https://doi.org/10.1051/epjconf/202533701264

Latest Developments of the PUNCH4NFDI Compute and Storage Infrastructures

Benoit Roland[1,*], *Manuel* Giffels[1], *Arman* Khalatyan[2], *Elena* Sacchi[2], *Harry* Enke[2], *Michael* Huebner[3], *Oliver* Freyermuth[3], *Christoph* Wissing[4], *Baida* Achkar[5], *Alexander* Drabent[6], *Matthias* Hoeft[6], and *Prateek* Gupta[6]

[1]Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany
[2]Leibniz-Institut für Astrophysik Potsdam (AIP), Potsdam, Germany
[3]Universität Bonn, Bonn, Germany
[4]Deutsches Elektronen-Synchrotron (DESY), Hamburg, Germany
[5]Georg-August-Universität Göttingen, Göttingen, Germany
[6]Thüringer Landessternwarte (TLS), Tautenburg, Germany

Abstract. The PUNCH4NFDI consortium, funded by the German Research Foundation for an initial period of five years, gathers various physics communities - particle, astro-, astroparticle, hadron and nuclear physics - from different institutions embedded in the National Research Data Infrastructure initiative. The goal of PUNCH4NFDI is the establishment of FAIR data management solutions for the participating communities. The federated compute and storage infrastructures made available to the consortium, Compute4PUNCH and Storage4PUNCH, comprise a variety of heterogeneous compute and storage systems. The compute resources are managed by an overlay batch system and COBalD/TARDIS meta-schedulers. The TARDIS resource manager is responsible for the provisioning of resources and their integration in the overlay batch system based on HTCondor, while the COBalD resource balancer optimises the resource utilization by matching the actual demand for a given type of resources. The access to the resources is standardised using a token-based authentication and authorization infrastructure. The refreshment of short-lived access tokens is automated using the HTCondor Credential Manager and the MyToken service. Login nodes define single entry points to the federation, while the use of containers and the CERN Virtual Machine File System ensures a scalable provisioning of virtualized software environments. The latest developments are presented, including the access tokens management and the integration of Compute4PUNCH as a compute backend of the REANA analysis platform.

1 Introduction

The PUNCH4NFDI consortium (Particles, Universe, NuClei and Hadrons for the NFDI) is funded by the German Research Foundation and embedded in the National Research Data Infrastructure initiative NFDI (Nationale Forschungsdateninfrastruktur). PUNCH4NFDI gathers various physics communities - particle, astro-, astroparticle, hadron and nuclear physics - from universities and research institutes in Germany that are facing similar data analysis

[*] e-mail: benoit.roland@kit.edu

challenges generated by the increasing amount of data. To better deal with the increasing need for compute and storage, these communities have decided to pool their expertise and resources within the PUNCH4NFDI consortium. This aims at providing a seamless, federated and standardised access to the resources supplied by the participating institutions by setting up a federated and FAIR science data platform enabling cross-disciplinary data analyses. The completion of this infrastructure is seeking to cover the variety of computing needs of the different communities while minimizing the potential modifications to be adopted by the resource providers.

2 Federation of the compute resources - Compute4PUNCH

The HTC, HPC and Cloud compute resources provided by the PUNCH4NFDI institutions are highly heterogeneous and characterised by a variety of architectures, operating systems and software environments, authentication methods and batch systems. The supplied resources are furthermore already in operation and utilised by different communities. Their integration in the PUNCH4NFDI compute infrastructure, Compute4PUNCH, should therefore interfere as little as possible with their operational status. This goal is achieved by relying on technologies recently developed for the federated compute infrastructure of the German High Energy Physics (HEP) community [1, 2] and by adapting them to the specific needs of PUNCH4NFDI. The key ingredients of Compute4PUNCH are the aggregation of the compute resources in a single overlay batch system (OBS) and the provisioning of these resources by a meta-scheduler.

The HTCondor Software Suite [3], a batch system specialised for compute-intensive jobs and designed to allow for a dynamic extension of its pool, has been chosen to aggregate the resources in a single OBS. The integration of a resource is achieved by executing placeholder jobs, known as pilots or drones, on that resource. Each drone is running an instance of the HTCondor StartD daemon which presents the resource to the federated pool as a machine capable of running jobs, also known as execution point or worker node. The StartD daemon also advertises some of the worker node attributes to the pool, allowing for a late binding of the jobs to the most appropriate resource. The drones rely on the Apptainer [4] container technology natively supported by HTCondor to provide the worker node environment suited for the PUNCH4NFDI applications. At the time of a resource integration, the configuration of the HTCondor daemons is pulled from a central GitLab repository to the drones using a dedicated HTCondor tool [5] that allows to dynamically configure an HTCondor node from a Git repository. The HTCondor central manager defines the extent of the federated pool and centralises the information about its state. Worker nodes made available to Compute4PUNCH authenticate to the HTCondor central manager via the use of IDTOKENS provided on request to the resource provider at the time of the first integration of their resource in the federated infrastructure.

The COBalD/TARDIS meta-scheduler [6–9] is used to provision resources according to the actual need. COBalD/TARDIS decides to either extend or reduce the number of integrated resources displaying specific attributes based on their current utilization. An instance of the COBalD/TARDIS meta-scheduler is running for each resource provider and acting as an interface between the local batch system or cloud API and the overlay batch system. The Opportunistic Balancing Daemon COBalD optimises the resource utilization by monitoring and matching the actual demand for a given type of resources, while the TARDIS resource manager is responsible for the provisioning of resources and their integration in the OBS. To achieve that task, the TARDIS plugin provides a collection of site adapters that allow the management of resources through the means of the most used batch systems.

Compute resources are provided to PUNCH4NFDI by the Karlsruhe Institute of Technology (KIT), the Universität Münster and the Georg-August-Universität (GAU) in Göttingen. At KIT, two compute clusters have been integrated using a dedicated fairshare between the different users. The first one is the local university cluster TOpAS (Throughput Optimised Analysis System) [10] that provides 8 NVidia V100 GPUs to PUNCH4NFDI - with 8 CPU cores and 20 GB of RAM per GPU drone. The second one is the GridKa cluster [11] that provides up to 2000 cores to PUNCH4NFDI - with 2-3 GB of RAM per core. In Münster, compute resources from the Uni Cloud Münster [12] have been integrated in Compute4PUNCH via the OpenStack cloud computing platform. In Göttingen, 160 CPU cores from the Tier-3 component of the Computing Center GoeGrid [13] are provided to Compute4PUNCH by the Georg-August-Universität.

3 Provisioning of entry points

The dynamic integration of compute resources presented in Sect. 2 was dealing with the community of resource providers. This section is dealing with the community of resource users to which the compute and storage infrastructures should be made available transparently via the provisioning of entry points. As of today, a single login node running the RHEL8 operating system and hosted at KIT serves as entry point to PUNCH4NFDI. This login node, also known as submit node or access point in the HTCondor framework, runs an instance of the HTCondor Schedd daemon. When compute jobs are submitted, the Schedd daemon presents the request for resources to the federated pool and stores and manages the jobs in the queue. As in the case of a worker node, a login node authenticates to the HTCondor central manager via the use of IDTOKENS provided on request to the login node provider at the time of its first integration in the infrastructure.

The access to the future login nodes is standardised and relies on the use of tokens. This will allow the users to connect to the future entry points using a single authentication mechanism already in use to access the existing login node at KIT. This mechanism is based on the Open ID Connect (OIDC) identity and authentication protocol [14] and is using the Helmholtz AAI [15] as the identity and authorisation management (IAM) system to arbitrate the access to the login node. The Helmholtz AAI is based on the Unity IdM [16] identity and authentication service and follows the AARC Blueprint architecture [17]. The authentication mechanism based on OIDC tokens is described below.

On the client side, users register to the Helmholtz AAI provider and install the oidc-agent [18], a set of tools to handle OIDC tokens from the command line in a similar way as SSH keys. After configuring the oidc-agent using their account at the Helmholtz AAI provider, the users can access the login node with the mccli client [19], a wrapper around the SSH client that enables to establish a SSH connection using OIDC access tokens. Users will be able to connect to the future entry points via the same mechanism relying on the OIDC access tokens provided by the Helmholtz AAI and the use of the mccli client.

On the server side, the login node is running the motley cue [20] service that maps federated OIDC identities to local identities on the node, and the pam-ssh-oidc [21] module, a pluggable authentication module (PAM) plugged to the SSH daemon that accepts OIDC access tokens for the authentication of users. The motley cue service also comprises a local user management (LUM) module that acts as an interface between motley cue and the local user management system and handles the lifecycle of the local user accounts mapped to the federated OIDC identities. The motley cue LUM module offers several backend plugins for the integration with various local user management systems. The login node is configured to use a dedicated LDAP [22] server to store the information about the local users and the motley cue LUM interface is therefore configured to use the LDAP backend. The configuration of the login node is managed and deployed using the configuration management tool Puppet [23] and the system management software Red Hat Satellite [24].

4 Provisioning of software environments

The provisioning of specific operating systems and software environments covering the needs of the PUNCH4NFDI communities relies on the Apptainer [4] container technology and the use of the CERN Virtual Machine File System CVMFS [25]. The Docker files containing the container build instructions defined by the users to execute their analysis workflow on the Compute4PUNCH infrastructure are stored in a common GitLab repository, the Container Stack repository. A Continuous Integration/Continuous Development (CI/CD) pipeline checks the validity of the build instructions and integrates security updates when required before building the containers and uploading them to a common GitLab Container Registry. A transparent and scalable provisioning of the containers is ensured by converting them into the Apptainer unpacked format before making them available to the compute infrastructure - with a latency of the order of a few hours - using the CERN CVMFS repository unpacked.cern.ch. Depending on the resource provider, the CVMFS file system is either directly installed on the bare-metal or mounted using a mount namespace. When submitting a compute job to Compute4PUNCH through the HTCondor batch system, the users only need to specify in the job configuration file the name of the specific container to be retrieved from CVMFS to run their analysis. Workflow examples and tutorials are provided for the users to familiarise with the provisioning described above.

Different workflows have been tested on the Compute4PUNCH infrastructure. Among others, the calibration of the Two-meter Sky Survey data from the LOFAR (LOw Frequency ARray) radio telescope [26] and the processing of cosmological model simulations have been performed by the community from the Thüringer Landessternwarte (TLS), while the reduction of interferometric data from the MeerKAT radio telescope [27] has been processed by the community from the Universitätssternwarte München (USM). Running these workflows has shed some light on differences between the data access and processing in HEP and astrophysics analysis and Compute4PUNCH has been able to adapt to serve the needs of both communities. The way the LOFAR data are processed is at the intersection between High-Throughput and High-Performance computing and powerful multicore single nodes with a scratch space as large as 5 to 10 TB are needed to performe the calibration. The LOFAR data are only available on tape. Their transfer to the Storage4PUNCH dCache instance at KIT and their provisioning to the worker nodes using a NFS distributed file system have been implemented to handle this specific data access mode.

5 Access tokens management

The access to the federated storage resources Storage4PUNCH (see Sect. 6) is standardised by using a token-based AAI that provides a single authentication and authorisation mechanism for the collection of storage resources. As for the login nodes, this mechanism is based on the OIDC protocol and is using the Helmholtz AAI provider as the identity and authorisation management system to arbitrate the access to the storage resources. For the purpose of preventing or at least limiting malicious use of the PUNCH4NFDI resources, OIDC access tokens obtained from the Helmholtz AAI provider have a limited lifetime of just over one hour. This value being shorter than the average job run-time, a mechanism had to be introduced within the HTCondor framework to refresh access tokens on the fly on the actual worker nodes running the jobs. The HTCondor Credential daemon Credd provides advanced features to refresh access tokens on the worker nodes via the use of Credential Monitoring (CredMon) plugins running on the submit node. The Credd daemon is agnostic to the type of token and only used for their distribution. The CredMon plugins are specific to the type of token and responsible for their manipulation and refreshment. A specific CredMon plugin had therefore to be developed to manipulate and refresh OIDC access tokens provided by the Helmholtz AAI. This plugin makes use of the Mytoken service [28] that aims at provisioning long-running compute jobs with OIDC access tokens in an easy and secure way. The Mytoken service provides credential artifacts known as Mytokens that allow a client application, the Helmholtz AAI provider, to provision new access tokens without resorting to the users. The Mytoken credentials are like OIDC refresh tokens which they supersede by enabling the enforcement of additional restrictions on the provided access tokens, among others regarding their lifetime, provided scopes, geolocation or audience claims.

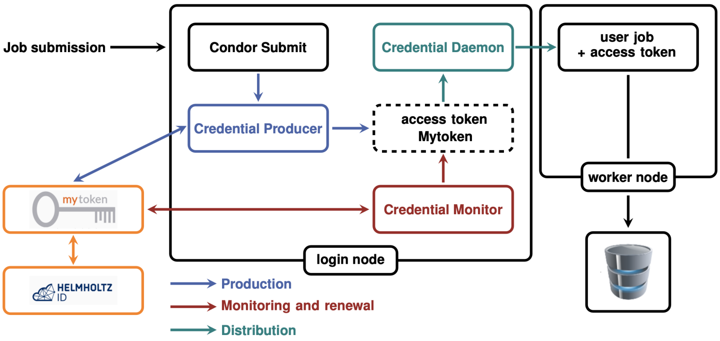

Figure 1. Implementation of the access token management in the HTCondor Software Suite.

The production and refreshment workflows are illustrated in figure 1. A plugin, known as the MytokenProducer [29] and based on the Mytoken libraries, is invoked during the HTCondor job submission to prompt the user to create a Mytoken credential artifact. This plugin enables the user to obtain a Mytoken by authenticating once to the Mytoken web service via a Helmholtz AAI OIDC flow. The obtained Mytoken is encrypted, securely stored on the submit node and used to create the first access token. This access token is stored on the submit node as well and made available to the specific worker node where the job is running. The Mytoken object residing only on the submit node is not exposed to the world. It can therefore be used during a relatively long period of time, typically of the order of one week, to refresh the OIDC access tokens embedded in the HTCondor compute jobs on the worker nodes. A plugin, known as the MyTokenCredMon [29], is periodically invoked by the HTCondor software to check the validity of the OIDC access tokens. When these are about to expire, the plugin replaces them by new access tokens via a Helmholtz AAI OIDC flow arbitrated by the Mytoken service. The new credentials are saved on the submit node and made available to the HTCondor Credd daemon that distributes them to all HTCondor Starters currently executing jobs of a particular user. This procedure ensures a long-term access to the Storage4PUNCH resources for the jobs running on the Compute4PUNCH infrastructure in a way transparent to the users. When a compute job is finalised, a file transfer service available in HTCondor is used to transfer the job output to the Storage4PUNCH resources using the Helmholtz AAI access token embedded in the job to authenticate to the storage element. The MytokenProducer can also be used - to create a Mytoken credential artifact together with the first access token - outside of the HTCondor software suite to allow for an automated submission of a collection of compute jobs afterwards. The HTCondor release [29] implementing the renewal of OIDC access tokens using the Mytoken service is also deployed by other scientific communities outside the PUNCH4NFDI consortium. It is used by the DARWIN (Dark Matter WIMP Search With Liquid Xenon) Collaboration [30, 31] and its deployment at the NIKHEF Computer Technology department is work in progress.

6 Federation of the storage resources - Storage4PUNCH

The federated storage infrastructure made available to PUNCH4NFDI, Storage4PUNCH, aims at a long term storage of data produced by the consortium. As mentioned in Sect. 5, the access to Storage4PUNCH relies on a single token-based authentication mechanism using the OIDC protocol and the Helmholtz AAI provider as the identity and authorisation management system. Dedicated network protocols have therefore to be used to access data on Storage4PUNCH. The experience of the German HEP community with the WebDAV and XRootD protocols used by the Worldwide LHC Computing Grid (WLCG) has led to the choice of these two protocols to access data on Storage4PUNCH. For the same reason, the distributed storage systems dCache [32] and XRootD [33] deployed in the WLCG have been chosen to handle the data on the heterogeneous storage resources made available to PUNCH4NFDI. The Storage4PUNCH infrastructure presently comprises four endpoints, two endpoints based on the dCache storage framework at DESY and KIT and two endpoints based on the XRootD storage framework at Bonn and GSI. The dCache and XRootD storage systems both support the WebDAV and XRootD access protocols and the authentication mechanism based on the OIDC access tokens provided by the Helmholtz AAI. The implementation of the token-based authentication mechanism in dCache follows the WLCG plans for token-based authentication [34]. The implementation in XRootD relies on the xrootd-scitokens plugin and the scitokens-cpp library [35] developed to handle SciTokens and extended to support the access tokens provided by the Unity IdM on which the Helmholtz AAI provider is based [36].

7 REANA job controller

One of the key elements of the PUNCH4NFDI federated and FAIR science data platform is the PUNCH4NFDI Data Portal. This portal will define the single entry point to the research products to be delivered by the consortium, a research product being defined as the complete collection of information related to a given analysis. A typical research product encompasses data, simulation, code, executable, workflow description, metadata and publication. A pillar of the Data Portal is the use of the Reproducible research data analysis platform REANA [37]. This analysis platform developed at CERN provides advanced features particularly suited for the development of a federated and FAIR science data platform. It offers a wrapper around computational workflow engines and a scalable model to use remote compute resources. It uses containers to run the analysis workflows, ensuring in that way interoperability and reusability, and presents a user-friendly graphical interface offering access to a Jupyter notebook and to the collection of information required to build a research product.

The REANA platform provides several job controllers that are responsible for the execution and the management of the jobs on the remote compute resources. Job controllers are available for the CERN flavours of the HTCondor and Slurm [38] batch systems and for the Kubernetes [39] containerised workflow management system. The job controller for the HTCondor batch system available in REANA does not meet the PUNCH4NFDI requirements related to the authentication and authorization infrastructure, for its use of the ticket-based Kerberos authentication protocol instead of the token-based OIDC protocol used by the PUNCH4NFDI compute and storage resources. A specific job controller using the Helmholtz AAI OIDC flow has therefore been developed to integrate the PUNCH4NFDI HTCondor software suite into the REANA analysis platform. Two REANA instances are hosted and managed by the Leibniz-Institut für Astrophysik (AIP) in Potsdam, a production instance for the user community and a test instance for the purpose of the integration and future developments. The PUNCH4NFDI-specific job controller [40] that is now part of the official REANA software release has been tested extensively in the submission of REANA jobs to the Compute4PUNCH infrastructure. The OIDC workflow is currently implemented in the following way: before submitting a REANA job, the user runs a dedicated executable that produces and monitors a Mytoken credential artifact, saves an encrypted version locally and uploads to the REANA server an unencrypted version in the form of a REANA secret. The job controller later invokes a second executable that uploads the Mytoken from the REANA server to the login node from which the HTCondor job associated to the REANA workflow is submitted. When the compute job is finalised, the job output is transferred to both the Storage4PUNCH resources and the REANA workspace where a direct access to the analysis results is possible via the REANA graphical user interface and the interactive session it provides.

8 Conclusion and outlook

The federated compute and storage infrastructures made available to the PUNCH4NFDI consortium, Compute4PUNCH and Storage4PUNCH, have been presented with a focus on the latest developments of the infrastructure, the automated access tokens management and the integration of Compute4PUNCH as a compute backend of the REANA analysis platform. Future developments include the improvement of the LOFAR data transfer to the Storage4PUNCH dCache instance at KIT, the evaluation of the Indigo IAM [41] as identity and authorisation management system, the authentication of the job submission to HTCondor using the Indigo IAM access tokens, the usage of the HTCondor Vault Token [42] to store and renew the access tokens and the provisioning of shared user directories to the PUNCH4NFDI communities.

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

References

[1] M. Böhler, R. Caspart, M. Fischer, O. Freyermuth, M. Giffels, S. Kroboth, E. Kuehn, M. Schnepf, F. von Cube, P. Wienemann, Transparent Integration of Opportunistic Resources into the WLCG Compute Infrastructure, EPJ Web of Conferences, **251**, 02039 (2021), https://doi.org/10.1051/epjconf/202125102039.

[2] A. Drabent, O. Freyermuth, M. Giffels, M. Hoeft, J. Künsemöller, B. Roland, D. Schwarz and C. Wissing, Federated Heterogeneous Compute and Storage Infrastructure for the PUNCH4NFDI Consortium, EPJ Web of Conferences, **295**, 07020 (2024), https://doi.org/10.1051/epjconf/202429507020.

[3] HTCondor Team, HTCondor (2024), https://doi.org/10.5281/zenodo.11397217.

[4] Apptainer Container System, https://apptainer.org, accessed on 20.02.2025.

[5] condor-git-config: dynamically configure an HTCondor node from a git repository, https://github.com/MatterMiners/condor-git-config.

[6] 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.

[7] M. Fischer, E. Kuehn, M. Giffels, M.J. Schnepf, A. Petzold, A. Heiss, Lightweight dynamic integration of opportunistic resources, EPJ Web of Conferences, **245**, 07040 (2020), https://doi.org/10.1051/epjconf/202024507040.

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

[9] M. Fischer, M. Giffels, A. Heiss, E. Kuehn, M. Schnepf, R.F. von Cube, A. Petzold, G. Quast, Effective Dynamic Integration and Utilization of Heterogenous Compute Resources, EPJ Web of Conferences, **245**, 07038 (2020), https://doi.org/10.1051/epjconf/202024507038.

[10] R. Caspart, M. Fischer, M. Giffels, R.F. von Cube, C. Heidecker, E. Kuehn, G. Quast, A. Heiss, A. Petzold, Setup and commissioning of a high-throughput analysis cluster, EPJ Web of Conferences, **245**, 07007 (2020), https://doi.org/10.1051/epjconf/202024507007.

[11] GridKa, https://www.scc.kit.edu/en/research/gridka.php, accessed on 20.02.2025.

[12] Uni Cloud Münster, https://cloud.uni-muenster.de, accessed on 20.02.2025.

[13] Jörg Meyer et al., ATLAS Tier-2 at the Compute Resource Center GoeGrid in Göttingen, J.Phys.Conf.Ser. **331** (2011) 072055, https://doi.org/10.1088/1742-6596/331/7/072055.

[14] What is OpenID Connect, https://openid.net/developers/how-connect-works, accessed on 20.02.2025.

[15] Helmholtz Authentication and Authorisation Infrastructure, https://hifis.net/aai, accessed on 20.02.2025.

[16] Unity Authentication and identity management, https://unity-idm.eu, accessed on 20.02.2025.

[17] AARC Blueprint Architecture (BPA), https://aarc-community.org/architecture, accessed on 20.02.2025.

[18] oidc-agent, https://indigo-dc.gitbook.io/oidc-agent, accessed on 20.02.2025.

[19] mccli, https://mccli.readthedocs.io/en/latest, accessed on 20.02.2025.

[20] motley cue, https://motley-cue.readthedocs.io/en/latest, accessed on 20.02.2025.

[21] ssh-oidc, https://github.com/EOSC-synergy/ssh-oidc, accessed on 20.02.2025.

[22] Lightweight Directory Access Protocol LDAP, https://ldap.com, accessed on 20.02.2025.

[23] Puppet, https://www.puppet.com, accessed on 20.02.2025.

[24] Red Hat Satellite, https://www.redhat.com/en/technologies/management/satellite, accessed on 20.02.2025.

[25] CVMFS, https://cernvm.cern.ch/fs, accessed on 20.02.2025.

[26] LOFAR radio telescope, https://www.astron.nl/telescopes/lofar, accessed on 20.02.2025.

[27] MeerKAT radio telescope, https://www.sarao.ac.za/science/meerkat, accessed on 20.02.2025.

[28] Mytoken Service, https://mytoken.data.kit.edu, accessed on 20.02.2025.

[29] C4P HTCondor CredMon, https://github.com/benoitroland/C4P-HTCondor/tree/credmon-jwt-token, accessed on 20.02.2025.

[30] DARWIN Collaboration, https://darwin.physik.uzh.ch/index.html, accessed on 20.02.2025.

[31] Robin Hofsaess, "A Lightweight Analysis and Grid Facility for the DARWIN Experiment", Proceedings of the CHEP 2024 conference (to be published).

[32] dCache, Distributed Storage for Scientific Data, https://www.dcache.org, accessed on 20.02.2025.

[33] F. Furano, A. Hanushevsky, Tech. Rep. CERN-IT-Note-2009-003, CERN, Geneva (2009), https://cds.cern.ch/record/1177151.

[34] B. Bockelman, A. Ceccanti, I. Collier, L. Cornwall, T. Dack, J. Guenther, M. Lassnig, M. Litmaath, P. Millar, M. Sallé et al., WLCG Authorisation from X.509 to Tokens, EPJ Web of Conferences, **245**, 03001 (2020), https://doi.org/10.1051/epjconf/202024503001.

[35] A C++ implementation of the SciTokens library with a C library interface, https://github.com/scitokens/scitokens-cpp, accessed on 20.02.2025.

[36] scitokens-cpp, issue 53: Compatibility with Unity IAM, https://github.com/scitokens/scitokens-cpp/issues/53, accessed on 20.02.2025.

[37] REANA, https://reanahub.io, accessed on 20.02.2025.

[38] Slurm workload manager, https://slurm.schedmd.com/sbatch.html, accessed on 20.02.2025.

[39] Kubernetes, https://kubernetes.io, accessed on 20.02.2025.

[40] REANA Pull Request 430, Add job controller, job monitor and tools to support Compute4PUNCH backend, https://github.com/reanahub/reana-job-controller/pull/430, accessed on 20.02.2025.

[41] Indigo IAM: Identity And Access Management for Scientific computing, https://indigo-iam.github.io/v/current, accessed on 20.02.2025.

[42] D. Dykstra, M. Altunay, J. Teheran, Secure Command Line Solution for Token-based Authentication, EPJ Web of Conferences **251**, 02036 (2021), https://doi.org/10.1051/epjconf/202125102036.