EPJ Web of Conferences **295**, 07020 (2024) https://doi.org/10.1051/epjconf/202429507020
CHEP 2023

Federated Heterogeneous Compute and Storage Infrastructure for the PUNCH4NFDI Consortium

Alexander Drabent[1], *Oliver* Freyermuth[2], *Manuel* Giffels[3,*], *Matthias* Hoeft[1], *Jörn* Künsemöller[4], *Benoit* Roland[3], *Dominik* Schwarz[4], and *Christoph* Wissing[5]

[1]Thüringer Landessternwarte Tautenburg, Germany
[2]University of Bonn, Germany
[3]Karlsruhe Institute of Technology (KIT), Germany
[4]Bielefeld University, Germany
[5]Deutsches Elektronen-Synchrotron (DESY), Germany

Abstract.
PUNCH4NFDI, funded by the Germany Research Foundation initially for five years, is a diverse consortium of particle, astro-, astroparticle, hadron and nuclear physics embedded in the National Research Data Infrastructure initiative. In order to provide seamless and federated access to the huge variety of compute and storage systems provided by the participating communities covering their very diverse needs, the Compute4PUNCH and Storage4PUNCH concepts have been developed. Both concepts comprise state-of-the-art technologies such as a token-based AAI for standardized access to compute and storage resources. The community supplied heterogeneous HPC, HTC and Cloud compute resources are dynamically and transparently integrated into one federated HTCondor-based overlay batch system using the `COBalD/TARDIS` resource meta-scheduler. Traditional login nodes and a JupyterHub provide entry points into the entire landscape of available compute resources, while container technologies and the CERN Virtual Machine File System (CVMFS) ensure a scalable provisioning of community-specific software environments. In Storage4PUNCH, community supplied storage systems mainly based on dCache or XRootD technology are being federated in a common infrastructure employing methods that are well established in the wider HEP community. Furthermore existing technologies for caching as well as metadata handling are being evaluated with the aim for a deeper integration. The combined Compute4PUNCH and Storage4PUNCH environment will allow a large variety of researchers to carry out resource-demanding analysis tasks. In this contribution we will present the Compute4PUNCH and Storage4PUNCH concepts, the current status of the developments as well as first experiences with scientific applications being executed on the available prototypes.

1 Introduction

Particles, Universe, NuClei and Hadrons (PUNCH) for the NFDI is a consortium funded by the National Research Data Infrastructure (German: Nationale Forschungsdateninfrastruktur – NFDI). PUNCH4NFDI represents around 9,000 scientists with PhD from universities

*e-mail: Manuel.Giffels@kit.edu

and research institutes, including the Max Planck Society, the Leibniz Association, and the Helmholtz Association in Germany.

The related scientific fields are facing similar data analysis challenges defined by the increasing amount of data generated by research infrastructures and complex algorithms that yield a high demand for computing resources. The fundamental idea behind forming the PUNCH4NFDI consortium is to benefit together from the experiences, concepts, and tools available in the diverse communities. The prime goal of PUNCH4NFDI is to set up a federated and "FAIR" science data platform, offering infrastructures and interfaces necessary to access and use data as well as computing resources of the involved communities and beyond [1]. This contribution focuses on the federation of the available heterogeneous computing and storage infrastructures in Germany and how to provide unified and seamless access to them.

2 Federated Heterogeneous Compute Infrastructure – Compute4PUNCH

A substantial amount of High-Throughput Compute (HTC), High-Performance Compute (HPC) and Cloud resources are provided as so-called in-kind resource contributions by the PUNCH4NFDI institutions distributed all over Germany. However, a most effective and user-friendly utilization of a variety of resources with different architectures, operating systems, software environments and authentication mechanisms poses tremendous challenges. This is even aggravated by the fact that all of those resources are already in operation and mostly shared among different communities, so that the mandatory requirements to resource providers and hence potential modifications of their systems should be minimized in order to minimize the interference with their operational concept.

To provide seamless, transparent and uniform access to these variety of community supplied heterogeneous compute resources for diverse communities, the Compute4PUNCH concept has been developed, following the idea of establishing a nationwide federated heterogeneous compute infrastructure for the PUNCH4NFDI sciences, adopting already existing technologies developed within the context of federated computing infrastructures in the German High Energy Physics (HEP) community [2] as much as possible in order to maximize synergies.

One basic building block of Compute4PUNCH is the dynamic and on-demand integration of all available compute resources into a single so-called Overlay Batch System (OBS) based on HTCondor [3] forming a single federated pool of heterogeneous compute resources. On the one hand HTCondor seems to be a good fit for the needs of the majority of PUNCH4NFDI communities and on the other hand it is perfectly suited for a dynamic extension of its compute resource pool. The actual integration is achieved by executing placeholder jobs, so-called pilots or drones containing HTCondor Startds on the compute resources allowing for a late binding of the tasks in the queue to the most appropriate integrated compute resource.

Another basic building block of Compute4PUNCH is the `COBalD/TARDIS` [4–7] meta-scheduler. `COBalD/TARDIS` is responsible for the decision making to further extend or reduce the number of integrated resources of a certain kind based upon the current resource utilization and therefore the actual demand for it, instead of trying to perform a complex and error-prone global matchmaking prediction for all kind of jobs and all kind of resources. In addition, `COBalD/TARDIS` takes care of the resource life-cycle management and comprises a variety of so-called site adapters to manage resources via the most common batch systems and cloud APIs.

The Compute4PUNCH infrastructure is designed to provide multiple entry points into the federated compute resource pool by utilizing the same Helmholtz authentication and au-

thorization infrastructure (AAI) [8] based upon the Unity IdM [9] and following the AARC Blueprint Architecture [10]. Traditional Linux login nodes for user job submission are provided by using an Open ID Connect (OIDC) enabled SSH service utilizing motley_cue – a service for mapping OIDC identities to local identities, pam-ssh-oidc – a PAM module that accepts OIDC access tokens for user authentication and mccli – an SSH client wrapper for connecting to an OIDC-enabled SSH server [11]. In addition, the deployment of a JupyterHub infrastructure utilizing HTCondor´s `condor_ssh_to_job` feature to exploit resources without inbound network connectivity [12] as well as traditional Grid Compute Elements primarily for HEP use-cases are foreseen in the near future.

The provisioning of dedicated operating systems and specific software environments to support the diverse PUNCH4NFDI communities is ensured by using modern container technologies like Apptainer [13] in combination with the CERN Virtual Machine Filesystem CVMFS [14]. The necessary container build instructions (Dockerfiles) are hosted in the PUNCH4NFDI GitLab instance. Daily and on-demand builds using a Continuous Integration (CI) workflow ensure the correctness of the build instructions as well as the integration of the latest security updates and in the end upload the ready-to-use container into the PUNCH4NFDI GitLab Container Registry. To allow for a scalable distribution of the containers, they are converted into the Apptainer unpacked format and distributed via the CERN's `unpacked.cern.ch` CVMFS repository[15]. Depending on the resource provider, CVMFS is either directly installed on the bare-metal or using the cvmfsexec tool allowing unprivileged users to mount CVMFS using a mount namespace.

A blueprint of the future federated heterogeneous compute and storage infrastructure foreseen for PUNCH4NFDI is depicted in fig. 1.

By implementing the above-described Compute4PUNCH concept, the mandatory requirements to the resource providers could be reduced to activated user and mount namespaces to support CVMFS and nested container execution as well as the availability of outgoing network connectivity from their worker nodes.

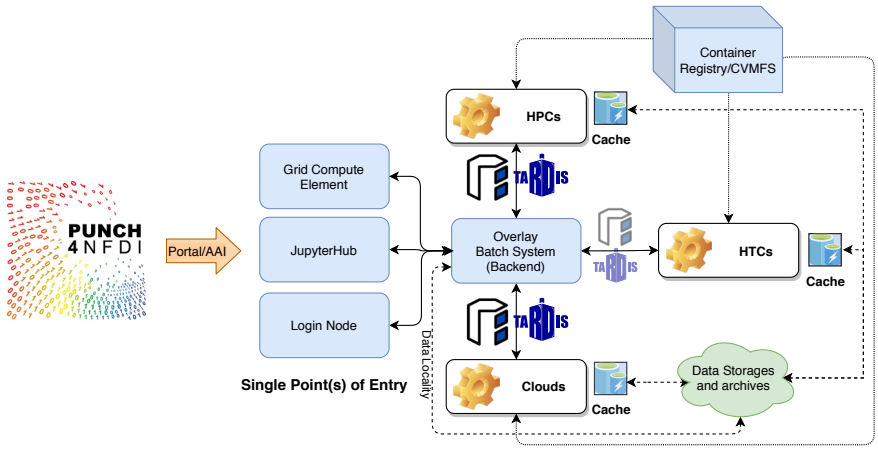


Figure 1: Blueprint of the future federated heterogeneous compute and storage infrastructure of PUNCH4NFDI.

As of the time of writing this paper, a demonstrator of the Compute4PUNCH infrastructure is available with three integrated computing resources, one of each kind HPC, HTC and Cloud. Three additional resources are in the progress of being integrated. The demonstra-

tor includes as well a traditional Linux login node for user job submission using an OIDC-enabled SSH service and a GitLab based container registry including CI workflows to build and upload containers.

3 Federated Storage Infrastructure – Storage4PUNCH

The distributed computing infrastructure as described above is complemented by a distributed storage infrastructure to allow for long term storage of data products. Access to the storage components is granted also via Tokens that are provided from the Helmholtz AAI. This implies that for data access dedicated protocols have to be used. Like in the Worldwide LHC Computing Grid (WLCG) the WebDAV protocol and the XRootD protocol are going to be used. Obvious choices for distributed storage components are the well known products that are deployed in WLCG. Since the German community has most experiences with dCache and XRootD systems prototype installations based on these two technologies have been opted for in the PUNCH4NFDI consortium.

dCache [16] is popular storage implementation to host and archive scientific data sets. A major fraction of all WLCG data is stored on dCache storage and for example the LOFAR Long Term Archive makes also use of dCache. dCache provides a wide range of access protocols, among others, in this context less relevant protocols, it implements an NFS 4.1 server, it provides WebDAV / HTTPS and the XrootD protocol. Also several methods for authentication and authorization are available. In the recent 8.2 "golden release" series a generic "oidc" module is provided to support WLCG SciTokens and tokens provided by e.g. the Helmholtz-AAI, following the WLCG plans for token based authentication [17].

For a prototype setup for PUNCH4NFDI an existing integration system at DESY has been used. This system serves also as a test system for WLCG and is further developed as a flexible storage for a wide range of scientific communities going beyond the domains of particle physics and astronomy. Satisfying the demands of the various communities has turned out to be rather challenging also due to the rapid developments in the context of token based authentication.

In addition to the dCache instance, XRootD [18] instances are set up both at the University of Bonn and GSI. XRootD itself is a flexible, modular storage framework. With limited configuration effort, it can be used to offer data via standard protocols (WebDAV / HTTPS) and its native XRootD protocol, which are both known to scale well and be interoperable with other storage solutions, which means they are a good match for the PUNCH requirements. In the context of Storage4PUNCH, authorization via the token-based PUNCH AAI is used. XRootD relies on a plugin called `xrootd-scitokens` which is part of the XRootD software distribution for token handling, which in turn uses the `scitokens-cpp` library [19]. In the context of the PUNCH project, the `scitokens-cpp` library was successfully extended to support the `at+jwt` tokens used by the Unity IdM, the underlying Identity Management system of the PUNCH AAI [20]. Furthermore, an extension of the `xrootd-scitokens` plugin is planned to extract group information from the PUNCH AAI tokens once they become available, which allows to use these for XRootD's authorization framework.

Noticeable advantages of XRootD in the context of supporting smaller communities are its large re-usability (additional authentication mechanisms, file distribution or file caching can easily be added to the configuration) and full local control by the data owners about the actual authorizations granted to the groups defined within the AAI.

There are efforts ongoing in PUNCH4NFDI to develop a generic metadata catalog, which should be used as initial application as a new catalog for the ILDG [21], the international lattice data grid and replace the existing catalog service. A main feature of the new catalog is the possibility to load a schema such that the service can be used also to describe metadata for

other science communities or to serve as a generic file and replica catalog. The later is going to be evaluated in PUNCH4NFDI for smaller research groups that typically have no resources to run or develop their own service for this. An obvious service for data management is also Rucio [22], which is widely used in the HEP community. PUNCH4NFDI foresees to evaluate Rucio for smaller groups. Another interesting option for PUNCH4NFDI is a storage federation, that could follow the well-established the AAA federation (Any data Any time Anywhere) [23] from the CMS experiment. In this setup the user just needs a logical filename, which is used to query a global or regional entry point. The service itself finds a storage source to which the request would be re-directed for the actual file access.

4 Integration of Federated Compute and Storage Infrastructure

The chosen technology stack of Storage4PUNCH comes at the price that it is not POSIX accessible anymore. This leads to two possible modes to access data on Storage4PUNCH depicted in fig. 2.

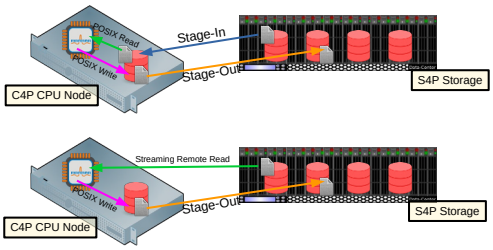


Figure 2: Two possible modes to access data on Storage4PUNCH when running on Compute4PUNCH. Upper drawing: Stage-in mode, lower drawing: Streaming mode.

The first mode is the so-called stage-in mode. It requires the entire data to be staged from Storage4PUNCH to a local POSIX-compliant storage before it can be accessed. This modus operandi is usually the most inefficient one because first, the CPU has to wait until the entire data has been transferred before starting the actual data processing, and second because the smallest entity to stage is a complete file even if only a part of it is actually needed by the job. However, for applications that strictly require POSIX access, it is usually the only possibility to access data stored on Storage4PUNCH.

The second mode is the so-called streaming mode. It is the most efficient and preferred method to access data on Storage4PUNCH, because the CPU can immediately start the actual data processing while streaming the data from Storage4PUNCH. Furthermore, only the parts of the file needed for data processing are transferred. However, this mode strictly requires your application and file format to support streaming mode, which is unfortunately not always the case.

Since the Helmholtz AAI access tokens that are mandatory to access Storage4PUNCH from Compute4PUNCH worker nodes have only a limited lifetime of about one hour, which is usually shorter than the average job run-time, a mechanism to on-the-fly refresh access tokens on the actual worker node running the job needs to be introduced. The so-called HTCondor Credential daemon - Credd - provides advanced features to refresh access tokens on the worker nodes via the use of so-called Credential Monitoring - CredMon - plugins running on the submit node. The Credd daemon is agnostic to the type of tokens and only used for their distribution. The CredMon plugin is specific to the type of tokens and responsible

for their manipulation and refreshment. A CredMon plugin has therefore been developed to manipulate and refresh access tokens of the Helmholtz AAI. This plugin makes use of the Mytoken web service developed to provide OIDC access tokens to long-running compute jobs in an easy and secure way [24]. The Mytoken service provides so-called mytokens to the user. These are similar to OIDC refresh tokens, which they supersede by enabling the enforcement of additional restrictions on the OIDC access tokens they provide, e.g. regarding their lifetime, provided scopes, geolocation or audience claims.

The refreshment workflow is shown in fig.3. During the job submission, a mytoken is requested by using an HTCondor-compliant script based on the Mytoken libraries. This script enables the user to obtain a mytoken by authenticating once to the Mytoken web service via an OIDC flow with the PUNCH AAI. The mytoken obtained is encrypted, securely stored on the submit node and used to create the first access token, which is also stored on the submit node and made available to the HTCondor compute job. The mytoken is latter used to refresh the OIDC access tokens embedded in the HTCondor compute jobs without being exposed to the world. This refreshment procedure is transparent to the users. The developed CredMon plugin - MyTokenCredMon - regularly checks the validity of the OIDC access tokens. When necessary, it refreshes them via the Mytoken service and transfers them to the HTCondor Credd on the submit node. The HTCondor Credd distributes the newly created access tokens to all HTCondor Starters that currently execute a job of a particular user, ensuring long-term access of the job to Storage4PUNCH. When a compute job is finalized, a file transfer service available in HTCondor ensures the transfer of the job output to the Storage4PUNCH resources, using the Helmholtz AAI access token embedded in the job to authenticate to the storage element.

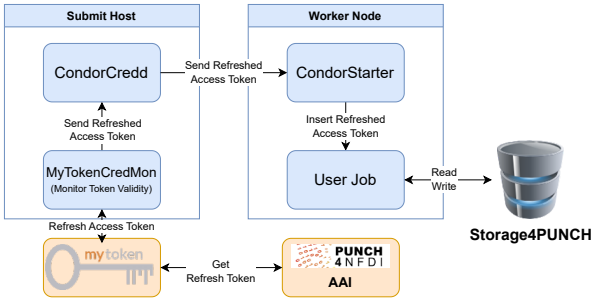


Figure 3: Workflow to transparently refresh access tokens before their expiration on the Compute4PUNCH worker nodes.

5 Workflows utilizing Compute4PUNCH and Storage4PUNCH

The PUNCH4NFDI consortium collected during the phase of proposal writing nearly 100 different use cases in a bottom-up approach from all participating communities to allow for a definition of community overarching tasks and deliverables. Two of those use cases have been selected to serve as demonstration workflows for the developed infrastructure.

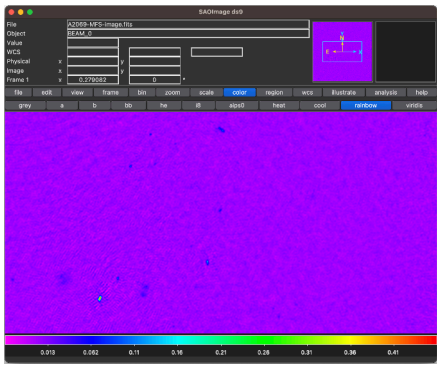
5.1 The LOFAR Radio Imaging Workflow

The LOw Frequency ARray (LOFAR) is a pan-European interferometer operating at very low radio-frequencies between 10 to 240 MHz. It is mainly built and operated by ASTRON,

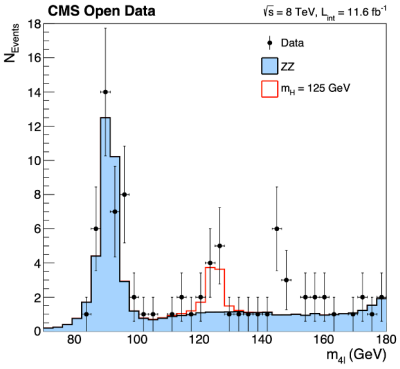
the Netherlands Institute for Radio Astronomy. Due to its wide bandwidth and its high time and frequency resolution, LOFAR enables astronomers to reconstruct the radio sky at low frequencies and with unprecedented detail and fidelity. Its main challenges comprise of the sheer amount of observational data (correlated data size of about 1-2 TB per hour), the efficient and automized mitigation of radio-frequency interference (RFI) and the proper calibration of phase distortions caused by a time-variable and direction-dependent ionosphere. The selected workflow performs a sky brightness reconstruction algorithm so-called "imaging" for a typical radio-interferometric data set recorded by LOFAR on Compute4PUNCH and Storage4PUNCH (see fig. 4a).

5.2 The CERN Open Data Workflow

For the first time in the history of particle physics the Compact Muon Solenoid (CMS) experiment made preceding research data available to the public in 2014 [25]. Further Large Hadron Collider (LHC) experiment followed this approach and nowadays research data of all four LHC experiments are publicly available in the CERN Open Data Portal [26]. The public availability of both, data taken at CMS detector as well as simplified versions of selected CMS analysis codes makes CERN Open Data perfectly suited for educational and demonstration purposes. By providing step-by-step instructions any member of the PUNCH4NFDI sciences can nowadays perform a simplified version of the CMS Higgs in four leptons ($H \rightarrow ZZ \rightarrow 4l$) analysis using data taken back in 2012 on Compute4PUNCH (see fig. 4b). The necessary software packages are provided via CVMFS and the around 13 GBs of data is directly streamed from the EOS file system [27] at CERN hosting CERN Open Data.



(a) Reconstructed LOFAR Radio Image



(b) Invariant four lepton mass spectrum.

Figure 4: Final results of demonstration workflows performed on Compute4PUNCH and Storage4PUNCH.

6 Summary and Outlook

In this contribution, we have introduced the concept of federated heterogeneous compute and storage infrastructures for the PUNCH4NFDI consortium, their interaction as well as selected workflows of diverse PUNCH4NFDI sciences. A demonstrator of the Compute4PUNCH infrastructure is available for testing including – at the time of writing – three integrated computing resources, one of each kind HPC, HTC, and Cloud. This prototype will be further

extended by integrating additional compute resources and additional entry points (JupyterHub and Grid CEs) in the future. Furthermore, it is planned to deploy unmanaged XRootD based on-the-fly disk caches on locally available distributed file systems if applicable and to introduce a data-locality-aware scheduling layer to the OBS in order to send jobs where the data is located. In addition, a demonstrator of the Storage4PUNCH infrastructure is available based on both dCache and XRootD technology. At the moment the Storage4PUNCH prototype comprises three storage systems, one dCache and two XRootD-based. This prototype will also be further extended by integrating additional storage resources. In addition, potential storage federation as well as data and meta-data management options are currently being evaluated. Access to both Compute4PUNCH and Storage4PUNCH is relying on an OIDC access token provided by the Helmholtz AAI, while an automated access token refresh workflow based on the mytoken service and HTCondor's Credd and CredMon ensures the validity of access tokens in long-running jobs. To demonstrate the capabilities of the infrastructure two demonstration workflows of different PUNCH4NFDI sciences have been introduced, one from astronomy using LOFAR data and one from HEP using CMS Open Data.

7 Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 460248186 (PUNCH4NFDI).

References

[1] C. Schneide, T. Schoerner, First results from the PUNCH4NFDI Consortium, Available in this proceedings (2024)

[2] Böhler, Michael, Caspart, René, Fischer, Max, Freyermuth, Oliver, Giffels, Manuel, Kroboth, Stefan, Kuehn, Eileen, Schnepf, Matthias, von Cube, Florian, Wienemann, Peter, EPJ Web of Conferences **251**, 02039 (2021)

[3] HTCondor Team, *HTCondor* (2023), `https://doi.org/10.5281/zenodo.8230603`

[4] M. Fischer, E. Kuehn, M. Giffels, M. Schnepf, S. Kroboth, T. M., O. Freyermuth, *Matterminers/cobald: v0.14.0* (2023), `https://doi.org/10.5281/zenodo.8199049`

[5] M. Fischer, E. Kuehn, M. Giffels, M.J. Schnepf, A. Petzold, A. Heiss, EPJ Web of Conferences **245**, 07040 (2020)

[6] M. Giffels, M. Fischer, A. Haas, S. Kroboth, M. Schnepf, E. Kuehn, PSchuhmacher, R. Caspart, F. von Cube, D. Sammel et al., *Matterminers/tardis: 0.7.1* (2023), `https://doi.org/10.5281/zenodo.7943895`

[7] M. Fischer, M. Giffels, A. Heiss, E. Kuehn, M. Schnepf, R.F. von Cube, A. Petzold, G. Quast, EPJ Web of Conferences **245**, 07038 (2020)

[8] *The Helmholtz Authentication and Authorisation Infrastructure*, `https://hifis.net/aai/`, accessed on 2023-08-15

[9] *Unity Authentication and identity management*, `https://unity-idm.eu/`, accessed on 2023-08-15

[10] *The AARC Blueprint Architecture (BPA)*, `https://aarc-project.eu/architecture/`, accessed on 2023-08-15

[11] *SSH directly with OIDC Access Tokens*, `https://wiki.geant.org/display/AARC/motley_cue%3A+SSH+directly+with+OIDC+Access+Tokens`, accessed on 2023-08-16

[12] O. Freyermuth, K. Kohl, P. Wienemann, Comput. Softw. Big Sci. **5**, 24 (2021)

[13] *Apptainer Container System*, https://apptainer.org/, accessed on 2023-08-16
[14] *CVMFS*, https://cernvm.cern.ch/fs/, accessed on 2023-08-14
[15] S. Mosciatti, J. Blomer, G. Ganis, R. Popescu, Journal of Physics: Conference Series **1525**, 012058 (2020)
[16] Mkrtchyan, Tigran, Chitrapu, Krishnaveni, Garonne, Vincent, Litvintsev, Dmitry, Meyer, Svenja, Millar, Paul, Morschel, Lea, Rossi, Albert, Sahakyan, Marina, EPJ Web Conf. **251**, 02010 (2021)
[17] Bockelman, Brian, Ceccanti, Andrea, Collier, Ian, Cornwall, Linda, Dack, Thomas, Guenther, Jaroslav, Lassnig, Mario, Litmaath, Maarten, Millar, Paul, Sallé, Mischa et al., EPJ Web Conf. **245**, 03001 (2020)
[18] F. Furano, A. Hanushevsky, Tech. Rep. CERN-IT-Note-2009-003, CERN, Geneva (2009), https://cds.cern.ch/record/1177151
[19] *A C++ implementation of the SciTokens library with a C library interface*, https://github.com/scitokens/scitokens-cpp, accessed on 2023-08-14
[20] *scitokens-cpp, issue #53: Compatibility with Unity IAM*, https://github.com/scitokens/scitokens-cpp/issues/53, accessed on 2023-08-14
[21] M. Beckett et al., Comput. Phys. Commun. **140**, 1208 (2011), 0910.1692
[22] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov, M. Elsing et al., Computing and Software for Big Science **3**, 11 (2019)
[23] K. Bloom, the CMS Collaboration, Journal of Physics: Conference Series **513**, 042005 (2014)
[24] *mytoken Documentation*, https://mytoken-docs.data.kit.edu/, accessed on 2023-08-16
[25] *CMS Open Data*, https://cms-opendata-guide.web.cern.ch/cmsOpenData/cmsopendata/, accessed on 2023-08-14
[26] *CERN Open Data Portal*, https://opendata.cern.ch/, accessed on 2023-08-14
[27] A. Peters, E. Sindrilaru, G. Adde, Journal of Physics: Conference Series **664**, 042042 (2015)