

## Web sémantique

olivier.corby@inria.fr



Web sémantique

2

## Introduction au Web sémantique

1. RDF : Resource Description Framework
2. RDFS : RDF Schema
3. SPARQL : RDF Query Language
4. OWL: Web Ontology Language

Web sémantique

2

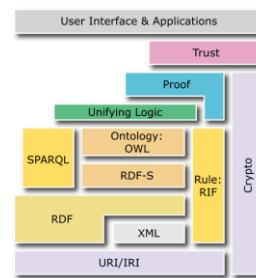
## Document

- <http://files.inria.fr/corese/slide/inrae.pdf>
- <http://files.inria.fr/corese/data/inrae.zip>
- <http://project.inria.fr/corese/download>  
– télécharger corese gui

Web sémantique

3

## W3C Semantic Web



Web sémantique

4

## Vocabulaire

- Triple Store : Base de données graphe RDF
- SPARQL Endpoint : Triple Store avec accès HTTP
- Linked Data : Données RDF liées par les URL
- Knowledge Graph : RDF est une sorte de KG

Web sémantique

5

## RDF

Représenter et échanger des données et des connaissances sur le Web, i.e. par HTTP.

Web sémantique

6

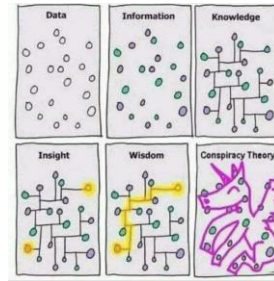
## Donnée vs Connaissance

- Donnée : 10, Nice, 6/4/2021
- Information : il fait 10° à Nice le 6 avril 2021
- Connaissance :
  - Nice est une ville située dans le département des Alpes-Maritimes dans le sud-est de la France
  - s'il fait moins de 15° il fait froid
  - s'il fait froid il faut bien se couvrir
- On parle de « connaissances » quand on peut faire des « déductions » ou « inférences »

Web sémantique

7

## Connaissance



Web sémantique

8

## Exemple : DBpedia

- Base de connaissances RDF
- Extraction automatique depuis Wikipedia
- Ressources nommées et décrites avec des URI

[http://fr.dbpedia.org/resource/Université\\_Nice\\_Sophia\\_Antipolis](http://fr.dbpedia.org/resource/Université_Nice_Sophia_Antipolis)  
 dbpedia-owl:city  
<http://fr.dbpedia.org/resource/Nice>

Web sémantique

9

The screenshot shows the DBpedia page for 'Université Nice Sophia Antipolis'. It includes a search bar, a list of categories and types, and a detailed description of the university in French. The page also features a 'Property: Value' table and a 'chercheurs et membres' section.

Web sémantique

10

## Déréférencer un URL

```
wget --header "Accept: application/rdf+xml"
http://fr.dbpedia.org/resource/Université_Nice_Sophia_Antipolis
```

```
curl -L -H "Accept: application/rdf+xml"
http://fr.dbpedia.org/resource/Université_Nice_Sophia_Antipolis
```

Web sémantique

11

## Résultat Déréférencement URL

```
<rdf:Description
rdf:about="http://fr.dbpedia.org/resource/Université_Nice_Sophia_Antipolis">
<rdf:type rdf:resource="http://schema.org/EducationalOrganization" />
<rdf:type rdf:resource="http://dbpedia.org/ontology/University" />
<rdf:type rdf:resource="http://www.wikidata.org/entity/Q43229" />
<rdf:type rdf:resource="http://dbpedia.org/ontology/EducationalInstitution" />
<rdfs:label xml:lang="en">University of Nice Sophia Antipolis</rdfs:label>
<dbpedia-owl:wikiPageWikiLink
rdf:resource="http://fr.dbpedia.org/resource/Institut_universitaire_de_technologie_de_Nice" />
<dbpedia-owl:city rdf:resource="http://fr.dbpedia.org/resource/Nice" />
<dbpedia-owl:region rdf:resource="http://fr.dbpedia.org/resource/Provence-Alpes-Côte_d'Azur" />
...
</rdf:RDF>
```

Web sémantique

12

## RDF en Turtle

prefix ex: <http://example.org/ns/>

```
ex:James rdf:type ex:Person ;
  ex:name "James" ;
  ex:knows ex:Jim, ex:Jesse ;
  ex:age 23 .
```

```
ex:Jim ex:name "Jim" .
```

Web sémantique

13

## Triplet RDF

resource property value

subject predicate object

Web sémantique

14

## Triplet RDF : relation binaire

James knows Jim

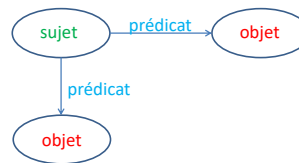
::=

knows(James, Jim)

Web sémantique

15

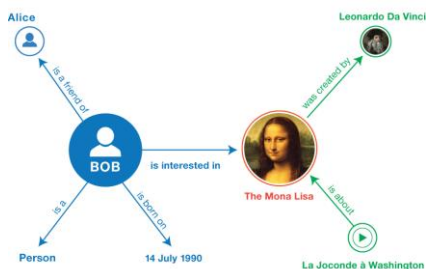
## Graphe RDF



Web sémantique

16

## RDF: Graphe orienté étiqueté



Web sémantique

Crédit: W3C

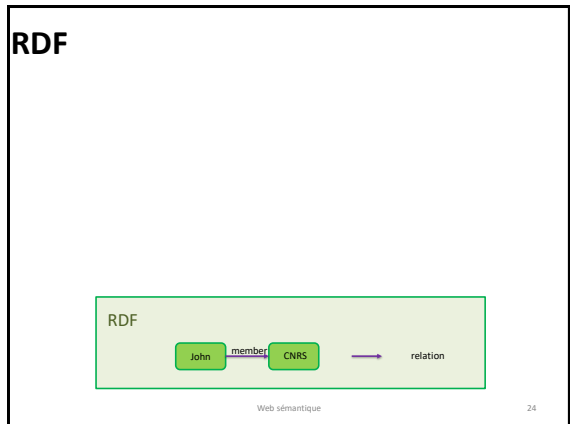
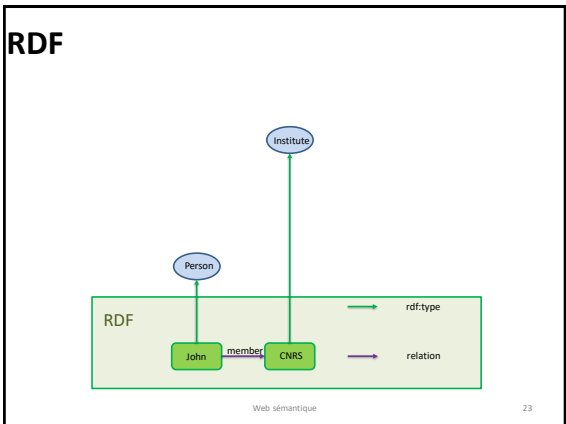
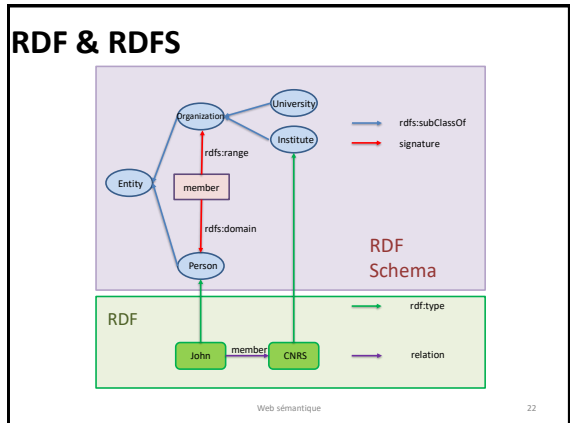
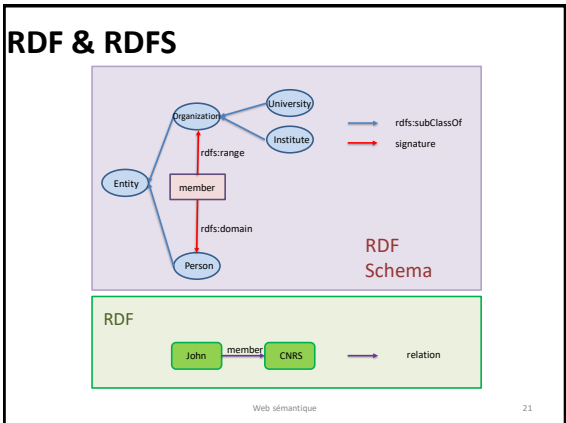
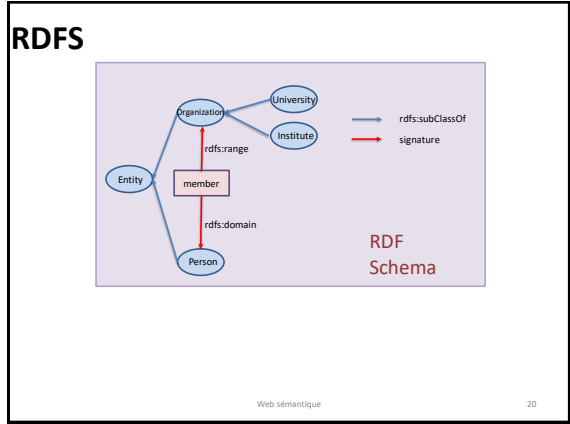
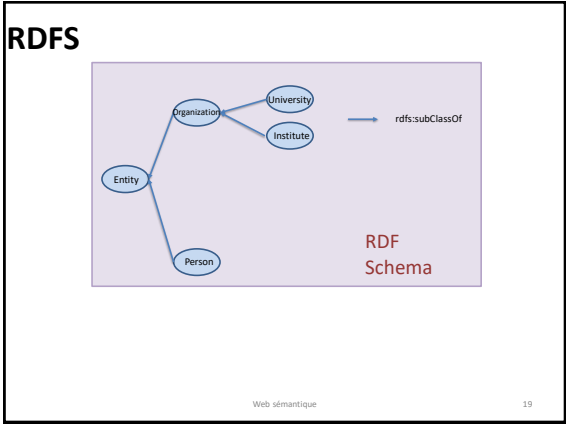
17

## RDF Schema

- Définir le vocabulaire utilisé dans le graphe RDF
- Hiérarchie de classes
- Hiérarchie de propriétés
- Signature (typage) des propriétés

Web sémantique

18



## Multi instantiation – Héritage Multiple

ex:Laurent **rdf:type** ex:Senior, ex:Programmer .

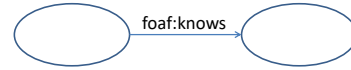
ex:Human **rdfs:subClassOf** ex:Good, ex:Evil .

Web sémantique

25

## Signature de Propriété

foaf:knows a rdf:Property ;  
**rdfs:domain foaf:Person ;**  
**rdfs:range foaf:Person .**

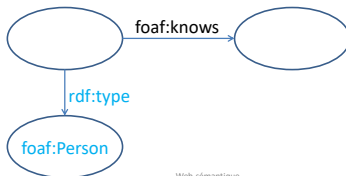


Web sémantique

26

## Inférence de type

foaf:knows a rdf:Property ;  
**rdfs:domain foaf:Person ;**  
**rdfs:range foaf:Person .**

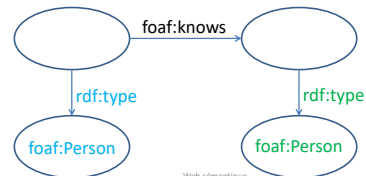


Web sémantique

27

## Inférence de type

foaf:knows a rdf:Property ;  
**rdfs:domain foaf:Person ;**  
**rdfs:range foaf:Person .**



Web sémantique

28

## RDF Dataset – Graphes nommés

- Graphe par défaut (default graph)
- Graphes nommés (named graph)
- { G, (uri<sub>1</sub>, G<sub>1</sub>), .. (uri<sub>n</sub>, G<sub>n</sub>) }

Web sémantique

29

## Graphe par défaut

ex:James a ex:Lecturer ;  
 foaf:name "James" .

ex:James a ex:Musician ;  
 foaf:name "Jimmy" .

Web sémantique

30

## Graphes nommés

```
graph ex:g1 {
  ex:James a ex:Lecturer ;
  foaf:name "James" .
}

graph ex:g2 {
  ex:James a ex:Musician ;
  foaf:name "Jimmy" .
}
```

Web sémantique

31

## Graphes nommés

```
graph ex:g1 {
  ex:James a ex:Lecturer ;
  foaf:name "James" .
}

graph ex:g2 {
  ex:James a ex:Musician ;
  foaf:name "Jimmy" .
}
```

Web sémantique

32

## Typier URI de graphe nommé

```
graph ex:g1 {
  ex:James a ex:Lecturer ;
  foaf:name "James" .
}
```

ex:g1 a ex:Context

Web sémantique

33

## Annoter URI de graphe nommé

```
graph ex:g1 {
  ex:James a ex:Lecturer ;
  foaf:name "James" .
}
```

ex:g1 ex:date "2014-11-28"^^xsd:date ;  
ex:author ex:ocorby

Web sémantique

34

## Annoter URI de graphe nommé

```
graph ex:g1 { ... }
graph ex:g2 { ... }

ex:g1 ex:before ex:g2
```

Web sémantique

35

## Graphes nommés

- Contextualiser les données
- Modéliser la provenance des données
- Annoter les URI des graphes nommés
- Versionning, Annotation temporelle
- Distinguer ontologie et données

Web sémantique

36

## SPARQL

- RDF Query Language
- HTTP Protocol

Web sémantique

37

## Select Query

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select *
where {
    ?x rdf:type foaf:Person ;
    foaf:name "John" .
}
```

Web sémantique

38

## Select Query

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select *
where {
    ?x rdf:type foaf:Person ;
    foaf:name "John", ?name .
    ?y foaf:knows ?x
}
```

Web sémantique

39

## Filter

```
select * where {
    ?x foaf:age ?age
    filter (?age >= 18)
}
```

Web sémantique

40

## Filter Language

- URI, Literal, Variable      ex:John, 3.14, ?x
- < <= >= >                ?age >= 18, ?n < "John"
- = !=                         ?x != ?y, ?y = "James"
- ( )
- + - \* /                      (?n \* (?n + 1))/2
- && || !                      !(?x < 0 && ?y < 0)
- Function                    datatype(?x)
- Exists                        exists { ?x foaf:knows ?y }

Web sémantique

41

## Function

- isBlank(?x)
- isURI(?x)
- isLiteral(?x)
- bound(?x)
- if (cond, exp, exp)
- coalesce(exp, exp)

Web sémantique

42

## Function

- datatype(?x)
- lang(?l)
- langMatches(lang(?l), "fr")
- str(<http://example.org>)
- uri("http://example.org")
- xsd:integer("123")
- xsd:string(12)
- strdt(str, datatype)
- strlang(str, lang)

Web sémantique

43

## Query Form

1. select \* where { }
2. ask { ex:Man ex:wise true }
3. construct { } where { }
4. describe URI

Web sémantique

44

## Construct

- Construct and return new RDF graph

```
construct { ?x rdfs:seeAlso ?z }
where { ?x foaf:knows ?y . ?y foaf:knows ?z }
```

1. Compute «where» results
2. Instantiate «construct» variables with result values
3. Merge «construct» triples into new RDF graph

Web sémantique

45

## Statement

1. Union `{?s a ex:Good} union {?s a ex:Evil}`
2. Optional `{?s a ex:Person} optional {?s ex:age ?n}`
3. Minus `{?s a ex:Person} minus {?s ex:child ?c}`
4. Filter Exists `filter exists {?s ex:knows ?x}`
5. Dataset `select from ex:g1 from named ex:g2`
6. Named Graph `graph ?g {?s ex:date ?d}`
7. Property Path `?s rdf:type/rdfs:subClassOf* ex:Person`
8. Nested Query `select where { select where { } }`
9. Bind `bind (uri(?str) as ?uri)`
10. Values `values (?p ?e) {(3.14 2.718)}`
11. Service `service <http://dbpedia.org/sparql> { }`

Web sémantique

46

## Result & Modifier

1. Select Exp `select * (year(?date) as ?y)`
2. Order By `order by desc(?date) ?name`
3. Group By `group by ?name`
4. Aggregate `(count(?x) as ?c)`
5. Having `having (?c > 100)`
6. Limit `limit 100`
7. Offset `offset 100`

Web sémantique

47

## Aggregate

- Combine several results into an aggregate value.

```
select (avg(?p) as ?avg) where {
  ?x ex:price ?p
}
```

1. count
2. sum
3. avg
4. min
5. max
6. group\_concat
7. sample

Web sémantique

48



## Aggregate & Group By

- Group results with similar values, aggregate in each group

```
select ?x (avg(?s) as ?score)
where {
  ?x ex:score ?s
}
group by ?x
```

Web sémantique

49

## SPARQL Update

Manage (modify) content of RDF Dataset

Web sémantique

50

## Load RDF document

```
load <http://example.org/data.ttl>
load <http://example.org/data.ttl> into graph ex:g
```

Web sémantique

51

## Insert Data

```
insert data {
  ex:John a foaf:Person ; foaf:age 18 .
  ex:Jill a foaf:Person ; foaf:age 81 .
}
```

data => there is no variable

Web sémantique

52

## Delete Data

```
delete data {
  ex:John a foaf:Person ; foaf:age 18 .
  ex:Jill a foaf:Person ; foaf:age 81 .
}
```

Web sémantique

53

## Insert Where

```
insert {
  ?x foaf:mail ?mail
}
where {
  ?x ex:firstName ?f ; ex:lastName ?l
  bind (concat(?f, ".", ?l, "@acme.com")
  as ?mail)
}
```

Web sémantique

54

## Delete Where

```
delete {
  ?x foaf:age ?age
}
where {
  ?x a foaf:Person ; foaf:age ?age
  filter (?age < 0)
}
```

Web sémantique

55

## Update: Delete Insert Where

```
delete { ?x foaf:age ?age }
insert { ?x foaf:age ?int }
where {
  ?x foaf:age ?age
  filter (datatype(?age) = xsd:string)
  bind (xsd:integer(?age) as ?int)
}
```

1. Compute solutions  $\{S_1, .. S_n\}$  of « where clause »
2. Apply delete on  $\{S_1, .. S_n\}$
3. Apply insert on  $\{S_1, .. S_n\}$

Web sémantique

56

## SPARQL Protocol

- Interact with SPARQL endpoint by means of HTTP

```
http://dbpedia.org/sparql?query=
select * where { ?x rdfs:label "Paris"@fr }
```

Query Results format is standardized

Web sémantique

57

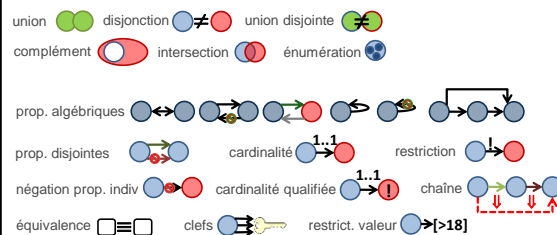
## Web Ontology Language (OWL)

- recommandation W3C
- logique de description
- classes et propriétés définies par des formules logiques
- permet de tirer plus de conclusions, de faire plus d'inférences.

58

## OWL in one...

une vue graphique des constructeurs logiques offerts



59

## Union/Intersection

```
ex:LegalAgent a owl:Class ;
  owl:equivalentClass [
    owl:unionOf
    ( ex:Person ex:Group ) ] .
```

```
ex:Man a owl:Class ;
  owl:equivalentClass [
    owl:intersectionOf
    ( ex:Person ex:Male ) ] .
```

Web sémantique

60

## Union disjointe

```
ex:Passenger a owl:Class ;
  owl:disjointUnionOf
    ( ex:Adult ex:Child ex:Pet ) .
```

Web sémantique

61

## Disjonction/Complement

```
ex:Good a owl:Class ;
  owl:complementOf ex:Evil .

ex:Square a owl:Class ;
  owl:disjointWith ex:Circle .

[] a owl:AllDisjointClasses ;
  owl:members
    (ex:Square ex:Circle ex:Triangle) .
```

Web sémantique

62

## Classe énumérée

définir une classe en énumérant tous ses membres

```
ex:EyeColor a owl:Class ;
  owl:oneOf
    ( ex:Blue ex:Green ex:Brown ) .
```

Web sémantique

63

## Identification par des clés

deux instances qui ont même(s) valeur(s) de clé(s) sont la même instance

```
ex:Person owl:hasKey (ex:hasSSN) .
```

Web sémantique

64

## Equivalence

deux classes représentent les mêmes ensembles d'instances.

```
ex:Human owl:equivalentClass
foaf:Person .
```

Web sémantique

65

## Ressources identiques/différentes

deux URI qui identifient le même individu.

```
ex:Bill owl:sameAs ex:William .
```

deux URI qui identifient des individus différents.

```
ex:Good owl:differentFrom ex:Evil .
```

Web sémantique

66

## Restriction des valeurs d'une propriété

classe pour laquelle toutes les valeurs d'une propriété sont issues d'une même classe

```
ex:Herbivore a owl:Class ;
  rdfs:subClassOf ex:Animal ;
  rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:eat ;
    owl:allValuesFrom ex:Plant ] .
```



**ATTENTION:** s'il n'y a aucune occurrence de la propriété (ex:eat) la restriction est tout de même vraie

67

## Restriction de certaines valeurs d'une propriété

certaines valeurs de la propriété sont instances d'une classe spécifique

```
ex:Sportive a owl:Class ;
  owl:equivalentClass
  [ a owl:Restriction ;
    owl:onProperty ex:hobby ;
    owl:someValuesFrom ex:Sport ] .
```



68

## Restriction à une seule valeur pour une propriété

la classe définie impose une valeur spécifique pour une propriété donnée

```
ex:Bicycle a owl:Class ;
  rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:nbWheel ;
    owl:hasValue 2 ] .
```

69

## Restriction de cardinalité

nombre de valeurs différentes possibles d'une propriété pour le même sujet : minimum, maximum, nombre exact

```
ex:Person a owl:Class ;
  rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:name ;
    owl:maxCardinality 1 ] .
```

70

## Restriction de cardinalité qualifiée

contrainte sur le nombre de fois qu'une propriété peut être utilisée avec des valeurs d'un certain type sur le même sujet : minimum, maximum, nombre exact

```
ex:Human a owl:Class ;
  rdfs:subClassOf
  [ a owl:Restriction ;
    owl:onProperty ex:hasParent ;
    owl:onClass ex:Male ;
    owl:qualifiedCardinality 1 ] .
```

71

## Propriétés (anti)symétriques

$$x R y \Rightarrow y R x$$

```
ex:hasSpouse a
  owl:SymmetricProperty .
```

$$x R y \Rightarrow \neg y R x$$

```
ex:hasChild a owl:AsymmetricProperty .
```

Web sémantique

72

## Propriétés inverses

deux relations qui existent simultanément en sens inverse (ex. parent\_de / enfant\_de)

$$x R_1 y \Leftrightarrow y R_2 x$$

```
ex:hasChild owl:inverseOf
ex:hasParent .
```

Web sémantique

73

## Propriétés transitives

une relation qui se propage de proche en proche (ex. Tom ancêtre Jim ancêtre Jules)

$$x R y \ \& \ y R z \Rightarrow x R z$$

```
ex:hasAncestor a
owl:TransitiveProperty .
```

Web sémantique

74

## Equivalence de propriétés

deux types de propriétés expriment la même relation.

```
ex:name owl:equivalentProperty
my:label .
```

Web sémantique

75

## Propriétés disjointes

Des relations qui ne peuvent pas exister en même temps sur le même sujet et le même objet

```
ex:hasSon owl:propertyDisjointWith
ex:hasDaughter .
```

Web sémantique

76

## Propriétés chaînées

relation définie par un chemin de propriétés  
 $x P/Q z \Rightarrow x R z$

```
ex:uncle owl:propertyChainAxiom
(ex:parent ex:brother) .
```

Web sémantique

77

## Propriétés (inverse) fonctionnelles

une ressource ne peut avoir qu'une valeur possible pour une propriété

$$x R y \ \& \ x R z \Rightarrow y = z$$

```
ex:hasFather a owl:FunctionalProperty
```

une relation pour laquelle une même valeur implique la même ressource (ex. NSS)

$$x R y \ \& \ z R y \Rightarrow x = z$$

```
ex:socialSecurityNumber a
owl:InverseFunctionalProperty .
```

Web sémantique

78

## What Else ?

**SKOS:** Simple Knowledge Organization System

- Thesaurus, Terminology, NLP

**SHACL:** Shapes Constraint Language

- Schema for RDF graph validation