VOSpace 1.0 Progress

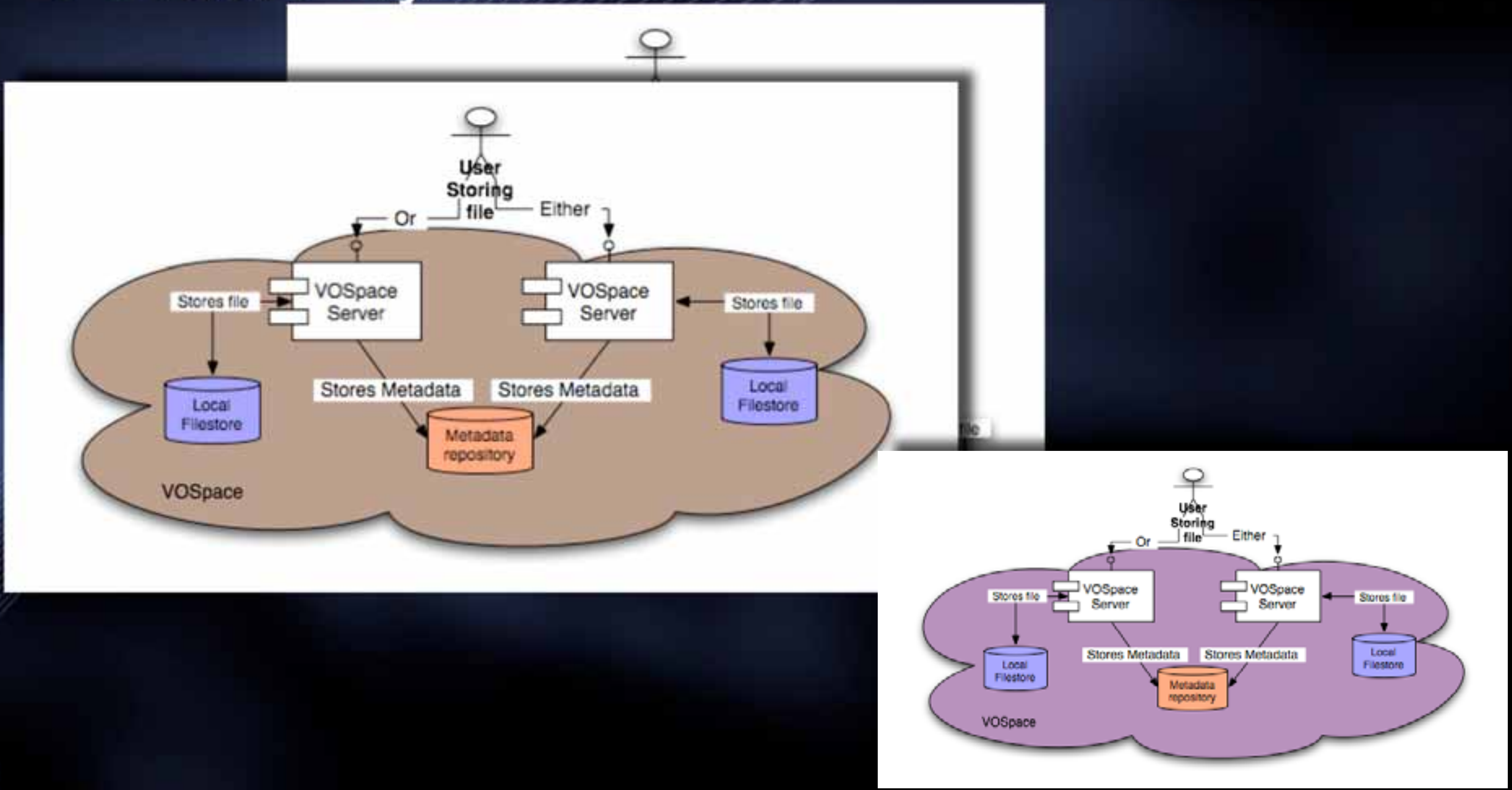
Paul Harrison ESO

Introduction

- Active Participants
  - Matthew Graham - Caltech
  - Paul Harrison - ESO
  - Dave Morris - AstroGrid
  - Guy Rixon - AstroGrid
- Much Email/IM interaction as well as physical meetings in Victoria, Cambridge and Leicester

Recap from last meeting

- VOSpace architecture changed - dropped VOStore layer.



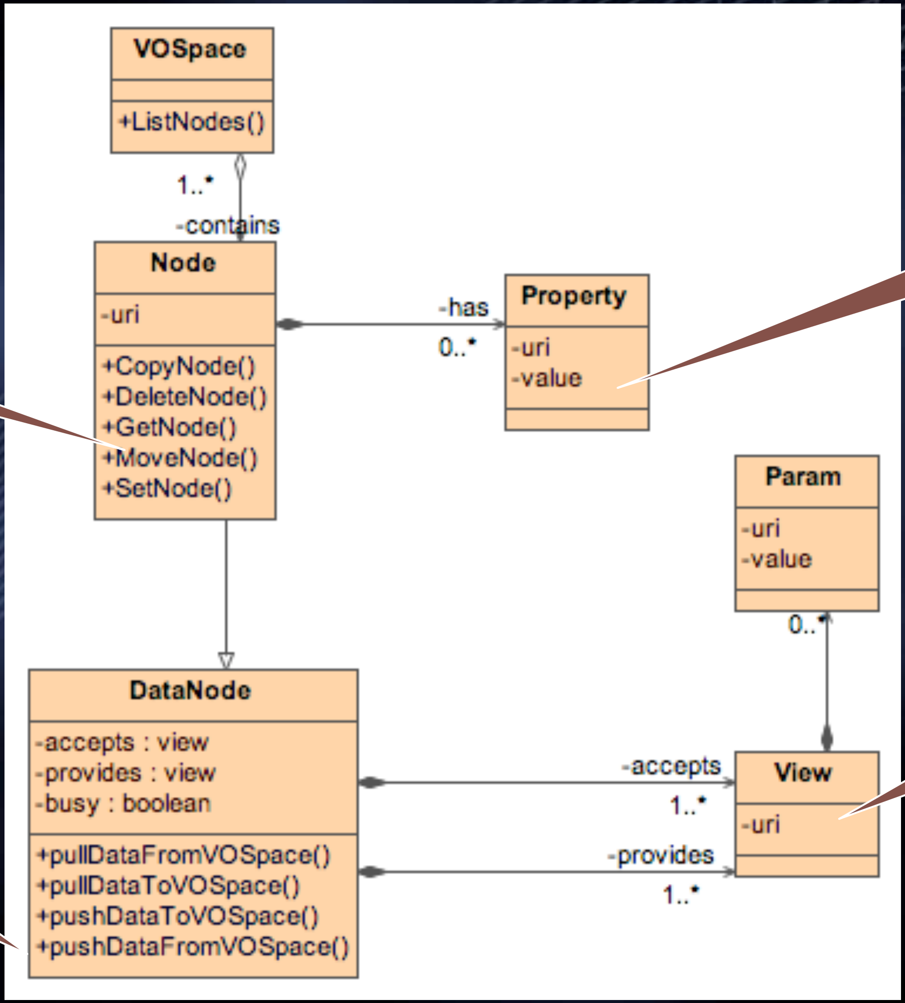
Basic Aims

- Strategy
  - Produce 1.0 that can be implemented now - suitable for interface to existing archive storage systems so that current standard services can return a vos: URL.
  - 2.0 (1.1+?) would be the version that could be used to provide an “end-user friendly” personal store.
- Design
  - Intended as a web service (SOAP) façade on existing systems that offer similar core functionality e.g. SRB, NGAS
  - Open, extensible, and permissive interface

What has been produced

- Draft 1.0 Standard
  - http://www.ivoa.net/twiki/bin/view/IVOA/VOSpace10Spec
  - Please look and comment
  - Intended to formally publish as 1.0 WD at Moscow IVOA interoperability meeting.
- WSDL for interface & Registry schema extension
  - http://www.ivoa.net/twiki/bin/view/IVOA/VOSpace10schema
  - You can build your own! - or better build clients…

VOSpace 1.0 Domain Model



Basic manipulations

Flexible metadata

Data transport

Flexible transformations

VOSpace Locators

- Want URI syntax
- Follow recommendations of RFC3986 and RFC2718
- In fact we want the subset of URIs that form hierarchical URLs
  - //……/ is the signature
  - Query part resolved by service
  - Fragment part resolved by client

$$\underbrace{\texttt{foo}}_{\text{scheme}}\texttt{://}\underbrace{\texttt{example.com:8042}}_{\text{authority}}\underbrace{\texttt{/over/there}}_{\text{path}}\texttt{?}\underbrace{\texttt{name=ferret}}_{\text{query}}\texttt{\#}\underbrace{\texttt{nose}}_{\text{fragment}}$$

Standard URL behaviour

- We proposed VOSpace locators as a new URI scheme vos:
  - We want to have standard relative URL syntax to work, as VOSpace will have links.
  - We want to be able to use the “Query” part to be able to express simple search functionality such as “all the VOTables in a container”
  - We want to be able to use the “Fragment” part to be able to specify functionality such as “the 2nd image in the FITS file”

VOSpace URL

- Suppose IVOA ID for the VOSpace is
  ivo://org.vo/vospace
- Data object in the VOSpace
  /path/to/myData
- Full URL of the data object
  vos://org.vo!vospace/path/to/myData
- Registry plays the role that DNS does in http URLs.

Views

- Originally there to support idea of VOSpace front end to data stored in table of RDB.
- Perhaps “view” not the best name
  - On import think of view as specifying data format of imported data
  - On export, the view is effectively the data format that the internally stored data should be transformed into.

View Example

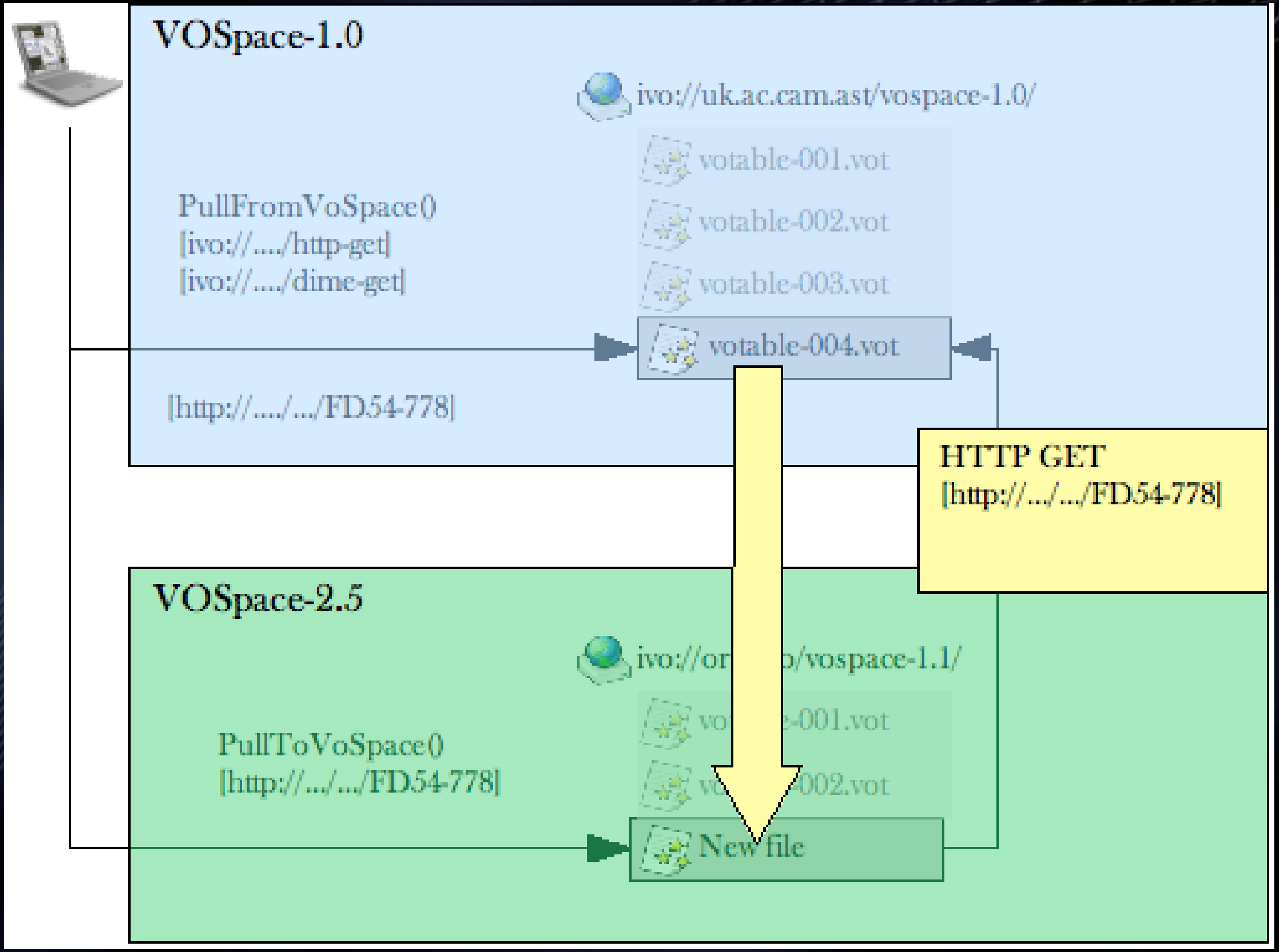
```
<node
  uri=”vos://uk.ac.cam.ast!vospace-1.0/image”>
          ....
 <views>
   <accepts>
     ....
   </accepts>
   <provides>
     <view uri=”....image-fits” original=”true”/>
     <view uri=”....thumb-jpeg”/>
   </provides>
 </views>
</node>
```

Data Transfer

- Asynchronous - client controls transfer of data, after the server has responded to SOAP request with a transfer endpoint URL.
  - PullFromVOSpace
  - PushToVOSpace
- Synchronous - server controls the transfer of data from the URL that the client supplies in the SOAP request which does not terminate until the transfer is complete.
  - PullToVOSpace
  - PushFromVOSpace

VOSpace 2 has asynchronicity as a priority

Data Transfer between Spaces



Protocols

- The protocol is the mechanism by which the bulk data are transported - the intention is that a VOSpace server could support several transport protocols.
- None are mandated, but there is a recommended list which is described in the registry.

Protocols - local file example

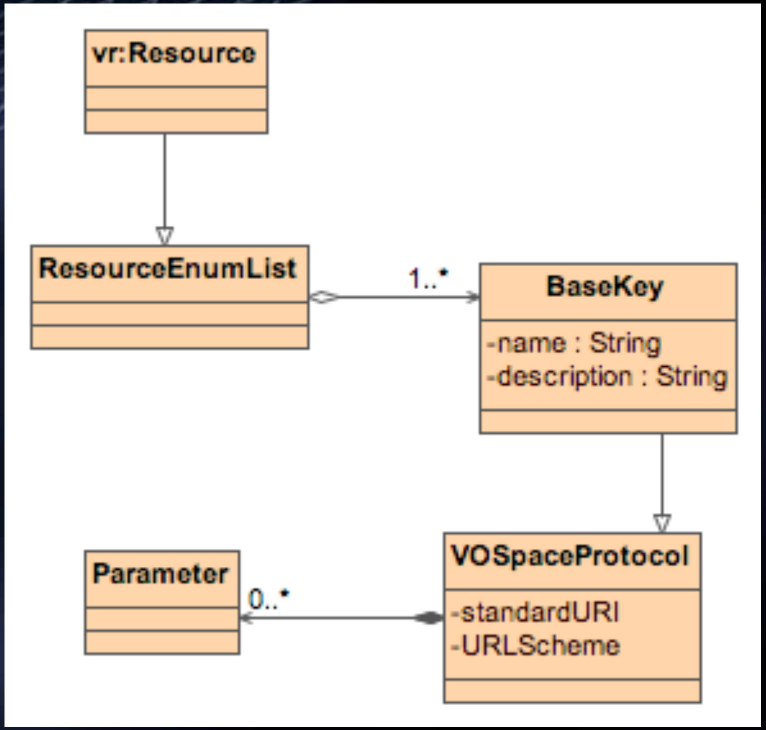
- Could add specific protocol to do local file transfer
  - e.g protocol called "…cambridge-nfs", which means available as a file on Cambridge LAN.
  - Client that knows it want to run CPU intensive job on Terabyte sized data at Cambridge can look in registry for VOSpace node that offers this protocol
- Makes workflow non-transportable, but does show flexibility of design - might be better if concept of network locality was coded better into VOSpace/Registry.

Detail - schema enumerations

- Rather than encode enumerations in the interface schema, these are encoded in the registry -
  - E.g. property names, protocol names, views etc.
- Pros
  - Easier to extend - do not have to issue new version of interface schema when a new enumeration value is required - simply edit the registry entry.
  - Easier for individual implementation to publish details of ‘non-standard’ enumeration values in a way that can be semi-automatically understood - e.g. by GUI tools to display a message to user.
- Cons
  - Allowed values are not enforced directly by the interface - up to the programmer to read registry.

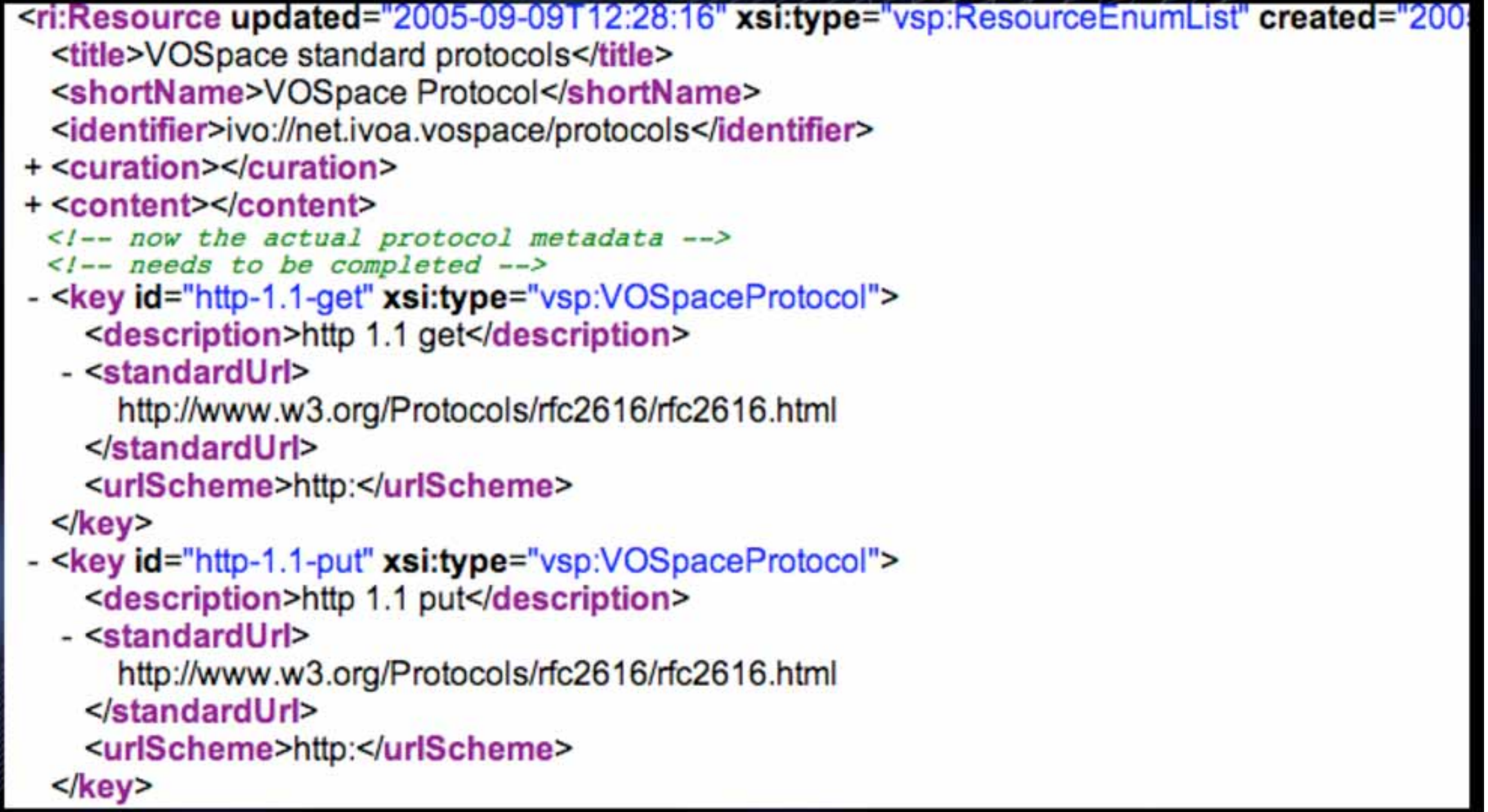
Detail - schema enumerations(2)

- Aim is to produce URI
- Multiple keys per registry entry
  - Only one copy of the Dublin core
  - Standard prefix
  - Use fragment # separator to indicate the enumeration key



ivo://net.ivoa.vospace/protocols#http-get

Detail - schema enumerations(3)



```
<ri:Resource updated="2005-09-09T12:28:16" xsi:type="vsp:ResourceEnumList" created="200
  <title>VOSpace standard protocols</title>
  <shortName>VOSpace Protocol</shortName>
  <identifier>ivo://net.ivoa.vospace/protocols</identifier>
+ <curation></curation>
+ <content></content>
  <!-- now the actual protocol metadata -->
  <!-- needs to be completed -->
- <key id="http-1.1-get" xsi:type="vsp:VOSpaceProtocol">
    <description>http 1.1 get</description>
  - <standardUrl>
      http://www.w3.org/Protocols/rfc2616/rfc2616.html
    </standardUrl>
    <urlScheme>http:</urlScheme>
  </key>
- <key id="http-1.1-put" xsi:type="vsp:VOSpaceProtocol">
    <description>http 1.1 put</description>
  - <standardUrl>
      http://www.w3.org/Protocols/rfc2616/rfc2616.html
    </standardUrl>
    <urlScheme>http:</urlScheme>
  </key>
```

Implementation Status

- Trying to produce 3 independent reference implementations for gaining IVOA Proposed Recommendation Status.
  - Caltech - with pluggable db backend for metadata
  - ESO - using NGAS for the bulk storage
  - AstroGrid - replacement for MySpace
- Is a well behaved GWS WG service
  - Standard interface
  - SSO compliant