All-VO registry in Aladin v10
The good, the bad...

IVOA Interop – May 2017 - Shanghai

Pierre Fernique
On behalf of the Aladin CDS team,
What's the plan?

1) Aladin and the VO registry
2) Evolution from v9 to v10
3) News possible use cases (demo)
4) Technical choices
5) The good, the bad, ...
Version 9

Interop - May 2017 – P.Fernique
1) Vo accessible only via “All VO” dedicated TAB

2) Only CS and SIAv1 + internal predefined Aladin collections (GLU dictionary)

3) Poor metadata information => only filtering on title and description

4) No space index => real queries on all selected servers to build the list of available collections in the user field
Version 10

Data Access Tree

Access selector
Version 10

Demo
1) “Data collection” oriented (not only “providers oriented”) => more intuitive

2) In the **main panel** => high visibility

3) **All** data: CS/SIAv1/SIAv2/SSA/TAP/HiPS VO data collections = 19 000 collections) => exhaustivity

4) One **integrated data access tree** => useful

5) Extremely **fast** (ms) => dynamicity
“Bring metadata next to the client”
VO reg & MocServer & Aladin v10

VO registry ➔ alasky@cds ➔ MocServer ➔ Your laptop

alasky@cds

Your laptop
One key element

MocServer

= Tomcat servlet containing a list of (MOCs+properties)

= metadata dictionary, queryable via URL, by criteria (key words and/or regions)
- Aladin v10 **innovation**: copy the registry content on the client
- Implementation of a **“local” MocServer** (same java classes)
- **Synchronized** each time Aladin starts with the “remote” MocServer localized at alasky@csd
- Contains **only 20 000 “properties” records** and not MOCs (5MB gzipped)
  + possible MOCs and properties record added locally
VO reg & MocServer & Aladin v10

Regtap server

alasky@cds

Your laptop

VO registry

XML 25MB

MocServer

Props 25MB
VO reg & MocServer & Aladin v10

Regtap server → alasky@cds → MocServer

MOCs 1GB Props 25MB

Other servers → VizieR → SIMBAD

Your laptop → Aladin
Aladin v10 internal architecture

MocServer

MOCs 1GB

Props 25MB

http://… ? update props at startup

Your laptop

scan (which collections by criteria)

MocServer

Props 25MB
Aladin v10 internal architecture

MocServer

MOCs 1GB

Props 25MB

http://... ? which collections in current view

Your laptop

scan (which collections by criteria)
Aladin v10 internal architecture

MocServer

MOCs 1GB

Props 25MB

Your laptop

scan(which collections by criteria)

http://... ? which collections in current view

+ 18

Props

25MB

MOCs

props
The good for Aladin 10...

1) **It works !**
   => probably used a lot compare to “All-VO” v9 tab
   => Thanks to RegTAP and MOCserver
   => see full demo in Apps2.

2) Thanks to **RegTAP** and **MOCserver**

3) It provides a high level of publicity for any VO collections
   => *should encourage the declarations in the VO*
The bad for Aladin 10...

1) **Inappropriate description level** in VO registry:
   ex: Aladin manipulates tables (homogeneous set), and not catalogs (pb on capabilities, IDs, descriptions, coverages, ...)

2) **Lack of meta data** information:
   ex1: Time, energy, space coverage fields rarely filled up
   ex2: No direct relation between table VO registry descriptions and the associated TAP schemas
Open questions ...

Exhaustivity also means:

1) Do we have to **filter the “bad” collections**
   (prototypes, not really maintained, bad data...)

2) **Which classification** should be used ?
   (by quality ? by data type ? by protocol ?
   by origin...)

=> Part of the response is probably political
Thanks!
Questions?

For testing the prototype
http://aladin.u-strasbg.fr/java/AladinProto.jar