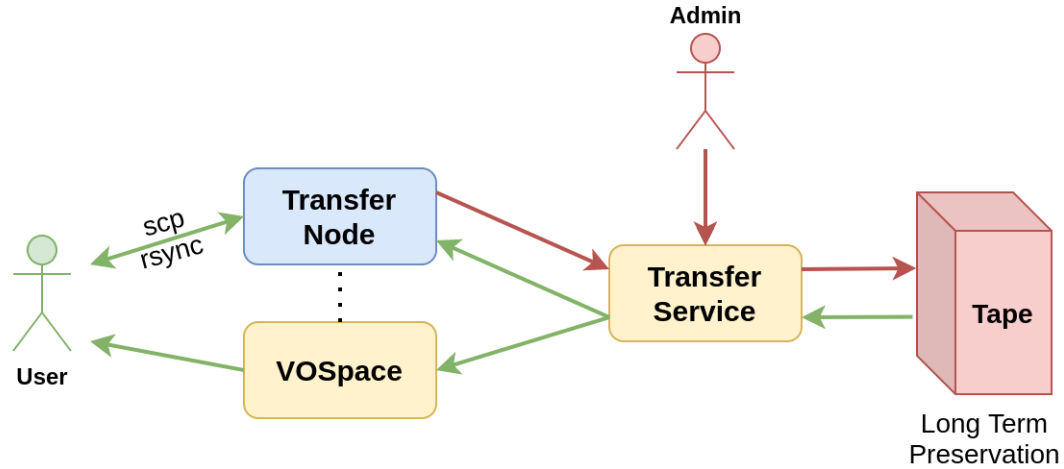


INAF VOSpace: status updates

Nicola Fulvio Calabria, Sonia Zorba, Cristiano Urban, Sara Bertocco, Cristina Knapic, Massimo Sponza




IA2 tape use case



New folder Upload files Async recall

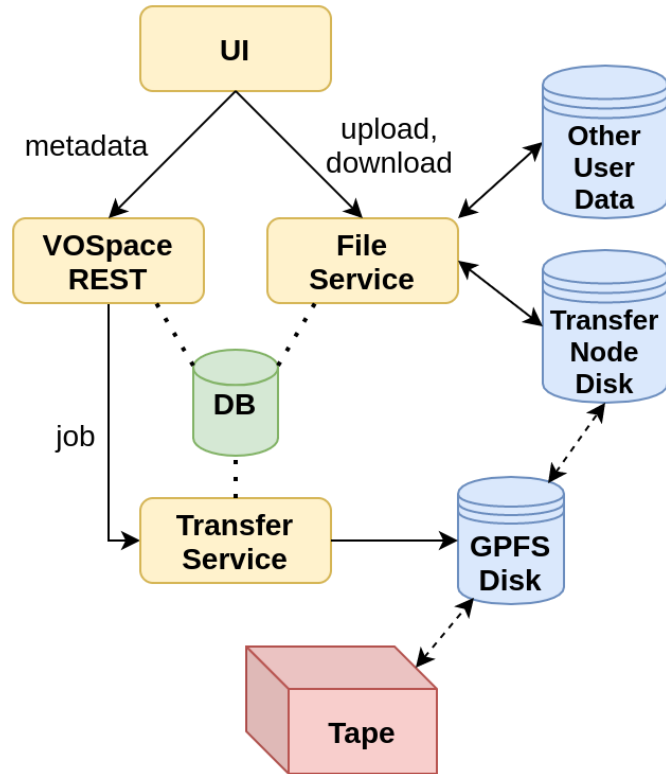
Name

 project1

 project2



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>
```

Recall multiple nodes from tape

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

Problems:

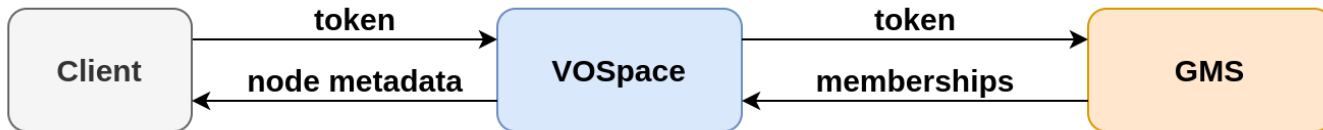
- VOSpace transfer operations expect **one target node**.
- We can't start multiple jobs (one for each node) because tape library software automatically optimizes the retrieval of a set of files from multiple cartridges.

Solution: a custom view (async-recall) is created. It contains the list of nodes to transfer using the “/include” parameter. The view populates a pullToVoSpace e target.

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0" version="2.1">
  <vos:target>vos://example.com!vospace/mynode</vos:target>
  <vos:direction>pullToVoSpace</vos:direction>
  <vos:view uri="ivo://ia2.inaf.it/vospace/views#async-recall">
    <vos:param uri="ivo://ia2.inaf.it/vospace/views#async-recall/include">test1.txt</vos:param>
    <vos:param uri="ivo://ia2.inaf.it/vospace/views#async-recall/include">test2.txt</vos:param>
    <vos:param uri="ivo://ia2.inaf.it/vospace/views#async-recall/include">test3.txt</vos:param>
  </vos:view>
</vos:transfer>
```

Versatile approach: we use other custom views to enable zip/tar archive creation by users

Communication with token-based GMS

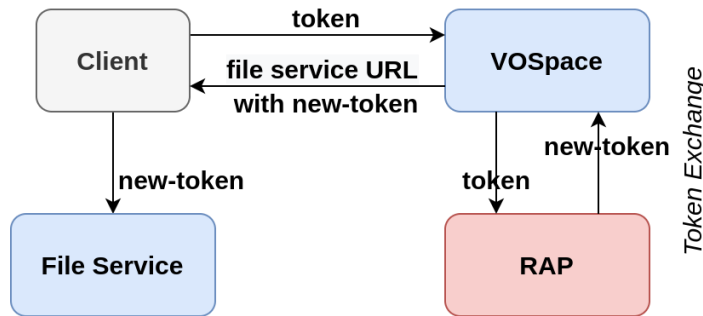


- The same token used for VOSpace is forwarded to GMS (token relay)
- All the groups are retrieved and result is cached for some minutes
- getNode() endpoint returns only the nodes that can be accessed by the user (according to groupRead property), so filtering each node using isMemberOf could be heavyweight

Implementation of OAuth2 token exchange in transfers:

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) for file transfer requests to File Service



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.

Plans

Undeletable nodes

Sticky nodes: users can edit their properties (e.g. groupRead) but can't delete them. Children nodes are not affected. Already implemented.

Immutable nodes: read-only nodes, their children inherit this property, proper abstraction of a Long Term Preservation node. Plan to implement.

Recursive groups update

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

Performances are good.

-> Switch to an asynchronous implementation for the REST setNode operation?

Thanks for your attention

Questions?

Contact:

nicola.calabria@inaf.it

sonia.zorba@inaf.it

BACKUP

“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”

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)

