The Virtual Research Environment: a multi-science analysis platform

Elena Gazzarrini[1,*], *Enrique* Garcia Garcia[1,**], *Domenic* Gosein[2], and *Xavier* Espinal[1]

[1]European Organization for Nuclear Research (CERN), 1211 Meyrin, Switzerland
[2]Mannheim University of Applied Sciences, Mannheim, Germany

Abstract. One of the objectives of the ESCAPE collaboration, within the context of the implementation of the European Open Science Cloud, is integrating analysis workflows from Cosmology, Astrophysics and High Energy Physics into a common framework, making it easier for researchers to share data, software and analysis code. The development of the ESCAPE Science Projects (Dark Matter and Extreme Universe) relies on the implementation of the Virtual Research Environment, a prototype analysis platform deployed in the respect of FAIR data principles. The Virtual Research Environment's main components are (1) an Authentication and Authorization layer, (2) a federated distributed storage solution, providing functionalities for data injection and replication through a Data Management and Orchestration software, (3) an interactive notebook interface that hides the complexity of the underlying infrastructure and facilitates browsing data catalogues and (4) computing processing power to run full end-to-end analyses through workflow schedulers and resource managers that also allow for pipelines' preservation. This contribution will depict the collection of services and technologies that compose the Virtual Research Environment, hosted on a cloud infrastructure and managed by an open source, publicly available repository.

1 Introduction

CERN, located in Geneva (Switzerland), is home to the largest and most powerful accelerator. The construction of the Large Hadron Collider (LHC) led to the discovery of the Higgs boson in 2012 [1], and paved the way for data-intensive physics computing. CERN experiments were the first ones to face the challenges of managing Petabytes (PB) of data per year and, therefore, to develop technologies to store, manage and share data among all users. As CERN collaborations extend across Europe (and globally), the field of High Energy Physics (HEP) has for long been focusing on a plethora of development projects on shared analysis platforms. Non-HEP sciences, like the new generation of sub-particle detectors, telescope arrays and antennas, are however more recently entering the Exabyte-scale era [2, 3], and therefore need the ability to efficiently track and process the generated data. Within this international scenario, the scope of EU-funded H2020 projects such as ESCAPE (European Science Cluster of Astronomy, Astroparticle and Particle Physics) [4, 5] and EOSC-Future

*e-mail: gazzarrini.elena@gmail.com
**e-mail: enrique.garcia.garcia@cern.ch

(European Open Science Cloud) [6] is to corroborate common IT research infrastructures across physics domains to provide, with a bottom-up approach, a sustainable prototype model to be adopted by multiple collaborations, while meeting FAIR (Findable, Accessible, Interoperable and Reusable) principles [7]. Such a model should tackle some of the current challenges in computing like the standardisation and usage of federated data management and distributed storage, data processing and orchestration on distributed computing resources and reproducibility of scientific results.

To address the aforementioned challenges, the Virtual Research Environment (VRE) analysis platform was initiated and developed at CERN, as an aggregation of services needed to run end-to-end physics analysis workflows. The VRE inherits its concept from the WLCG (World LHC Computing Grid) [8] and from the user-oriented analysis platforms deployed by astrophysical observatories. Its novelty consists in the example workflows from various physics domains that have been successfully run on it, confirming the idea that adopting a common framework across particle physics and astrophysics workflows would promote research collaboration and analysis sharing.

2 Scientific use case

A clear example of the versatility of such platform is provided by the ESCAPE "Dark Matter Science Project" [1] [9] pilots (funded through the H2020 EOSC-Future project), which combine workflows from Particle Physics (CERN), High-energy Astrophysics (CTA [10], FermiLAT [11]), Neutrino Observations (KM3NET [12], Darkside [13]), Radio Astronomy (SKA [14], LOFAR [15]) and Gravitational Waves searches (LIGO [16], Virgo [17]).

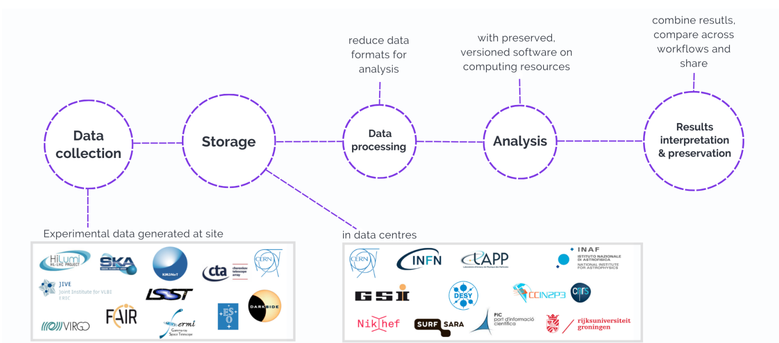

Figure 1. A generalised representation of the steps of a physics analysis workflow. The logos correspond to all the experiments' sites and storage sites that contribute to the VRE development.

All of the above follow a workflow that generally looks like the one in Figure 1 and investigates Dark Matter (DM) mass limits in different ways. Direct Detection methods study the interaction of particles inside underground detectors, Collider physics produces DM candidates from accelerating protons, Astrophysics observes distant phenomena in the sky and compares them with the theory to detect abnormal behaviours, while Indirect Detection methods analyse annihilating DM by looking at its decay products, such as neutrinos. Combining all of these results would allow the development of better, more universal theories.

[1] https://escape2020.pages.in2p3.fr/virtual-environment/home/sp-dark_matter/

3 VRE technical specifications

To aid scientific collaboration, the VRE exposes four core services, represented in Figure 2 and encompassing (1) user management through a reliable Authorisation and Authentication Infrastructure (AAI), (2) a data management framework, (3) exposure to a high-level notebook environment that facilitates the interaction with the underlying infrastructure and (4) access to computing resources managed by various work schedulers. A detailed explanation of these services can be found in the next sections.

CERN's OpenStack virtualised cloud machines constitute the VRE hardware backend; the technical specifications of the cluster can be found in Table 1. The 23 worker nodes are mostly 2x Intel(R) Xeon(R) Silver 4216 2.10GHz CPUs with 192GB DDR4 2933Mhz of RAM, which are connected to CERN's 10Gbit ethernet network, for an overall count of 184 vCPUs and 335.8 GB RAM. The characteristics are indicative, as the cluster creation is serialized, and the scheduler maps resources on specific hypervisors based on the occupation of the cloud.

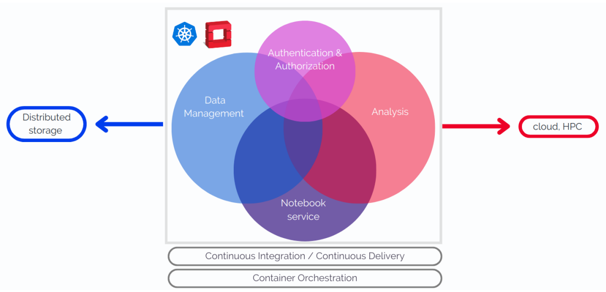

Figure 2. The VRE building blocks, namely a common authentication and authorization layer, a data management framework (connected to storage elements), a notebook service to hide the platform's complexity and computing resources to run analysis (on cloud or HPC nodes).

vCPUs	RAM (GB)	Masters	Nodes	Remote Storage (TB)	CephFS (TB)
184	335.8	3	23	646	1.8

Table 1. The VRE technical components. The first four columns refer to the cloud infrastructure managed with K8s, while the last two columns refer to the total quota of the remote storage elements and the shared object storage (CephFS) attached to the processing nodes.

The current VRE is deployed with the Kubernetes (K8s) template version 1.21; the VRE services, applications and packages are installed on the nodes by exploiting the K8s manifests, which manages the dynamical provisioning of resources to services. The cluster details are embedded in Terraform HCL files, adaptable to meet the specifications of any other cloud platform (i.e. commercial cloud providers). The software is installed via Helm charts, synchronised to the cluster automatically by Flux v2 upon being pushed to the code repository, hosted on a public GitHub organisation [18] that has significantly evolved since the latest publication [19]. The VRE central repositories are

i an onboarding documentation built with GitHub Pages [18], along with a more technical Wiki following each deployment phase,

ii Helm configuration charts for the core services,

iii Docker images comprising the environments displayed on the log-in page of the JupyterHub instance, iv reproducible analyses of each science project [2],

v a service uptime status page [3] informing users of any unexpected service incidentt.

Recent efforts have been focused on achieving a portable, modular and well-documented VRE model that could be re-deployed on any cloud infrastructure. In this way, the repository could serve as a blueprint for IT administrators at scientific institutions that might be interested in replicating the VRE model.

The VRE operations rely on a hybrid combination of industry-standard and CERN-specific technologies, including:

i Sealed Secrets to safely encrypt K8s secrets,

ii nginx proxy server as a load-balancing solution to manage the internet traffic outside the CERN firewall,

iii INDIGO Identity and Access Management (IAM) to manage users,

iv Ceph as an object storage solution,

v distributed disk storage solutions such as EOS [20], StoRM [21], dCache [22], DPM [23] and XRootD [24],

vi CERN's DB On Demand service for database management,

vii CERN Virtual Machine File System (CVMFS [25]) to access experiments' software,

viii CERN's Elasticsearch and Grafana instances to collect and display data transfers.

4 Authorisation and Authentication

The VRE users are authenticated by the IAM service [26], deployed on a K8s cluster at INFN-CNAF in Bologna. Users can authenticate via a username and a password, EduGAIN or X.509 certificates/Virtual Organization Membership Service (VOMS [27]) attribute provisioning services. Following authentication, authorisation is in place to ensure that the user is allowed to perform the requested action, and to delegate access rights to distributed resources. Instead of passing the user's credentials directly to VRE applications, a proxy in the form of a token with set expiration date that acts on behalf of the user is employed as a safer solution. The VRE grants OpenID Connect (OIDC) tokens, which are simple JSON-based identity tokens (JWT) that fit web applications and are delivered via the Open Authorization 2.0 (Auth 2.0) protocol.

The VRE users (the resource owners) are allowed to use the multiple available resources through the IAM service, by authorising a client (an application acting on behalf of the user) to obtain a final access token from the authorisation server of each resource. The addition of the IAM provider as a way to uniquely access all VRE resources, some of which were initially only accessible through CERN's Single Sign On method extends the platform's useability to non-CERN physics communities. The management of users on the platform is automatised via K8s cronjobs that send periodic (daily) JSON based requests to the IAM server to check for new users and update the databases accordingly, giving immediate access to new scientific collaborators.

[2] https://github.com/vre-hub/science-projects

[3] https://vre-hub.github.io/status/

5 Data Management, the Data Lake

The data management and storage orchestration for the VRE prototype is based on the open-source scientific software developed at CERN, Rucio [28], initially developed by the ATLAS [29] experiment for managing community data. It provides services and associated libraries to manage large volumes of data spread across facilities at multiple institutions and organizations.

The VRE Rucio instance is deployed on the aforementioned cloud infrastructure (Table 1) and is upgraded to the latest software version (v1.30). As depicted on the right-hand side of Figure 3, Rucio's servers manage API requests and users' authentication, while Rucio daemons – processes that continually run on the infrastructure – are in charge of data upload, access, download and replication. The servers and daemons are connected to a central PostgreSQL (v14.6) relational database hosted at CERN, providing backup services in case of significant disruptions. While the metadata information is contained in the Rucio database, the raw experimental data is stored in multiple Rucio Storage Elements (RSEs), with quotas varying from 1 to 300 TB, hosted at each partner institution, the location of which can be seen in the middle of Figure 3. The RSEs support several storage technologies (EOS, StoRM, dCache, DPM, XRootD) and use diverse back-ends (classic RAID systems, Ceph, erasure coding and multi-replica). Data can be accessed through gridFTP, HTTP(S), XRootD and S3 protocols. Such policy-driven, reliable, distributed data infrastructure, commonly referred to as the Data Lake [30, 31], is able to deliver data on-demand at low latency to all types of processing facilities.

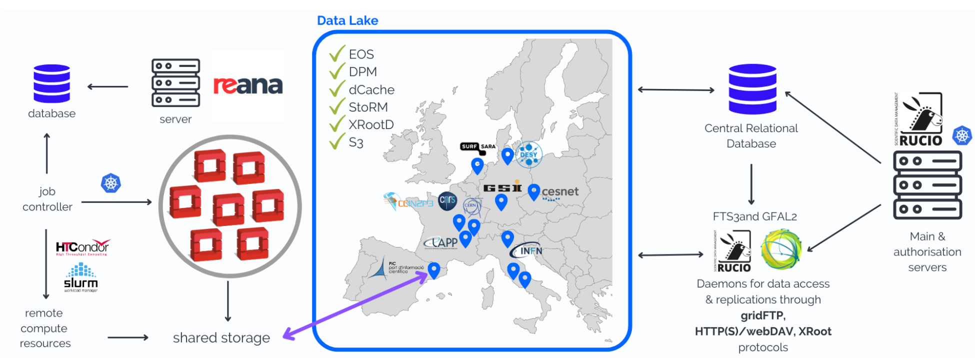

Figure 3. The VRE Cloud infrastructure hosting a Rucio and Reana instance. The first is composed of (i) servers and daemon processes taking care of data transfers, (ii) a central relational database (right) and of (iii) storage elements (RSEs) supporting different technologies (EOS, StoRM, dCache, DPM, XRootD, S3) distributed across Europe, which make up the 'Data Lake' (centre). The latter is composed of (i) a server to manage authentication, (ii) a central relational database, (iii) job controllers to distribute the workflow and (iv) a shared cloud storage to temporarily store the files needed during the analysis.

The main functionalities that Rucio offers are data upload, download, streaming, executed by exploiting the GFAL [32] library, and third-party asynchronous transfers (between RSEs), achieved instead by CERN's File Transfer Service (FTS) [33], which is given permission to access the various RSEs. The Rucio `hermes` daemon collects messages about the transfer status and stores them in an ActiveMQ queue, which sends them to a CERN-hosted Elasticsearch instance. The data gets eventually displayed on Grafana dashboards, an example of which can be found in Figure 4. The Data Lake and the status of the RSEs are monitored by testing the infrastructure with daily uploads and transfers.

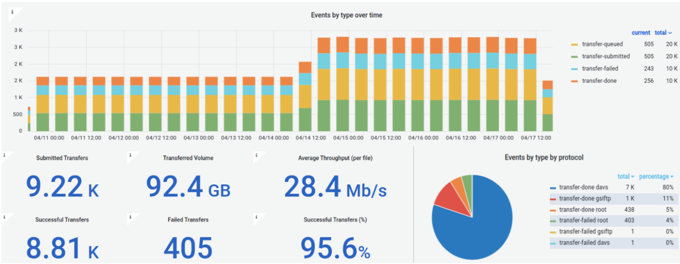

Figure 4. The VRE Rucio continuous monitoring machinery, populated by K8s cronjobs and hosted on CERN's Grafana dashboards.

6 Distributed computing, notebooks and analysis preservation

Once the experimental raw data is safely stored, it needs to be accessed and processed. The advent of multi-core processors [34] has made parallel computing techniques mandatory within large physics analyses. Software has therefore adapted to hybrid parallelism, and workflow schedulers are adopted to efficiently distribute tasks to the underlying hardware and make sure that idle resources are avoided. Once the results are obtained, it is imperative to preserve them for later re-use: literature studies [35] drastically highlight how half of researchers cannot reproduce their own results.

Consequently, the VRE development focuses on:

i finding efficient ways to make the Data Lake catalogue accessible to users,

ii exploiting workflow schedulers to harness the available compute processing power,

iii exploring frameworks to preserve end-to-end analyses.

The platform addresses these challenges in the following ways:

i the VRE exposes a JupyterHub notebook front-end interface [4] enhanced by a Rucio plug-in extension [36], enabling the user to browse the data in the Data Lake and make a copy of it on a CERN RSE of ~0.5 TB. The volume is FUSE mounted on the JupyterHub nodes, bringing the data to the same location of the processing power. Experiment-specific Docker images, pulled from the VRE's GitHub container registry, can be selected at log-in time, sparing the user the need to install the necessary software to analyse data. CVMFS can also be used to load software libraries into the user's JupyterHub session.

ii the VRE employs two main work schedulers to distribute physics analyses in concurrent tasks: CERN's reproducible analysis platform, Reana [37], and the parallel computing library Dask [38]. The VRE Reana cluster [5] exposes a UI interface to track the analysis' progress, it is accessible by the JupyterHub terminal and it is agnostic of the programming language. As can be seen on the left-hand side of Figure 3, the Reana job controller allows to connect to different computing backends to scale out on demand, given the assumption that the user has access rights to remote resources. The default scheduler is K8s, but HTCondor or Slurm can be selected if the goal is maximising data throughput (High Throughput Computing) or computational performance (High Performance Computing) respectively. On the other hand, Dask is installed on JupyterHub along with Dask Gateway, thereby enabling users to dynamically spawn Dask clusters from their JupyterHub session. Dask allows parallelisation of tasks on the cloud infrastructure and display dashboards tracking worker nodes' load.

[4] https://jhub-vre.cern.ch/
[5] https://reana-vre.cern.ch/

iii Reana allows to package a full analysis by exploiting a recent feature [39] that allows connecting to any Rucio instance and pulling the data close to the computing power as the first step of the analysis (exemplified by the purple arrow in Figure 3). It then dispatches the workflow to the computing nodes. The reproducibility of the analysis relies on an easily interpretable `reana.yaml` file (as in Figure 5), where the user specifies the (i) input data and parameters, (ii) code files, (iii) computing environment and (iv) computational steps. In this way, scientists can maintain and compare lists of past runs and share the results with colleagues.

```
1  version: 0.8.1
2  inputs:
3    directories:
4      - python/
5  workflow:
6    type: serial
7    specification:
8      steps:
9        - name: fetchdata-rucio
10         voms_proxy: true
11         rucio: true
12         environment: ghcr.io/vre-hub/vre-rucio-client:v0.1.2-1-0487cc0
13         compute_backend: kubernetes
14         commands:
15           - rucio whoami
16           - rucio get ATLAS_LAPP_SP:DMsummary.dileptonReinterpretation_14TeV.2018
17       - name: SetLimits
18         environment: 'reanahub/reana-env-root6:6.18.04'
19         compute_backend: kubernetes
20         kubernetes_memory_limit: '9Gi'
21         commands:
22           - mkdir plots/
23           - python python/MakeLimit.py
24 outputs:
25   directories:
26    - plots/
```

Figure 5. An example of the Reana declarative language in a typical `reana.yaml` file. The first step is authenticating to Rucio (line 11) and moving the data from the desired RSE to the Reana shared storage represented in Figure 3. The resource manager is in this example K8s (line 13).

Conclusions

In its endeavour to provide physicists with a prototype platform built on common software solutions to manage and process scientific data, the VRE project has successfully demonstrated its value to multiple physics disciplines spanning from High Energy Physics to Astrophysics in the context of pilot use cases from Science Projects in European Commission funded projects. It has gained interest from communities that are currently entering the Exabyte-scale era and are looking for adequate computing solutions. The ESCAPE Open Collaboration Agreement [40] has been signed in January 2023 and has seen new experiments (such as the Einstein Telescope [41]) onboarding the VRE (which currently grants access to almost 300 users) and expressing interest in reproducing it at their home institution. Researchers from various domains used the VRE to share data through the Rucio framework and preserve analyses through the Reana platform. Moreover, the open source and publicly available VRE cluster and its accessibility to communities outside of CERN have made it a valuable prototype to develop combined analyses and a costly testing ground for large-scale physics experiments. The VRE represents a fundamental achievement under both sociological and technological aspects for Landmark European Research Infrastructures facing new data management and computing challenges in the next decade.

Acknowledgements

Authors acknowledge support from the ESCAPE and EOSC-Future projects. ESCAPE has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement 824064. EOSC-Future is co-funded by the European Union Horizon Programme call INFRAEOSC-03-2020, Grant Agreement 101017536.

References

[1] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. Abdel Khalek, A. Abdelalim, O. Abdinov, R. Aben, B. Abi, M. Abolins et al., Physics Letters B **716**, 1 (2012)

[2] M. Rahman, D. Lang, R. Hlozek, J. Bovy, L. Perreault-Levasseur (2019)

[3] R. Bolton, *Global Data Management in Astronomy and the link to HEP: are we collaborators, consumers or competitors*, in *CHEP* (2023)

[4] *Escape: European science cluster of astronomy particle physics esfri research infrastructures*, `https://projectescape.eu/`

[5] G. Lamanna, *The ESCAPE collaboration*, in *CHEP* (2023)

[6] *Eosc future project*, `https://eoscfuture.eu/`

[7] *Fair principles*, `https://www.go-fair.org/fair-principles/`

[8] *Wlcg: Worldwide lhc computing grid*, `https://wlcg.web.cern.ch/`

[9] J. Little, *The ESCAPE Dark Matter Science Project in the European Science Cloud*, in *CHEP* (2023)

[10] M. Actis, G. Agnetta, F. Aharonian, A. Akhperjanian, J. Aleksić, E. Aliu, D. Allan, I. Allekotte, F. Antico, L. Antonelli et al., Experimental Astronomy **32**, 193 (2011), m1 - Journal Article

[11] J.M. Casandjian, *The fermi-lat model of interstellar emission for standard point source analysis* (2015), `https://arxiv.org/abs/1502.07210`

[12] A. Margiotta, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **766**, 83 (2014)

[13] M. Bossa, Journal of Instrumentation **9**, C01034 (2014)

[14] A. Weltman, P. Bull, S. Camera, K. Kelley, H. Padmanabhan, J. Pritchard, A. Raccanelli, S. Riemer-Sørensen, L. Shao, S. Andrianomena et al., Publications of the Astronomical Society of Australia **37** (2020)

[15] M.P. van Haarlem, M.W. Wise, A.W. Gunst, G. Heald, J.P. McKean, J.W.T. Hessels, A.G. de Bruyn, R. Nijboer, J. Swinbank, R. Fallows et al., Astronomy & Astrophysics **556**, A2 (2013)

[16] B.P. Abbott, R. Abbott, R. Adhikari, P. Ajith, B. Allen, G. Allen, R.S. Amin, S.B. Anderson, W.G. Anderson, M.A. Arain et al., Reports on Progress in Physics **72**, 076901 (2009)

[17] F. Acernese, M. Agathos, K. Agatsuma, D. Aisa, N. Allemandou, A. Allocca, J. Amarni, P. Astone, G. Balestri, G. Ballardin et al., Classical and Quantum Gravity **32**, 024001 (2014)

[18] *The virtual research environment documentation*, `https://github.com/vre-hub`

[19] E. Gazzarrini, E. Garcia, D. Gosein, X. Espinal, *The virtual research environment: towards a comprehensive analysis platform* (2023), `2305.10166`

[20] A. Peters, E. Sindrilaru, G. Adde, Journal of Physics: Conference Series **664**, 042042 (2015)

[21] R. Zappi, S.D. Pra, M. Dibenedetto, E. Ronchieri, Journal of Physics: Conference Series **331**, 062047 (2011)

[22] P. Fuhrmann, V. Gülzow, *dCache, Storage System for the Future*, in *Euro-Par 2006 Parallel Processing*, edited by W.E. Nagel, W.V. Walter, W. Lehner (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006), pp. 1106–1113, ISBN 978-3-540-37784-9

[23] A. Álvarez Ayllón, A. Beche, F. Furano, M. Hellmich, O. Keeble, R. Rocha, Journal of Physics Conference Series **396**, 2015 (2012)

[24] *Xrootd*, `https://xrootd.slac.stanford.edu/`

[25] *Cvmfs: Cern virtual machine file system*, `https://cernvm.cern.ch/fs/`

[26] A. Ceccanti, M. Hardt, B. Wegh, A. Millar, M. Caberletti, E. Vianello, S. Licehammer, Journal of Physics: Conference Series **898**, 102016 (2017)

[27] *Voms*, `https://github.com/italiangrid/voms`

[28] 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)

[29] G. Aad et al. (ATLAS), JINST **3**, S08003 (2008)

[30] R. Di Maria, R. Dona, *ESCAPE Data Lake-Next-generation management of cross-discipline Exabyte-scale scientific data*, in *EPJ Web of Conferences* (EDP Sciences, 2021), Vol. 251, p. 02056

[31] R. Bolton, S. Campana, A. Ceccanti, X. Espinal, A. Fkiaras, P. Fuhrmann, Y. Grange, EPJ Web Conf. **245**, 04019 (2020)

[32] *Fgal2*, `https://dmc-docs.web.cern.ch/dmc-docs/gfal2/gfal2.html`

[33] E. Karavakis, A. Manzi, M. Arsuaga Rios, O. Keeble, C. Garcia Cabot, M. Simon, M. Patrascoiu, A. Angelogiannopoulos, EPJ Web Conf. **245**, 04016 (2020)

[34] B. Venu, CoRR **abs/1110.3535** (2011), `1110.3535`

[35] M. Baker, Nature **533** (2016)

[36] *Rucio jupyterlab extension*, `https://github.com/rucio/jupyterlab-extension`

[37] T. Šimko, L. Heinrich, H. Hirvonsalo, D. Kousidis, D. Rodríguez, *REANA: A system for reusable research data analyses*, in *EPJ web of conferences* (EDP Sciences, 2019), Vol. 214, p. 06034

[38] M. Rocklin, *Dask: Parallel computation with blocked algorithms and task scheduling*, in *Proceedings of the 14th python in science conference* (Citeseer, 2015), 130-136

[39] *Reana and rucio*, `https://docs.reana.io/advanced-usage/access-control/rucio/`

[40] *The escape open collaboration agreement*, `https://indico.in2p3.fr/event/27202/contributions/109871/attachments/70182/99379/ESCAPE-Collab-Agree-210422.pdf`

[41] A. Grado, Journal of Physics: Conference Series **2429**, 012041 (2023)