Fig. 1

Fig. 2

1. IVOA Registry for Data Providers

(cf. Fig. 1)

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

(cf. Fig. 2)

- Why have the Registry?
- Does it work?
- Resource Records…
- …and how to get them in.

2. Why Registry?

Imagine ADS went down for good.

How would you find you literature? Go to each publisher's web site separately? And click around it until you've figured it out? Or use Google and wade through junk from the wider internet?

That's been the situation for data publishing before the Registry. Except…

- …it was worse because there's many more data publishers than journal publishers
- …few worried about it because everyone had their pet data collection and rarely looked further. And there was already VizieR.

3. Does it work?

Check out TOPCAT……or WIRR……or SPLAT.

4. Resource Records

Per resource (i.e., thing you want registered), there's one **resource record**.

Physically, it's a heap of ugly, **pedantic XML**.

Logically, it's a **beautiful metadata structure** containing:

- Title, authors, publishers, contact in case of trouble.
- Human-readable description, subject keywords, related publications
- Coverage in space, time, and spectrum
- Table and column metadata of material exposed
- Access modalities of one or more services ("capabilities")

5. Identifiers

The Registry essentially is a database of such resource records.

The primary key into this database is the IVOA idenifier (IVOID):

```
ivo://<authority>/<resource-key>[<local-part>]
```

The **resource-key** tells apart different resources you operate.

The **local-part** you probably only care about if you're assigning dataset identifiers.

The **authority** is what you need to agree upon with everyone else.

6. Authority

The IVOID is a primary key and therefore must be **globally unique**.

To let you assign IVOIDs as you see fit, you'll have to **claim an authority**; nobody else can then make IOVIDs with it.

Claim an authority = register a record `ivo://<authority>`.

To make things even more complex, an authority also needs to define an organisation that manages the authority. The distinction is a bit subtle; the upshot is, though, is if you're going to publish in your own name, you'll usually create two resource records before even starting out.

7. Authorities

In case you're planning to have your own authority, you'll have to **choose a name**. There are no fixed rules for this, but to give you an idea what others did:

- omegacen – just a project name
- org.gavo.dc – Java-style inverted DNS name
- byurakan.phys.uniroma1.it – plain DNS name
- vopdc.obspm – hierarchical by institute
- ar.nova – hierarchical by parent project

As long as you keep away from names other people might plausibly claim ("astro"), it doesn't really matter; keeping the authority short helps keeping IVOIDs short, which is a good thing.

All authorities present in the current VO can be obtained by running

```
select * from rr.authorities
```

on the TAP service at `http://reg.g-vo.org/tap`.

8. A Record for your Resource

Unless you're developing your software, do *not* write your resource records from scratch.

For few-resource sites, use a **web-based form interface**. http://registry.euro-vo.org/insert.jsp

Many-resource sites want a **publishing registry** of their own. Publishing toolkits like DaCHS contain everything you need for that.

We're considering offering **harvesting records** directly from services for intermediate sites. If you'd like to go with that, talk to me.

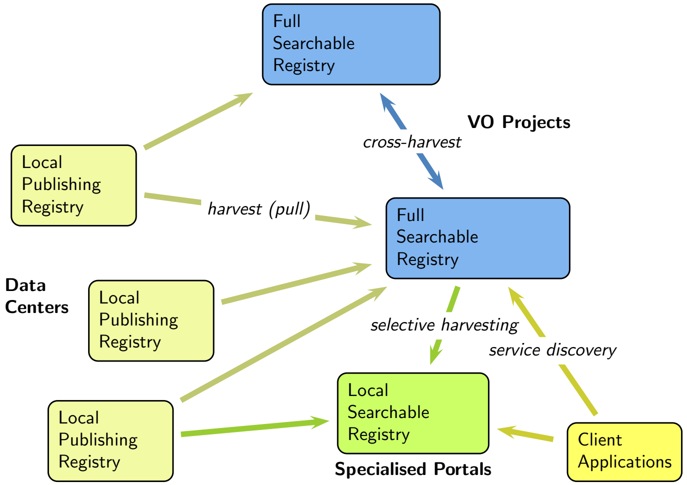

Fig. 3

9. Behind the Scenes

(cf. Fig. 3)

The Registry is distributed, which always makes for a bit of complexity.

You won't be concerned with more than two of these arrows.

10. Conclusion

If you're not in the Registry, clients won't (easily) see you. The Registry is a distributed database, which always makes for a bit of complexity. Fortunately, we're here to hold your hand as you discover what's behind it.