

# Engineering and Semantic Web

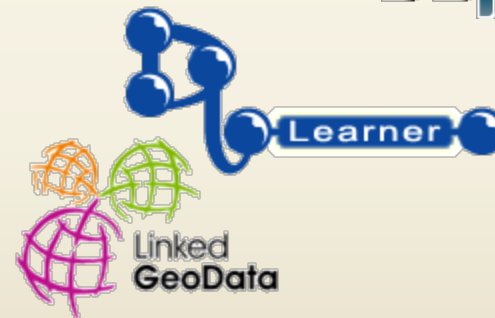
## AKSW aims:

- Contributing to the advancement of science in Semantic Web, Knowledge Engineering, Software Engineering
- Cost efficient, high-impact R&D, which proves usefulness at an early stage
- Bridge the gap between science and applications



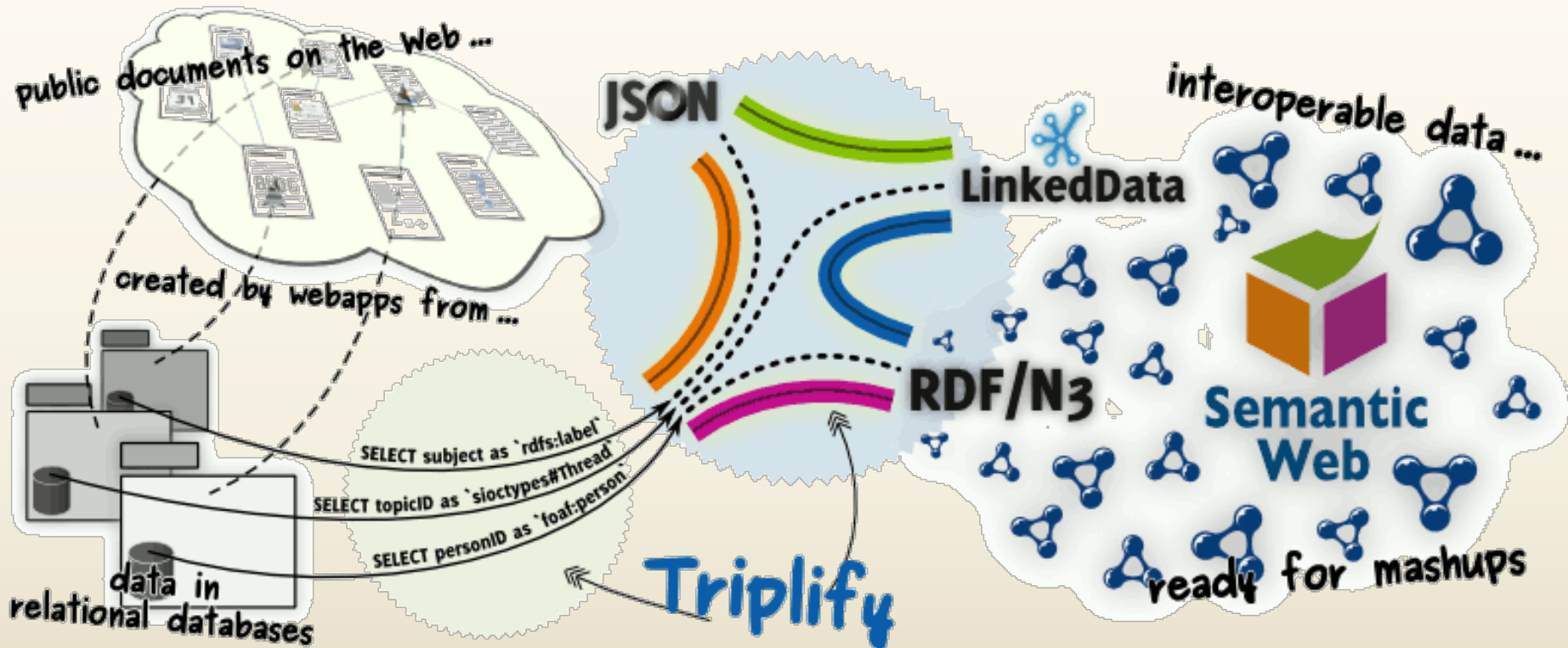
## Main AKSW results:

- **OntoWiki** - Semantic Data Wiki,
- **DBpedia** - Wikipedia knowledge extraction,
- **DL-Learner** - Ontology Machine Learning,
- **Triplify** - light-weight semantic integration,
- **LinkedGeoData** – triplification of OpenStreetMap



**AKSW actively educates students in Semantic Technologies and serves the community** by (co-) organizing events such as Conference on Social Semantic Web, I-Semantics, Scripting for The Semantic Web workshop series etc.

# Triplify Big Picture



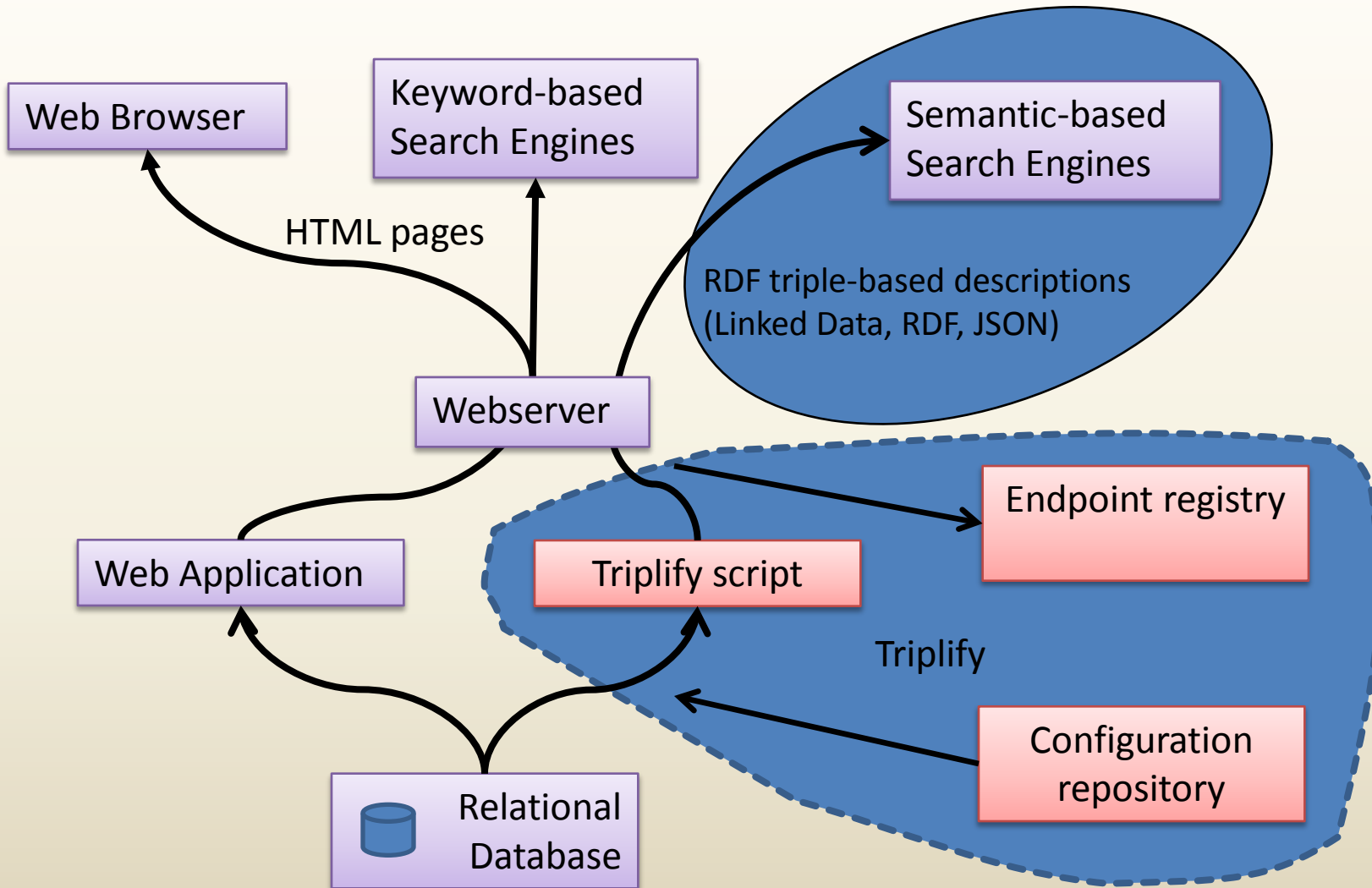
# Triplify Motivation

- **overcome the chicken-and-egg dilemma of missing semantic representations and search facilities on the Web**
- Triplify leverages relational representations behind existing Web applications:
  - often open-source, deployed hundred thousand times
  - structure and semantics encoded in relational database schemes (behind Web apps) is not accessible to Web search engines, mashups etc.

Project	Area	Downloads
<a href="#">phpBB</a>	discussion forum	235480
<a href="#">Gallery</a>	photo gallery	166005
<a href="#">XOOPS</a>	CMS	115807
<a href="#">Coppermine</a>	photo gallery	113854
<a href="#">Typo3</a>	CMS	63641
<a href="#">Liferay Portal</a>	Portal	39615
<a href="#">eGroupWare</a>	group ware	33865
<a href="#">Alfresco</a>	CMS	31914
<a href="#">e107</a>	CMS	19996
<a href="#">Lifetype</a>	Blogging	16730
<a href="#">Plone</a>	CMS	13993
<a href="#">Compiere</a>	ERP + CRM	13718
<a href="#">WebCalendar</a>	Calendar	12832
<a href="#">Nucleus</a>	Blogging	12739
<a href="#">Tikiwiki</a>	Wiki	6368

Monthly Web application downloads at Sourceforge

# Overview



# Solution overview

- SQL is the industry standard language for **relational transformations**
- **Extend SQL with a few syntactic constructs** opaque to the SQL query processor
- **Map** URL patterns to sets of SQL query patterns
- For a concrete URL request, **replace placeholders** in the query patterns, execute the query
- **Transform resulting relations** into various RDF serializations (multiple view to class approach)

# Triplify Solution:

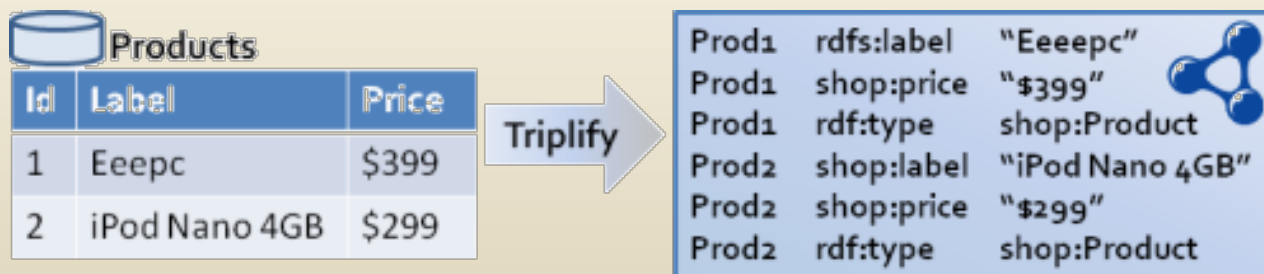
## SQL-SELECTs map relational to RDF

### Triplify Configuration:

- number of *SQL queries* selecting information, which should be made publicly available.

### Special SQL query result structure required (in order to convert results into RDF:

- *first column must contain identifiers* for generating instance URIs (i.e. the primary key of DB table)
- *column names are used to generate property URIs*, renaming columns allows to reuse properties from existing vocabularies such as Dublin Core, FOAF, SIOC
  - e.g. `SELECT id, name AS 'foaf:name' FROM users`
- *individual cells contain data values or references to other instances* (eventually constitute the objects of resulting triples)



# Example: Wordpress Blog Posts

Associate the URL path fragment 'post' with a number of SQL patterns:

**http://blog.aksw.org/triplify/post/(xxx)**

1 **SELECT** id, post\_author **AS** 'sioc:has\_creator->user',  
 post\_title **AS** 'dc:title',  
 post\_content **AS** 'sioc:content', post\_date **AS**  
 'dcterms:modified^^xsd:dateTime',  
 post\_modified **AS** 'dcterms:created^^xsd:dateTime'  
**FROM** posts  
**WHERE** post\_status='publish' (**AND** id=xxx)

Object property

Datatype property

2 **SELECT** post\_id id, tag\_label **AS** 'tag:taggedWithTag'  
**FROM** post2tag **INNER JOIN** tag **ON**(post2tag.tag\_id=tag.tag\_id)  
 (**WHERE** id=xxx)

3 **SELECT** post\_id id, category\_id **AS** 'belongsToCategory->category'  
**FROM** post2cat  
 (**WHERE** id=xxx)

# RDF Conversion

<http://blog.aksw.org/triplify/post/1>



1	id	post_author	post_title	post_content	post_date	post_modified
	1	5	New DBpedia release	Today we released ...	200810201635	200810201635

2	id	tag:taggedWithTag
	1	DBpedia
	1	Release
	..	

3	id	belongsToCategory
	1	34
	...	



<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>  
<http://blog.aksw.org/triplify/post/1>

sioc:has\_creator  
 dc:title  
 sioc:content  
 dcterms:modified  
 dcterms:created  
 tag:taggedWithTag  
 tag:taggedWithTag  
 belongsToCategory

<http://blog.aksw.org/triplify/user/5>  
 "New DBpedia release"  
 "Today we released ..."  
 "20081020T1635"^^xsd:dateTime  
 "20081020T1635"^^xsd:dateTime  
 "DBpedia"  
 "Release"  
<http://blog.aksw.org/triplify/category/34>

# Triplify Implementation: Simplicity

- Expose semantics as simple as possible
  - **No** (new) **mapping languages** – easy to learn
  - **Few lines of code** – easy to plug-in
  - Simple, **reusable** configurations
- Available for most popular Web app languages
  - PHP (ready), Ruby/Python under development
- Works with most popular Web app DBs
  - MySQL (extensively tested), PHP-PDO DBs (SQLite, Oracle, DB2, MS SQL, PostgreSQL etc.) should work, not needed for Virtuoso 😊
- Triplify exposes RDF/Ntriples, LinkedData and RDF/JSON

# Example Config

```

<?php
include('../wp-config.php');

$triplify['namespaces']=array(
  'vocabulary'=>'http://triplify.org/vocabulary/Wordpress/',
  'foaf'=>'http://xmlns.com/foaf/0.1/',
  ... );

$triplify['queries']=array(
  'post'=>array(
    "SELECT id,post_author 'sioc:has_creator->user',post_date 'dcterms:created',post_title 'dc:title', post_content 'sioc:content',
      post_modified 'dcterms:modified' FROM {$table_prefix}posts WHERE post_status='publish'",
    "SELECT post_id id,tag_id 'tag:taggedWithTag' FROM {$table_prefix}post2tag",
    "SELECT post_id id,category_id 'belongsToCategory' FROM {$table_prefix}post2cat",
  ),
  'tag'=>"SELECT tag_ID id,tag 'tag:tagName' FROM {$table_prefix}tags",
  'category'=>"SELECT cat_ID id,cat_name 'skos:prefLabel',category_parent 'skos:narrower' FROM {$table_prefix}categories",
  'user'=>array(
    "SELECT id,user_login 'foaf:accountName',SHA(CONCAT('mailto:',user_email)) 'foaf:mbox_sha1sum',
      user_url 'foaf:homepage',display_name 'foaf:name' FROM {$table_prefix}users",
    "SELECT user_id id,meta_value 'foaf:firstName' FROM {$table_prefix}usermeta WHERE meta_key='first_name'",
    "SELECT user_id id,meta_value 'foaf:family_name' FROM {$table_prefix}usermeta WHERE meta_key='last_name'",
  ),
  'comment'=>"SELECT comment_ID id,comment_post_id 'sioc:reply_of',comment_author AS 'foaf:name',
    SHA(CONCAT('mailto:',comment_author_email)) 'foaf:mbox_sha1sum', comment_author_url 'foaf:homepage',
    comment_date AS 'dcterms:created', comment_content 'sioc:content',comment_karma,comment_type
    FROM {$table_prefix}comments WHERE comment_approved='1'",
  );

$triplify['objectProperties']=array(
  'sioc:has_creator'=>'user', 'tag:taggedWithTag'=>'tag', 'belongsToCategory'=>'category','skos:narrower'=>'category','sioc:reply_of'=>'post');

$triplify['classMap']=array('user'=>'foaf:person', 'post'=>'sioc:Post', 'tag'=>'tag:Tag', 'category'=>'skos:Concept');

$triplify['TTL']=0; // Caching

$triplify['db']=new PDO('mysql:host='.DB_HOST.'.dbname='.DB_NAME,DB_USER,DB_PASSWORD);
?>

```

# Configuration repository

- Triplify configurations are shared at:  
<http://Triplify.org>
- Existing configurations for  
OpenConf, Wordpress, WackoWiki, Drupal,  
OJS, Joomla, osCommerce, Gallery, phpBB,  
OMDB ...

# Triplify Endpoint Registry

- Simple REST endpoint registry:  
[http://triplify.org/Registry/?url=%rdf\\_source\\_URL%](http://triplify.org/Registry/?url=%rdf_source_URL%)
- Itself available as Linked Data endpoint
- Enables building of mashups, vertical search and other applications using information from many sources – product search, blog search etc.

# Triplify Temporal Extension

**Problem:** How do next generation search engines know something changed on the Data Web?

## Different solutions:

- *Try to crawl always everything:* currently deployed on the Web
- *Ping a central update notification service:*  
PingTheSemanticWeb.com – will probably not scale if the Data Web gets really deployed
- *Each linked data endpoint publishes an update log:*  
Triplify Update Logs

# Triplify Temporal Extension

special update path and vocabulary

<http://example.com/Triplify/update>

<http://example.com/Triplify/update/2007> rdf:type **update:UpdateCollection** .

<http://example.com/Triplify/update/2008> rdf:type **update:UpdateCollection** .

<http://example.com/Triplify/update/2008>

<http://example.com/Triplify/update/2008/Jan> rdf:type **update:UpdateCollection** .

<http://example.com/Triplify/update/2008/Feb> rdf:type **update:UpdateCollection** .

Nesting continues until we finally reach an URL, which exposes all updates performed in a certain second in time...

<http://example.com/Triplify/update/2008/Jan/01/17/58/06>

<http://example.com/Triplify/update/2008/Jan/01/17/58/06/user123>

**update:updatedResource** <http://example.com/Triplify/users/JohnDoe> ;

**update:updatedAt** "20080101T17:58:06"^^<xsd:dateTime> ;

**update:updatedBy** <http://example.com/Triplify/users/JohnDoe> .

# LOD Update log generation

Updates have to be logged in the DB

Update log queries have to expose a date as first column:

```
$triplify['queries']=array(
```

```
...
```

```
'update'=>"SELECT p.changed AS id,  
p.id AS 'update:updatedResource->project'  
FROM project p",
```

```
);
```

# Triplify Spatial Extension: Linked Open Geo Data



Spatial data is crucial for the Data Web in order to interlink geographically linked resources.

Open Street Map project (OSM) collects, organizes and publishes geo data the wiki way:

- **80.000 OSM users** collected data about **22M km ways** (roads, highways etc.) on earth, **25T km are added daily**
- OSM contains a vast amount **points-of-interest** descriptions e.g. shops, amenities, sports venues, businesses, touristic and historic sights.

Goal: publish OSM geo data, interlink it with other data sources and provide efficient means for browsing and authoring:

- **Open Street Map data extraction** works on the basis of OSM database dumps, a bi-directional live integration of OSM and our Linked Geo Data browser and editor is currently in the works.
- **Triplify spatial data publishing**, the Triplify script for publishing linked data from relational databases is extended for publishing geo data, in particular with regard to the retrieval of information about geographical areas.
- **LinkedGeo Data browser and editor** is a facet-based browser for geo content, which uses an OLAP inspired hypercube for quickly retrieving aggregated information about any user selected area on earth.

# Triplify Spatial Extension



How to publish geo-data using Triplify?

Lon
Lat
Radius
Attribute
Value  
<http://linkedgedata.org/near/48.213056,16.359722/1000/amenity=Hotel>

<http://linkedgedata.org/node/212331>

<http://linkedgedata.org/node/944523>

<http://linkedgedata.org/node/234091>

<http://linkedgedata.org/way/56719>

node/150760824

amenity	"pub";
created_by	"JOSM";
distance	"5995";
name	"La friolera";
geo#lat	"40.4474";
geo#long	"-3.7173".

# Faceted Linked-Geo-Data Browser



LinkedGeoData.org

This faceted Linked Geo Data browser is based on data obtained from the [OpenStreetMap project](#) (released under [CC-BY-SA](#)) and was developed by [AKSW research group](#).

Facets Instances Search: powered by Namefinder Link PDF RDF-Export

**Class hierarchy**

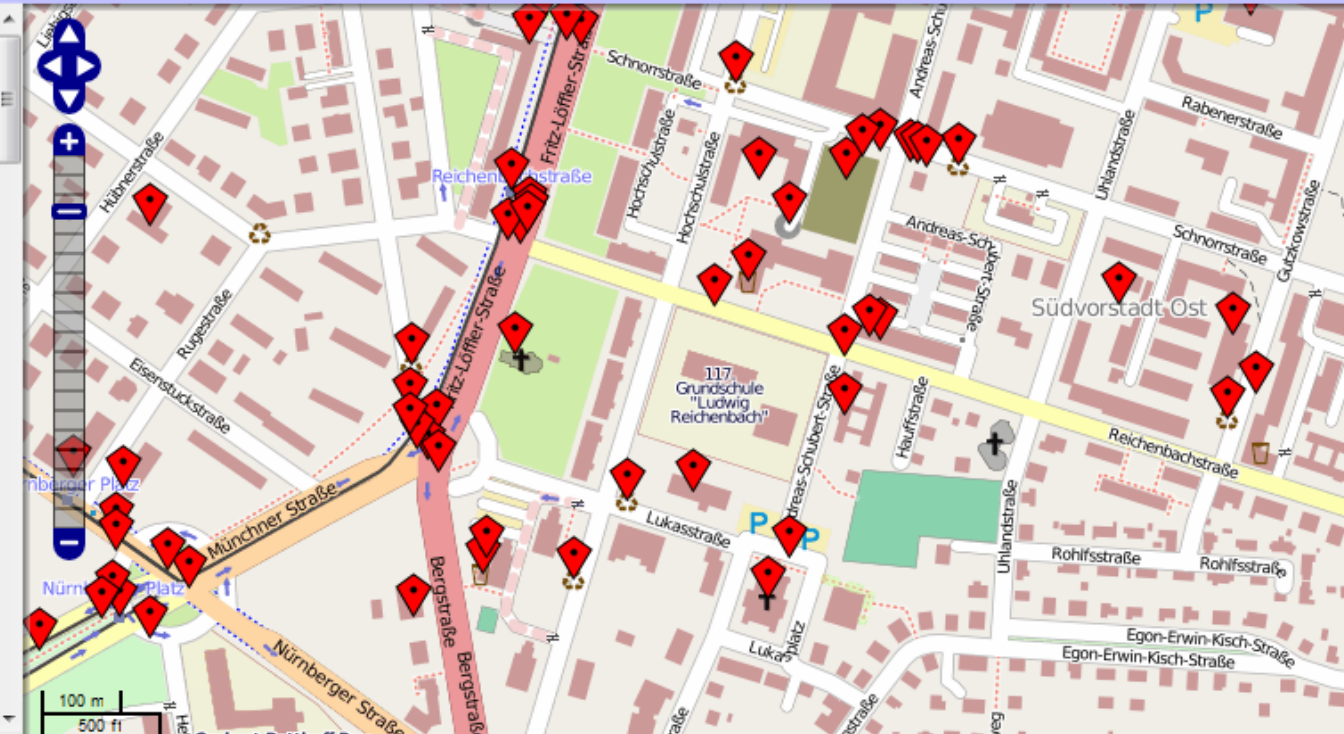
- [+] [amenity](#)(30)
- [+] [highway](#)(17)
- [+] [leisure](#)(1)
- [+] [man\\_made](#)(1)
- [+] [railway](#)(4)
- [+] [shop](#)(7)

**Properties**

- [+] [cuisine](#)(1)
- [+] [denomination](#)(2)
- [+] [place](#)(1)
- [+] [religion](#)(2)

Query took 19ms.

1. highway: traffic\_signals
2. highway: traffic\_signals
3. **NÄ¼rnberger Platz**  
railway: tram\_stop
4. **NÄ¼rnberger Platz**  
railway: tram\_stop
5. **NÄ¼rnberger Platz**  
highway: bus\_stop  
railway: tram\_stop
6. amenity: post\_box
7. amenity: telephone
8. amenity: bench
9. amenity: bicycle\_parking
10. amenity: recycling
11. **Club 11**  
amenity: pub
12. man\_made: MDF
13. amenity: bicycle\_parking



# RDB2RDF tool comparison

Tool	Triplify	D2RQ	Virtuoso RDF Views
Technology	Scripting languages (PHP)	Java	Whole middleware solution
SPARQL endpoint	-	X	X
Mapping language	SQL	RDF based	RDF based
Mapping generation	Manual	Semi-automatic	Manual
Scalability	Medium-high (but no SPARQL)	Medium	High

More at: <http://esw.w3.org/topic/Rdb2RdfXG/StateOfTheArt>

# Conclusion

- Triplify **supports the “long tail”** of deployed Web applications
- Publishing RDF and Linked Data is **simple**
- Support for **temporal** and **spatial** data dimensions
  - LOD Update Logs enable differential crawling
  - Linkedgeodata.org provides spatial identifiers for most parts of the world
- More comprehensive solutions are (still) required for SPARQL support



# Thanks!

**Sören Auer**

[auer@informatik.uni-leipzig.de](mailto:auer@informatik.uni-leipzig.de)

Research group Agile Knowledge Engineering & Semantic Web  
(AKSW): <http://aksw.org>

- [Triplify.org](http://Triplify.org)
- [DBpedia.org](http://DBpedia.org)
- [OntoWiki.net](http://OntoWiki.net)
- [OpenResearch.org](http://OpenResearch.org)
- <http://aksw.org/projects/xOperator>
- [DL-Learner.org](http://DL-Learner.org)
- [Cofundos.org](http://Cofundos.org)