



Fig. 1

# 1. Data Discovery in pyVO

Markus Demleitner  
*msdemlei@ari.uni-heidelberg.de*

- The VO Registry and RegTAP
- Motivating the change
- The new API

(cf. Fig. 1)

## 2. The VO Registry

A collection of highly structured metadata (Dublin Core plus service infos plus table infos) on  $\sim 26'000$  resources ( $\approx$  data collections) with  $\sim 70'000$  tables.

Publishers usually run OAI-PMH endpoints to let everyone harvest their metadata records.

STScI, ESAC, and GAVO do that in order to feed RegTAP services.

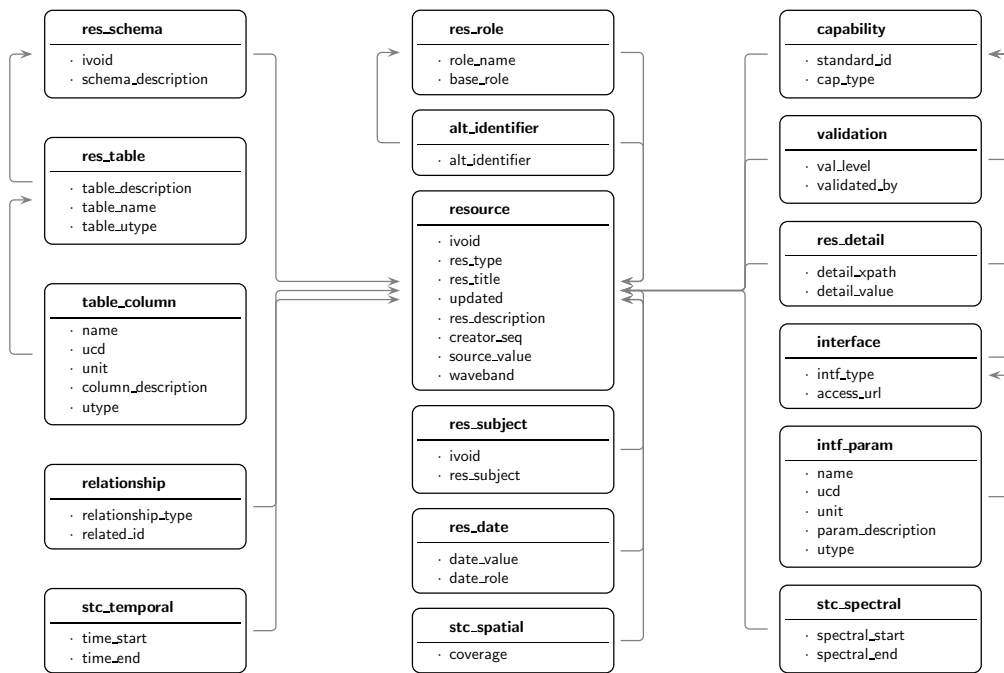


Fig. 2

### 3. RegTAP

RegTAP is 13...20 tables containing a relational mapping of the Registry contents queryable with ADQL and TAP.

(cf. Fig. 2)

The tables are designed such that you can look for the metadata items you want to retrieve and/or constrain and simply NATURAL JOIN the tables you need. For instance, to find the URLs (retrievable from the interface table) for TAP services (constrainable in capability) having data for M32 (constrainable in stc\_spatial) created by a Mr. Copernicus (constrainable in res\_role), you would write something like

```
SELECT access_url
FROM
  rr.interface
  NATURAL JOIN rr.capability
  NATURAL JOIN rr.spatial
  NATURAL JOIN rr.res_role
WHERE
  [details irrelevant]
```

Yes, this assumes people are writing ADQL. Perhaps surprisingly, in practice the data model is a larger problem than the ADQL.

See also Demleitner, M., Harrison, P., Taylor, M., and Normand, J., 2015: *Client interfaces to the Virtual Observatory Registry*, *Astronomy and Computing*, **11**, pp. 91-101.

## 4. pyVO Registry Before

The legacy pyVO API to the registry has essentially been:

```
registry.search(keywords, servicetype)
```

```
-> service-descriptors
```

– when you wanted infrared spectra, you would look for, roughly, “SSAP services with infrared”.

That’s not good enough any more: Services and data collections are *really* not 1:1 any more (extreme case: several  $10^4$  tables within the single TAPVizieR service).

Also, with libraries like pyVO you don’t care too much any more whether you get something through TAP, SCS, or SIAP, as long as you get it.

## 5. pyVO Registry After

The idea is that you combine Constraints and get back registry records that you can ask about how they can be queried.

Try it:

```
http://blog.g-vo.org/media/2022/data-discovery-demo.ipynb
```

## 6. Parting Words

The code discussed here will probably end up in pyVO release 1.4 (or so). I’d be great if you could exercise it a bit before then.

<https://blog.g-vo.org/towards-data-discovery-in-pyvo.html> tells you how to do this without nuking your system’s pyVO.

... Thanks!