

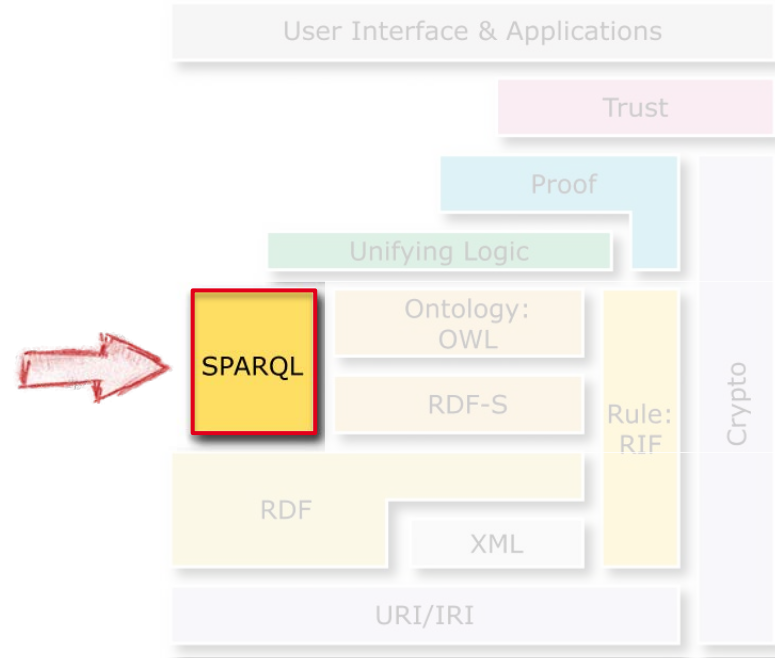
Introduction to a Web of Linked Data

Week 3: SPARQL Query Language

Accessing Data Sources on the Web

Week 3: SPARQL Query Language

Query RDF triple stores published on the Web



Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
3. Filter, constraint and function
4. Pre and post processing
5. Several query forms
6. Results and update

Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
3. Filter, constraint and function
4. Pre and post processing
5. Several query forms
6. Results and update

SPARQL Protocol And RDF Query Language

1. Query Language (Turtle syntax)

- SPARQL 1.1 Query Language - W3C REC 21 Mar. 2013

SPARQL Protocol And RDF Query Language

1. Query Language (Turtle syntax)

- SPARQL 1.1 Query Language - W3C REC 21 Mar. 2013
- SPARQL 1.1 Update - W3C REC 21 Mar. 2013

SPARQL Protocol And RDF Query Language

1. Query Language (Turtle syntax)

- SPARQL 1.1 Query Language - W3C REC 21 Mar. 2013
- SPARQL 1.1 Update - W3C REC 21 Mar. 2013

2. Result format

- SPARQL Query Results XML Format - W3C REC 21 Mar. 2013

Query with SPARQL

SELECT *what you want*
FROM *from where you want*
WHERE { *as you want* }

SPARQL triples

- Turtle syntax with question marks for variables:

```
?x    rdf:type    ex:Person
```

SPARQL triples

- Turtle syntax with question marks for variables:

```
?x    rdf:type    ex:Person
```

- Specify graph pattern to be found:

```
SELECT ?subject ?property ?value  
WHERE { ?subject ?property ?value }
```

SPARQL triples

- Turtle syntax with question marks for variables:

```
?x    rdf:type    ex:Person
```

- Specify graph pattern to be found:

```
SELECT ?subject ?property ?value  
WHERE { ?subject ?property ?value }
```

- A basic graph pattern is a **conjunction** of triples

```
SELECT ?x WHERE  
{ ?x    rdf:type    ex:Person .  
  ?x    ex:name     ?name . }
```

Same Abbreviations as Turtle

- Triples with common subject:

```
SELECT ?name ?fname
WHERE {?x a ex:Person;
       ex:name ?name ;
       ex:firstname ?fname ;
       ex:author ?y . }
```



```
SELECT ?name ?fname
WHERE{?x rdf:type ex:Person.
      ?x ex:name ?name .
      ?x ex:firstname ?fname .
      ?x ex:author ?y . }
```

Same Abbreviations as Turtle

- Triples with common subject:

```
SELECT ?name ?fname
WHERE {?x a ex:Person;
       ex:name ?name ;
       ex:firstname ?fname ;
       ex:author ?y . }
```



```
SELECT ?name ?fname
WHERE{?x rdf:type ex:Person.
      ?x ex:name ?name .
      ?x ex:firstname ?fname .
      ?x ex:author ?y . }
```

- Several values:

```
?x ex:firstname "Fabien", "Lucien" .
```

Same Abbreviations as Turtle

- Triples with common subject:

```
SELECT ?name ?fname  
WHERE { ?x a ex:Person ;  
        ex:name ?name ;  
        ex:firstname ?fname ;  
        ex:author ?y . }
```



```
SELECT ?name ?fname  
WHERE { ?x rdf:type ex:Person .  
        ?x ex:name ?name .  
        ?x ex:firstname ?fname .  
        ?x ex:author ?y . }
```

- Several values:

```
?x ex:firstname "Fabien", "Lucien" .
```

- Blank nodes as anonymous variables:

```
[ ex:firstname "Fabien" ]  
[ ] ex:firstname "Fabien" .
```

Declare Prefixes and Namespaces

- Declare **prefixes** for vocabularies used in the query:

```
PREFIX mit: <http://www.mit.edu#>
SELECT ?student
WHERE {
  ?student mit:registeredAt ?x .
}
```

Declare Prefixes and Namespaces

- Declare **prefixes** for vocabularies used in the query:

```
PREFIX mit: <http://www.mit.edu#>
```

```
SELECT ?student
```

```
WHERE {
```

```
  ?student <http://www.mit.edu#registeredAt> ?x .
```

```
}
```


Declare Prefixes and Namespaces

- Declare **prefixes** for vocabularies used in the query:

```
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student
WHERE {
  ?student mit:registeredAt ?x .
  ?x foaf:homepage <http://www.mit.edu> .
}
```

Declare Prefixes and Namespaces

- Declare **prefixes** for vocabularies used in the query:

```
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student
WHERE {
  ?student mit:registeredAt ?x .
  ?x foaf:homepage <http://www.mit.edu> .
}
```

- Declare base namespace for relative URIs

```
BASE <http://ns.inria.fr/>
```

Specify Language and Datatype of Literals

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x ?f WHERE {
  ?x foaf:name "Fabien"@fr ; foaf:knows ?f .
}
```

Specify Language and Datatype of Literals

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x ?f WHERE {
  ?x foaf:name "Fabien"@fr ; foaf:knows ?f .
}
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x WHERE {
  ?x foaf:name "Fabien"@fr ;
     foaf:age "21"^^xsd:integer .
}
```

Picture credits

- Semantic Web stack of standards, W3C®
- Icons (iconfinder.com)
 - Right icon, Aleksandra Wolska - Free for commercial use

Week 3: SPARQL Query Language

1. RDF graph pattern matching
- 2. Statements**
3. Filter, constraint and function
4. Pre and post processing
5. Several query forms
6. Results and update

Optional Pattern

When part of graph pattern is not mandatory

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE {
  ?person foaf:homepage <http://fabien.info> .
  OPTIONAL { ?person foaf:name ?name . }
}
```

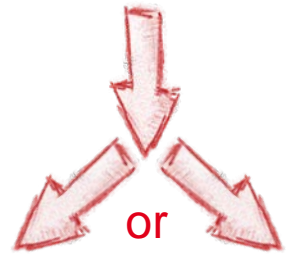


?name variable may be « unbound » in result

Alternative Patterns

Union of results of graph patterns

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE {
  ?person foaf:name ?name .
  {
    ?person foaf:homepage <http://fabien.info> .
  }
  UNION
  {
    ?person foaf:homepage <http://bafien.org> .
  }
}
```



Negation

Remove results that match a pattern

```
PREFIX ex: <http://www.example.abc#>
```

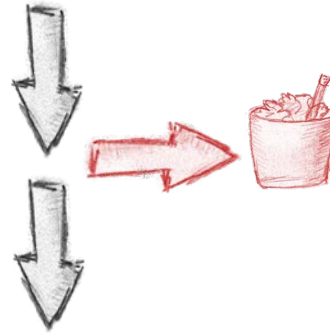
```
SELECT ?x
```

```
WHERE {
```

```
  ?x a ex:Person
```

```
  MINUS { ?x a ex:Man }
```

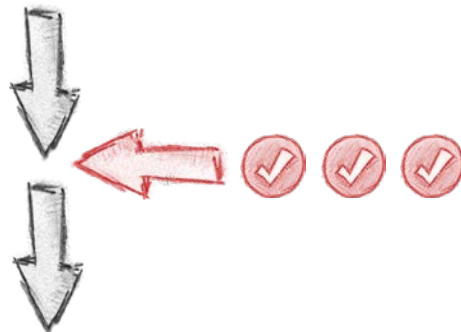
```
}
```



Predefined Variable Values

Results where part of the bindings are predefined

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE {
  ?person foaf:name ?name .
}
VALUES ?name { "Peter" "Pedro" "Pierre" }
```

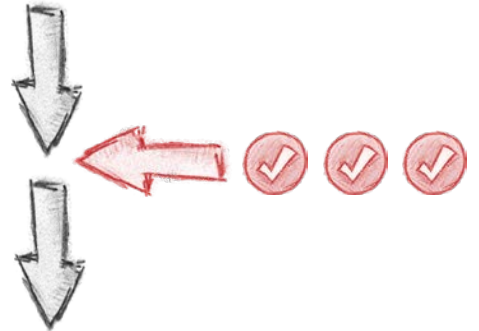


Variable Binding

Results where part of the bindings are computed

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE {
  ?person ex:fname ?fname ;
         ex:lname ?lname .

  BIND (concat(?fname, ?lname) AS ?name)
}
```

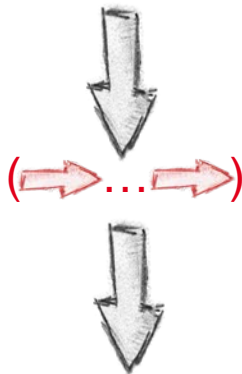


Property path

Regular expressions on property path between resources

/	: sequence		: alternative
+	: one or several	*	: zero or several
?	: optional	^	: reverse
!	: negation		

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?friend WHERE {
  ?x foaf:name "Fabien Gandon" ;
  foaf:knows+ ?friend .
}
```



Keep Distinct Results

Keep one occurrence of similar results with same values for same variables

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT DISTINCT ?name  
WHERE { ?person foaf:name ?name . }
```

Picture credits

- Icons ([iconfinder.com](https://www.iconfinder.com))
 - Right icon, Aleksandra Wolska - Free for commercial use
 - Full, garbage, recycle bin, trash icon, Aleksandra Wolska - Free for commercial use
 - Tick icon, Aleksandra Wolska - Free for commercial use

Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
- 3. Filter, constraint and function**
4. Pre and post processing
5. Several query forms
6. Results and update

Filter Results by Values

Declare constraints on variable values

- `select` = select values to be returned
- `where` = graph pattern
- **`filter`** = constraints in the `where` clause with expressions and functions

E.g. Person At Least 18 Years Old

```
PREFIX ex: <http://inria.fr/schema#>
SELECT ?person ?name
WHERE {
  ?person    rdf:type  ex:Person ;
             ex:name   ?name ;
             ex:age    ?age .
  FILTER (xsd:integer(?age) >= 18)
}
```

Test Values

Test and compare constants, variables and expressions

- Comparators: `<`, `>`, `=`, `<=`, `>=`, `!=`
- Regular expressions: `regex(?x, "A.*")`
- Test variable values: `isURI(?x)`, `isBlank(?x)`,
`isLiteral(?x)`, `bound(?x)`

Strings and Literals

`CONTAINS(lit1, lit2)`, `STRSTARTS(lit1, lit2)`, `STREND(lit1, lit2)`

string inclusion

`STRDT(value, type)`

create literal with datatype

`STRLANG(value, lang)`

create literal with language

`CONCAT(lit1, ..., litn)`

concatenate strings

`SUBSTR(lit, start [, length])`

extract substring

`ENCODE_FOR_URI(str)`

encode string for URI

`UCASE(str)`, `LCASE(str)`

change case

`STRLEN(str)`

string length



Other Functions

YEAR(Date), MONTH(Date), DAY(Date)

HOURS(Date), MINUTES(Date), SECONDS(Date)

NOW()



ABS(Val), CEIL(Val), FLOOR(Val), ROUND(Val)

isNumeric(Val)

RAND()



COALESCE(val₁,..., val_n)

IRI(str), URI(str)

BNODE(ID)

Boolean Connectors

- **And:** &&
- **Or:** ||
- **Not:** !
- **()**



Branching Expression

Usual test: **if... then... else...**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT * where {
  ?x foaf:name ?name ; foaf:age ?age .
  FILTER ( if (langMatches(lang(?name), "FR"),
           ?age >= 18, ?age >= 21) )
}
```



Verify Presence / Absence of a Pattern

exists checks whether a pattern occurs in the graph

not exists checks whether a pattern does not occur in the graph

```
SELECT ?name
WHERE {
  ?x foaf:name ?name .
  FILTER NOT EXISTS { ?x foaf:age -1 }
}
```



Picture credits

- Icons (iconfinder.com)
 - Right icon, Aleksandra Wolska - Free for commercial use
 - Notepad icon, Aleksandra Wolska - Free for commercial use
 - Calendar icon, Aleksandra Wolska - Free for commercial use
 - Alarm, clock, history, time icon, Aleksandra Wolska - Free for commercial use
 - Analytics, chart, pie, statistic icon, Aleksandra Wolska - Free for commercial use
 - Clasp icon, Aleksandra Wolska - Free for commercial use
 - Pause, remove, sketch, stop icon, Aleksandra Wolska - Free for commercial use

Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
3. Filter, constraint and function
- 4. Pre and post processing**
5. Several query forms
6. Results and update

Specify Default Graph

```
PREFIX mit: <http://www.mit.edu#>
SELECT ?student
FROM <http://www.mit.edu/data1.rdf>
FROM <http://www.mit.edu/data2.rdf>
WHERE { ?student mit:registeredAt ?x . }
```



Specify Named Graphs

```
PREFIX mit: <http://www.mit.edu#>
SELECT ?g ?student
FROM NAMED <http://www.mit.edu/data1.rdf>
FROM NAMED <http://www.mit.edu/data2.rdf>
WHERE {
  GRAPH ?g {
    ?student mit:registeredAt ?x .
  }
}
```



Query Remote SPARQL Endpoint

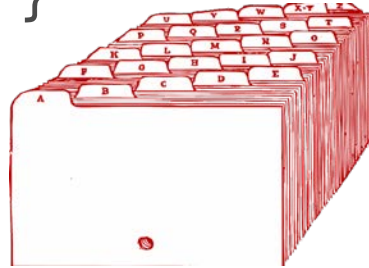


```
SELECT ?x
WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?x rdfs:label "Auguste"@fr .
  }
}
```

Order and Limit Results

E.g.: sort results by *name* from n° 21 to n° 40

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?name  
WHERE { ?x foaf:name ?name . }  
ORDER BY ?name  
LIMIT 20  
OFFSET 20
```



Aggregate Results

Group results by variable(s) values: `group by`

Aggregate values: `count`, `sum`, `min`, `max`, `avg`, `group_concat`, `sample`

Filter aggregated values: `having`

```
PREFIX mit: <http://www.mit.edu#>
SELECT ?student
WHERE { ?student mit:score ?score . }
GROUP BY ?student
HAVING(AVG(?score) >= 10)
```



Nested Queries

Use results of subquery in embedding query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name WHERE {
  { SELECT (max(?age) as ?max)
    WHERE { ?person foaf:age ?age } }
  ?senior foaf:age ?max .
  ?senior foaf:name ?name
}
```



Select Expressions

Extend *select* clause with expressions

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?x (month(?date) as ?month)  
WHERE { ?x foaf:birthday ?date . }
```



Picture credits

- folder-146153_640 by OpenClips – License CC0 Public Domain
<http://pixabay.com/fr/dossier-fichier-cab-bureau-fichier-146153/>
- doll-314345_640 by PublicDomainPictures – License CC0 Public Domain
<http://pixabay.com/fr/poup%C3%A9e-russe-russie-moscou-jeu-314345/>
- Calculation, calculator, numbers icon - shopping website by raja Yuvan
https://www.iconfinder.com/icons/292489/calculation_calculator_numbers_icon#size=128
- Icons (iconfinder.com)
 - Database icon, Aleksandra Wolska - Free for commercial use
 - RSS, signal icon, Aleksandra Wolska - Free for commercial use
 - Document, documents, file, files, report icon, by Raja Yuvan – Free for commercial use
 - Config, configuration, engine, options, preferences, settings icon, Aleksandra Wolska - Free for commercial use

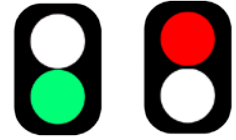
Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
3. Filter, constraint and function
4. Pre and post processing
- 5. Several query forms**
6. Results and update

Check the Existence of a Solution

Do not enumerate all solutions, just answer *true* or *false*

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
ASK { ?person foaf:age 111 . }
```



Construct a Result Graph

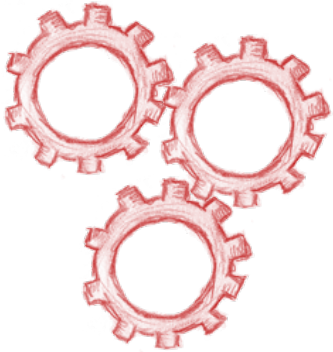
Result of query is a fresh new RDF graph

```
PREFIX mit: <http://www.mit.edu#>
```

```
PREFIX corp: <http://mycorp.com/schema#>
```

```
CONSTRUCT { ?student a corp:FuturExecutive . }
```

```
WHERE      { ?student a mit:Student . }
```



Describe a Resource

Discover unknown data

```
DESCRIBE <http://fabien.info>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
DESCRIBE ?x WHERE { ?x foaf:name "Fabien" }
```



Picture credits

- traffic-light-149581_640 by OpenClips – License CC0 Public Domain
<http://pixabay.com/fr/feu-de-red-stop-feux-de-circulation-149581/>
- gear-472009_640 by blickpixel – License CC0 Public Domain
<http://pixabay.com/fr/gear-engrenages-graphique-472009/>

Week 3: SPARQL Query Language

1. RDF graph pattern matching
2. Statements
3. Filter, constraint and function
4. Pre and post processing
5. Several query forms
- 6. Results and update**

SPARQL Query Result

- Select, Ask: XML Results format
- Construct, Describe: RDF/XML
- JSON

XML SPARQL Query Results Format

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>   <variable name="student"/> </head>
  <results>
    <result>
      <binding name="student">
        <uri>http://www.mit.edu/data.rdf#ndieng</uri>
      </binding>
    </result>
    <result>
      <binding name="student">
        <uri>http://www.mit.edu/data.rdf#jdoe</uri>
      </binding>
    </result>
  </results>
</sparql>
```

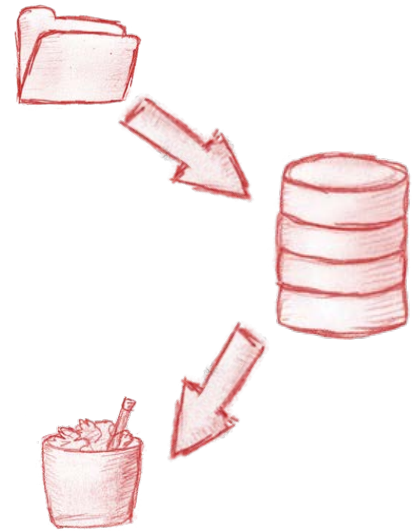


SPARQL Update

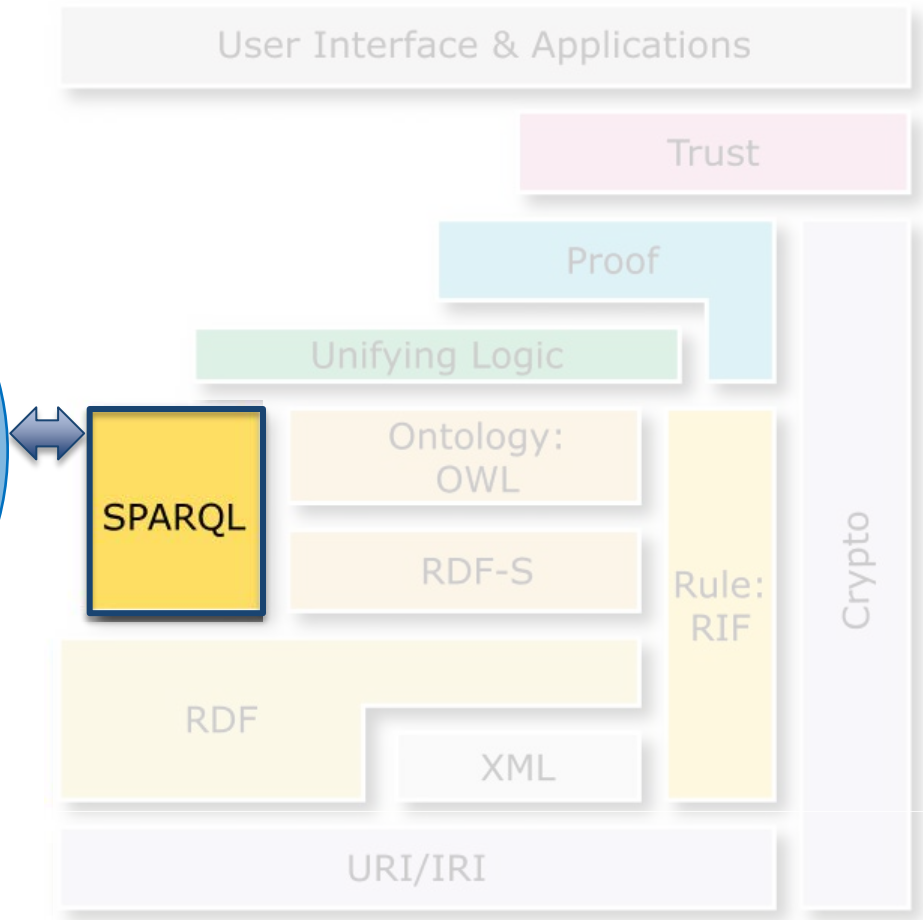
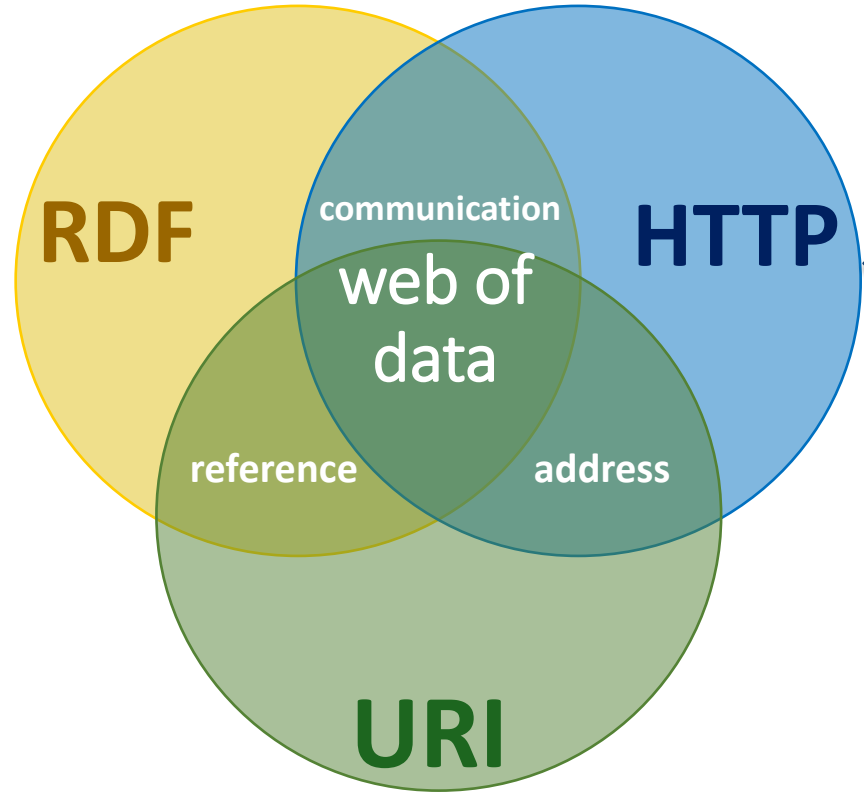
Manage the content of triple store

- Load
- Delete
- Insert
- Copy
- Move
- Add

...



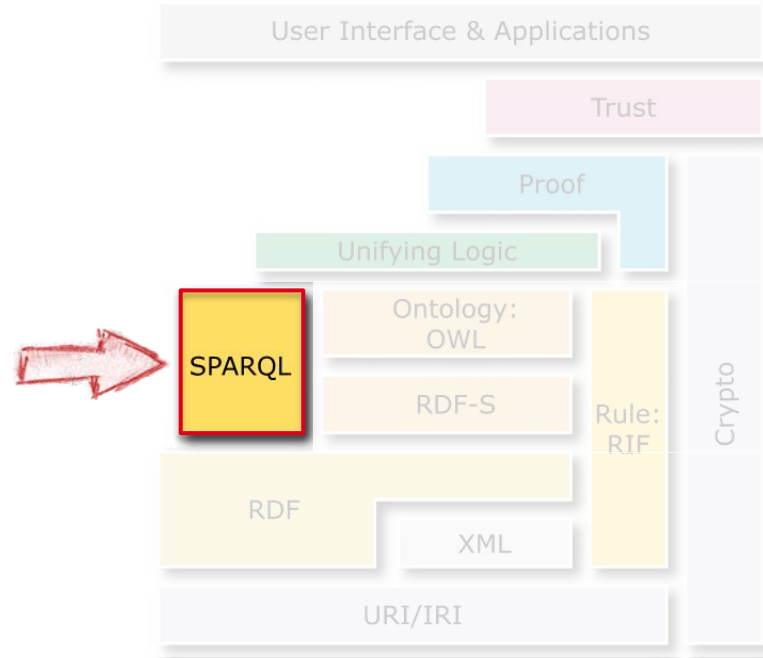
Stack of standards



W3C® Web of data stack of standards

Week 3: SPARQL Query Language

Query RDF triple stores published on the Web



Picture credits

- Semantic Web stack of standards, W3C®
- Icons (iconfinder.com)
 - Paper icon, Web Design Creatives – GPL Licence
 - Folder icon, Aleksandra Wolska - Free for commercial use
 - Right icon, Aleksandra Wolska - Free for commercial use
 - Full, garbage, recycle bin, trash icon, Aleksandra Wolska - Free for commercial use
 - Database icon, Aleksandra Wolska - Free for commercial use