design patterns for theoretical data access

David Languignon

ASOV - Paris - 14/01/28

how it began

problem

- variety of code domains, quantities, units, …

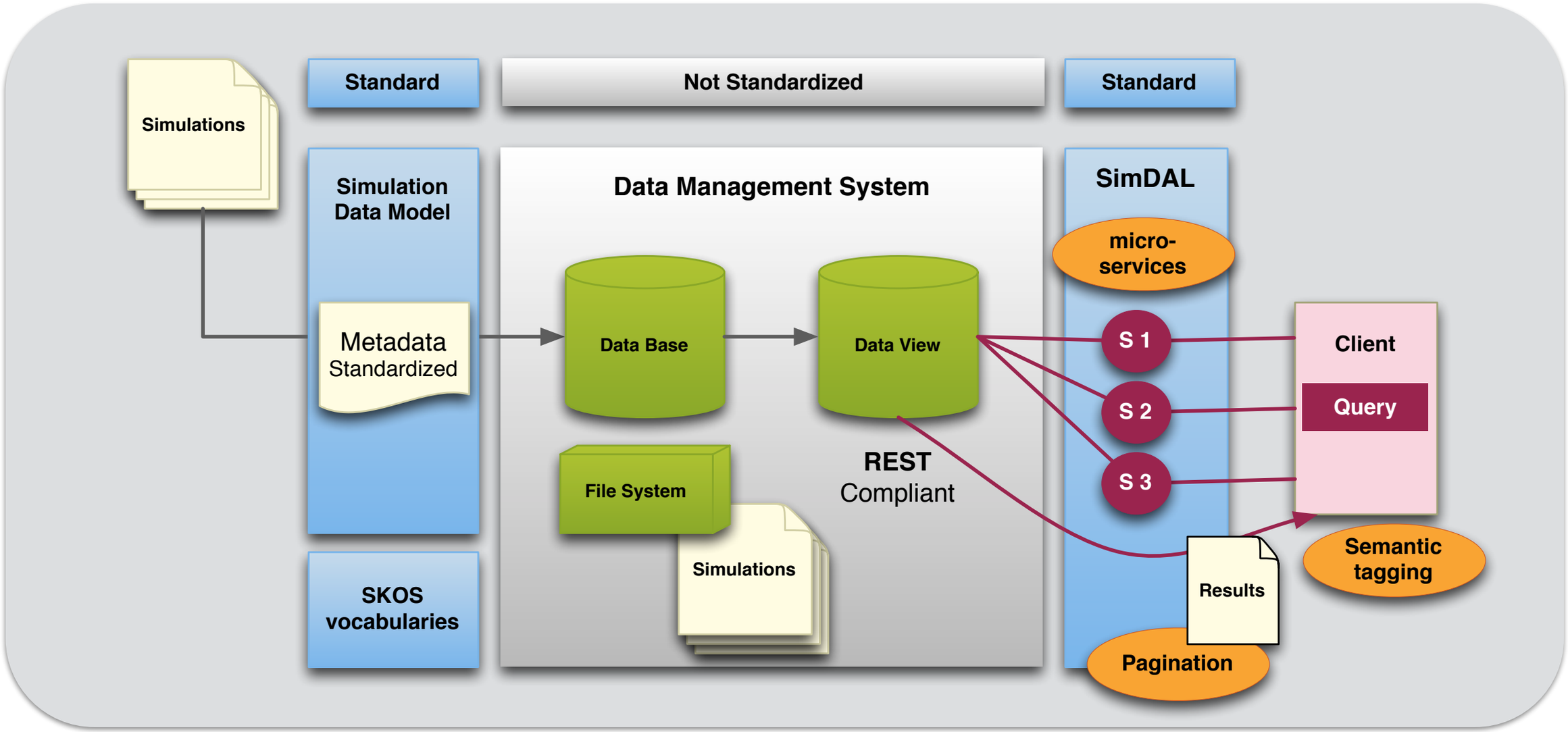
solution

- **standard description format** as an abstract data-model : SimDM

the end already ?

- practical implementations of the standard format
  - descriptions **can be huge**
    - ✓ deep hierarchy of nested atomic elements
    - ◉ long nested **collections of large elements**
- code **raw output data** handling issue remains

theoretical service architecture



solution patterns encountered in our problem solving journey

- meta-model
- micro services architecture
- generalized space cutout
- pagination
- semantic tagging
- streams*

meta-model

- Simulation Data-Model
  - Model whose instance is also a model
    - Abstract model to make concrete models
      - 1 concrete model per simulation code
      - ex of concrete model for observations : Simple Spectral DataModel

micro services

- micro service 1
    - get available semantic tags **like “proton density”** for **project P**
- micro service 2
    - get models from project P producing objects with **“proton density” < T**
- micro service 3
    - get the **datalink file** for a specific model

micro services

- composite service
    - get **datalink file** for models from **project P** having calculated **property matching “proton density”** string **< T**
- orchestrate : ms1 -> ms2 -> ms3

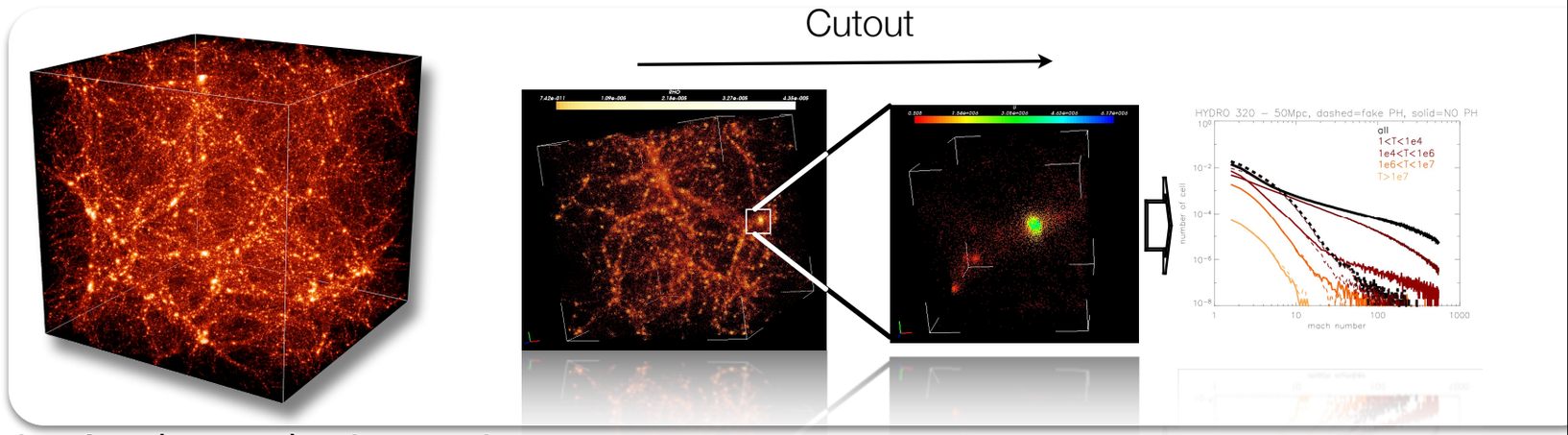
micro services

- do 1 small thing well
  - simple api (simple restful service)
- ~~easy~~ very easy to debug / maintain
- composable / reusable
  - layered web service system

micro services

- TAP based services (ADQL)
  - **tightly coupled to Relational Database Systems**
    - bottleneck number of columns
    - bottleneck query complexity, EAV pattern
  - TAP compliant service is not trivial to setup
- API based services
  - **loosely coupled with implementation details**
    - specific technologies can be used for specific problems
    - more user/developer friendly

generalized space cutout

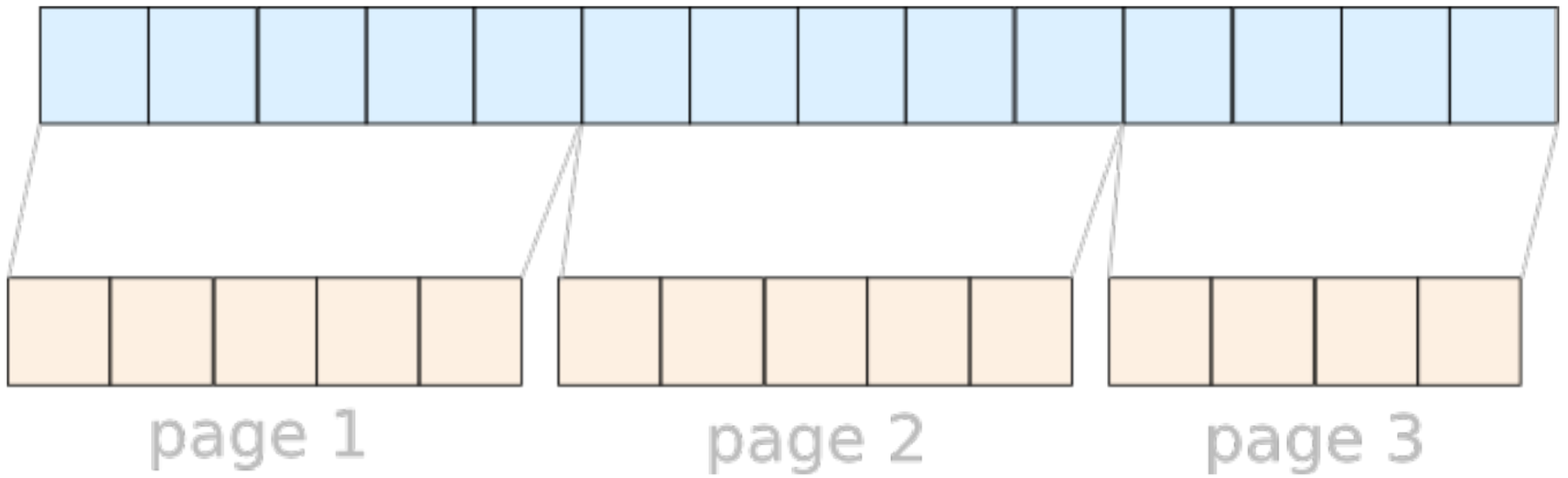


- common specific case
  - (x,y,z, property)
  - get cube inside the whole (x,y,z) domain

- general case (numerical simulation calculated objects)
  - (property1, property2, …)
  - **get hyper-cube inside the whole properties space**
  - ex : cutout on (mass, velocity, temperature)
  - generalized cutout on any axis system through simple query language “ala” sql

pagination

- **split potentially infinite collections into pages**



- manageable amount of memory & network latence. **Allows interactive interface**

pagination

- display a huge query result through a web interface
  - page 1 from result 0 to result 10
  - page 2 from 10 to 20 …
- The user very rarely goes farther than the first pages. **Avoid unnecessary backend work**

Google virtual observatory

Web Images Maps Shopping More Search tools

About 5,850,000 results (0.11 seconds)

Cookies help us deliver our services. By using our services, you agree to our use of cookies.
OK Learn more

The European Virtual Observatory EURO-VO | Euro-VO
www.euro-vo.org/
The **Virtual Observatory** is an international astronomical community-based initiative. It aims to allow global electronic access to the available astronomical data ...

Virtual observatory - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Virtual_observatory**
Virtual observatory (**VO**) is a collection of interoperating data archives and software tools which utilize the internet to form a scientific research environment in ...

IVOA
www.ivoa.net/
The **Virtual Observatory** (VO) is the vision that astronomical datasets and other resources should work as a seamless whole. Many projects and data centres ...
Documents - IvoaEvents - Using the VO - VO Glossary

Searches related to **virtual observatory**

virtual observatory **tools**	**astrophysical** virtual observatory
virtual observatory **software**	virtual observatory **standards**
science virtual observatory	virtual **astronomical** observatory

Goooooooooogle ›
1 2 3 4 5 6 7 8 9 10 **Next**

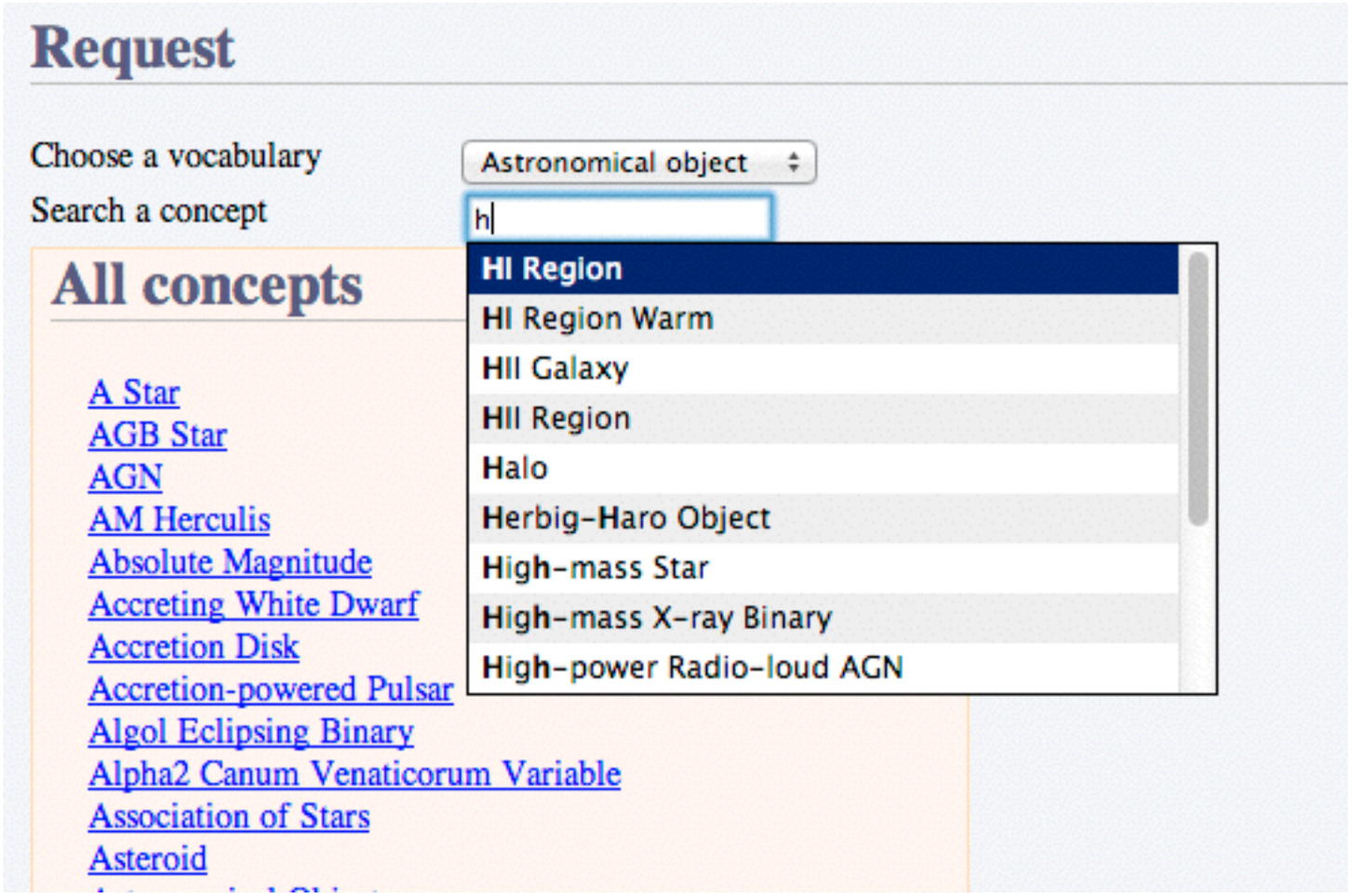
pagination

```
Simplified SimDM description

{
    property1 : atom,
    property2: object,
    property3 : list
}


Simplified SimDM description instance example

{
    property1 : "atom1",
    property2 : {
        property21 : "atom21",
        property22 : "atom22"
    },
    property3: [
        {property31: "atom31", property32 : "atom32"},
        {property31: "atom31", property32 : "atom32"}
        ...
    ]
}
```

semantic tagging

- user1 calls “halo mass” *massh1*
- user2 calls “halo mass” *masst2*
- How do we achieve consistency ?
  - semantic tagging from standard vocabulary
- How do we deal with very large amount of metadata/ semantic tags ?
  - smart autocomplete VS infinite length select box

autocomplete



http://votheory.obspm.fr

(Nicolas Moreau)

Conclusion

- Hide implementation details : API/Rest VS ADQL
- splited, manageable collections
- small, manageable pieces of software
- standard exchange format : datamodel serialization
- generalized interface to extract cube of any space