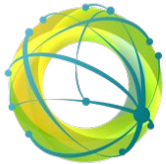
FTS & XRootD Workshop

CNRS Rhône Auvergne Delegation, Lyon – 14-18th September 2026



CERN FTS4: Establishing QA Procedures

Mihai Patrascoiu
on behalf of the FTS team

The ~~Elephant in the Room~~

CopyModePush Incident

Friday 04th September 2026

Dimitrios Christidis 05:56

Since yesterday around 16:40 CEST, we have been observing a surge of transfer errors across the Grid. For the most part, they appeared related to authentication, like:

```
copy: https: Permission denied: copy HTTP 401 : Authentication Error
```

Unlike the issue earlier this week related to the IGTF CA bundles, this affected many more sites and applied to both certificate and token transfers. While investigating, I disabled the use of tokens at one site and the error became:

```
copy: https: Host is down: copy HTTP 400 : Server Error
```

But now it appears to me that this issue is exclusive to FTS4. We're reassigning transfers back to FTS3 for the time being. Kindly review the situation at your earliest convenience.

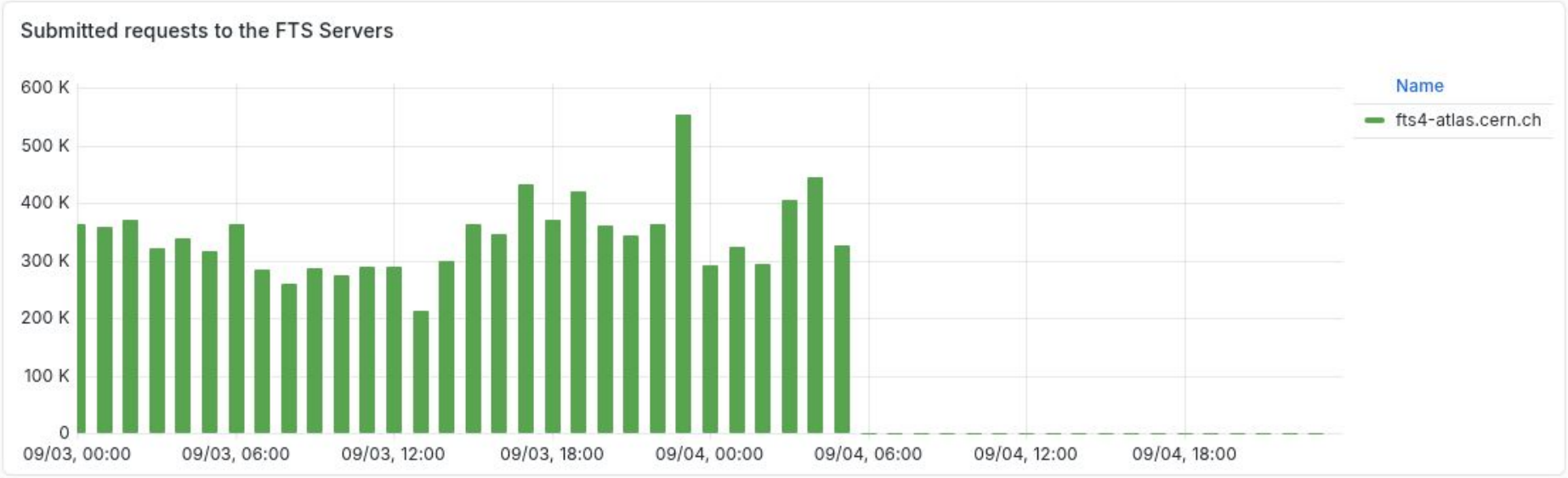
Joao Pedro Lopes 09:11

Hi Dimitrios,
We will have a look

Petr Vokac 09:20

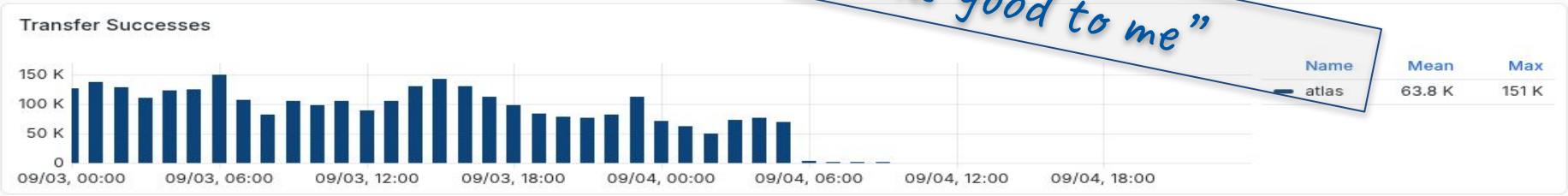
> new FTS4 version introduced incorrectly support for HTTP-TPC push?
>
> - HTTP COPY now include both Source: (HTTP-TPC pull) and Destination: (HTTP-TPC push) headers
> - Source: HTTP-TPC header use wrong davs:// protocol scheme

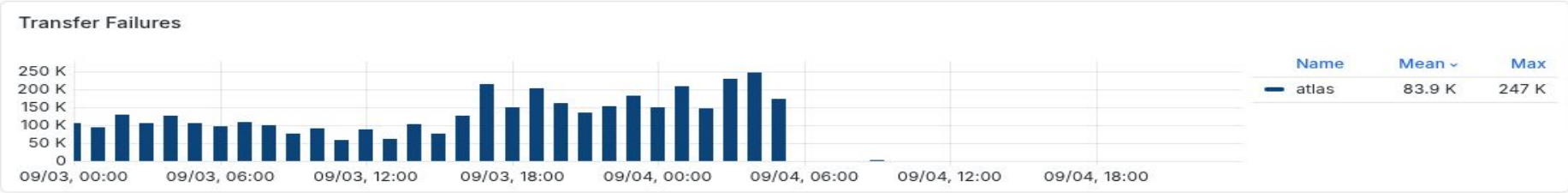
ATLAS submissions stopped (4th September, ~06:00)

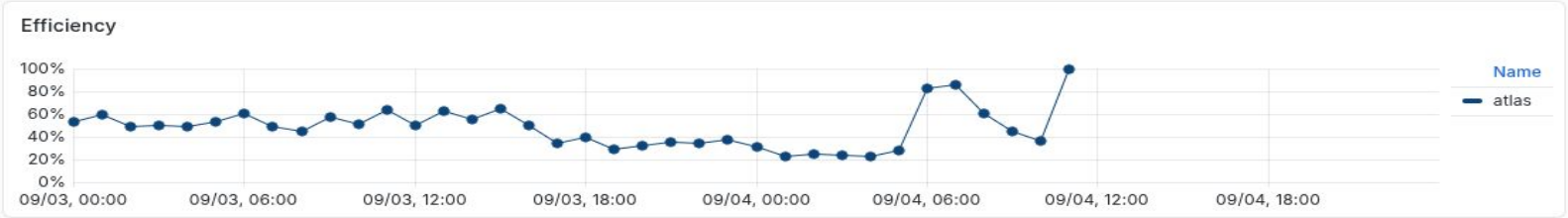


Monitoring

"Looks good to me"







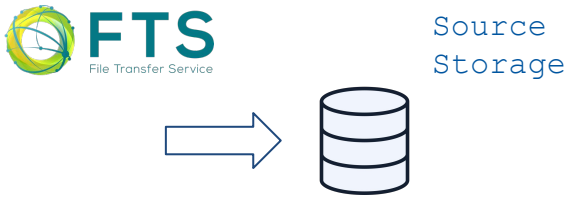
Monitoring

Moment
faulty code deployed
Transfer Successes
150 K
100 K
50 K
0
09/03, 00:00
09/03, 06:00
09/03, 12:00
09/03, 18:00
09/04, 00:00
09/04, 06:00
09/04, 12:00
09/04, 18:00
Name
Mean
Max
atlas
63.8 K
151 K
Transfer Failures
250 K
200 K
150 K
100 K
50 K
0
09/03, 00:00
09/03, 06:00
09/03, 12:00
09/03, 18:00
09/04, 00:00
09/04, 06:00
09/04, 12:00
09/04, 18:00
Name
Mean
Max
atlas
83.9 K
247 K
Efficiency
100%
80%
60%
40%
20%
0%
09/03, 00:00
09/03, 06:00
09/03, 12:00
09/03, 18:00
09/04, 00:00
09/04, 06:00
09/04, 12:00
09/04, 18:00
Name
atlas

Mechanism of action

1. FTS sends a malformed HTTP COPY Push request to the source storage
2. Source storage receives a malformed HTTP COPY from FTS
3. Confused source storage attempts to copy source file unto itself
4. File ends up truncated or deleted

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```



Timeline

- FTS4 change is deployed on ATLAS instance, Thursday, 3rd September, at 16:46
  - Malformed `HTTP COPY Push` requests are sent to the storages
  - Storages attempt to interpret the malformed requests

  → In some cases, the source file is deleted

- ATLAS spots unusual errors Friday, 4th September, at ~06:00
  - Redirects traffic from FTS4 to FTS3

- Situation is investigated by the FTS team Friday, 4th September, starting ~09:30
- FTS4 instances are shut off entirely Friday, 4th September, starting at 10:35

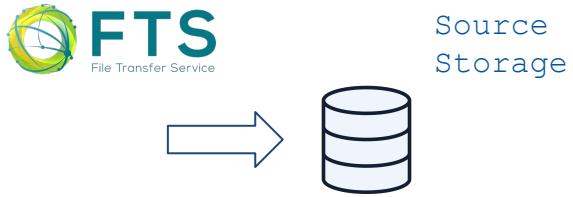
- Preliminary list of impacted files provided Friday, 4th September, at ~13:00
- Detailed investigation is launched

Incident Analysis

Behind the scenes… *FTS*

FTS sends a malformed HTTP COPY Push request to the source storage

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```



Invalid request (HTTP-TPC documentation*)

- Source [..] Must not be specified in the same request that defines the `Destination` header.
- Destination [..] Must not be specified in the same request that defines the `Source` header.

* WLCG HTTP TPC Technical Wiki

Behind the scenes... *Storages*

The Good

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```

```
HTTP 400: Bad Request
```

File left intact

The Bad

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```

```
HTTP 202: Accepted
```

Source file to be rewritten with source file

The Ugly

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```

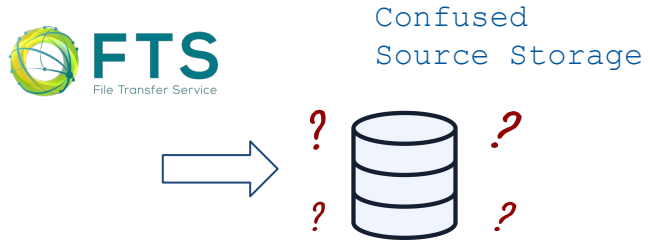
```
HTTP 500: Server Error
```

File left intact (fortunately)

Behind the scenes… *The Bad*

Source storage receives a malformed HTTP COPY from FTS

```
HTTP COPY <source-file>
  Host:        <source-se>
  Source:      <source-file>
  Destination: <dest-file>
...
```



Presence of `Source` header

→ Interpret as TPC Pull

→ Begin download from `<source-file>` to `<source-file>`

→ **File ends up truncated or deleted**

HTTP-TPC and Overwrite?

The HTTP-TPC page* mentions the following on overwrite:

> Controls whether the client desires the TPC request to overwrite an existing file.
>
> Valid values are T (overwrite any existing file)
> and F (fail request if file already exists).
>
> If not specified then T is used.

FTS does not specify the `Overwrite` header
→ Default true

* WLCG HTTP TPC Technical Wiki

HTTP-TPC and Macaroons?

The HTTP-TPC page* mentions the following on macaroons:

> Once FTS has the two macaroons (M_A and M_B) it never uses the delegated X.509 credential [not completely true, but doesn't matter].

Did not age like wine

FTS uses macaroons for both the active and passive parties, but also sends the certificate to the active party

→ Certificate grants write permission on the active party

Harmless when active party is the destination

Impactful when active party is the source

* WLCG HTTP TPC Technical Wiki

Putting it all together

1. FTS4 ported feature of configuring “Storage TPC Support Level”
    - Port yielded more `HTTP-TPC Push` transfers than should have
    - FTS4 only supported `HTTP-TPC Pull` → No Impact (undetected)

2. `HTTP-TPC Push` support implemented in FTS4
    - Problem in 1. triggers unexpected `HTTP-TPC Push` transfers

3. *Overwrite header is skipped since forever → Default true*
4. *Certificate is sent along with macaroons since forever → Write permissions on active party*

5. `HTTP-TPC Push` implementation in FTS4 has a bug
    - `Source` header is always added to the `HTTP-TPC Push` request
    - Compounded by 1., 3. and 4.

6. Storage interprets request with both `Source` and `Destination` as `HTTP-TPC Pull`
7. Storage attempts to write same file at same location → File may be lost

Incident Forensics

File detection

- Script to parse FTS4 logs is written, looking for the malformed `HTTP-TPC Push` signature in the log file
- A JSON entry for the impacted file is created

```
{
  "input": "<transfer-log-path>",
  "transfer_host": "fts4-atlas-004.cern.ch",
  "job_id": "a8c776e8-a5b5-11f1-a158-0c42a198fe6c",
  "file_id": 161779018,
  "source_se": "davs://eosctaatlas.cern.ch",
  "dest_se": "davs://eosatlas.cern.ch",
  "auth_method": "certificate",
  "activity": "Data Carousel Analysis",
  "source_surl": "<source-url>",
  "dest_surl": "<dest-url>",
  "user_filesize": 3983167528,
  "user_checksum_adler32": "52b418b3",
  "error_message": "<fts-error-message>",
  "err_time": "2026-09-03T18:19:13+02:00",
  "storage": "EOSCTA"
}
```

Running the scan

- **FTS4-ATLAS**
  - 14M+ log files analyzed, 1.4M faulty transfers identified
  - 367,410 unique files identified
  - 62 distinct storages involved

- **FTS4-CMS**
  - 53k log files analyzed, 0 faulty transfers identified

- Made available to ATLAS:
  - Complete dump of faulty transfers involved
  - Dump of unique files involved (first detected transfer, in order of time)
  - Summary table (impacted files / storage)

Forensics findings

- dCache storage **not impacted**
  → Correctly refuses request with `HTTP 400 Bad Request`
- StoRM storage **(presumed) not impacted**
  → Accepts request, fails with `HTTP 500 Server Error`, leaves file intact
- OAuth2 token authentication **not impacted**
  → Storage refuses request with `HTTP 401 Authentication Error`
- `Functional Test` activity → Impacted, but loss acceptable

pleasant surprise

- XRootD Storage → **truncates file** (0-size file in the namespace)
- EOS Storage → **deletes file** (disk + namespace)
- EOSCTA (EOS underneath) → **deletes disk replica** (file metadata + content on tape preserved)

Collected numbers

Storage	Impacted files	Status
dCache	240,218	Unaffected
XRootD	36,552	**Truncation risk**
StoRM	36,418	(Presumed) Unaffected
EOSCTA	31,140	**Deletion risk (disk replica)**
EOS	23,082	**Deletion risk**

Total impacted files – 367,410

Files at loss risk – **90,774**

OAuth2 files – 165,452 (45.03%)

`Functional Test` files – 157,725 (42.92%)

**At risk counts include OAuth2 & Functional Test activity*

Revised numbers & Recovery

- Upon excluding `Functional Test` activity and `OAuth2`:
  - EOSCTA: 31,140 files → Recovered by CTA team (tape safety measures in place)
  - XRootD: 12,955 files truncated
  - EOS: 6,272 files deleted

  → 19,227 files lost on disk

- ATLAS checked impacted files in Rucio catalogue for replication

  → Final tally: 6015 lost files

> *Sending our thanks again to ATLAS and EOSCTA colleagues !*

Future-proofing

How did it happen?

Perfect combination of:

- Hidden inaccuracies for years (i.e.: x509 + macaroon, missing `Overwrite` header)
- Porting of new features to FTS4
- Human mistake

… but most importantly

⚠ FTS4 has gradually evolved into running production traffic, whilst not having in-place the QA workflows of FTS production instances !

Software Follow-up

FTS

- [FTS-2528]: HTTP-TPC “CopyModePush” Incident
  - Already addressed the `TPC Push` error (Friday, 4th September 2026)
  - Address x509 + macaroon, `Overwrite` header
  - When transferring from tape, check source file existence at end-of-transfer or raise alarm otherwise
- Some of these changes will require synchronization with storages or experiments

XRootD

- [xrootd/xrootd#2925]: [XrdHttpTPC] Files can be deleted if src = destination
- [xrootd/xrootd#2930]: [XrdHttpTPC] Return 400 Bad request if Source AND Destination

Moving forward

- FTS4 has evolved past the “testing” or “prototype” phase
- Production QA workflows must be applied for the FTS4 instances
  - Intentional RPM deployment (Puppet + versionlocking)
  - Gradual deployment to an FTS4-Preprod instance
  - Experiments to validate with part of their activity
- Call to revive “WLCG dream” of having a test setup with FTS & real storages !

Conclusion

- Serious incident involving file deletion
  - Many “harmless looking” inaccuracies combined for a disastrous effect
  - Uncovered weaknesses in both FTS and storages
- Quick action from ATLAS and FTS team to contain the incident
- Tape T0 file recovery exercised successfully !
- FTS4 has matured very fast from testing to moving production traffic
  - Production QA workflows will be applied → We want the production traffic to continue

Thank you!

Discussion

- Tape safety & recovery methods were exercised at T0
  - Are WLCG T1s similarly prepared?
- Move to a tiered release + pre-prod deployment model system
  - Can experiments accommodate this?
  - FTS4 development and testing continues, but deployment will slow down
- OAuth2 authentication does not suffer from same credential area blast
  - Consider this a good incentive to advance on the token front?
- Views from experiments?