FTS Service Evolution and LHC Run-3 Operations

CHEP 2023

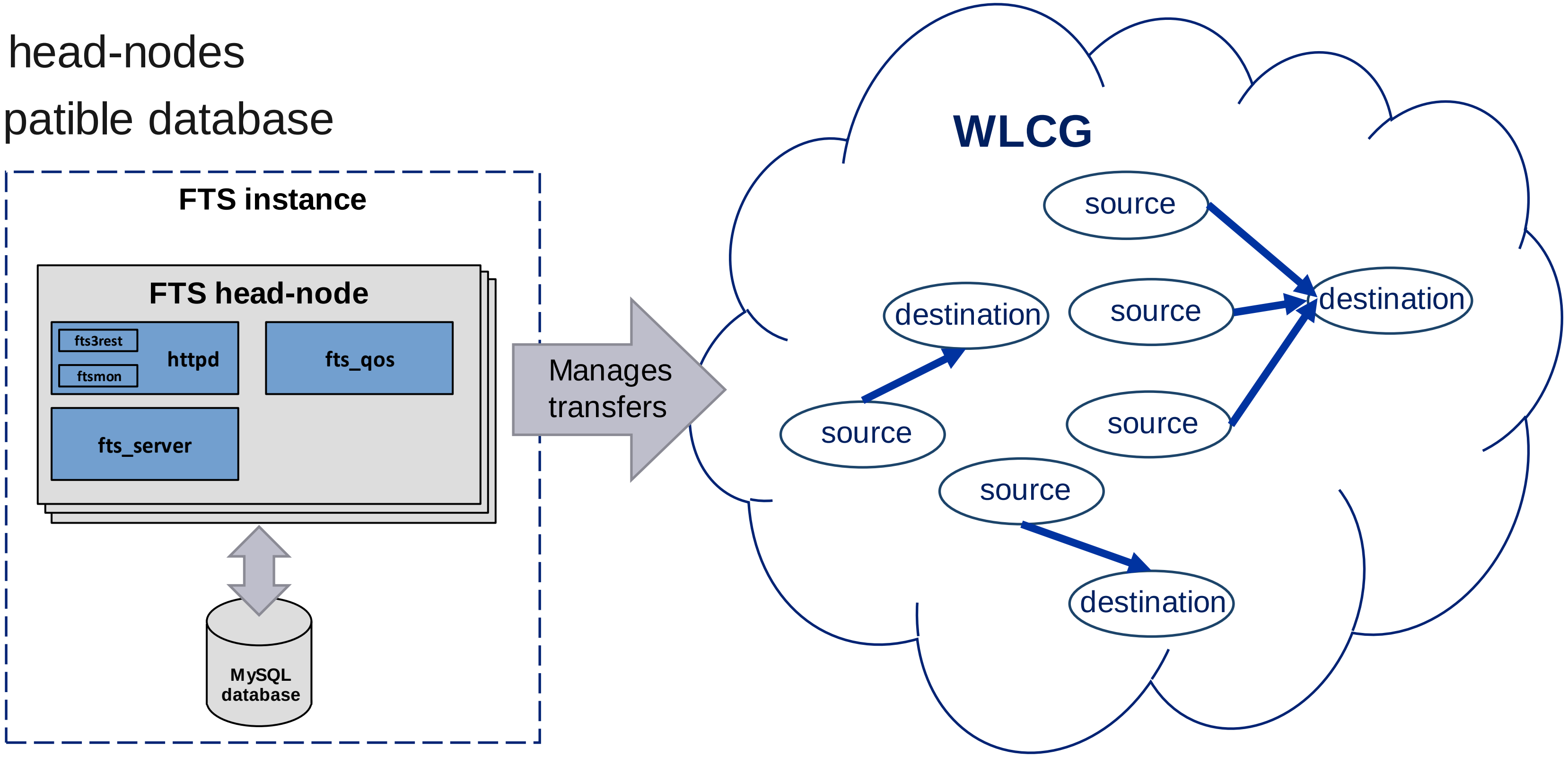
Presenter: Steven Murray

Authors: Joao Pedro Lopes, Shubhangi Misra, Steven Murray and Mihai Patrascoiu

Thursday 11th May 2023

What is FTS?

- **The File Transfer Service (FTS) queues, schedules and executes file transfers**
- **Used across the Worldwide LHC Computing Grid (WLCG)**
- **A typical FTS instance:**
  - Multiple identical head-nodes
  - A MySQL or compatible database



How users interact with FTS

- **Users submit file-transfer requests to FTS using command-line tools or a HTTP API**
- **Users can monitor the current progress of their transfers via FTS web pages**
- **FTS relies on the Data Management Clients (DMC) project (same team):**
  - Grid File Access Library version 2 (GFAL2)
    - GFAL2 Python bindings
    - GFAL2 command-line tools
  - Client library and command-line tools for HTTP file transfers - DaviX
  - SRM client library for GFAL2 and FTS - SRM-IFCE
  - Secure gSOAP client and server libraries - CGSI-GSOAP

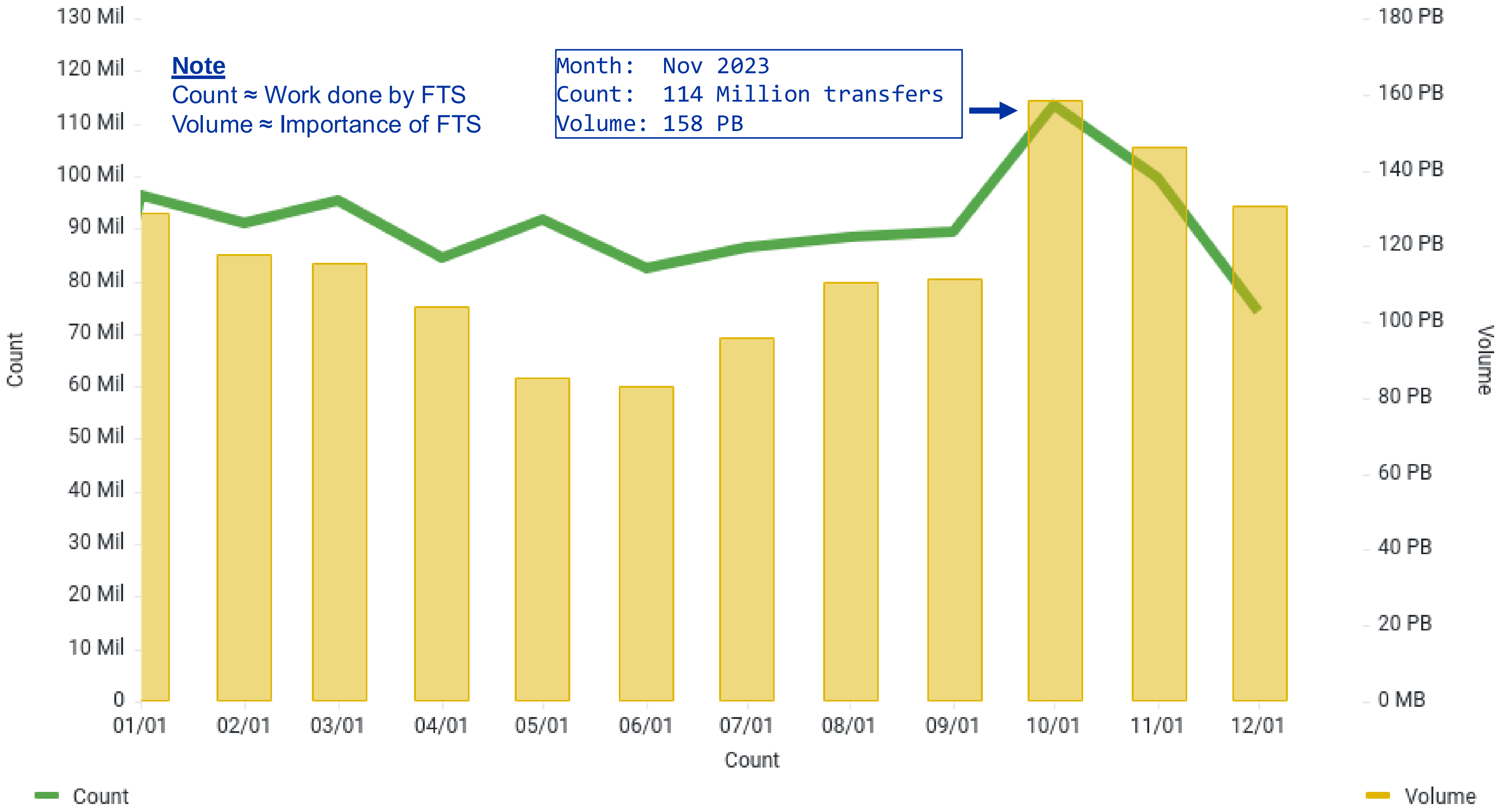
FTS team

Internal team		
Mihai Patrascoiu	Project leader	CERN
Steven Murray	Service manager	CERN
João Lopes	C++ / Python developer	CERN
Shubhangi Misra	C++ / Python developer	CERN

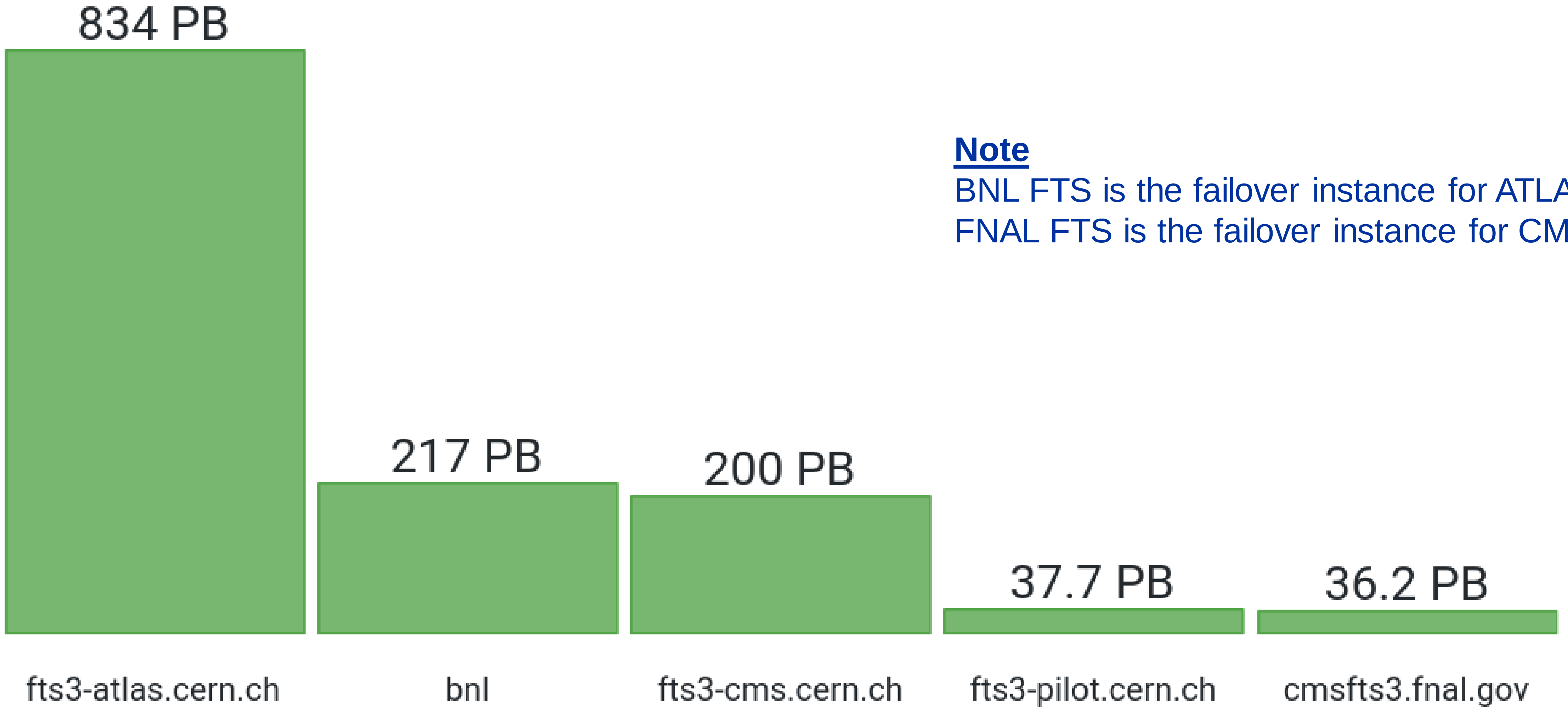
External contributors		
Ed Dambik	C++ developer	ATLAS / Indiana University Bloomington, USA
Eraldo Silva Junior	Python developer	ATLAS / LHCb /CERN / CBF, Brazil

Many thanks to other contributor both past and present

Successful WLCG FTS file transfers per month - 2022



Data volume transferred during 2022
Top 5 WLCG FTS instances



Note
BNL FTS is the failover instance for ATLAS
FNAL FTS is the failover instance for CMS

Migration from Python 2 to 3

- **During 2022 Python2 has been removed from all the FTS components**
  - Clients, REST interface and Web Monitoring
- **Migration did not bring functional changes to end-users in any of the components**
- **New Python3 client is now distributed via PyPI**
  - $ pip install fts3
- **New RPM is also available in the CERN repos (link) and EPEL**
  - $ yum install fts-rest-client
- **Old C++ client package will also be deprecated after v3.12.x**

Migration from Pylons to Flask

- **New project was created with this migration: FTS-REST-FLASK**
  - https://gitlab.cern.ch/fts/fts-rest-flask/
- **Flask was the chosen framework:**
  - Big user community, good documentation, simplicity and a rich ecosystem
  - Good integration with SQLAlchemy
- **No changes to the API**
  - The goal was to copy the structure and code as much as possible to avoid breaking things
- **Running in production at CERN since February 2022**
- **Distributed via RPM's in the FTS repositories [1]**
  - $ yum install fts-rest-server

[1] https://fts-repo.web.cern.ch/fts-repo/el7/x86_64/

CERN database deployment is “cloud” like

- **CERN Database on Demand (DBoD) service provides MySQL 8.0 databases to FTS**
- **Enables the CERN IT department to optimize the cost of running the FTS service**
- **The DBoD service priorities running hundreds of databases**
- **The FTS team and the DBoD service have made FTS more efficient to stay within the performance bounds of DBoD:**
  - Optimized FTS query used to decide what to stage from tape
  - Running OPTIMIZE TABLE once a week allows for the MySQL RAM cache to warm up in less than 20 minutes after a cold start
  - Migration to MySQL8
  - On-line Data Definition Language (DDL) operations (schema changes) remove the need to run with a dedicated and local flash drive storage solution
- **A fully puppetized, replicated MySQL database running on dedicated hardware with local SSD storage is ready for use in case of any unforeseen problems**

Deploying replica databases at CERN

- **Each CERN FTS instance has its own database backend:**
  - ATLAS
  - CMS
  - DAQ
  - LHCb
  - Pilot
  - Public
- **Each instance has a main and a replica database**
  - Main - Mission critical On-Line Transaction Processing (OLTP)
  - Replica - Monitoring On-Line Analytical Processing (OLAP)
- **The execution of fast mission critical database statements is isolated and protected from the execution of relatively slow and non-critical monitoring database statements**
- **A replica database could be used to recover the service if the main database failed**

Quick overview of the tape REST API

FTS
File Transfer Service

- **Modern and uniform way of managing tape data movements across the WLCG**
  - Replacement for SRM
- **Developed by EOSCTA, dCache and STORM as storages**
- **Developed by FTS/GFAL2 as clients**
- **Tape REST API Reference document [v1, May 2022]: Link**
- **GFAL2 v2.21.0 introduced support for the tape REST API**
- **Full support in FTS after v3.12.2:**
  - Deployed in all CERN production instances since Jan 2023
  - Allows experiments to pass staging and archiving metadata to tape endpoints

```
///src/plugins/http/gfal_http_plugin.cpp#L1160
extern "C" gfal_plugin_interface gfal_plugin_init(...)
{
    //[...]
    http_plugin.bring_online = &gfal_http_bring_online;
    http_plugin.release_file = &gfal_http_release_file;
    http_plugin.archive_poll = &gfal_http_archive_poll;
    http_plugin.bring_online_poll = &gfal_http_bring_online_poll;
    http_plugin.abort_files = &gfal_http_abort_files;
    //[...]
}
```

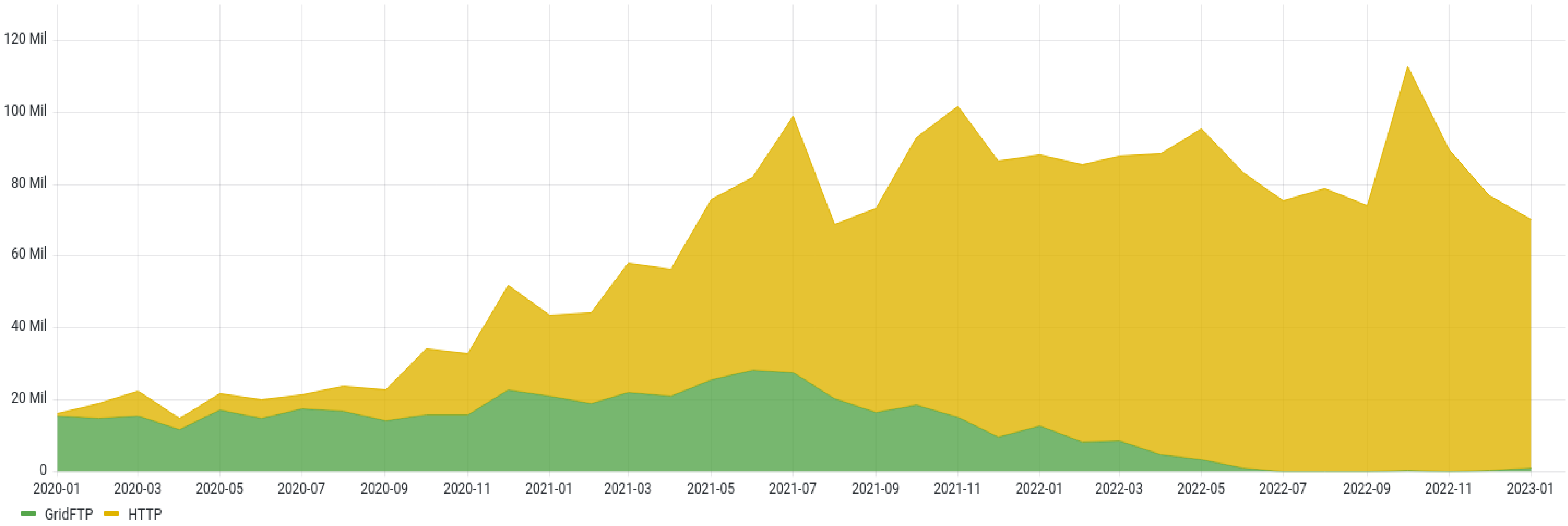
The tape REST API within FTS

FTS File Transfer Service

- **FTS must allow experiments to access their tape storage in the Grid**
- **The QoS daemon of FTS manages the necessary tape operations (QoS ≈ tape):**
  - Archive Monitoring:
    - After transferring a file, FTS tracks the file's transition from the disk buffer to tape
    - Allows experiments to mark on their catalogs data as safely stored on tape avoiding data-loss
  - Bring-online / Stage-in:
    - Brings data from tape storage to disk storage
  - Release file after recall:
    - When files are no longer needed on disk, FTS instructs the storage to delete them from the disk buffer
    - Crucial to manage small disk buffers
- **FTS uses GFAL2 as the client library to talk to remote storages endpoints**
- **Support for the tape REST API was added to FTS via GFAL2**

GridFTP is being phased out

Transfers per month managed by the CERN FTS instances

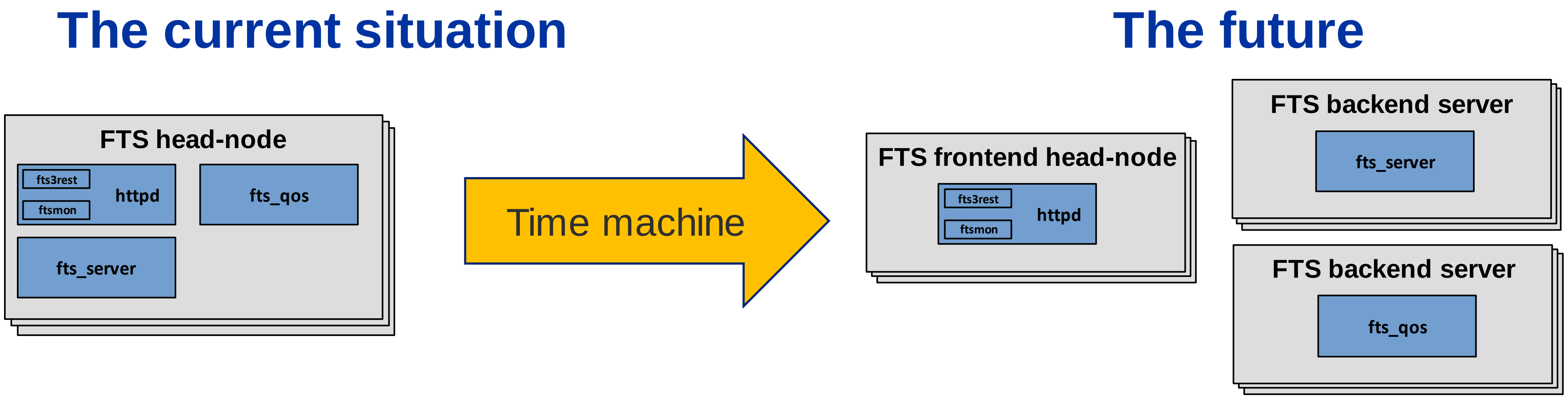


Future – Exascale Tokens for FTS

- **X509 proxy certificates are the main authentication method used by FTS**
- **FTS currently provides "unrefined" token support for HTTP file transfers**
- **The "Exascale tokens for FTS" project has started:**
  - FTS should allow storages to transition from X509 proxy certificates to 100% tokens
  - This is a 2 year project
- **This work is a pre-requisite for the WLCG's transition from X509 certificates to tokens**
- **The main stakeholders are:**
  - Worldwide LHC Computing Grid (WLCG)
  - Large Hadron Collier (LHC) experiments
  - Storage providers
  - Authentication and authorization services
  - European Grid Infrastructure (EGI) community
  - Open Science Grid (OSG) community
- **Current discussions center around how WLCG specific FTS should be:**
  - Implementing WLCG specific workflows offloads and centralizes development work from Rucio, Dirac and small to medium sized experiments
  - Being fully generic allows FTS to be used by "everything"

Future – True micro-service model

- The current FTS daemons must all run on the same machine
- When a new FTS head-node is added an instance of each daemon is added
- FTS is effectively a “Monolithic micro-service” – Is this a World first?
- To scale it should be possible to add machines dedicated to specific daemons
- We plan to decouple the FTS daemons so they can be deployed on different machines



Future - Outlook

- **Will protocol simplifications allow us to consolidate effort on client data management tools?**
  - GridFTP has nearly been phased out
  - SRM will eventually be replaced by the Tape REST API
- **We would like to improve the FTS scheduler to:**
  - Improve its predictability
    - The current splitting of scheduler decisions across all head-nodes breaks the strict FIFO order between some transfer requests
  - Revisit the model used to take decisions
    - Should we support priorities source storage endpoints that send files to a common destination storage endpoint?
- **Bottom-line – Support Run-4!**