OpenCADC – cadcVOS

- provides domain classes for some Node types
- provides XML read/write Node and Transfer
- provides restlet application for /nodes resource
- provides customisable NodeDAO class
- provides VOSpaceClient
  - library and command-line app

http://opencadc.googlecode.com/

OpenCADC – cadcVOS

- uses SSO (X509 certficates) for authentication
  - http: anonymous
  - https: requires certificate
- does not provide a /transfers resource, but we use cadcUWS library to implement one
  - service implementors need to write their own to work with their storage system
  - our impl delegates to a separate web service...
  - will likely provide a simple filesystem-based impl

beyond the spec: authorization

- authorization
  - owner can always read/write
  - group-read property = <URI to a group> (RO)
  - group-write property = <URI to a group> (RW)
  - public-read property = <boolean>

read permission	access node metadata
	negotiate transfer from VOSpace
write permission	on ContainerNode to create child nodes
	on DataNode to negotiate transfer to VOSpace

beyond the spec: groups

- how to allow access to multiple groups?
  - multiple group-read properties (e.g. same uri)?
  - one group-read property with a list of groups, e.g. multiple values?
  - owner can just make a new custom group (union), e.g. punt to GMS?

beyond the spec: quotas

- admin tool creates a ContainerNode, sets owner and quota
  - quota stored as node property of this container

    `ivo://cadc.nrc.ca/vospace/properties#quota`
  - DataNode size stored after transfer using property

    `ivo://ivoa.net/vospace/core#length`
- create transfer (to vospace) fails when quota exceeded
- actual transfer fails when quota exceeded, even mid-stream

beyond the spec: quotas

- ContainerNode space used property

  `ivo://ivoa.net/vospace/core#length`?

  - ContainerNode aggregate size stored after transfer or delete by walking up the tree and updating

- ContainerNode property

  `ivo://ivoa.net/vospace/core#availableSpace`

  - is it the same as quota?
  - is it the amount of space remaining?
  - should it only be reported for the root container?

thorny issues: vos URI

- vos URI is easy to parse

  `vos://cadc.nrc.ca!vospace/pdowler/foo.png`

  `vos` (scheme)

  `cadc.nrc.ca!vospace` (authority)

  `/pdowler/foo.png` (path)

- using it is fairly easy

  `ivo://cadc.nrc.ca/vospace` (replace ! with / in authority, use ivo scheme)

  `http://www.cadc.hia.nrc.gc.ca/vospace + {node resource} + path` (lookup service base URL in registry)

thorny issues: vos URI

- BUT: that ! is a pain on the command-line

  `vos://cadc.nrc.ca!vospace/pdowler/foo.png`

- could we use a different character?
  - has to be unreserved: _ - ! . ~ ' ( ) *

    `vos://cadc.nrc.ca~vospace/pdowler/foo.png`

- could we use alternate parts of URI?
  - <scheme>:<authority><path>#<fragment>

    `vos://cadc.nrc.ca/vospace#pdowler/foo.png`

    `ivo://<authority><path>` (service URI)

    `ivo:<scheme-specific-part>` (service URI)

thorny issues: DistinguishedName

- owner identity is stored when nodes are created

  `ivo://ivoa.net/vospace/core#creator`

  - comparing DNs for equality is tricky
- a user DN may change when they renew certificate
  - probably a bad idea to store the DN directly with node metadata
  - store internal ID with node, map of ID<->DN
  - allow users to update their stored DN if it changes

thorny issues: storage

- our VOSpace uses new iRODS-based archive system (ad2): <archive,namespace,fileID>
  - metadata in a RDBMS (node and property tables)
  - storage: <vospace,$path,$name>
  - path is the sequence of container nodes

    ```
    internal vospace move requires changing
    namespace(s) of arbitrary number of files
    in ad2 – ouch!
    ```
- good idea to decouple VOSpace metadata from storage
  - storage: <vospace,null,$nodeID>

cadcTestVOS

- provides unit tests for all the REST interactions
  - GetContainerNodeTest, GetDataNodeTest
  - CreateContainerNodeTest, CreateDataNodeTest
  - UpdateContainerNodeTest, UpdateDataNodeTest
  - DeleteContainerNodeTest, DeleteDataNodeTest
  - PushToVOSpaceTest, PullFromVOSpaceTest
- we use as integration tests
- could be made into a decent conformance test

cadcTestVOS

- missing some tests
  - PushFromVOSpaceTest, PullToVOSpaceTest
  - PropertiesTest, ProtocolsTest, ViewsTest
- our implementation does not pass :-)
  - error handling is dodgy: wrong codes and text
  - VOSpaceClient is POSTing and cadcUWS is accepting vanilla parameters instead of XML for transfer jobs
  - our VOSpace transfer runner expects parameters in the UWS job parameters list...

VOSpace transfers

- client POSTs something to the /transfers resource to create an async job (UWS)
- parameters not sufficient because transfer jobs can have multiple protocol-endpoint pairs
- current WD says to post a <uws:job> document
  - UWS specifies how fields of the job are set and this looks like it adds a new way to do that
- proposal: post a <vos:transfer> document
  - already defined and used as a document
  - stick it into the <uws:jobInfo>, WD examples OK

CADC VOSpace Summary

- incomplete but usable
  - service and a cheating command-line client
  - in use by real users for storing and sharing data, VMs
- authentication using SSO (X509 certificates)
- authorization as owner or group member (GMS)
- mostly open-source but code a bit rough at this point

http://opencadc.googlecode.com/