Semantic in SimDB

Nicolas Moreau, Norman Gray,
Fabrice Roy, Franck Lepetit, Benjamin Ooghe
LERMA

l'Observatoire de Paris LERMA

Laboratoire d'Étude du Rayonnement et de la Matière en Astrophysique

l'Observatoire de Paris LUTH

Laboratoire Univers et Théories

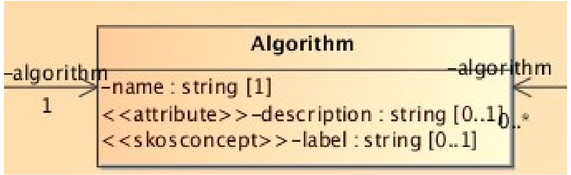
l'Observatoire de Paris Virtual Observatory Paris Data Centre

I Applying semantic concepts in SimDB
II Creating and editing vocabularies
III Concepts identification
IV An example of implementation
V Evolutions

I Applying semantic concepts in SimDB
II Creating and editing vocabularies
III Concepts identification
IV An example of implementation
V Evolutions

Applying semantic concepts in SimDB

- SimDM allows to tag data with semantic concepts to help the user searching DB


-algorithm
1
Algorithm
-algorithm
-name : string [1]
<<attribute>>-description : string [0..1]
0..*
<<skosconcept>>-label : string [0..1]

-Taggable classes are :

Protocol package : Physics, Algorithm, InputParameter, ParameterGroup,
RepresentationObjectType
Experiment package : TargetProcess, TargetObjectType
Object package : Property, PropertyGroup

- There are currently 7 vocabularies :
Algorithms
AstronomicalObjectTypes
JournalKeywords
PhysicalProcesses
Properties
RepresentationObjects
ProductType

- Following Semantic WG recommendations, vocabularies use the Simple Knowledge Organization System (SKOS)

- Normalized by W3C, common data model for sharing and linking knowledge organization systems via the Web

A vocabulary is a list of concepts

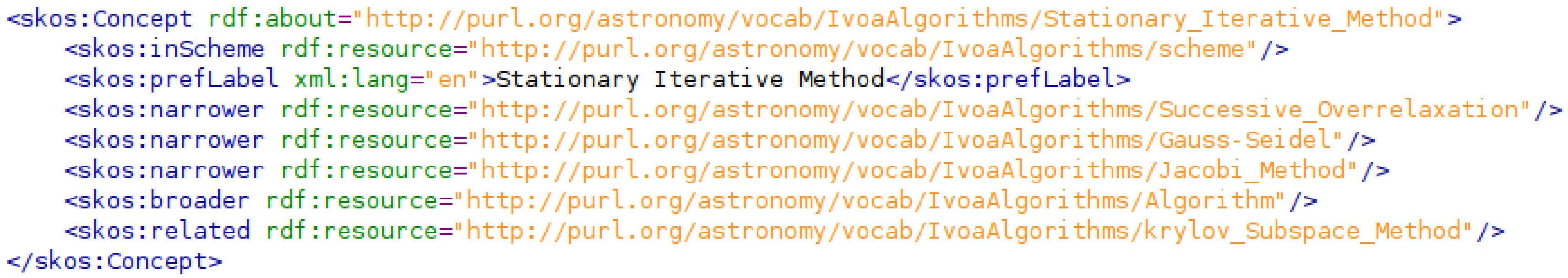
Each concept has :

An identifier (a single URI mainly for use by computers)
A single prefered label in each supported language
Zero or more altLabel (simple synonyms or commonly-used aliases)
A description

Concepts are linked by a limited set of relations :

“broader”, “narrower” or simply “related”

Example :



```
<skos:Concept rdf:about="http://purl.org/astronomy/vocab/IvoaAlgorithms/Stationary_Iterative_Method">
    <skos:inScheme rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/scheme"/>
    <skos:prefLabel xml:lang="en">Stationary Iterative Method</skos:prefLabel>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Successive_Overrelaxation"/>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Gauss-Seidel"/>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Jacobi_Method"/>
    <skos:broader rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Algorithm"/>
    <skos:related rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/krylov_Subspace_Method"/>
</skos:Concept>
```

I Applying semantic concepts in SimDB

II Creating and editing vocabularies

III Concepts identification

IV An example of implementation

V Evolutions

Creating and editing vocabularies : PoolParty

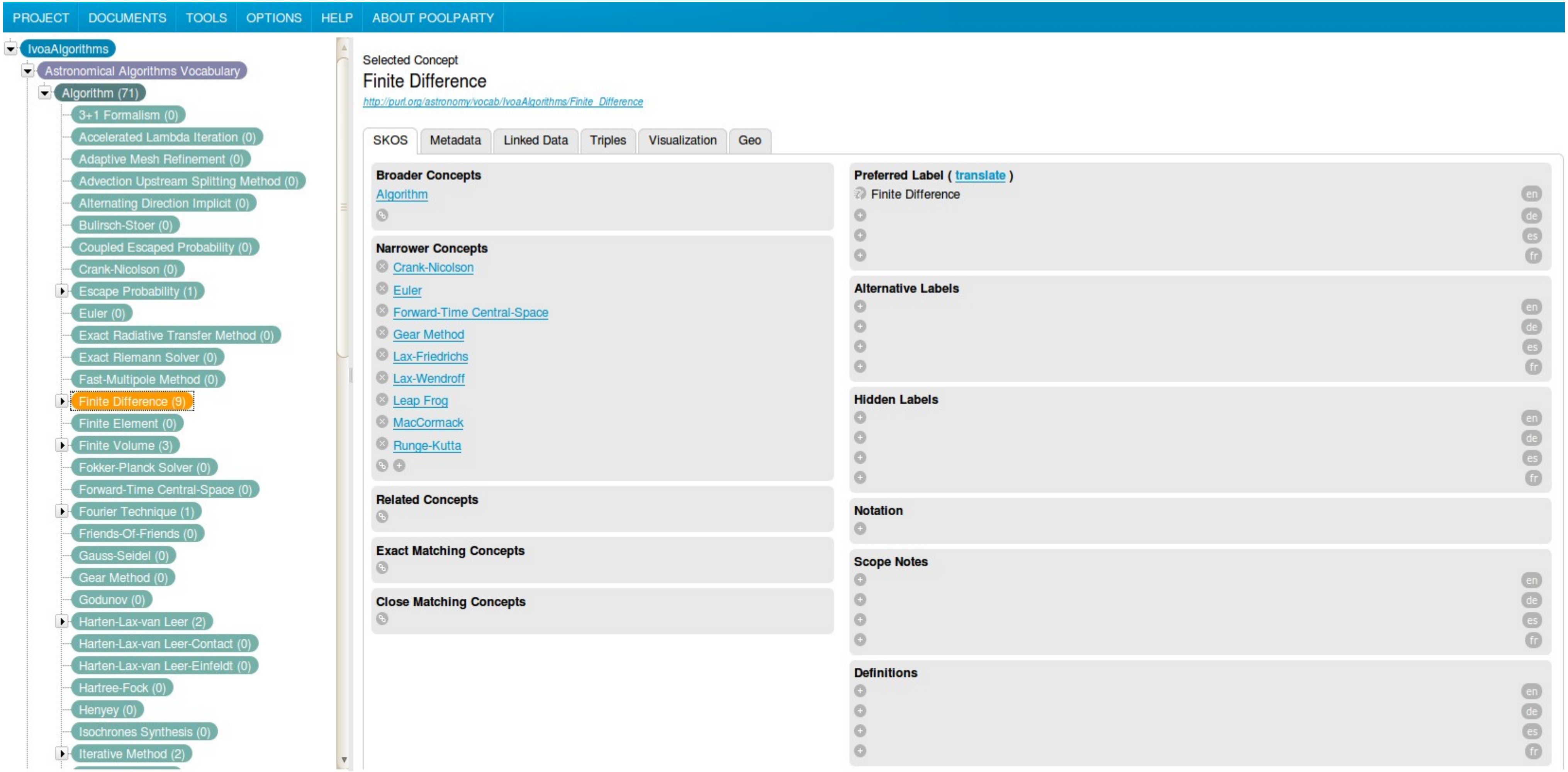
- SKOS vocabularies can be written in XML
- However creation/maintenance is not user friendly

-PoolParty is a commercial web application allowing :

  - creation and edition of vocabularies
  - a wiki frontend for public access
  - a clear graphical interface to browse and edit them in admin mode
  - import/export utilities
  - vocabulary validation tools
  - programmatic access through SPARQL endpoint and HTTP web services :
    getSemanticRelations
    getThesauri
    getConcepts
    ...

- Fortunately we got an academic licence

Poolparty interface



IvoaAlgorithms
- Astronomical Algorithms Vocabulary
  - Algorithm (71)
    - 3+1 Formalism (0)
    - Accelerated Lambda Iteration (0)
    - Adaptive Mesh Refinement (0)
    - Advection Upstream Splitting Method (0)
    - Alternating Direction Implicit (0)
    - Bulirsch-Stoer (0)
    - Coupled Escaped Probability (0)
    - Crank-Nicolson (0)
    - Escape Probability (1)
    - Euler (0)
    - Exact Radiative Transfer Method (0)
    - Exact Riemann Solver (0)
    - Fast-Multipole Method (0)
    - Finite Difference (9)
    - Finite Element (0)
    - Finite Volume (3)
    - Fokker-Planck Solver (0)
    - Forward-Time Central-Space (0)
    - Fourier Technique (1)
    - Friends-Of-Friends (0)
    - Gauss-Seidel (0)
    - Gear Method (0)
    - Godunov (0)
    - Harten-Lax-van Leer (2)
    - Harten-Lax-van Leer-Contact (0)
    - Harten-Lax-van Leer-Einfeldt (0)
    - Hartree-Fock (0)
    - Henyey (0)
    - Isochrones Synthesis (0)
    - Iterative Method (2)

Selected Concept

Finite Difference

http://purl.org/astronomy/vocab/IvoaAlgorithms/Finite_Difference

Broader Concepts

Algorithm

Narrower Concepts

Crank-Nicolson
Euler
Forward-Time Central-Space
Gear Method
Lax-Friedrichs
Lax-Wendroff
Leap Frog
MacCormack
Runge-Kutta

Related Concepts

Exact Matching Concepts

Close Matching Concepts

Preferred Label (translate)

Finite Difference

Alternative Labels

Hidden Labels

Notation

Scope Notes

Definitions

- A Poolparty instance has been created for us
- No access to the server

How do we use it :

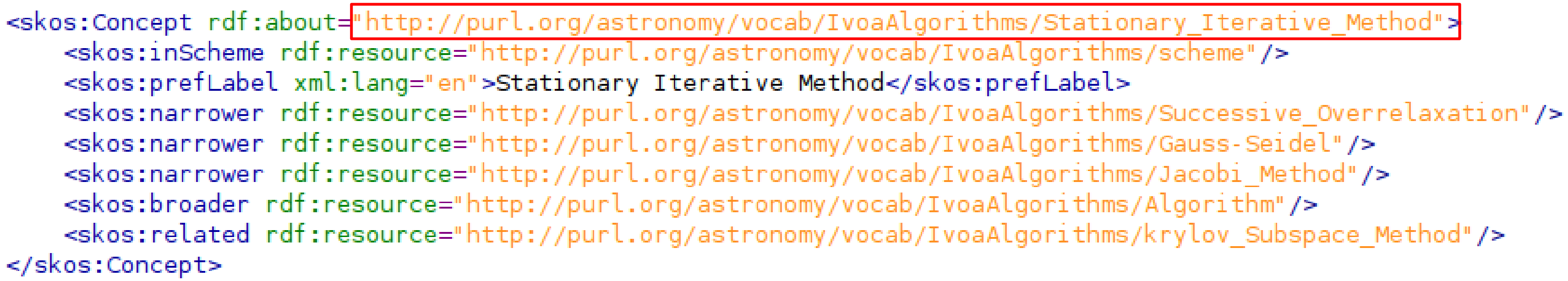
- We have to stay independant from the application
- The aim is to simplify the creation and update of vocabularies
- Eventually they will be exported and stored elsewhere

Demo

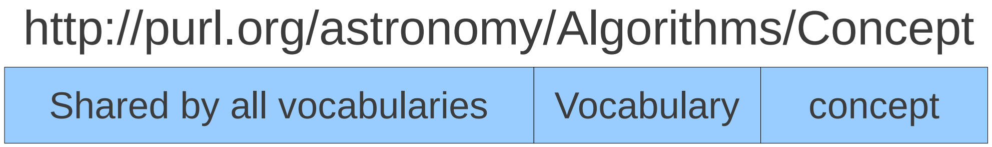
I Applying semantic concepts in SimDB
II Creating and editing vocabularies
III Concepts identification
IV An example of implementation
V Evolutions

Concepts identification : purl

- Each SKOS concept has a unique identifier
- We use it in simDB to assign a concept to an object



```
<skos:Concept rdf:about="http://purl.org/astronomy/vocab/IvoaAlgorithms/Stationary_Iterative_Method">
    <skos:inScheme rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/scheme"/>
    <skos:prefLabel xml:lang="en">Stationary Iterative Method</skos:prefLabel>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Successive_Overrelaxation"/>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Gauss-Seidel"/>
    <skos:narrower rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Jacobi_Method"/>
    <skos:broader rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/Algorithm"/>
    <skos:related rdf:resource="http://purl.org/astronomy/vocab/IvoaAlgorithms/krylov_Subspace_Method"/>
</skos:Concept>
```



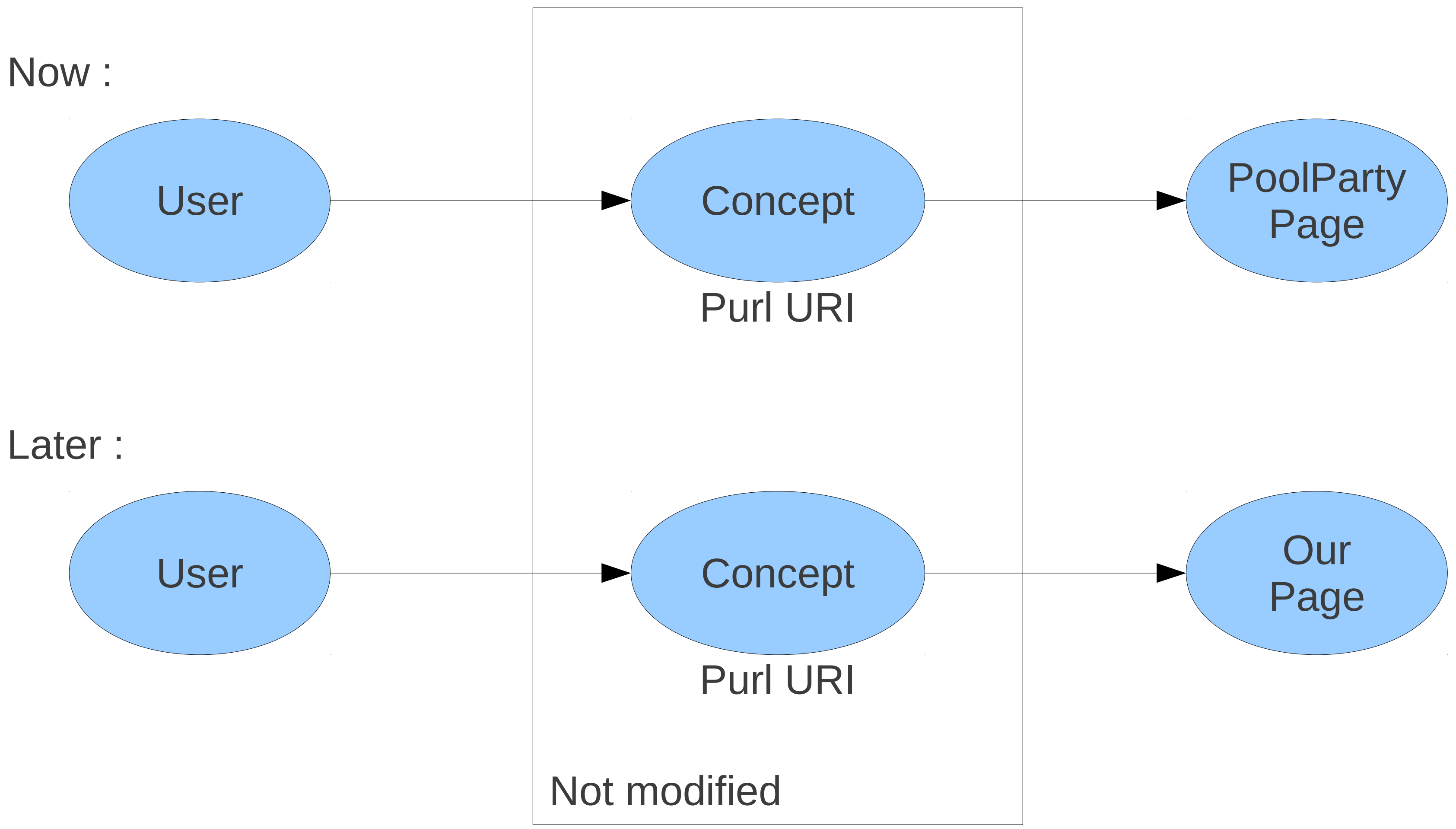
http://purl.org/astronomy/Algorithms/Concept

Shared by all vocabularies | Vocabulary | concept

Vocabulary : always plural, starts with an uppercase
Concept : prefLabel with blank replaced by undescores

- It can provide a link to a page describing the concept
- If the uri follows the default Poolparty pattern : http://prod.poolparty.punkt.at/myconcept a HTML page is automatically created for each concept but we are tied to the application
- They told us that they can create this page even with another uri
- We uses PURLs (Persistent Uniform Resource Locators) that are Web addresses acting as permanent identifiers and redirecting to a page that can be modified

Moving a description



Now :

User

Concept

Purl URI

PoolParty Page

Later :

User

Concept

Purl URI

Our Page

Not modified

I Applying semantic concepts in SimDB

II Creating and editing vocabularies

III Concepts identification

IV An example of implementation

V Evolutions

An example of implementation

- We inserted some concepts in 2 databases : PDR and Starformat
- Concerned classes are :
  - Physics
  - Algorithms
  - TargetObjectTypes

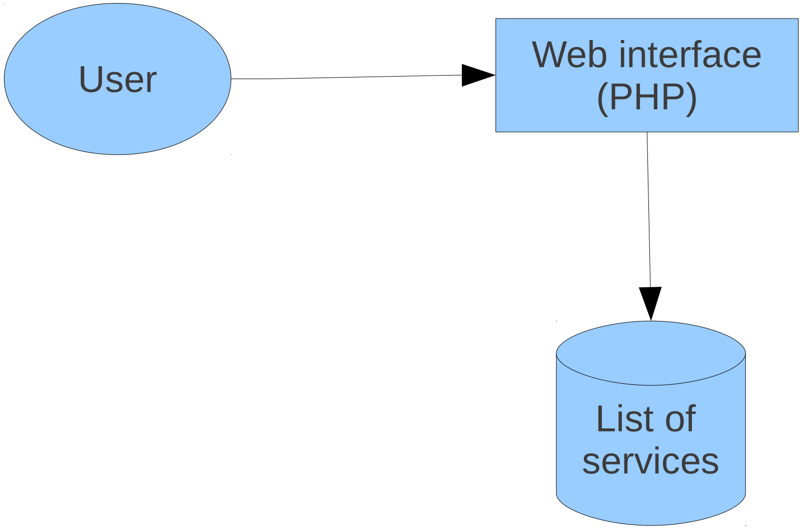
The aim is to find Protocol/Projects according to those concepts

For now, basic use case :

A user chooses one or several concepts in one or several vocabularies
The choice is done among prefLabels/altLabels
He got a list of protocols and a list of projects (both with a number of experiments) matching the criteria in the databases

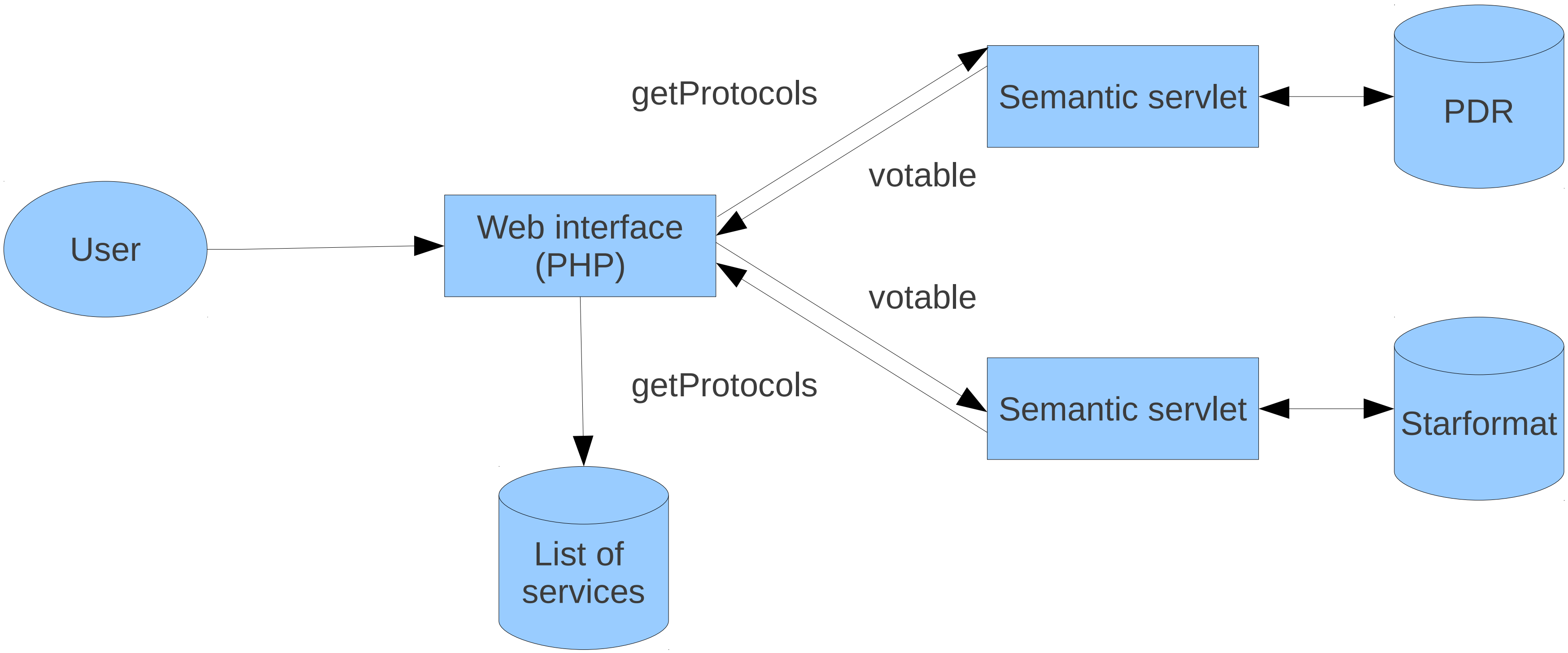
Relations between concepts are not used yet in this example.

Architecture



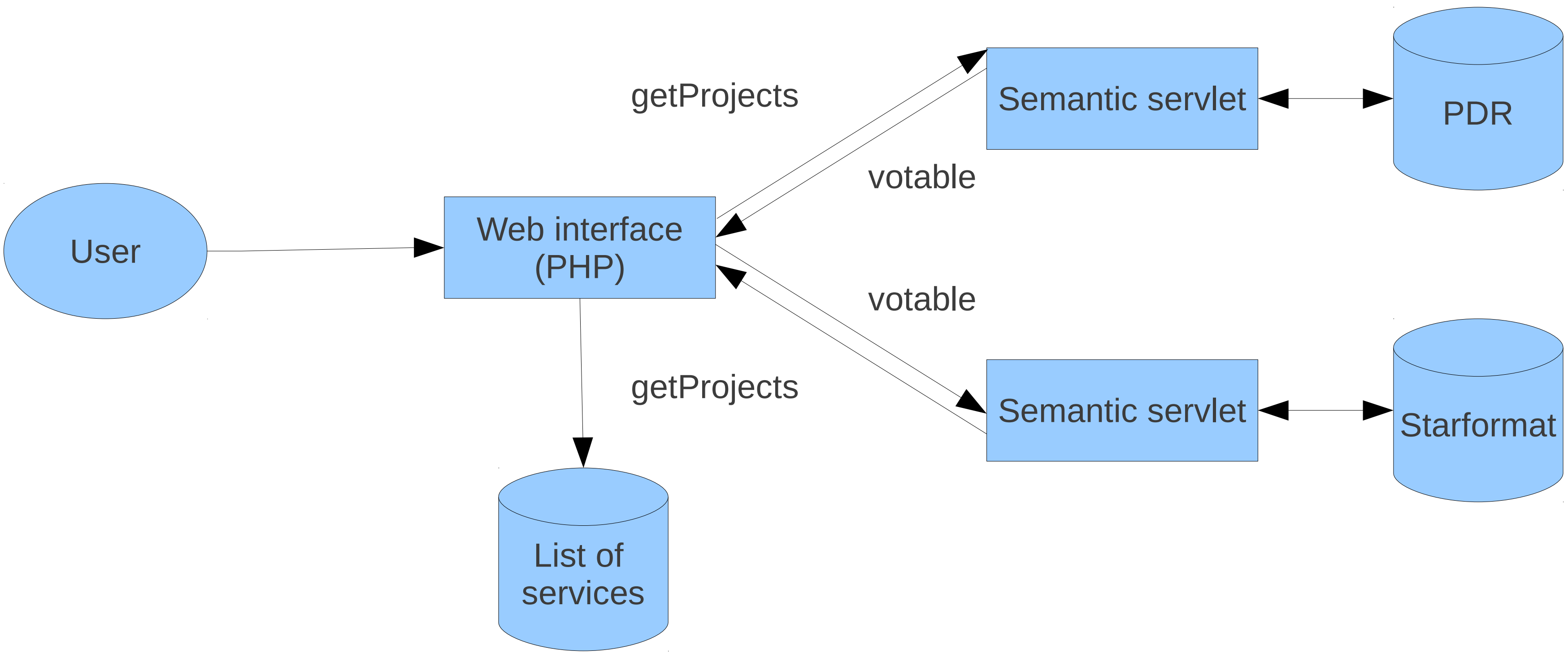
1. User select some terms
2. A request to get the protocols is sent to all services contained in the list
3. A request to get the projects is sent to all services contained in the list
4. VOTables are displayed

Architecture



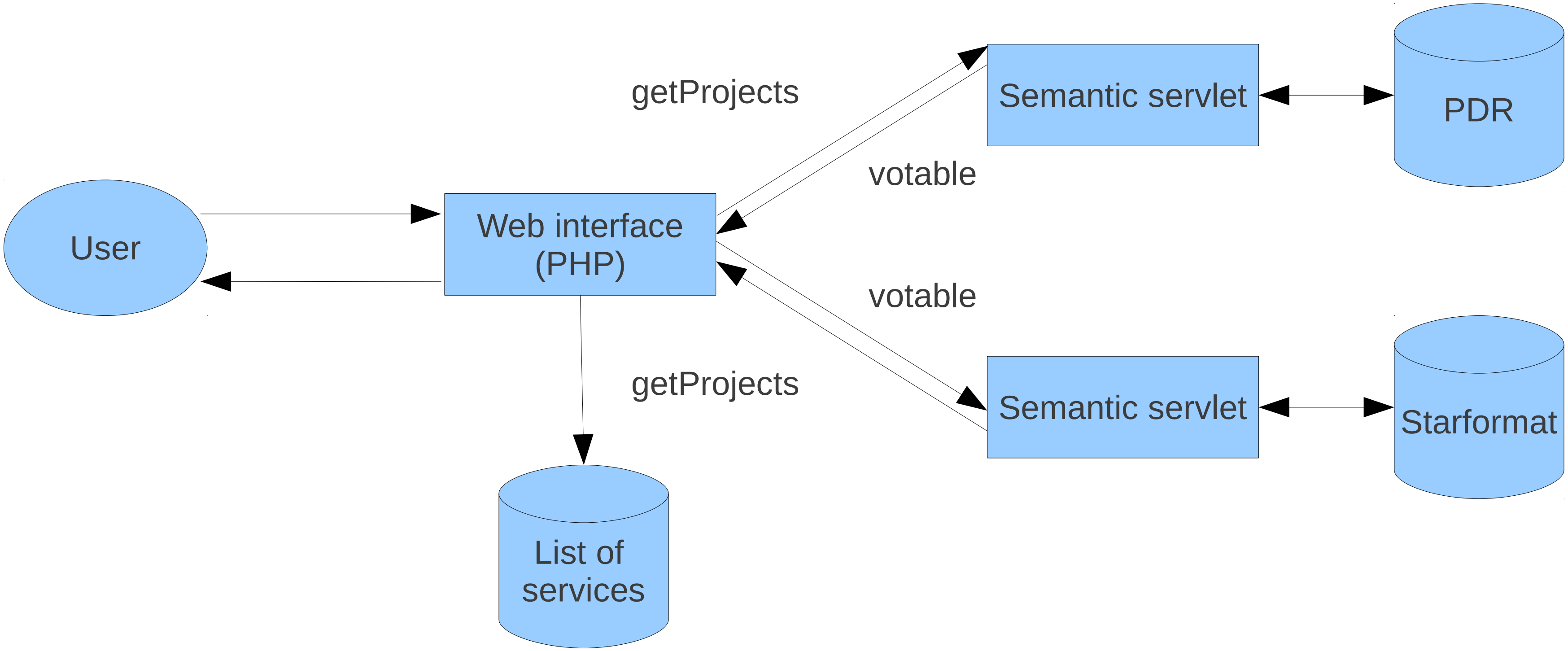
1. User select some terms
2. A request to get the protocols is sent to all services contained in the list
3. A request to get the projects is sent to all services contained in the list
4. VOTables are displayed

Architecture



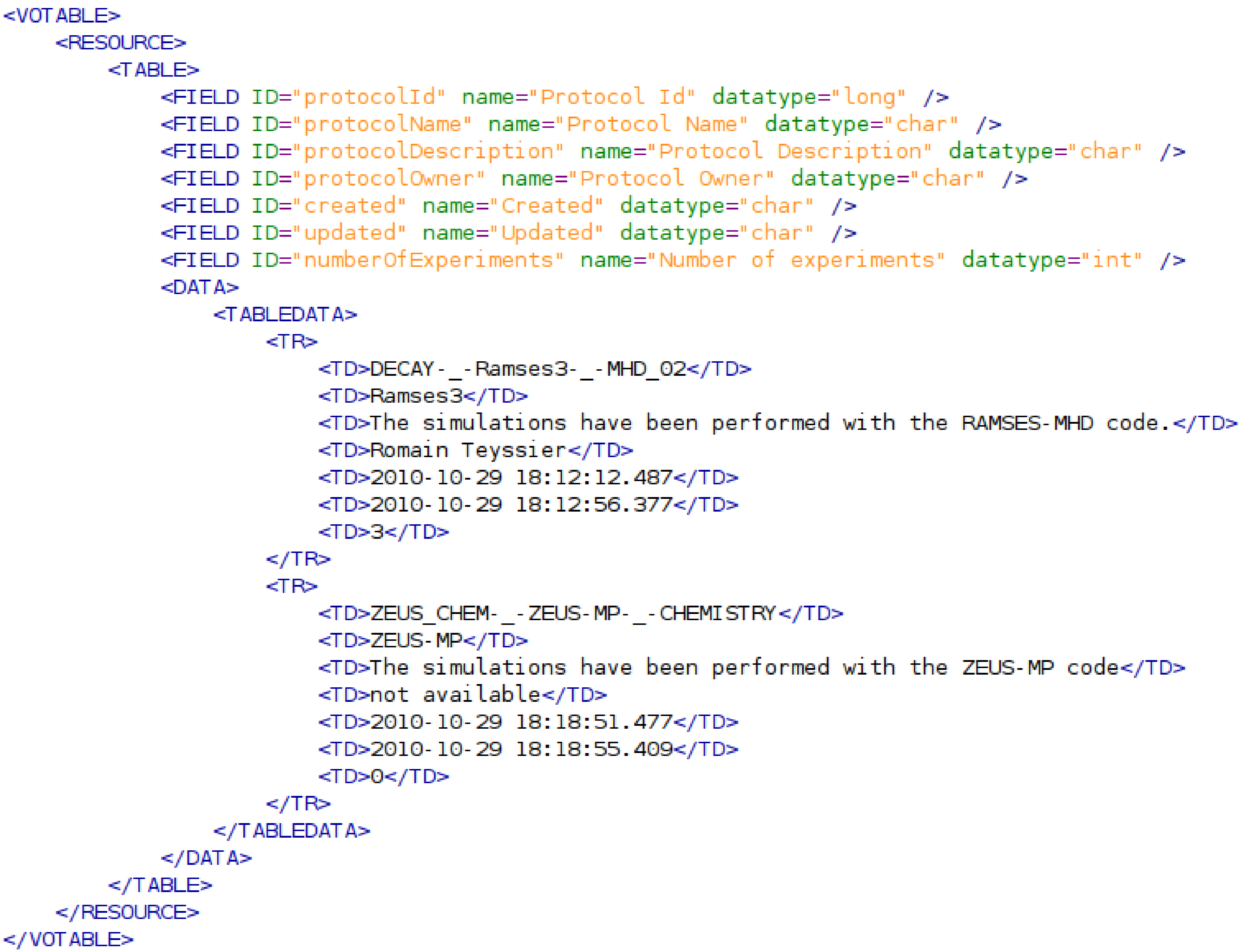
1. User select some terms
2. A request to get the protocols is sent to all services contained in the list
3. A request to get the projects is sent to all services contained in the list
4. VOTables are displayed

Architecture



1. User select some terms
2. A request to get the protocols is sent to all services contained in the list
3. A request to get the projects is sent to all services contained in the list
4. VOTables are displayed

Output example



```
<VOTABLE>
    <RESOURCE>
        <TABLE>
            <FIELD ID="protocolId" name="Protocol Id" datatype="long" />
            <FIELD ID="protocolName" name="Protocol Name" datatype="char" />
            <FIELD ID="protocolDescription" name="Protocol Description" datatype="char" />
            <FIELD ID="protocolOwner" name="Protocol Owner" datatype="char" />
            <FIELD ID="created" name="Created" datatype="char" />
            <FIELD ID="updated" name="Updated" datatype="char" />
            <FIELD ID="numberOfExperiments" name="Number of experiments" datatype="int" />
            <DATA>
                <TABLEDATA>
                    <TR>
                        <TD>DECAY-_-Ramses3-_-MHD_02</TD>
                        <TD>Ramses3</TD>
                        <TD>The simulations have been performed with the RAMSES-MHD code.</TD>
                        <TD>Romain Teyssier</TD>
                        <TD>2010-10-29 18:12:12.487</TD>
                        <TD>2010-10-29 18:12:56.377</TD>
                        <TD>3</TD>
                    </TR>
                    <TR>
                        <TD>ZEUS_CHEM-_-ZEUS-MP-_-CHEMISTRY</TD>
                        <TD>ZEUS-MP</TD>
                        <TD>The simulations have been performed with the ZEUS-MP code</TD>
                        <TD>not available</TD>
                        <TD>2010-10-29 18:18:51.477</TD>
                        <TD>2010-10-29 18:18:55.409</TD>
                        <TD>0</TD>
                    </TR>
                </TABLEDATA>
            </DATA>
        </TABLE>
    </RESOURCE>
</VOTABLE>
```

Demo

I Applying semantic concepts in SimDB
II Creating and editing vocabularies
III Concepts identification
IV An example of implementation
V Evolutions

Evolutions

- Adding SimDBs into registries (to replace local list of services)
- Using relations between terms in the services

- Defining vocabularies creation and update processes

How will they be created :

For now, several big vocabularies
Other projects will have specific needs

2 Solutions:

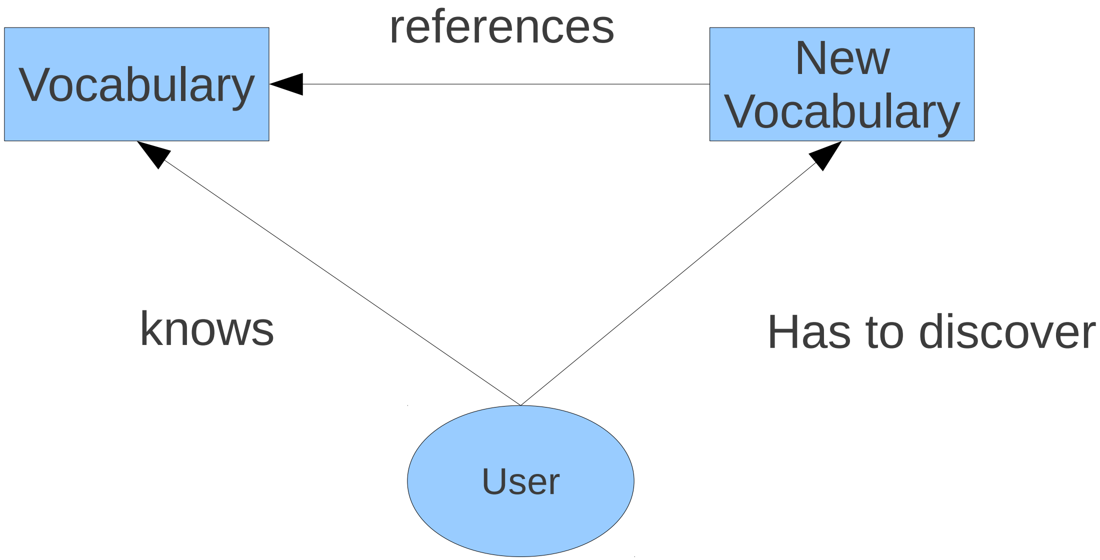
Centralized architecture :

Adding terms in global vocabularies
People ask for one/several terms, an authority decide to add them
Who will host the vocabularies, who will maintain them

Evolutions

Distributed architecture:

New projects can create new vocabularies referencing previous ones



How can applications know where the new vocabulary is located ?
If a query is done on the referenced vocabulary, how does it know he is referenced ?