NRC·CNRC

Integrating Anonymous & Authenticated Access to VO Services

Patrick Dowler

Canadian Astronomy Data Centre

2016-05-10

Authentication at CADC

- anonymous access
- multiple authentication mechanisms supported
  - TLS with client certificate (X509 bare and proxy certificates)
  - username and password (because users want that)
  - cookie (token in the web browser)
  - delegated token (token + limited scope + expiry that can be ~safely shared)
  - adding support for other external identity providers like EduGain, OpenID, etc: conceptually straightforward because “users” have multiple identities

Endpoints

- for each capability, we provide multiple endpoints with different authentication mechanism
  - capability – interface – securityMethod (VOResource / VOSI)
- we always use the common/standard endpoints for anonymous access
  - e.g. /tap/sync, /tap/async, /tap/tables
- and an alternate path for username+password
  - e.g. /tap/auth-sync, /tap/auth-async, /tap/auth-tables
- currently only change protocol to https for tls-with-cert; would prefer to use alternate path as well
  - e.g. /tap/x509-sync, /tap/x509-async, /tap/x509-tables
- then could provide http and/or https where applicable

VOSI: capability - interface - securityMethod

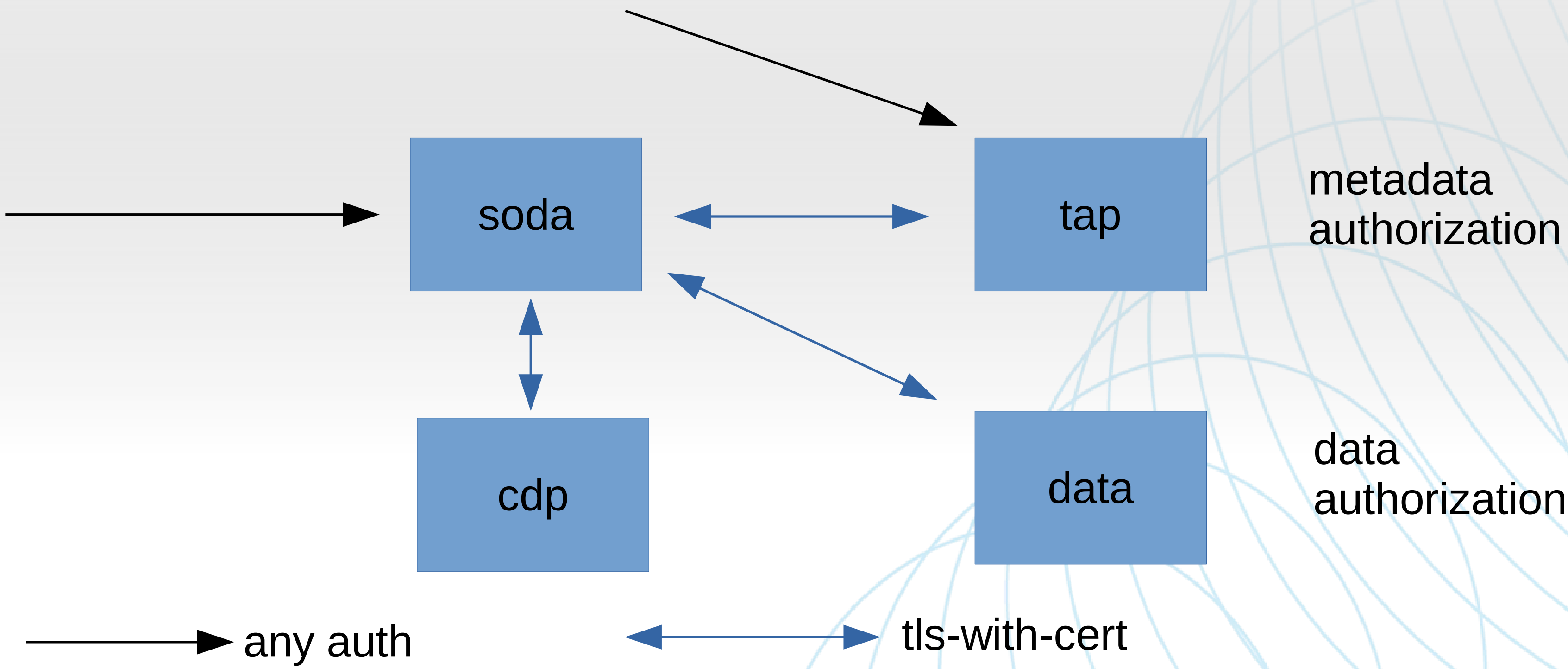
```
<capability standardID="ivo://ivoa.net/std/DataLink#links-1.0">
    <interface xsi:type="vs:ParamHTTP" role="std" version="1.0">
      <accessURL use="base">http://www.cadc.hia.nrc.gc.ca/caom2ops/datalink</accessURL>
    </interface>
    <interface xsi:type="vs:ParamHTTP" version="1.0">
      <accessURL use="base">http://www.cadc.hia.nrc.gc.ca/caom2ops/auth-datalink</accessURL>
      <securityMethod standardID="http://www.w3.org/Protocols/HTTP/1.0/spec.html#BasicAA" />
    </interface>
    <interface xsi:type="vs:ParamHTTP" version="1.0">
      <accessURL use="base">https://www.cadc.hia.nrc.gc.ca/caom2ops/datalink</accessURL>
      <securityMethod standardID="ivo://ivoa.net/sso#tls-with-certficate" />
    </interface>
</capability>
```

Why authenticate users?

- so the service can authorize access to non-public resources, e.g.
- data (files)
  - distribute proprietary archive files to PI and collaborators
  - archive pipeline processing running on CANFAR cloud
  - telescope staff / operations support
- metadata (rows in database tables)
  - observatory policies require this to protect PI projects (rare)
- resources created/owned by users: jobs, user info, group info, credentials, files and directories (vospace), …
  - users control the permissions on resources they create

How a typical service call works

- all CADC and CANFAR services are callable by users
- services call other services using identity of the user



NRC·CNRC

Implementation – Java specfic – implemented throughout OpenCADC codebase

- every request run in an AccessControlContext:

```
Subject s = AuthenticationUtil.getSubject(httpRequest);
Subject.doAs(s, handleRequest(httpRequest));
```

- down in the code where you need to perform authorization check

```
AccessControlContext acc = AccessController.getContext();
Subject subject = Subject.getSubject(acc);
// check if subject is allowed to do it…
```

- Subject.doAs(…) also applicable in client applications
- does not mess up API: can be added to existing code because caller identity never part of functional APIs or call stack!

Summary

- authentication supported by existing VOResource + VOSI-capabilities
  - model of 1 capability → N interaces with different securityMethod: correct, implementable, works
- concept of external identity providers (currently: CA) crucial to global reach and the dream of Single Sign On
- CADC uses this so that core infrastructure (TAP, SIA, DataLink, SODA, CDP) are VO services that satisfy all archive data centre use cases
  - including proprietary data access for PIs and collaborators
  - and public data access for everyone