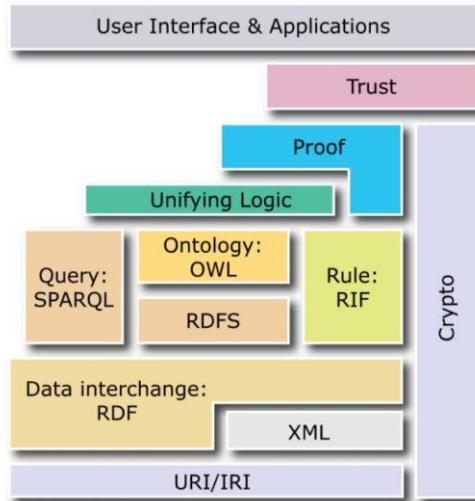# Web Technologies and Data Storage 2

⚠️**Semantic Web [语义万维网]**

An extension of the current web in which information is given well defined meaning, better enabling computers and people to work in cooperation.


Why we need it?

•Tasks often require combining data across the Internet

•Humans understand how to combine this information
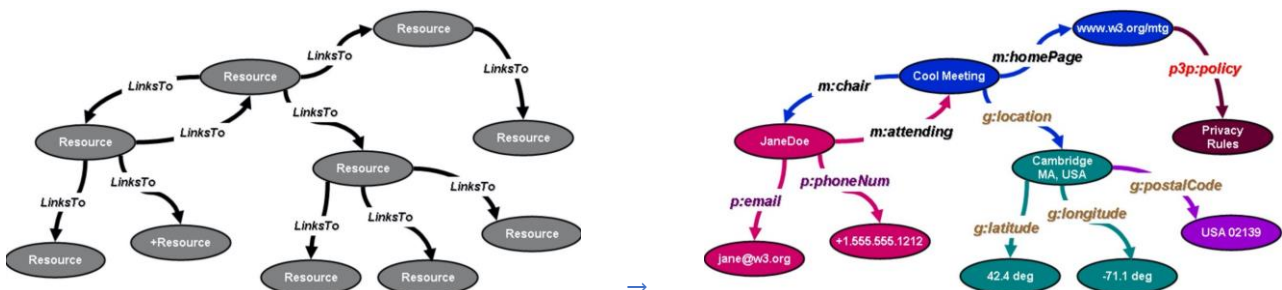
⚠️**Resource description framework**

RDF is a framework for representing information on the Web.

Facilitate data merging even if the underlying schemas differ.

The core structure of the abstract syntax is a set of triples, each consisting of a subject, a predicate and an object.
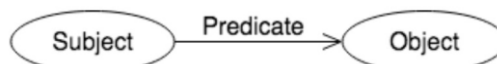
A set of such triples is called an RDF graph. Each triple is represented as a node-arc-node link.

RDFS: Resource Description Framework Schema

  •Provides basic capabilities for describing RDF vocabularies

OWL: Web Ontology Language

  •Built on top of RDFS and RDF which is a computational logic-based language

  •Provides additional capabilities in knowledge representation

•OWL is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things

<RDFS and OWL documents, known as ontologies>

🔺 Ontology/knowledge base (graph)

•An ontology is an explicit specification of a conceptualization [概念化].

•While a conceptual schema defines relations on data, an ontology defines terms to represent knowledge.

    Data: ground atomic facts [表层最小事件]

    Knowledge: expressible in logical sentences with existentially and universally quantified variables

•In the context of Semantic Web, ontology and knowledge base sometimes are used interchangeably (but with some subtle differences).
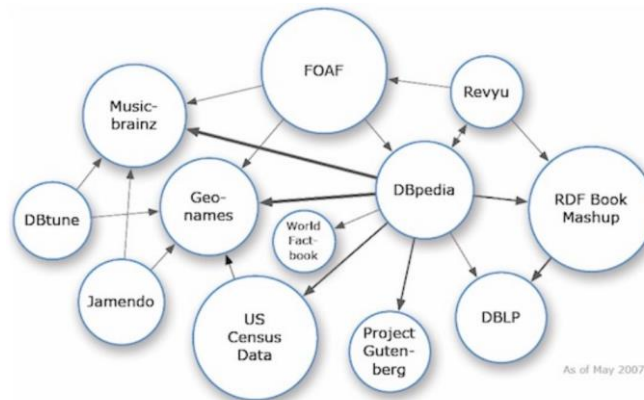
🔺 Linked Open Data

Linked data design principles:

    Use URIs as names for things

    Use HTTP URIs so that people can look up those names.

    When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)

    Include links to other URIs, so that they can discover more things.

The Linked Open Data Cloud (lod-cloud.net)

🔺 Query the Web of Data
🔺 SPARQL

Express queries across diverse data sources whether the data is stored natively as RDF or viewed as RDF via middleware. [在不同数据源之间表示查询，无论数据是作为 RDF 原生存储的还是通过中间件作为 RDF 查看的]

Contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. [包含查询必需的和可选的图形模式及其连接和分离的功能]

Supports extensible value testing and constraining queries by source RDF graph and the results of SPARQL queries can be results sets or RDF graphs. [支持可扩展的值测试和通过源 RDF 图约束查询，SPARQL 查询的结果可以是结果集或 RDF 图]

Queries are based on patterns (triples).

    •RDF can be seen as a set of relationships among resources (i.e. RDF triples);

    •SPARQL queries provide one or more patterns against such relationships.

    •These triple patterns are similar to RDF triples, except that one or more of the constituent resources references are variables.

A SPARQL engine would return the resources for all triples that match these patterns.

Tutorial: Tutorial 5: Querying Semantic Data (linkeddatatools.com)

Website: OpenLink Virtuoso SPARQL Query Editor (dbpedia.org)

[1]

- Retrieve country instances:

```
SELECT ?country
  WHERE
    {
          ?country rdf:type dbo:Country
    } LIMIT 100
```

SELECT distinct ?country WHERE {?country rdf:type dbo:Country} LIMIT 100

[2]

- Retrieve artist instances whose birth place is a country contains the string of "United Kingdom".

```
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT ?artist ?place
  WHERE
    {
        ?artist rdf:type dbpedia-owl:Artist.
        ?artist dbprop:birthPlace ?place.
        ?place rdf:type dbpedia-owl:Country.
        ?place rdfs:label ?label.
        FILTER regex(?label, "^United Kingdom").
    }
```

PREFIX dbprop: <http://dbpedia.org/property/> PREFIX dbpedia-owl: <http://dbpedia.org/ontology/> SELECT ?artist ?place WHERE {?artist rdf:type dbpedia-owl:Artist. ?artist dbprop:birthPlace ?place. ?place rdf:type dbpedia-owl:Country. ?place rdfs:label ?label. FILTER regex(?label, "^United Kingdom"). }