EPJ Web of Conferences **295**, 01048 (2024) https://doi.org/10.1051/epjconf/202429501048
CHEP 2023

The CERN Tape Archive Beyond CERN

An Open Source Data Archival System for HEP

Michael Davis[1,*], *João* Afonso[1], *Richard* Bachmann[1], *Vladimír* Bahyl[1], *Jorge* Camarero Vera[1], *Julien* Leduc[1], *Pablo* Oliver Cortés[1], *Fons* Rademakers[1], *Lasse* Wardenær[1], and *Volodymyr* Yurchenko[1]

[1]CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland

Abstract. The CERN Tape Archive (CTA) is the successor to CASTOR and the tape backend to EOS. It was designed to meet the needs of archival storage of data from LHC Run–3 and other experimental programmes at CERN.

In the wider Worldwide LHC Computing Grid (WLCG), the tape software landscape is quite heterogeneous, but we are entering a period of consolidation. A number of sites have reevaluated their options and are choosing CTA for their future tape archival storage needs. However, CTA's original mandate imposed several design constraints which are not necessarily optimal for external sites.

In this contribution, we show how CERN has engaged with the wider HEP community and collaborated on improvements which allow CTA to be adopted more widely. We detail community contributions to allow CTA to be used as the tape backend for dCache; to facilitate migrations from other tape systems such as OSM and Enstore; and improvements to CTA building and packaging to remove CERN-specific dependencies and to allow easy distribution to external sites. Finally, we present a roadmap for the community edition of CTA.

1 Introduction

Until recently, the tape software landscape in High-Energy Physics was very heterogeneous. In March 2021, a report on tape evolution [1] at the WLCG[1] Grid Deployment Board, highlighted the diversity of tape software and disk frontends currently in use (Fig. 1).

Already by 2021, the situation was evolving. Several of the widely-used Free and Open Source software solutions were at end-of-life. CASTOR—in service at CERN since the late 1990s [2]—had been replaced by the CERN Tape Archive (CTA) [3, 4]. Open Storage Manager (OSM) [5] and Enstore [6] were no longer being actively developed. On the other hand, changing licensing models and costs for commercial tape software were making that option unattractive to many scientific institutes.

The tape software landscape has now entered a period of consolidation. Several sites have evaluated CTA and decided to adopt it as the successor to their legacy systems. CTA is an attractive choice for several reasons: it is Free and Open Source Software (FOSS), actively developed by CERN, with a long-term roadmap; it uses a modern software stack, integrated with the latest WLCG standards and protocols; it is performant, designed to meet the demands of LHC data processing; and it includes a suite of operational management tools.

*e-mail: michael.davis@cern.ch

[1]Worldwide LHC Computing Grid, https://wlcg.web.cern.ch/

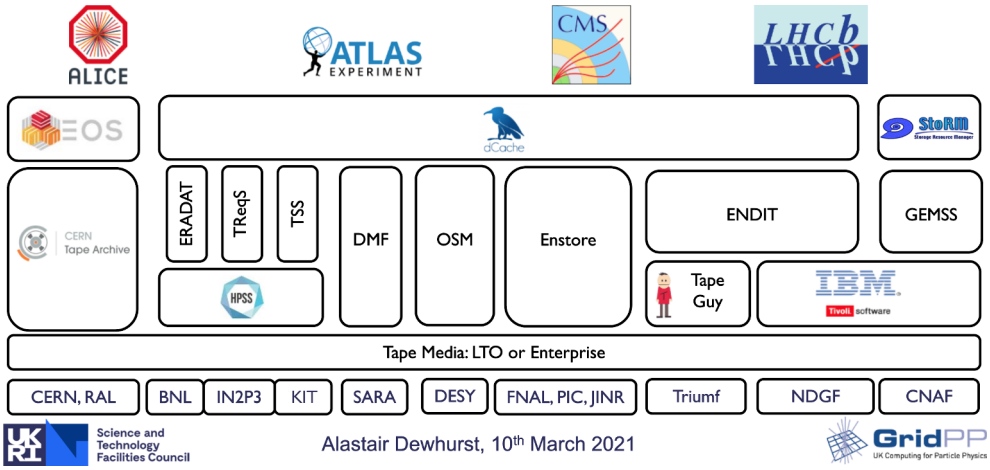


Figure 1: WLCG Tape Landscape in 2021. Site names are listed across the bottom. Each column in the figure represents the disk and tape software stack in use at each site.

Despite these advantages, there were certain obstacles to deploying and operating CTA beyond CERN. This paper describes how those obstacles were identified and overcome. The rest of the paper is organised as follows: § 2 discusses the development work required to make CTA usable at other sites. § 3 describes issues relating to software distribution and deployment, in particular CERN-specific software dependencies. § 4 describes how CERN supports operations at other sites. The paper concludes with a summary in § 5.

2 Making CTA usable beyond CERN

CTA was designed as the successor to CASTOR and as the tape backend to the EOS disk system [7]. In this section, we discuss how these design goals facilitated—or hindered—its use at external sites, and the problems which had to be overcome.

2.1 Migrating existing tape archives to CTA

To replace CASTOR, around 340 PB of data had to be migrated from CERN's existing tape archive. Physically copying this amount of data would have taken several years, with a major impact on operations. Consequently, it was decided that CTA should use the same physical tape format as CASTOR, so that migration could be a pure metadata operation [3, 4]. A set of migration tools were developed to inject file and tape metadata from the CASTOR database into the EOS namespace and CTA Catalogue.

A metadata-only migration is desirable for other sites, for exactly the same reasons. However, for most sites—apart from RAL, who were also migrating from CASTOR—their existing tapes were not readable by CTA. At a round-table with all stakeholders [8], it was agreed that CTA should be able to read (but not write) data in non-native formats, in order to facilitate migration, while writing all post-migration data in CTA's native format.

2.1.1 Reading non-native file formats

CTA's interface to the physical tape hardware (libraries, drives, cartridges) is handled by the tape server daemon. The tape server would need to read two additional tape file formats:

Figure 2: ANSI AUL tape data format [11] used by CASTOR/CTA, with descriptors for volume label (VOL1), headers (HDR), user headers/trailers (UHL/UTL), end-of-file (EOF); tape marks (TM); and the payload (DATA). Descriptors are stored in tape blocks of 80 bytes; for more details see [12].

OSM (tape backend to dCache [9]) and Enstore. There are two parts to the tape format: volume label and file descriptors (Fig. 2). The volume label contains a unique identifier and other tape-specific metadata. Enstore's volume label is compatible with CTA; OSM has a 64 KB label with a different structure. The file descriptors contain metadata including each file's unique file sequence number and starting block ID, used to check that the tape head is correctly positioned before starting a read/write operation. CTA has fixed-length descriptors; the Enstore and OSM tape file formats are based on CPIO [10].

The CTA tape server now had to be modified to read the new tape formats. The code is an evolution of the CASTOR Tape Server, which had been re-written and optimised in 2015 [13, 14]. Re-using the CASTOR code in CTA guaranteed compatibility with the CASTOR tape format and included this recent performance optimisation work. However, it also brought CASTOR's legacy data structures with it. In CASTOR, all file reading and writing operations were handled by the `File` C++ class. A single monolithic `ReadSession` method handled checking tape volume labels, positioning, reading headers and reading the file. As this was not easy to generalise to other tape formats, the code was refactored, with a new class for each discrete operation (Fig. 3a). The three classes concerned with reading files—`ReadSession`, `HeaderChecker` and `FileReader`—were specialised for each file format (Fig. 3b). (The other classes do not need to be specialised as CTA writes files only in its native format). Further implementation and testing details were presented at the 2023 EOS Workshop [15].

CTA not only has the capability to read tapes written by Enstore or OSM; the refactoring work allows other tape file formats to be added as required. The last piece of the puzzle is to populate the CTA Catalogue with the metadata for each migrated tape.

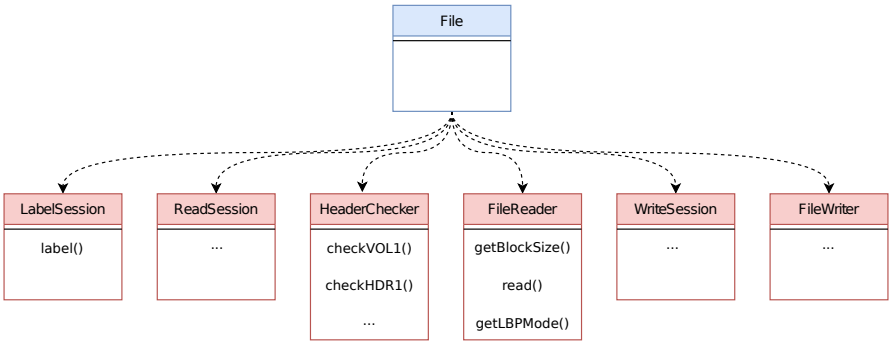
2.1.2 Migrating metadata

At CERN Tier–0, the CASTOR tape archive was migrated to CTA in two separate metadata operations [4]: tape and tape file metadata migration from the CASTOR DB to the CTA Catalogue DB; and disk file metadata injection into the EOS namespace.

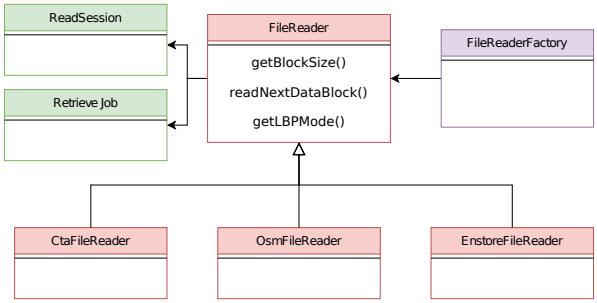
Other sites migrating from CASTOR were able to use the CERN migration tools. RAL took this opportunity to consolidate two CASTOR instances into a single CTA instance [16].

Sites migrating from Enstore or OSM had to develop their own metadata migration scripts, using the CERN scripts as a template. For sites deploying CTA behind EOS, a slightly-modified version of the EOS namespace injection tools is provided [17].

For migrations from Enstore, one item of metadata is missing: the block ID of each file. CTA has the capability to seek by file sequence number (fSeq) or by the ID of the first block in the file. Searching by block ID is more efficient, because the ID of the block at the end of each tape wrap is stored in the cassette's internal memory. Furthermore, CTA's Recommended Access Order feature [18, 19] depends on the block ID. As Enstore stores only the fSeq, migrated files suffer a seek time penalty in CTA. In theory, the block IDs could be subsequently entered in the CTA Catalogue, but as this requires positioning to every file, it is unfeasible at the scale of the entire archive. The performance penalty for migrated files will gradually disappear as the archive is repacked to new media written in CTA's native format.



(a) The monolithic `File` class was refactored into six new classes, each with clearly-defined scope



(b) Each tape format has its own concrete implementation of the new `FileReader` abstract (base) class

Figure 3: Refactoring of the CTA Tape Server code to allow reading multiple tape formats

2.2 Choice of disk system

Some WLCG sites would like to use CTA with dCache [20] rather than EOS. In principle, CTA is agnostic to the disk system in front. Whereas CASTOR was a Hierarchical Storage Manager (HSM), including a persistent disk cache, CTA was designed as a pure tape archival system, optimised for throughput. Disk functions are delegated to EOS, but a different disk system can be used, as long as it provides the disk functions listed in Table 1. In practice, CTA made a limiting assumption that the disk file ID would be a 64-bit unsigned integer (as in EOS), which made it incompatible with dCache, which uses a (UUID-like) string.

Table 1: Mapping of tape system functions to software components

Function	**Software component**
File Metadata Operations	EOS (MGM/XRootD)
Namespace	EOS (QuarkDB)
Disk Buffer for Staging	EOS (FST)
Tape File Metadata Operations	CTA (Frontend)
Archive/Recall Requests	CTA (Scheduler DB)
Tape File Catalogue	CTA (Catalogue DB)
Tape Operations (libraries, drives, cartridges)	CTA (Tape Server)

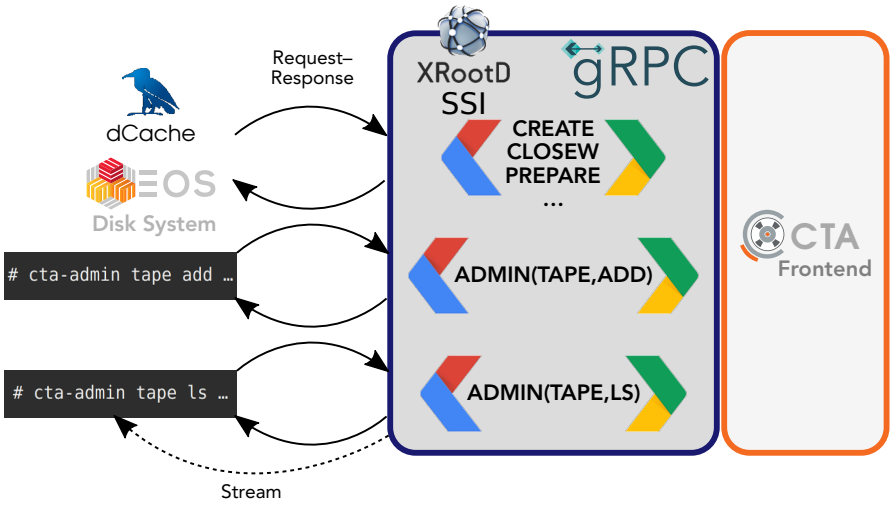


Figure 4: CTA Frontend Transport protocol. Payloads are Google Protocol Buffers; transport is XRootD SSI (EOS) or gRPC (dCache). CTA Frontend request-handling code is common to both transport protocols. `cta-admin` streaming and non-streaming commands are implemented for both protocols.

2.2.1 Type of the Disk File ID

The disk file ID problem was not as serious as it first appeared. CTA's internal representation was already a string, so changing the CTA Frontend protocol to also use a string was fairly straightforward. However, there were several places in the CTA code where the string representation was converted back into an integer: the command to list tape files and the "recycle bin" tools to restore deleted files [17]. In order to decouple CTA from any specific disk system, this functionality will be removed from the CTA Frontend and re-implemented in the client-side tools. (An equivalent set of tools will need to be implemented for dCache).

2.2.2 CTA Frontend Transport Protocol

A second obstacle was the transport protocol between the disk system and CTA. There are two types of request: (1) events from the disk system ("create", "close write", "prepare", "delete"), which enqueue a request to the CTA Scheduler and (2) requests from the CTA command-line interface to control tape drives, configure tapepools and archive routes, manage queues or list tape and tape file metadata. For commands with an arbitrarily-long response (such as listing files on a tape), the response is returned as a stream. Requests, responses and stream items are all serialised using Google Protocol Buffers (protobuf) [21]. See Fig. 4.

The protobufs are transported using XRootD with the Scalable Service Interface (SSI) extensions [22]. XRootD SSI is performant and has the necessary functionality to handle both request-response and streaming requests. However, the dCache XRootD client implementation does not include the SSI extensions. Conversely, the protobufs used for the payload have their own native transport, Google Remote Procedure Calls (gRPC).

To avoid the dependency on XRootD SSI, it was decided to implement gRPC transport for the CTA Frontend. A reference implementation was contributed by the dCache developers [9], including a proof-of-concept for gRPC streaming and Kerberos authentication, for the CTA Admin commands. The CTA Frontend code was refactored to separate the transport protocol layer from the unpacking of the protobufs and command dispatcher code. The gRPC transport layer is currently being integrated with the CTA Frontend code base and will be available in a future release.

3 CTA Software Dependencies and Distribution

The first CTA releases were source code only, which external sites had to compile and build. This was challenging due to CTA's dependencies on packages internal to CERN. An additional complication was that CERN used Oracle as its production database, but many external sites do not have an Oracle license. This section describes how the CTA team responded to the need for easier distribution and deployment.

3.1 CTA Catalogue Database

From its inception, CTA was designed to be agnostic to the database backend. While CASTOR relied on Oracle-specific optimisations and PL/SQL, CTA's strategy was to use only standard SQL tables and queries, to maintain portability and avoid vendor lock-in [23]. Initially, CTA supported four DB backends: Oracle, PostgreSQL, MySQL and SQLite. Support for MySQL was implemented by IHEP [3] but was dropped in 2021 when IHEP moved to Postgres [24]. SQLite is used for unit tests and is not intended for production deployment.

Thus CTA now has two supported databases for production use: Oracle (used at CERN) and PostgreSQL. The CTA Catalogue library depends on both database implementations, even though only one will be used. This introduced a dependency on a CERN-specific Oracle RPM which could only be distributed internally, for licensing reasons. In 2022, the code was repackaged to remove this dependency in favour of the publicly-available Oracle RPMs. In an upcoming release, the hard dependency will be replaced by a database backend plugin, obviating the need to install Oracle RPMs for sites running Postgres.

One other notable development is a set of tools to manage the CTA Catalogue schema upgrade procedure, to allow transparent upgrades on the CTA production service [25].

3.2 CTA Public Release

The CERN Tape Archive source code was released in 2016 under GNU GPL v3 [26]. Removing the dependencies on CERN-internal RPM packages was the first step to a public binary release.

The CTA team now provides a public repository for all CTA dependencies [27] (except Oracle libraries, which cannot be redistributed, as mentioned above). In 2023, the CTA team announced [28] the first public binary release for CTA's production deployment platform, CERN CentOS 7. Binary releases for other RHEL versions and clones will follow in 2024.

4 CTA Operations and Support

4.1 CTA Operations Utilities

The CTA operations tools were originally developed for CERN internal use, but in response to many requests, the CTA team have invested effort in generalising and packaging these tools for use at other sites. The CTA Operations utilities [29] are a suite of tools for tasks such as media lifecycle management, disk buffer management and monitoring (Table 2). The tools are written in Python and available as `pip` packages, under a GNU GPL v3 license.

In addition to the code, the repository provides a selection of monitoring examples which may be useful for sites with a similar setup to CERN [31].

Table 2: CTA Operations Tools.

Package	Function
ctautils	A collection of helpers and wrappers used across all operations tools
tapeadmin	Library for interacting with CTA and tape infrastructure
ATRESYS	Automated Tape REpacking SYStem [30]
ctaopsadmin	Configurable wrapper for `cta-admin` and hardware interaction scripts
ctaopseos	Tools for EOS–CTA interactions
poolsupply	Automation for supply of tape pools with new cartridges
tapeverify	Tools for automatic data integrity checks

4.2 The CTA Community

The CTA team organises an annual CTA Day co-located with the EOS Workshop [32]. At the 2023 workshop, there was an active exchange of ideas, with six external sites sharing their experiences with CTA. The collaboration between CERN and external sites is mutually beneficial, as other sites contribute code (§ 2, 3), documentation [33], bug reports and fixes. Support for external sites is provided through the CTA Community forum [34].

5 Summary

This paper describes the work done by the CTA team and contributors in the CTA Community to make the CERN Tape Archive usable at WLCG Tier–1s and other sites beyond CERN. The scope of the work includes: the ability to migrate existing data archives from legacy tape systems; compatibility with other disk systems besides EOS; a choice of database backends; and the CTA public binary release. In addition, we described how CERN provides support to external sites through the CTA operations tools and the CTA Community.

DESY's experience of migrating from OSM to CTA is published in [9]. We look forward to similar reports from other sites in due course.

Future work includes full integration of gRPC support for the CTA Frontend, and a plugin for the database backend. Additional operations tools will include the EOS–CTA namespace reconciliation scripts [35] and tools for managing ACLs and tape drive metadata. In addition, we plan to package the tools as an RPM, to allow them to be version-locked to compatible CTA versions, for better management of external dependencies (not handled by `pip`).

Acknowledgements. This work has been a true community effort. We acknowledge the contributions of the dCache developers and CTA users, in particular DESY, FNAL, IHEP, RAL, PIC and AARNet.

References

[1] A. Dewhurst, *Tape Evolution pre-GDB report*, https://indico.cern.ch/event/876787/ (2021), WLCG Grid Deployment Board

[2] G. Lo Presti et al., *CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN*, in *24th IEEE Conf. Mass Storage Systems Tech.* (MSST 2007), pp. 275–280, `https://doi.org/10.1109/MSST.2007.4367985`

[3] E. Cano et al., EPJ Web Conf. **245**, 04013 (2020)

[4] M. Davis, *Migration from CASTOR to the CERN Tape Archive*, `https://indico.cern.ch/event/1078853/contributions/4579745/` (2021), HEPiX Autumn 2021 Workshop

[5] S. Brand, P. Fuhrmann, Computer Physics Communications **110**, 131 (1998)

[6] J. Bakken et al., *The Fermilab data storage infrastructure* (2003)

[7] S. Murray et al., Journal of Physics: Conference Series **898**, 062013 (2017)

[8] M. Davis, *CTA tape format support: BoF discussion*, https://indico.cern.ch/event/1103358/contributions/4760546/ (2022), 6th EOS Workshop
[9] T. Mkrtchyan, J. Chodak, M. Karimi, R. Lueken, S. Meyer, P. Suchowski, C. Voss, *dCache integration with CERN Tape Archive* (2024), to be published in proceedings of CHEP 2023
[10] *IEEE Std 1003.1–1988 (ANSI/IEEE Standard Portable Operating System Interface for Computer Environments)*, https://archive.org/details/POSIX.1-1988 (1988)
[11] *ISO/IEC 1001:2012. Information technology—File structure and labelling of magnetic tapes for information interchange*, https://www.iso.org/standard/60220.html (2012)
[12] *CAStor Tape Server's Handbook*, https://gitlab.cern.ch/castor/CASTOR/-/blob/master/castor/tape/tapeserver/documentation/TapeServer.pdf (2013)
[13] S. Murray et al., Journal of Physics: Conference Series **396**, 042042 (2012)
[14] E. Cano et al., Journal of Physics: Conference Series **664**, 042007 (2015)
[15] J.C. Vera, *External tape readers: integration into CTA and OSM/Enstore cases*, https://indico.cern.ch/event/1227241/contributions/5335997/ (2023), 7th EOS Workshop
[16] T. Byrne, *Technical challenges of tape instance consolidation at RAL*, https://indico.cern.ch/event/1227241/contributions/5335998/ (2023), 7th EOS Workshop
[17] L.T. Wardenaer, *Disk File Metadata for Tape Files—Migrating, Restoring, Replicating*, https://indico.cern.ch/event/1227241/contributions/5335992/ (2023), 7th EOS Workshop
[18] C.G. Moraru, Master's thesis, University Politehnica of Bucharest, Hungary (2017), https://cds.cern.ch/record/2282014/files/CERN-THESIS-2017-131.pdf
[19] R. Bachmann, *CERN's Run 3 Tape Infrastructure*, https://indico.cern.ch/event/1123214/contributions/4821966/ (2023), 7th EOS Workshop
[20] T. Mkrtchyan et al., *dCache: Inter-disciplinary storage system* (2021), https://github.com/dCache/dcache
[21] *Google Protocol Buffers*, https://protobuf.dev/overview/
[22] M. Davis, *Building client-server APIs using the XRootD Scalable Service Interface*, https://indico.cern.ch/event/656157/contributions/2866317/ (2018), 2nd EOS Workshop
[23] M.C. Davis et al., EPJ Web Conf. **214**, 04015 (2019)
[24] Y. Bi, *EOS and CTA Status at IHEP*, https://indico.cern.ch/event/1103358/contributions/4760540/ (2022), 6th EOS Workshop
[25] C. Caffy, M. Barros, *Updating the database schema for services in production*, https://indico.cern.ch/event/1163544/ (2022), CERN internal presentation
[26] *GNU General Public License version 3*, https://www.gnu.org/licenses/gpl-3.0.en.html (2007)
[27] *CTA software dependencies repository*, https://cta-public-repo.web.cern.ch/cta-4/el-7/cta-dependencies/
[28] *CTA public binary release announcement*, https://cta-community.web.cern.ch/t/public-release-4-5-8-7-1-announcement/217 (2023)
[29] *CTA Operations Utilities*, https://gitlab.cern.ch/cta/cta-operations-utilities/-/wikis/home
[30] V. Bahyl, *ATRESYS—Automated Tape REpacking System, a tool for managing CTA repacks and tape lifecycle*, https://indico.cern.ch/event/1227241/contributions/5366313/ (2023), 7th EOS Workshop
[31] R. Bachmann, *Monitoring your EOSCTA deployment—the general recipe*, https://indico.cern.ch/event/1227241/contributions/5335995/ (2023), 7th EOS Workshop
[32] *CTA Day at the 7th EOS Workshop*, https://indico.cern.ch/event/1227241/timetable/#20230426.detailed (2023)
[33] *The CTA+dCache Book*, https://ctadcache.gitbook.io/the-cta-book/
[34] *CTA Community Discourse forum*, https://cta-community.web.cern.ch/
[35] R. Bachmann, *Maintaining consistency in an EOSCTA system*, https://indico.cern.ch/event/1103358/contributions/4760538/ (2022), 6th EOS Workshop