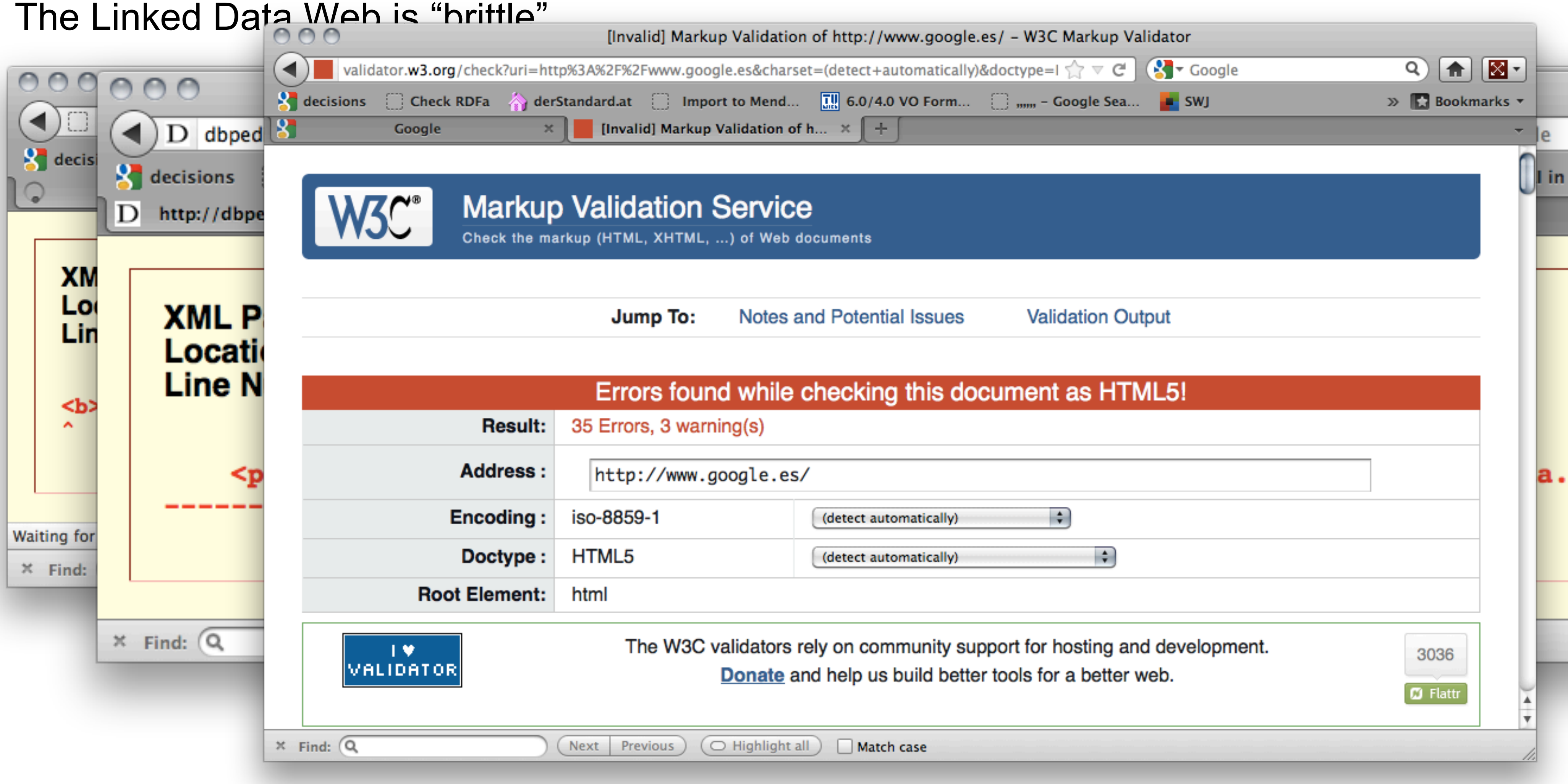
- Intellectual property
- Standardization and regulation
- Information research

Corporate Technology (CT)

Challenges ahead...

Challenges/Problems

The Linked Data Web is “brittle”



[Invalid] Markup Validation of http://www.google.es/ – W3C Markup Validator

validator.w3.org/check?uri=http%3A%2F%2Fwww.google.es&charset=(detect+automatically)&doctype=I

Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: Notes and Potential Issues   Validation Output

Errors found while checking this document as HTML5!

Result:	35 Errors, 3 warning(s)
Address :	http://www.google.es/
Encoding :	iso-8859-1 (detect automatically)
Doctype :	HTML5 (detect automatically)
Root Element:	html

I ♥ VALIDATOR

The W3C validators rely on community support for hosting and development.
Donate and help us build better tools for a better web.

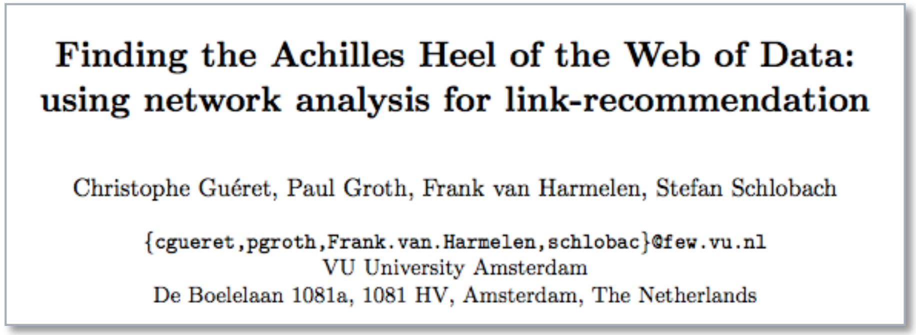
3036 Flattr

Just like the normal Web is (did you ever try to run an HTML validator on google.com)?

How good/bad is published Linked Data?

ISWC2010

“Almost all infrastructural connectivity on the WoD is mediated by 3 servers, xmlns.com, dbpedia.org and purl.org, making the system very brittle.”

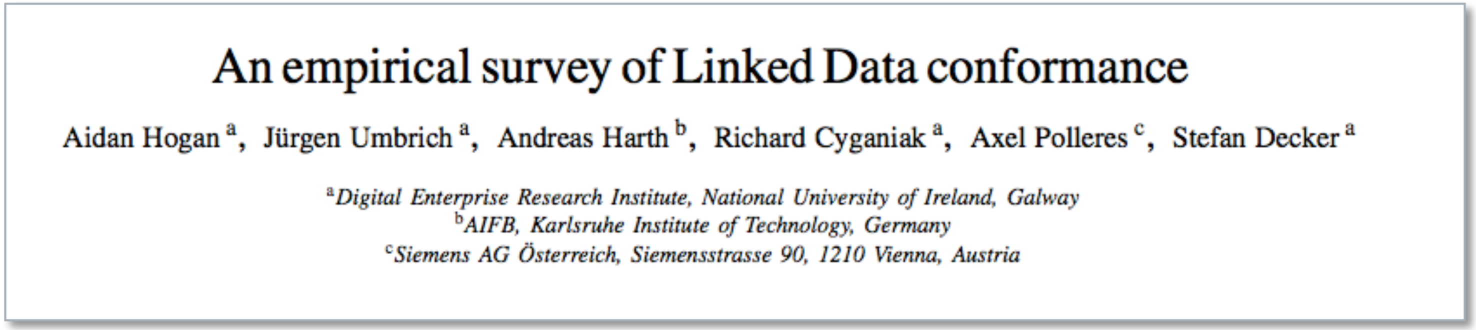


Finding the Achilles Heel of the Web of Data: using network analysis for link-recommendation

Christophe Guéret, Paul Groth, Frank van Harmelen, Stefan Schlobach

`{cgueret,pgroth,Frank.van.Harmelen,schlobac}@few.vu.nl`
VU University Amsterdam
De Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands

Journal of Web Semantics (forthcoming)



An empirical survey of Linked Data conformance

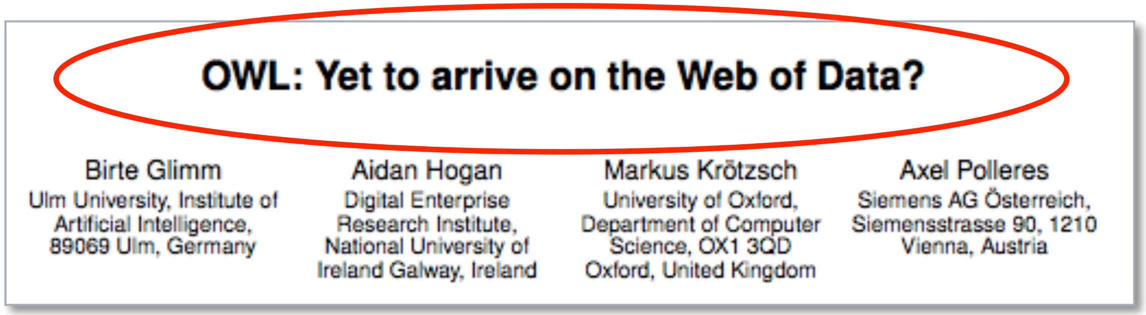
Aidan Hogan [a], Jürgen Umbrich [a], Andreas Harth [b], Richard Cyganiak [a], Axel Polleres [c], Stefan Decker [a]

[a] *Digital Enterprise Research Institute, National University of Ireland, Galway*
[b] *AIFB, Karlsruhe Institute of Technology, Germany*
[c] *Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria*

“conformance of data providers varies significantly for the different Linked Data guide- lines highlighted, which in turn may have implications for ad hoc consumers operating over the Web of Data.”

How much OWL is on the Web of Data?
What’s missing for using Linked Data?

LDOW workshop @ WWW2012



OWL: Yet to arrive on the Web of Data?

Birte Glimm
Ulm University, Institute of Artificial Intelligence, 89069 Ulm, Germany

Aidan Hogan
Digital Enterprise Research Institute, National University of Ireland Galway, Ireland

Markus Krötzsch
University of Oxford, Department of Computer Science, OX1 3QD Oxford, United Kingdom

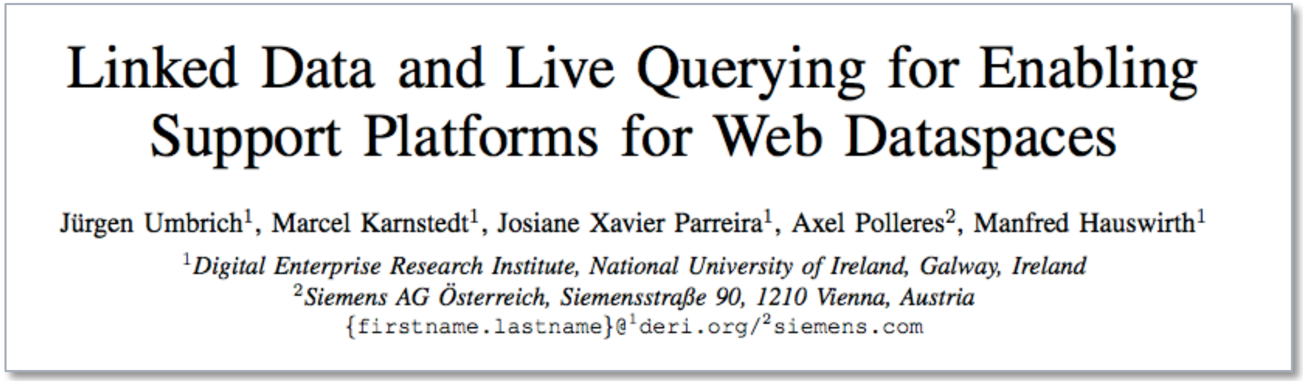
Axel Polleres
Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria

“Single-triple expressible OWL RL axioms are most prominent on the Web.”

DESWEB workshop @ICDE2012

“indexes for Linked Data in the Web are often incomplete and outdated.”

→ Needs rethinking in terms of applying traditional Database techniques.



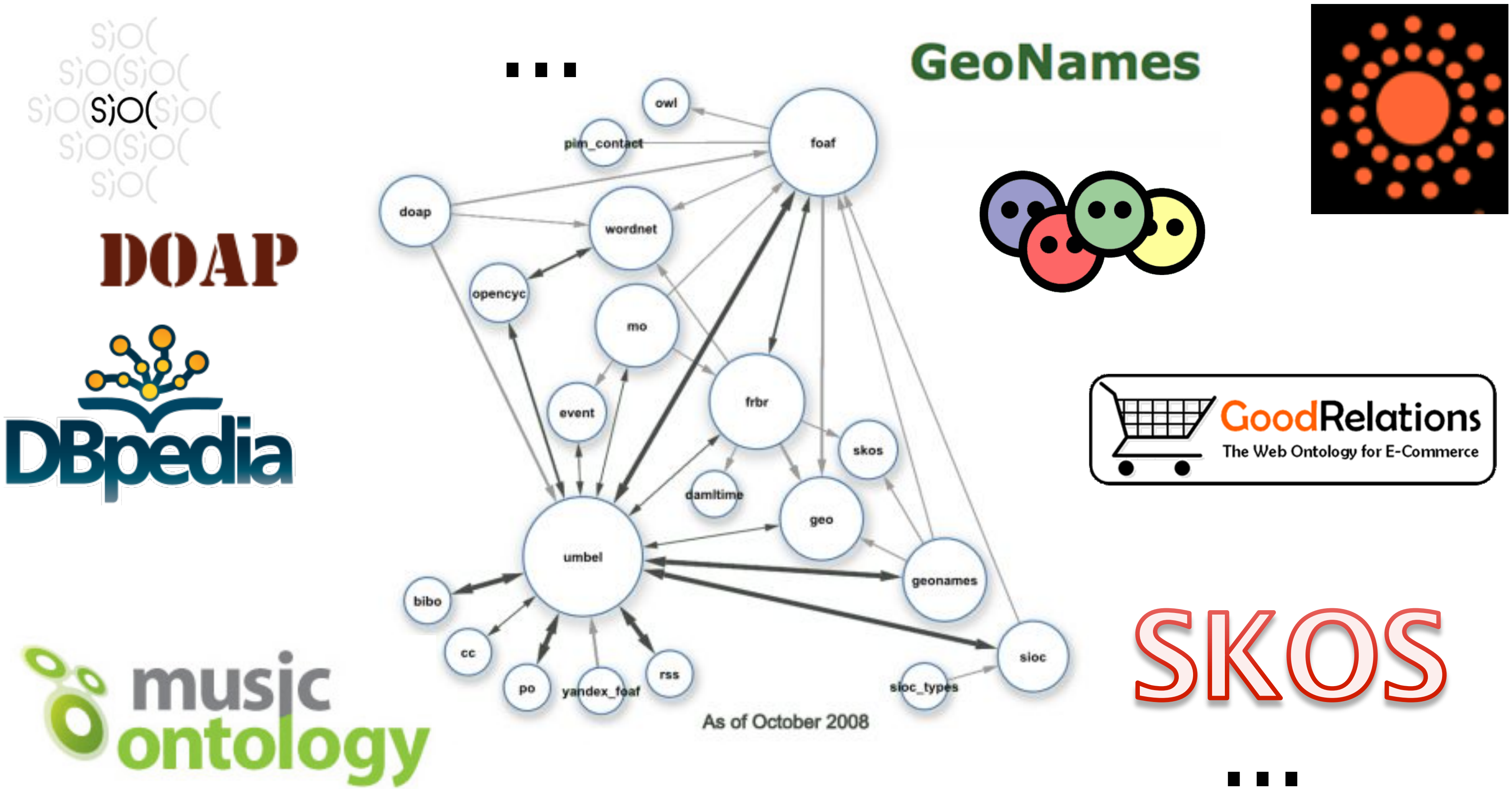
Linked Data and Live Querying for Enabling Support Platforms for Web Dataspaces

Jürgen Umbrich[1], Marcel Karnstedt[1], Josiane Xavier Parreira[1], Axel Polleres[2], Manfred Hauswirth[1]

[1]*Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland*
[2]*Siemens AG Österreich, Siemensstraße 90, 1210 Vienna, Austria*
{firstname.lastname}@[1]deri.org/[2]siemens.com

Linked Data, RDFS and OWL: **Linked Vocabularies**

SIEMENS



As of October 2008

Image from http://blog.dbtune.org/public/.081005_lod_constellation_m.jpg: Giasson, Bergman

So what OWL is used out there?

Looked at Billion Triple Challenge 2011 Dataset

- 2.1 billion quadruples, crawled from…
- 7.4 million RDF/XML documents, covering…
- 791 (pay-level) domains

Count OWL features used in the dataset:

- Per use
- Per document
- Per domain
- Can be skewed by data

Ranked OWL features using ***PageRank***:

- Rank documents based on dereferenceable links
- For each OWL feature, sum the rank of documents using it
- **Intuition: Approximates probability of encountering an OWL feature during a random walk of the data**

Results of ranking (see paper for all details)

1	`rdf:Property`	5.74E-1
2	`rdfs:range`	4.67E-1
3	`rdfs:domain`	4.62E-1
4	`rdfs:subClassOf`	4.60E-1
5	`rdfs:Class`	4.45E-1
6	`rdfs:subPropertyOf`	2.35E-1
7	`owl:Class`	1.74E-1
8	`owl:ObjectProperty`	1.68E-1
9	`rdfs:Datatype`	1.68E-1
10	`owl:DatatypeProperty`	1.65E-1
11	`owl:AnnotationProperty`	1.60E-1
12	`owl:FunctionalProperty`	9.18E-2
13	`owl:equivalentProperty`	8.54E-2
14	`owl:inverseOf`	7.91E-2
15	`owl:disjointWith`	7.65E-2

SIEMENS

Results of ranking (see paper for all details)

…

16	`owl:sameAs`	7.29E-2
17	`owl:equivalentClass`	5.24E-2
18	`owl:InverseFunctionalProperty`	4.79E-2
19	`owl:unionOf`	3.15E-2
20	`owl:SymmetricProperty`	3.13E-2
21	`owl:TransitiveProperty`	2.98E-2
22	`owl:someValuesFrom`	2.13E-2
23	`rdf:_*`	1.42E-2
24	`owl:allValuesFrom`	2.98E-3
25	`owl:minCardinality`	2.43E-3
26	`owl:maxCardinality`	2.14E-3
27	`owl:cardinality`	1.75E-3
28	`owl:oneOf`	4.13E-4
29	`owl:hasValue`	3.91E-4
30	`owl:intersectionOf`	3.37E-4
31	`owl:NamedIndividual`	3.37E-4

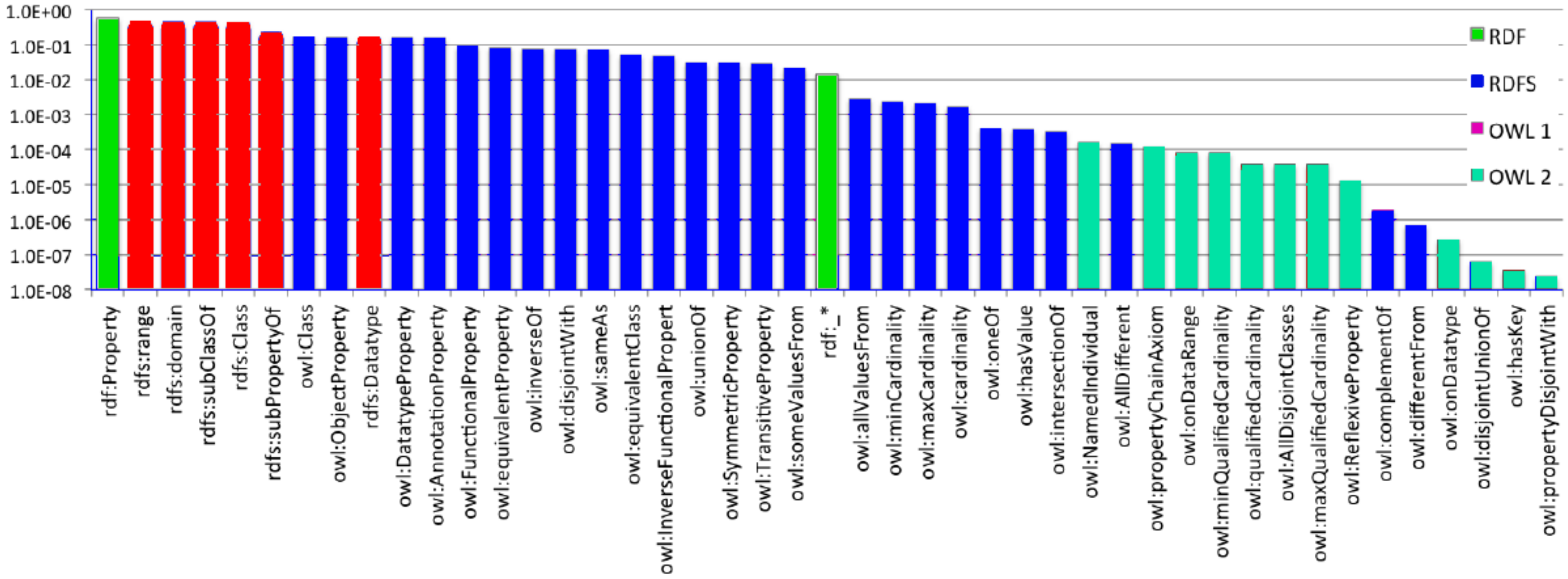
Observations?

RDFS features amongst the most prominently used

OWL 2 features not yet used prominently

RDF | RDFS | OWL | OWL 2

x-axis is log-scale!



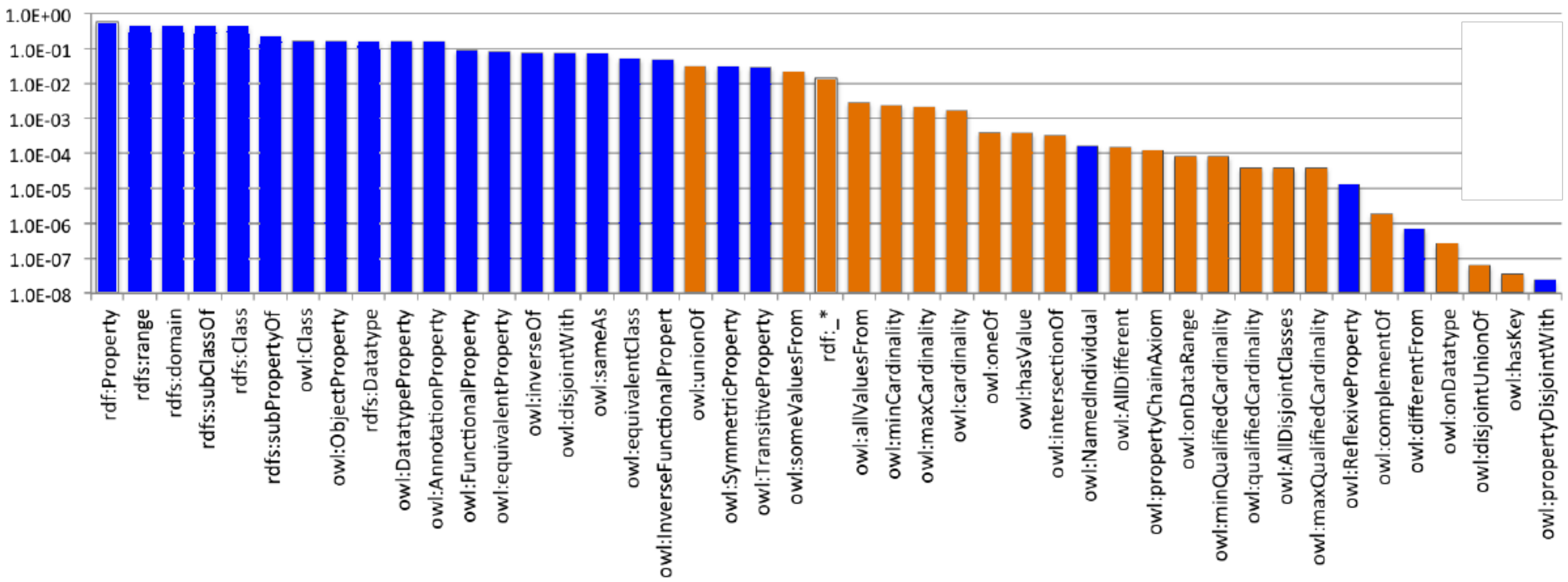
Observations?

(OWL) Features expressed with a single RDF triple are most prominent

- Roughly speaking, features not *requiring* blank nodes
  - e.g., sub-class/-property, inverse-of, equivalent property/class, sameas, domain/range, disjoint with, etc.
- Not those requiring lists or *n*-ary predicate in RDF mapping
  - e.g., union, intersection, cardinalities, all-disjoint, some/all/has-value restrictions, hasKey, pCAs, etc.

Single Triple (No BNodes) | Multi-Triple (Needs BNodes)

x-axis is log-scale!

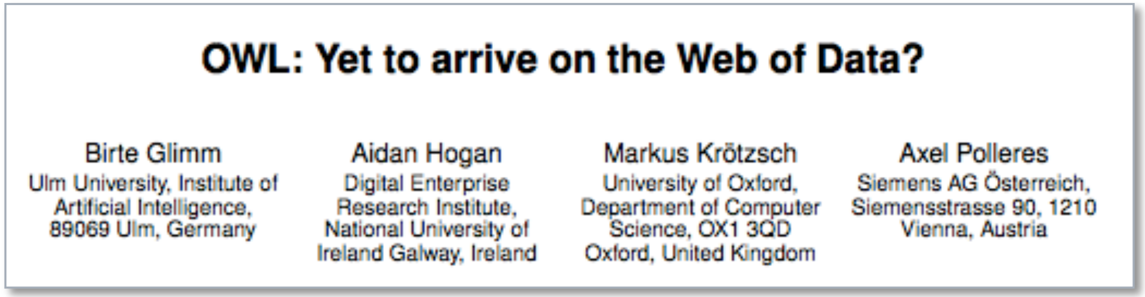


What Reasoning is needed?

Bottomline:

A **subset of OWL 2 RL** (which is efficiently implementable, i.e. without ABox-joins) is sufficient to cover reasoning on most Linked Data sources!

Details, cf.



OWL: Yet to arrive on the Web of Data?

Birte Glimm
Ulm University, Institute of Artificial Intelligence, 89069 Ulm, Germany

Aidan Hogan
Digital Enterprise Research Institute, National University of Ireland Galway, Ireland

Markus Krötzsch
University of Oxford, Department of Computer Science, OX1 3QD Oxford, United Kingdom

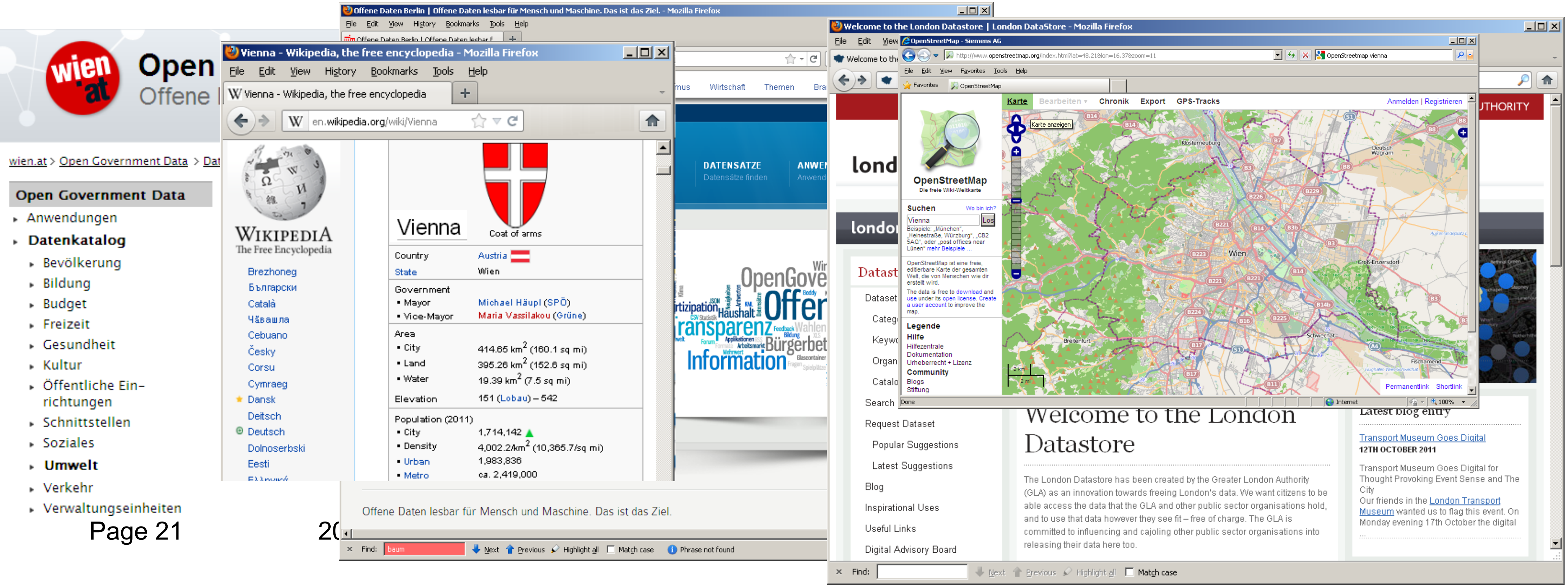
Axel Polleres
Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria

However…

Not all Web Data is RDF (and OWL):

In fact, most Web Data is still in other formats:
XML, CSV, JSON…

→ We need approaches to deal with these formats!



wien.at > Open Government Data > Dat

Open Government Data
- Anwendungen
- **Datenkatalog**
  - Bevölkerung
  - Bildung
  - Budget
  - Freizeit
  - Gesundheit
  - Kultur
  - Öffentliche Einrichtungen
  - Schnittstellen
  - Soziales
  - **Umwelt**
  - Verkehr
  - Verwaltungseinheiten

Vienna - Wikipedia, the free encyclopedia - Mozilla Firefox

en.wikipedia.org/wiki/Vienna

Vienna — Coat of arms

Country	Austria
State	Wien
Government	
• Mayor	Michael Häupl (SPÖ)
• Vice-Mayor	Maria Vassilakou (Grüne)
Area	
• City	414.65 km² (160.1 sq mi)
• Land	395.26 km² (152.6 sq mi)
• Water	19.39 km² (7.5 sq mi)
Elevation	151 (Lobau) – 542
Population (2011)	
• City	1,714,142
• Density	4,002.2/km² (10,365.7/sq mi)
• Urban	1,983,836
• Metro	ca. 2,419,000

Offene Daten Berlin | Offene Daten lesbar für Mensch und Maschine. Das ist das Ziel. - Mozilla Firefox

Offene Daten lesbar für Mensch und Maschine. Das ist das Ziel.

Find: baum — Phrase not found

Welcome to the London Datastore | London DataStore - Mozilla Firefox

OpenStreetMap - Siemens AG

OpenStreetMap — Die freie Wiki-Weltkarte

Suchen: Vienna

Welcome to the London Datastore

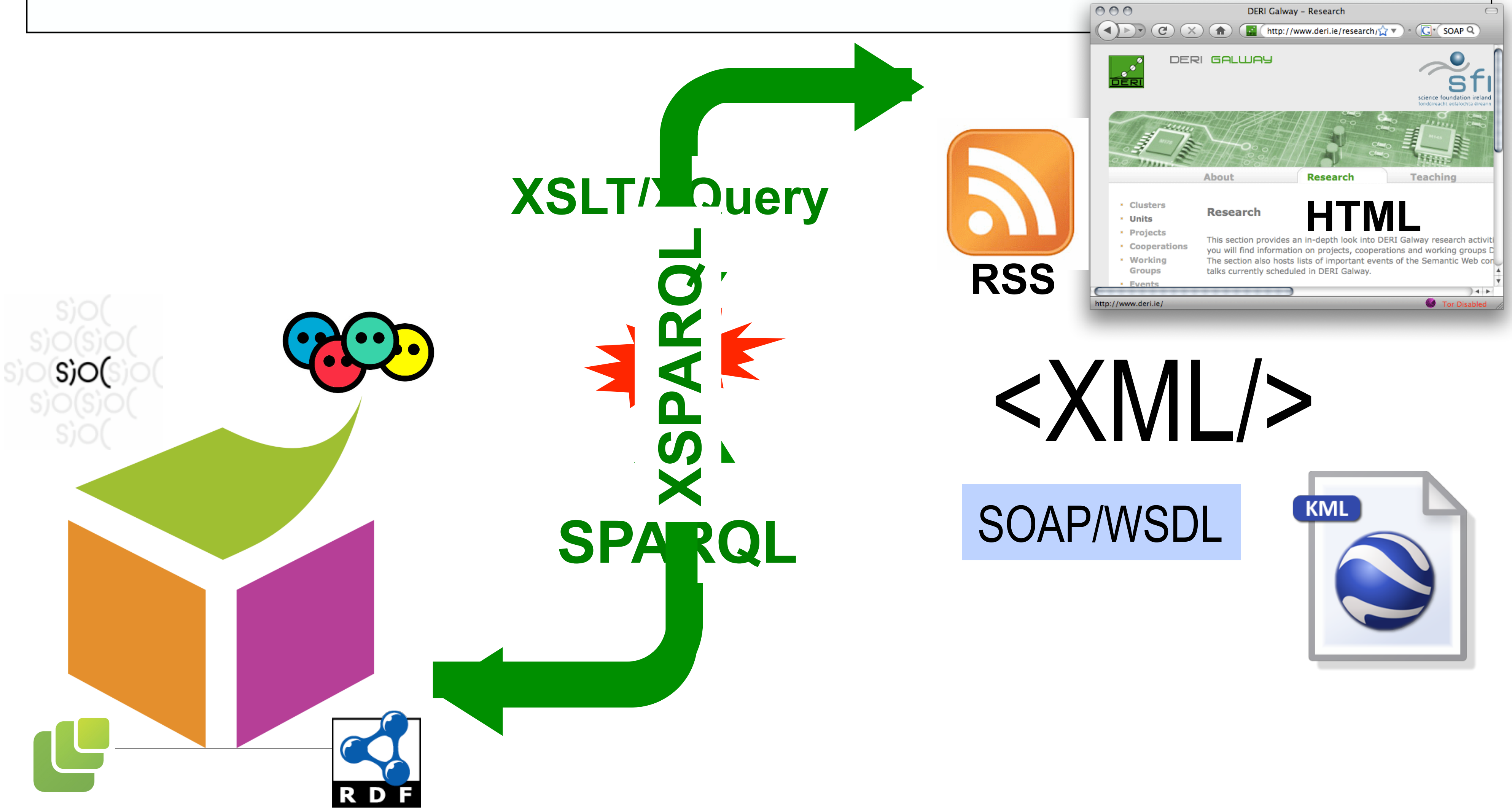
The London Datastore has been created by the Greater London Authority (GLA) as an innovation towards freeing London's data. We want citizens to be able access the data that the GLA and other public sector organisations hold, and to use that data however they see fit – free of charge. The GLA is committed to influencing and cajoling other public sector organisations into releasing their data here too.

Latest blog entry

Transport Museum Goes Digital
12TH OCTOBER 2011

Transport Museum Goes Digital for Thought Provoking Event Sense and The City
Our friends in the London Transport Museum wanted us to flag this event. On Monday evening 17th October the digital

XML & RDF: one Web – two formats



DERI Galway – Research

http://www.deri.ie/research/

SOAP

DERI GALWAY

sfi science foundation ireland

About | Research | Teaching

- Clusters
- Units
- Projects
- Cooperations
- Working Groups
- Events

Research

This section provides an in-depth look into DERI Galway research activiti
you will find information on projects, cooperations and working groups D
The section also hosts lists of important events of the Semantic Web con
talks currently scheduled in DERI Galway.

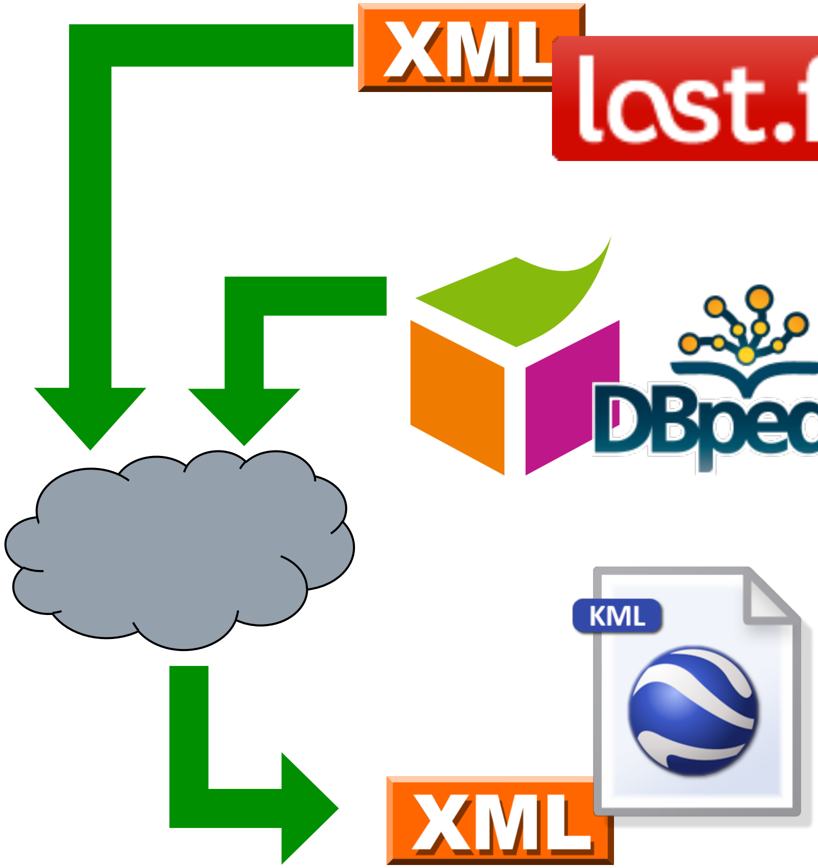
http://www.deri.ie/

Tor Disabled

A Sample Scenario…

Example: Favourite artists location

Display information about your favourite artists on a map



Last.fm knows what music you listen to, your most played artists, etc.

Using RDF allows to combine Last.fm info with other information on the web, e.g. location.

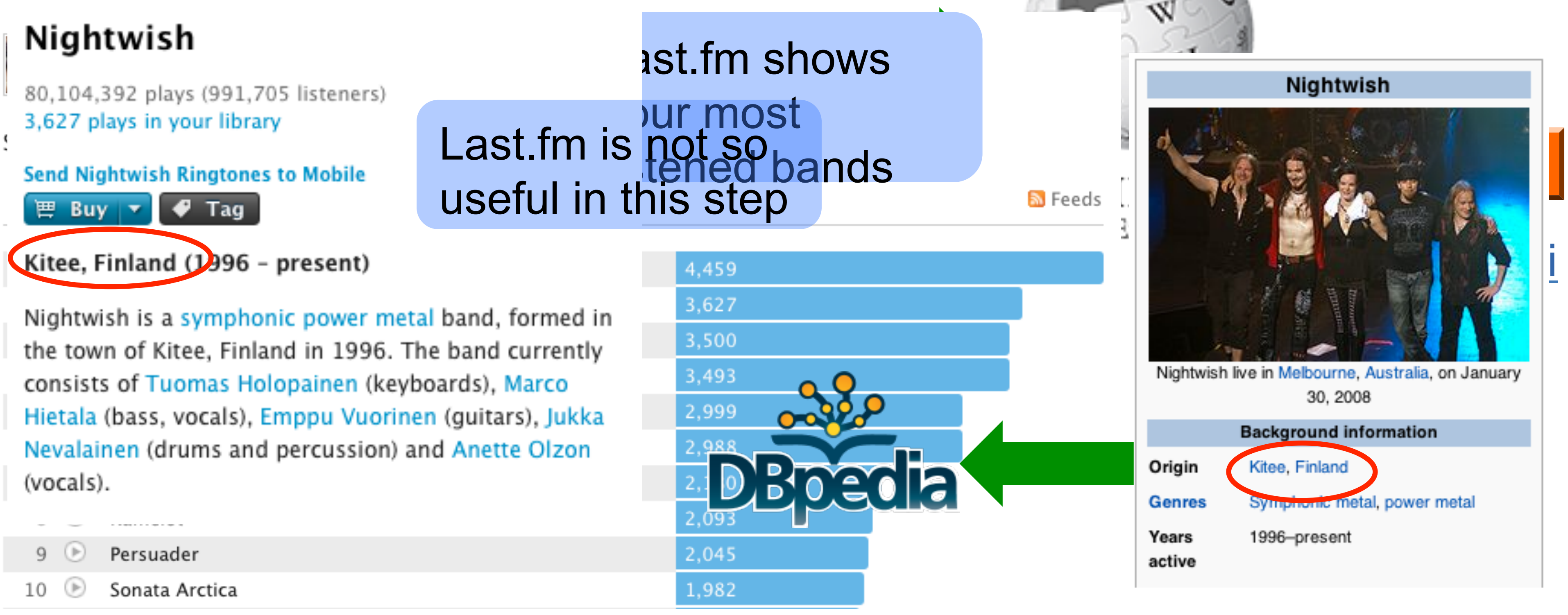
Show your top bands hometown in Google Maps.

SIEMENS

Example: Favourite artists location

How to implement the visualisation?

1) Get your favourite bands
2) Get the hometown of the bands
3) Create a KML file to be displayed in Google Maps



ast.fm shows
our most
tened bands

Last.fm is not so useful in this step

Nightwish

80,104,392 plays (991,705 listeners)

3,627 plays in your library

Send Nightwish Ringtones to Mobile

Kitee, Finland (1996 – present)

Nightwish is a symphonic power metal band, formed in the town of Kitee, Finland in 1996. The band currently consists of Tuomas Holopainen (keyboards), Marco Hietala (bass, vocals), Emppu Vuorinen (guitars), Jukka Nevalainen (drums and percussion) and Anette Olzon (vocals).

9	Persuader	2,045
10	Sonata Arctica	1,982

Nightwish

Nightwish live in Melbourne, Australia, on January 30, 2008

Background information

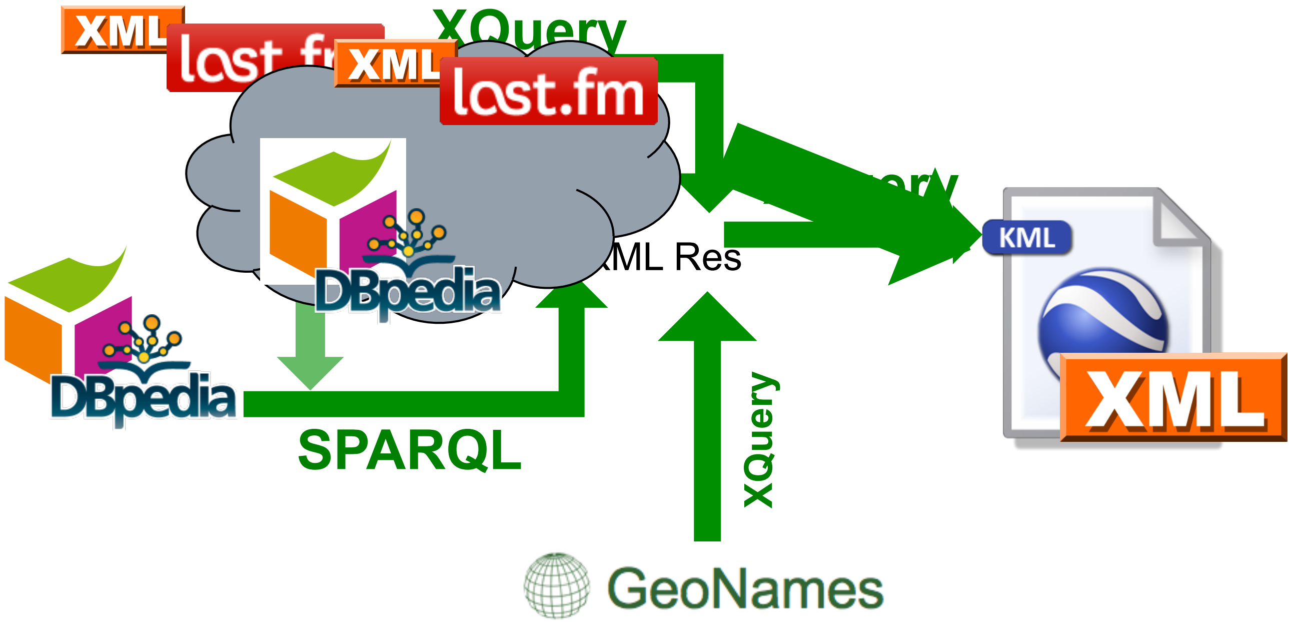
Origin	Kitee, Finland
Genres	Symphonic metal, power metal
Years active	1996–present

SIEMENS

Example: Favourite artists location

How to implement the visualisation?

1) Get your favourite bands
2) Get the hometown of the bands
3) Create a KML file to be displayed in Google Maps



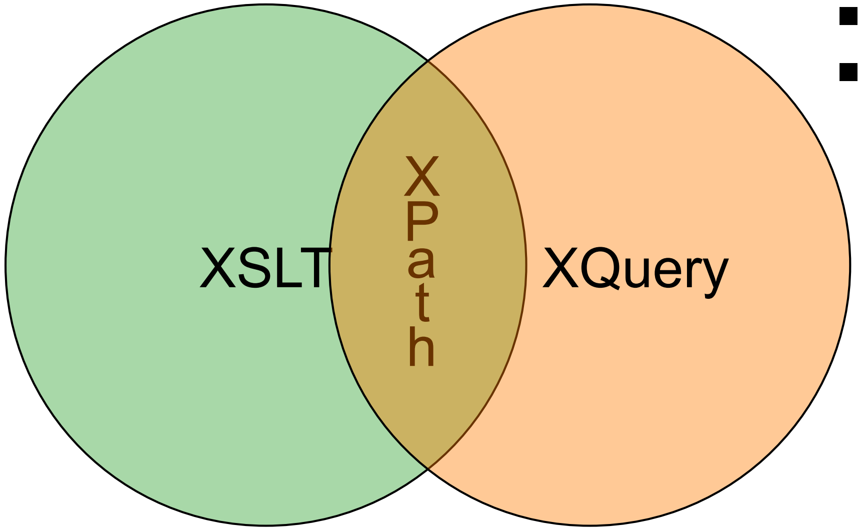
Transformation and Query Languages

XML world

RDF world

XML Transformation Language
Syntax: XML

- XML Query Language
- non-XML syntax

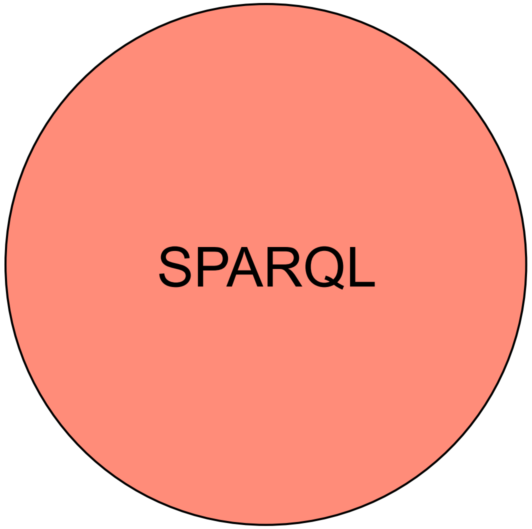


- XPath is the common core
- Mostly used to select nodes from an XML doc

SPARQL XML Result format

RDF/XML… ambiguous

- Query Language for RDF
- Pattern based
- declarative



Querying XML Data from Last.fm with XQuery 1/2

```
<lfm status="ok">
  <topartists type="overall">
    <artist rank="1">
      <name>Therion</name>
      <playcount>4459</playcount>
      <url>http://www.last.fm/music/Therion</url>
    </artist>
    <artist rank="2">
      <name>Nightwish</name>
      <playcount>3627</playcount>
      <url>http://www.last.fm/music/Nightwish</url>
    </artist>
  </topartists>
</lfm>
```

Last.fm API format:

- root element: “lfm”, then “topartists”
- sequence of “artist”

Querying this document with XPath:

XPath steps: `/lfm` Selects the “lfm” root element

`//artist` Selects all the “artist” elements

XPath Predicates `//artist[@rank = 1]` Selects the “artist” with rank 1

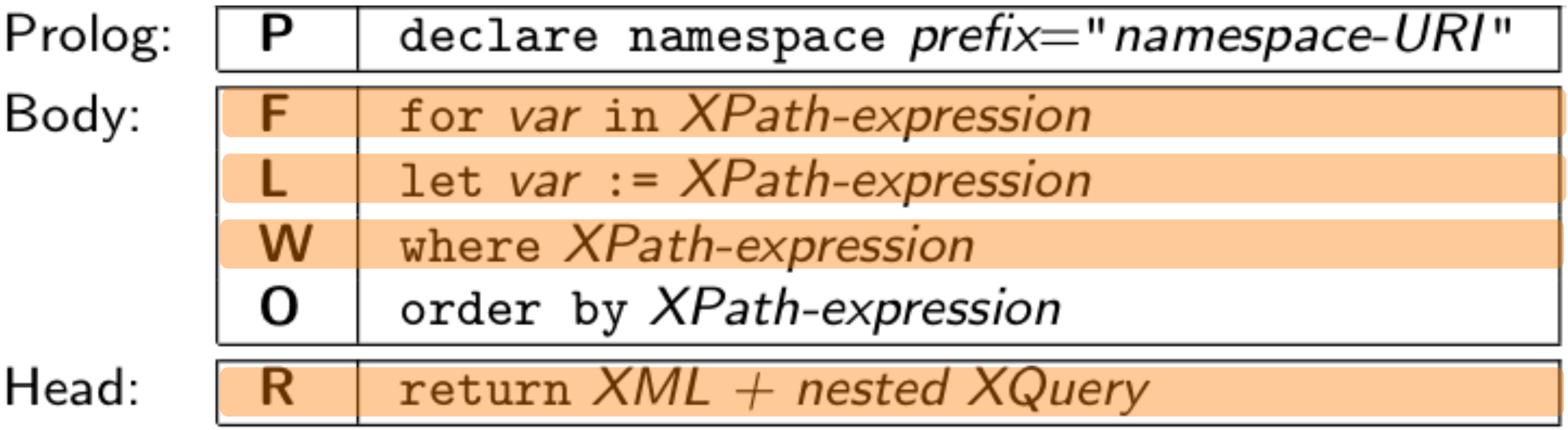
Querying XML Data from Last.fm with XQuery 2/2

iterate over sequences

assign values to variables

filter expressions

create XML elements



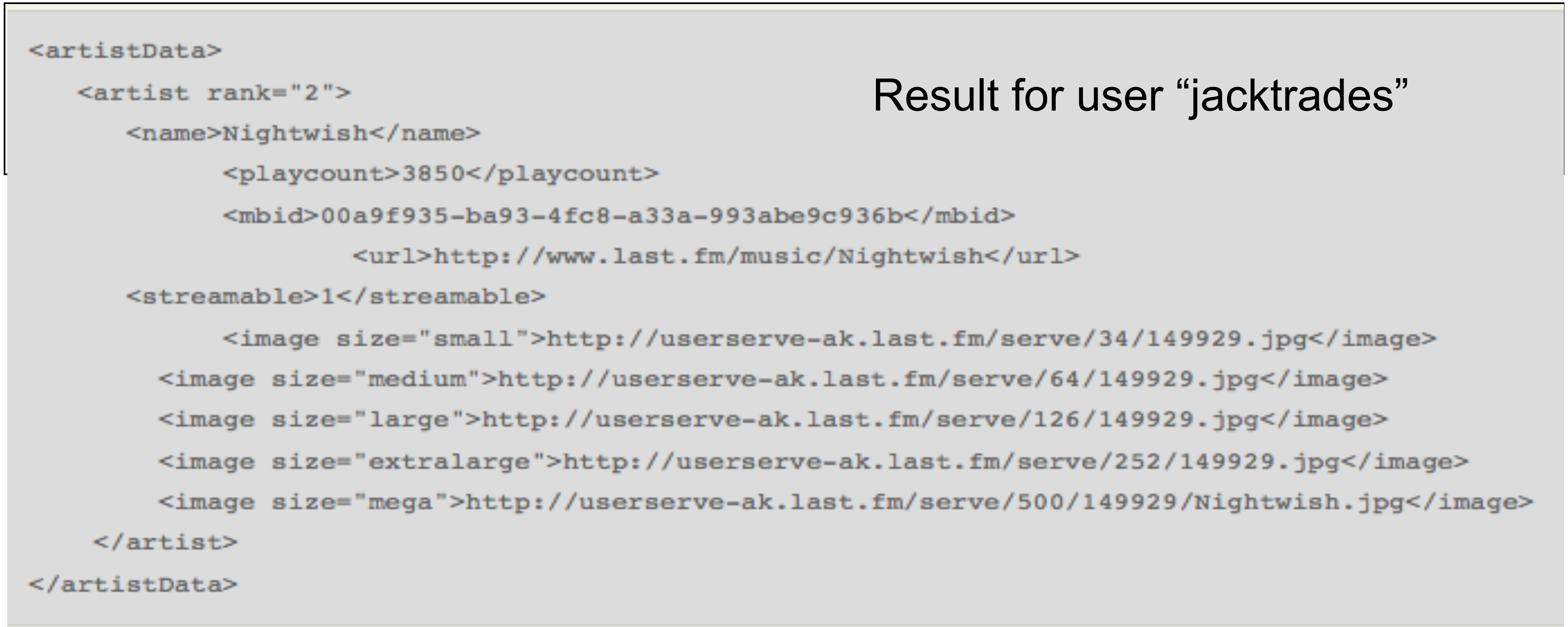
Prolog:	P	declare namespace *prefix*="*namespace-URI*"
Body:	F	for *var* in *XPath-expression*
	L	let *var* := *XPath-expression*
	W	where *XPath-expression*
	O	order by *XPath-expression*
Head:	R	return *XML + nested XQuery*

Query:

Retrieve information regarding a users' 2nd top artists from the

Last.fm API

```
let $doc := "http://ws.audioscrobbler.com/2.0/user.gettopartist"
for $artist in doc($doc)//artist
where $artist[@rank = 2]
return <artistData>{$artist}</artistData>
```



Result for user “jacktrades”

```
<artistData>
   <artist rank="2">
      <name>Nightwish</name>
            <playcount>3850</playcount>
            <mbid>00a9f935-ba93-4fc8-a33a-993abe9c936b</mbid>
                    <url>http://www.last.fm/music/Nightwish</url>
      <streamable>1</streamable>
            <image size="small">http://userserve-ak.last.fm/serve/34/149929.jpg</image>
        <image size="medium">http://userserve-ak.last.fm/serve/64/149929.jpg</image>
        <image size="large">http://userserve-ak.last.fm/serve/126/149929.jpg</image>
        <image size="extralarge">http://userserve-ak.last.fm/serve/252/149929.jpg</image>
        <image size="mega">http://userserve-ak.last.fm/serve/500/149929/Nightwish.jpg</image>
    </artist>
</artistData>
```

Query:

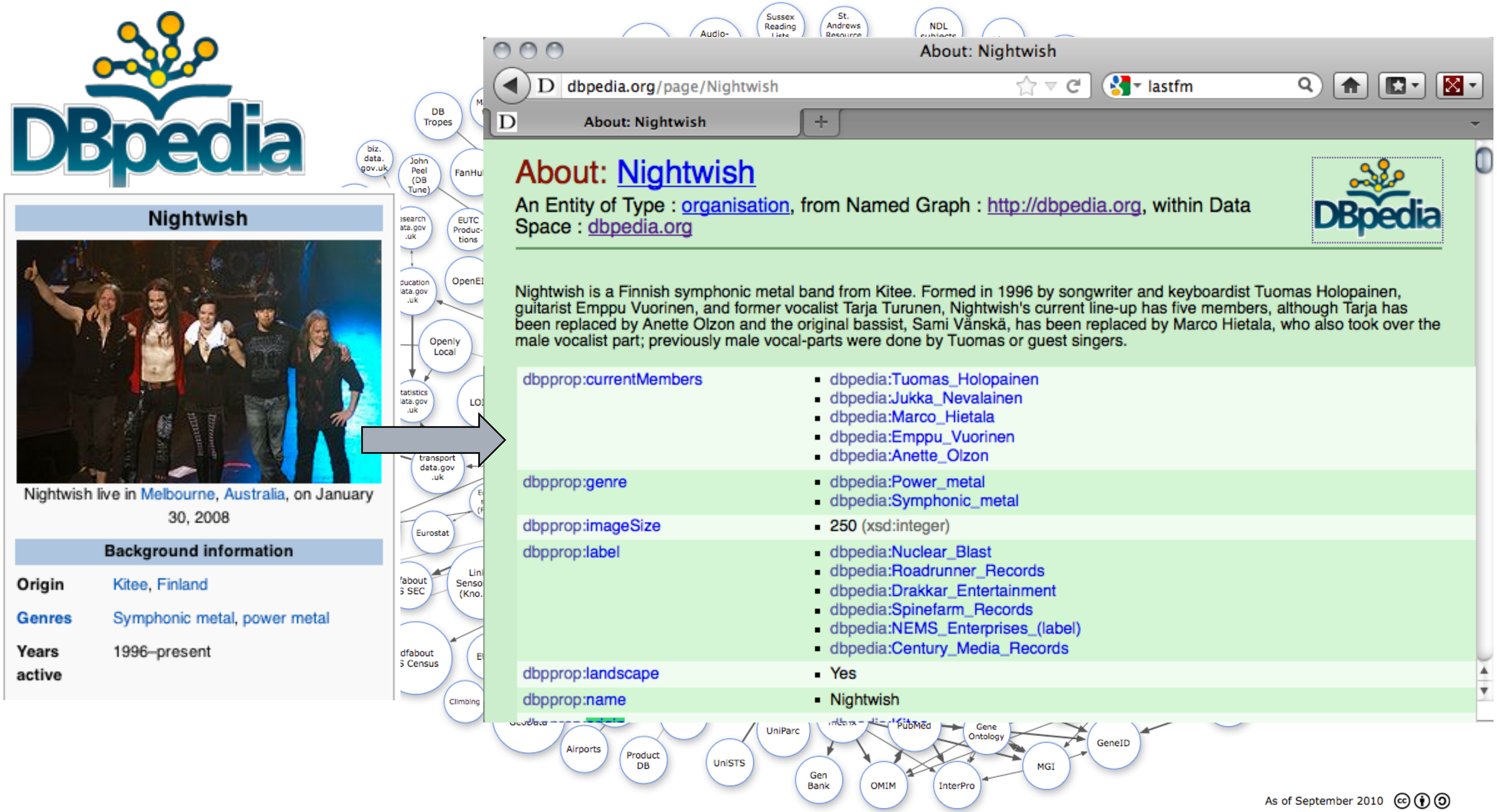
Retrieve information regarding a users' 2nd top artists from the

Last.fm API

```
let $doc := "http://ws.audioscrobbler.com/2.0/user.gettopartist"
for $artist in doc($doc)//artist
where $artist[@rank = 2]
return <artistData>{$artist}</artistData>
```

Now what about RDF Data?

Lots of RDF Data out there, ready to “query the Web”



Nightwish

Nightwish live in Melbourne, Australia, on January 30, 2008

Background information

Origin	Kitee, Finland
Genres	Symphonic metal, power metal
Years active	1996–present

About: Nightwish

An Entity of Type : organisation, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

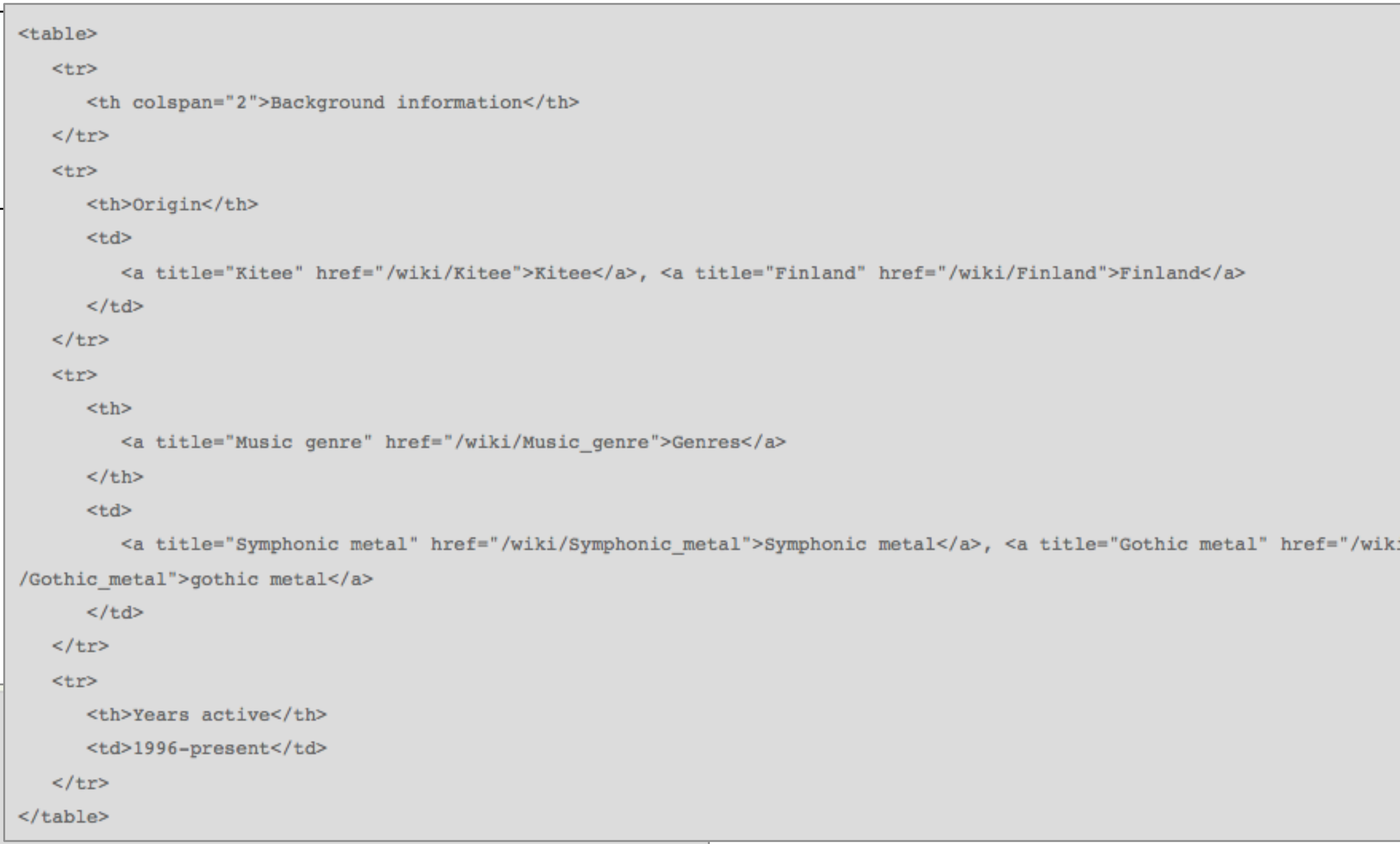
Nightwish is a Finnish symphonic metal band from Kitee. Formed in 1996 by songwriter and keyboardist Tuomas Holopainen, guitarist Emppu Vuorinen, and former vocalist Tarja Turunen, Nightwish's current line-up has five members, although Tarja has been replaced by Anette Olzon and the original bassist, Sami Vänskä, has been replaced by Marco Hietala, who also took over the male vocalist part; previously male vocal-parts were done by Tuomas or guest singers.

dbpprop:currentMembers	dbpedia:Tuomas_Holopainen dbpedia:Jukka_Nevalainen dbpedia:Marco_Hietala dbpedia:Emppu_Vuorinen dbpedia:Anette_Olzon
dbpprop:genre	dbpedia:Power_metal dbpedia:Symphonic_metal
dbpprop:imageSize	250 (xsd:integer)
dbpprop:label	dbpedia:Nuclear_Blast dbpedia:Roadrunner_Records dbpedia:Drakkar_Entertainment dbpedia:Spinefarm_Records dbpedia:NEMS_Enterprises_(label) dbpedia:Century_Media_Records
dbpprop:landscape	Yes
dbpprop:name	Nightwish

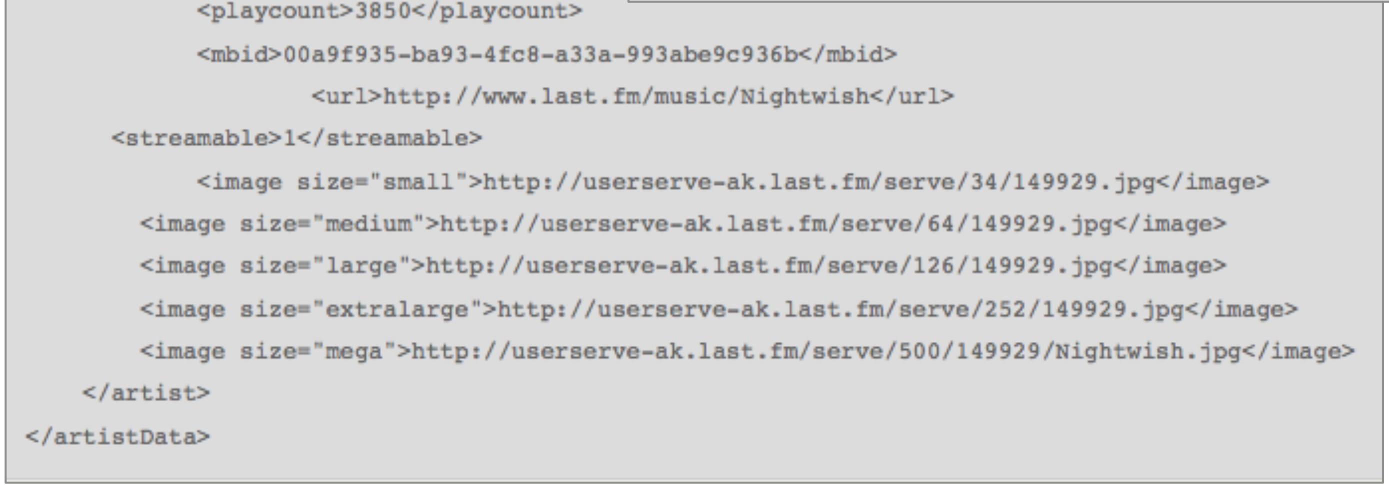
As of September 2010

XML vs. RDF

***XML:** "treelike" semi-structured Data (mostly schema-less, but "implicit" schema by tree structure… not easy to combine, e.g. how to combine lastfm data with wikipedia data?*



```
<table>
   <tr>
      <th colspan="2">Background information</th>
   </tr>
   <tr>
      <th>Origin</th>
      <td>
         <a title="Kitee" href="/wiki/Kitee">Kitee</a>, <a title="Finland" href="/wiki/Finland">Finland</a>
      </td>
   </tr>
   <tr>
      <th>
         <a title="Music genre" href="/wiki/Music_genre">Genres</a>
      </th>
      <td>
         <a title="Symphonic metal" href="/wiki/Symphonic_metal">Symphonic metal</a>, <a title="Gothic metal" href="/wik
/Gothic_metal">gothic metal</a>
      </td>
   </tr>
   <tr>
      <th>Years active</th>
      <td>1996-present</td>
   </tr>
</table>
```

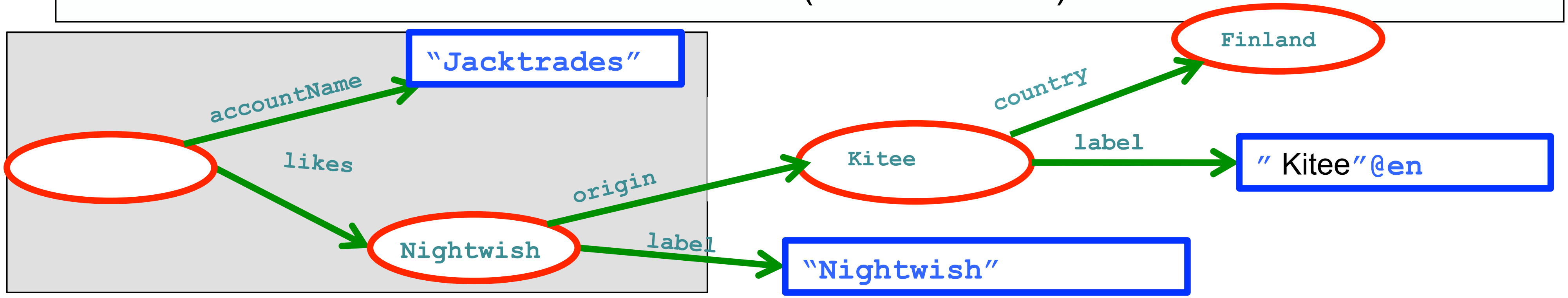


```
<artistData>
   <artist rank="2">
      <name>Nightwish</name>
            <playcount>3850</playcount>
            <mbid>00a9f935-ba93-4fc8-a33a-993abe9c936b</mbid>
                    <url>http://www.last.fm/music/Nightwish</url>
      <streamable>1</streamable>
            <image size="small">http://userserve-ak.last.fm/serve/34/149929.jpg</image>
        <image size="medium">http://userserve-ak.last.fm/serve/64/149929.jpg</image>
        <image size="large">http://userserve-ak.last.fm/serve/126/149929.jpg</image>
        <image size="extralarge">http://userserve-ak.last.fm/serve/252/149929.jpg</image>
        <image size="mega">http://userserve-ak.last.fm/serve/500/149929/Nightwish.jpg</image>
    </artist>
</artistData>
```

RDF

Simple, declarative, graph-style format
based on dereferenceable URIs (= Linked Data)

SIEMENS

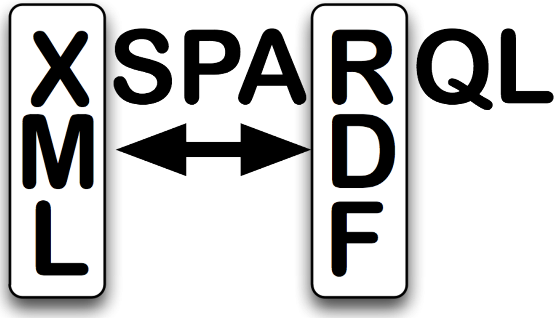


```
<http://dbpedia.org/resource/Nightwish> <http://dbpedia.org/property/origin>
                            <http://dbpedia.org/resource/Kitee> .
<http://dbpedia.org/resource/Nightwish> <http://www.w3.org/2000/01/rdf-schema#label>
                           ”Kitee”@es .
```

```
_:x <http://xmlns.com/foaf/0.1/accountName> “Jacktrades" .
_:x <http://graph.facebook.com/likes> <http://dbpedia.org/resource/Nightwish> .
```

XSPARQL

Idea: One approach to conveniently query XML and RDF side-by-side: XSPARQL



- Transformation language
- Consume and generate XML and RDF
- Syntactic extension of XQuery, ie.
  XSPARQL = XQuery + SPARQL

XSPARQL: Syntax overview (I)

Prefix declarations	P	declare namespace *prefix*="*namespace-URI*" or prefix *prefix*: <*namespace-URI*>	
Body:	F	for *var* [at *posVar*] in *FLOWR' expression*	
	L	let *var* := *FLWOR' expression*	
	W	where *FLWOR' expression*	
Data Input (XML or RDF)	O	order by *FLWOR' expression*	or
	F'	for *varlist* [at *posVar*]	
	D	from / from named (<*dataset-URI*> or *FLWOR' expr.*)	
	W	where { *pattern* }	
	M	order by *expression* limit *integer* > 0 offset *integer* > 0	
Data Output (XML or RDF)	C	construct { *template (with nested FLWOR' expressions)* }	or
	R	return *XML+ nested FLWOR' expressions*	

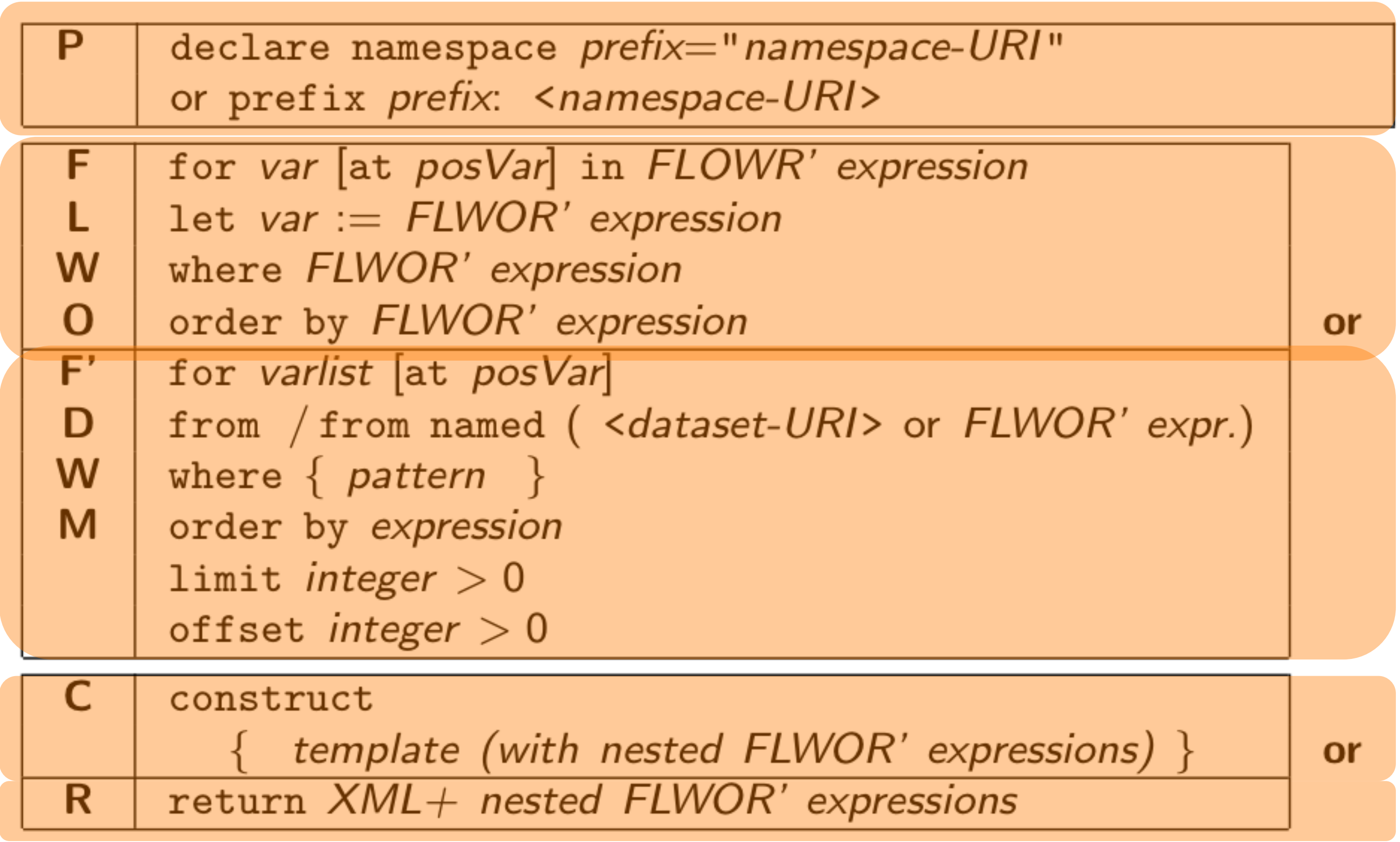
XSPARQL Syntax overview (II)

XQuery or SPARQL prefix declarations

Any XQuery query

“SPARQL-FOR-Clause” represents a SPARQL query

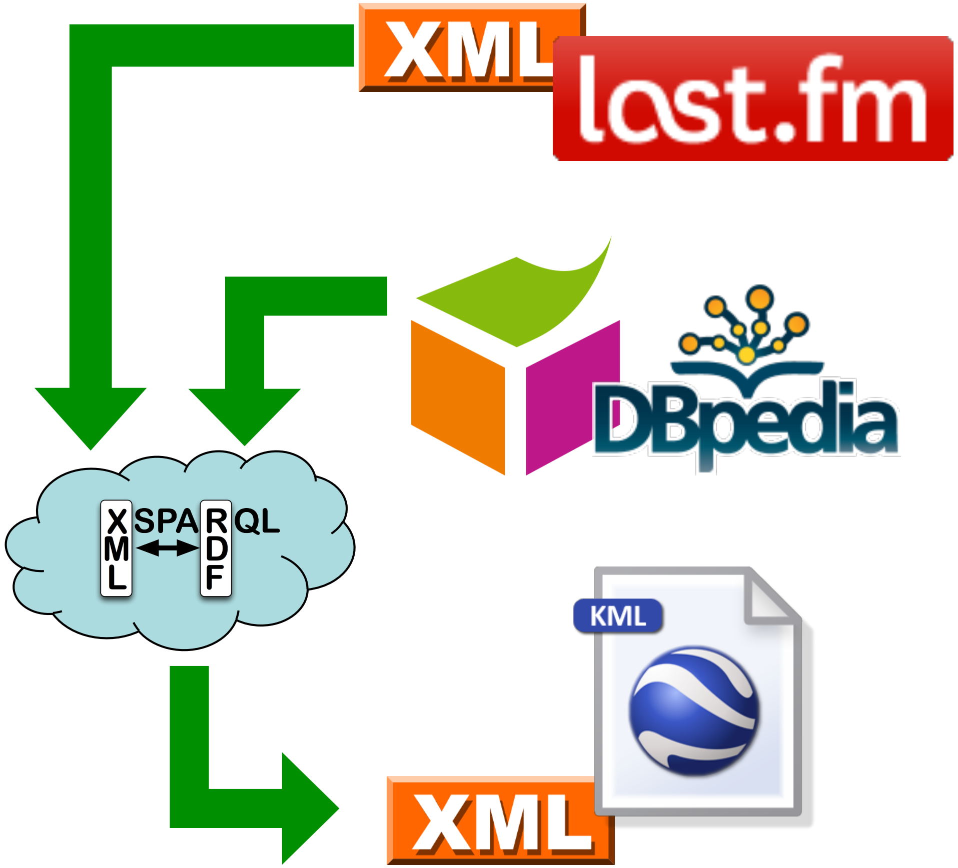
construct allows to create RDF



P	declare namespace *prefix*="*namespace-URI*" or prefix *prefix*: <*namespace-URI*>	
F	for *var* [at *posVar*] in *FLOWR' expression*	
L	let *var* := *FLWOR' expression*	
W	where *FLWOR' expression*	
O	order by *FLWOR' expression*	or
F'	for *varlist* [at *posVar*]	
D	from / from named (<*dataset-URI*> or *FLWOR' expr.*)	
W	where { *pattern* }	
M	order by *expression*	
	limit *integer* > 0	
	offset *integer* > 0	
C	construct { *template (with nested FLWOR' expressions)* }	or
R	return *XML+ nested FLWOR' expressions*	

SIEMENS

Use case



XSPARQL: Convert XML to RDF

Query:

Convert Last.fm top artists of a user into RDF

```
prefix lastfm: <http://xsparql.deri.org/lastfm#>

let $doc := "http://ws.audioscrobbler.com/2.0/?method=user.gettopartists"
for $artist in doc($doc)//artist
where $artist[@rank < 6]
construct { [] lastfm:topArtist {$artist//name};
               lastfm:artistRank {$artist//@rank} . }
```

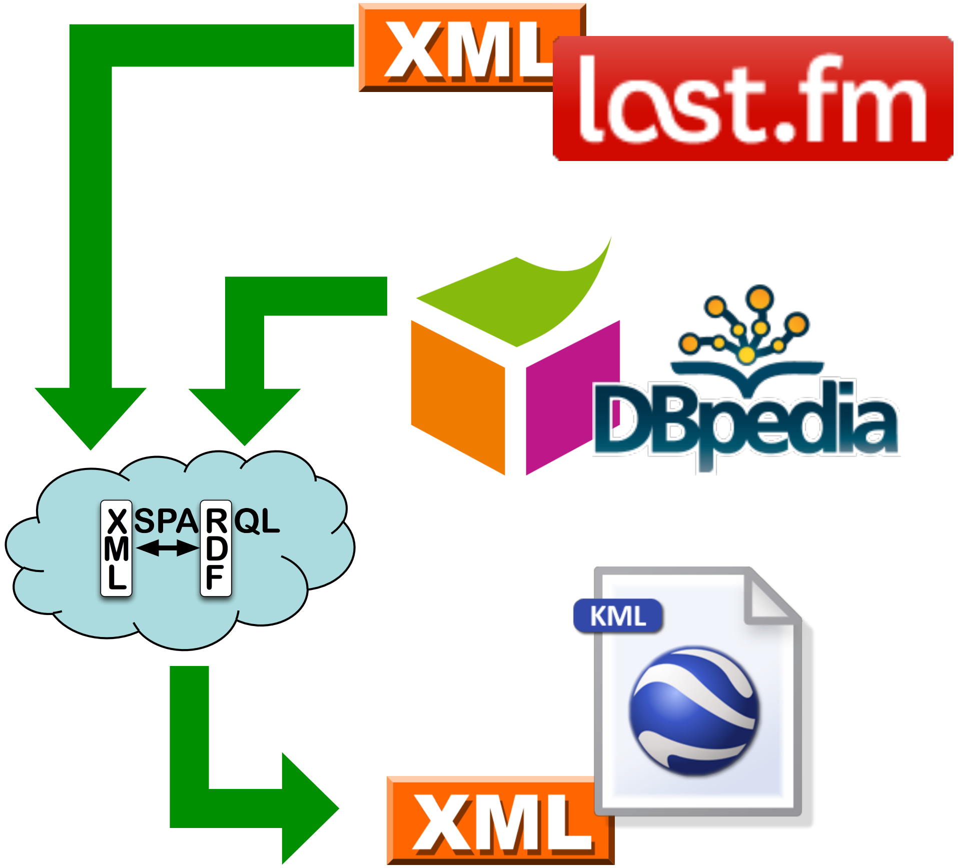
Result:

```
@prefix lastfm: <http://xsparql.deri.org/lastfm#> .

[ lastfm:topArtist "Therion" ;  lastfm:artistRank "1" ] .
[ lastfm:topArtist "Nightwish" ;  lastfm:artistRank "2" ] .
[ lastfm:topArtist "Blind Guardian" ;  lastfm:artistRank "3" ] .
[ lastfm:topArtist "Rhapsody of Fire" ;  lastfm:artistRank "4" ] .
[ lastfm:topArtist "Iced Earth" ;  lastfm:artistRank "5" ] .
```

XSPARQL construct generates valid Turtle RDF

SIEMENS

Use case



XSPARQL: Integrate RDF sources

Query:

Retrieve the origin of an artist from DBPedia: Same as the SPARQL query

```
prefix dbprop: <http://dbpedia.org/property/>
prefix foaf:   <http://xmlns.com/foaf/0.1/>

construct { $artist foaf:based_near $origin }
from <http://dbpedia.org/resource/Nightwish>
where { $artist dbprop:origin $origin }
```

Issue: determining the artist identifiers

DBPedia does not have the map coordinates

GeoNames

XML

XSPARQL: Integrate RDF sources

Query:

Retrieve the origin of an artist from DBPedia ***including map coordinates***

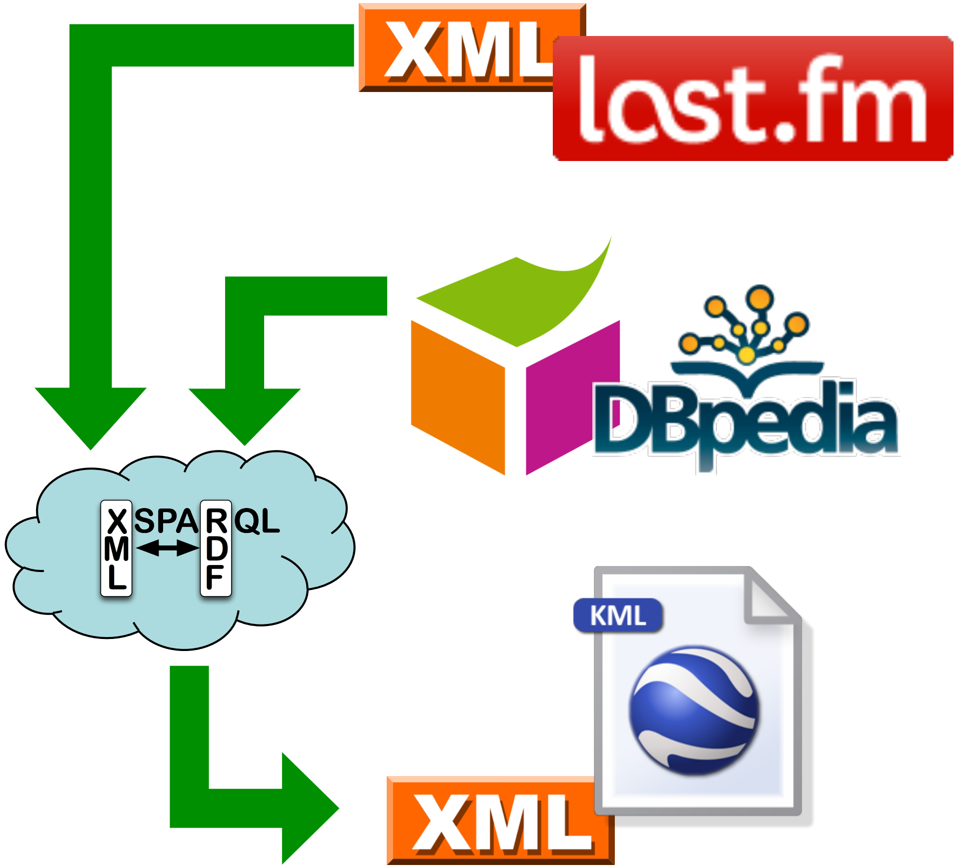
```
prefix wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
prefix dbprop: <http://dbpedia.org/property/>

for * from <http://dbpedia.org/resource/Nightwish>
where { $artist dbprop:origin $origin }
return
let $hometown :=
  fn:concat("http://api.geonames.org/search?type=rdf&q=",fn:encode-for-uri($origin))
for * from $hometown
where { [] wgs84_pos:lat $lat; wgs84_pos:long $long }
limit 1
construct { $artist wgs84_pos:lat $lat; wgs84_pos:long $long }
```

DBPedia does not have the map coordinates

GeoNames

XML

Use case



Output: KML XML format

```
<kml xmlns="http://www.opengis.net/kml/2.2">
   <Document>
      <Placemark>
         <name>Hometown of Nightwish</name>
         <Point>
            <coordinates>
              30.15,62.1,0
            </coordinates>
         </Point>
      </Placemark>
 </Document>
</kml>
```

KML format:

- root element: “kml”, then “Document”
- sequence of “Placemark”
- Each “Placemark” contains:
  - “Name” element
  - “Point” element with the “coordinates”

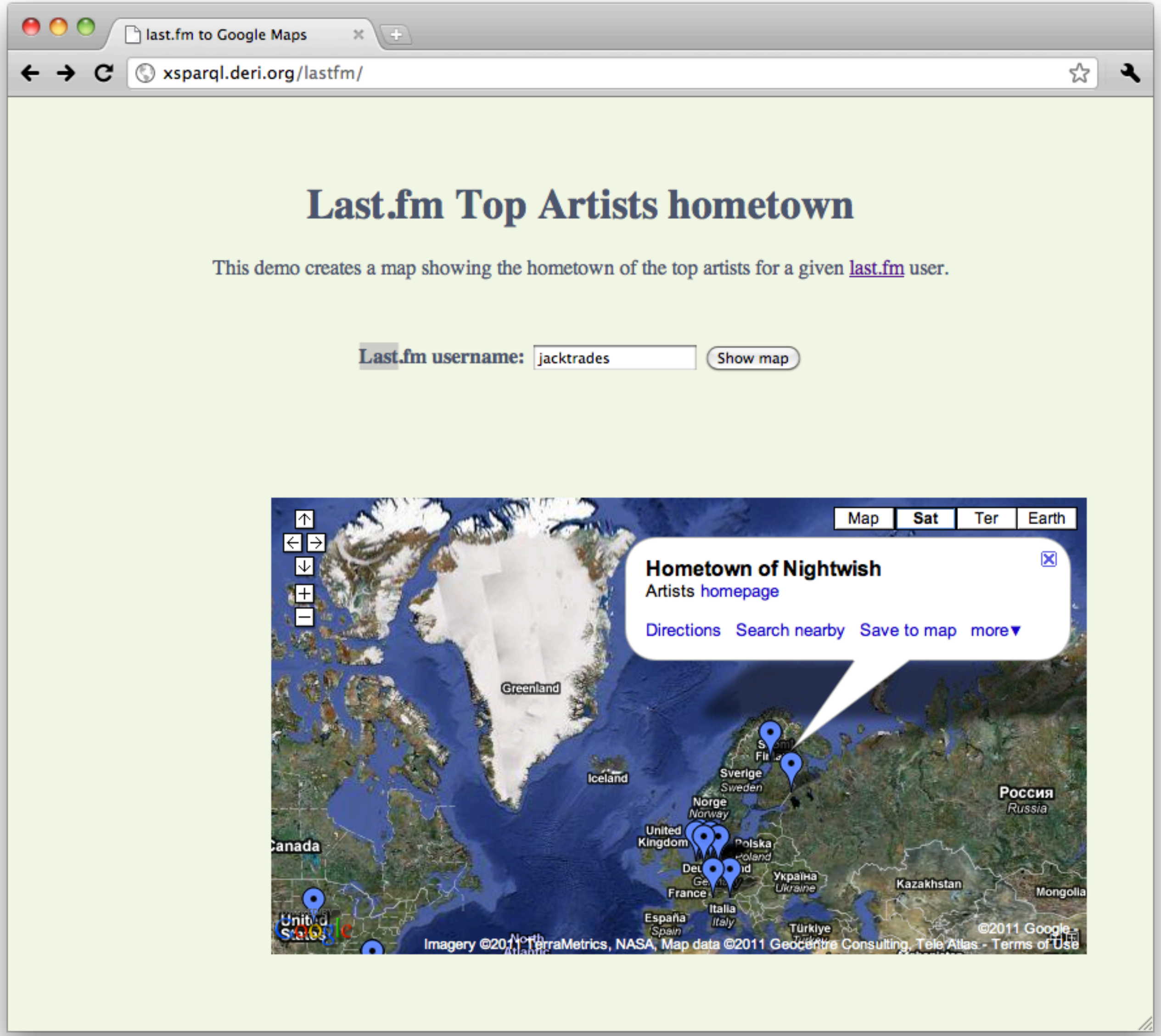
XSPARQL: Putting it all together

Query: Display top artists origin in a map

```
prefix dbprop: <http://dbpedia.org/property/>

<kml><Document>{
let $doc := "http://ws.audioscrobbler.com/2.0/?method=user.gettopartists”
for $artist in doc($doc)//artist
return let $artistName := fn:data($artist//name)
  let $uri := fn:concat("http://dbpedia.org/resource/", $artistName)
  for $origin from $uri
  where { [] dbprop:origin $origin }
  return
   let $hometown := fn:concat("http://api.geonames.org/search?type=rdf&q=",
                    fn:encode-for-uri($origin))
   for * from $hometown
   where { [] wgs84_pos:lat $lat; wgs84_pos:long $long }
   limit 1
   return <Placemark>
            <name>{fn:concat("Hometown of ", $artistName)}</name>
            <Point><coordinates>{fn:concat($long, ",", $lat, ",0")}
            </coordinates></Point>
           </Placemark>
}</Document></kml>
```

XSPARQL: Demo

http://xsparql.deri.org/lastfm



last.fm to Google Maps

xsparql.deri.org/lastfm/

Last.fm Top Artists hometown

This demo creates a map showing the hometown of the top artists for a given last.fm user.

Last.fm username: jacktrades Show map

Hometown of Nightwish

Artists homepage

Directions Search nearby Save to map more▼

XSPARQL: another example…

Federated Queries in SPARQL1.1

Find which persons in DBPedia have the same birthday as Axel (foaf-file):

SPARQL 1.1 has new feature SERVICE to query remote endpoints

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
{ [ foaf:birthday ?MyB ].

  SERVICE <http://dbpedia.org/sparql> { SELECT ?N WHERE {
    [ dbpedia2:born ?B; foaf:name ?N ]. FILTER ( Regex(str(?B),str(?MyB)) ) } }
}
```

Doesn't work!!! **?MyB** unbound in SERVICE query

Federated Queries in SPARQL1.1

Find which persons in DBPedia have the same birthday as Axel (foaf-file):

SPARQL 1.1 has new feature SERVICE to query remote endpoints

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?N ?MyB
FROM <http://polleres.net/foaf.rdf>
{ [ foaf:birthday ?MyB ].

  SERVICE <http://dbpedia.org/sparql> { SELECT ?N WHERE {
    [ dbpedia2:born ?B; foaf:name ?N ]. } }

 FILTER ( Regex(Str(?B),str(?MyB)) )
}
```

Doesn’t work either in practice ☹ as SERVICE endpoints often only returns limited results…

Federated Queries

Find which persons in DBPedia have the same birthday as Axel (foaf-file):

In XSPARQL:

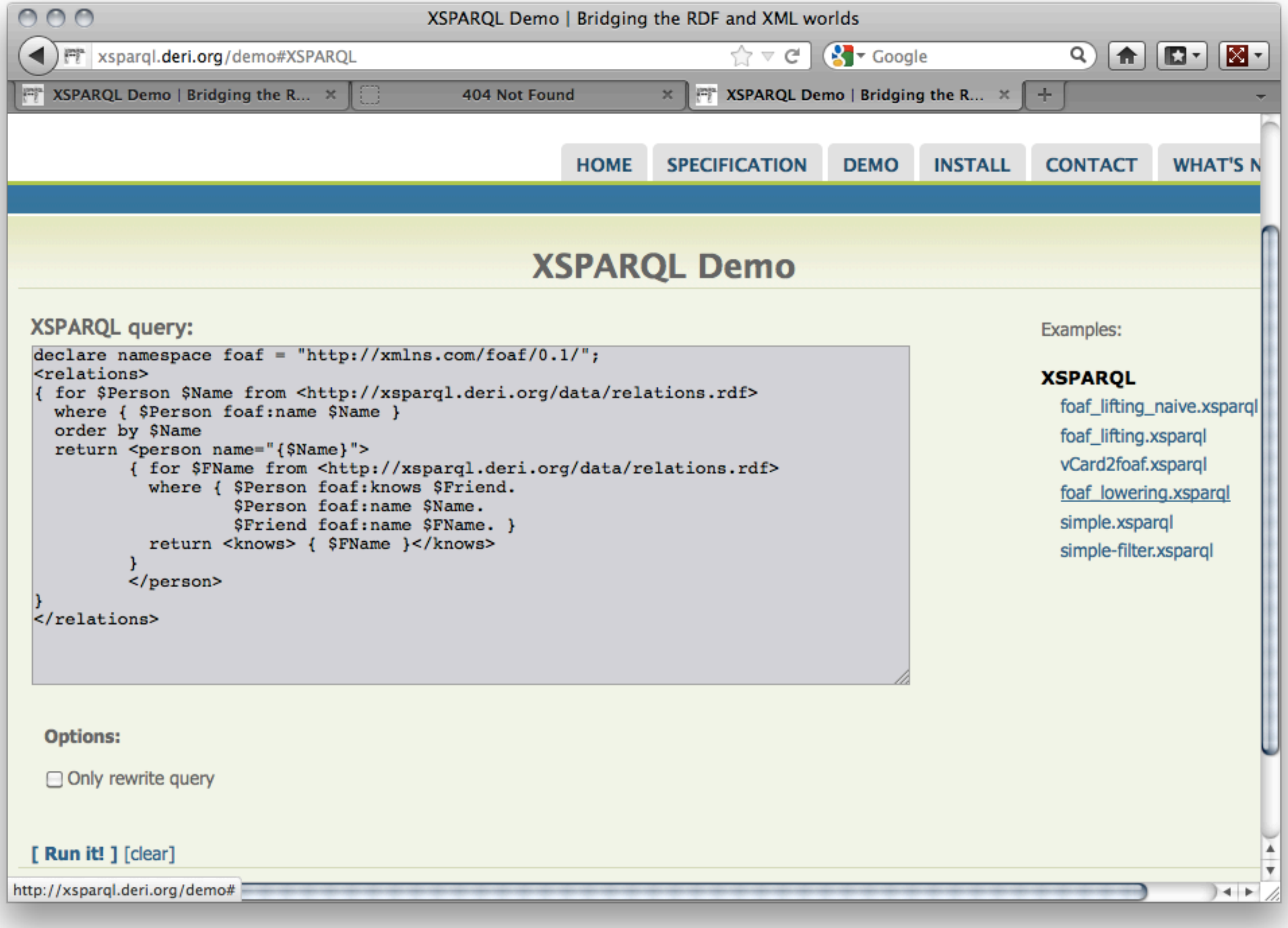
```
prefix dbprop: <http://dbpedia.org/property/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix : <http://xsparql.deri.org/bday#>

let $MyB := for * from <http://polleres.net/foaf.rdf>
           where { [ foaf:birthday $B ]. }
           return $B

for * from <http://dbpedia.org/> endpoint <http://dbpedia.org/sparql>
where { [ dbprop:born $B; foaf:name $N ].
           filter ( regex(str($B),str($MyB)) )  }
construct { :axel :sameBirthDayAs $N }
```

Specifies the endpoint to perform the query, similar to SERVICE in SPARQL1.1

Works! In XSPARQL bound values (**?MyDB**) are **injected** into the SPARQL subquery
→ More direct control over "query execution plan"

Test Queries and play around…

http://xsparql.deri.org/demo



XSPARQL Demo | Bridging the RDF and XML worlds
xsparql.deri.org/demo#XSPARQL
Google
XSPARQL Demo | Bridging the R...
404 Not Found
XSPARQL Demo | Bridging the R...
HOME
SPECIFICATION
DEMO
INSTALL
CONTACT
WHAT'S N
http://xsparql.deri.org/demo#

XSPARQL Demo

XSPARQL query:

```
declare namespace foaf = "http://xmlns.com/foaf/0.1/";
<relations>
{ for $Person $Name from <http://xsparql.deri.org/data/relations.rdf>
  where { $Person foaf:name $Name }
  order by $Name
  return <person name="{$Name}">
           { for $FName from <http://xsparql.deri.org/data/relations.rdf>
             where { $Person foaf:knows $Friend.
                     $Person foaf:name $Name.
                     $Friend foaf:name $FName. }
             return <knows> { $FName }</knows>
           }
           </person>
}
</relations>
```

Examples:

XSPARQL

- foaf_lifting_naive.xsparql
- foaf_lifting.xsparql
- vCard2foaf.xsparql
- foaf_lowering.xsparql
- simple.xsparql
- simple-filter.xsparql

Options:

□ Only rewrite query

[Run it!] [clear]

Details about XSPARQL1.1 semantics and implementation

SIEMENS

Check our Technical Report (just accepted at Springer's Journal of Data Semantics):

Stefan Bischof, Stefan Decker, Thomas Krennwallner, Nuno Lopes, Axel Polleres. **Mapping between RDF and XML with XSPARQL**. Technical Report 2011. http://www.deri.ie/fileadmin/documents/DERI-TR-2011-04-04.pdf

BTW: First author started in this lecture two years ago!
→ If you are interested in Internships, Diploma theses, PhD theses let me know!)