TAP-1.1 review

Patrick Dowler
Canadian Astronomy Data Centre

Data Access Layer - Wed May 15

TAP-1.1

- things were removed
- clarifications were made
- datatype mapping changes
- other tap_schema changes
- VOSI and UWS related changes
- TAP, Authentication, and Authorization

TAP-1.1

- things were removed
  - all mention of PQL, including examples
  - REQUEST parameter, including examples
  - VERSION parameter

- clarifications
  - QUERY parameter can be re-used for other values of LANG
  - defer to other standards (ADQL, DALI, TapRegExt)
    - required vs optional ADQL geometric functions
    - datatype support and formats
  - interaction between MAXREC parameter and use of TOP within the query (ADQL)
  - use of delimited identifiers, including “size” in tap_schema
  - valid table names for UPLOAD
  - multiple UPLOAD(s) accumulate

TAP-1.1

- datatype mapping
  - added **arraysize** and **xtype** to tap_schema.columns
  - enables consistent use of VOTable+DALI type system: **<datatype,arraysize,xtype>**
  - removed suggested/explicit mapping of TAP to RDBMS types (now: implementation detail)

- other tap_schema changes
  - added **schema_index**, **table_index**, **column_index** to tap_schema
  - enables service-recommended presentation order

TAP-1.1

- UWS related changes
  - async implements UWS-1.1 or later
  - examples include use of blocking WAIT

TAP-1.1

- VOSI related changes
  - tables implements VOSI-tables-1.1 or later
  - availability resource decoupled from TAP base URL (DALI)
  - prefer a single interface for TAP capability
    - accessURL (mirrorURL) is the base URL
    - 0..* securityMethod that co-exist on that URL
    - this usage is consistent with SSO-2.0
    - proven to be implementable

- **previously**: VOSI-capabilities described the way a provider deployed their service (flexibility)
- **now**: we are saying saying more about how they deploy their service (more restricted)

- TBD: multiple interfaces: forbidden? discouraged? allowed?

TAP-1.1

```
<capability standardID="ivo://ivoa.net/std/TAP">

   <interface xsi:type="vs:ParamHTTP" role="std" version="1.1">

      <accessURL use="base"> https://example.net/srv </accessURL>

      <!-- anonymous or cookie or client certificate -->
      <securityMethod/>
      <securityMethod standardID="ivo://ivoa.net/sso#cookie"/>
      <securityMethod standardID="ivo://ivoa.net/sso#tls-with-certificate"/>

   </interface>

 </capability>
```

TAP-1.1

- TAP & Authentication
  - CADC operates all services (VO and custom) supporting both anonymous and authenticate access > 10 years
  - we have supported up to 5 different authentication methods
  - have tested that common web servers/front-ends can be configured to support multiple* optional authentication methods on a single URL: apache, tomcat, nginx, haproxy
  - SSO-2.0 states that username-password auth is to be used to “login to a session” aka “acquire a token” and not for direct access

- **current gap: where to get the credentials?**
  - client certificate: a valid CA (not all issuers in default ca-bundle)
  - token? TBD
  - BUT: authentication has wider scope and is loosely coupled

TAP-1.1

- TAP & Authorization experience at CADC
  - minimal: authenticating gives authorization to see your own jobs (UWS job listing)
  - YouCat (unregistered prototype):
    - table-level owner and group permissions
    - anon can only see public tables
    - authorized to see public & non-public tables
  - archive storage system (unregistered):
    - row level permissions: based on rules
    - anon can query tap_schema
    - query result limited by modifying ADQL → SQL
  - archive metadata service (unregistered):
    - service-level group permissions
    - no anon access

TAP-1.1

- TAP & Authorization at CADC
  - CAOM allows one to specify public/proprietary metadata & data
  - primary (CAOM) TAP service (registered):
    - row-level public and group permissions
    - query results limited by modifying ADQL → SQL

- what is authorized varies
- results can be subtle
- our experience in other services
  - permission grants live with the resources they protect
  - help users / self-serve: expose the permissions that are in effect via service API (VOSpace: yes - TAP: working on it)