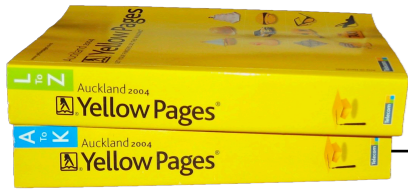# Linking capabilities with tablesets: an update
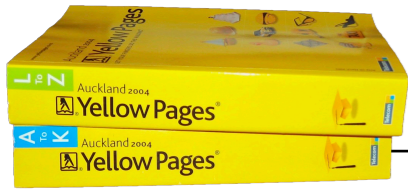
Renaud Savalle (PADC/Obs Paris) and Gilles Landais (CDS)

2024-05-21 - Registry WG
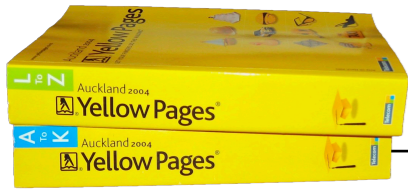
v1.1

Outline

‣ What are <capability> and <tableset> in VOResources

‣ CDS / Vizier problem and proposal

‣ Report of Running Meeting
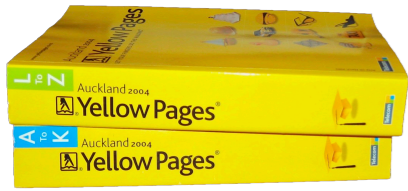
‣ The mini-resources pattern

‣ Discussion ?

<capability> and <tableset> in VOResources

Initial discussion and Running Meeting

‣ http://mail.ivoa.net/pipermail/registry/2024-March/005572.html
CDS: "We didn't find any clean method to link capabilities with tablesets described in a VORegistry record."…

‣ IVOA_Reg_RM20240506.pdf
- CDS described a new proposal
- A lively discussion followed

CDS proposal to link tablesets with services in the registry

Issue: one resource in registry possibly contain a collection of datasets and a collection of services but it is not possible to attach a service to the datasets.

→ A general issue in VizieR, Esa, Ukidss, etc.

example: ivoid: ivo://CDS.vizier/i/355

Consequences: clients (eg: PyVO, registry module) are not capable today to link properly the datasets with the services

Examples :

- Vizier catalogue:
  https://cdsarc.cds.unistra.fr/viz-bin/cat/J/AJ/165/45

- VizieR Notebook:
  https://cdsarc.cds.unistra.fr/vizier/notebook.gml?source=J/AJ/165/45

Proposal: Update VOResource
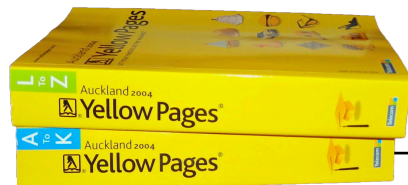
```
<capability xsi:type="cs:ConeSearch"
      standardID="ivo://ivoa.net/std/ConeSearch">

      <serves>
          <reftable name='table1' />
          <reftable name='table2' />
      </serves>

</capability>
...
<tableset>
      <schema>
          <table>
              <!-- todo ADD CONTENT-TYPE HERE -->
              <name>table1</name>
....
```
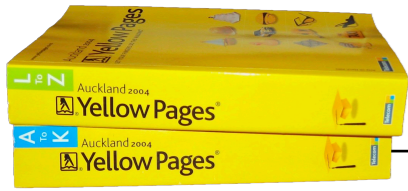
Don't break the current framework

Credits: Gilles Landais, CDS

The Resource Granularity Problem

‣ "collections" in VOResources seem to be the problem
  - Legacy: Vizier Catalog <=> VOResource: 1-1 mapping
  - History: VOResource first written **manually**: factorization of metadata

‣ And also…
  - High granularity VORs have caused a similar problem with TAP before
  - Solved with #aux cap - cf Endorsed Note <u>Discovering Data Collections within Services</u> (2019)

‣ BUT
  - Such large VOR tend to "hide" their details, hindering discovery efforts:
    Ex: which table(s) ? is concerned by a IsServedBy relationship?
  - Redundancy not an issue since VORs are programmatically **generated**

The "mini-VOResources" pattern

‣ Split such "large" VOResouces into smaller ones
  - Associate *closely related* <capabilities> and <table>s
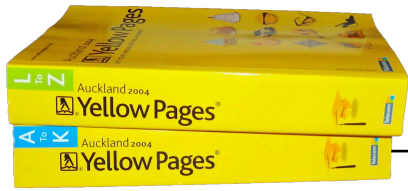 - Limit case (SCS): 1 <capability> and/or 1 <tableset> with 1 <table>

‣ Benefits
  - No change in Standards, just a "new" usage pattern
  - Solves CDS issue: a service is naturally attached to a (set of) table(s)
  - Several VOR elements become more accurate (as they apply to one 1 capability/table): <source>, <altIdentifier>, <coverage> etc.
  - Useful when we'll add <productTypeServed>

‣ *Closely Related* ?
  - Can discoverability drive this criteria ?
  - Ex: Tables describing *similar kind of* data (same productTypeServed) ?
  - NB: "~~Tables which can usefully be joined together~~: not a good criteria

Impacts of using "mini-VORs"

‣ For existing Publishers:
- Discoverability will increase only when you "upgrade"
- Upgrade could be costly for some (design new publishing layer)
- Some ivoids may change (OK, they are not PIDs)
+ Need to use new <relationship>s to GROUP mini-VORs together

‣ For Users:
- They will see more resources in results
+ They can discover better "labeled" data more easily

‣ For Harvesters:
- may see same DOIs attached to several resource. Is it an issue ?

‣ For Clients Apps and their ADQL queries:
- They might have to display related VORs using <relationship>s

Conclusions

‣ Consensus:
  - Discoverability is our challenge, and should drive changes
  - Small resources ease data discovery

‣ CDS will explore the impact of their proposal on discovery queries

‣ Maybe it's not necessary to modify the schema…

…but the way we use it ?