

# Meeting notes

2014.12.02

# 3 parts design

- clarify the independence thing
  - list of what should be provided (xml files, service, ...) by user, provider, publisher in the standard, as examples. Herve suggests a table
  - Link components through VOTable LINK element towards parent (protocol.xml / project.xml) in other component.

# Repository harvesting

- How to deal with freshness / version / last-update of retrieved data ?
- See how Registry deal with it ?
- -> move it to next SimDAL version & let the design to registry Group

# SimDAL registration in Registries ?

- Repository must be registered in registries
- Search also registered in registries because no longer dependency with Repository thanks to <LINK> to protocol.xml in Votable header (view schema).
- In particular for providers having only a Search/DataAccess service

# Experiment(s).xml in Search service ?

- Optional
- Available values in VOTable <VALUES> elements : gives all the necessary meta from experiment.xml to query the view

# Response format

- Json / VOTable ?
  - VOTable mandatory everywhere: comply with current IVOA practice
  - Json optional

# Json as input POST

- Marco suggests giving inputs to François for AccessData
- Marco suggests passing the input to DataAccess cutout as a POST query=`<json txt query>` instead of a json document to be compatible with AccessData way of passing input.

# Pagination

- Define our own standard for pagination
  - Problem of maximum number of page reload in browser ? -> Herve -> move it to implementation note of the std
  - allow optional special keyword “ALL” for per\_page, in case the user wants to get the whole result (and knows what he is doing)
- Define VOTable serialisation (and json ? -> not for now)
- Provide total number of pages when possible (i.e it's optional for the service)
- Provide ORDER BY parameter in the pagination api!!!



# TAP compat

- ADQL support for v2 ?
  - We postpone that for when Gerard will be available -> that is his suggestion back at Madrid intro

# Sync / Async

- DataAccess
  - both optional, but at least one must be defined
- Search
  - At least one, both optional
  - think not only to data size but also to large processing time for /async use case -> see SimDAL draft document (§ “async”, part 6.2)

# Summary/stat feature for cutout

- Think about it

# Search api endpoints

- `/views`
- `/views/metadata/search`
- `/views/:view/metadata`, `/views/metadata?view=:view`
- `/views/:view/metadata/:field`
- `/views/:view/metadata/search`
- `/views/:view/cutout`

# Schedule

- Dec14: FLP: Get Gerard feedback
- Dec14: DL: Francois / link to AccessData
- Jan14: ALL: updated document from meeting feedback
- Feb15: CR: Design / Fitting summary feature
- Feb15: DL: Pagination
- Mar15: ALL: updated doc
- May15: WD to doc coordinator
- Jun15: ALL: Iterate document, ML