DAL Running Meeting #19

Date: Wed. 11 December 2024 - 04:00 PM UTC

Participants: Tess Jaffe (TJ), Anastasia Laity (AL), Markus Demleitner(MD), Grégory Mantelet (GM), Joshua Fraustro (JF), Pat Dowler (PD), Mark Taylor (MT), Dave Morris (DM), Renaud Savalle, Abdu Zoghbi, Jon Juaristi Campillo, Kim Gillies

Useful links

- Tess' Talk in Nov. 2024 Interop
- Anastasia's talk in Nov. 2024 Interop
- DAL session notes
- Email from François Bonnarel *(see below)*

Meeting notes

- **Preferences (according to the Interop session):**
  - **PD** and **MT** (running meeting): Service descriptor
  - **MT** (interop) and **François B.**: `local-semantics`
  - **TJ:** new column (`cloud_access`)...but likes it less and less
  - **MD:** Semantics (`#this-secondary`?) and perhaps extra info in `local_semanics` (but won't people know from the URIs?)

- **PD:** multiple Datalink interrogations work surprisingly fine
  - **MD:** don't use the implementation in PyVO (bad batch management)
- **MD:** What is the functionality the client is supposed to effect? Use the links and choose where they want to go manually?
  - **TJ:** Either, it's offering the option. Either a user can do it themselves, or it could be automated by preferences.
  - **MD:** If it's done automatically, we wouldn't need to specify the link type (cloud AWS, etc.)
  - **TJ:** We don't want them looking at the URL's... It depends on what the client wants. Some people want to know what cloud type-- give them the option to pick.
- **DM:** As machine-readable as possible. This is useful for ExecutionBroker-- folks might want to pick the AWS region something executes in.
  - **MD:** That means we'd need to enumerate vocabulary-- what's the minimum we can get away with? Just needs to be written down somewhere, note, standard, and clients need to agree. But avoid defining as many vocabulary terms as clouds and locations combinations.

- **PD:** If you call datalink and get multiple results, there's nothing currently to suggest anything other than that these are all the links to download. If presented with two links, it implies a user has two things to download. A client that doesn't understand differing local semantics will use both.
    - Introduce `local_semantics` "package", contains all the alternative links
    - We need a way to convey relation-ship between links with the same semantic
- **MD** believes multi-`#this` has always meant "alternatives"; and I don't think there is a client yet that would actually pull all liks. But yes: We have to figure out and define what multi-`#this` means.
- **MT:** Datalink currently underdescribed. Local semantics currently something the user interacts with. Service descriptor probably the best way to proceed-- availability of descriptive text for users to view
- **MD:** Multiple links has always been implied as alternatives. It might be time to really delve into what does it mean to have multiple links with local semantics. Don't think clients currently care about multiple links (interpreting as multiple downlaods).
    - **PD:** I don't think we can practically say 'id' means 'alternatives'. Some data providers might provide data split into multiple files.
    - **MD:** Perhaps another column with some values that imply alternatives/grouping.
- **AL (in chat):**
    - Sample `cloud_access` column at IRSA (SIA results):
    - Let's the user/client see what region, and potentially the access policy, in case those things matter to their choice

```
{
  "aws": {
    "bucket_name": "nasa-irsa-spitzer",
    "key"        :
"spitzer/seip/seip_science/images/6/0095/60095931/2/60095931-
12/60095931.60095931-12.MIPS.1.median_mosaic.fits",
    "region"     : "us-west-2"
  }
}
```

- **PD:** If we can imply/denote that links are alternatives, what can we embed in `access_url` to imply host data (provider/store/region etc.)?
    - **TJ/AL:** A need to describe access policy / protected locations, etc. separately from the URL. (User might need to authenticate to access the data.
    - **PD:** There are other columns in datalink to denote authentication needed. A user might need different credentials, but it at least informs them some auth is needed. Put the blob of information in the `access_url` or something else. (**TJ:** something else.)
    - **TJ:** Some NASA desire to have a gateway/layer over the direct link; they shouldn't know the exact S3 bucket name.
    - **PD:** If you give a client an HTTP, can they use S3 specific packages?
        - **JF:** I think boto3 is smart enough to do the handshake to know it's S3 underneath.

- **MD:** Assuming we can do the description of grouping with extra table lines, etc. How do we include the extra metadata (e.g. localisation, policy ; specific cloud parameters in this example) in the response? (See Anastasia example above)
- **DM (in chat):** Extra complexity - you could have replicas in two AWS regions that would have the same download URL.
- **DM:** Good idea to just try something for awhile and see what happens-- don't call a provider "aws" for now, maybe "irsa-aws" to allow experimentation for a time. The metadata also doesn't need to be explicitly written for now. Many platforms use the S3 'standard' for interacting with stores (linode, openstack, etc.)
- **JF:** Just use the terminology that the providers use (S3, Google, etc.) and don't try to standardize across everyone.
- **MD:** Given current vagaries, perhaps best to use an extra column for metadata for now, maybe doesn't even need to be defined for now. Current step to address is denoting the grouping of links in the results. "For this group of this things, it's all the same." A 'link group' column-- all the results with the same group are alternatives.
    - **MT:** If you have one row with all the access locations in that column, you don't need multiple `#this`, `#this-aws` rows.
    - **PD:** The extra column/metadata doesn't yet address that they alternatives in location.
- **GM:** If we have multiple services that are interested in this method of denoting cloud locations, should we publish a note on what current implementors are experimenting with?
    - **TJ:** Close to having this implentation working for datalink service. Not data discovery yet.

- **Conclusion/Agreements:**
    - For her use-case, TJ will add an extra custom column to denote locations ; it is fine according to the standard to add custom columns. This is an immediate solution, standard-compatible, to her use-case while waiting for a standard way to declare alternative links (see below)
    - On Datalink side, add a clarification about what happens to multiple links with the same `semantic` value and add a possibility to have alternative links.
        - A GitHub issue has been created for this task: DataLink#117

François Bonnarel's email (just before the meeting)

> Dear all,
>
> I will not attend the meeting, but here are my 2 cents on the questions raised
>
> a. DAL at IRSA :
>
> - in Obscore there is a distinction between "observation" identified by obs_id and the individual unit "dataset" identified by obs_publisher_did
>   This allows to group by a common obs_id several datasets considered to belong to the same "observation". The definition of what is an observation is actually provider dependent. SO in the mapping with CAOM I guess that planeid is more aligned with obs_publisher_did and the dataset concept.

- for "fast" DataLink as far as I understand the question, the ID parameter in the DataLink service is not required to be obs_publisher_did, so any more direct index to the CAOM artifact table can be used there.

b. How should ObsTAP+DataLink provide options to retrieve the same product from multiple locations (including cloud)?

- prefered provider : the DataLink access is not required to be given by the access_url FIELD (or equivalent) in a data discovery response. It can be provided through a DataLink service descriptor as a secondary RESOURCE in the respeonse. In that case the access_url can provide direct link to the preferred provider.
- multi provider issue : with the current standard, I think providing the provider tag in the local_semantics is the "less bad" solution if not the better. local_semantics is a free text FIELD, but should repeat the same strings for different datasets in the main service in order to identify related or similar raws in the DataLink response. So something like

```
ID = id1
semantics : #this
content_type : application/fits
content_qualifier : image
local_semantics : aws
............
ID = id1
semantics : #this
content_type : application/fits
content_qualifier : image
local_semantics : google
............
ID = id1
semantics : #progenitor
content_type : application/fits
content_qualifier : cube
local_semantics : aws
............
ID = id1
semantics : #progenitor
content_type : application/fits
content_qualifier : cube
local_semantics : google
............
ID = id1
semantics : #progenitor
content_type : hdf5
content_qualifier : cube
local_semantics : aws
............
ID = id1
semantics : #progenitor
content_type : hdf5
content_qualifier : cube
local_semantics : google
```

and then the same kind of answer for ID = id2

> will allow the client : to find out all accesses to the full (image) fits format dataset served by aws (or google or ...) to find out all accesses to the (cube) progenitor in fits (or hdf5) served by google (or aws ...) etc....
>
> Cheers
> François