SemWeb.Pro

État des travaux en cours au W3C et perspectives

1ères journées professionnelles du Web Sémantique
Paris, le 17 Janvier 2011

Ivan Herman
W3C

W3C Semantic Web

Travaux en cours

- Groupes “technologiques”
  - actifs :
    - RDFa 1.1
    - SPARQL 1.1
    - RDB2RML
    - Media annotation
    - Media addressing
  - en création :
    - RDF “next” (en cours d’approbation par W3C)
    - RDF API (sans doute à venir)
    - Provenance

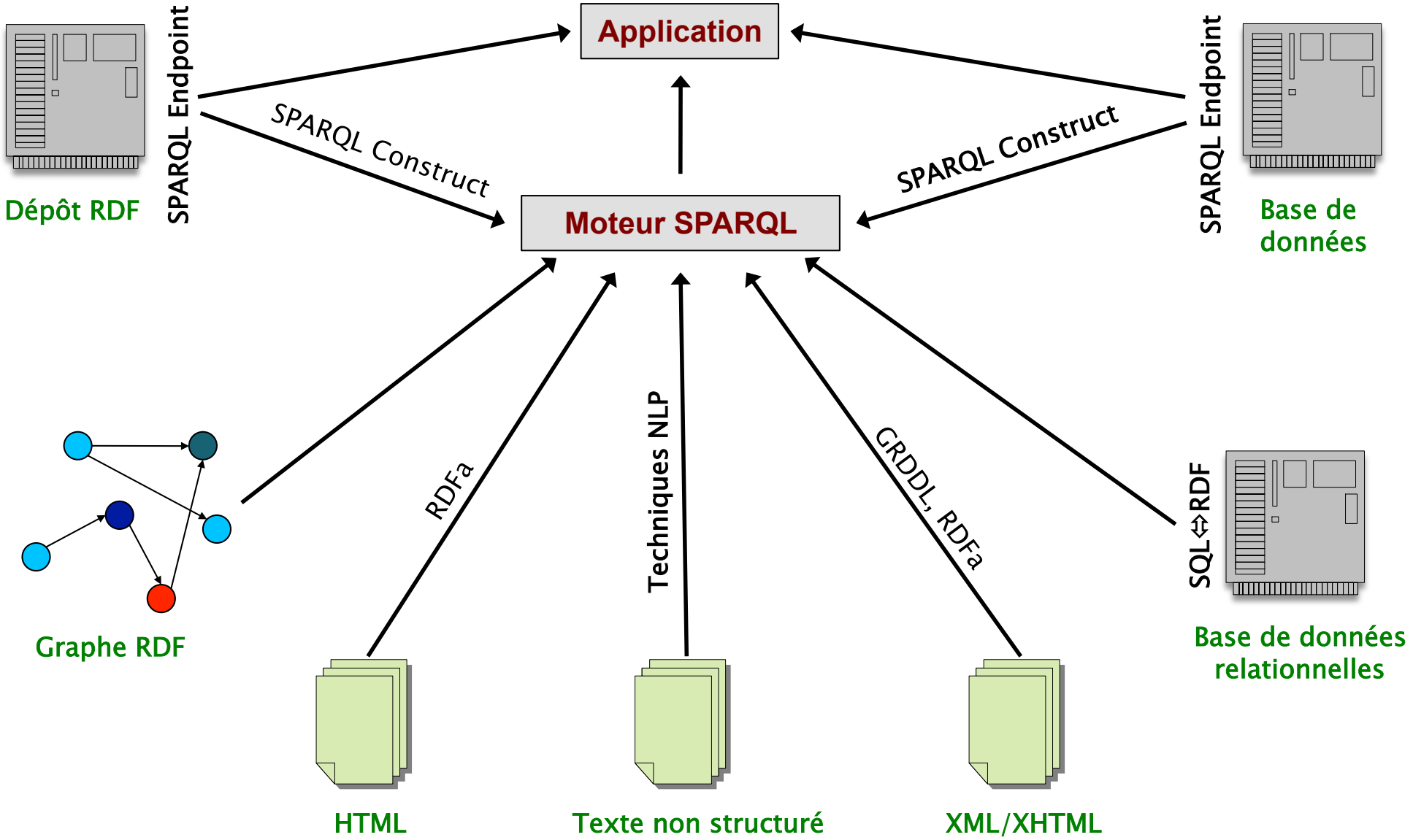
Travaux en cours

- Relations avec d'autres communautés
  - HCLS (un groupe d'intérêt très actif)
  - Publications (bibliothèques, journaux comme NYT, IPTC, etc.)
  - Gouvernements

SPARQL 1.1

W3C Semantic Web

Rôle de SPARQL



W3C Semantic Web

Tous succès a un prix…

- L'utilisation de SPARQL a révélé des problèmes…
  - SPARQL est un langage de requêtes, or nous voulons aussi *modifier* des bases de données
  - aucune relation entre SPARQL et OWL/RDFS/RIF
  - fonctionnalités manquantes
  - etc.
- SPARQL 1.1 essaye de trouver un compromis entre
  - ajouter de nouvelles fonctionnalités nécessaires, et
  - ne pas rendre SPARQL illisible, inaccessible, trop difficile a implémenter

Agrégation

- *Trouver le prix minimum pour un livre :*

```
SELECT (MIN(?prix) AS ?prix_minimum)
WHERE {
  ?livre a:price ?prix .
}
GROUP BY ?livre
```

- Autres exemples : MAX, AVG, SUM, …

W3C Semantic Web

Requêtes imbriquées

- *Trouver les gens que je connais avec leurs noms le plus court :*

```
SELECT ?y ?nom_court
WHERE {
  :Je v:connais ?y .
  {
    SELECT ?y (MIN(?nom) AS ?nom_court)
    WHERE {
      ?y v:nom ?nom .
    } GROUP BY ?y
  }
}
```


W3C Semantic Web

Variables et affectation

- *Générer un URI à partir d'un ISBN, et construire un triplet avec celui-ci comme sujet :*

```
CONSTRUCT {
  ?id a:title "Mon titre" .
}
WHERE {
  ?book <Book#ISBN> ?isbn .
  BIND (IRI(fn:concat("http://a.b.c/",?isbn)) AS ?id)
}
```

- J'aurais pu utiliser un SELECT imbriqué pour ?id ; cette forme est, néanmoins, bien plus simple…

Chemins de relations

```
{
  ?x foaf:knows/foaf:knows/foaf:name ?y .
  ?x foaf:knows{2}/foaf:name ?y .
  ?ancestor (ex:motherOf|ex:fatherOf)+ <#me> .
  ?z rdf:type/rdfs:subClassOf* ?type .

  …
}
```

Attention !

```
{
  <#a> rdf:rest*/rdf:first ?listElement .
}
```

- …n’est *pas* une solution complète pour une requêtes sur une liste : l’ordre des variables n’est pas garanti dans les réponses…

Négation I.

- *Trouver les personnes sans noms :*

```
SELECT ?personne
WHERE
{
    ?personne rdf:type  foaf:Person .
    FILTER NOT EXISTS { ?personne foaf:name ?name }
}
```


W3C Semantic Web

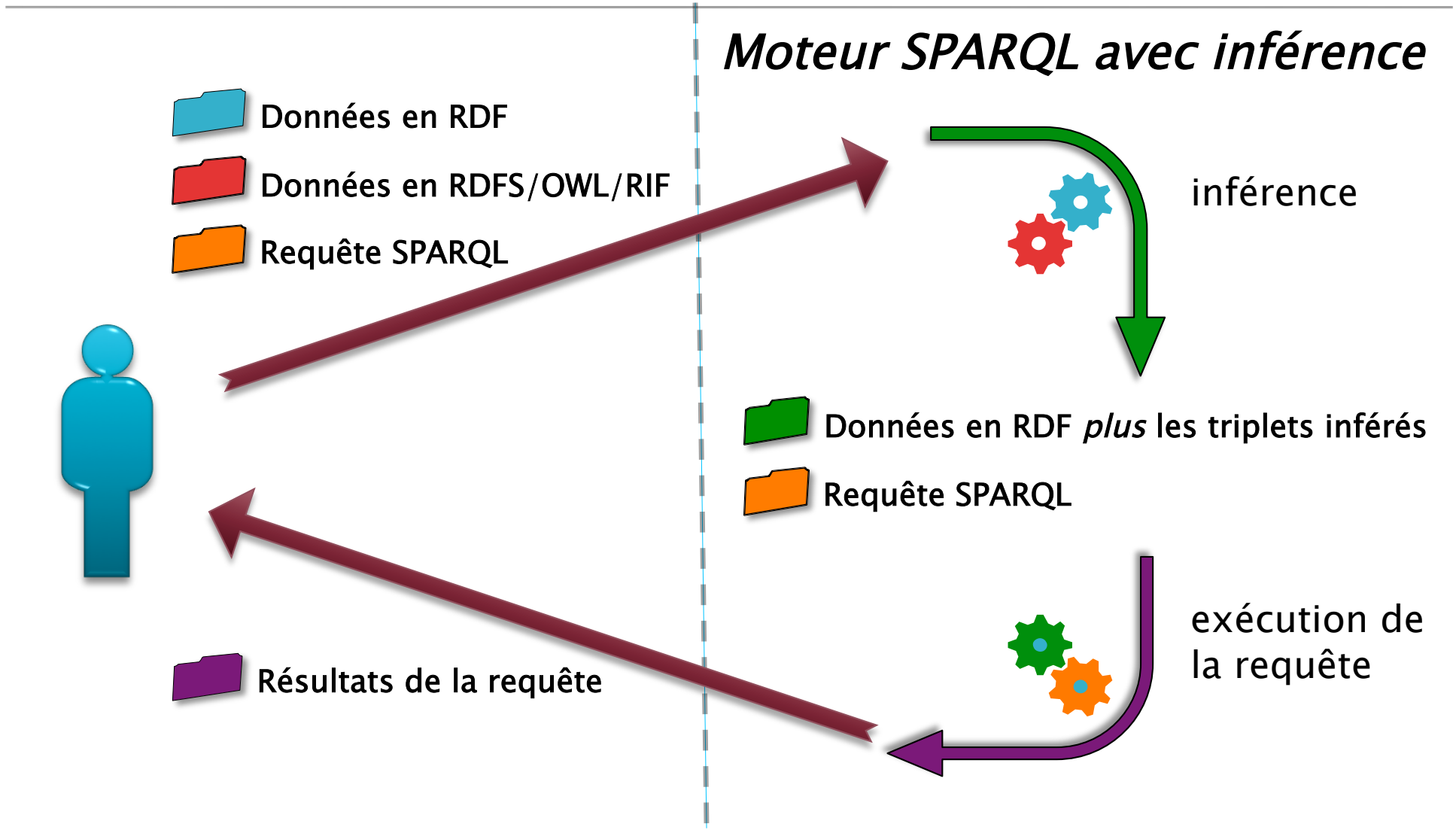
Négation II.

- *Trouver les personnes sans noms :*

```
SELECT ?personne
WHERE {
   ?personne rdf:type  foaf:Person .
   MINUS {
      ?personne foaf:name ?name .
   }
}
```


W3C Semantic Web

SPARQL 1.1 et RDFS/OWL/RIF



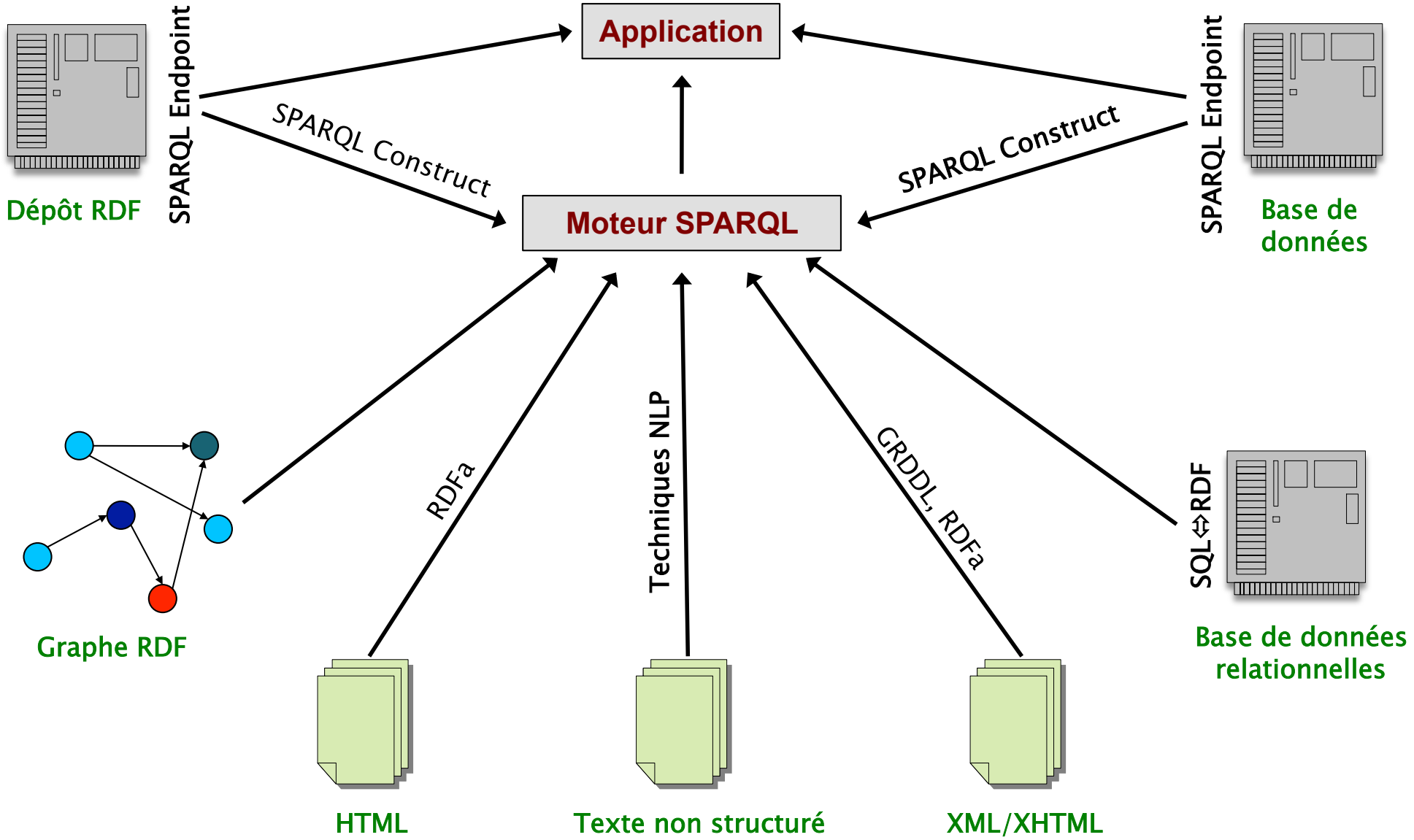
SPARQL Update

- Langage pour *changer* le contenu d'un dépôt

```
DELETE { ?livre ?p ?v } WHERE { … }
INSERT { ?livre ?p ?v } WHERE { … }
LOAD <#blabla> INTO GRAPH <#toto>
CLEAR GRAPH <#blabla>
CLEAR ALL
CREATE GRAPH <#blabla>
DROP GRAPH <#blabla>
…
```

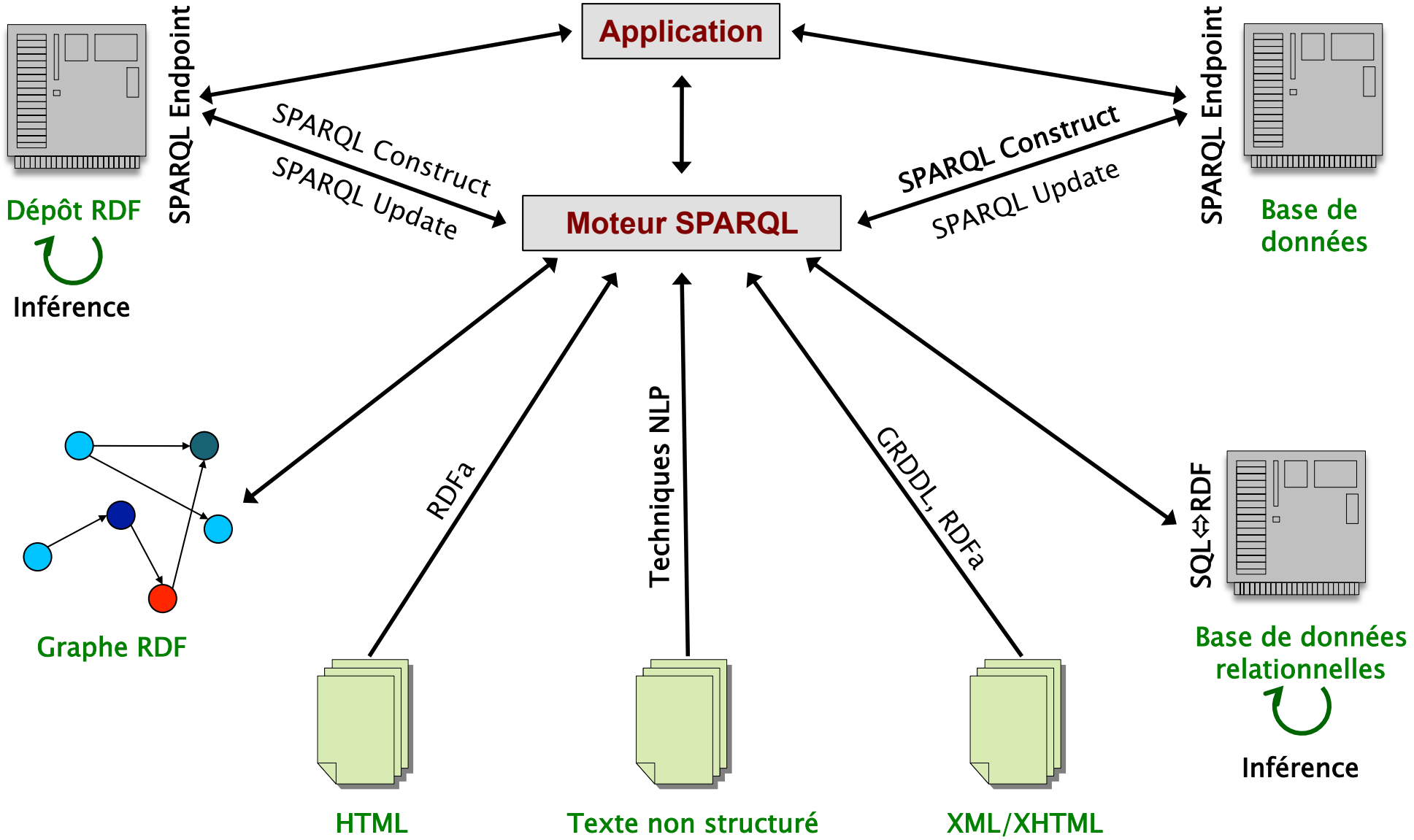

W3C Semantic Web

Rôle de SPARQL



W3C Semantic Web

Rôle de SPARQL 1.1



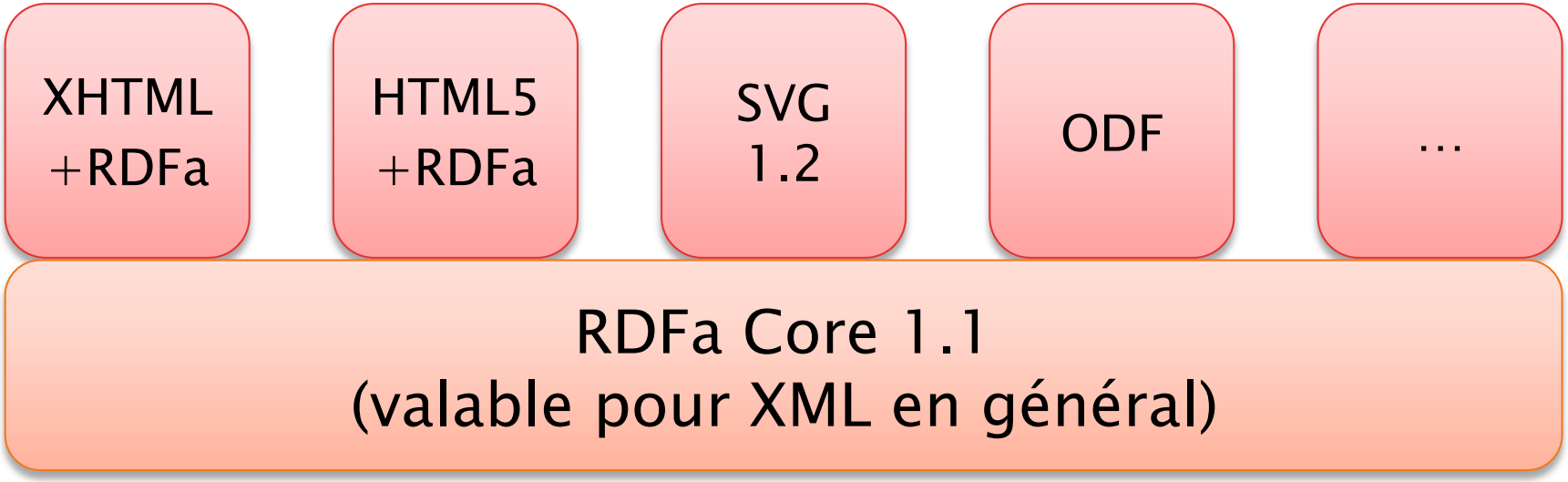
RDFa 1.1

W3C Semantic Web

Un peu comme avec SPARQL…

- Déploiement important de RDFa
  - Facebook, BestBuy, Tesco, Yahoo!, …
- Mais des problèmes apparaissent :
  - une structure un peu compliquée ex. pour des URI-s
  - une réaction défavorable à la syntaxe genre `xmlns:`…
  - l'utilisation de RDFa en dehors de XHTML
  - l'accès à RDFa à partir d'un programme

XML ou (X)HTML?



- RDFa WG définit Core et XHTML
- HTML WG définit HTML5

Problème des CURIE-s

- RDFa 1.0 introduit la notion des CURIE-s:
  - les préfixes sont définis par
    - `xmlns:p="http://a.b.c#"`
  - le CURIE `p:machin` est égale à
    - `http://a.b.c#machin`
      (p.ex. `<span rel="p:machin"...>`)

W3C Semantic Web

Plusieurs problèmes sont apparus

- L'utilisation des CURIE-s n'est pas le même partout
  - utilisable pour **`rel`**, **`rev`**, **`property`**, …
  - mais utilisable seulement comme « safe CURIE » pour **`about`**
    - **`about="[p:machin]"`**
- L'utilisation de la syntaxe **`xmlns`** est controversée
- Certains utilisateurs préfèrent utiliser uniquement des URI

CURIE-s dans RDFa 1.1

- Les préfixes peuvent être définis par :
  - **`<span prefix="p: http://a.b.c# q: http://…" … >`**
  - équivalent à **`xmlns`** mais l'utilisation de **`xmlns`** est déconseillée
- Utilisation universelle des CURIE-s, sauf pour **`href`** et **`src`**
  - les « safe CURIE » perdent leur utilité
- Utilisation des URI, si telle est la préférence :
  - **`rev="pr:b"`** veut dire:
    - si **`pr`** est un préfixe défini pour un CURIE, la valeur est l'URI défini par les règles pour les CURIE-s
    - autrement, **`pr:b`** est traité comme un URI à part entière

W3C Semantic Web

Les termes

- Les utilisateurs « non-RDF » ne veulent pas voir des URI (ni en forme URI, ni en forme CURIE)
- Ils préfèrent q.q. chose comme :

```
<body …
  <p property="name">Ivan Herman</p>
  …
</body>
```

Une solution : les vocabulaires

- Concaténation automatique du vocabulaire et du terme :

```
<body>
  <div vocab="http://xmlns.com/foaf/0.1/"
     …
     <p property="name">Ivan Herman</p>
     …
  </div>
</body>
```


W3C Semantic Web

L’importance des termes

- Les utilisateurs traditionnels de (X)HTML ne connaissent pas RDF
  - il faut leur « cacher » les URI le plus possible
- Mais : que faire si plusieurs vocabulaires sont utilisés, p.ex. **`foaf`** et **`dc`**?
  - il faut pouvoir définir, d’une manière invisible, des transformations **`terme ⇒ URI`**

Autre solution : les profils

- Un fichier *externe* à la source de RDFa
- Peut contenir des transformations pour:
  - `terme ⇒ URI`
  - `préfixe ⇒ URI`
- Le format du fichier est RDFa, RDF/XML, ou Turtle
- Les gros utilisateurs (p.ex. Google) ou les auteurs de vocabulaires (p.ex. `foaf`) peuvent publier leurs profils

Exemple de profil

- Le fichier `http://www.example.org/myProf` :

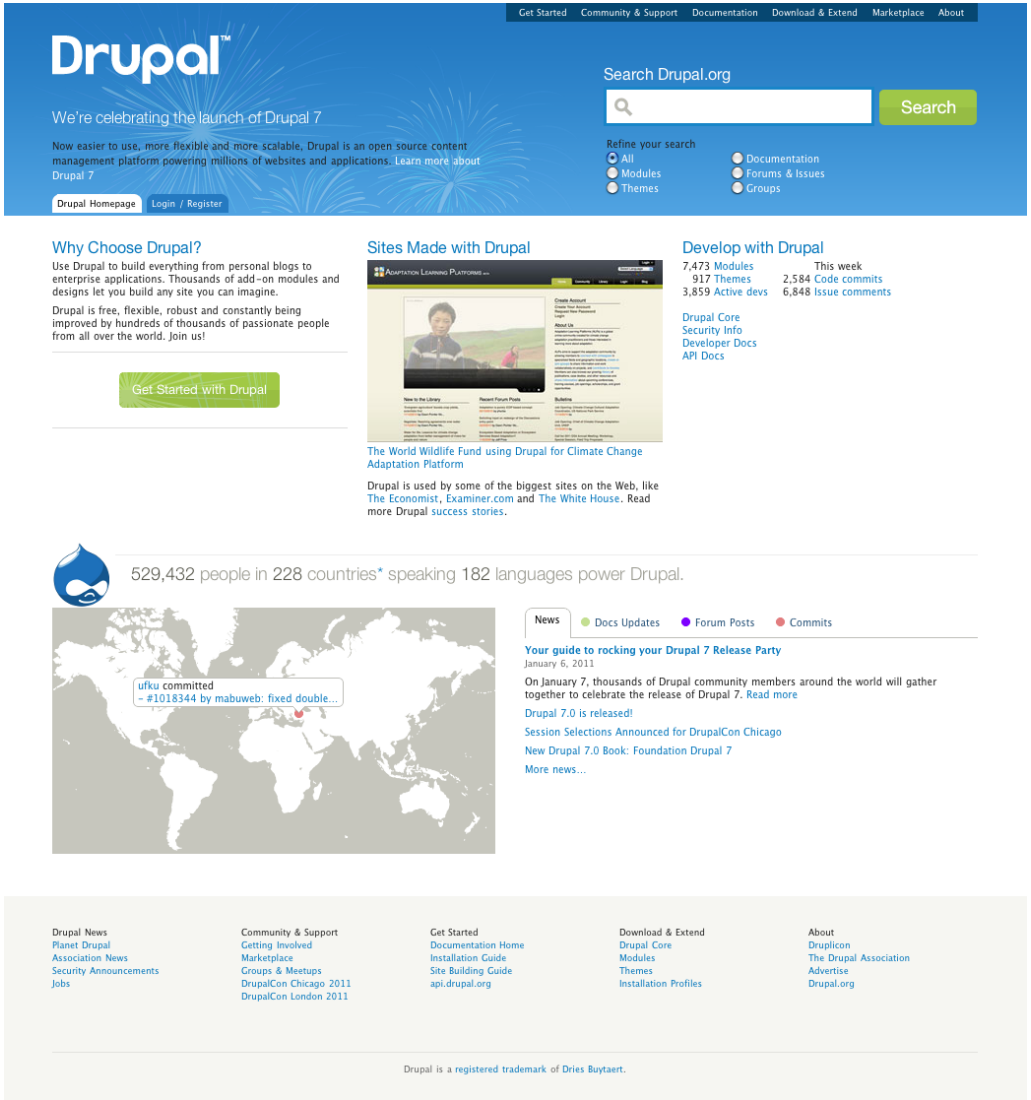
```
[ rdfa:prefix "rdfs" ;
  rdfa:uri "http://www.w3.org/2000/01/rdf-schema#" ].
[ rdfa:term "name" ;
  rdfa:uri "http://xmlns.com/foaf/0.1/name" ].
[ rdfa:term "Person" ;
  rdfa:uri "http://xmlns.com/foaf/0.1/Person" ].
```

Exemple de données

```
<div profile="http://www.example.org/myProf">
  <div about="#me" typeof="Person">
    <p>Name: <span property="name">Ivan Herman</span></p>
    <p>See my <a rel="rdfs:seeAlso"
       href="http://www.ivan-herman.net/foaf">foaf file</a>
      for more details.</p>
  </div>
  …
</div>
```


W3C Semantic Web

Une étape importante pour RDFa : Drupal 7



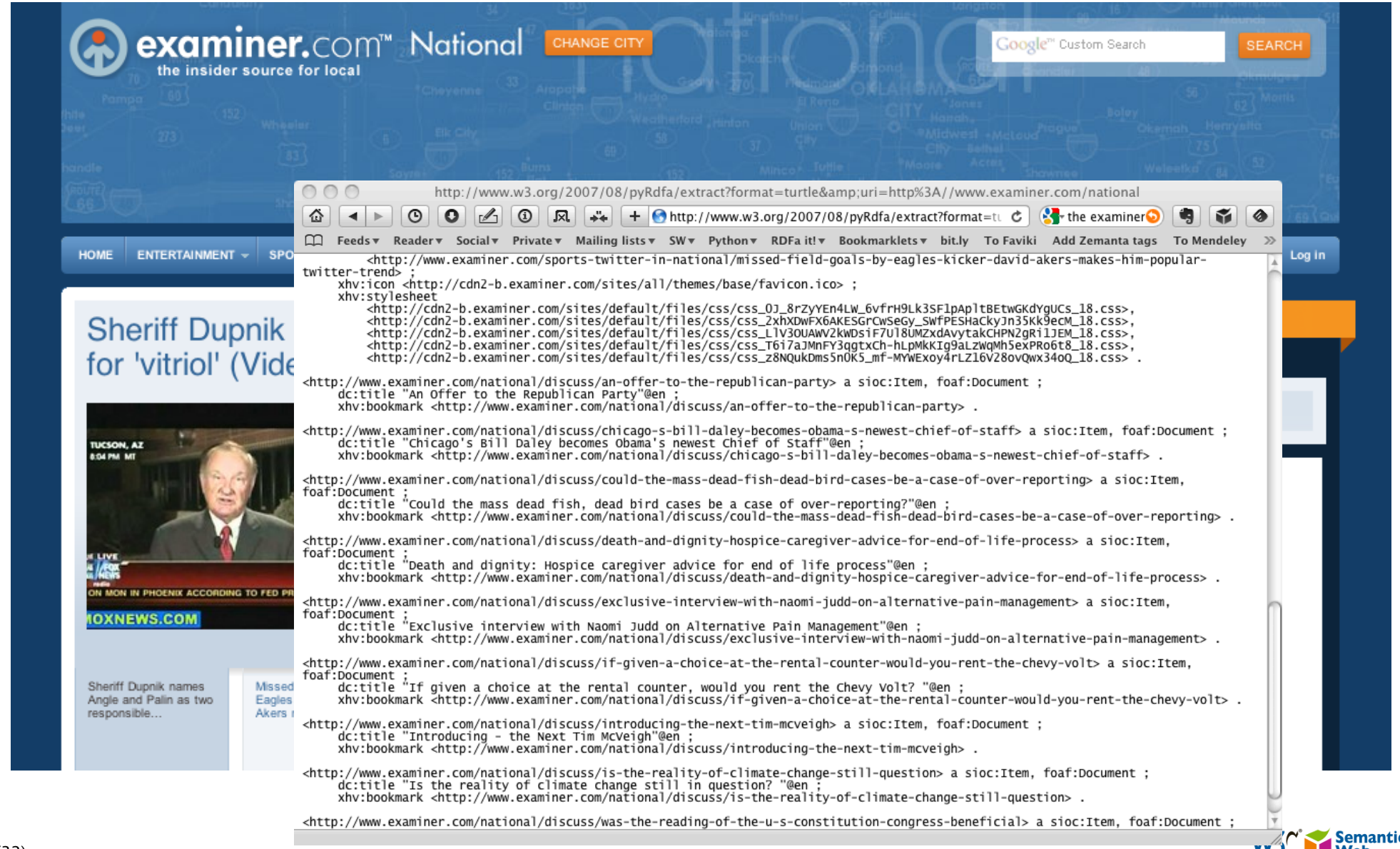
- Un système CMS important
- Génère des pages avec RDFa
- Des millions de triplets générés !

W3C Semantic Web

Examiner.com (2 jours après la publication de Drupal 7…)

W3C Semantic Web

Examiner.com (2 jours après la publication de Drupal 7…)



R2RML et « Direct Mapping »

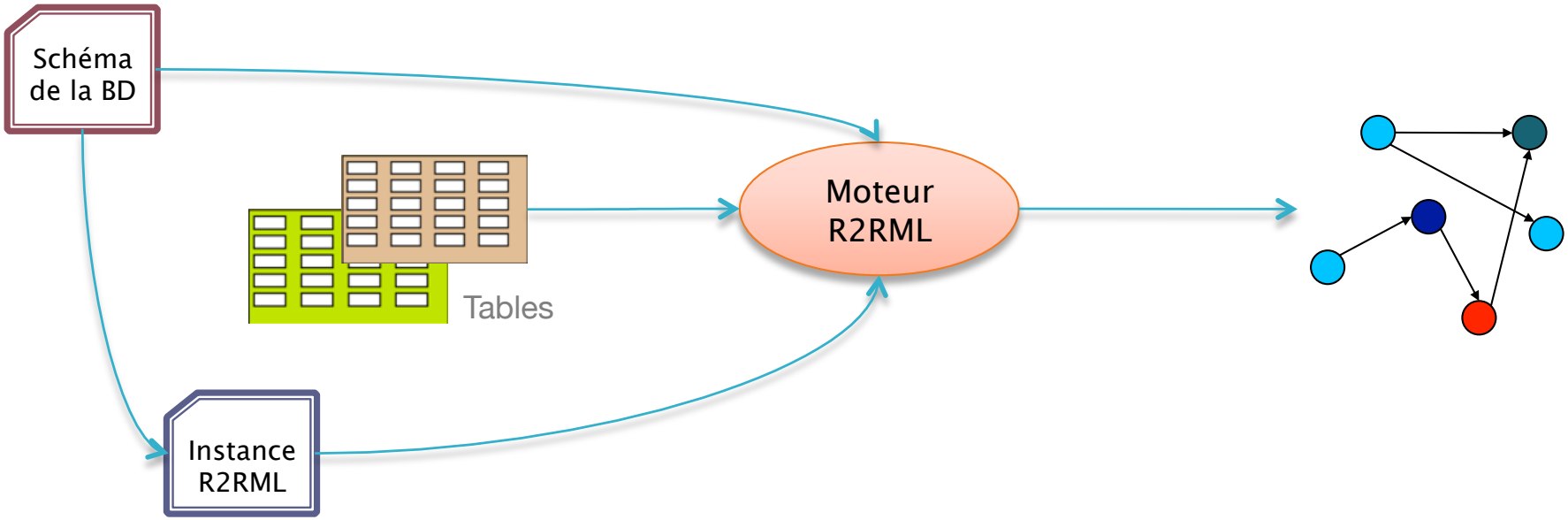
W3C Semantic Web

Quel est le rôle de R2RML ?

- Définit une transformation standard d'une table relationnelle en RDF
  - le schéma relationnel est utilisé :
    - les cellules d'une ligne produisent des triplets avec un sujet commun
    - les noms des colonnes définissent les prédicats
    - chaque cellule produit un objets
  - les différentes tables d'une même base de données sont aussi liées entre elles

Quel est le rôle de R2RML ?

- Un moteur R2RML a accès :
  - au schéma de la base de données
  - une instance de R2RML
  - les tables elles-mêmes
- … et génère un graphe RDF



W3C Semantic Web

Exemple

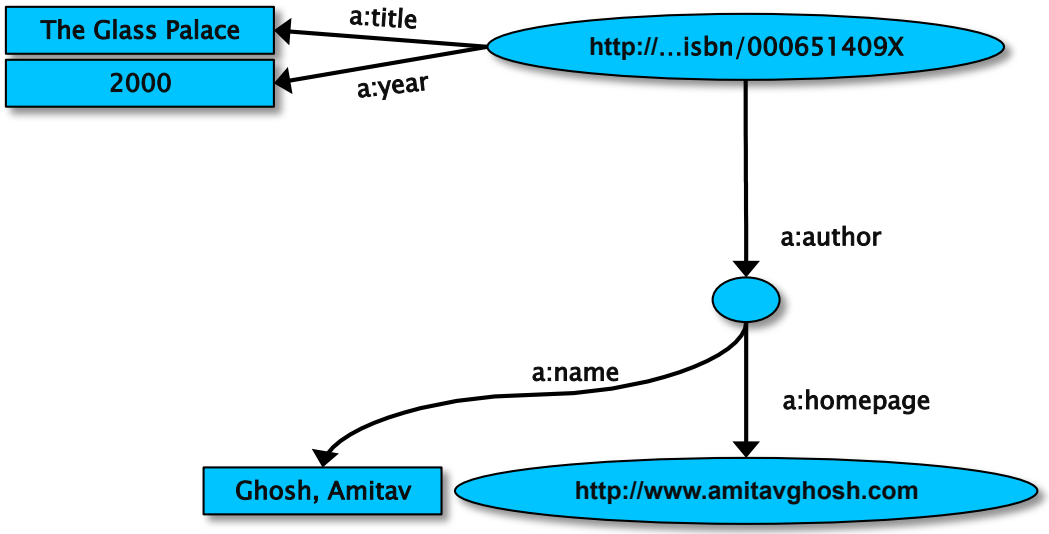
ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

Exemple

ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

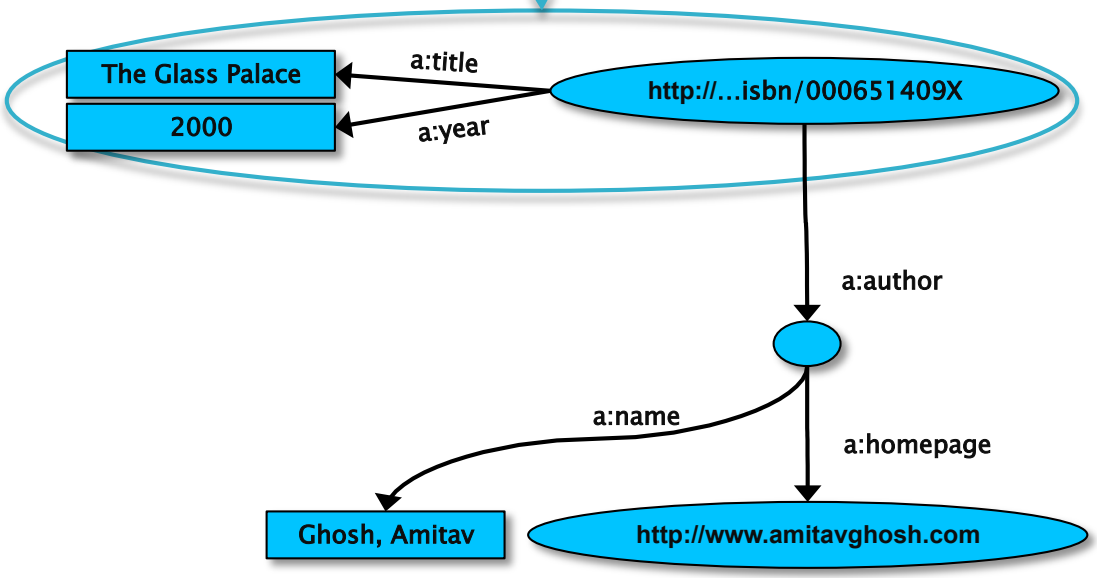


W3C Semantic Web

Exemple

ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

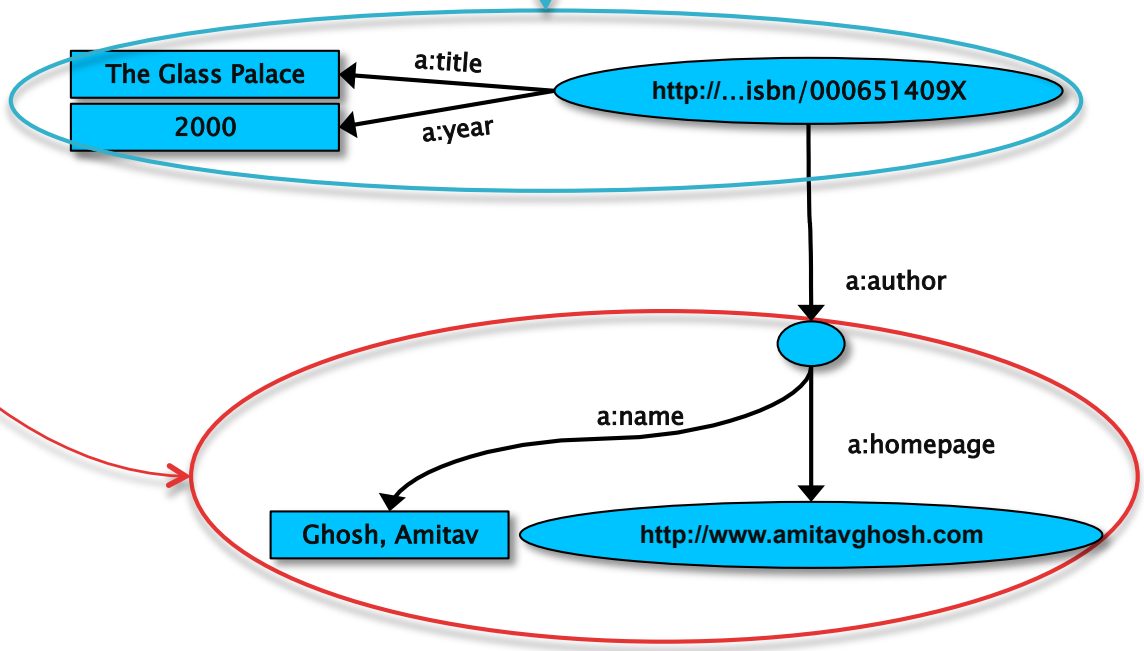


W3C Semantic Web

Exemple

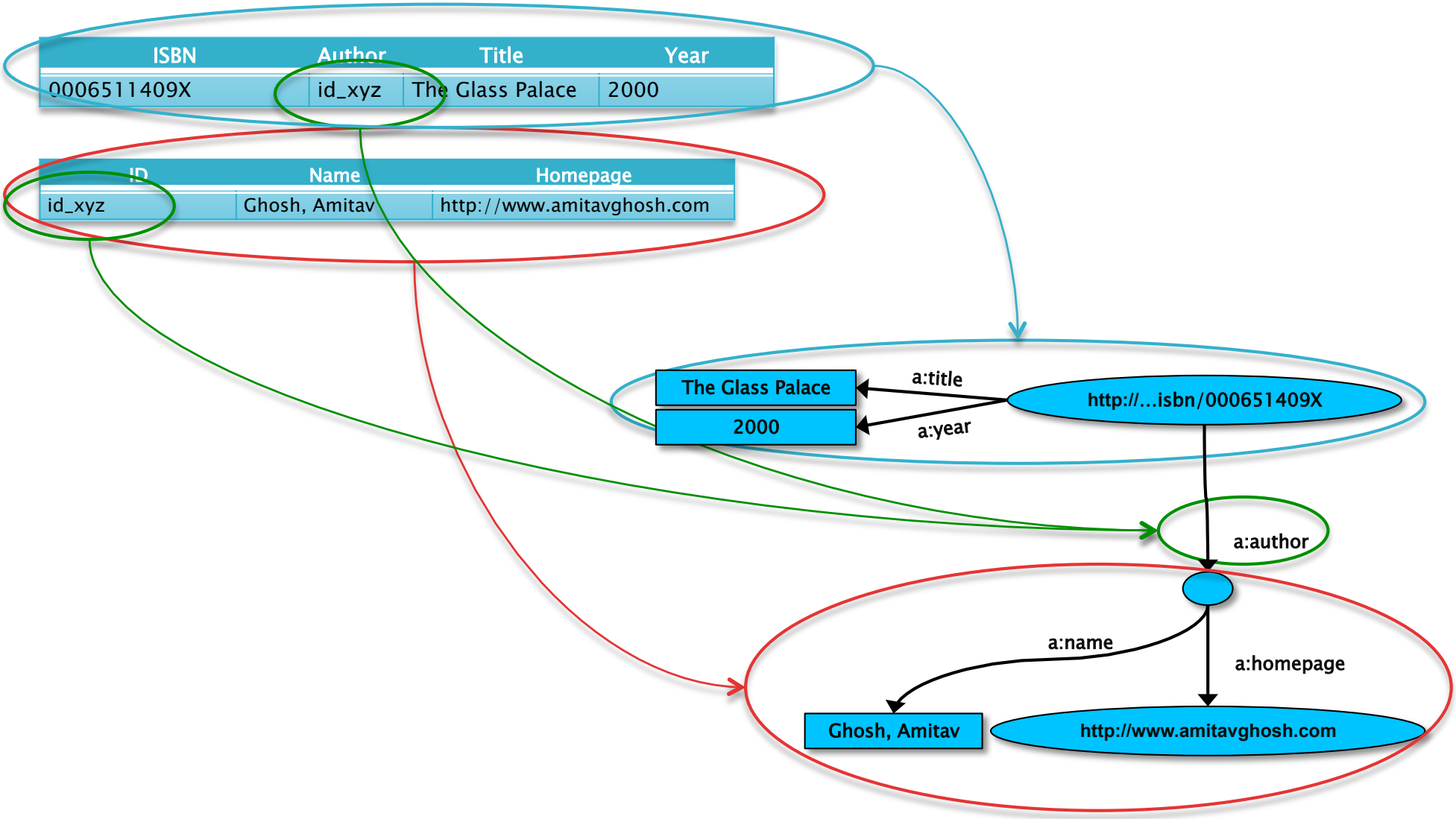
ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com



W3C Semantic Web

Exemple



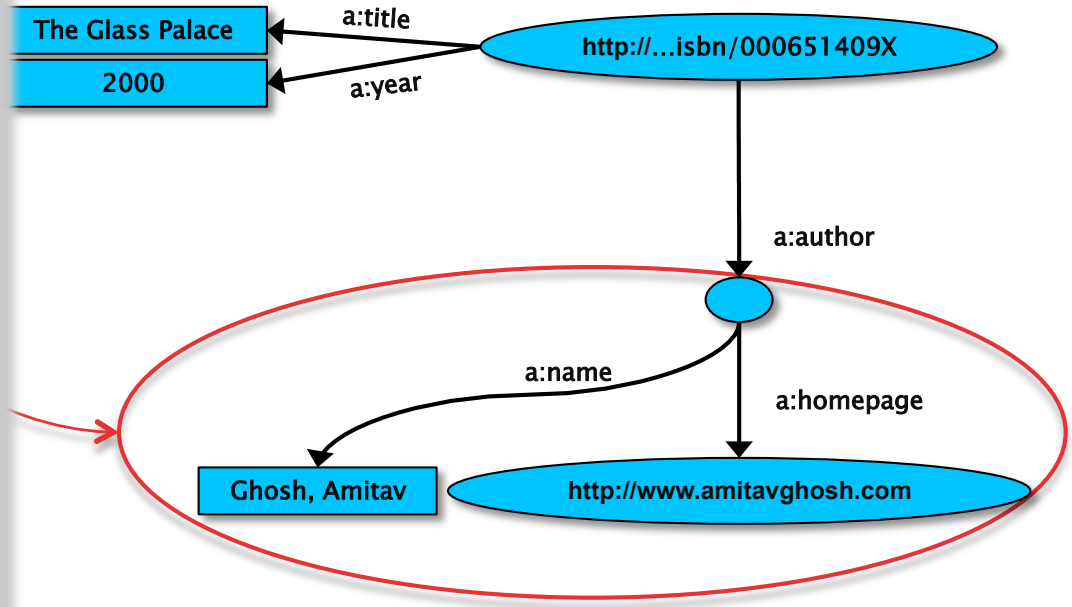
W3C Semantic Web

Première étape : transformation de la table des personnes

ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

```
:P_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [
    rr:termtype "BlankNode" ;
    rr:column "ID" ;
  ] ;
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:predicate a:name
    ];
    rr:objectMap [
      rr:column "Name"
    ]
  ] ;
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:predicate a:homepage
    ];
    rr:objectMap [
      rr:column "Homepage" ;
      rr:termtype "IRI"
    ]
  ] ;
```



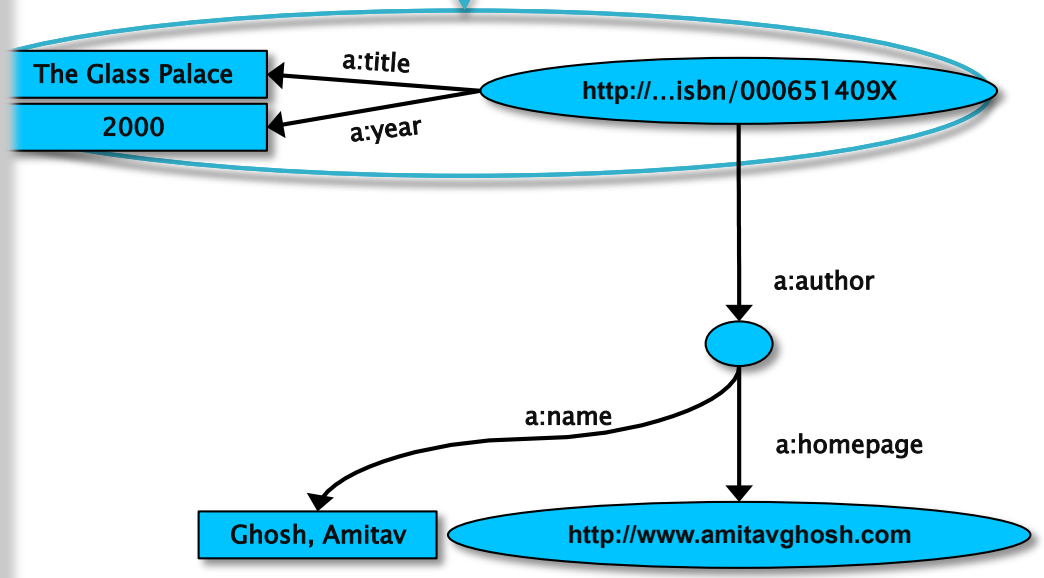
W3C Semantic Web

Deuxième étape : transformation de la table des livres

ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

```
:B_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [
    rr:template "http://...isbn/{ISBN}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:predicate a:title
    ];
    rr:objectMap [
      rr:column "Title"
    ]
  ] ;
  rr:predicateObjectMap [
    rr:predicateMap [
      rr:predicate a:year
    ];
    rr:objectMap [
      rr:column "Year" ;
    ]
  ] ;
```

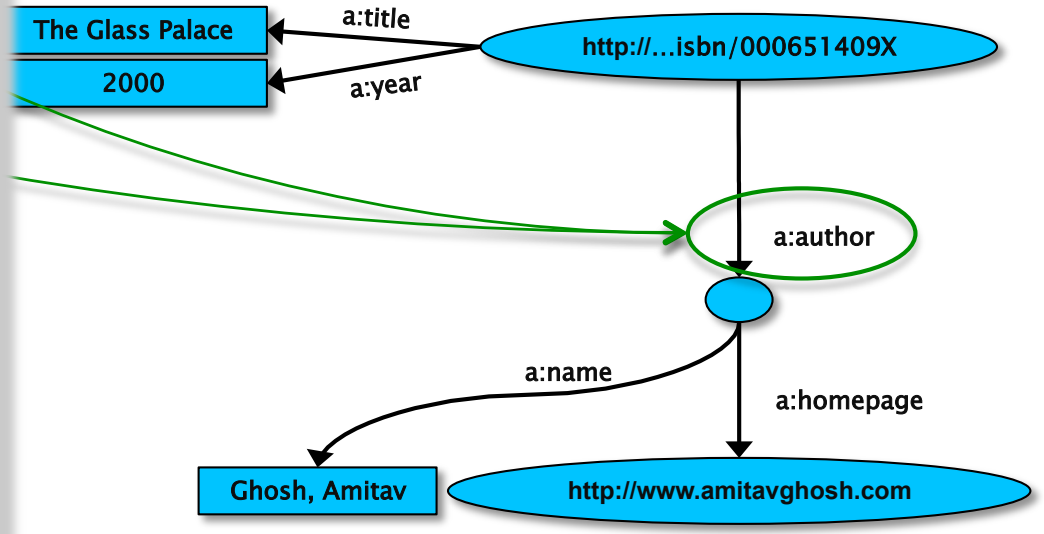


Troisième étape : lier les deux tables entre elles

ISBN	Author	Title	7
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

```
:B_Table a rr:TriplesMap ;
  ...
  rr:refPredicateObjectMap [
    rr:refPredicateMap [
      rr:predicate a:author
    ];
    rr:refObjectMap [
      rr:parentTriplesMap :P_Table ;
      rr:joinCondition
        "{child}.Author = {parent}.ID"
      ]
    ]
  ].
```



W3C Semantic Web

Autre possibilités offertes par R2RML

- Définir les types des données
- Combinaisons plus complexes pour la définitions des objets (p.ex., pour une colonne, les objets peuvent être définis par une *autre* colonne)

```
:B_Table rdf:type rr:TriplesMap ;
  ...
  rr:predicateObjectMap [
    ...
    rr:objectMap [
      rr:column "Year" ;
      rr:datatype xsd:year
    ]
  ] ;
```

Tables virtuelles

- Nous avions, dans notre exemple :

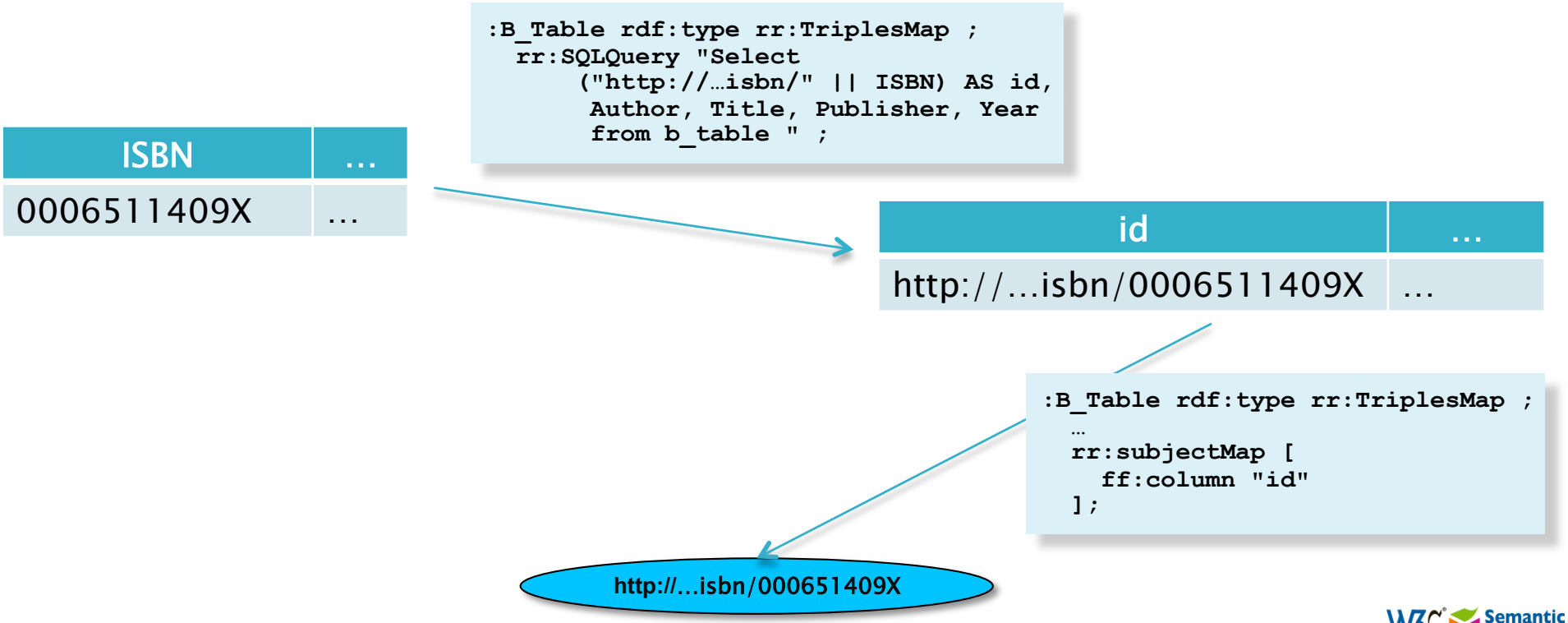
ISBN	…
0006511409X	…

```
:B_Table rdf:type rr:TriplesMap ;
  rr:subjectMap [
    rr:template "http://...isbn/{ISBN}";
  ];
  …
```

W3C Semantic Web

Tables virtuelles

- Nous aurions pu utiliser SQL
  - pour générer un table virtuelle (« logical table »)
  - toutes les autre définitions se réfèrent à cette table
- Ce serait bien trop compliqué pour notre exemple, mais ouvre beaucoup de possibilités en général!



W3C Semantic Web

Export direct

- Un moteur R2RML peut être compliqué
  - contient tout un moteur SQL
  - o.k. quant inclus dans un système important
- L'alternative : générer un graphe RDF simple
  - pas de transformations spéciales
  - objets littéraux sauf pour les liens entre les tables
  - structure des noms fixe
  - l'équivalent d'un « R2RML zéro »
- Le graphe peut être transformé avec RIF, SPARQL, etc.

Retour à notre exemple

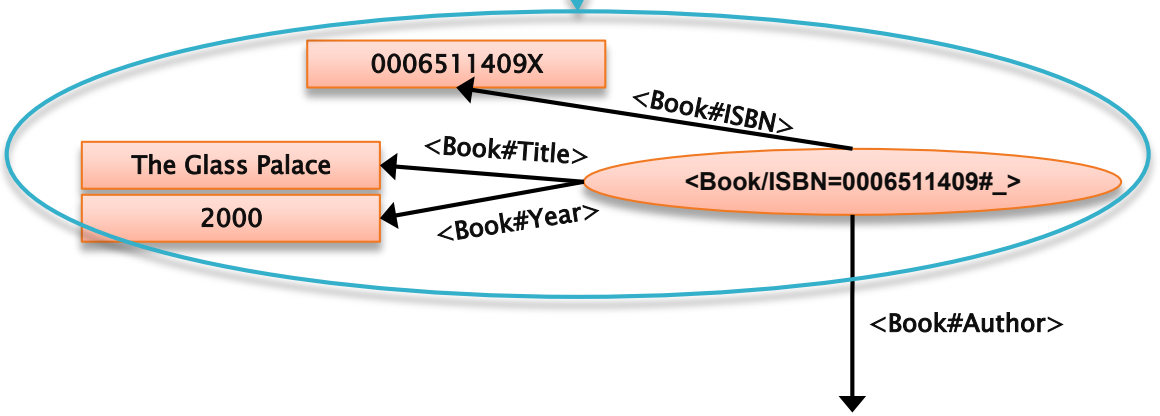
ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

Retour à notre exemple

ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

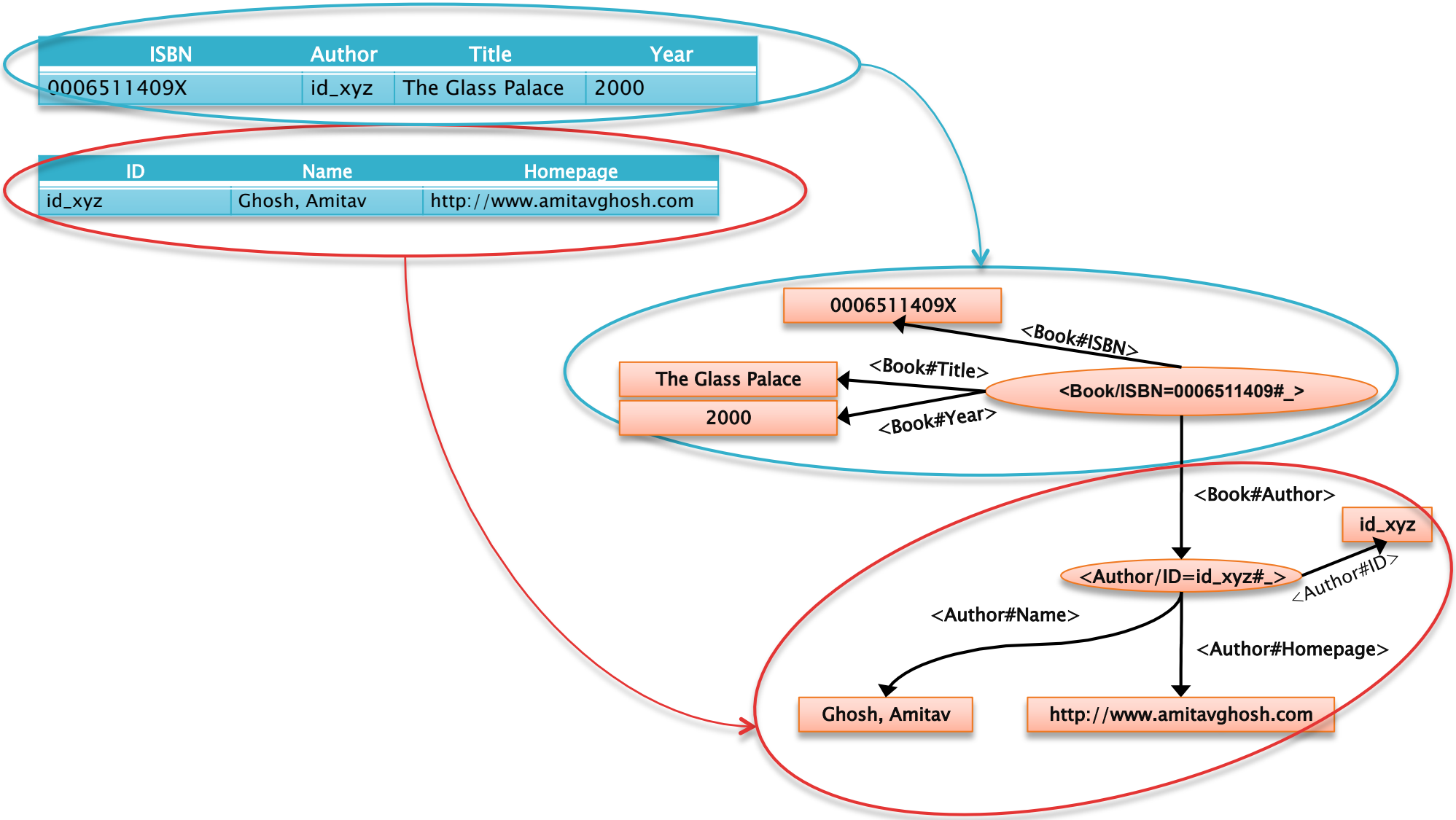
ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com



Retour à notre exemple

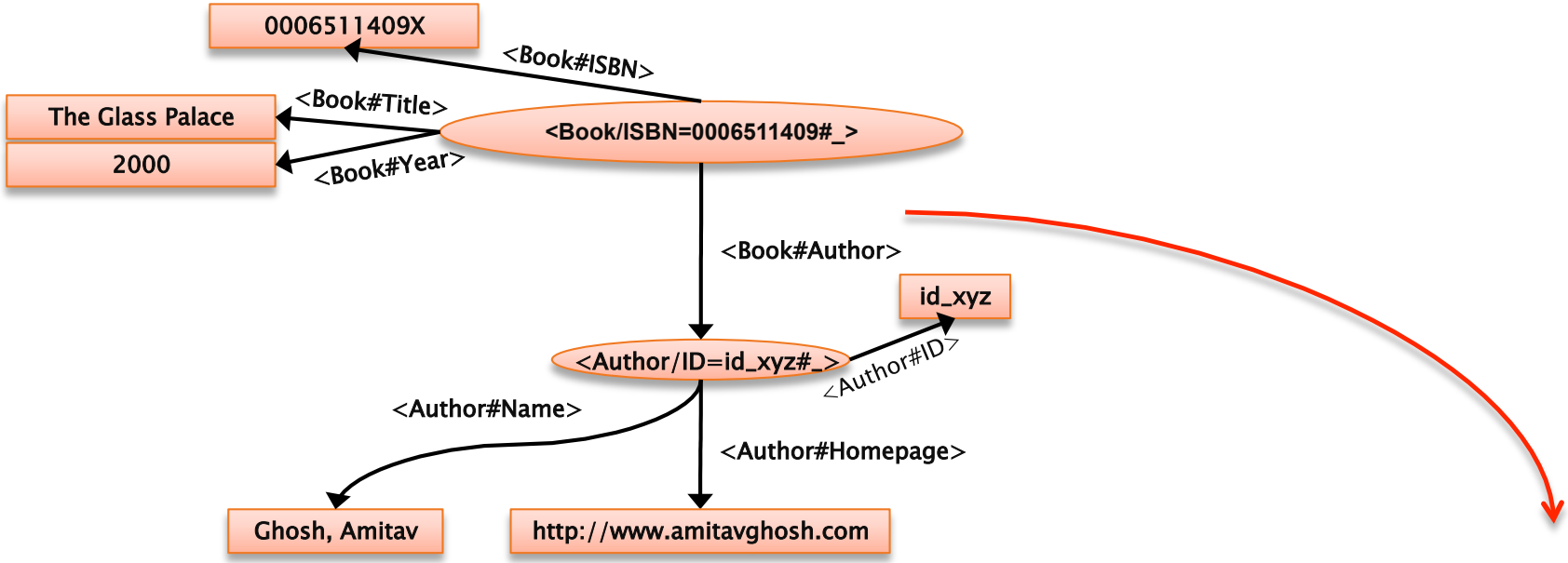
ISBN	Author	Title	Year
0006511409X	id_xyz	The Glass Palace	2000

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

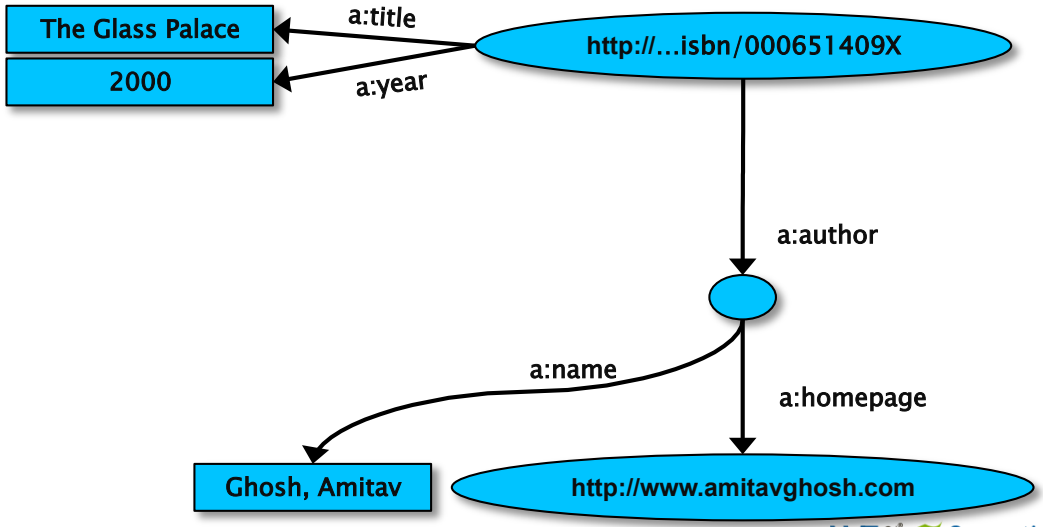


W3C Semantic Web

Le graphe direct doit être transformé

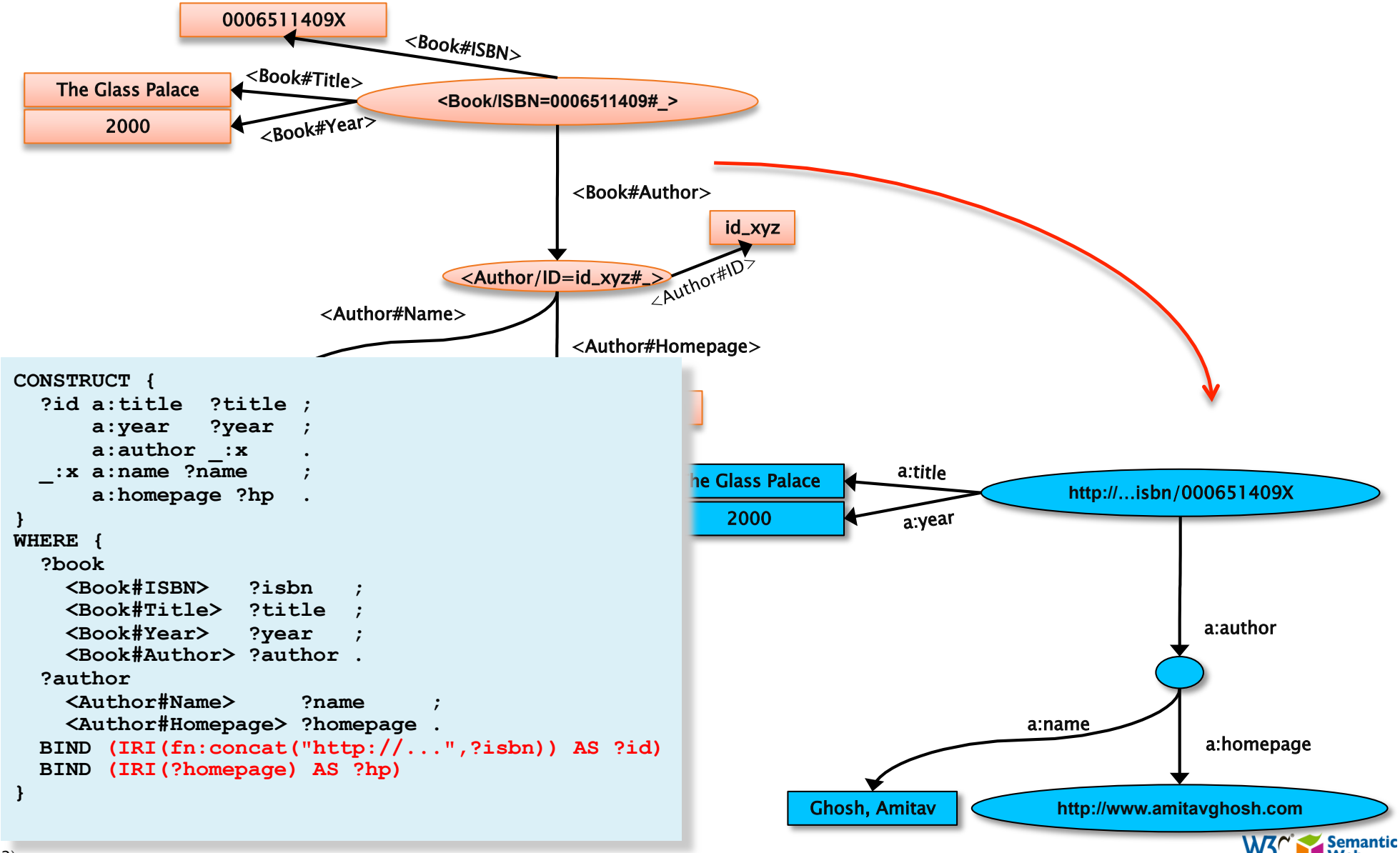


- Les prédicats et les URI-s voulus doivent être créés
- Des littéraux doivent être remplacés par des URI-s

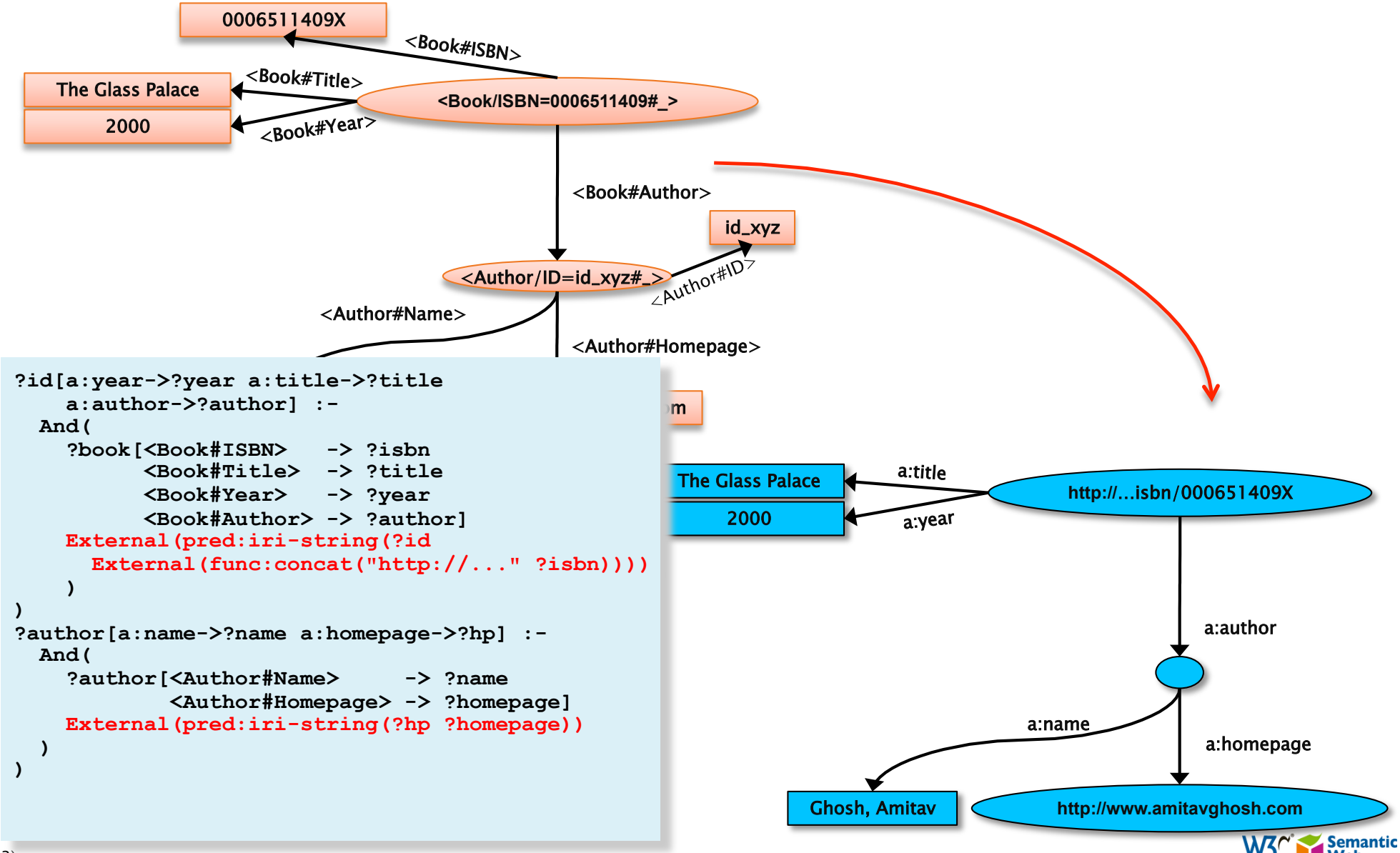


W3C Semantic Web

Transformation avec SPARQL 1.1



Transformation avec RIF Core




```
?id[a:year->?year a:title->?title
    a:author->?author] :-
  And(
    ?book[<Book#ISBN>   -> ?isbn
          <Book#Title>  -> ?title
          <Book#Year>   -> ?year
          <Book#Author> -> ?author]
    External(pred:iri-string(?id
      External(func:concat("http://..." ?isbn))))
    )
)
?author[a:name->?name a:homepage->?hp] :-
  And(
    ?author[<Author#Name>     -> ?name
            <Author#Homepage> -> ?homepage]
    External(pred:iri-string(?hp ?homepage))
  )
)
```

Nouvelle version de RDF

Un peu d'histoire

- Version actuelle de RDF publiée en 2004
  - avec, depuis, un déploiement important
- Mais :
  - il y des fonctionnalités qui manquent
  - il y a des erreurs
  - des communautés importantes n'ont pas adopté RDF
    ex. les développeurs de Web

La question

- Est-ce que nous devons
  - vivre avec ?
  - recommencer à zéro ?
  - faire des changements minimaux ?

L'atelier W3C « RDF Next Step »

- L'atelier a eu lieu en juin 2010
  - 32 soumissions, 28 acceptées pour publication, 18 présentées à l'atelier
  - 2 jours bien remplis à Stanford (merci à NCBO)

Participants de l'atelier

Ce que nous avons fait…

- Essayer de répondre à la question : vivre avec, recommencer, ou changer
  - s'il y a des changements à faire, lesquels et avec quelles priorités
- Créer une liste de travaux à mener, avec priorités

Le sentiment général…

- Ok de regarder qq. problèmes de plus près
- Mais nous devons être extrêmement vigilant à ne pas envoyer de mauvais signaux à ceux qui veulent adopter la technologie
- limiter au maximum les changements

La liste informelle produite par l'atelier

Proposed Work Item	WG must do it	WG should/may do it	WG must not do it
Standardize Model for Graph Identification	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✔✔✔	
Modify Semantics to Support Graph Identification	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✔✔	
Switch to Improved Inference Rules	✔✔✔✔✔✔✔✔✔✔	✔✔✔✔✔✔✔✔	
Apply Fixes to Known Spec Errors	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔		
Standardize a JSON RDF Syntax	✔✔✔✔✔✔✔✔	✔✔✔✔✔✔✔✔✔✔✔✔	
Make Turtle a W3C Standard	✔✔✔✔✔✔✔✔✔✔✔✔	✔✔✔	
Add Graphs to Turtle	✔✔✔✔✔✔✔✔✔✔	✔✔✔✔✔	
Add Graphs to RDF/XML	✔✔✔✔✔	✔✔✔✔✔✔✔✔✔✔✔✔✔	✖✖
Revise Blank Node Semantics	✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	
Specifiy Linked Data Style of RDF	✔✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✖
Weakly Deprecate some RDF/XML Features	✔✔✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	
Define alternatives to owl:sameAs	✔✔✔✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✖✖✖
Weakly Deprecate some Data Model Feature	✔✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✖
Namespace Profiles		✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	
Weakly Deprecate some RDF Semantics Features		✔✔✔✔✔✔✔✔✔✔✔✔✔✔	
Have Explicit Support for Annotations		✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✖✖✖
Align RDF Semantics with SPARQL	✔	✔✔✔✔✔✔✔✔✔✔✔✔✔✔	✖✖✖✖
Improve rdf:List Support in RDF/XML		✔✔✔✔✔✔✔✔✔✔✔✔	✖✖✖✖
Explain How to Determine What a URI Means		✔✔✔✔✔✔✔✔✔✔	✖✖✖✖✖✖
Allow Literals as Subjects		✔✔✔✔✔✔✔	✖✖✖✖✖✖✖✖✖✖✖✖✖

W3C Semantic Web

Étapes suivantes

- Rapport publique de l'atelier :
  - http://www.w3.org/2009/12/rdf-ws/Report.html
- L'équipe du W3C commence à travailler sur la charte du groupe
- … mais avec sentiment qu'une consultation plus large serait nécessaire
- Un questionnaire voit le jour en août 2010
  - http://www.w3.org/2002/09/wbs/1/rdf-2010/results
- Et, naturellement, des discussions sur des forums diverses
- Charte envoyée aux membres de W3C en décembre 2010

Quelques détails de la charte

Avant d'entrer dans les détails…

- Quelle est l'attitude à prendre avec les graphes existants ?
- Le charte demande que :
  - tout graphe RDF valable reste valable
  - toute déduction à base de RDF(S) reste valable

Les questions évidentes

- Il y a des erreurs :
  - relations exactes v.à.v. des IRI-s
  - références exactes aux différentes versions de XML
  - il y a une erreur mathématique dans le « Entailment Lemma »

Identification des graphes

- Connu sous les noms de « named graph », « quoted graphs », « knowledge bases », etc.
- Priorité absolue pour toute la communauté
  - SPARQL a déjà une notion de graphe
  - il y a déjà une utilisation répandue
- Mais la sémantique n’est pas claire
  - parlons-nous d’une collection immuable de triplets ?

Sérialisation en Turtle

- L’autre demande d’une grande priorité
- Nous avons :
  - une « team submission » comme spécification
  - SPARQL Query Language
  - N3
- Mais il n’y a pas de standard

Sérialisation en JSON

- Semble être essentielle pour des applications Web où Javascript domine
- La syntaxe ne sera pas *nécessairement* complète
  - pas de blank nodes mais uniquement une syntaxe pour des nœuds skolemisés ?
  - cacher la différence entre des littéraux et les URI ?
  - etc.

Sérialisation en JSON

- Nous ne partons pas de rien
  - RDF/JSON de Ian Davis
  - JSON-LD de Manu Sporny et Mark Birbeck
  - les idées de Sandro Hawke

Réconcilier des documents sur la sémantique

- Un nombre de documents contenants des éléments de sémantique ont été publiés depuis 2004
- Leur intégration logique serait rassemblée dans un même document RDF
- Il ne s'agit pas de nouvelles définitions, mais de rassembler, éventuellement, ce qui existe déjà

Réconcilier des documents sur la sémantique : exemples

- rdf:plainLiteral : littéral simple avec langage, mais comme un type de donnée
  - nécessaire pour OWL 2 et RIF
  - logiquement, le type devrait être cité dans RDFS
- ponts entre les URI et les littéraux de POWDER :
  - **`wdrs:matchesregex`** et **`wdrs:notmatchesregex`** relient les ressources URI et leur représentations en temps que littéraux
  - une extension de RDFS

Réconcilier des documents sur la sémantique : exemples

- SPARQL 1.1 définit une version “finie” de RDF(S)
  - fait partie de la définition des inférences pour SPARQL 1.1
  - un « sous-ensemble sémantique » de RDFS, réalisable par un moteur de règles
    - une variante de la sémantique de « ter Horst »

Beaucoup d'autres choses…

- Et beaucoup d'autres choses dont je ne vous ai pas parlé:
  - provenance, données liées, données gouvernementales, etc.
- Il y du travail pour tout le monde!
- Essayez de
  - convaincre votre employeur de rejoindre à W3C
  - … et, après, rejoignez un des groupes de travails !

Merci pour votre attention !

Ces slides sont aussi sur le Web :

http://www.w3.org/2011/Talks/0117-Paris-IH/