PlanetData

- Network of Excellence funded by EU (7th FWP)
- Aim: establish an interdisciplinary, sustainable European community on large-scale data management
  - Purposeful data exposure
  - Novel and improved applications

Objectives

- Addressing challenges through integrated research
- Data and technology through PlanetData Lab
- Impact through training, dissemination, standardization and networking
- Openness and flexibility through PlanetData Programs

http://www.planet-data-eu
Workplan highlights

- Methods and techniques to publish, access and manage stream-like data
- **Quality assessment** of interlinked data sets, including best practices for the representation and usage of spatio-temporal information
- **Provenance and access control** framework for Linked (Stream) Data
- **Data sets** and **vocabularies**, including **best practices** for publishing and managing self-descriptive data
- **Linked Services** and **Processes** as an instrument to develop applications

- Yearly **summer school** co-located with Extended Semantic Web Conference

- **PlanetData Programs**
Linked Data and APIs
Session 1: Linked Data

Günter Ladwig, Steffen Stadtmüller, Andreas Harth
Outline

- Session 1: Linked Data (9:00 – 10:30)
  - Introduction
  - Linked Data Principles
  - Publishing Linked Data
  - Accessing Linked Data

- Session 2: Linked APIs (10:45 – 12:15)
  - Motivation
  - Preliminaries
  - Linked API Guideline
  - Real World Example
INTRODUCTION
The Web today

- Web of Documents
- Content readable by humans
- Links allow discovery of new content
Data on the Web

http://www.homepage1.de

Web Application 1

Database 1

http://www.homepage2.de

Web Application 2

Database 2
Web of Data

- Linking data instead of documents
- Machine readable
- “Web as a database”
Motivation

- With increased use of computers more and more data is being stored
  - Organisations rely on data for business decisions
  - Data drives policy decisions in government
  - Individuals rely on data from the Web for information and communication

- Data volumes explode
  - More and more data available on the Web is represented in Semantic Web standards
  - Linking Open Data (LOD) initiative

- Semantic Web technologies facilitate the integration of data from multiple sources
- Combining data from multiple sources enables insights
Linked Data Now!

http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html
Linked Data on the Web
Linked Data on the Web
Linked Data on the Web
Linked Data on the Web

2008-03
Linked Data on the Web

As of September 2008
Linked Data on the Web
Linked Data on the Web

As of July 2009

2009-07
Linked Data on the Web
Linked Data on the Web
Types of Data in the Linking Open Data Cloud

http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (Sept 2010)
DBpedia

- Linked Data version of Wikipedia
- Scripts that extract data (text, links, infoboxes) from Wikipedia
- Published as Linked Data
- Interlinking hub in the Linked Data web

- Berlin
  - http://dbpedia.org/resource/Berlin

- Hegel
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel

- Marlene Dietrich
  - http://dbpedia.org/resource/Marlene_Dietrich
BBC Music

- Data about BBC (radio) programmes, artists, songs…
- Combination of BBC-internal data (playlists), MusicBrainz (artists, albums), Wikipedia (artists)
- Underpinning the BBC Music website
- Data published according to Linked Data principles

Marlene Dietrich

- http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5
Virtual International Authority File (VIAF)

- Joint project of national libraries and related organisations
  - 21 institutions, among them the Deutsche Nationalbibliothek
- Provide access to “authority files”
- Matching and interlinking collections from participating institutions

- Hegel
  - http://viaf.org/viaf/89774942
- Marlene Dietrich
  - http://viaf.org/viaf/97773925
Semantic Technologies

- Semantic Web technologies, standardised by the W3C, are mature:
  - RDF recommendation in 1999, update in 2004
  - RDFa (RDF in HTML) note in 2008
  - RDFS recommendation in 2004
  - SPARQL recommendation in 2008
  - OWL recommendation in 2004, update in 2009

- Linked Data is a subset of the Semantic Web stack, including web architecture:
  - IRI (IETF RFC 3987, 2005)
  - HTTP (IETF RFC 2616, 1999)
LINKED DATA PRINCIPLES
Linked Data Principles

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs, so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData
Use URIs as names for things
1. Use URIs as Names for Things

- Use a unique identifier to denote things
- URIs are defined in RFC 2396

Hegel, Georg Wilhelm Friedrich
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
  - http://viaf.org/viaf/89774942
  - ...

Hegel, Georg Wilhelm Friedrich: Gesammelte Werke / Vorlesungen über die Logik
  - urn:isbn:978-3-7873-1964-0
Names for Things

“Now! That should clear up a few things around here!”
2

Use HTTP URIs
2. Use HTTP URIs

- Enables “lookup” of URIs
- Via Hypertext Transfer Protocol (HTTP)
- Piggy-backs on hierarchical Domain Name System to guarantee uniqueness of identifiers
- Uses established HTTP infrastructure
- Connects logical level (thing) with physical level (source)
- Important: distinction between name/“thing URI” and location/“source URI” („other resource“/„non-information resource“ vs. „information resource“)
Uniform Resource Identifiers (URIs)

Every URI identifies one “entity”

Semantic Web URIs usually use HTTP (Hypertext Transfer Protocol)
- QName (Qualified Name) as abbreviation, e.g., dbpedia:India
- Useful for looking up common prefixes: http://prefix.cc/
- IRI: Unicode-based “Internationalized Resource Identifiers”
Uniform Resource Identifiers (URIs)

- Every URI identifies one “entity”
- Semantic Web URIs usually use HTTP (Hypertext Transfer Protocol)
  - QName (Qualified Name) as abbreviation, e.g., dbpedia:India
  - Useful for looking up common prefixes: [http://prefix.cc/](http://prefix.cc/)
  - IRI: Unicode-based “Internationalized Resource Identifiers”

```
Protocol
    http://dbpedia.org/resource/India
  Hostname
```

01.12.2011
Uniform Resource Identifiers (URIs)

- Every URI identifies one “entity”
- Semantic Web URIs usually use HTTP (Hypertext Transfer Protocol)
  - QName (Qualified Name) as abbreviation, e.g., dbpedia:India
  - Useful for looking up common prefixes: http://prefix.cc/
  - IRI: Unicode-based “Internationalized Resource Identifiers”
Information Resources vs. Other Resources

Marlene Dietrich, the person

File containing data about Marlene Dietrich

Name?
Creator?
Birth date?
Last change date?
License?
Copyright?
...

RDF
Correspondence between thing-URI and source-URI („hash URIs“)

User Agent

HTTP
GET

RDF

Web Server

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5#artist

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf
Hypertext Transfer Protocol (HTTP)

$ curl -H "Accept: application/rdf+xml" -v http://www.w3.org/People/Berners-Lee/card#i

> GET /People/Berners-Lee/card HTTP/1.1
> User-Agent: curl/7.21.0
> Host: www.w3.org
> Accept: application/rdf+xml

< HTTP/1.1 200 OK
< Date: Mon, 28 Mar 2011 17:16:30 GMT
< Server: Apache/2
< Content-Location: card.rdf
< Content-Type: application/rdf+xml; qs=0.9
< Connection: close
Correspondence between thing-URI and source-URI („slash URIs“)

User Agent

HTTP GET 303 HTTP GET RDF

Web Server

http://dbpedia.org/resource/Marlene_Dietrich

http://dbpedia.org/data/Marlene_Dietrich

http://dbpedia.org/page/Marlene_Dietrich
URI Schemes

- Use identifier scheme within your namespace
- Mint crisp, short URIs without mention of underlying technology
- Keep your URI stable over time
- Often, there will be three URIs related to a given resource
  - An identifier for the resource
  - An identifier for a related resource suitable for human consumption in a browser (typically HTML)
  - An identifier for a related resource suitable for software agents (in RDF)
  - E.g.,
    - http://estatwrap.ontologycentral.com/id/tsieb010 (the resource)
    - http://estatwrap.ontologycentral.com/page/tsieb010 (the HTML page)
    - http://estatwrap.ontologycentral.com/data/tsieb010 (the
Provide useful information
3. Provide Useful Information

- When somebody looks up a URI, return data using the standards (RDF*, SPARQL)
- Resource Description Framework, a format for encoding graph-structured data (with URIs to identify nodes/vertices and links/edges)
Resource Description Framework

- Directed, labeled graph
- triple(subject, predicate, object)
  - subject: URI (or blank node)
  - predicate: URI
  - object: URI (or blank node) or RDF literal (string, integer, date…)

```
RDF Primer: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
```

```
dbpedia:Berlin

Subject

พวกเรา:Berlin

Predicate

dbpedia:capital

Subject

dbpedia:Germany

Predicate

rdfs:label

“Deutschland”@de

Object

rdfs:label

“Germany”@en
```

- RDF/XML is the most widely deployed serialisation
- Other serialisations possible (N-Triples, Turtle, Notation3…)
- RDF Primer: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
3. Provide Useful Information

@prefix dbpedia: <http://dbpedia.org/resource/>.
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.

dbpedia:India rdf:type dbpedia-owl:Country.
dbpedia:India rdfs:label "India"@en.
dbpedia:India dbpedia-owl:leaderName dbpedia:Manmohan_Singh.
dbpedia:India owl:sameAs
dbpedia:India owl:sameAs

http://dbpedia.org/resource/India
Merging Data with RDF
RDF Vocabularies

- RDF schema (RDFS) is the RDF vocabulary description language
  - Specifies classes and properties to describe vocabularies
- Classes
  - rdfs:Resource
  - rdfs:Class
  - rdfs:Property
- Properties
  - rdf:type
  - rdfs:label
  - rdfs:subClassOf and rdfs:subPropertyOf
  - rdfs:range and rdfs:domain
- Many pre-defined vocabularies are available
- Most conform to Linked Data principles, i.e. you can get more information about a schema term by looking it up via HTTP!
Reusing Existing Terms

- If there already exist terms for your problem, just reuse them
- Find terms by
  - Asking an expert
  - Inspecting examples
  - Using a semantic web search engine (e.g., SWSE, Sindice)

- Friend-of-a-Friend (FOAF) for describing people
- Dublin Core (DC) defines general metadata attributes (digital libraries)
- Semantically-Interlinked Online Communities (SIOC) for representing online communities.
- Simple Knowledge Organization System (SKOS) for representing taxonomies
- Creative Commons (CC) for describing license terms.

http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/
Creating Own Vocabularies

- Only define terms in your own namespace!
- Publish vocabularies according to linked data principles
- Assume both humans and machines look up vocabulary descriptions
- Link to existing vocabulary terms/established ontologies (e.g., via rdfs:subClassOf, rdfs:subPropertyOf)
- Neither overly specify nor overly constrain
Link to other URIs
4. Link to Other URIs

- Enable people (and machines) to jump from server to server
- External links vs. internal links (for any predicate)

- Using external vocabularies enables linking
- Vocabularies might be interlinked, too

- Special owl:sameAs links to denote equivalence of identifiers (useful for data merging)
Links to Other URIs

@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

dbpedia:India rdf:type dbpedia-owl:Country .
dbpedia:India rdfs:label "India"@en .
dbpedia:India dbpedia-owl:leaderName dbpedia:Manmohan_Singh .
dbpedia:India owl:sameAs <http://sw.opencyc.org/concept/Mx4rvVj7XJwpEbGdrcN5Y29ycA> .

http://dbpedia.org/resource/India
Interlinking Resources

- Implicitly link via inverse functional properties
- Explicitly via reusing URIs of other datasets
- Equivalence: owl:sameAs
- Some relation: rdfs:seeAlso
- Other object properties: foaf:knows, foaf:topic_interest…

- (Semi-)automatic interlinking
  - E.g. using similarity measures
  - Via toolkits such as Silk

- Data publishers should try to get external links

- Use established datasets, e.g. DBpedia as interlinking hub
Finding Related Datasets

- CKAN is the Comprehensive Knowledge Archive Network, a registry of open knowledge packages and projects (and a few closed ones)

- Useful for both marketing your own datasets and finding existing ones

http://ckan.net/about
Benefits of Linked Data

- Explicit, simple data representation
  - Common data representation (Resource Description Framework, RDF) hides underlying technologies and systems

- Distributed System
  - Decentralised distributed ownership and control facilitates adoption and scalability

- Cross-referencing
  - Allows for linking and referencing of existing data, via reuse of URIs

- Loose coupling with common language layer
  - Large scale systems require loose coupling, via HTTP as common access protocol

- Ease of publishing and consumption
  - Simple and easy-to-use systems and technologies to facilitate uptake

- Incremental data integration
  - Start with merged RDF graphs and provide mappings as you go
Challenges

- Ramp-up cost for data conversion
  - May be alleviated by semi-automatic mappings and adequate tool support for manual conversion
- Integrated data may be messy at first
  - But can be refined as need arises
- Distributed creation and loose coordination may result in inconsistencies
  - Can be detected, diagnosed, and fixed with appropriate tools
The Pedantic Web Group

- Get the community to contact publishers about errors/issues as they arise
- Get involved: http://pedantic-web.org/
- ~200 members!

Acknowledgements to: Aidan Hogan, Alex Passant, Me, Antoine Zimmermann, Axel Polleres, Michael Hausenblas, Richard Cyganiak, Stéphane Corlosquet
PUBLISHING LINKED DATA
Enterprise and Legacy Data

- Large amounts of data are managed in relational databases
- Web datasets are often published as JSON APIs
- Data published in XML or CSV

- Step 1: Ontology modelling driven by the current legacy data schema
- Step 2: Mappings expressed in procedural code/declarative languages
  - Text documents
  - Data dumps
  - Rewriting HTTP lookups to return RDF
  - Rewriting queries from SPARQL to SQL
Linking Text to Data

- Named entity recognition in text documents
- Establish links between text and structured description of entities
- OpenCalais detects entities (identified using internal IDs)
- WM Wikifier (Milne & Witten 2008) detects entities (identified using Wikipedia URIs)
- “SAP acquires Sybase for $5.8 billion, but why?” -> “SAP acquires Sybase for $5.8 billion, but why?”
Data Dumps

- Once-off data conversion via scripts that convert spreadsheet data to RDF
- Using Java/Perl/Python/Ruby/bash/XSLT…

- E.g. convert Eurostat CSV files to RDF using Java code
- Provide URIs for entities, e.g. [http://estatwrap.ontologycentral.com/dic/geo#de](http://estatwrap.ontologycentral.com/dic/geo#de) to denote Germany
Rewriting HTTP Lookups

- On-demand conversion via wrappers

- E.g. information about Facebook (instance of type Company)

- HTML via site
  - [http://www.crunchbase.com/company/facebook](http://www.crunchbase.com/company/facebook)

- JSON via API
  - [http://api.crunchbase.com/v/1/company/facebook.js](http://api.crunchbase.com/v/1/company/facebook.js)

- RDF/XML via Wrapper
  - [http://cbasewrap.appspot.com/company/facebook#id](http://cbasewrap.appspot.com/company/facebook#id)
Relational Data to RDF: D2R

- Express mappings between a database schema and an RDFS vocabulary

- Rewriting HTTP lookups on entity URIs
- Rewriting SPARQL queries to SQL

Bizer and Cyganiak, ISWC 2007
5 Star Data

★ Available on the web (whatever format), but with an open licence
★★ Available as machine-readable structured data (e.g. excel instead of image scan of a table)
★★★ as (2) plus non-proprietary format (e.g. CSV instead of excel)
★★★★ All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
★★★★★ All the above, plus: Link your data to other people’s data to provide context

http://www.w3.org/DesignIssues/LinkedData.html
ACCESSING LINKED DATA
Accessing Linked Data

- We already know of one way to access Linked Data!

- But what about queries?

http://www.w3.org/2005/ajar/tab.html
SPARQL

“SQL for RDF”
- Where SQL is about relations, SPARQL is about graphs
- Similar syntax: SELECT ... FROM ... WHERE ...

- Supports four query types: SELECT, ASK, DESCRIBE, CONSTRUCT
- SPARQL based on graph patterns that are matched on RDF graphs
Basic building block of SPARQL queries

A BGP is a conjunction of **triple patterns**

```sql
{ ?x rdf:type dbpedia-owl:Country .
}
```

BGPs *matches a* subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph.
Basic Graph Patterns

- Basic building block of SPARQL queries
- A BGP is a conjunction of **triple patterns**

```{?x rdf:type dbpedia-owl:Country.
}
```

- BGPs *matches a* subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph
Basic Graph Patterns

- Basic building block of SPARQL queries
- A BGP is a conjunction of **triple patterns**

```{Variable
} Constant
```

- BGPs *matches* a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph
Basic Graph Patterns

- Basic building block of SPARQL queries
- A BGP is a conjunction of triple patterns

```
{ ?x rdf:type dbpedia-owl:Country .
}
```

- BGPs *matches a* subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph
Basic Graph Patterns

- Basic building block of SPARQL queries
- A BGP is a conjunction of **triple patterns**

{  
}

- BGPs _matches a_ subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is an RDF graph equivalent to the subgraph
SELECT

SELECT queries have the form

```javascript
SELECT <variables>
WHERE {
  <BGP>
}
```

Result: **table of all bindings for selected variables**
SELECT example

dbpedia:India rdf:type dbpedia-owl:Country .
dbpedia:India rdfs:label "India"@en .
dbpedia:Germany rdf:type dbpedia-owl:Country .
dbpedia:Germany rdfs:label "Germany"@en .
dbpedia:France foaf:name "France"@en .

SELECT ?x ?y WHERE {
}

<table>
<thead>
<tr>
<th>?x</th>
<th>?y</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:India</td>
<td>“India”@en</td>
</tr>
<tr>
<td>dbpedia:Germany</td>
<td>“Germany”@en</td>
</tr>
</tbody>
</table>
CONSTRUCT

- CONSTRUCT queries make a new graph based on
  - the whole set of matches and the variable bindings in each match
  - a second graph pattern which shares some of the variables

CONSTRUCT { <Template BGP> }  
WHERE { 
  <BGP> 
}

Result: set of RDF graphs formed from template

For each query solution, an RDF graph is formed by substituting all variables in the template pattern with bindings in the query solution
CONSTRUCT

CONSTRUCT {
  ?x rdf:type geo:Country .
  ?x foaf:name ?y .
}

WHERE {
}

<table>
<thead>
<tr>
<th>?x</th>
<th>?y</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia:India</td>
<td>“India”@en</td>
</tr>
<tr>
<td>dbpedia:Germany</td>
<td>“Germany”@en</td>
</tr>
</tbody>
</table>

dbpedia:India rdf:type geo:Country .
dbpedia:India foaf:name “India”@en .
dbpedia:Germany rdf:type geo:Country .
dbpedia:Germany foaf:name “Germany”@en .
More SPARQL

- SPARQL supports way more than just basic graph patterns
  - OPTIONAL patterns
  - FILTER conditions
  - ORDER BY
  - LIMIT, OFFSET

```
SELECT ?x ?y WHERE {
  OPTIONAL {
    ?x owl:sameAs ?z .
  }
  FILTER (lang(?y) = "en")
}
```

- “SPARQL by example”
  http://www.cambridgesemantics.com/2008/09/sparql-by-example/

- SPARQL 1.1
  http://www.w3.org/TR/sparql11-query/
Querying Linked Data: Architectures

Warehousing/Crawl-Index-Serve

1. Query  
2. Answer  
0. Crawl-Index

Virtual Integration/Distributed Querying

1. Query  
2. Answer
Warehousing/Materialization

- Download datasets and store in a local database
  - Some datasets provide RDF dumps
  - Otherwise, **crawl** Linked Data by following links (=web crawlers)

- Triple stores
  - Purpose-built databases for storage and retrieval of RDF
  - Open source and commercial systems available

- Pro
  - Control, does not depend on 3rd parties
  - Performance

- Contra
  - Operating cost (hardware, software, maintenance, ...)
  - Updates
Virtual Integration/Distributed Querying

SPARQL Endpoint Federation

Linked Data Query Processing

Mediator

Query Engine

HTTP GET

HTTP GET
65% of datasets offer SPARQL endpoints (for free!)

However: sometimes unreliable, query limits, ...
SPARQL Endpoint Federation

- Virtual integration of (remote) data sources
- Mediator
  - Splits queries and sends parts to SPARQL endpoints that contain matching data
  - Processes partial results to obtain query answers

```sparql
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia:President .
  ?nytPresident owl:sameAs ?President .
}
```
SPARQL Endpoint Federation


- SPARQL 1.1 Federation Extensions http://www.w3.org/TR/2010/WD-sparql11-federated-query-20100601/
- Sesame 2.6 http://www.openrdf.org
- DARQ http://darq.sourceforge.net/
- SPLENDID [Görlitz et al. 2011]
- ANAPSID [Acosta et al. 2011]
Linked Data Query Processing

- Works directly over Linked Data, no SPARQL endpoints necessary
- Linked Data sources are retrieved via HTTP during query processing
- Query processing is done locally
- Source discovery and selection is crucial

**Pro**
- Always up-to-date
- No reliance on SPARQL endpoint, any published Linked Data can be queried

**Contra**
- Fairly slow as sources do not support indexes
- High number of possible sources
- Source discovery and selection complex
Linked Data Query Processing

Source discovery
- Run-time: discover sources by following links [Hartig et al. 2009]
- Source index: pre-computed index of sources [Harth et al. 2010]
- Mixed: use both strategies [Ladwig et al. 2010]

Source selection/ranking
- There are (too) many sources, only process most “relevant”
- Sources ranked by their expected results [Harth et al. 2010]
- Refine ranking at run-time [Ladwig et al. 2010]

Query processing
- Network access might lead to blocking, stalling query execution
- Non-Blocking Iterator [Hartig et al. 2009]
- Stream-based query processing [Ladwig et al. 2010]
Accessing Linked Data Summary

- SPARQL is the query language of choice for RDF
  - Graph pattern matching
  - Query types: SELECT, CONSTRUCT, ASK, DESCRIBE
  - Triple patterns, basic graph patterns
  - Solution modifiers: ORDER BY, FILTER, OPTIONAL, ...
- Architectures: Warehousing vs. Virtual Integration
- Warehousing
  - RDF dumps or crawl Linked Data and store locally
- Virtual Integration
  - SPARQL endpoint federation
  - Linked Data Query Processing

What the best technique is depends on the problem!
TOOLS
Tools

- Talis Platform, SaaS, Cloud-based storage for RDF data and binary objects, SPARQL access, REST APIs [http://www.talis.com/platform](http://www.talis.com/platform)
- ARC (PHP) [http://arc.semsol.org/](http://arc.semsol.org/)
- Sesame (Java) [http://www.openrdf.org](http://www.openrdf.org)
- Jena (Java) [http://jena.sourceforge.net/](http://jena.sourceforge.net/)
- NxParser (Java) [http://sw.deri.org/2006/08/nxparser/](http://sw.deri.org/2006/08/nxparser/)
- Spark visualisation library [http://km.aifb.kit.edu/sites/spark/](http://km.aifb.kit.edu/sites/spark/)
Tools

- **Frameworks**
  - Paget [http://code.google.com/p/paget](http://code.google.com/p/paget)

- **Other Tools (debug, etc.)**
  - curl
  - Live HTTP headers (FF plug in)
Tools and Applications

- Further resources regarding the publishing process
  - http://linkeddata.org/docs/how-to-publish
  - http://events.linkeddata.org/iswc2008tutorial/
  - http://videolectures.net/iswc08_heath_hpdlw/
  - http://www.w3.org/TR/swbp-vocab-pub/
  - http://vapour.sourceforge.net/
  - http://swse.deri.org/RDFAlerts/
Summary

- The Linked Data Web is a large, decentralized and complex system built on simple principles
  - identify resource via HTTP URIs
  - provide RDF that links to other URIs upon lookup
- Current trend around Linked Data allows for a re-think of components in Semantic Web Layer Cake
- Data publishers and consumers coordinate little
- Web of Data grows rapidly and covers a large variety of domains
- Algorithms operating over a common access protocol and data model
- First commercial applications emerging

Linked Data: Evolving the Web into a Global Data Space
http://linkeddatabook.com/editions/1.0/
References I


References II


Attribution

- Slides from previous tutorials: WWW 2010 SILD tutorial, ISWC 2011 SILD tutorial, ASWS 2011 summer school and INFORMATIK 2011 tutorial
  - Andreas Harth, Denny Vrandecic, Elena Simperl, Aidan Hogan, Daniel Herzig, Barry Norton
- SWT-2 lectures of Andreas Harth at KIT
- Images of Berlin, Hegel and Dietrich via Wikipedia