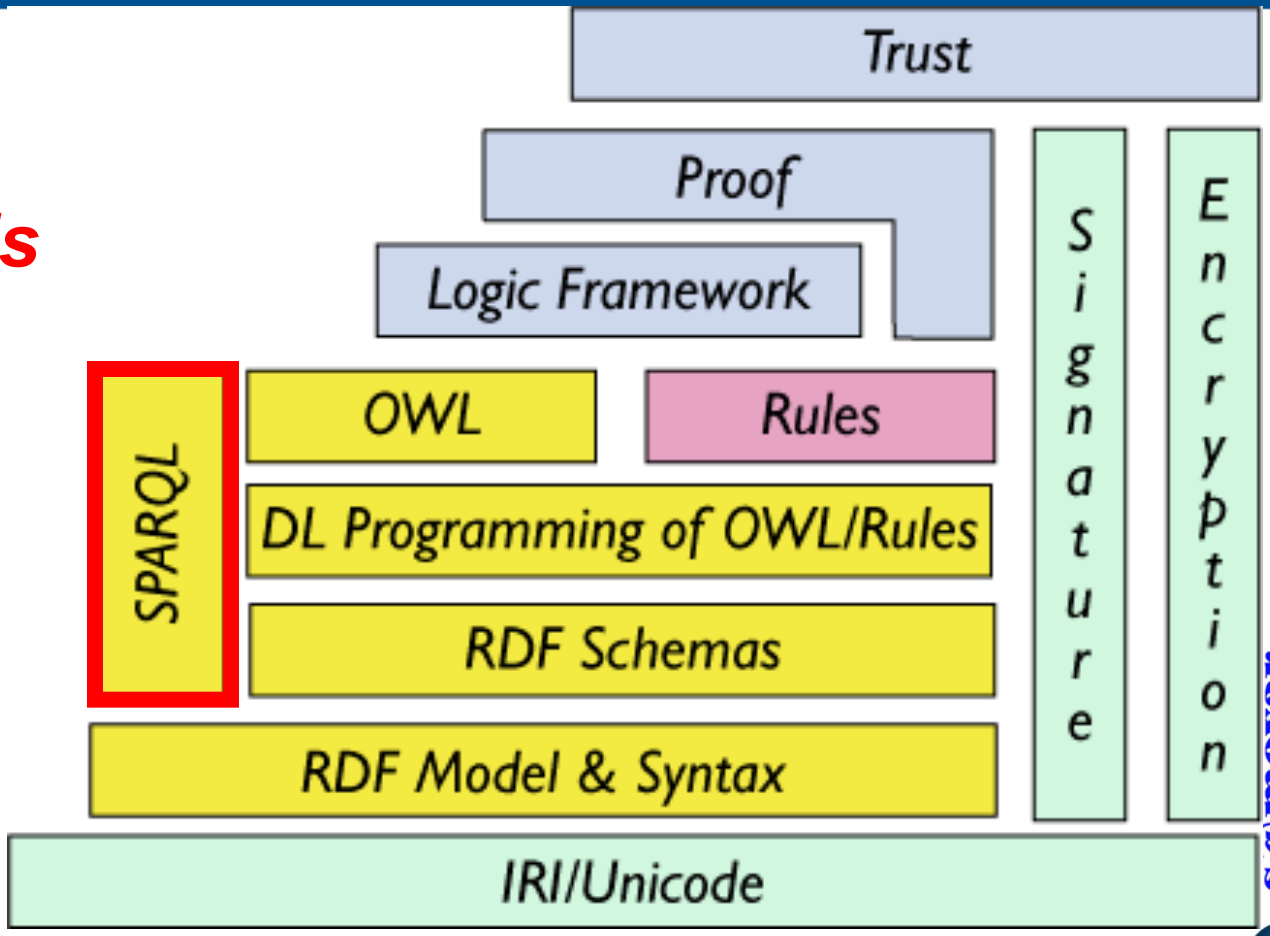
TELECOM SudParis

Introduction to SPARQL

Semantic Web Layer Cake…

Everything is triple



W3C, T Berners-Lee, Ivan Herman

A query language for triples

- **Query language on RDF data**
  - **Graph matching / projection**
  - **The three-clause principle imitating SQL :**
    - Select: clause selecting the values to be returned
    - Where: a pattern of triples to match
    - Filter: constraints

A query language for triples

- « Select » equivalent to « SQL Select » returs a table

  `Select` …
  `From` … Identifies the data sources to be queried

- `Where { … }` the triple/graph pattern
  to match with RDF
  triples/graphes
  a conjunction of triples

- PREFIX to declare the schema used in the query

A query language for triples

- **Example of the simplified triple syntax**

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

- **Language of patterns to be matched**

```
select ?subject ? property ?value
where {?subject ? property ?value }
```

- **The pattern is by default a conjunction of triples**

```
{ ?x rdf:type     ex:Personne .
  ?x  ex:nom      ?nom }
```

- **Two possible forms for the presentation of results:**
  - **the binding i.e. the list of values selected for each response;**
  - **sub-graphs of responses in RDF**

Simple query and namespace

- **Full names of authors :**

```
SELECT ?nom ?prenom
WHERE {
 ?x nom ?nom .
 ?x prenom ?prenom .
 ?x auteur ?y .
}
```

- **To use namespaces :**

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
SELECT ?nom
WHERE {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:option3A tsp:ASR.

 ?etudiant tsp:nom ?nom
}
```

Abbreviated syntax

- Triples with a common root can be simplified as well as the typing relationship:

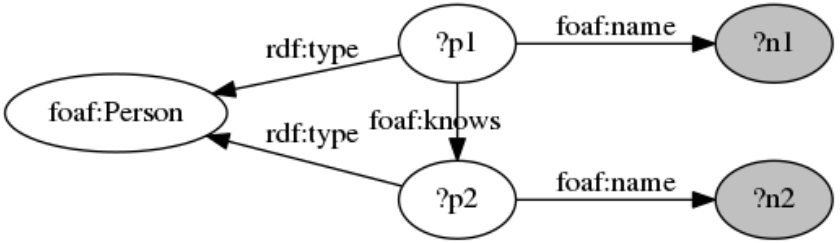
```
SELECT ?nom ?prenom
WHERE {
 ?x a Person;
    nom ?nom ;
    prenom ?prenom ;
    auteur ?y . }
```

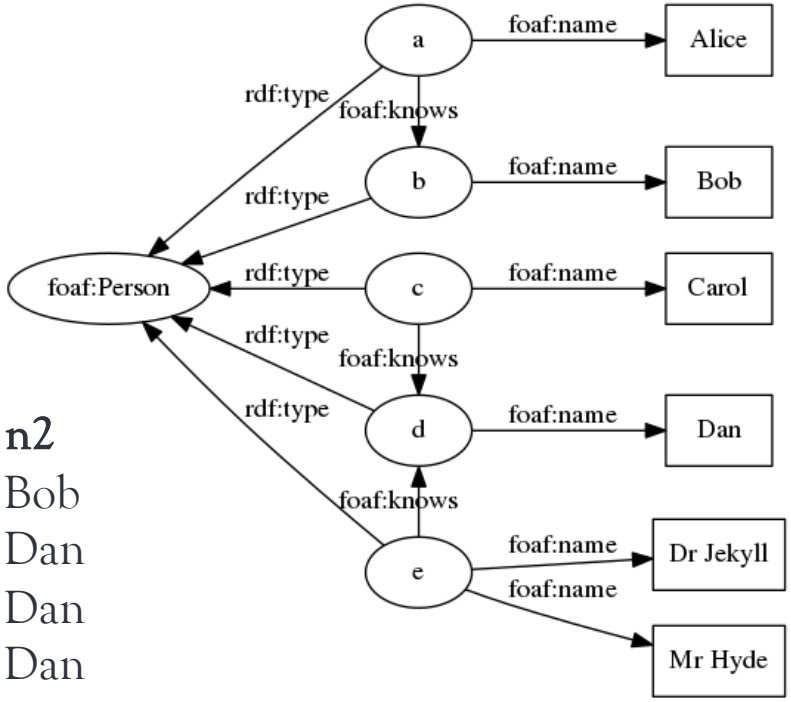
- Several values

```
?x prenom "Tim", "Ivan" .
```

Simple Example

```
PREFIX foaf: http://xmlns.com/foaf/0.1/
SELECT ?n1 ?n2
WHERE {
?p1 a <http://xmlns.com/foaf/0.1/Person> ;
foaf:name ?n1 ;
foaf:knows ?p2 .
?p2 a foaf:Person ;
 foaf:name ?n2 .}
```



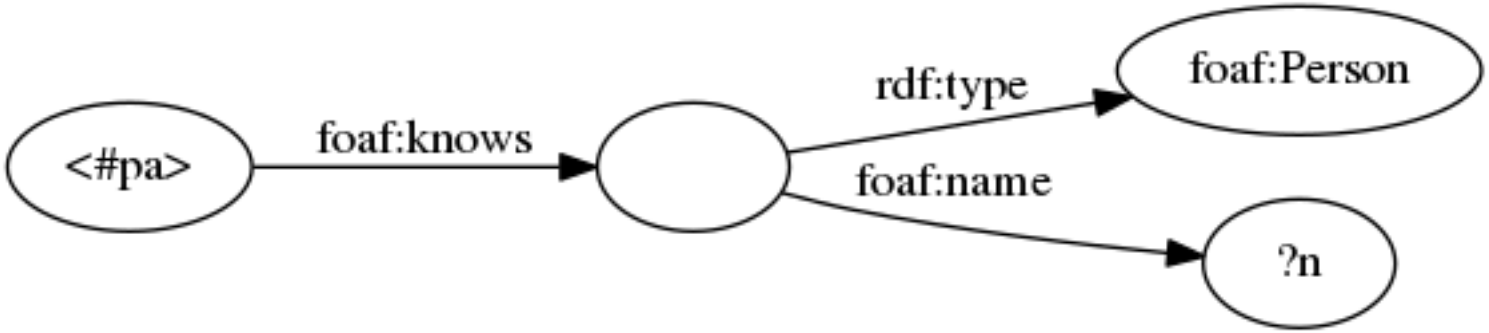


Results

n1	n2
Alice	Bob
Carol	Dan
Dr Jekyll	Dan
Mr Hyde	Dan

Blank nodes

```
<#pa> foaf:knows _:someone.
      _:someone a foaf:Person ;
        foaf:name ?n .
```



Several notations: "_:abc", or "[]".

[foaf:name ?name ; foaf:mbox <mailto:alice@example.org>] is equivalent to :
_:b18 foaf:name ?name .
_:b18 foaf:mbox <mailto:alice@example.org> .

Optional Pattern

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
SELECT ?etudiant ?age
WHERE {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:siteweb <http://www.telecom-sudparis.eu>.
 OPTIONAL {?etudiant tsp:age ?age }
 }
```

Union

- to give alternative patterns :

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
SELECT ?etudiant
WHERE {
 ?etudiant tsp:inscrit ?x .
 {
  {
   ?x tsp:siteweb <http://telecom-sudparis.eu>
  }
  UNION
  {
   ?x tsp:siteweb <http://https://www.imt-bs.eu/>
  }
 }
}
```

Sort, filter and limit answers

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
SELECT ?etudiant ?nom
WHERE {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:siteweb <http://www.telecom-sudparis.eu#> .
 ?etudiant tsp:nom ?nom .
 ?etudiant tsp:age ?age .
 FILTER (?age > 22)
}
ORDER BY ?nom
LIMIT 20
OFFSET 20
```

- Students over 22 sorted by name, answers from #21 à #40

Filter operators

- In the FILTER clause:
  - Comparators : <, >, =, <=, >=, !=
  - Operators: + * - /
  - Tests on variable binding :
    isURI(?x), isBlank(?x), isLiteral(?x), bound(?x)
  - Regular expression filters regex(?x, "A.*")
  - Casting functions xsd:integer(?x)
  - Logic operators &&, ||, !
- In the WHERE clause: @fr , ^^xsd:integer
  - Examples of literals
    - 'chat' @fr
    - 15 => "15"^^xsd:integer
    - 2,5 => "2,5"^^xsd:decimal
    - true => "true"^^xsd:boolean
- In the SELECT clause: distinct

Other clauses

- « Select » returns all or part of the variables linked in a query pattern match
- « Construct » returns an RDF graph
- « Ask » returns a boolean indicating whether a query pattern matches
- « Describe » returns an RDF graph that describes the matched resources

Ask if there are any answers

```
PREFIX tsp: <http://www.telecom-
sudparis.eu#>
ASK {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:siteweb <http://www.telecom-
sudparis.eu> .
 ?etudiant tsp:age ?age .
 FILTER (?age > 30)
}
```

Construct or describe a result

- An output format can be created from scratch :

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
CONSTRUCT
{ ?etudiant rdf:type tsp:FuturIngenieur . }
WHERE {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:siteweb <http://www.telecom-sudparis.eu> .
}
```

- A general description may be requested :

```
PREFIX tsp: <http://www.telecom-sudparis.eu#>
DESCRIBE ?etudiant
WHERE {
 ?etudiant tsp:inscrit ?x .
 ?x tsp:siteweb <http://www.telecom-sudparis.eu> .
}
```

Sparql endpoints

- Dbpedia : http://dbpedia.org/sparql
- Data.gov : http://services.data.gov/sparql
- Data.gov.uk : http://services.data.gov.uk/sparql
- Music Brainz : http://dbtune.org/musicbrainz/
- Linked Geo Data : http://linkedgeodata.org/sparql/
- GeoLinkedData : http://geo.linkeddata.es/web/guest/endpoints
- BBC programmes : http://dbtune.org/bbc/programmes/test/
- Données bibliographiques de la British Library : http://bnb.bibliographica.org/sparql
- Open University : http://data.open.ac.uk/query
- …

- Uberblic : http://api.talis.com/stores/uberblic/services/sparql
- Linked Data semantic repository : http://www.ontotext.com/ldsr/
- LOD cache : http://lod.openlinksw.com/sparql
- Linked Open Commerce : http://linkedopencommerce.com/sparql

SPARQL: *SPARQL Protocol and RDF Query Language*

Ressources

- http://en.wikipedia.org/wiki/SPARQL
- http://www.w3.org/TR/rdf-sparql-query/
- http://jena.sourceforge.net/ARQ/Tutorial/
- http://esw.w3.org/topic/SparqlImplementations
- http://arc.semsol.org/home
- http://virtuoso.openlinksw.com/wiki/main/Main/