Authentication

context, requirements, strawman proposal

Context

- VO services currently describe supported authentication methods via IVOA registry records and/or VOSI-capabilities documents

```
<capability standardID="ivo://ivoa.net/stc/TAP" version=”1.1”>
  <accessURL> https://example.net/query </accessURL>
  <securityMethod />
  <securityMethod standardID="ivo://ivoa.net/std/SSO#cookie"/>
  <securityMethod standardID="ivo://ivoa.net/std/SSO#tls-with-certificate"/>
  <securityMethod standardID="ivo://ivoa.net/std/SSO#OAuth"/>
  <securityMethod standardID="http://example.net/auth#token"/>
</capability>
```

- this does not convey where to get the cookie, OAuth token, or custom token
  - the dream of SSO-1.0 was that everyone had a personal client certificate from a trusted CA

Requirements

- How does a service tell users (client software) **how** and **where** to **log in?**
- must convey:
  - **a URL for login endpoint(s)**
  - **the API of the login endpoint(s)**
  - **the kind of credential the endpoint(s) will return**
- should convey:
  - scope so client software knows when it can re-use an existing credential …
  - … and when it needs a new one

Requirements

- describe VOSI-capabilities style?

```
<capability standardID="ivo://ivoa.net/stc/SSO#cookie">
  <accessURL>https://example.net/login</accessURL>
  <securityMethod standardID="ivo://ivoa.net/std/SSO#BasicAA"/>
</capability>
```

  - clients have to find and read /capabilities doc for definitive info

Requirements

- describe VOSI-capabilities style?

```
<capability standardID="ivo://ivoa.net/stc/SSO#cookie">
  <accessURL>https://example.net/login</accessURL>
  <securityMethod standardID="ivo://ivoa.net/std/SSO#BasicAA"/>
</capability>
```

  - clients have to find and read /capabilities doc for definitive info
- challenge via HTTP headers? RFC7235

```
WWW-Authenticate: {challenge} [, {challenge} . . .]

e.g. WWW-Authenticate: bearer realm="foo"
```

  - in the OAuth example, the “bearer” challenge conveys the type of credential needed by the service and the realm conveys scope information (but not **where** to get a bearer token)
  - WWW-Authenticate challenge is quite extensible: let’s explore...

Strawman for using HTTP headers

- define {challenge} for each credential type -- use SSO identifiers or industry standard strings where available, e.g.

```
WWW-Authenticate: "bearer" realm="foo",
     "ivo://ivoa.net/std/SSO#cookie", "ivo://ivoa.net/std/SSO#tls-with-certificate"
```

- use challenge params to convey the required info, e.g.

```
WWW-Authenticate: ivo://ivoa.net/std/SSO#cookie accessURL="https://example.net/login"
     securityMethod="ivo://ivoa.net/std/SSO#BasicAA" [, {challenge}, . . .]
```

- feasible to co-exist with and/or describe industry standards... not sure about adding params to existing challenges though
- prior art: https://tools.ietf.org/id/draft-broyer-http-cookie-auth-00.html

Complications and Extras

- services support anonymous and authenticated access on a single endpoint
  - how to provoke a 401 in order to proceed with authentication?
  - hack: submit a known invalid authentication attempt, e.g.

```
Authorization: tell-me-how-to-auth
```

- help clients know what is going on
  - services tell clients whether or not they successfully authenticated:

```
X-VO-Authenticated: true|false
```

- standardise/describe a very common implementation:
  - ivo://ivoa.net/std/SSO#FormLogin

```
POST username=foo&password=bar https://example.net/login
```