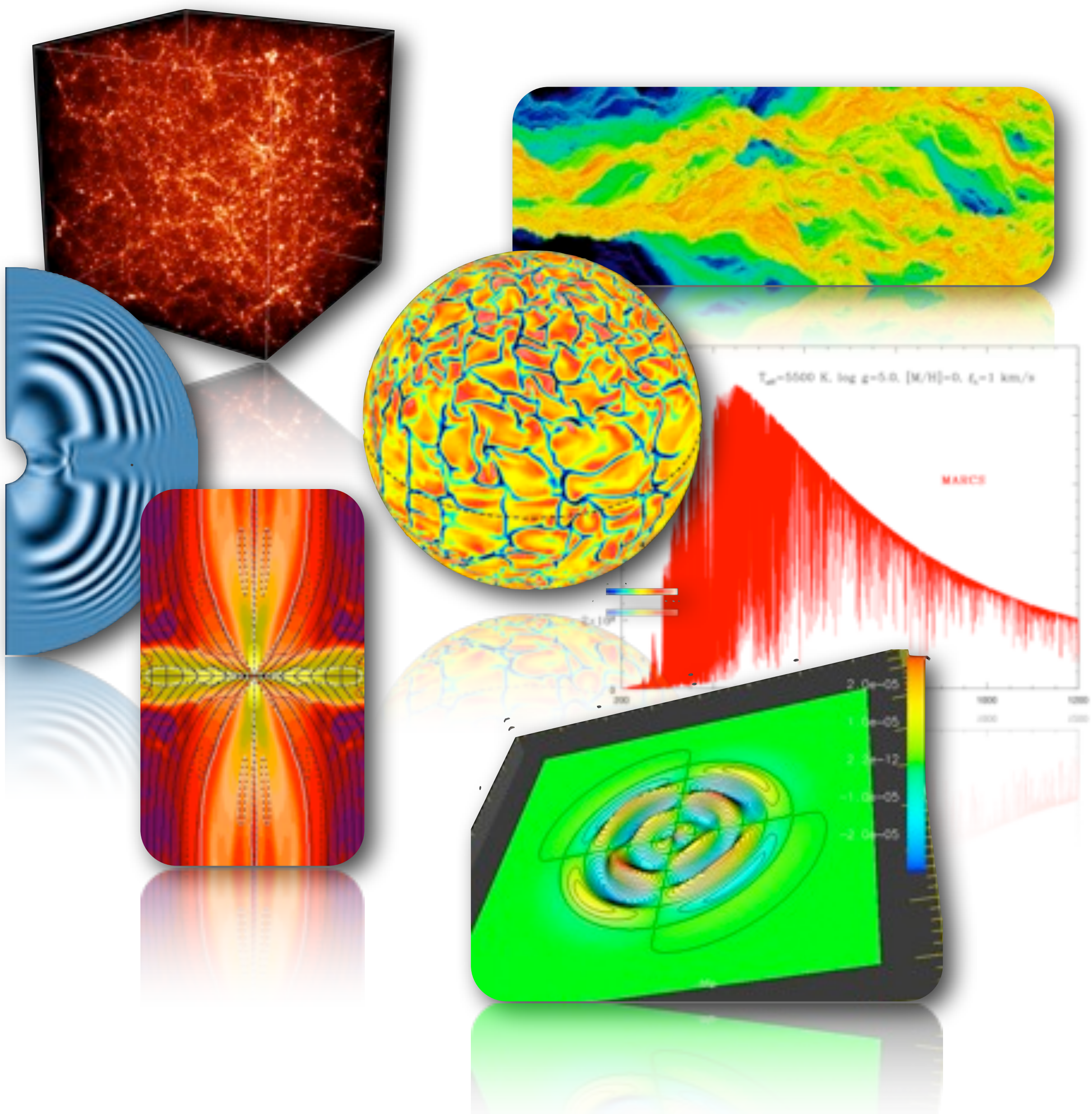
Implementations of SimDM

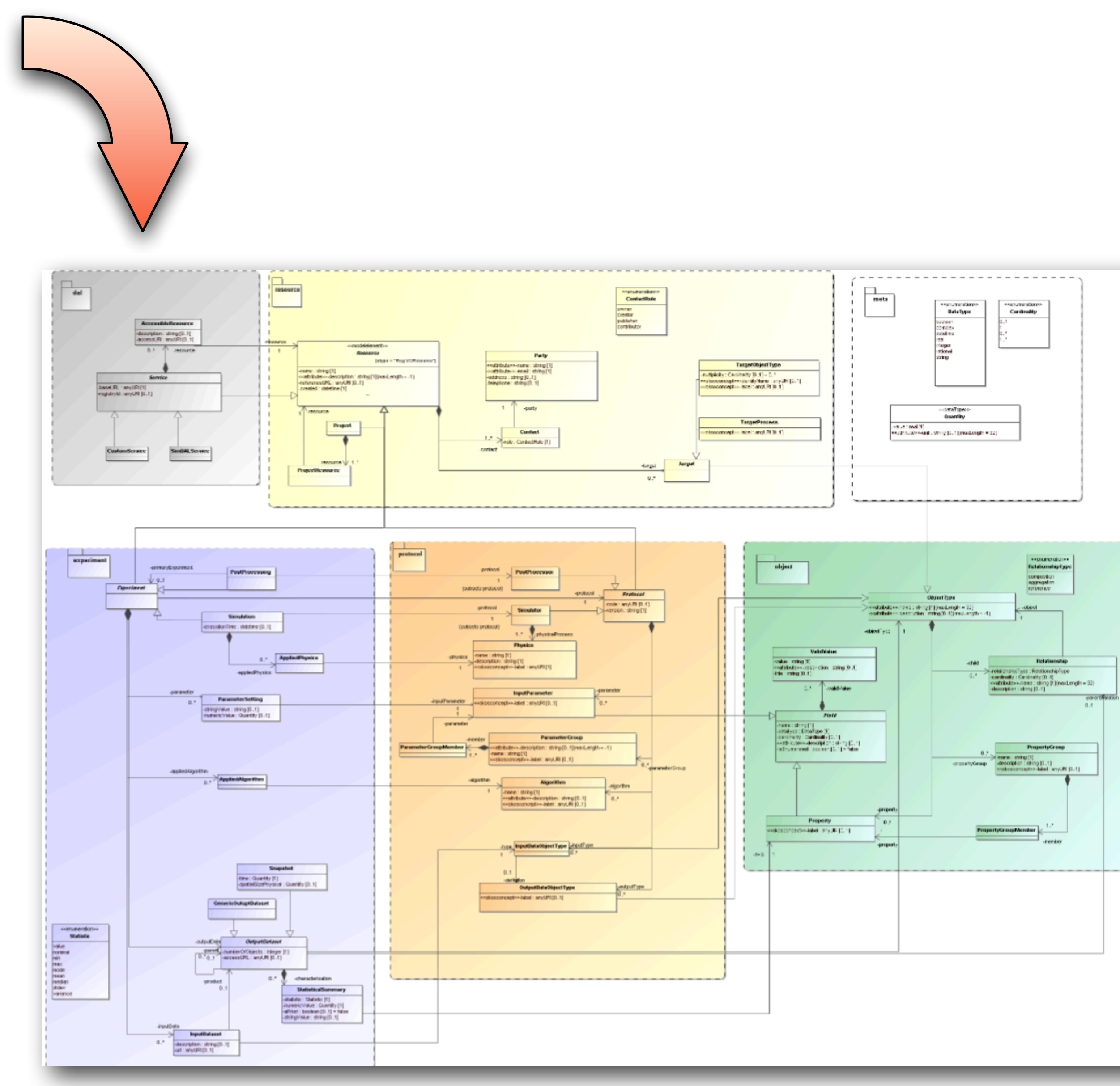
Franck Le Petit

Simulation data model

SimDM :

- high level DM
- implies that it may be difficult to implement
- but it is general and can cover many categories of simulations





4 implementations :

- PDRDB
- Starformat
- DEUVO
- GAVO (G. Lemson)

GAVO example

G. Lemson et al. (MpI - Garching)

Simulation Database - Browser - 1.0-20110520 - Browser

SimDB Browser - Home :

Browse Model

- All Resources
  - All Projects
  - All Experiments
    - All Simulations
    - All PostProcessings
  - All Protocols
    - All Simulators
    - All PostProcessors
  - All Services
    - All SimDALServices
    - All CustomServices
- All Partys

Documentation

Browse documentation of the data model.

SKOS Concepts

Browse SKOS vocabularies used in the model.

SQL Interface

Query the SimDM database using ADQL (well, SQL).

XML Validator

Validate an XML document against the data model's XML schemas.

XML Loader

Load an XML document into the database.
Requires a login.

Log management (requires a login!)

Cosmological simulations

PDR database

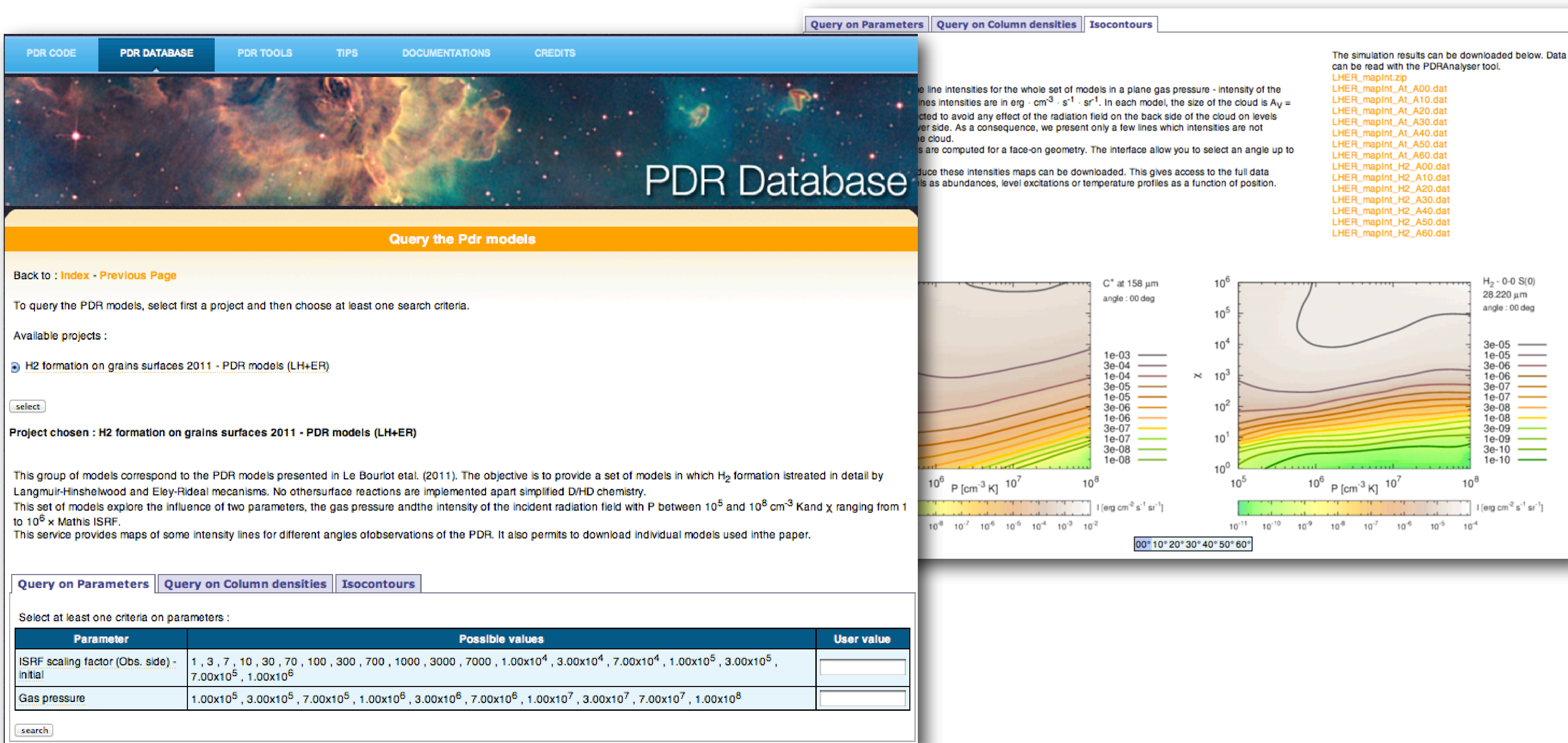
F. Le Petit, D. Languignon, N. Moreau, B. Ooghe, L. Bourgès

http://pdr.obspm.fr

Models of interstellar clouds (micro-physics simulations)

- temperature,
- abundances of hundreds chemical species : H, H_2, C^+, C, CO, H_2O, ...
- grains properties
- Lines intensities
- ...

Facilitate interpretations of HERSCHEL and ALMA observations



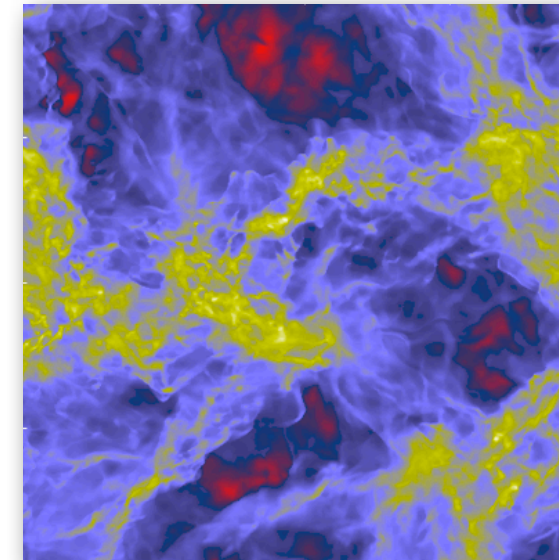
Starformat

P. Hennebelle, N. Moreau, C.M. Zwolf, B. Ooghe (LERMA, Paris Observatory)

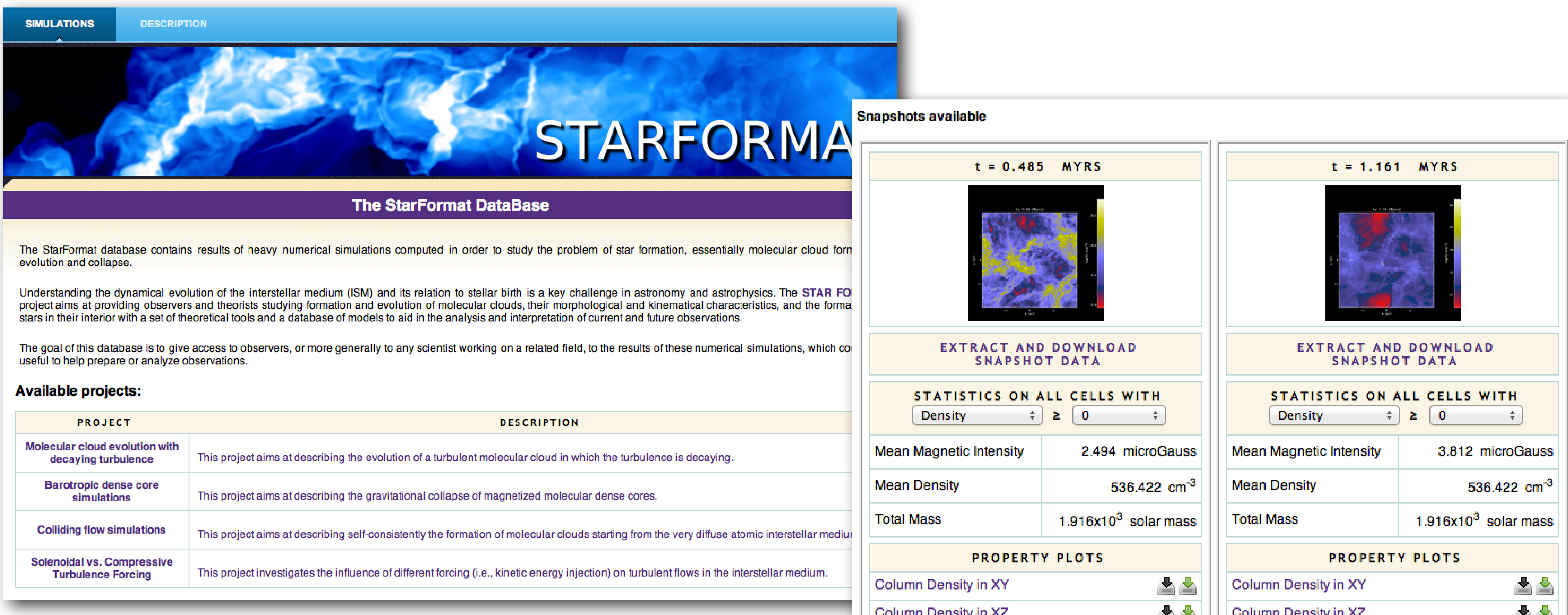
http://starformat.obspm.fr

MHD simulations for the interstellar medium

- density, velocity, magnetic fields, ...
- Dense cores, IMF
- Cooling flows
- MHD + chemistry



Preparation of ALMA observations



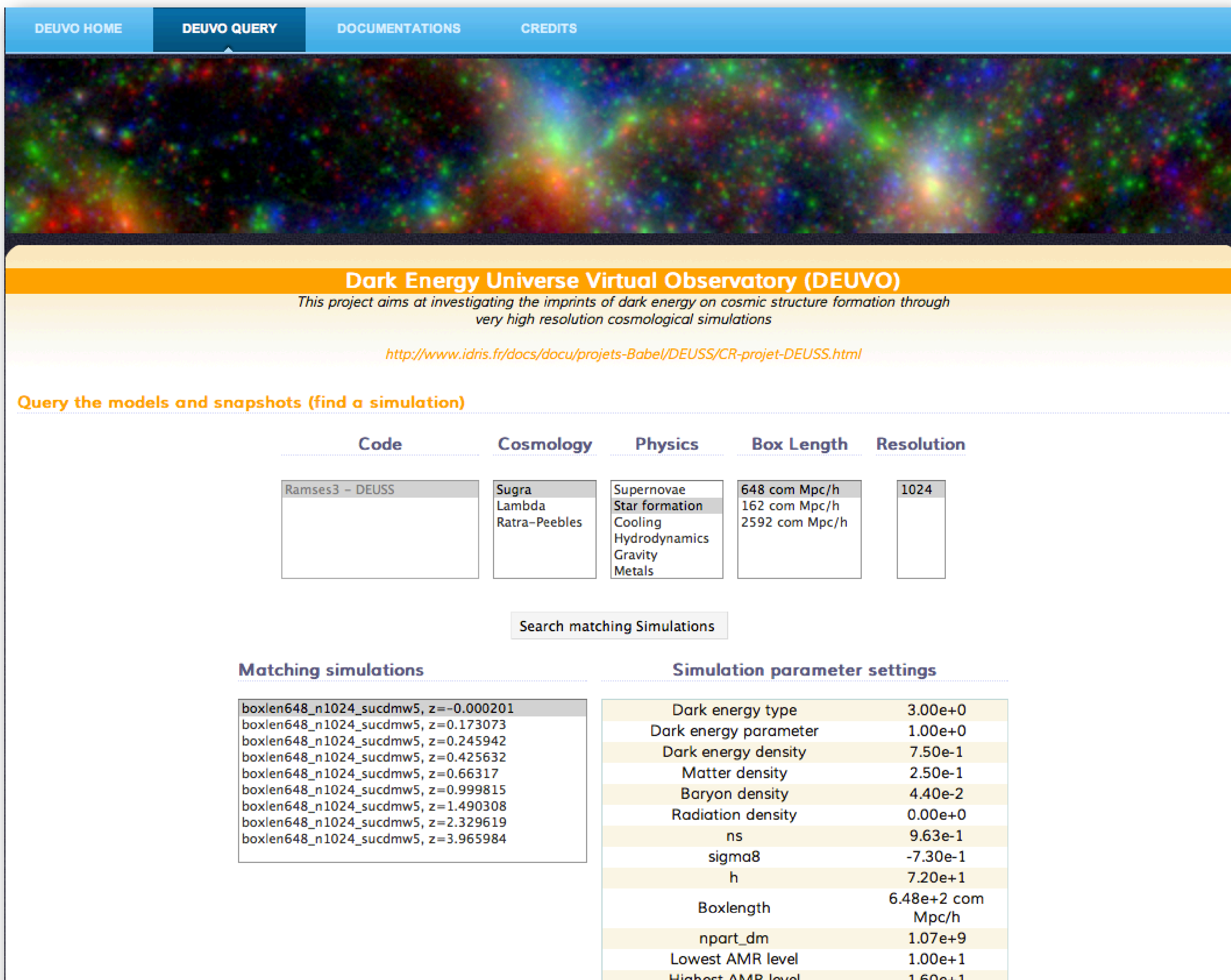
DEUVO

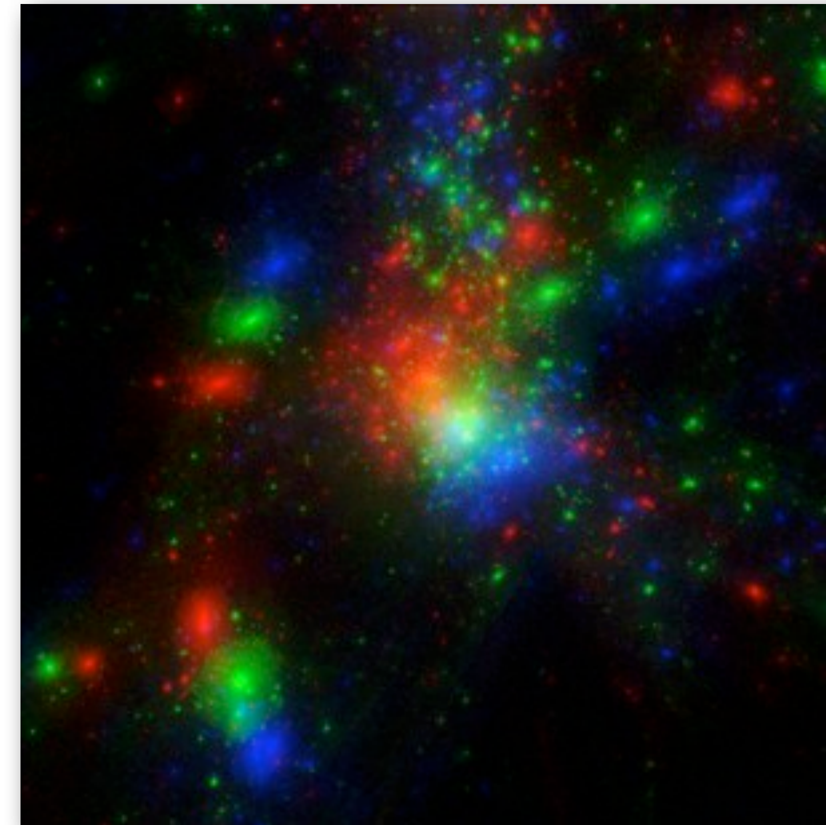
Y. Rasera, J.-M. Alimi, D. Languignon, B. Ooghe (LUTH - Paris Observatory)

in development

Publication of Grand Challenge cosmological simulations

- DEUSS : up to 9 billions particles
- 50 To of data produced
- Access to halos properties



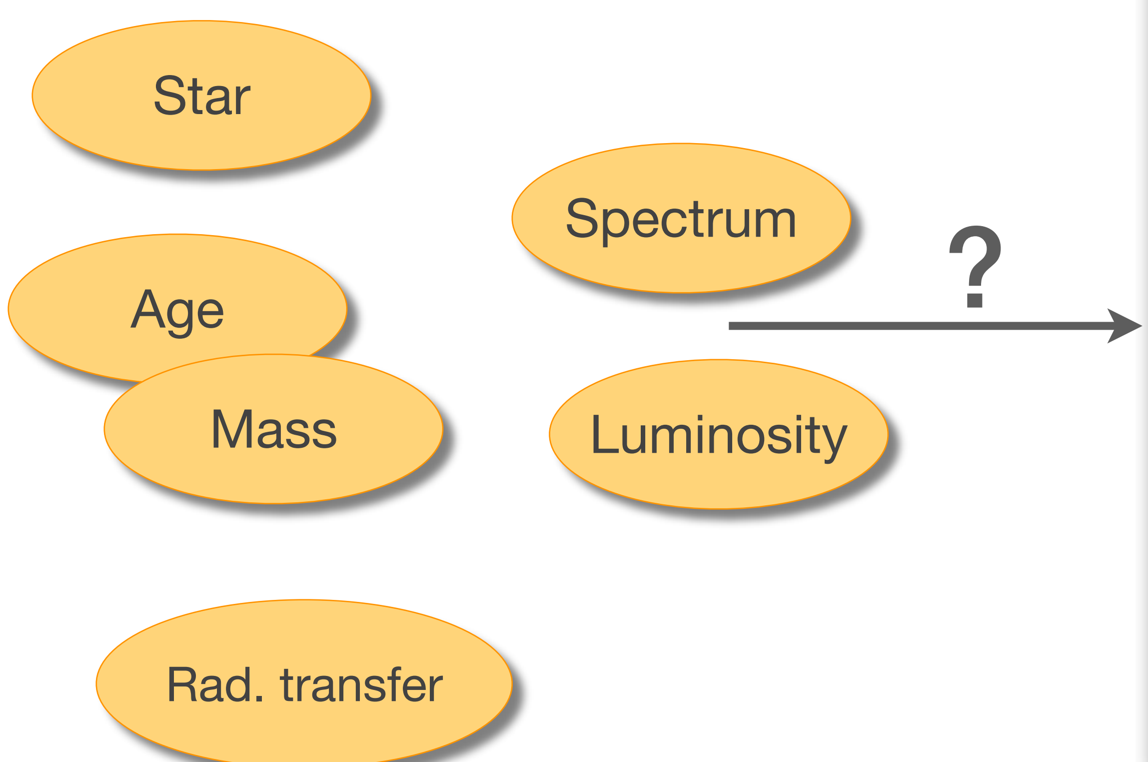


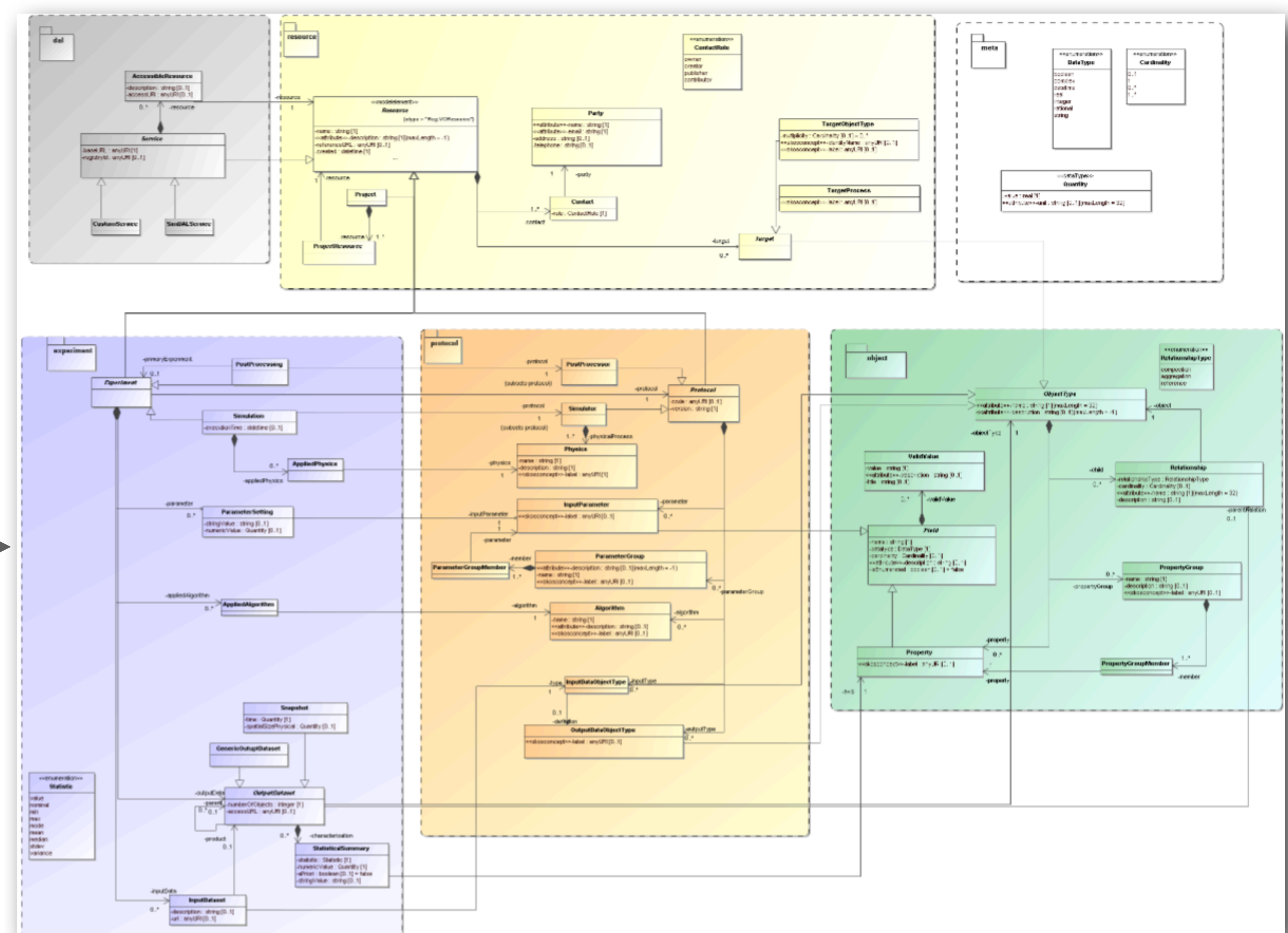
Implementations

1 - Mapping of the DM
2 - Set up a relational data base
3 - Set up an ingestion pipeline
4 - Queries

1 - Mapping on the DM

SimDM is an abstract model : mapping can be difficult





1 - Mapping on the DM (2)

Mapping requires training :
Having several examples in the implementation note can help scientists to do this.

Up to now : 2 detailed descriptions in the implementation note

Some pieces of advice :

- Some SimDM classes have SKOS concepts
  - try to map those ones first

- Do not try to fill the full SimDM
  - Publisher should ask himself how he wishes users discover simulation

  Example : if the discovery of simulations is only done with input parameters, it is pointless to fill all the statistics.

All concepts

3+1 Formalism
8-Wave Scheme
Accelerated Lambda Iteration
Adaptive Mesh Refinement
Advection Upstream Splitting Method
Algorithm
Alternating Direction Implicit
BiConjugate Gradient
BiConjugate Gradient Stabilized
Block Based AMR
Bulirsch-Stoer
Cell Based AMR
Cell Centred
Central Difference Scheme
Chebyshev Iteration
Conjugate Gradient Method
Conjugate Gradient Squared Method
Constrained Transport
Coupled Escaped Probability
Crank-Nicolson
Discontinuous Galerkin
Discontinuous Galerkin methods
Escape Probability
Euler
Exact Radiative Transfer Method
Exact Riemann Solver
Extended Finite Element Method
Fast-Multipole Method

List of vocabularies :
http://votheory.obspm.fr

2 - Set up a relational database based on SimDM

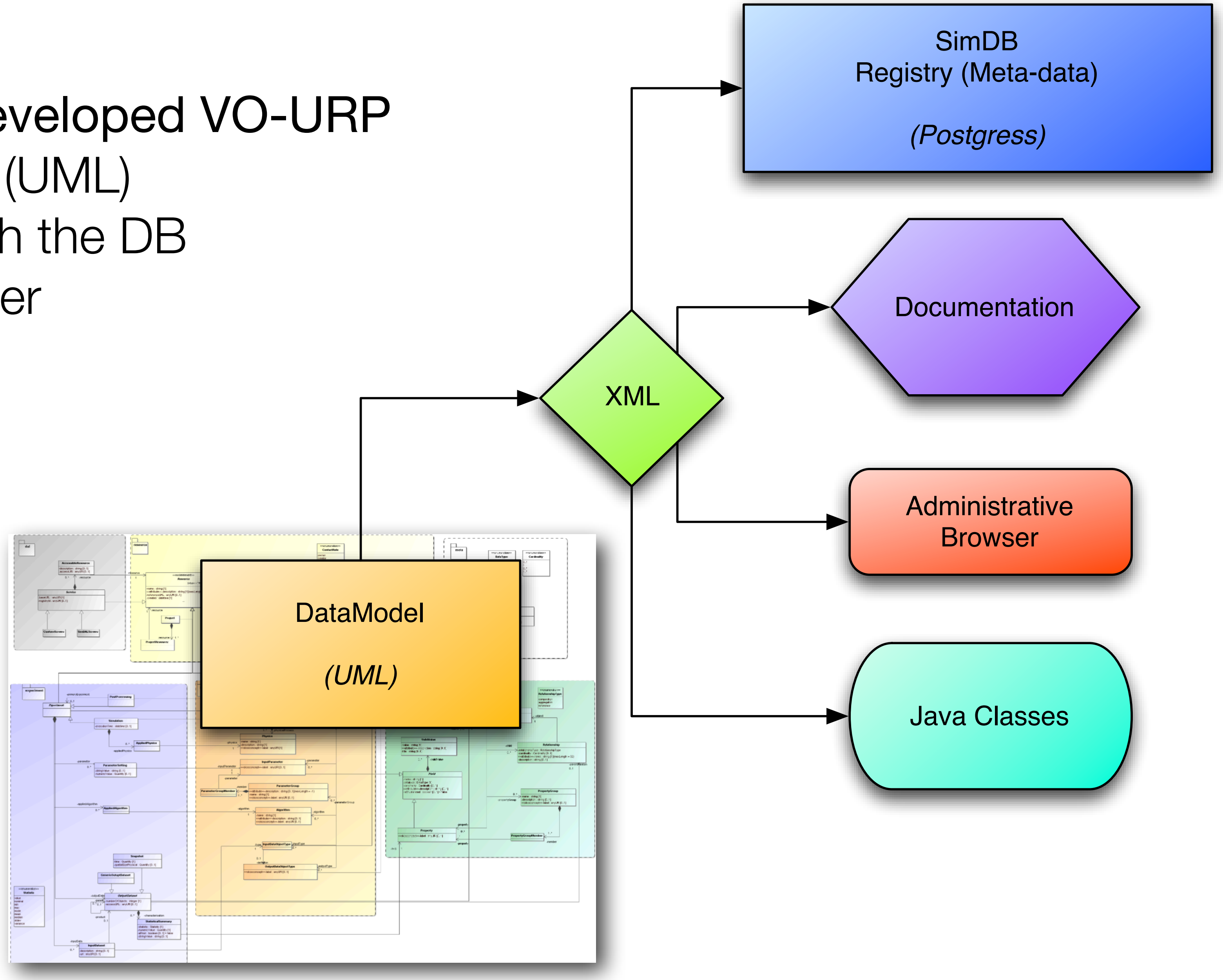
The creation of the relational data base based on SimDM can be complex

Gerard Lemson & Laurent Bourges developed VO-URP

- generate a database from SimDM (UML)
- provides APIs to communicate with the DB
- generates an administrative browser
- provides documentations

It requires time to learn how to use VO-URP but it is a benefit on the long term

Technology : Java JSP



Some tests are done in Strasbourg with Saada

3 - Ingestion pipeline

How to create metadata and store them in the data base ?

- Depending on the code, one can have hundreds or thousands metadata to produce.
- Moreover, if the code evolves frequently, the metadata to store evolves also (and we do not wish to re-build the database and queries each time ...)

Example : micro-physics simulations

- *code computes line intensities*
- *new lines can be added frequently*
- *new metadata (properties, statistics, ...)*

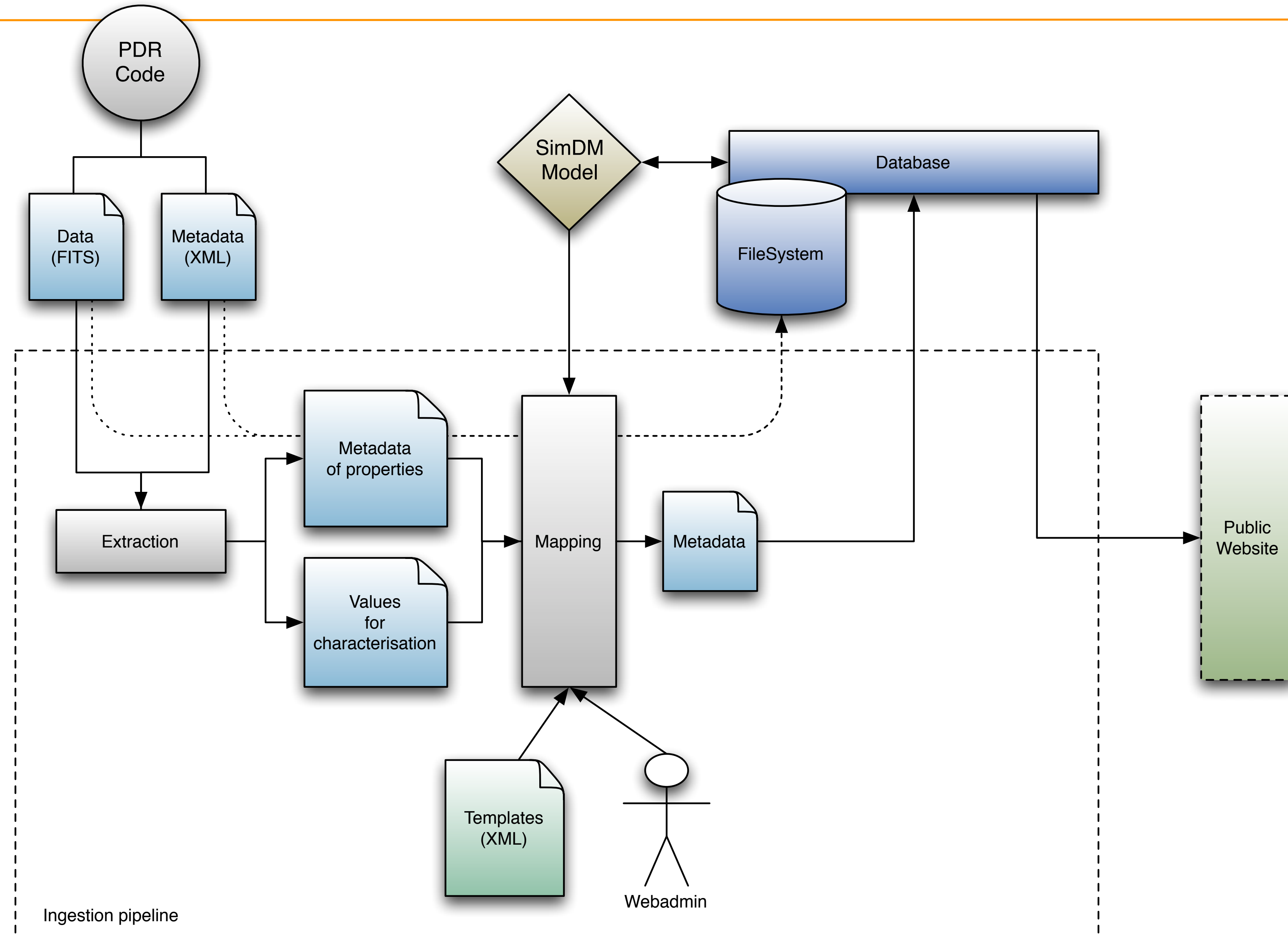
Generation of metadata :

- after a run of the code
- modifications of the code to generate automatically metadata

Ingestion pipeline

- When possible the ingestion pipeline should adapt itself to new metadata
- Automatic modification of the DB

3 - Ingestion pipeline

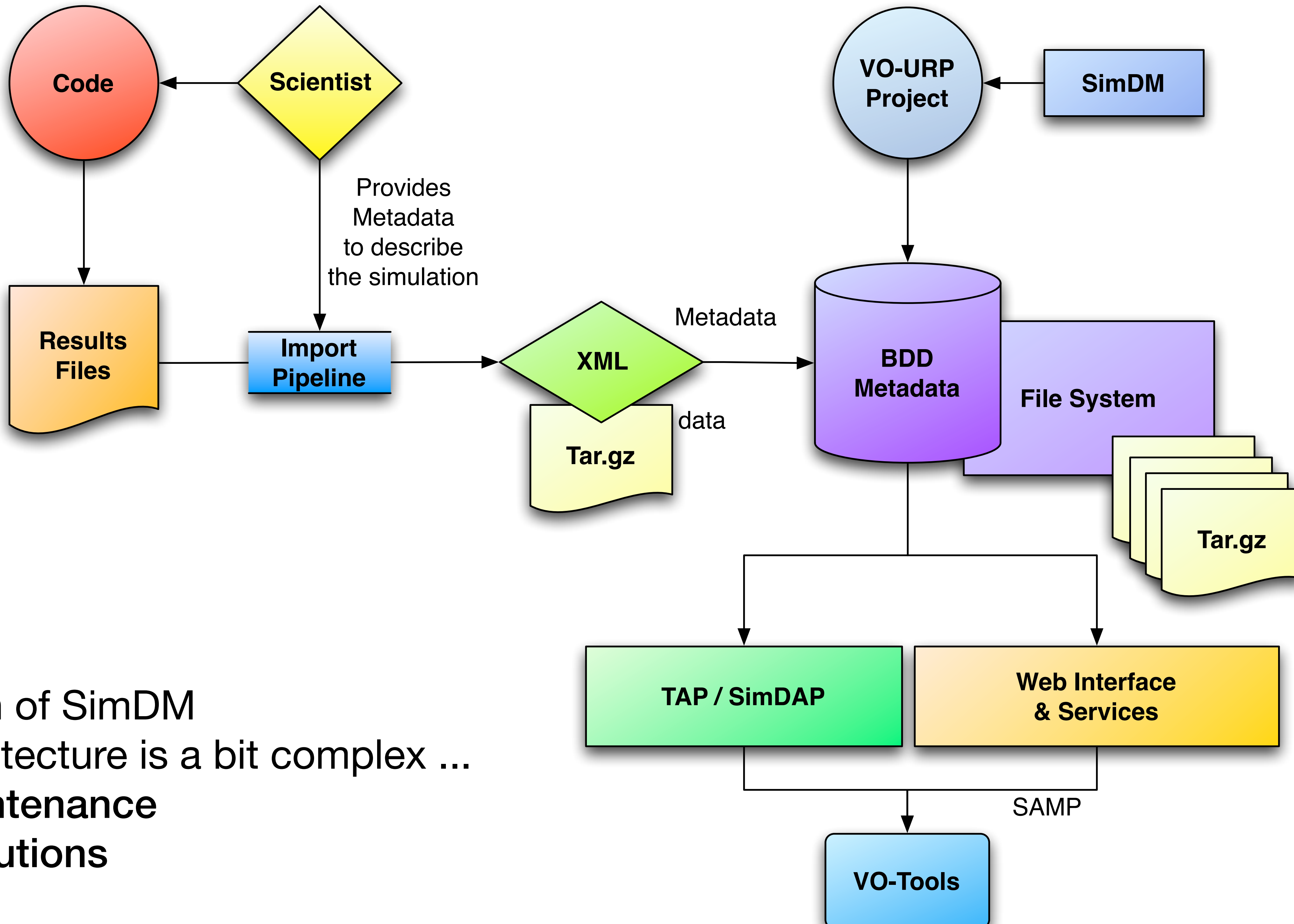


4 - Queries

- Queries can be difficult : hierarchical model - lots of joints
- Optimization of queries required on large data bases (*Ex : millions of halos*)

Typical queries should be simplified by the access protocol : SimDAL (SimTAP)
(see D. Languignon’s talk)

Complex architecture



Implementation of SimDM

- the final architecture is a bit complex ...
- requires **maintenance**
- requires **evolutions**

=> Most evolutions have to be thought from the beginning.

Conclusions

SimDM : high level DM that can cover many different kind of simulations

It is difficult to set up a new service based on SimDM

- Mapping :
  - IVOA can help with a comprehensive implementation note
  - Share experiences

Before any implementation future evolutions / maintenance of the services have to be planned

Discussions in the VO-Theory I.G. :

- What global architecture for VO-Theory services in the VO ?
  - Which link with the registries ?
  - Are SimDBs required ?

- The access protocol (SimDAL) could simplify the implementation

- Answers should be given quickly :
  - more and more people want to share simulations
  - if we want it is done in a VO context, we have to provide good advices for implementation