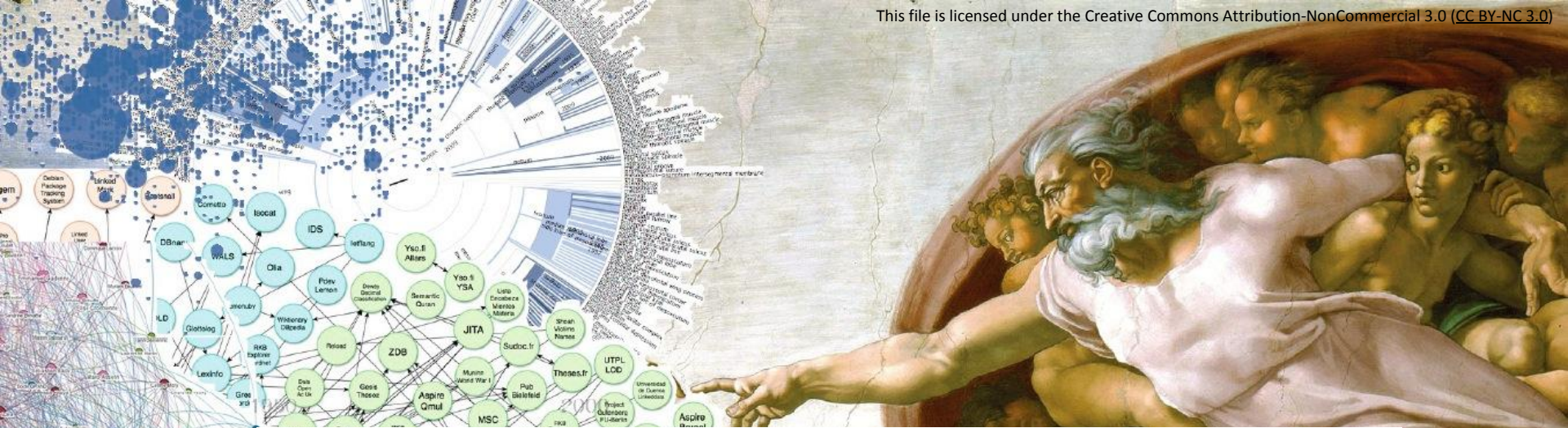




Information Service Engineering

Lecture 4 - Linked Data Engineering II

4.4 How to query RDF with SPARQL

 **FIZ Karlsruhe**

Leibniz Institute for Information Infrastructure

 **Hasso Plattner Institut**

 **KIT**
Karlsruher Institut für Technologie

Prof. Dr. Harald Sack & Dr. Maria Koutraki
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Spring 2018

Information Service Engineering

Lecture 4: Linked Data Engineering II

4.1 Linked Data Principles

4.2 Web Ontology Language OWL

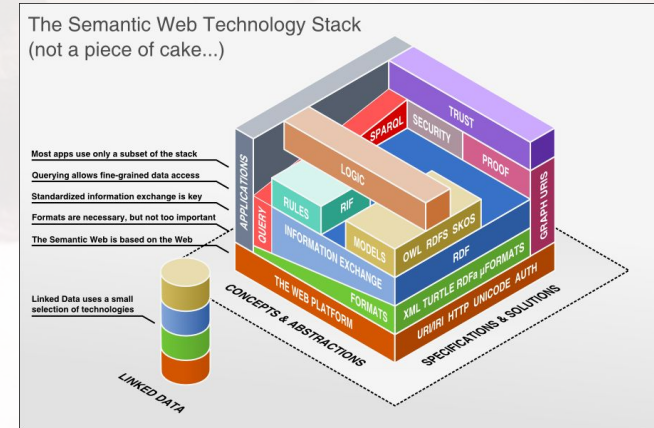
4.3 Web Ontology Language OWL II

4.4 How to Query RDF with SPARQL

4.5 Complex Queries with SPARQL

4.6 More Complex SPARQL Queries

4.7 Linked Data Knowledge Graphs



The Semantic Web Technology Stack (not a piece of cake...)

Most apps use only a subset of the stack

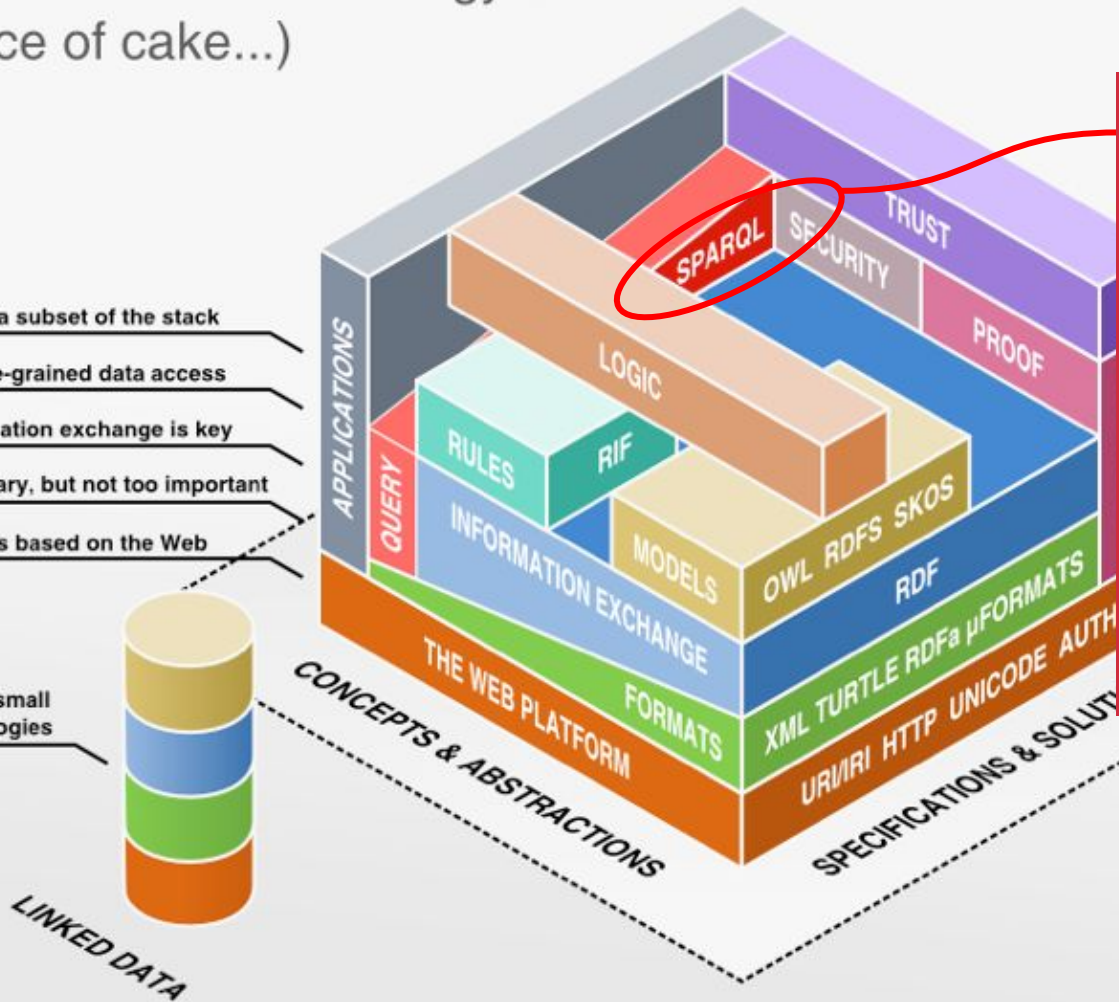
Querying allows fine-grained data access

Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

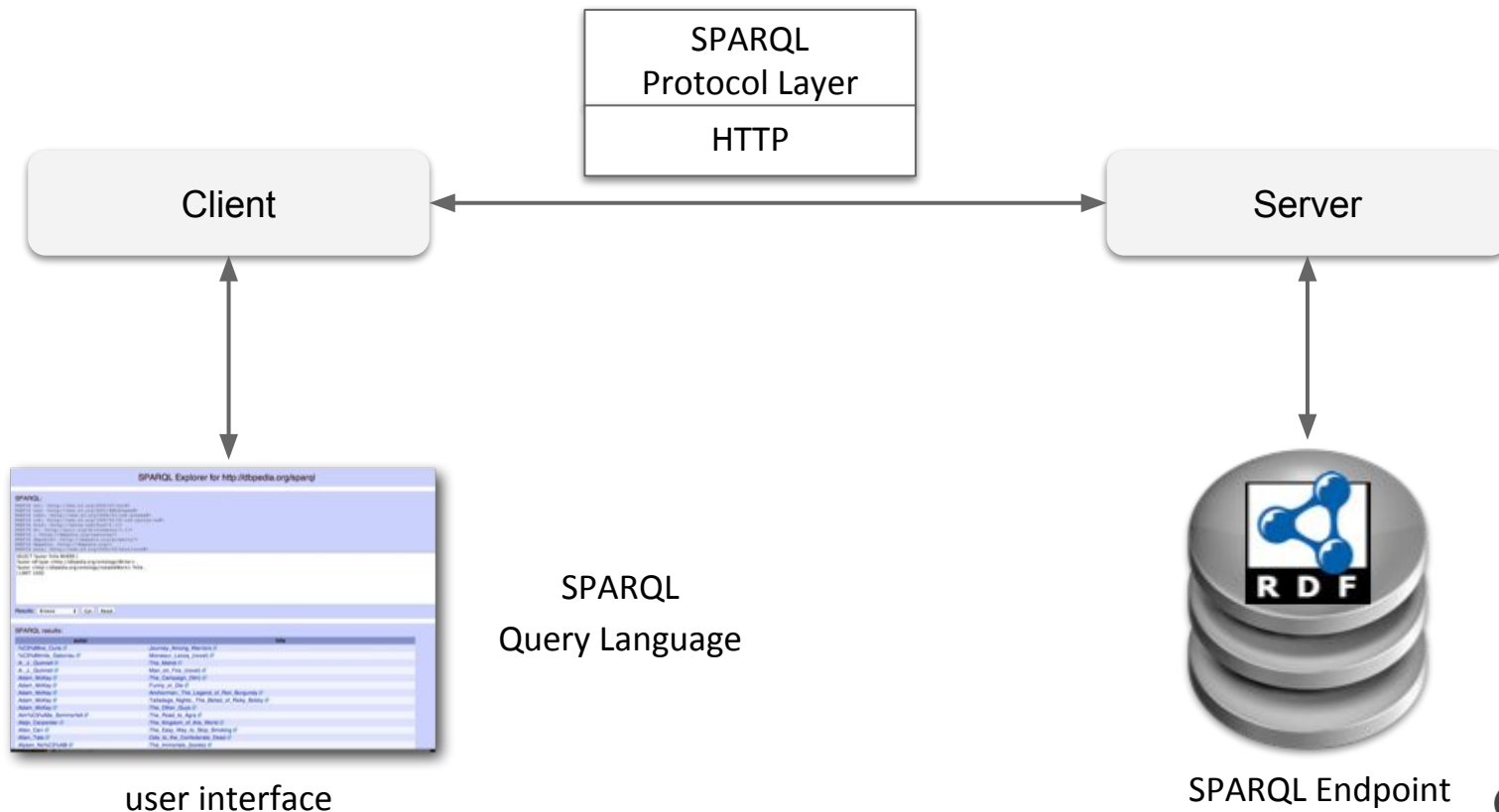
Linked Data uses a small selection of technologies



USE
SPARQL
AND
QUERY
ON

SPARQL - A Query Language for RDF(S)

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL



SPARQL Endpoint Example

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

<http://dbpedia.org/sparql>

← → ↻ dbpedia.org/sparql 🔍 ☆ 👤 📺 ⋮

Apps 24 Google Kalender Pin It Projekte In Refinder merken! postBookmark myBibSonomy postPublication Page Metadata MEDIAGLOBE SocialBro - Browse... » Andere Lesezeichen

Virtuoso SPARQL Query Editor

[About](#) | [Namespace Prefixes](#) | [Inference rules](#) | [ISPARQL](#)

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[ ] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: ☒ Strict checking of void variables ☐ Log debug info at the end of output (has no effect on some queries and output formats)

(The result can only be sent back to browser, not saved on the server, see [details](#))

Copyright © 2017 [OpenLink Software](#)
Virtuoso version 07.20.3218 on Linux (686-generic-linux-glibc212-64), Single Server Edition

SPARQL - A Query Language for RDF(S)

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- **SPARQL Protocol and RDF Query Language** is
 - a **Query Language** for RDF graph traversal
(*SPARQL Query Language Specification*)
 - a **Protocol Layer**, to use SPARQL via http
(*SPARQL Protocol for RDF Specification*)
 - an **XML Output Format Specification** for SPARQL queries
(*SPARQL Query XML Results Format*)
 - W3C Standard (SPARQL 1.1, Mar 2013)
 - inspired by SQL

For Queries we need Variables

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- SPARQL **Variables** are bound to RDF terms
 - e.g. **?title, ?author, ?address**
- In the same way as in SQL,
a **Query for variables** is performed via **SELECT statement**
 - e.g. **SELECT ?title ?author ?published**
- A SELECT statement returns Query Results as a **table**

SPARQL Query

?title	?author	?published
1984	George Orwell	1948
Brave New World	Aldous Huxley	1932
Fahrenheit 451	Ray Bradbury	1953

SPARQL Result

SPARQL Graph Pattern Matching

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- SPARQL is based on **RDF Turtle serialization** and **basic graph pattern matching**.
- A **Graph Pattern (Triple Pattern)** is a RDF Triple that contains variables at any arbitrary place (Subject, Property, Object).

(Graph) Triple Pattern = Turtle + Variables

- Example:
Look for books and their authors:
?book **dbo:author** **?author** .
- A **Basic Graph Pattern (BGP)** is a set of Triple Pattern

SPARQL Graph Pattern Matching

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

Triple Pattern

?book **dbo:author** **?author** .

RDF Graph

dbr:Oliver_Twist rdf:type dbo:Book .

dbr:Oliver_Twist **dbo:author** dbr:Charles_Dickens .

dbr:Oliver_Twist dbo:illustrator dbr:George_Cruikshank .

dbr:I,_Robot rdf:type dbo:Book .

dbr:I,_Robot **dbo:author** dbr:Isaac_Asimov .

dbr:I,_Robot dbo:releaseDate "1950-12-02"^^xsd:date .

...

SPARQL Complex Query Patterns

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- SPARQL Graph Pattern can be combined to form **complex (conjunctive) queries** for RDF graph traversal

- *Find books, their authors, and their release date:*

```
?book dbo:author ?author .  
?book dbo:releaseDate ?date .
```

- *Given a FOAF URI, find the name of a person and her friends:*

```
<http://hpi-web.de/id#haraldsack> foaf:name ?surname ;  
                                   foaf:knows ?friend .  
  
?friend foaf:name ?friend_surname .
```

SPARQL General Query Format

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- *search for all authors and the titles of their notable works:*

specifies namespaces

```
PREFIX :      <http://dbpedia.org/resource/>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo:   <http://dbpedia.org/ontology/>
```

```
SELECT ?author_name ?title
```

specifies output variables

```
FROM <http://dbpedia.org/>
```

specifies graph to be queried

```
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?title .
}
```

*specifies graph pattern
to be matched*

SPARQL General Query Format

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- *search all authors and the titles of their notable works **ordered by** authors in ascending order and **limit** the results to the first 100 results starting the list at **offset** 10 position:*

[query SPARQL endpoint](#)

```
PREFIX :      <http://dbpedia.org/resource/>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo:   <http://dbpedia.org/ontology/>

SELECT ?author_name ?title

FROM <http://dbpedia.org/>

WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?title .
}

ORDER BY ASC (?author_name)
LIMIT 100
OFFSET 10
```

solution sequence
modifiers

SPARQL General Query Format

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

[query SPARQL endpoint](#)

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work dbo:numberOfPages ?pages .
    FILTER (?pages > 500)
    ?work rdfs:label ?title .
} LIMIT 100
```

*specifies constraints
for the result*

- **FILTER** expressions contain operators and functions

SPARQL Unary Operators

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

Operator	Type(A)	Result Type
!A	xsd:boolean	xsd:boolean
+A	numeric	numeric
-A	numeric	numeric
BOUND (A)	variable	xsd:boolean
isURI (A)	RDF term	xsd:boolean
isBLANK (A)	RDF term	xsd:boolean
isLITERAL (A)	RDF Term	xsd:boolean
STR (A)	literal/URL	simple literal
LANG (A)	literal	simple literal
DATATYPE (A)	literal	URI

SPARQL Filter Constraints

Information Service Engineering - Lecture 4 Linked Data Engineering II - 4.4 How to query RDF with SPARQL

- Example: Filter results **only for English labels**

[query SPARQL endpoint](#)

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    FILTER (LANG(?author_name)="en")
    ?author dbo:notableWork ?work .
    ?work dbo:numberOfPages ?pages .
    FILTER (?pages > 500)
    ?work rdfs:label ?title .
    FILTER (LANG(?title)="en")
} LIMIT 100
```

RHINOCERVS

A

Complex Queries with
SPARQL

Picture References:

- P.3: The Semantic Web Technology Stack, via http://bnode.org/media/2009/07/08/semantic_web_technology_stack.png
- P.17: Albrecht Dürer, The Rhinoceros (1515), https://commons.wikimedia.org/wiki/File:D%C3%BCrer%27s_Rhinoceros,_1515.png