



An HTTP REST API for Tape-backed Storage

João Afonso
joao.afonso@cern.ch

Tuesday, 9 May 2023



Introduction

WLCG Tape REST API project

- New protocol created by the *WLCG Tape REST API work group*.
- A common HTTP REST interface allowing clients to manage access to files stored on tape.
 - All tape file transfer & management operations can be done with HTTP.
- Storage-neutral API: Same protocol to be used by all storage systems.

Collaboration between different actors

- **WLCG storage providers:** EOS+CTA, dCache, StoRM.
- **T0 main clients:** FTS

Introduction

WLCG Tape REST API project

- New protocol created by the *WLCG Tape REST API work group*.
- A common HTTP REST interface allowing clients to manage access to files stored on tape.
 - All tape file transfer & management operations can be done with HTTP.
- Storage-neutral API: Same protocol to be used by all storage systems.

Collaboration between different actors

- **WLCG storage providers:** EOS+CTA, dCache, StoRM.
- **T0 main clients:** FTS

Focus of this presentation



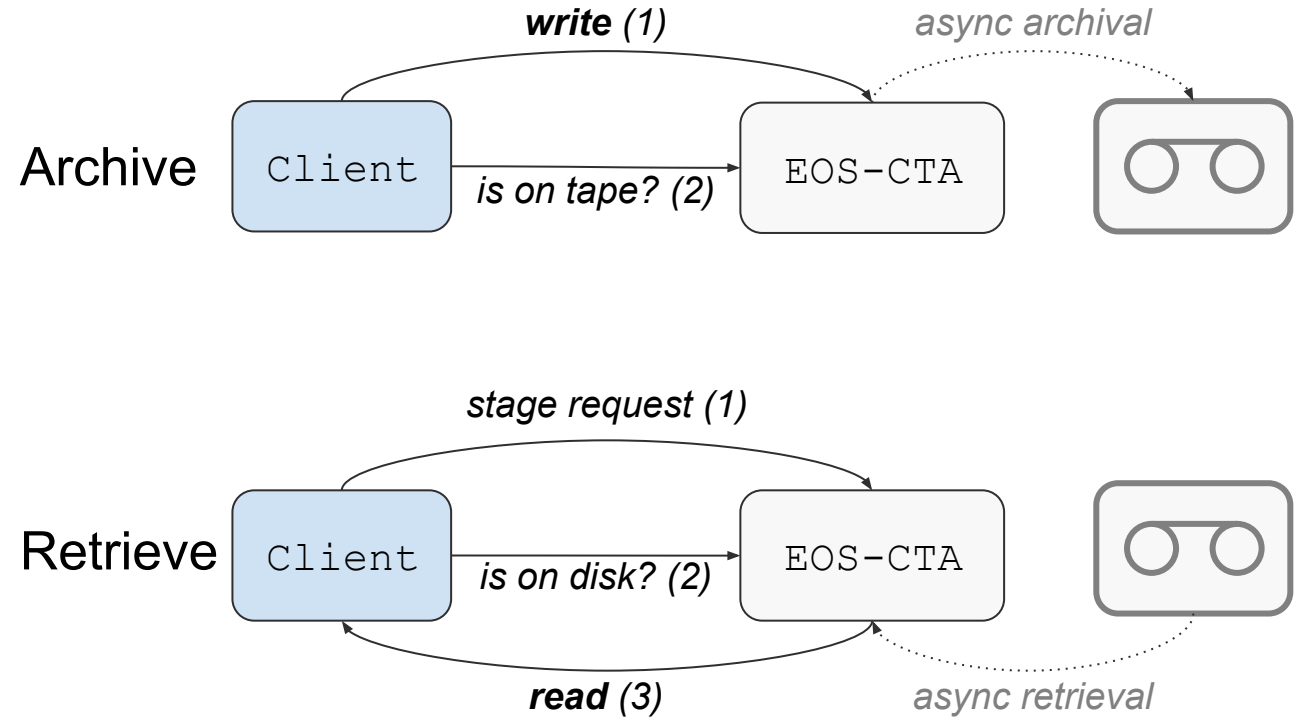
Introduction

Complexity of reading/writing files on tape:

- Multiple steps required
- Async operation

Previous HTTP limitations:

- *Check-On-Tape* could not be done in HTTP
- *Staging* operations could not be done in HTTP



Use cases

Use case	Functionality
Tracking file archival on tape	<ul style="list-style-type: none">● Check disk/tape residency of files being archived to tape
Staging a set of files from tape	<ul style="list-style-type: none">● Submit files for staging● Track staging files● Cancel files from staging
Releasing recalled files	<ul style="list-style-type: none">● Evict copies from the disk buffer

Specifications - short summary

Use case	XRootD	HTTP Tape REST API
Tracking file archival on tape	<code>xrdfs query prepare <fake_id> /path/file.txt</code>	<code>POST /api/v1/archiveinfo [with JSON containing files]</code>
Staging a set of files from tape	<code>xrdfs prepare -s /path/file.txt</code>	<code>POST /api/v1/stage [with JSON containing files]</code>
	<code>xrdfs query prepare <request_id> /path/file.txt</code>	<code>GET /api/v1/stage/<request_id></code>
	<code>xrdfs prepare -a <request_id> /path/file.txt</code>	<code>POST /api/v1/stage/<request_id>/cancel [with JSON containing files]</code>
Releasing recalled files	<code>xrdfs prepare -e /path/file.txt</code>	<code>POST /api/v1/release/<request_id> [with JSON containing files]</code>

Copy of HTTP Tape REST API specification document (v1):

- [Click to get document](#)

EOS-CTA: Protocol example

Tracking archival on tape

```
$ curl --head -L --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}
https://eosctapps.cern.ch:8444/eos/ctapps/tabtest/ --upload-file ./test_file
[...]
HTTP/1.1 201 CREATED
Connection: Keep-Alive
Content-Length: 0
Date: Fri, 28 Apr 2023 09:20:23 GMT
```

```
$ # Wait a few mins...
```

```
$ FILES='{ "paths" : [ "/eos/ctapps/tabtest/test_file" ] }'
$ curl -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}
https://eosctapps.cern.ch:8444/api/v1/archiveinfo -X POST -H "Content-Type: application/json" -d
"${FILES}" | jq '.'
[
  {
    "locality": "TAPE",
    "path": "/eos/ctapps/tabtest/test_file"
  }
]
```

Possible localities:

- DISK
- TAPE,
- DISK_AND_TAPE

EOS-CTA: Protocol example

Staging a file from tape, releasing file and deleting request (1)

```
$ FILE_REQ='{ "files": [ { "path": "/eos/ctapps/tabtest/test_file" } ] }'  
$ curl -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT} https://eosctapps.cern.ch:8444/api/v1/stage -X POST  
-H "Content-Type: application/json" -d "${FILE_REQ}" | jq '.'  
{  
  "requestId": "11938de2-e5c5-11ed-8e47-ac1f6b4cd618"  
}
```

```
$ # Wait a few mins...
```

```
$ curl -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}  
https://eosctapps.cern.ch:8444/api/v1/stage/11938de2-e5c5-11ed-8e47-ac1f6b4cd618 -X GET | jq '.'  
{  
  "createdAt": 1682687008,  
  "files": [  
    {  
      "onDisk": true,  
      "path": "/eos/ctapps/tabtest/test_file"  
    }  
  ],  
  "id": "11938de2-e5c5-11ed-8e47-ac1f6b4cd618",  
  "startedAt": 1682687008  
}
```


EOS-CTA: Protocol example

Staging a file from tape, releasing file and deleting request (2)

```
$ FILES='{ "paths" : [ "/eos/ctapps/tabtest/test_file" ] }'  
$ curl --head -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}  
https://eosctapps.cern.ch:8444/api/v1/release/11938de2-e5c5-11ed-8e47-ac1f6b4cd618 -X POST -H "Content-Type: application/json" -d "${FILES}"  
HTTP/1.1 200 OK  
Connection: Keep-Alive  
Content-Length: 0  
Date: Fri, 28 Apr 2023 15:08:34 GMT
```

```
$ curl -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}  
https://eosctapps.cern.ch:8444/api/v1/stage/11938de2-e5c5-11ed-8e47-ac1f6b4cd618 -X GET | jq '.'  
{  
  "createdAt": 1682687008,  
  "files": [  
    {  
      "onDisk": false,  
      "path": "/eos/ctapps/tabtest/test_file"  
    }  
  ],  
  "id": "11938de2-e5c5-11ed-8e47-ac1f6b4cd618",  
  "startedAt": 1682687008  
}
```

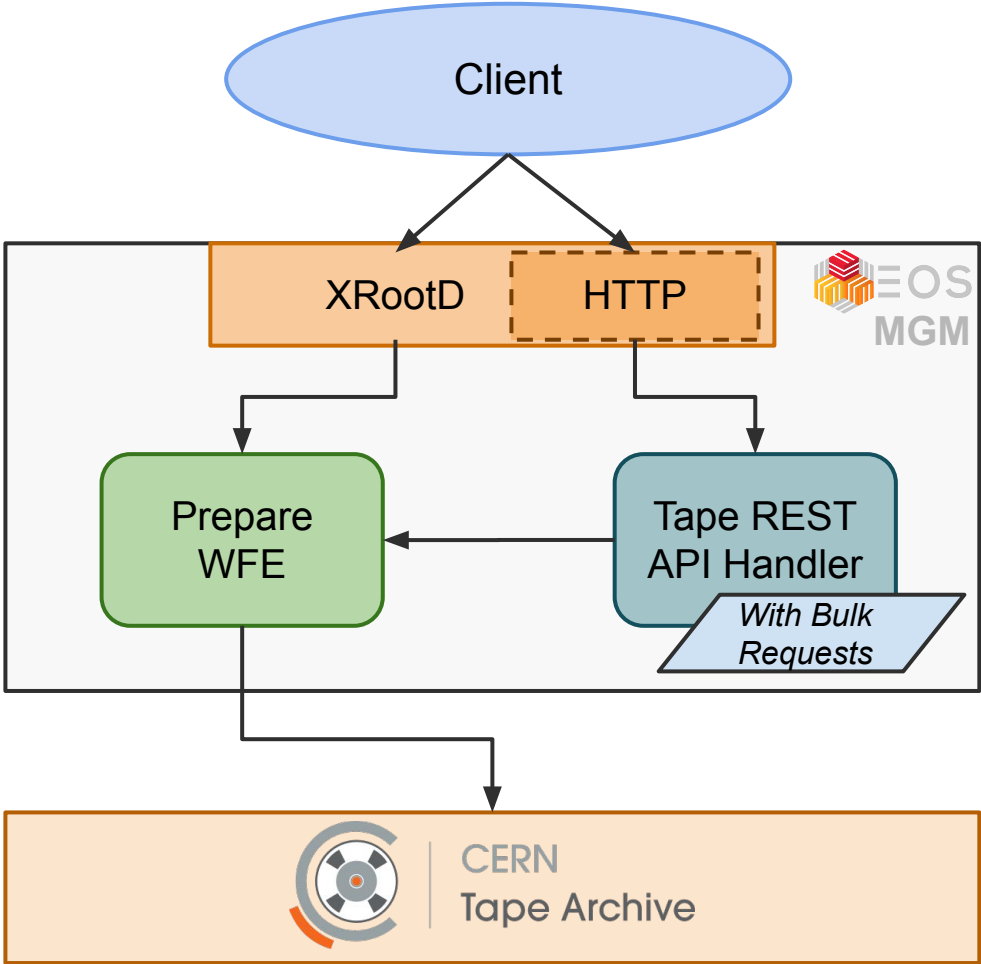
EOS-CTA: Protocol example

Staging a file from tape, releasing file and deleting request (3)

```
$ curl --head -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}
https://eosctapps.cern.ch:8444/api/v1/stage/11938de2-e5c5-11ed-8e47-ac1f6b4cd618 -X DELETE
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 0
Date: Fri, 28 Apr 2023 13:43:00 GMT
```

```
$ curl -sL --capath /etc/grid-security/certificates -E ${VOMS_CERT} --cacert ${VOMS_CERT}
https://eosctapps.cern.ch:8444/api/v1/stage/11938de2-e5c5-11ed-8e47-ac1f6b4cd618 -X GET | jq '.'
{
  "status": 404,
  "title": "Not found"
}
```

EOS-CTA: Implementation



EOS-CTA: Activation instructions

- First, configure [EOS HTTP transfers with XrdHttp](#).
- Then, [enable the HTTP Tape REST API](#) in the EOS-CTA mgm config file:

```
mgmofs.tapeenabled true  
taperestapi.sitename cern-cta-xxxx
```

Finally, the Tape REST API can be activated with these EOS commands:

- Tape REST API STATUS (enable *fileinfo* required for *Check-On-Tape*):

```
eos space config default taperestapi.status=on
```

- Tape REST API STAGE (staging, staging request tracking, abortion, eviction):

```
eos space config default taperestapi.stage=on
```

EOS-CTA: Deployment for LHC Run-3 operations

HTTP Tape REST API version 1.0 specifications implemented in EOS-CTA:

- Included in the software stack of CTA version 4/5.8.7-1 (EOS 4.8.98/5.1.9):
 - *Public release available*
- Deployed in production at CERN during March 2023

Enabled on T0 EOS-CTA instances:

- ***eosctalhcb***: Started being used on 23 March 2023 (stage requests and *Check-On-Tape* for T0↔T1 transfers)
- ***eosctaatl***: Started being used on 18 April 2023 (Rucio switched to HTTP-only)

EOS-CTA: Deployment for LHC Run-3 operations

Example: FTS archive monitoring (eosctaatl)



- Only HTTP operations after 18 April 2023

Conclusion

WLCG Tape REST API is fully implemented in EOS-CTA:

- Tape metadata only operation
- All tape file transfers can now be done purely with HTTP:
 - No XRootD required
- Specification, implementation and **deployment** at T0 complete:
 - **Ready to be used!**
- [Specification document link](#)

Thank you



home.cern