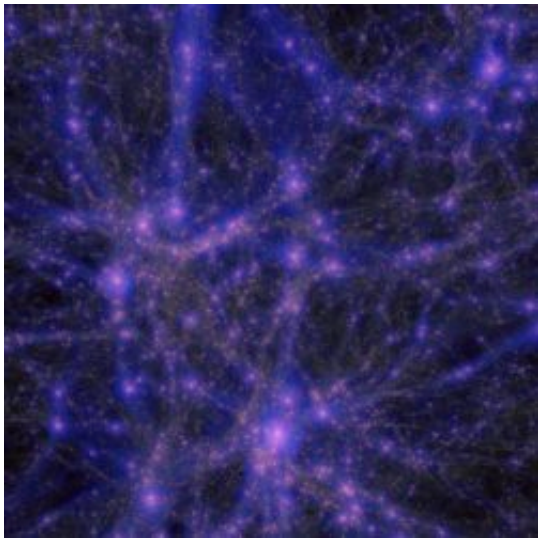
Implementing the Simulation Data Model (SNAP DM) for Horizon Halo & Galaxy catalogs (Galics …)



SNAP Meeting, CRAL, 04/03/2008

Laurent Bourgès (EURO-VO DCA Project)
Hervé Wozniak, Jérémy Blaizot (CRAL)

Presentation Agenda

- Horizon catalog database
- Horizon schema
- First implementation of the SNAP DM (demo)
- New database schema (SNAP compliant) : metadata & data (many runs)
- SNAP DM changes
- Questions ?

Horizon catalog database 1/2

Description

- Various kind of particle simulations (SNAP Level 0) : gallcs, mare nostrum …
- Several post processing softwares :
  - Halomaker (v1, v2)
  - Treemaker
  - Galaxymaker
  - GalaxyFinder (mare nostrum)
  - MoMaF (light cones)

=> Existing database & web application (mysql 5, tomcat) with a fully normalized schema (JPLF)

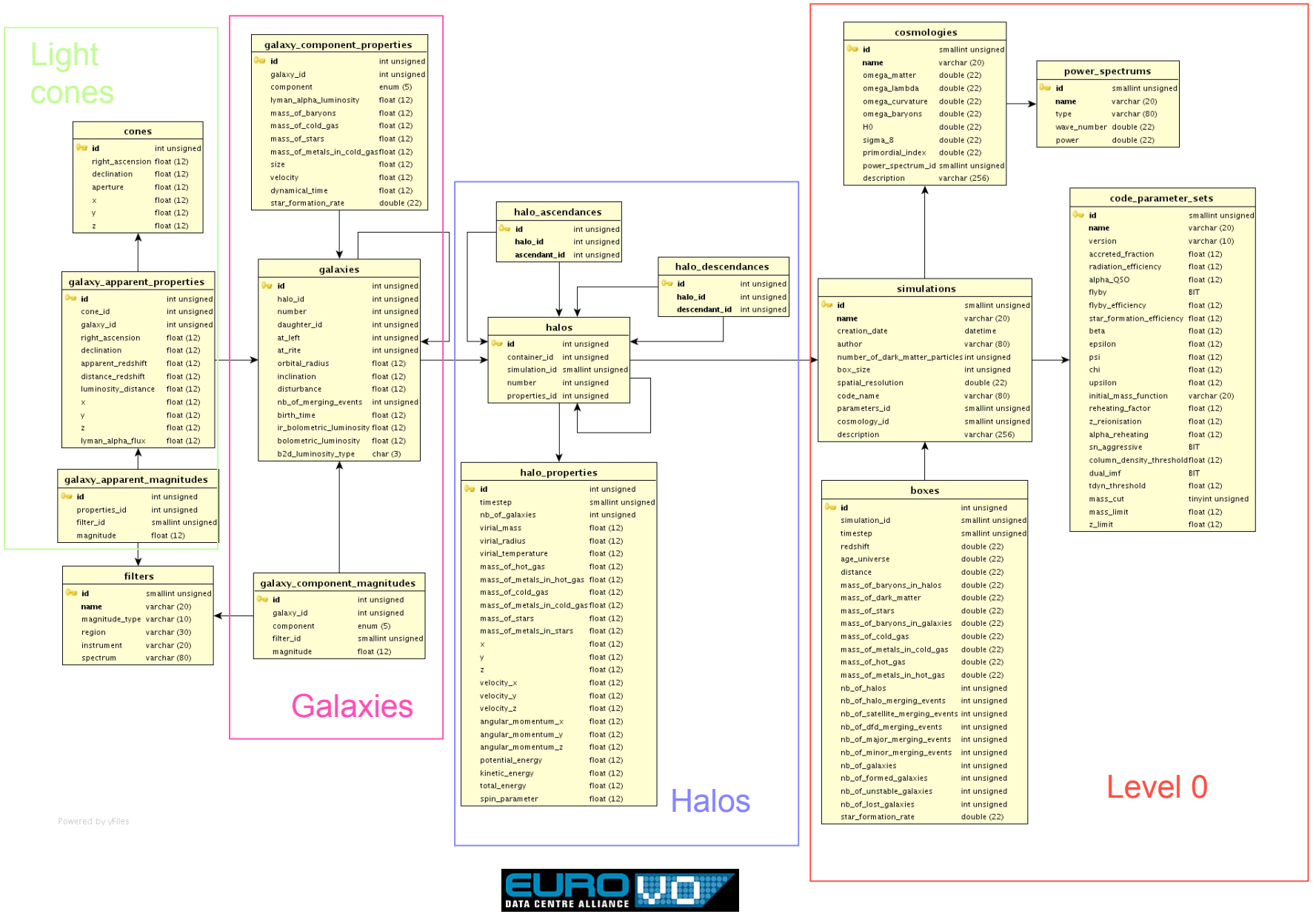
Horizon catalog database 2/2

Schema Description

- Particle simulations : parameters
- Halo catalogs : properties, tree (parents & children for every halo)
- Galaxy catalogs : properties, parent halo, sub components (bulge, burst, disc)
- Magnitudes computed for many filters for each galaxy component
- Light cones

Content : only 2 galics test simulations (light &heavy) with halos, galaxies & magnitudes

Horizon database Schema



EURO VO DATA CENTRE ALLIANCE

Horizon SNAP Implementation 1/2

Mapping entities

SNAP DM (2/10/2007 version)

- 2 protocol instances : halomaker, galaxymaker
- One SnapPostProcessing instance per run
- Representation Objects :
  - Halomaker : Halos
  - Galaxymaker : Galaxies, Galaxy Bulge, Galaxy Burst, Galaxy Disc

  Many properties (see demo)

- Characterisation : min / max / mean / stddev
- Target Objects : in discussion with Gerard
- Parameters, Physics & Algorithms : waiting for Jeremy

Solution : postgres 8.x, Java 5 (JPA, tomcat) : data loader & web application (SNAP DM browser)

Horizon SNAP Implementation 2/2

Demo & Problems

- Demo : http://horizon-vo.univ-lyon1.fr:8080/HorizonSNAPWeb/
- Problems :
  - Need property groups & avoid duplication
  - Missing meta data in existing database schema (parameters for each run)
  - Database loading uses XML files (hard to generate)
  - Queries are complex & slow : many joins on big tables (magnitudes)
  - 8 Gb for 1 big run of galaxymaker (magnitudes : 7 Gb)

⇒ Partitioning is required to store different runs in different tables to allow table scans & have less records

⇒ New optimized schema

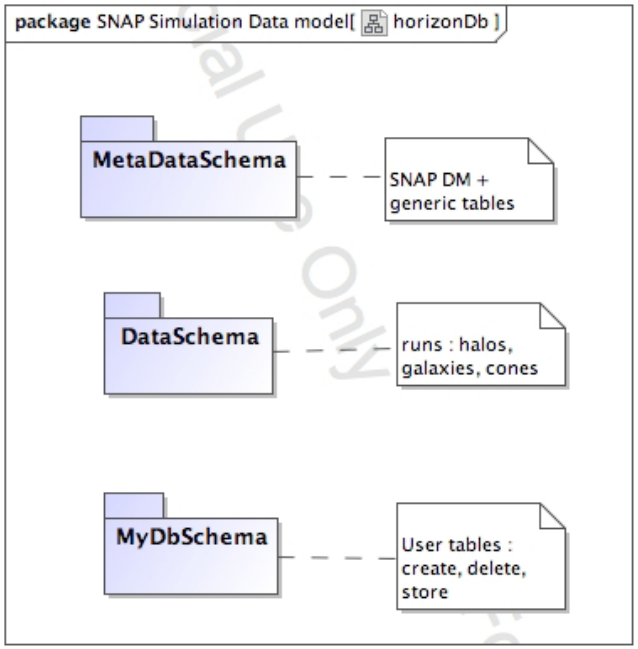
New database schema

Description

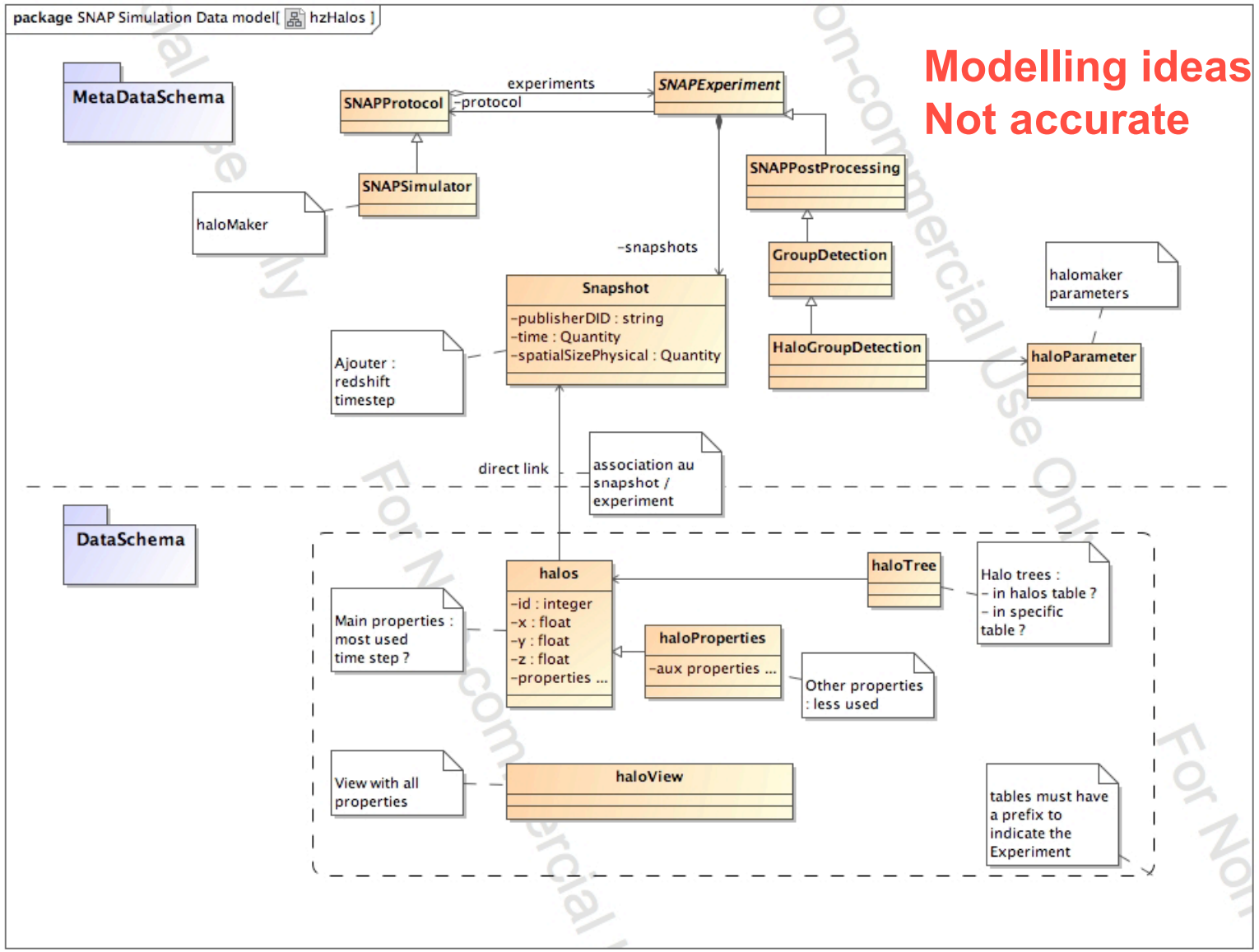
- 3 schemas instead
  - SNAP DM (meta data)
  - Data (runs)
  - MyDb (users)

- SNAP DM is useful to provide documentation too (run pipeline, objects, properties)

- Each run has its own table :
  - Table scan are possible
  - Jeremy can add properties, indexes …

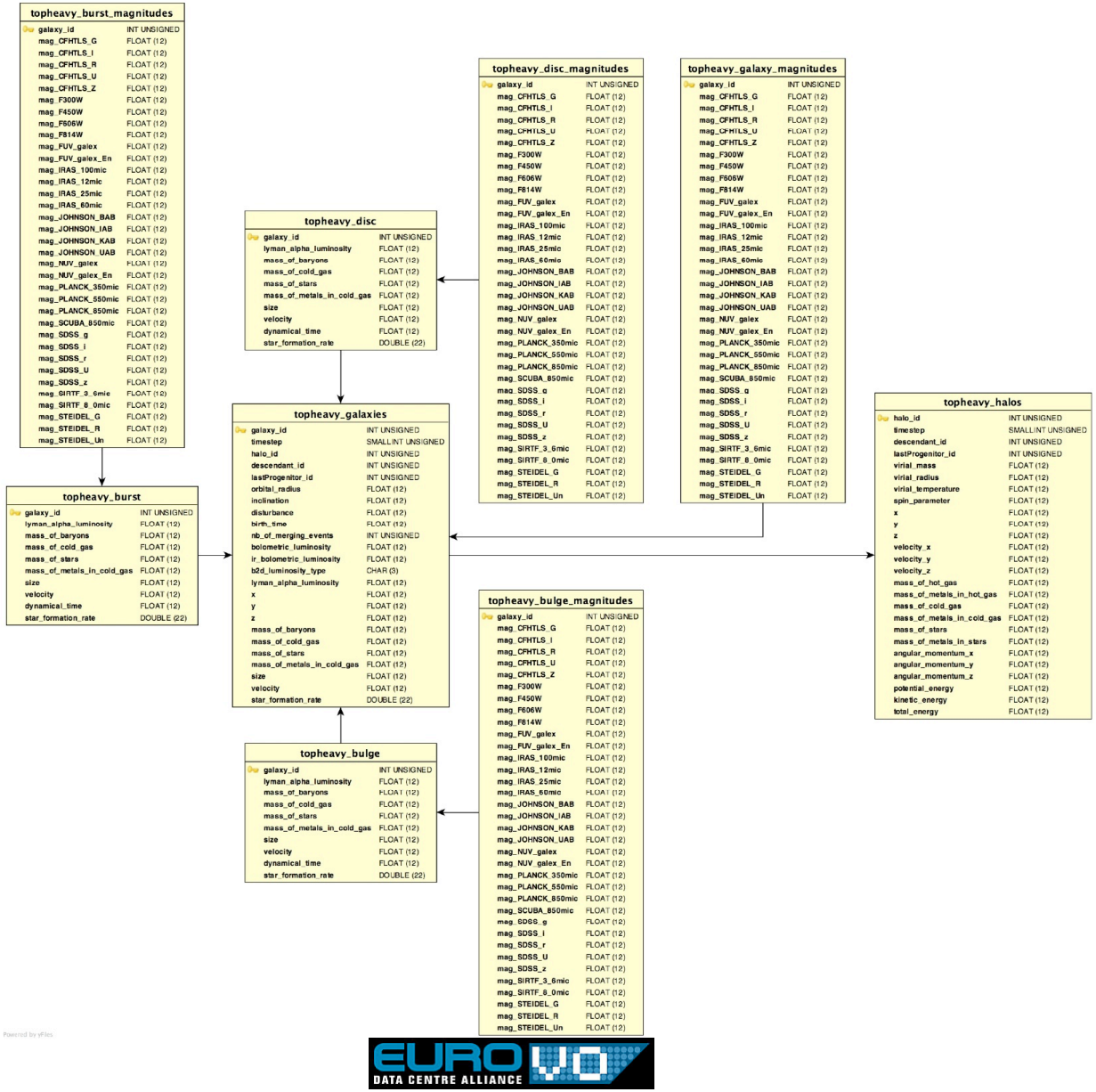


Ideas for halo catalogs



Modelling ideas : Not accurate

Prototype for halo & galaxy catalogs



Prototype tests

- Queries are simpler (views can help)
- Database is smaller : 1 Gb vs 8 Gb
- Table scans are very faster

- Query example :

```
select stellar_mass from mag1, mag2 where mag1.filter1 =
    'SDSS_G' and mag2.filter = 'SDSS_U' and mag1.magnitude -
    mag2.magnitude between -20 and -10
```

- Existing schema : timestep = 39 : 45s 13244 records
- Prototype schema : timestep = 39 : 1.53s 13244 records

=> 25 times faster (mysql 5)

SNAP DM Changes

- Gerard update the SNAP DM (13/02/2008)
- New idea : SNAPProject to hold shared objects (objects, properties, parameters)
- SNAPExperiment changed in the good way
- Work in progress :

  svn :

  http://code.google.com/p/volute/
- UML model will have full descriptions for every entities & attributes
- XML schema generated by XSLT

Conclusion

- New schema will contain SNAP DM + data
- New Schema is designed to be simple :
  - Queries are easy to write by scientists
  - Queries are as fast as possible (cache, disks)
- Data Loader need to be updated to generate meta data (SNAP DM) & fill run tables
- Questions ?