Institut
Mines-Télécom

Programmer pour le Web sémantique et le Web des données

Télécom ParisTech
Jean-Claude Moissinac – Juin 2020
(et l'aide de F.Suchanek)

Problématiques

Exemples

- Créer par programme un graphe de connaissances
- Utiliser par programme un graphe de connaissances
- Exploiter des connaissances accessibles sur le Web

Web sémantique et standards W3C

- Expression de faits (RDF)
- Identificateurs uniques (URI)
- Expression de concepts (RDFS)
- Partage de vocabulaires
- Et description de contraintes (OWL, SHACL)
- Requêtes ‘sémantiques’ (SPARQL)
- Publication de données liées
- Publications ‘sémantisées’ (RDFa)

Représentation des connaissances

Granules de connaissances

- **Les triplets RDF**

- **(sujet)(prédicat)(objet)**

- **Sujet: l'entité sur laquelle porte la connaissance**
- **Prédicat: l'affirmation qu'on fait sur le sujet; une propriété applicable au sujet**
- **Objet: valeur qu'on associe au prédicat (valeur de la propriété**

L'ensemble constitue une connaissance sur le sujet

Resource Description Framework

RDF

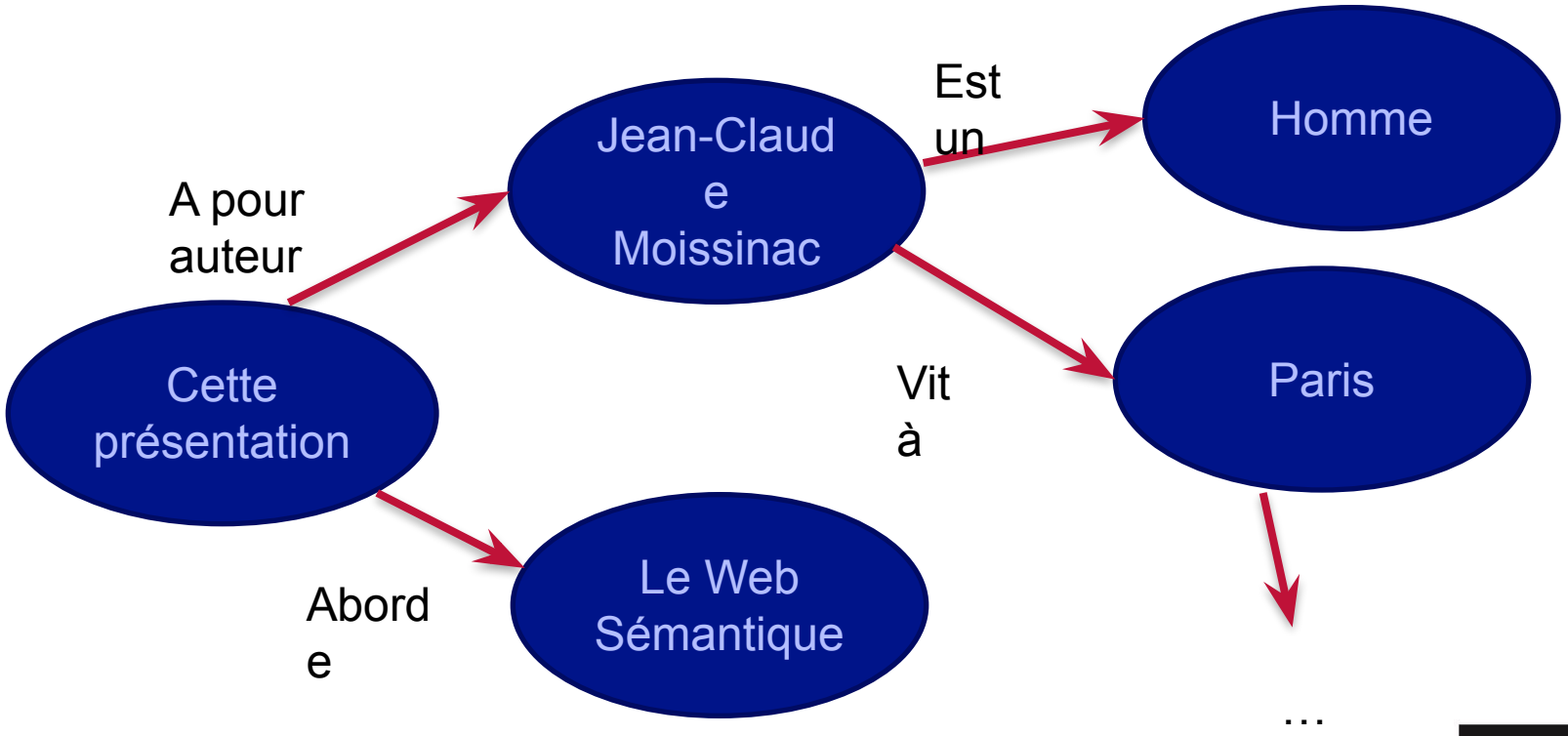
- **Resource**
  - Pages, images, vidéo, données…
  - Accessibles par une URI (ex: http://monsite.fr/...)
- **Description**
  - Propriétés et relations de la ressource
- **Framework**
  - Modèle (simple), langage, syntaxes pour ces descriptions

RDF, le modèle

- Décrire tout ce qu'on peut par des triplets
- (sujet, prédicat, objet)
- Cette présentation a pour auteur Jean-Claude Moissinac et aborde le Web Sémantique
- (cette présentation, a pour auteur, Jean-Claude Moissinac)
- (cette présentation, aborde, le Web Sémantique)

RDF définit des graphes

- **Un ensemble de triplets RDF peut être vu comme un graphe orienté et étiqueté**



Stockage d'ensemble de triplets RDF

Formats de représentation du RDF

- **RDF/XML**
- **Turtle**
- **N3**
- **RDFa**
- **Json-ld**

- **Conversions de format**
  - Voir EasyRdfVoir EasyRdf (php), RDF Translator, les possibilités en lecture/écriture de rdflib (python)

Exemple Turtle (extrait)

```
@prefix ns1: <http://erlangen-crm.org/current/> .
@prefix ns2: <http://datamusee.givingsense.eu/onto/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://datamusee.givingsense.eu/onto/event/creation/226737>
        a       ns1:E12_Production,       ns1:E65_Creation ;
    ns1:P108_has_produced
<http://datamusee.givingsense.eu/onto/work/physical/226737> ;
    ns1:P4_has_time_span <http://datamusee.givingsense.eu/onto/time/226737> ;
    ns1:P94_has_created
<http://datamusee.givingsense.eu/onto/work/concept/226737> .

<http://datamusee.givingsense.eu/onto/time/226737> a ns1:E52_Time-Span .

<http://datamusee.givingsense.eu/onto/work/concept/226737>
        rdfs:label "Soleil couchant sur la Seine à Lavacourt, effet d'hiver"@fr ;
        ns2:DME10_URL
"https://www.parismuseescollections.paris.fr/fr/petit-palais/oeuvres/soleil-couchan
t-sur-la-seine-a-lavacourt-effet-d-hiver" ;
```

Exemple RDf/XML (extrait)

Exemple précédent traduit à l'aide de http://www.easyrdf.org/converter

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ns0="http://erlangen-crm.org/current/"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:ns1="http://datamusee.givingsense.eu/onto/"
        xmlns:ns2="http://parismusees.givingsense.eu/onto/">

  <rdf:Description
rdf:about="http://datamusee.givingsense.eu/onto/event/creation/226737">
    <rdf:type rdf:resource="http://erlangen-crm.org/current/E12_Production"/>
    <rdf:type rdf:resource="http://erlangen-crm.org/current/E65_Creation"/>
    <ns0:P108_has_produced
rdf:resource="http://datamusee.givingsense.eu/onto/work/physical/226737"/>
    <ns0:P14_carried_out_by>
      <ns0:E21_Person
rdf:about="http://joconde.givingsense.eu/onto/artist/8de61f58-e275-3996-82b4-584e
b63d3c74">
        <rdfs:label xml:lang="fr">Monet, Claude</rdfs:label>
        <ns0:P100_was_dead>
          <ns0:E69_Death
…
```

Exemple Json-Ld (extrait)

Exemple traduit avec http://rdf-translator.appspot.com/


```
{
  "@context": {
    "ns0": "http://erlangen-crm.org/current/",
    "ns1": "http://datamusee.givingsense.eu/onto/",
},
  "@graph": [
    {
      "@id": "http://datamusee.givingsense.eu/onto/work/concept/226737",
      "ns0:P65_is_shown_by": {
              "@id": "http://datamusee.givingsense.eu/onto/work/physical/226737"
      },
      "ns1:DME10_URL":
"https://www.parismuseescollections.paris.fr/fr/petit-palais/oeuvres/soleil-couchant-sur-la-sei
ne-a-lavacourt-effet-d-hiver",
      "rdfs:label": {
        "@language": "fr",
        "@value": "Soleil couchant sur la Seine à Lavacourt, effet d'hiver"
      }
…
```

RDF Store – TripleStore - QuadStore

- **Base de données spécialisée pour le stockage et les requêtes sur des ensembles de triplets RDF**
- **Quad -> les triplets appartiennent à un graphe**
  - (graphe, (sujet, prédicat, objet))
- **Certains sont des surcouches d'une base de donnée SQL**
- **SPARQL**
  - Langage de requête le plus utilisé
    - Voir plus loin
- **Services REST d'envoi de requête**

Jena- Fuseki

- **Source: Fondation Apache**
- **Jena**
  - Librairie Java de manipulation de données RDF
- **Fuseki**
  - Triple store basé sur Jena et interfacé avec Jena

- **Très utilisé**
- **Bonnes performances**
- **Possibilité de couplage avec des raisonneurs**

Sesame/Eclipse RDF4J

- **Source: Fondation Eclipse**
  - librairie Java de manipulation de données RDF
- **Très utilisé**
  - initialement connu sous le nom Sesame
- **Bonnes performances**
  - Possibilités de préconstruire divers index (spoc, posc…)
- **Possibilité de couplage avec des raisonneurs**

Virtuoso

- **Source: Open Link Software**
- **Open Source ou commercial**
- **Plus large qu'un Triple Store**
  - Base XML, serveur de pages Web…
- **Très utilisé**
- **Exemple: DBPediaExemple: DBPedia, Europeana**
  - Nombreuses requêtes
  - Graphe très grand
    - DBPedia > 3 milliards de triplets
  - Distribution de charge

MarkLogic

- **Ingestion de documents**
  - XML, TXT, JSON, …
- **Représentation par des triplets**
  - Avec liens vers les sources
- **Interrogations SPARQL et XQUERY**
- **Connu pour un très bon passage à l'échelle avec de nombreux documents et de nombreux triplets**

Autres

- Neo4J, graph database
- BlazeGraph
- AllegroGraph
- StarDog
- GraphDB
- 4store
- CubicWeb
- Dydra
- TopBraid

Source https://db-engines.com/en/ranking_trend/rdf+store

Benchmark

- **Voir https://www.researchgate.net/publication/316975670_LargeRDFBench_A_Billion_Triples_Benchmark_for_SPARQL_Endpoint_Federation**

- **Outil de comparaison**

SPARQL

SPARQL

- Un langage de requête sur un graphe de triplets
- Inspiré de SQL
- S'appuie sur la notion de triplet

SPARQL: requêtes sur des triplets

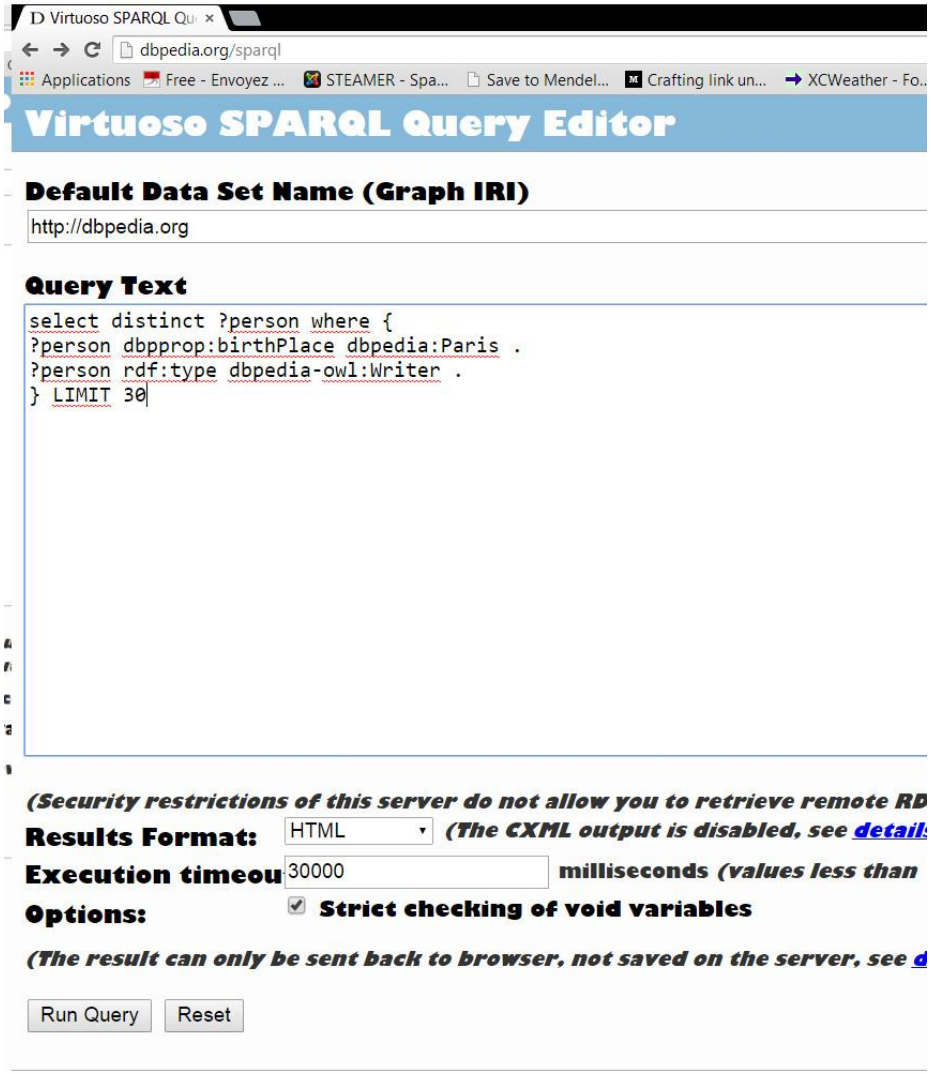
```
select distinct ?p where
{
?p dbpedia-owl:birthPlace dbpedia:Paris .
?p dbpedia-owl:occupation dbpedia:Writer
} LIMIT 30
```

Sélectionne moi les personnes nées à Paris et qui ont comme activité écrivain, donne moi les 30 premiers

(en fait, sélectionne moi tous les triplets qui remplissent la condition, récupère la valeur à la place de la variable ?p)

Demo

Exemple de requête



person
http://dbpedia.org/resource/Jaime_de_Angulo
http://dbpedia.org/resource/Hans_Stefan_Santesson
http://dbpedia.org/resource/Antoine_Blondin
http://dbpedia.org/resource/Pierre-Antoine-Augustin_de_Piis
http://dbpedia.org/resource/Ana%C3%AFs_Nin
http://dbpedia.org/resource/Anatole_France
http://dbpedia.org/resource/St%C3%A9phane_Mallarm%C3%A9
http://dbpedia.org/resource/Cecile_de_Brunhoff
http://dbpedia.org/resource/Pierre_de_Marivaux
http://dbpedia.org/resource/Anna_de_Noailles
http://dbpedia.org/resource/Florian_Zeller
http://dbpedia.org/resource/Henri_Loyrette
http://dbpedia.org/resource/Eric_O'Keefe
http://dbpedia.org/resource/Jean_de_La_Bruy%C3%A8re
http://dbpedia.org/resource/Prosper_M%C3%A9rim%C3%A9e
http://dbpedia.org/resource/Augustin_Cochin_(historian)
http://dbpedia.org/resource/Florence_Delay
http://dbpedia.org/resource/Louis_Racine
http://dbpedia.org/resource/Marquis_de_Sade
http://dbpedia.org/resource/Antoine_Fureti%C3%A8re
http://dbpedia.org/resource/Stuart_Cloete
http://dbpedia.org/resource/Nicolas_Iljine
http://dbpedia.org/resource/Elisabeth_Beresford
http://dbpedia.org/resource/Alexandre_Mercereau
http://dbpedia.org/resource/Maurice_Druon

Commandes SPARQL: consultation

- **SELECT**
  - Sélection de triplets dans un graphe
  - Résultat dans divers formats demoRésultat dans divers formats demo demo
- **CONSTRUCT**
  - Construit des triplets à partir d'une sélection de triplets
  - Résultat dans divers formats demo
- **ASK**
  - Teste si un triplet existe
- **DESCRIBE**
  - Décrit un nœud du graphe à l'aide d'un ensemble de triplets qui le concernent
  - Dépend de l'implémentation demo

Autre exemple de CONSTRUCT

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>

CONSTRUCT
{
    ?resource2 ?property1 ?resource1 .
}
WHERE
{
   ?property1 owl:inverseOf ?property2 .
   ?resource1 ?property2 ?resource2 .
}
```

Commandes SPARQL: modification

- **INSERT**
  - Sur le modèle d'un CONSTRUCT
  - Introduit dans le graphe les triplets construits
- **DELETE**
- **DROP GRAPH <uri>**
  - S'applique à un graphe entier
- **CREATE GRAPH <uri>**

Requête fédérative

- **Requête qui accède à des points d'accès distants, différents de celui sur lequel est fait la requête**
- **Pas supporté par tous les points d'accès**
  - Fuseki OK
  - Virtuoso 7 OK
- **Exemple**

```
1   PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2   PREFIX prop-fr: <http://fr.dbpedia.org/property/>
3   PREFIX refc:
4   <http://givingsense.eu/onto/refCulture/refHistArts.rdf#>
5   PREFIX dbpedia-owl: <http://fr.dbpedia.org/ontology/>
6
7   SELECT  distinct ?vtag  ?s WHERE {
8     ?fiche refc:tags ?tags .
9     ?tags ?v ?vtag .
10    BIND (STRLANG(?vtag,"fr") AS ?tag) .
11    SERVICE <http://fr.dbpedia.org/sparql>
12    {
13      ?s rdfs:label ?tag .
      }
    } LIMIT 100
```

Accès par programme d'un point d'accès SPARQL

- **Envoi d'une requête à un serveur**
  - Avec un type de résultat souhaité: JSON, Turtle, XML,…
  - Méthode: AJAX
  - Problème éventuel: CORS (cf javascript)
- **Récupération de la réponse**
  - SELECT renvoie des données
  - ASK renvoie un booléen
  - CONSTRUCT renvoie des triplets
  - DESCRIBE renvoie des triplets

Référence

- **Learning SPARQL**
  - Bob du Charme
  - Edition O'Reilly

Programmer

En Java

- **Apache Jena**
  - Création/modification de graphe RDF
  - Stockage en mémoire ou permanent
  - Support de SPARQL
- **Eclipse RDF4J**
  - Possibilités similaires

Python

- **sparqlwrapper**
  - Interrogation de point d'accès SPARQL
- **rdflib**
  - Création/modification de graphe
- **Pymicrodata**
  - Parse du HTML5 pour en extraire les microdata
- **pyrdfa3**
- **OWL-RL**
  - Raisonneur qui étend un graphe suivant les règles d'une ontologie

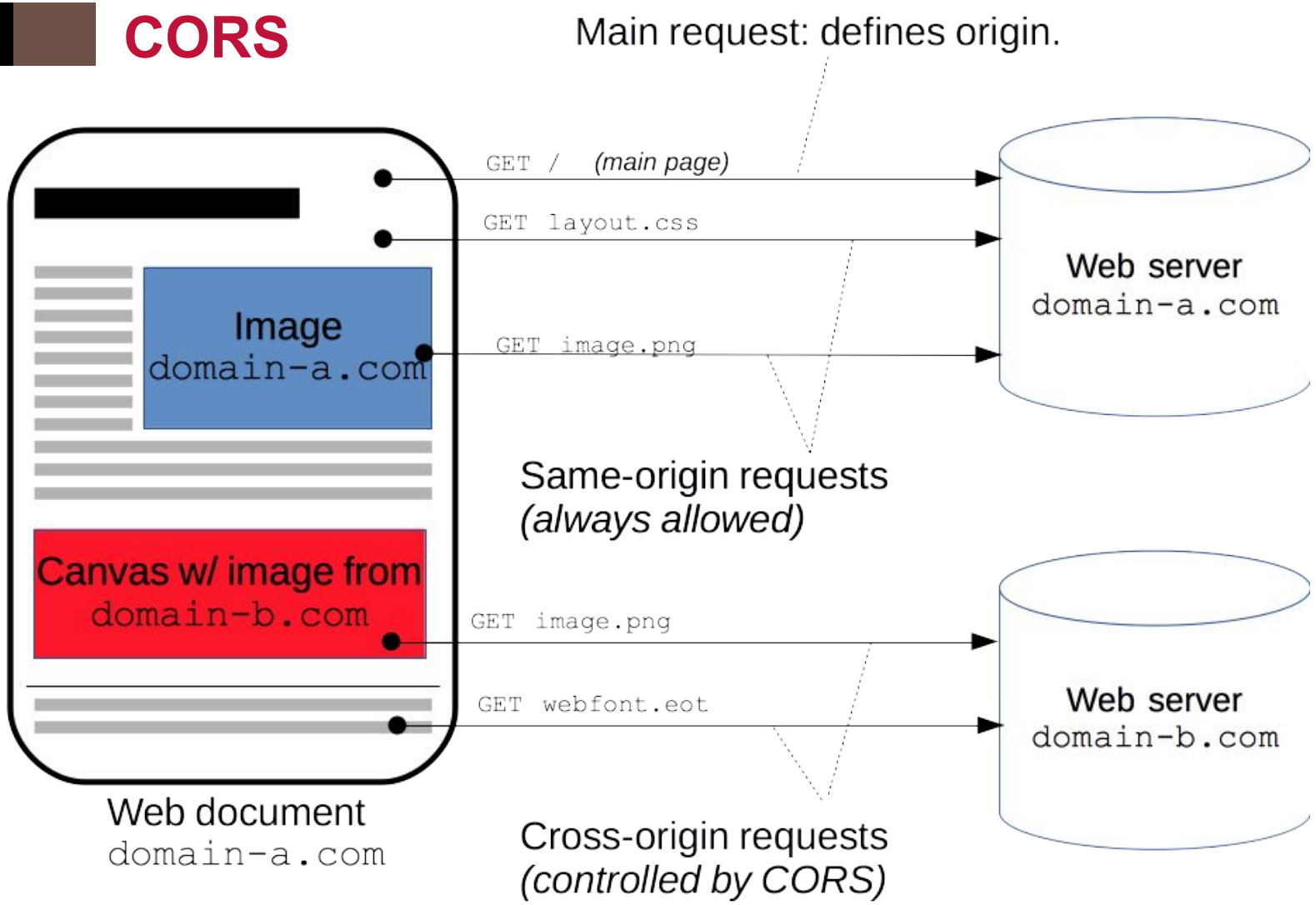
PHP

- **Easyrdf**
  - Manipulation de graphe RDF en PHP
  - Fournit une interface vers un point d'accès SPARQL
- **Semsol/Arc2**
  - **Easy RDF and SPARQL for LAMP systems**
  - Solution simple pour hébergement PHP
  - Stockage MySQL

Javascript, navigateur et nodejs

- Rdflib.js
  - Manipulation de données RDF en JS
- green-turtle
  - Parser de RDFa, microdata, turtle
- rdfstore-js
  - RDF Store écrit en javascript
- **D3-sparql**
  - Envoie d'une requête sparql et récupération de tableaux de données pour intégration d3

CORS



Source: https://developer.mozilla.org/fr/docs/HTTP/Access_control_CORS

CORS et le web sémantique

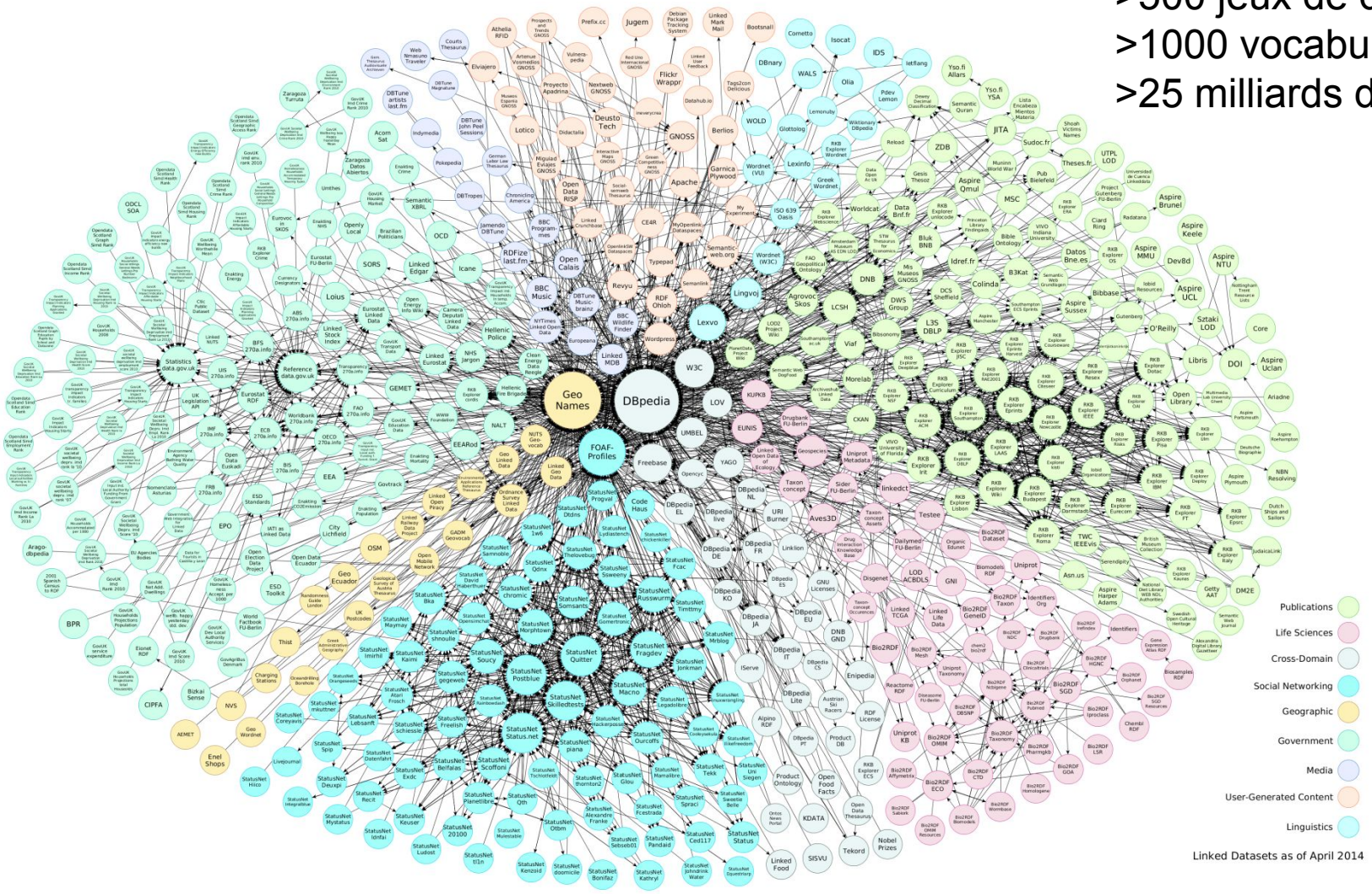
- Voir
- https://onsem.wp.imt.fr/2015/07/28/cors-web-semantique-et-donnees-liees/

Connaissances issues du web

Vocabulaires

Linked Open data

>500 jeux de données
>1000 vocabulaires
>25 milliards de triplets



TELECOM ParisTech

Ontologie (informatique)

- Ensemble structuré des termes et concepts représentant le sens d'un champ d'informations
- Constitue un modèle de donnéesConstitue un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts.
- Permet de raisonner à propos des objets du domaine concerné.
- Les conceptsLes concepts sont organisés dans un grapheLes concepts sont organisés dans un graphe dont les relations peuvent être :
  - des relations sémantiques ;
  - des relations de subsomption.
- objectif principal: modéliser un ensemble dans un domaine donné

Publication de vocabulaires

- **Des vocabulaires sont régulièrement publiés**
- **But: partager les mêmes URIs pour exprimer des choses de la même façon dans une communauté d'utilisateurs**
  - D'un même domaine
  - De domaines transverses
- **Facilite le croisement et la déduction par combinaison de jeux de données**

Vocabulaires généraux

- **Rdf**
  - Rdf:type
- **Rdfs**
  - *rdfs:subClassOf, rdfs:property, rdfs:domain, rdfs:range*
- **Dublin Core**
  - xmlns:dc=http://purl.org/dc/elements/1.1/
  - **dc:title … description de documents**
- **(Dolce)**
- **Geo84**
  - Geo:lat, geo:lon
- **Foaf**
  - **foaf:Person -> foaf:name**
- **…**

Trouver un vocabulaire

- **Lov**
- **http://lov.okfn.org/dataset/lov/**
- **Demo**
- **Focus**
  - Schema.org

Créer son vocabulaire

- **Toute organisation peut créer ses URIs**
  - Simplicité de mise en œuvre de linked data
- **Plus un jeux de données sera décrit avec des vocabulaires partagés, plus des liens entre données pourront être créés (trouvés, cf SPARQL)**

Créer son vocabulaire: exemple

- **Datamusee.fr possédé par datamusee**
- **Création d'un sous-domaine:**
  - Onto.datamusee.fr
- **Besoin d'associer une url à une œuvre d'art**
  - Utilisation globale du vocabulaire CIDOC-CRM
  - Création d'une propriété
  - http://onto.datamusee.fr/has_url
  - Domaine: toute entité
  - Range: une url (literal de type string)
  - Description: associe une url d'une page web relative à l'entité

Recommandations pour la publication de données

Five Stars

http://5stardata.info/en/

- ★
  - make your stuff available on the Web (whatever format) under an open license[1]
- ★★
  - make it available as structured data (e.g., Excel instead of image scan of a table)[2]
- ★★★
  - make it available in a non-proprietary open format (e.g., CSV as well as of Excel)[3]
- ★★★★
  - use URIs to denote things, so that people can point at your stuff[4]
- ★★★★★
  - link your data to other data to provide context[5]

Requête sur des données sémantiques

SPARQL, points d'accès

- http://dbpedia.org/sparql
- http://fr.dbpedia.org/sparql
- http://data.bnf.fr/sparql
- http://www.rechercheisidore.fr/sqe/
- http://europeana.ontotext.com/sparql
- … et bien d'autres

- Voir

http://www.w3.org/wiki/SparqlEndpoints

Actions de bas niveau

- Extraire des données RDF d'une page HTML
- Interroger un RDF Store
- Ajouter des données dans un RDF Store
- Générer des données RDF
- Vérifier des données RDF et les convertir

Autres objectifs

- **Associer traitement de la langue, traitement des données et sémantique**
- **Exemple: Traiter des textes pour les marquer sémantiquement**
  - DBPedia Lookup demo
  - DBPedia Spotlight demo

Exploiter un graphe de connaissances

- **Comme variable dans un programme**
  - Chargé et exploité:
  - Par exemple avec rdflib en python ou js
  - Avec Jena en Java
- **Comme 'base de données'**
  - Installer les données dans un point d'accès sparql (serveur) et effectuer des requêtes sparql
  - Par exemple avec sparql-warapper en python ou d3-sparql en javascript

Lier source de données et Web Sémantique

Reconciliation, entity linking, entity matching

- **Source table**
  - Voir par exemple OpenRefine
  - Associer des colonnes à un type de données
  - Associer des valeurs spécifiques à des URIs (ex Paris -> http://dbpedia.org/resource/Paris
- **Source texte**
  - Reconnaissance d'entités nommées
  - Exemple spotlight (cf demo)
- **Source BDD**
  - Définir des correspondances entre des tables, leurs liens et des triplets
  - Décrire formellement ces correspondances ex: D2RQ

Etapes typiques d'entity linking

Etapes inspirées de DBpedia Spotlight

- **Repérage de candidats**
  - Opération de TALN pour identifier des mots ou séquences de mots qui pourraient être des entités
- **Sélection de candidats**
  - Sélection de candidats pour lesquels on a une bonne probabilité d'association avec une entité DBpedia
- **Désambiguïsation**
  - Utilisation du contexte d'apparition de l'entité pour choisir lorsque plusieurs entités sont possibles
    - ex: Paris (France), Paris (Texas), Paris sportifs

Evaluation de la qualité des données

- **Préliminaire**
  - Hypothèse du monde ouvert
  - (Open-World Assumption)

- **Complétude**
- **Homogénéité**
- **Exactitude**
  - Ex: Yago versus DBpedia
- **Richesse**
- **Couverture**

- **Systèmes de règles: SHACL, ShEx**

Avancées récentes

- **Modélisation**
  - Graph embeddings -> construction de modèles
  - Un vecteur est construit par une méthode destinée à refléter le rôle de l'entité dans le graphe
  - Chaque entité (nœud) du graphe est alors représentée par un vecteur
  - On peut alors faire des opérations sur ces vecteurs
    - Calcul de similarité…

    Roi -> Homme

    Reine -> Femme

    V(Roi)-V(Homme)~V(Reine)-V(Femme)

    V(Reine)~V(Roi)-V(Homme)+V(Femme)

Un exemple de chaine de traitement

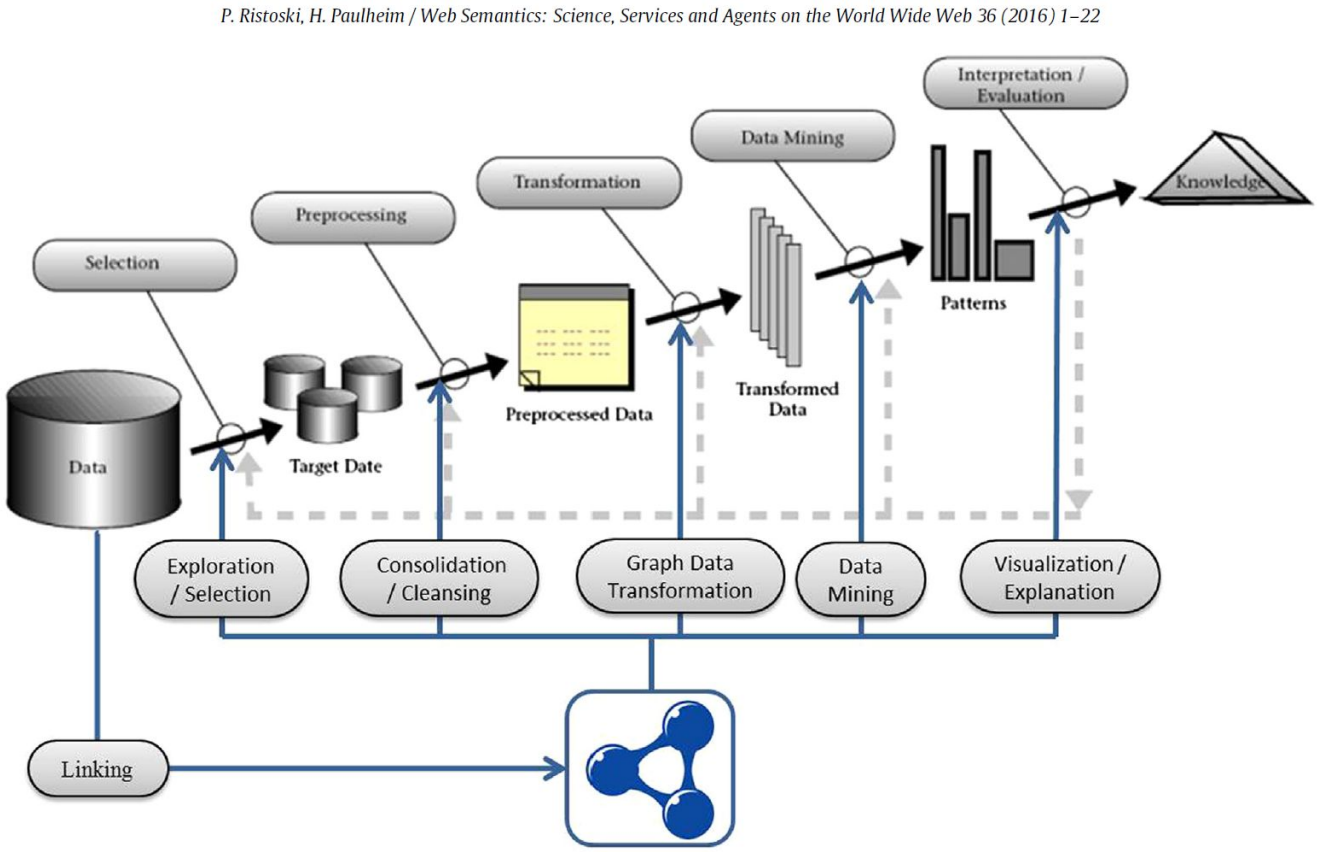


Fig. 2. An overview of the steps of the linked open data enabled KDD pipeline.

Autre exemple

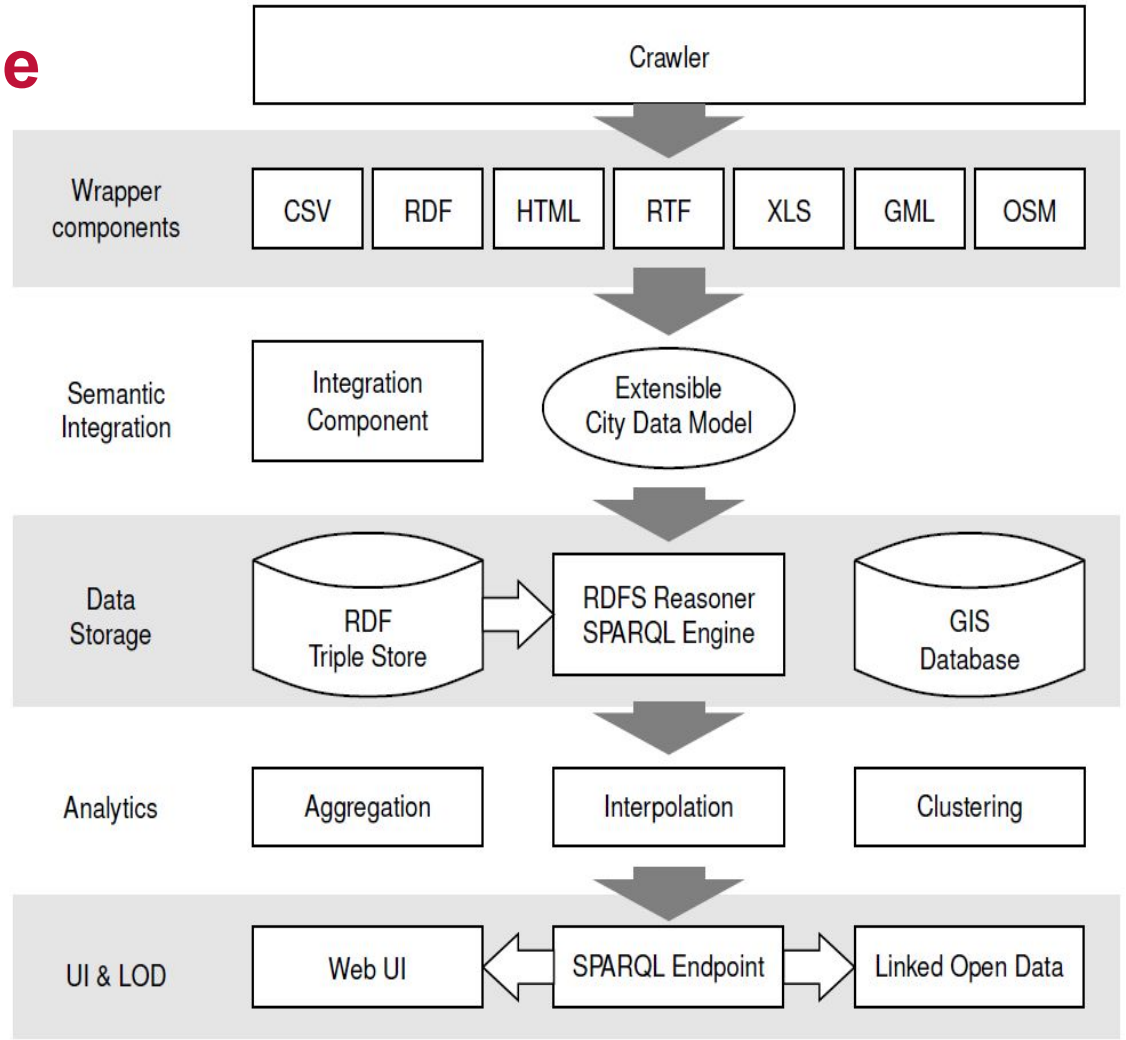


Fig. 1: City Data Pipeline architecture showing components for crawling wrapping, cleaning, integrating, and presenting information