NRC·CNRC

VOSpace v2.1 and Discussion

Brian Major
Canadian Astronomy Data Centre

Contents

- 2.1 changes overview
- Transfers:
  - Review of transfer methods
  - Optimized with REQUEST=redirect (WD change, example)
  - Usefulness of async pullFromVoSpace, pushFromVoSpace
  - Access control in transfers (WD change, example)
  - Server-to-server transfers—client orchestrated?
- Distributable / Sharable VOSpace URL?
- Interoperable authorization
  - Node authorization example
- Node searching

Overview of v2.1 changes

Changes in this WD:

- Added REQUEST=redirect to optimized pullFromVoSpace and documented supported modes
- Changed Transfer in XSD to accept parameters (such as Content-length).
- Redid all webservice operation examples
- Changed XSD element authType to SecurityMethod to match IVOA standard
- Fixed general inconsistencies and added more text regarding transfers, access control
- Ported source to ivoatex, editorial changes

Main changes in the last 2.1 WD:

- Addition of optimized HTTP GET method of data transfer for pushToVoSpace, pullFromVoSpace
- Addition of authType to Protocol in XML schema for transfer negotiation.
- Deprecated view=data convenience method of data transfer

The most common VOSpace transfer: downloads

- pullFromVoSpace: the only mandatory transfer operation
- There are currently 4 ways to pullFromVoSpace:
  1. Full asynchronous transfer negotiation with XML POST
  2. Full synchronous transfer negotiation with XML POST
  3. Optimized synchronous negotiation using URL parameters:

```
?TARGET=<VOS URI>&DIRECTION=pullFromVoSpace&
  PROTOCOL=ivo://ivoa.net/vospace/core#httpget
```

  4. Optimized synchronous negotiation with direct download using:

```
&REQUEST=REDIRECT
```

URL Parameter Download

```
curl "http://www.canfar.phys.uvic.ca/vospace/synctrans?
  TARGET=vos://cadc.nrc.ca!vospace/majorb/file&
  DIRECTION=pullFromVoSpace&
  PROTOCOL=ivo://ivoa.net/vospace/core%23httpget”
```

```
<?xml version="1.0" encoding="UTF-8"?>
  <vos:transfer xmlns:vos=http://www.ivoa.net/xml/VOSpace/v2.1>
  <vos:target>vos://cadc.nrc.ca!vospace/majorb/file</vos:target>
  <vos:direction>pullFromVoSpace</vos:direction>
  <vos:protocol uri="ivo://ivoa.net/vospace/core#httpget”>
    <vos:endpoint>http://storage1.example.com/file
    </vos:endpoint>
  </vos:protocol>
  <vos:protocol uri="ivo://ivoa.net/vospace/core#httpget”>
    <vos:endpoint>http://storage2.example.com/file
    </vos:endpoint>
  </vos:protocol>
</vos:transfer>
```

URL Parameter Download with REQUEST=redirect

```
curl "http://www.canfar.phys.uvic.ca/vospace/synctrans?
  TARGET=vos://cadc.nrc.ca~vospace/majorb/file&
  DIRECTION=pullFromVoSpace&
  PROTOCOL=ivo://ivoa.net/vospace/core%23httpget&
  REQUEST=REDIRECT”
```

Results in a direct download:

```
HTTP/1.1 303 See Other
Location: http://storage1.example.com/trans/file
```

The preferred endpoint is returned by the service.

REQUEST=redirect is not supported for pushToVoSpace

Do we need asynchronous upload and download?

Asynchronous pullFromVoSpace, pushToVoSpace:

What advantage do they bring?

Since the client needs to transfer the bytes, they are aware when the transfer is complete or encounters an error. There is no need to check the job phase.

SecurityMethod in transfer negotiation

- 2.1 clients can request what authentication methods they wish to use for byte transfer
  - TLS client certificates, HTTP basic authentication, etc…
- SecurityMethod is necessary because you cannot rely on the incoming protocol or the protocol in the transfer document to determine how authentication is to be done.

SecurityMethod – backwards compatibility

- If a 2.1 service is talking to a 2.0 client, the securityMethod should NOT be included in the protocol.
- If a 2.1 service is talking to a 2.1 client, the securityMethod should be included in the protocol.
- No security methods means public (anonymous) access.

SecurityMethod example - Request

```
<?xml version="1.0" encoding="UTF-8"?>
  <vos:transfer xmlns:vos=http://www.ivoa.net/xml/VOSpace/v2.1>
  <vos:target>vos://cadc.nrc.ca!vospace/majorb/file</vos:target>
  <vos:direction>pullFromVoSpace</vos:direction>
  <vos:protocol uri="ivo://ivoa.net/vospace/core#httpsget”>
    <vos:endpoint>http://storage1.example.com/trans/file
    </vos:endpoint>
    <vos:securityMethod
      uri=“ivo://ivoa.net/sso#tls-with-certificate” />
  </vos:protocol>
</vos:transfer>

curl –H “Content-Type: text/xml” –d @transfer.xml
  "http://www.canfar.phys.uvic.ca/vospace/synctrans”
```

SecurityMethod example - Response

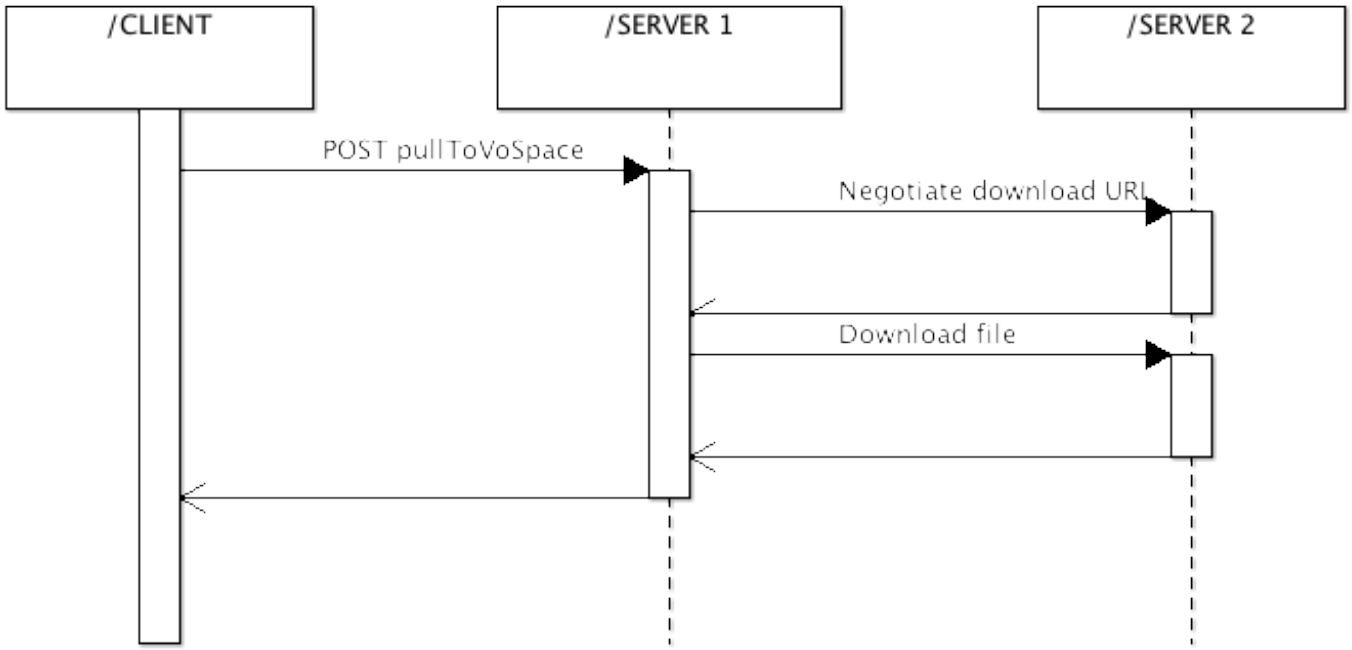
```
<?xml version="1.0" encoding="UTF-8"?>
  <vos:transfer xmlns:vos=http://www.ivoa.net/xml/VOSpace/v2.1>
  <vos:target>vos://cadc.nrc.ca!vospace/majorb/file</vos:target>
  <vos:direction>pullFromVoSpace</vos:direction>
  <vos:protocol uri="ivo://ivoa.net/vospace/core#httpsget”>
    <vos:endpoint>https://storage1.example.com/trans/file
    </vos:endpoint>
    <vos:securityMethod
      uri=“ivo://ivoa.net/sso#tls-with-certificate” />
  </vos:protocol>
</vos:transfer>
```

```
curl –E mycert.pem "https://storage1.example.com/trans/file”
```

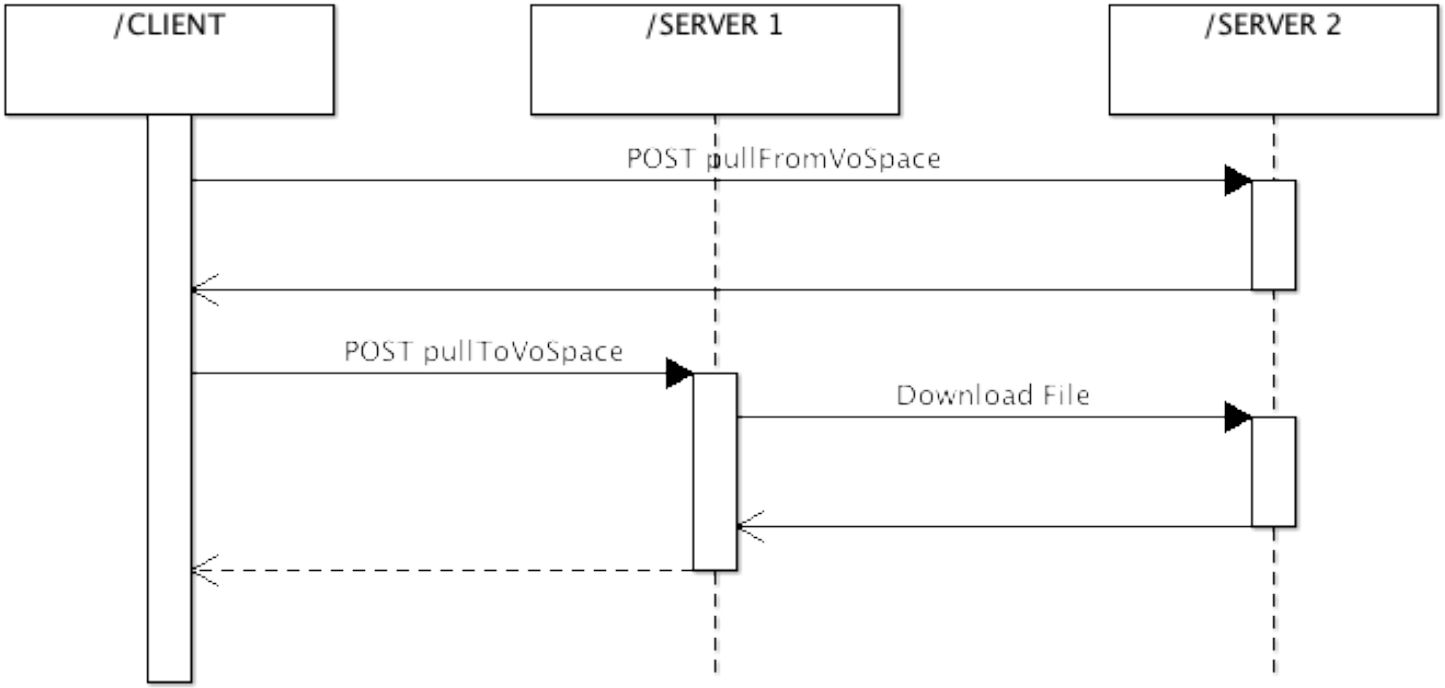
Server to server transfers

- Two of the transfer operations (pullToVoSpace, pushFromVoSpace) are for transfer between two different VOSpace instances
- Currently the negotiations to the peer VOSpace server are handled by the VOSpace service that receives the initial transfer request.
- I propose we move that responsibility to the client so VOSpace servers do not need to negotiate on behalf of the client.

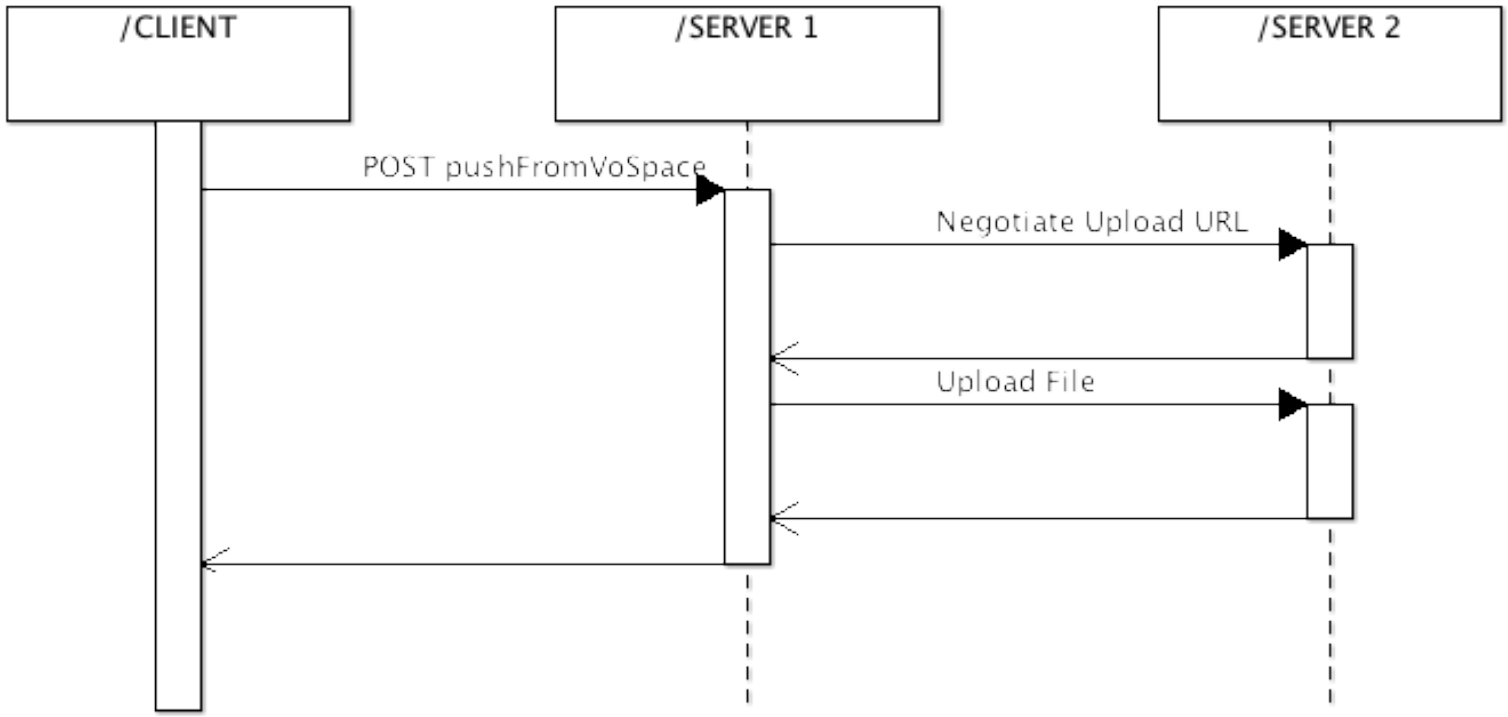
pullToVoSpace: current



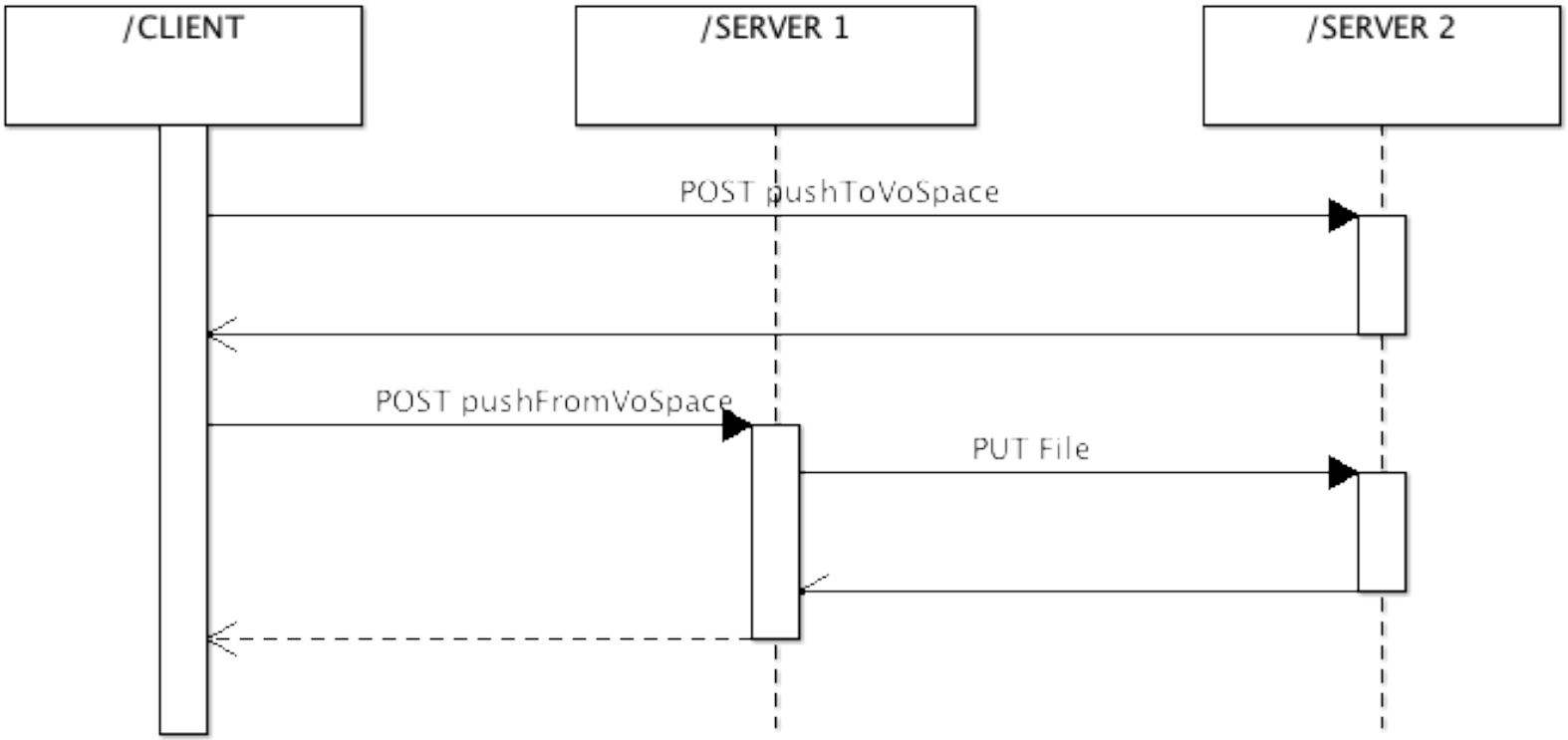
pullToVoSpace: proposed



pushFromVoSpace: current



pushFromVoSpace: proposed



Sharable, Distributable VOSpace Download URL

- Users are asking for a single sharable / distributable URL
- /synctrans REQUEST=redirect does this, but it is hard to read and write and is not RESTful
- ?view=data : awkward and not RESTful

- Is this a reasonable request? VOS URI resolution is a key part of VOSpace

- What about a /vospace/data endpoint for downloads?

Interoperable Authentication & Authorization

- Runs orthogonal to services (such as VOSpace)
- Interoperability use cases are here!

Interoperable Authentication

- See latest SSO Working draft for a list of support authentication protocols

- After authentication, the service needs to hold onto to the user's identity for use in authorization calls.
  (in java, this is the 'Subject')

Interoperable Authorization

- Each service making use of authorization must store authorization information that relates to that resource.

- With the user's identity in hand, services can then call:
  - The Credential Delegation Protocol (CDP) to obtain the delegated identity.
  - Some authorization service, such as a Group Management Service (GMS) to determine user membership

CANFAR VOSpace Authorization

- CANFAR VOSpace authorization is Node based, stored in the Node properties:
  - ivo://ivoa.net/vospace/core#ispublic – read access for all if true
  - ivo://ivoa.net/vospace/core#owner – full access for this user
    - e.g. ivo://canfar.net/user#C=ca,O=hia,OU=cadc,CN=majorb
  - ivo://ivoa.net/vospace/core#groupread – groups with read access
    - e.g. ivo://canfar.net/gms#myreadgroup
  - ivo://ivoa.net/vospace/core#groupwrite – groups with write access
    - e.g. ivo://canfar.net/gms#mywritegroup
- Inter-service calls all use X509 Client Certificates
- Similar to Linux, users must have read access to the root Node

Checking group membership example:

1. Get the groupread property:
   `ivo://canfar.net/gms#mygroup`
2. Resolve the access URL from a registry:
   `https://www.canfar.net/gms/groups`
3. Using CDP, get the calling user's proxy certificate
   `http://www.canfar.net/cdp/proxyCert > proxy.pem`
4. Call the GMS service as calling user to check membership:

```
curl –E proxy.pem https://www.canfar.net/gms/
   groups/mygroup
```

Interoperable authorization is achieved by fully referencing the authorization authority in the groupread URI (ivo://canfar.net/gms)

Node searching

```
<uws:jobInfo>
  <vos:search>
    <vos:detail>properties</vos:detail>
    <vos:matches>
      ivo://ivoa.net/vospace/core#description='galax'
    </vos:matches>
    <vos:node uri="vos://example.com!vospace/my/node" />
  <vos:search>
<uws:jobInfo>
```

Currently can't search/filter by node name/path or type… ideas?