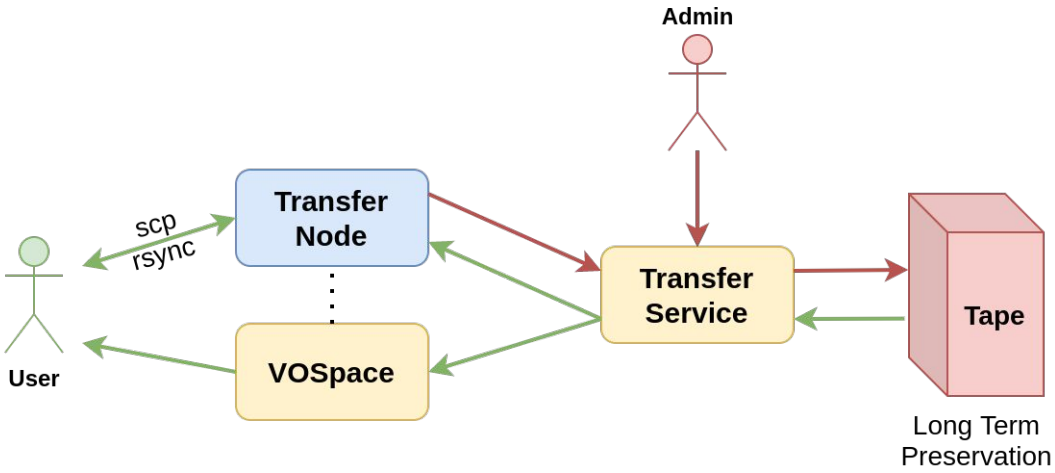
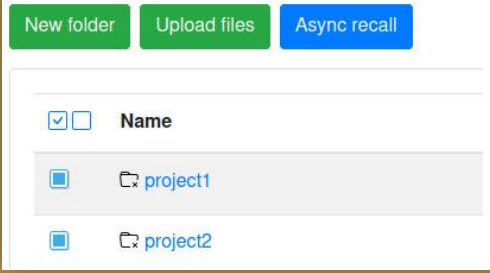
A VOSpace implementation with tape support

Sonia Zorba, Cristiano Urban, Sara Bertocco, Nicola Fulvio Calabria

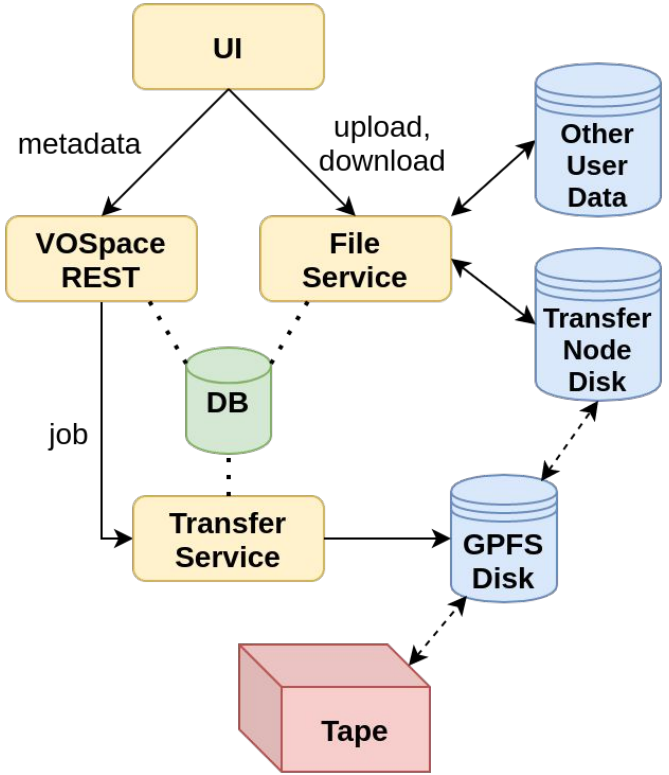
Virtual Interoperability Meeting - May 2021

IA2 tape use case





General overview



- pushToVoSpace and pullFromVoSpace endpoints point to File Service
- Transfer Service performs import and export of files from the storage reachable from File Service to the tape and vice versa
- file metadata is stored in a shared database
- A&A based on OAuth2

Tape async recall

Custom property in node to indicate that data is not immediately available for download:

```
<vos:properties>
    <!-- ... -->
    <vos:property uri="urn:ia2:async-trans">true</vos:property>
</vos:properties>
```

Custom protocol in pullToVoSpace transfer:

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0" version="2.1">
   <vos:target>vos://example.com!vospace/mydir</vos:target>
   <vos:direction>pullToVoSpace</vos:direction>
   <vos:protocol uri="urn:ia2:async-recall"></vos:protocol>
</vos:transfer>
```

List of files transfer

Use case: user wants to perform an asynchronous recall of multiple nodes (e.g. a subset of files inside a directory).

Problems:

- VOSpace transfer operations expect **one target node**.
- We can't start multiple jobs (one for each node) because tape library software automatically optimizes the retrieval of a set of files from multiple cartridges.

Current workaround: a temporary StructuredDataNode with a custom view (urn:ia2:list-of-files) is created. It contains the list of nodes to transfer. The pullToVoSpace operation specifies this node as the target.

Communication with token-based GMS

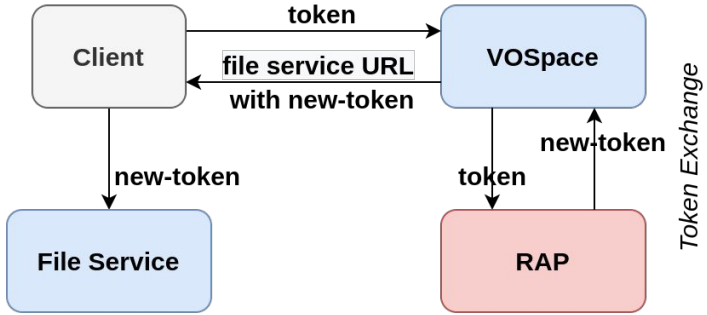
- The same token used for VOSpace is forwarded to GMS (token relay)
- All the groups are retrieved and result is cached for some minutes
- getNode() endpoint returns only the nodes that can be accessed by the user (according to groupRead property), so filtering each node using isMemberOf could be heavyweight

OAuth2 token exchange in transfers

```
<vos:transfer ...>
    <vos:target>vos://example.com!vospace/mynode</vos:target>
    <vos:direction>pushToVoSpace</vos:direction>
    <vos:protocol uri="ivo://ivoa.net/vospace/core#httpput">
        <vos:endpoint>http://file-service/mynode?token=eyJ0eXA...</vos:endpoint>
    </vos:protocol>
</vos:transfer>
```

RFC 6819: «*If access tokens are sent via URI query parameters, such tokens may leak to log files and the HTTP "referer".*»

→ a new token with a narrower scope, jti claim and shorter expiration is generated using Token Exchange (RFC 8693)



“Single-user groups” sharing

Use case: user wants to share a node with another user.

We are using special groups in the GMS that are associated with a single user.

```
<vos:property uri="ivo://ivoa.net/vospace/core#groupread">mygroup1 people.user2</vos:property>
```

group of users

special “single-user group”

Other discussion points

Undeletable nodes
For some nodes we want that the user can edit their properties (e.g. groupRead) but can't delete them (long term preservation data can't be deleted). We are using a custom property for this (sticky, similar to CADC locked?).

Recursive groups update
We added a parameter to the setNode operation for performing the groupRead and groupWrite update recursively on all child nodes.

Folder size
Total size? Zero? 4 KB (file system block size)?

Pagination API
What if a node contains too many child nodes?

Used technologies

- File metadata is stored in a PostgreSQL database with ltree extension
- JAXB beans automatically generated from XSD files using XJC
- Spring Boot for the REST service and the File Service
- Jackson combined with JAXB to support both XML and JSON payloads
- Python for the Transfer Service with Redis queues for handling jobs
- Spring Boot + Vue.js for the UI

Current status

Development is in progress, ready for production in the next months.
Code available here: https://www.ict.inaf.it/gitlab/vospace (still partially private)

Tests with CADC vofs module shown a good level of compatibility.

Thanks for your attention

Questions?

Contact:
sonia.zorba@inaf.it