W3C Current Activities

Dagstuhl Perspective Workshop: Semantic Web Reflections and Future Directions

Dagstuhl, Germany, June 30, 2009

Ivan Herman
ivan@w3.org

- Lots of things are happening at W3C:
  - technology work
    - POWDER, OWL 2, RIF, SPARQL, RDFa, SKOS, media (primarily video) annotations, Linked Open Data community
  - thematic Interest Groups (acting or planned)
    - Health Care and Life Sciences, XBRL, eGovernment
  - incubator groups
    - Semantic Sensor Webs, Social Spaces, Product Modelling

Linked Open Data Community

- It has a strange relationship to W3C:
  - was started as a W3C SWEO “community project”
  - SWEO is no longer active, LOD very much is
  - In fact, LOD lives its own life, uses some of W3C’s infrastructure, but it is not a “formal” part of W3C
    - I hope I can say that the informal ties are very strong, though…

Already done…

- RDFa
  - a way of adding attributes to XHTML and yielding RDF data
  - begins to be well deployed: Slideshare, Best Buy, London Gazette, various UK governmental sites, thesauri published by LOC, …
  - Yahoo! has adopted as part of SearchMonkey
  - Google has adopted as part of Rich Sniplets
  - next version of Drupal will have RDFa “built-in”

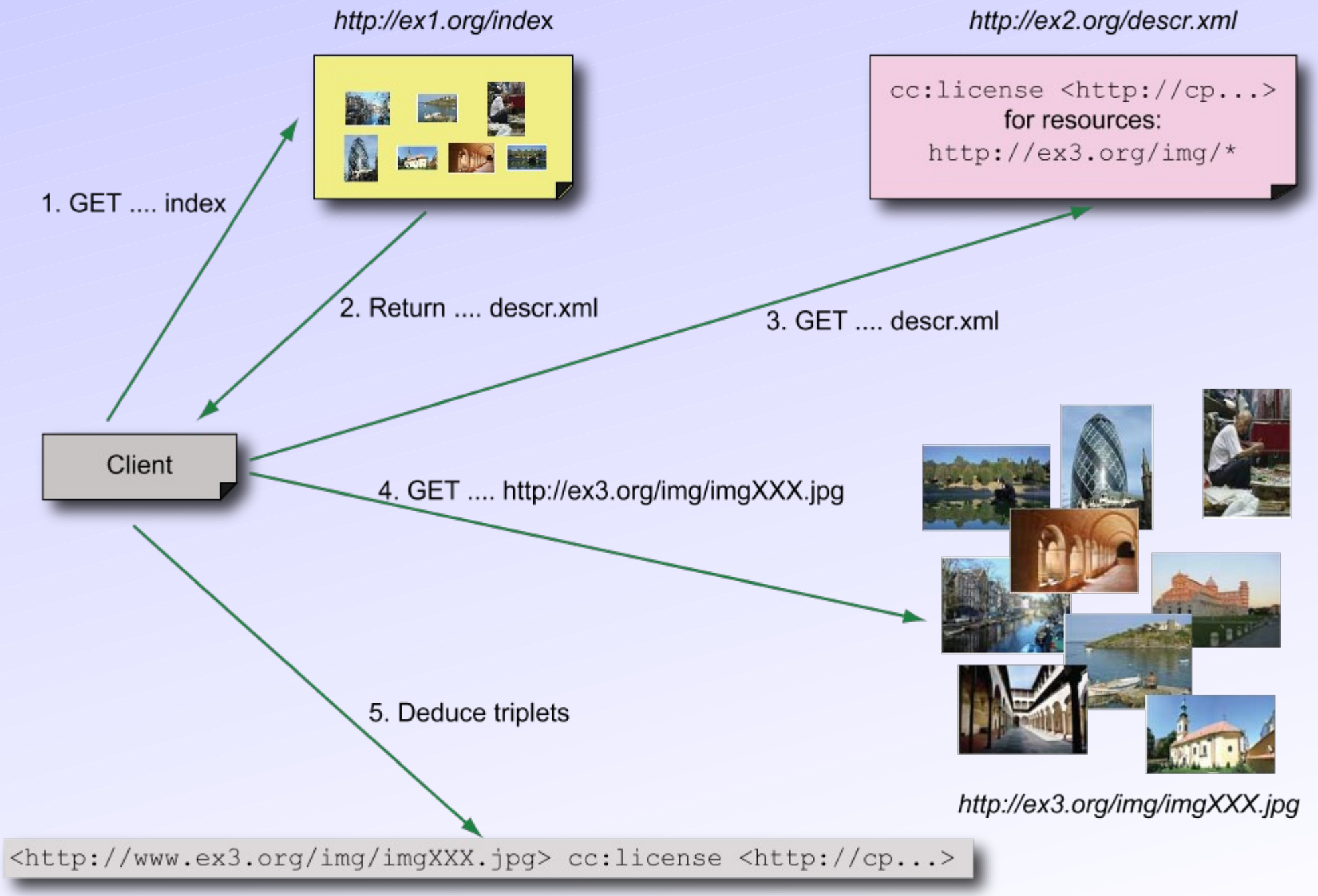
… but there are issues raised around RDFa

- Who defines the vocabularies, where are they stored, how are they managed, archived, found
  - this of course is not specific by RDFa, but the question has been come to the fore used to its success
- RDFa is currently defined for XHTML1.1, what about HTML5
- W3C may set up a separate RDFa IG
  - outreach, tutorials, use cases, etc
  - technical directions (HTML5, general XML dialects, etc)

POWDER (Protocol for Web Description Resources)

- Lets you define predicates that can be automatically assigned to a *set* of resources

POWDER scenario: copyright for photos



The technical details…

- The “description resource” is an XML file:

```
<powder xmlns="http://www.w3.org/2007/05/powder#"
        xmlns:cc="http://creativecommons.org/ns#">
  <attribution>
    <issuedby src="http://www.ivan-herman.net/me"/>
  </attribution>
  <dr>
    <iriset>
      <includehosts>www.ex2.org</includehost>
      <includepathstartswith>/img/</includepathstartswith>
    </iriset>
    <descriptorset>
      <cc:license rdf:resource="http://cp:..."/>
    </descriptorset>
  </dr>
```

POWDER Service

- Online POWDER service can also be set up:
  - a Web service with
    - submit a URI and a resource description file
    - return the RDF statements for that URI
  - such service should be set up, eg, at W3C

POWDER Status

- Proposed Recommendation since June 5

SKOS
(Simple Knowledge Organization System)

- Represent and share classifications, glossaries, thesauri, etc
- Define classes and properties to add those structures to an RDF universe
  - allow for a quick port of this traditional data, combine it with other data

Typical example: LC Subject Headings

Authorities & Vocabularies (Library of Congress): Historical fiction - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://id.loc.gov/authorities/sh85061165

LIBRARY OF CONGRESS | ASK A LIBRARIAN | DIGITAL COLLECTIONS | LIBRARY CATALOGS

The Library of Congress > Authorities & Vocabularies > Historical fiction

Authorities & Vocabularies

Return

Search

Enter search terms... GO

Details | Visualize

Historical fiction

URI: <http://id.loc.gov/authorities/sh85061165#concept>

Type: Topical Term

Broader Terms:

- Fiction

Done

Typical example: LC Subject Headings

Authorities & Vocabularies (Library of Congress): Historical fiction - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://id.loc.gov/authorities/sh85061165

LIBRARY OF CONGRESS | ASK A LIBRARIAN | DIGITAL COLLECTIONS | LIBRARY CATALOGS

The Library of Congress > Authorities & Vocabularies > Historical fiction

Authorities & Vocabularies

Return

Search

Enter search terms... GO

Details | Visualize

skos:prefLabel

Historical fiction

URI: <http://id.loc.gov/authorities/sh85061165#concept>

skos:broader

Type: Topical Term

Broader Terms:

- Fiction

Done

W3C Semantic Web

Typical example: LC Subject Headings

Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/DOCUME~1/IVANHE~1/LOCALS~1/Temp/extract-4

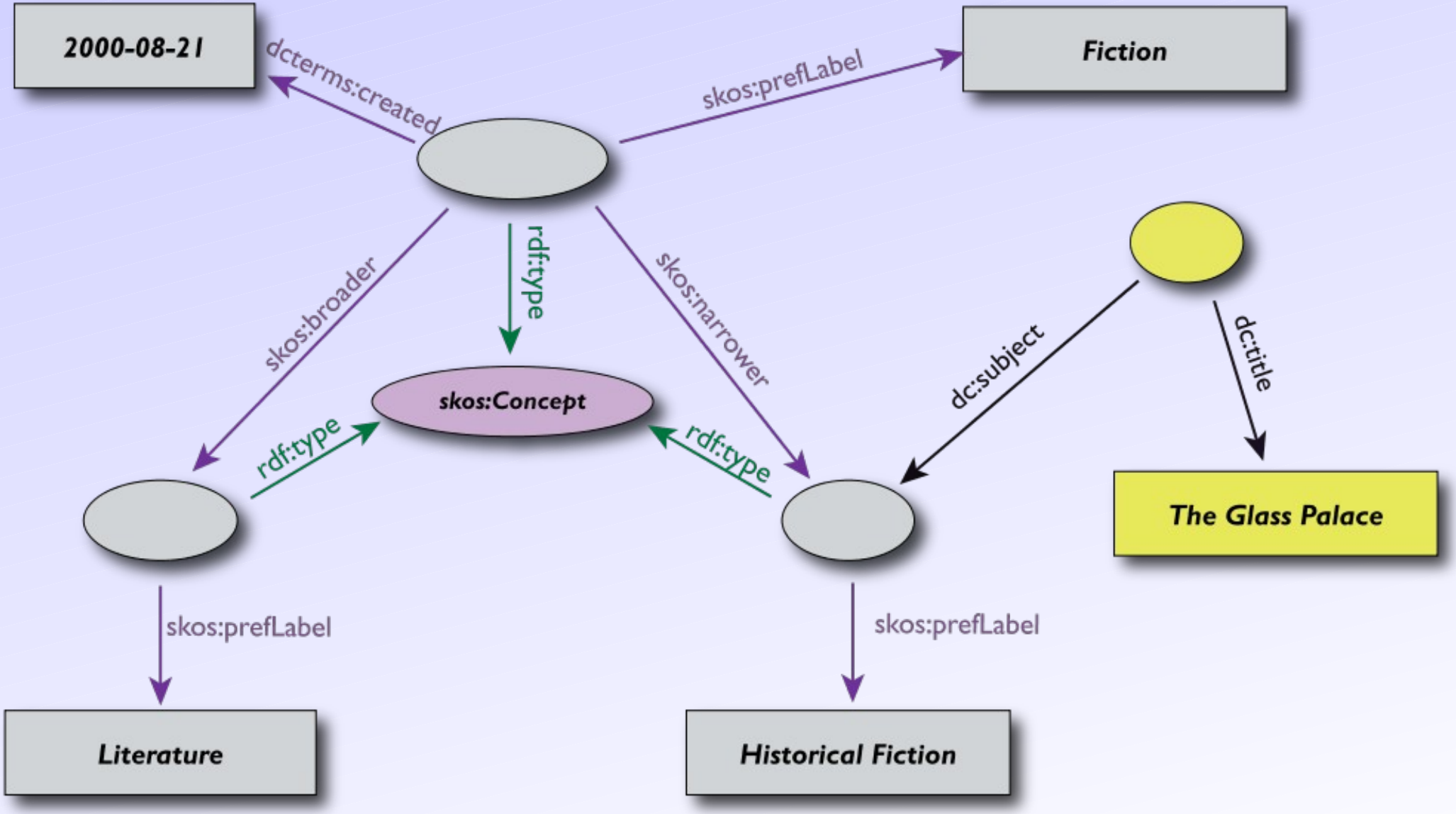
```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xhv: <http://www.w3.org/1999/xhtml/vocab#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xs: <http://www.w3.org/2001/XMLSchema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .


<http://id.loc.gov/authorities/sh85061165> xhv:alternate
         <http://id.loc.gov/authorities/feed/>,
         <http://id.loc.gov/authorities/sh85061165.json>,
         <http://id.loc.gov/authorities/sh85061165.nt>,
         <http://id.loc.gov/authorities/sh85061165.rdf> ;
     xhv:icon <http://www.loc.gov/favicon.ico> ;
     xhv:stylesheet <http://id.loc.gov/static/css/subject_headings_print.css>, <http://id.l

<http://id.loc.gov/authorities/sh85061165#concept> a skos:Concept ;
     dcterms:created "2000-08-21T00:00:00-04:00"^^xsd:dateTime ;
     dcterms:modified "2000-10-04T10:47:15-04:00"^^xsd:dateTime ;
     dcterms:source "GSAFD, 2000 (Historical fiction. UF Fiction, Historical; Historical no
     skos:broader <http://id.loc.gov/authorities/sh85048050#concept> ;
     skos:closeMatch <http://stitch.cs.vu.nl/vocabularies/rameau/ark:/12148/cb119808101> ;
     skos:inScheme <http://id.loc.gov/authorities#conceptScheme>, <http://id.loc.gov/author
     skos:prefLabel "Historical fiction"@en .
```

Done

Using the LCSH terms…



SKOS and OWL

- SKOS is geared towards specific (though large) use cases, like
  - taxonomies, glossaries, …
  - annotations of complex structures (including ontologies)
- SKOS is a based on a very simple usage of OWL
  - using some simple OWL Full constructions
  - the emphasis is on *organization* and not on logical inferences
- "OWL is a Harley-Davison, SKOS is a mountain bike" — (Tom Baker, co-chair of the relevant WG)

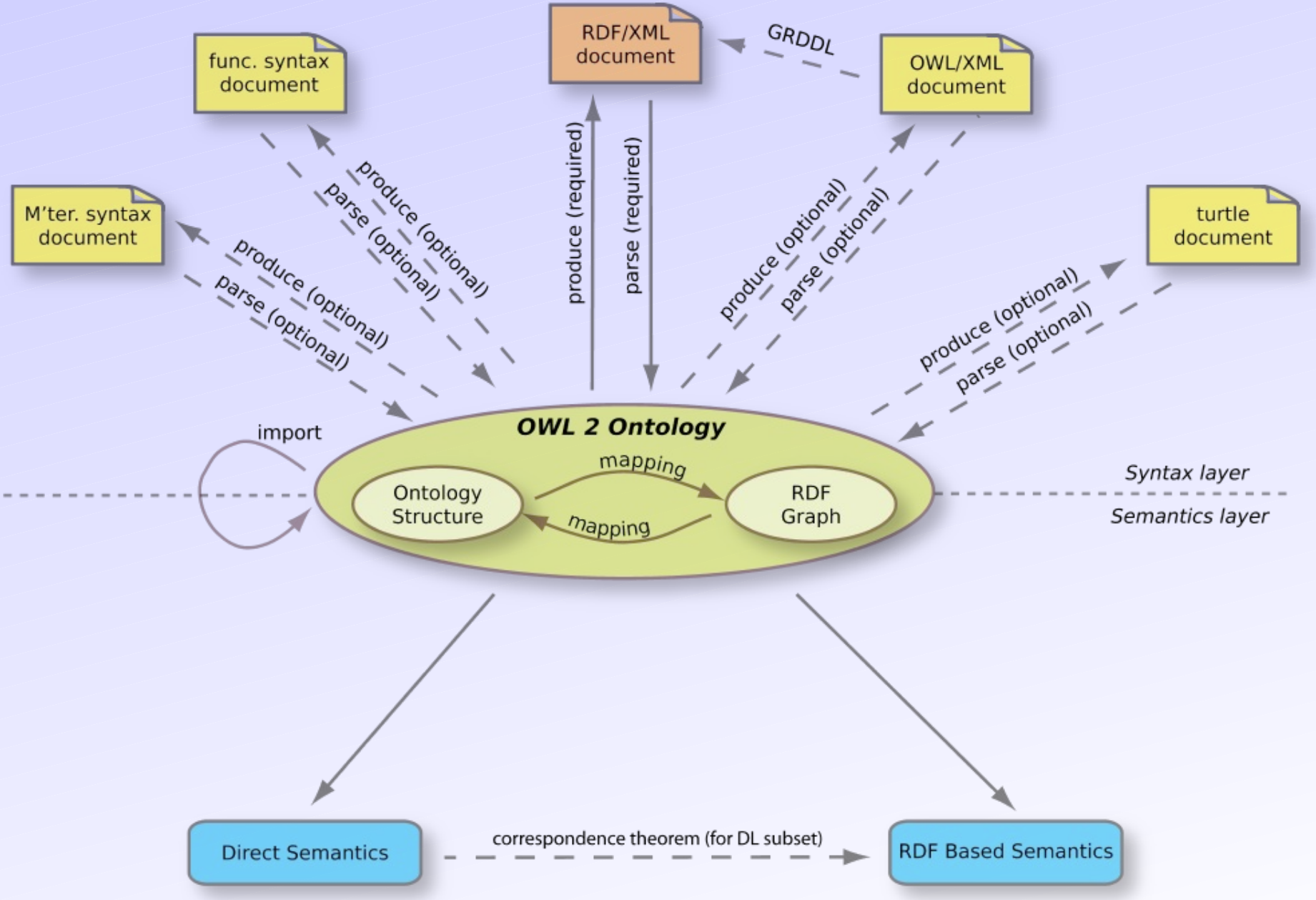
SKOS Status

- Proposed Recommendation since June 15

OWL Working Group

- A new Working Group works on the revision of OWL (a.ka. OWL 2)
- The goal of the group:
  1. add a few extensions to current OWL that are useful, and is known to be implementable
     - many things happened in research since 2004
  2. define “profiles” of OWL that are:
     - smaller, easier to implement and deploy
     - cover important application areas and are easily understandable to non-expert users

The overall structure has not changed



Some new features in OWL 2

- Syntactic sugars
  - eg, disjoint union of classes
- New constructs for properties
  - property chains, reflexive properties
- Richer annotations
- Extended datatype facilities
  - define a numerical interval as an OWL Datatype class
- Profiles

New constructs for properties

- Disjoint properties
- Reflexive, irreflexive, asymmetric properties
- Self restriction (a.k.a. local reflexivity):
  - collect of individual for which the `(x p x)` holds
- Qualified cardinality restrictions
  - eg, “class instance must have two black cats”

New constructs for properties: chains

- Properties, when applied one after the other, may be subsumed by yet another one:
  - “if a person «P» was born in city «A» and «A» is in country «B» then «P» was born in country «B»”
  - more formally:

```
ex:born_in_country owl:propertyChainAxiom
    (ex:born_in_city ex:city_in_country).
```

- More than two constituents can be used
- There are some restrictions for DL to avoid “circular” specifications

Keys

"if two persons have the same email and the same homepages then they are identical"

- Identification is based (in this example) on the identical values of *two* properties
- The rule applies to persons only

Extended datatypes facilities

- OWL 1 just takes over RDF datatypes, but that is it
- But we might want datatypes “restrictions” in a more complex way; eg, numeric intervals
  - “I am interested in a price range between €5 and €15”
- In OWL 1, one has to use XML Schema datatypes
  - but it is very complex, users and reasoners have to understand a whole different syntax

OWL 2 datatype facets

- For each datatype one may have restrictions “facets” (min, max, length, etc)
- New datatypes can be defined as *datatype ranges*
- These can be used via the usual restriction mechanisms

OWL 2 defines “profiles”

- Further restrictions on how terms can be used and what inferences can be expected
- The semantic approaches are identical, but restrictions may ensure even more manageable implementations

OWL 2 profiles

- Classification and instance queries in polynomial time: *OWL-EL*
- Implementable on top of conventional relational database engines: *OWL-QL*
- Implementable on top of traditional rule engines: *OWL-RL*

An example: OWL-RL

- Goal: to be implementable through rule engines
- Usage follows a similar approach to RDFS:
  - merge the ontology and the instance data into a big RDF graph
  - use a rule engine to add new triples (as long as it is possible)
  - then, for example, use SPARQL to query the resulting (expanded) graph
- This application model is very important for RDF based applications

Example: OWL-QL

- The RL “model” has a downside: a large number of extra triples are added to the graph before query
  - big database vendors have means to handle that
  - but simpler implementations may have a problem there…
- QL is different: map a query directly on SQL
  - ie, the data can be queried without “touching” the dataset

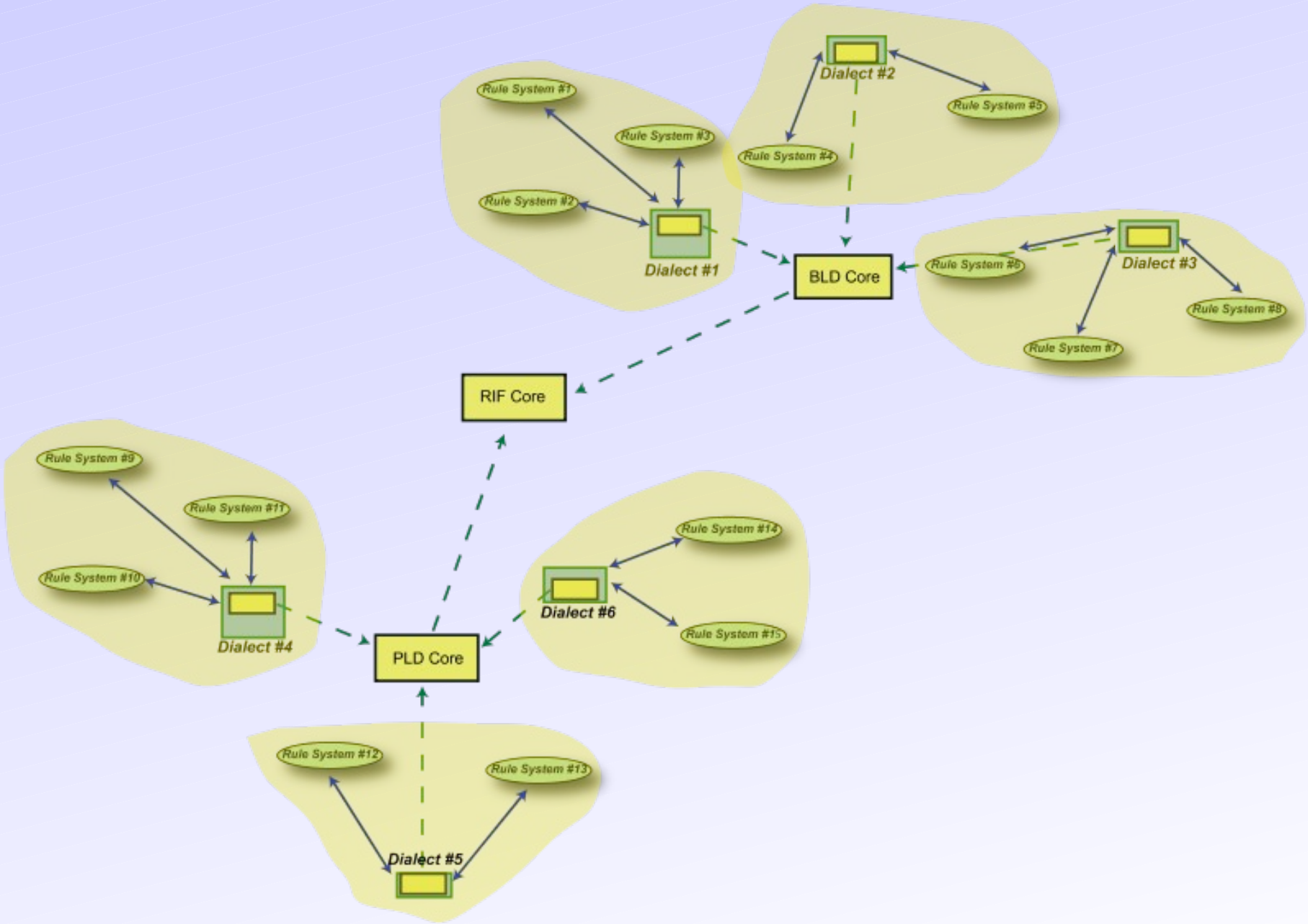
OWL 2 Status

- Candidate Recommendation since June 15

RIF

- Has a long (and tumultuous…) history
- Started with a wish to exchange rule sets among any rule systems
- This has proven to be unrealistic
  - the distance between production rule systems and “logical” rules is too large
- The model is a hierarchy of “cores” with dialects

Hierarchy of cores



Schematically…

- The “BLD (Basic Logic Dialect)” is of the form:
  - “**`if condition true then this is true`**”
  - conditions may include functions, hierarchies
- The “PLD (Production Logic Dialect)” is of the form:
  - “**`if condition is true then do something`**”
- The “Core”: shared subset of major languages
  - technically: positive Horn without function terms, with some simple datatypes

Some RIF Core/BLD features

- RIF BLD includes some extra features
  - built-in datatypes and predicates
  - notion of “local names”, a bit like RDF’s blank nodes
  - a “frame-based” syntax (beyond predicates and functions):
    - `p[prop1->v1 prop2->v2]`
  - built-in abstractions for classes, subclassing, and typing:
    - `m # C`, `C1 ## C2`
- RIF Core/BLD’s semantics follows the “usual” approach in logic

RIF Syntaxes

- RIF has a normative XML syntax
  - *the* syntax used to exchange rules among engines
- There is also a non-normative “presentation syntax”

RIF BLD example

```
Document(
  Prefix(cpt http://example.com/concepts#)
  Prefix(ppl http://example.com/people#)
  Prefix(bks http://example.com/books#)

  Group
  (
    Forall ?Buyer ?Item ?Seller (
        cpt:buy(?Buyer ?Item ?Seller):- cpt:sell(?Seller ?Item ?Buyer)
    )
    cpt:sell(ppl:John bks:LeRif ppl:Mary)
  )
)
```

infers the following relationship:

```
cpt:buy(ppl:Mary bks:LeRif ppl:John)
```

What about RDF(S), OWL, and RIF?

- Typical scenario: applications exchange rules that refer to RDF data
- To make that work:
  - RDF facts/triples have to be representable in BLD
  - harmonization on the concepts is necessary (eg, classes)
  - the formal semantics of the two worlds should also be aligned
- There is a separate document that brings these together

What about RDF(S), OWL, and BLD?

- Triples can be expressed in BLD using the frame syntax:
  - `(s p o)` is written as `s[p->o]`
  - subclassing and typing of BLD are defined to be equivalent to their RDFS counterpart
  - the datatypes are identical to OWL 2

An example

We describe/exchange the rules:

```
Group
(
  Forall ?Buyer ?Item ?Seller (
      ?Buyer[cpt:buy->?Item cpt:from->?Seller] :-
          ?Seller[cpt:sell->?Item cpt:to->?Buyer]
  )
)
```

An example

We describe/exchange the rules:

```
Group
(
  Forall ?Buyer ?Item ?Seller (
      ?Buyer[cpt:buy->?Item cpt:from->?Seller] :-
          ?Seller[cpt:sell->?Item cpt:to->?Buyer]
  )
)
```

We then *import* the RDF data

```
ppl:Mary
   cpt:sell bks:LeRif;
   cpt:to   ppl:John .
```

An example

We describe/exchange the rules:

```
Group
(
  Forall ?Buyer ?Item ?Seller (
      ?Buyer[cpt:buy->?Item cpt:from->?Seller] :-
          ?Seller[cpt:sell->?Item cpt:to->?Buyer]
  )
)
```

We then *import* the RDF data, and infer:

```
ppl:Mary
   cpt:sell bks:LeRif;
   cpt:to   ppl:John .

ppl:John
   cpt:buy bks:LeRif;
   cpt:from ppl:Mary .
```

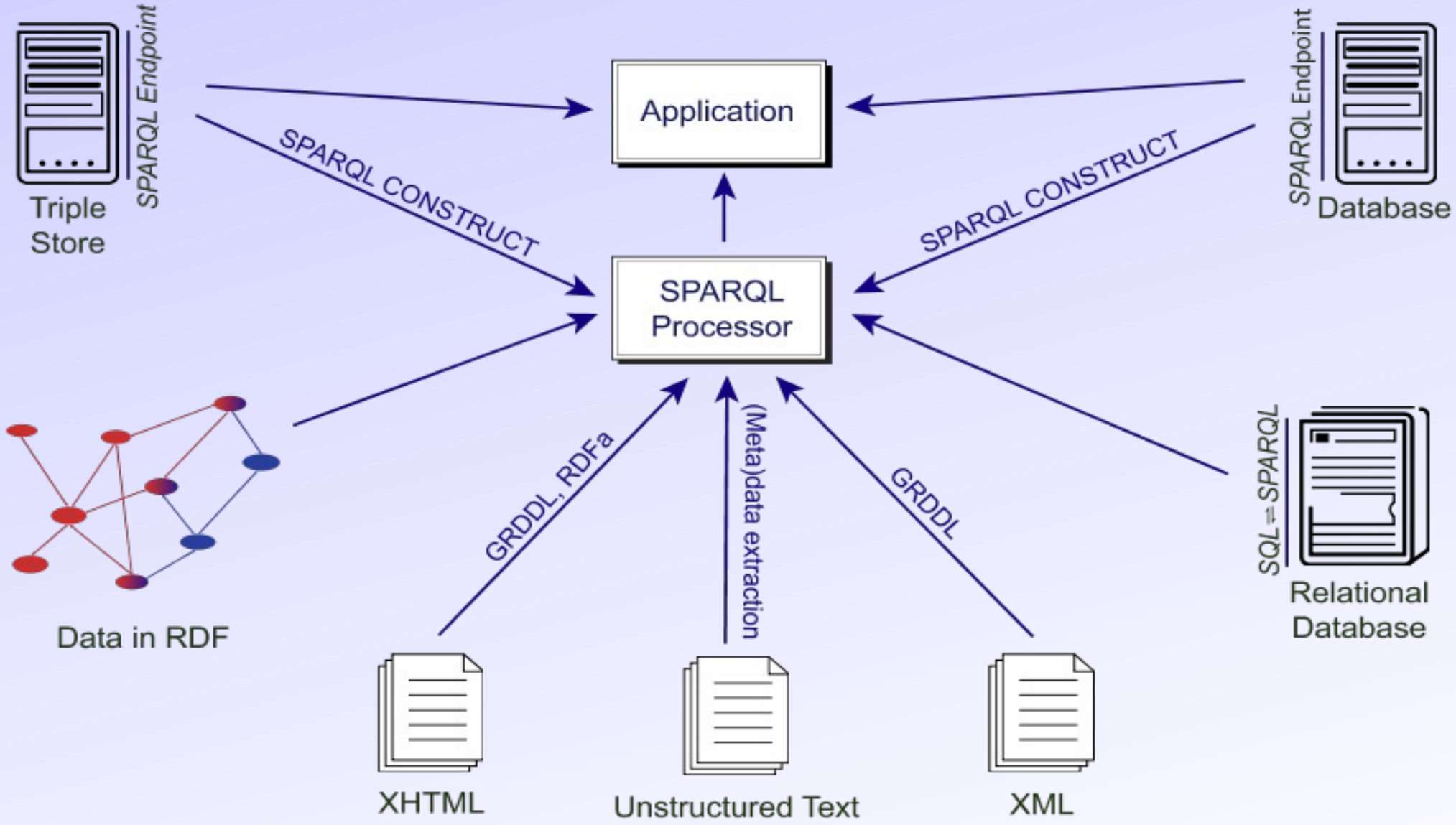
Issues around RIF

- An attempt to bring together various communities
- The more “down-to-Earth” aspect to provide a simple rule level for RDF is not really central in RIF
- We should work on a more RDF-centric dialect? sublanguage? syntax? once RIF is published
  - unclear what the exact format will be

RIF Status

- 2nd Last Call any day now

SPARQL as a unifying point!



New SPARQL WG: Goals

- To define a small set of extensions to SPARQL
- No complex change, backward compatibility
- Listen to user and implementation experiences of the past few years
- Group started in February 2009

Planned features

- Update, ie, ability to change the RDF store
- Service description framework
  - what type of extensions, inference possibilities, etc, are available at the endpoint
- Addition to the query language
  - aggregate functions
  - subqueries
  - negation
  - project expressions

Planned features (tentative syntax examples)

- Aggregate functions and project expressions:

```
SELECT AVG(?age) AS average_age WHERE { .... }
SELECT (?age < 18) AS minor WHERE { ... }
```

- Subqueries:

```
SELECT ?person (SELECT ?n WHERE { ?person foaf:name ?n } LIMIT 1)
WHERE { <http://www.ivan-herman.net/me> foaf:knows ?person. }
```

- Negation:

```
SELECT *
WHERE { ?x :p ?v. UNSAID { ?x :q ?v. } }
```

Planned features (time permitting)

- Definition of “entailment regimes”
  - RDFS, OWL Profiles, RIF
- Property paths
- Commonly used functions (eg, string manipulation)
- Basic control for federated queries
- Additional query language syntax
  - commas in select lists, some operators in filters

How to access a database

- Most data come from relational databases
- Issue: how to “map” a relational database content to RDF
  - different tools exist (Virtuoso’s RDF view, D2RQ, Triplify, R2O, Dartgrid toolkit, Asio, RDBToOnto)
  - the W3C RDB2RDF Incubator Group published a survey:
    - http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf

How to access a database (cont.)

- A new RDB2RDF Working Group is planned
- Goal:
  - “standardize a language for mapping relational data and relational database schemas into RDF and OWL”
  - how to assign public identifiers to database entries
  - group should start in September, watch the news and join!

Where to go from here?

- There are lots R & D issues that the community has already identified
  - distributed/federated SPARQL queries on huge datasets
  - how to automatically interlink large datasets among themselves (eg, two “blobs” in the LOD cloud)
  - how to handle an “ID jungle”, ie, when lots of URI aliases are generated to the same “thing”
  - versioning of, say, vocabularies
  - updating/refreshing the RDF model
  - scaling
  - etc
- Some extra longer term questions

How do we interface to other technologies?

- In the SW we do
  - (Web) data integration on the largest possible scale
  - some level of inferences to discover new relationships among data
- But the same integrated data could/should be exploited by other, eg, numerical means
  - Wolfram's Alpha is an example
- How do we "bind" these two worlds in a standard way?
  - eg, do we have to build in hooks into our various technologies?

Uncertainty

- Even if we concentrate on logics, we use a binary, a.k.a. “black-and-white” logic
- This is obviously not good enough
  - e.g., the LOD community uses owl:sameAs a lot
    - it is, in many cases, wrong in view of pure OWL
    - we would need some sort of a “fuzzy” way of expressing sameness
- There are fuzzy-logic and probabilistic approaches, but how to integrate that into the SW is still a challenge

Provenance, access control, security…

- The first questions asked after a presentation…
  - "I do not want to give away (all) my data to everyone"
  - "How do I know that the originator of a statement is really the person who says he is"
  - etc
- We do *not* have a proper story to tell on these…
- … but it is absolutely essential to have one!
- Personally, I believe this is the most pressing problem to solve
  - playing with the idea of a relevant workshop in 2010…

Thanks for your attention!