https://doi.org/10.1051/epjconf/202125102060

The ESCAPE Data Lake: The machinery behind testing, monitoring and supporting a unified federated storage infrastructure of the exabyte-scale

Rizart Dona[1,*] and *Riccardo* Di Maria[1,**] on behalf of the ESCAPE project

[1]European Organization for Nuclear Research (CERN), Geneva, Switzerland

Abstract. The EU-funded ESCAPE project aims at enabling a prototype federated storage infrastructure, a Data Lake, that would handle data on the exabyte-scale, address the FAIR data management principles and provide science projects a unified scalable data management solution for accessing and analyzing large volumes of scientific data. In this respect, data transfer and management technologies such as Rucio, FTS and GFAL are employed along with monitoring enabling solutions such as Grafana, Elasticsearch and perfSONAR. This paper presents and describes the technical details behind the machinery of testing and monitoring of the Data Lake – this includes continuous automated functional testing, network monitoring and development of insightful visualizations that reflect the current state of the system. Topics that are also addressed include the integration with the CRIC information system as well as the initial support for token based authentication / authorization by using OpenID Connect. The current architecture of these components is provided and future enhancements are discussed.

1 Introduction

The European Union funded ESCAPE project (European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures) [1] [2] consists of a synergy between ESFRI projects and science organizations [3] which aim at establishing a single collaborative cluster of next generation facilities in the area of astronomy and accelerator-based particle physics in order to implement a functional link between those and EOSC [4]. The project spans over six work packages which reflect its goals. On the technical level these concern the implementation and deployment of a data infrastructure for open science [5] and the creation of a a flexible science platform (ESAP) [6] for the analysis of open access data available through the EOSC environment.

One of the main objectives of the Data Infrastructure for Open Science work package (DIOS) is building a prototype scalable federated data infrastructure, a Data Lake, that would be able to handle data on the exabyte-scale and address the FAIR [7] data management principles, that is, the data would need to be Findable, Accessible, Interoperable and Reusable. Furthermore, this infrastructure would facilitate the access to the scientific data via the ESAP science platform and provide the tools and documentation for such platforms to seamlessly

[*] e-mail: rizart.dona@cern.ch

[**] e-mail: riccardo.di.maria@cern.ch

integrate with it. The biggest challenge of the project is the creation and exposure of a common tool-set for not only HEP but also non-HEP communities who need to manage their data. In that aspect, the ESCAPE Data Lake is highly adaptive and technologically inclusive to the needs of different experiments.

The ESCAPE project has been active since February 2019 and it has successfully delivered a pilot version of the Data Lake which serves the needs and use cases of the participant science organizations and partners. This paper presents an overview of the architecture of this system and focuses on the methodology and technologies that are used in order to test and monitor the data management capabilities of it. More specifically, these topics are covered in the next sections as follows: in section 2 the Data Lake architecture is provided along with the software stack that is used in order to deploy it, in section 3 the testing infrastructure is presented and in section 4 the monitoring capabilities are detailed. Finally, in section 5 some conclusions are drawn and future plans are discussed.

2 Data Lake Architecture

The Data Lake consists of several components that work with each other in order to provide a unified namespace to users that wish to upload, download or access data. On the lowest level one can find the various storage technologies that are deployed and integrated into it, these include storages such as EOS [8] [9], DPM [10], dCache [11], StoRM [12] and XRootD [13]. Currently more than ten deployments are present in different locations and institutes [14].

The data transfer technology stack can be attributed to three software solutions, GFAL, FTS and Rucio. GFAL (Grid File Access Library) [15] acts as a a multi-protocol data management library providing an abstraction layer of the grid storage system complexity. It supports protocols like GridFTP [16], HTTP [17] and Root [13], all three protocols are currently supported in the Data Lake as well through several storages [14]. The FTS (File Transfer Service) [18] [19] acts as the middleware and provides reliable data transfer at a large scale between the storage systems, it enables TPC (Third Party Copy) to transfer data between two storages that support the same protocol by using a direct link between the two. Parallel transfers optimization can also be achieved through heuristics that gather metrics from the network state at the time of the transfers. Rucio [20] [21] acts as the data orchestrator, it is the enabling technology that implements many of the concepts of the Data Lake such as QoS (Quality of Service) [22] [23] and file transitions, distributed redundancy and data policies. It is also the technology that provides a proto-common namespace for the users to interact with the data. One can see the interaction of those three components in Figure 1. Rucio uses GFAL for upload/download operations and FTS in order to perform TPC transfers. FTS employs GFAL in order to perform the actual transfer during a TPC. GFAL finally works on the storage file system level with the supported protocols.

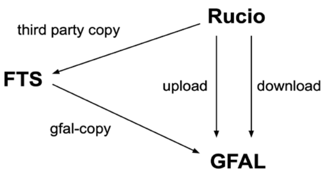


Figure 1: The Data Lake transfer stack

The current deployment of ESCAPE Rucio is based on a Kubernetes [24] cluster that runs on top of Openstack [25], both of those services are provided by the CERN IT Cloud Services [26]. In particular, components of Rucio that run there include the server, the authentication server, the WebUI portal and the various daemons that are responsible for most of the data management logic. Rucio also keeps state in a database which in this case consists of an OracleDB [27] instance provided by the CERN IT Oracle Database Service. The same cluster also hosts all the transfer and access testing code as well as synchronization scripts that integrate different subsystems inside the Data Lake. CRIC (Computing Resource Information Catalogue) [28] serves as the information catalogue and it's where the configuration of the Rucio Storage Elements (RSE) happens, this configuration is also accessible via REST API as a JSON response. The instance is provided and maintained by the CRIC team at CERN. The FTS instance that is used in the Data Lake is provided by the CERN IT FTS Service [29].

The authentication and authorization schema that allows users to to interact with the Data Lake ecosystem currently relies on the use of X.509 certificates [30]. Following the example of WLCG [31], the VOMS [32] stack is used in order to grant access control in distributed services and storages. These capabilities are implemented through the INDIGO IAM (Identity and Access Management) [33] service which is hosted at INFN-CNAF [34]. Nevertheless, the goal is to move towards token-based authorization by extending the work done in the context of the WLCG Authorization Working group [35] and more specifically on the WLCG JWT profile [36]. This transition will be achieved by using the OpenID Connect (OIDC) [37] identity layer on top of the OAuth2 protocol [38]. Rucio supports OIDC authentication [39] and this is already being tested in the DOMA Rucio instance [40]. In the context of ESCAPE an initial testing phase is also ongoing where those functionalities are examined with the aim of demonstrating completely X.509 free access to the resources.

3 Testing Infrastructure

The ESCAPE Data Lake testing infrastructure consists of all the tools and technologies that are deployed in order to ensure that the transfer stack functions seamlessly. For this reason, each level of the stack is challenged with various functional and stress testing scenarios. Additionally, the network performance is measured with well established tools that are specifically targeted at federated systems that have the characteristics of the Data Lake.

3.1 Transfer Capabilities

The Data Lake transfer capabilities are essentially derived from the functionality of the transfer stack that was described in section 2, that is GFAL, FTS and Rucio. A set of tools and scripts is built around these technologies.

Gfal Functional Testing

For the functional testing that concerns GFAL, a Python based script is used [41]. All RSEs in Rucio consist of one or more endpoints that are associated with a supported protocol, this testing flow examines the basic data operations one can perform on the storage level and provides results per RSE per endpoint. There are three types of operations that are being tested:

- **Upload** of a file that is a few bytes long to all the endpoints of all RSEs
- **Download** of the file that was uploaded in the previous step

- **Deletion** of the file that was uploaded in the first step

The results are pushed into an Elasticsearch [42] data source that is provided by the CERN IT Monitoring Service (MONIT) [43]. This script is also integrated with CRIC, it fetches the configuration of the RSEs before each run from the JSON endpoint ensuring that once something is configured centrally it will also be reflected on the testing scenario. The frequency that this procedure runs is every minute providing this way real time results.

FTS TPC Testing

In order to test FTS TPC transfers, a Python based toolkit is used [44]. In this case, the same endpoints as in the GFAL tests are examined and the goal is to trigger TPC transfers between all possible endpoint pairs that participate in the Data Lake. The toolkit reads from a configuration file all endpoint pairs that are to be tested as well as other parameters like number of jobs that are to be triggered, file sizes to be used, number of files per job and whether or not to enable checksum verification. Custom metadata attributes that can be attached to FTS jobs or to the files that participate in these jobs are also supported.

Before each mock transfer the endpoints that serve as the source in the copy are populated with the relevant testing files, a mechanism is in place to label an endpoint problematic if this attempt is not successful. In that case, that endpoint will not participate in the transfers. The nature of this testing is asynchronous, the toolkit will trigger the required FTS jobs and then will start polling the server in order to get the job states, as soon as a job finishes it will record the final job state on the output and remove the transferred files from the destination endpoint. The data is not piling up on the destination endpoints and thus no quota is exceeded. Extensive error handling is performed in every step making sure that the testing flow will continue even if endpoints fail to respond mid-test. This concerns mainly GFAL operations that are used to prepare the endpoints for the transfers.

The current deployed configuration performs two sets of tests. The first one transfers 1MB files with 4 files per job and all Data Lake endpoints participate making it a N:N transfer mesh (an FTS job is initiated for every endpoint pair). The second one transfers 1GB files with 4 files per job and all Data Lake endpoints participate except for some testing ones with low quota. Testing results are automatically pushed from the FTS server to an Elasticsearch data source provided by MONIT. The frequency that this procedure runs is every 30 minutes.

Rucio Testing

The testing of the Rucio data orchestrator can be broken down to a few functionalities. The main goal is to perform uploads of files by using the Rucio client and then assigning what in Rucio are called rules in order to trigger data movement across RSEs (Rucio replication mechanism). An agent is being deployed that serves this purpose [45], it uploads files to all RSEs and then it assigns rules that trigger the movement of those files to all other RSEs making this also a N:N transfer mesh like in the FTS case. Similar to the GFAL testing, this agent is integrated with CRIC and it fetches the configuration of the RSEs before each run.

The current deployed configuration concerns of file sizes of 1MB, 10MB, 100MB, 1GB and 5GB with running frequencies of 15, 30, 60, 240 and 480 minutes respectively. A second toolkit is also used [46] in order to perform the same type of tests. This is developed and deployed by the SKAO [47] team that works for ESCAPE and runs hourly tests for uploads and replication across all RSEs with 100KB files. Testing results are handled by Rucio built-in machinery and other messaging systems that are explained in detail in section 4.1.

3.2 Network Capabilities

In order to measure network performance in the Data Lake and establish end-to-end usage expectations among the RSEs the perfSONAR [48] toolkit is employed. perfSONAR is a network measurement toolkit designed to provide federated coverage of paths and end-to-end network measurements. It consists of multiple network tools brought together (e.g. iperf3, ping, traceroute) [49] in order to provide a framework which can be used to collect and analyze network metrics.

WLCG [31] and OSG [50] jointly operate a network of perfSONAR agents deployed worldwide, ESCAPE participates in this network through hosts [14] that represent the RSEs of the Data Lake. It is required that each RSE has two hosts associated with it in the system, one dedicated for bandwidth tests and one for latency. These hosts form the ESCAPE network mesh and network tests are being ran among all possible pairs that participate in it. The test specifications are the following:

- **Latency** tests that use the *owping* tool in order to perform the one-way delay test between the hosts. These tests run continuously in the background and only IPv4 is tried
- **Traceroute** tests that use the *traceroute* tool in order to measure the number of hops that are required for hosts to reach each other. These tests run every 10 minutes and no IP version is forced (both IPv4 and IPv6 are tried as supported)
- **Throughput** tests that use the *iperf3* tool in order to perform the throughput test between the hosts. These tests run every 23 hours, each test has a duration of 25 seconds and the TCP protocol is used. Both IPv4 and IPv6 are tried given that they are supported by the hosts

The results from these tests are accessible via the MaDDash [51] monitoring. They are also pushed to an Elasticsearch data source, more details are given in section 4.1.

4 Monitoring Infrastructure

The ESCAPE monitoring infrastructure consists of all the tools and technologies that are deployed in order to monitor the Data Lake along with the services that support it. It includes insightful visualizations derived from the produced data traffic of the testing activities that were described in section 3.

4.1 Architecture

The monitoring infrastructure can be divided into several technologies. The visualization platform that is used is Grafana [52], a multi-platform open source analytics and interactive visualization web application. It supports multiple storage backends (Elasticsearch, InfluxDB, MySQL, etc.) as data sources which one can query in order to fetch data and create a panel, the basic visualization building block. A dashboard is an aggregation of multiple panels, in the case of ESCAPE we organize different topics into different dashboards. Moreover, the only employed data source is Elasticsearch (ES) while the messaging technology that is used is ActiveMQ [53].

A diagram of the architecture can be seen in Figure 2. Most services are provided by the MONIT service and the CERN IT Messaging Service [54]. Two external Elasticsearch data sources are ported into Grafana, one based in STFC Cloud [55] that is operated by SKAO and one based in the University of Chicago [56] which serves the needs of the WLCG/OSG network of perfSONAR boxes.

https://doi.org/10.1051/epjconf/202125102060

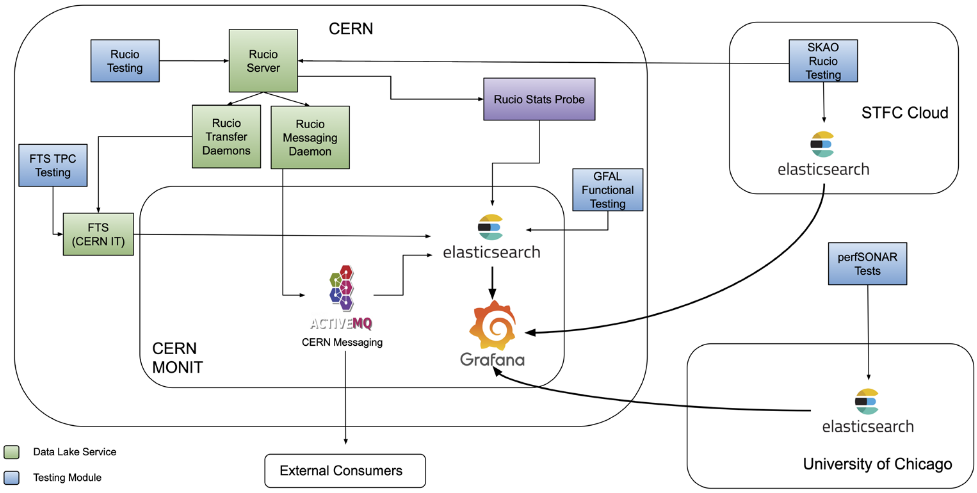


Figure 2: The Data Lake monitoring architecture

As can be seen in the diagram, the GFAL testing results get pushed directly to an ES data source while the FTS TPC testing results pass through the FTS server first before ending up to ES. Rucio testing data follows a different path, once the Rucio server becomes aware of the traffic it stores various useful information in the backend database in the form of data points (Rucio events), those are then fetched by a messaging daemon and are pushed to an ActiveMQ broker before ending up in ES. The ActiveMQ broker provides access for the same events to external consumers from the experiments that want to build tools that use them. Furthermore, since Rucio makes use of FTS, part of the TPC data comes from the transfer daemons.

The SKAO Rucio testing activity reflects on the monitoring in two different ways. Firstly, via the standard traffic that is generated on the Rucio server, this follows the Rucio Events route that was described in the previous paragraph. The second way concerns the STFC Cloud ES data source which is accessible through Grafana. This data source holds custom data that is manually pushed and is not in the chain of the standard Rucio messaging workflow.

4.2 Dashboards

The Grafana dashboards are eventually where users, experiment data managers and storage infrastructure providers go to in order to analyse and visualize data. It's where the integrity of data storage, distribution and processing can be verified. They provide a general view of the Data Lake status while at the same time they support refined views that enable users to identify issues and debug RSEs. A description of the main dashboards follows.

GFAL Dashboard

This dashboard presents the results of the GFAL operations as they were described in section 3.1. Metrics that are plotted include number of successful/failed operations per RSE while filtering capabilities exist for protocols. The user is able to identify potential issues at the storage level and observe the historical resiliency of particular RSEs over time. A representative panel can be seen in Figure 3.

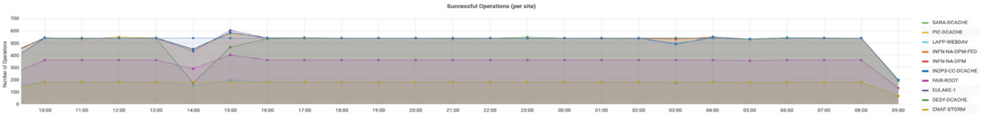
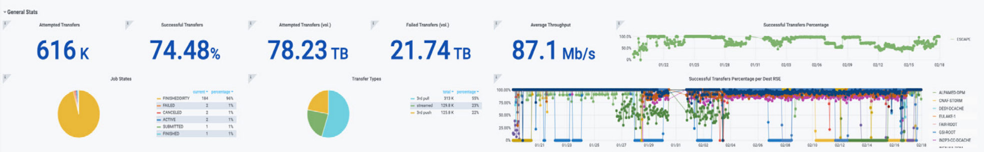


Figure 3: GFAL Successful Operations (per site)

FTS Dashboard

This dashboard presents the results of the FTS TPC transfers as they were described in section 3.1. The main highlights for a user would be the several aggregated statistics that are available (total data transferred, mean throughput, successful transfers, etc.) as well as an efficiency matrix where one can see the percentage of successful transfers from one RSE to another. This dashboard also presents debugging views of error codes and links for the logfiles of the failed transfers. The general stats view can be seen in in Figure 4, the data volume and metrics presented in this view are only indicative of the functionality of the dashboard and cover a 1 month period.


General Stats

Attempted Transfers 616 K

Successful Transfers 74.48%

Attempted Transfers (vol.) 78.23 TB

Failed Transfers (vol.) 21.74 TB

Average Throughput 87.1 Mb/s

Job States

	current	percentage
FINISHEDDIRTY	184	96%
FAILED	2	1%
CANCELED	2	1%
ACTIVE	2	1%
SUBMITTED	1	1%
FINISHED	1	1%

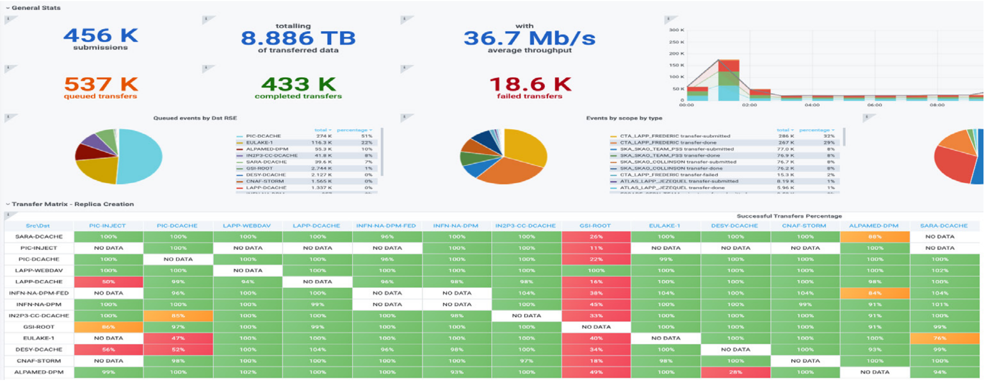
Transfer Types

	total	percentage
3rd pull	313 K	55%
streamed	129.8 K	23%
3rd push	125.8 K	22%

Figure 4: FTS transfers dashboard (general stats view - 1 month)

Rucio Events Dashboard

This dashboard presents the results of the Rucio (testing) traffic as it was described in section 3.1. It monitors the creation of replicas as well as the deletion of those. The user is able to navigate through metrics such as the total throughput from RSE to RSE, the total volume transferred and the failed or successful deletions over time. Support also exists for a refined search of specific files that participated in the replica creation process in order to pinpoint the viewing to specific cases. A part of this dashboard is displayed in Figure 5.


General Stats

456 K submissions

totalling 8.886 TB of transferred data

with 36.7 Mb/s average throughput

537 K queued transfers

433 K completed transfers

18.6 K failed transfers

Queued events by Dst RSE

	total	percentage
PIC-DCACHE	274 K	51%
EULAKE-1	116.3 K	22%
ALPAMED-DPM	55.3 K	10%
IN2P3-CC-DCACHE	41.8 K	8%
SARA-DCACHE	39.6 K	7%
GSI-ROOT	2.744 K	1%
DESY-DCACHE	2.127 K	0%
CNAF-STORM	1.565 K	0%
LAPP-DCACHE	1.337 K	0%

Events by scope by type

	total	percentage
CTA_LAPP_FREDERIC transfer-submitted	286 K	32%
CTA_LAPP_FREDERIC transfer-done	267 K	29%
SKA_SKAO_TEAM_PSS transfer-submitted	77.0 K	8%
SKA_SKAO_TEAM_PSS transfer-done	76.9 K	8%
SKA_SKAO_COLLINSON transfer-submitted	76.7 K	8%
SKA_SKAO_COLLINSON transfer-done	76.2 K	8%
CTA_LAPP_FREDERIC transfer-failed	15.3 K	2%
ATLAS_LAPP_JEZEQUEL transfer-submitted	8.19 K	1%
ATLAS_LAPP_JEZEQUEL transfer-done	5.96 K	1%

Transfer Matrix - Replica Creation

Successful Transfers Percentage

Src\Dst	PIC-INJECT	PIC-DCACHE	LAPP-WEBDAV	LAPP-DCACHE	INFN-NA-DPM-FED	INFN-NA-DPM	IN2P3-CC-DCACHE	GSI-ROOT	EULAKE-1	DESY-DCACHE	CNAF-STORM	ALPAMED-DPM	SARA-DCACHE
SARA-DCACHE	100%	100%	100%	100%	96%	100%	100%	26%	100%	100%	100%	88%	NO DATA
PIC-INJECT	NO DATA	100%	NO DATA	NO DATA	NO DATA	100%	100%	11%	NO DATA	NO DATA	NO DATA	100%	NO DATA
PIC-DCACHE	100%	NO DATA	100%	100%	96%	100%	100%	22%	99%	100%	100%	100%	100%
LAPP-WEBDAV	100%	100%	NO DATA	100%	100%	100%	100%	100%	100%	100%	100%	100%	102%
LAPP-DCACHE	50%	99%	94%	NO DATA	96%	98%	98%	16%	100%	100%	100%	98%	100%
INFN-NA-DPM-FED	NO DATA	96%	100%	100%	NO DATA	NO DATA	104%	38%	104%	100%	104%	84%	104%
INFN-NA-DPM	100%	100%	100%	99%	NO DATA	NO DATA	100%	45%	100%	100%	99%	91%	101%
IN2P3-CC-DCACHE	100%	85%	100%	100%	100%	98%	NO DATA	33%	100%	100%	100%	91%	100%
GSI-ROOT	86%	97%	100%	99%	100%	100%	100%	NO DATA	100%	100%	100%	91%	99%
EULAKE-1	NO DATA	47%	100%	100%	100%	100%	100%	40%	NO DATA	100%	100%	100%	76%
DESY-DCACHE	56%	52%	100%	104%	96%	98%	100%	34%	100%	NO DATA	100%	93%	99%
CNAF-STORM	NO DATA	98%	100%	100%	100%	100%	97%	18%	98%	100%	NO DATA	100%	100%
ALPAMED-DPM	99%	100%	102%	100%	100%	93%	100%	49%	100%	28%	100%	NO DATA	94%

Figure 5: Rucio events dashboard (FDR day)

Rucio Stats Dashboard

This dashboard consists of data that is fetched from the Rucio Stats Probe which is displayed in the architecture diagram as seen in Figure 2. The Rucio Stats Probe is a Python based script which communicates with the Rucio server through a client and queries for information like the total amount of storage used per RSE and the number of files that exist in each RSE. These metrics are also aggregated per experiment. The visualizations of this dashboard allows a user to have a general view of the Data Lake and observe the usage patterns over time across RSEs and experiments. Examples can be seen in Figures 6 and 7.

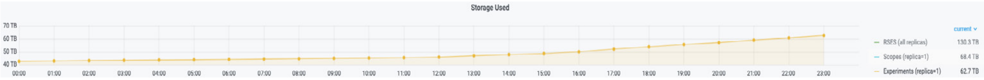
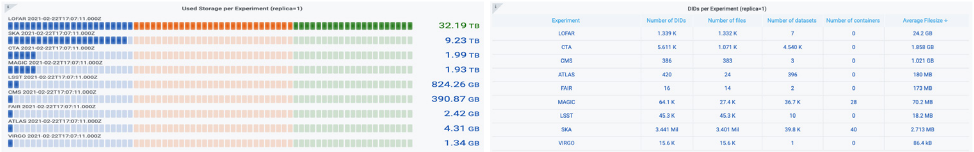


Figure 6: Used storage over time for experiments (FDR day)


Used Storage per Experiment (replica=1)
LOFAR 2021-02-22T17:07:11.000Z
32.19 TB
SKA 2021-02-22T17:07:11.000Z
9.23 TB
CTA 2021-02-22T17:07:11.000Z
1.99 TB
MAGIC 2021-02-22T17:07:11.000Z
1.93 TB
LSST 2021-02-22T17:07:11.000Z
824.26 GB
CMS 2021-02-22T17:07:11.000Z
390.87 GB
FAIR 2021-02-22T17:07:11.000Z
2.42 GB
ATLAS 2021-02-22T17:07:11.000Z
4.31 GB
VIRGO 2021-02-22T17:07:11.000Z
1.34 GB

(a) Used storage per experiment

DIDs per Experiment (replica=1)

Experiment	Number of DIDs	Number of files	Number of datasets	Number of containers	Average Filesize ↓
LOFAR	1.339 K	1.332 K	7	0	24.2 GB
CTA	5.611 K	1.071 K	4.540 K	0	1.858 GB
CMS	386	383	3	0	1.021 GB
ATLAS	420	24	396	0	180 MB
FAIR	16	14	2	0	173 MB
MAGIC	64.1 K	27.4 K	36.7 K	28	70.2 MB
LSST	45.3 K	45.3 K	10	0	18.2 MB
SKA	3.441 Mil	3.401 Mil	39.8 K	40	2.713 MB
VIRGO	15.6 K	15.6 K	1	0	86.4 kB

(b) Files per experiment

Figure 7: Rucio stats dashboard

The Full Dress Rehearsal Experience

The Full Dress Rehearsal (FDR) [57] was a full scale exercise when the experiments made extensive usage of the Data Lake capabilities by injecting data and replicating it according to their scientific workflow scenarios. This exercise lasted for 24 hours. During this period the monitoring infrastructure proved to be well prepared and essential in order to track (and improve by debugging) the performance/status of the RSEs and validate that the workflows were actually successful. Views that concern the FDR can be found in Figures 5 and 6.

5 Conclusions

The ESCAPE Data Lake has successfully reached its pilot phase and it has demonstrated a robust architecture that serves the needs and use cases of the participant experiments and science facilities. The testing infrastructure along with the monitoring capabilities allow users to validate the system status and efficiently track their activities through the functional elements that bring together the data management ecosystem.

The next steps involve work towards the Data Lake prototype. Consolidation of topics like token based authentication, quality of service mechanisms and network optimized transfers are some of the features that are to be implemented. In this aspect, ESCAPE is closely related to the WLCG activities and shares many of the data challenges that the DOMA project deals with. Synergies between the two can be very beneficial for the community in general.

Acknowledgements

Authors acknowledge support from the ESCAPE project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824064 (ESCAPE, the European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures).

References

[1] Bolton, Rosie, Campana, Simone, Ceccanti, Andrea, Espinal, Xavier, Fkiaras, Aristeidis, Fuhrmann, Patrick, Grange, Yan, EPJ Web Conf. **245**, 04019 (2020)

[2] *ESCAPE Website*, https://projectescape.eu (2021), accessed: 2021-02-20

[3] *ESCAPE Experiments & Partners*, https://wiki.escape2020.de/index.php/Experiment_and_partners (2021), accessed: 2021-02-20

[4] *EOSC Portal*, https://eosc-portal.eu (2021), accessed: 2021-02-20

[5] *Data Infrastructure for Open Science*, https://projectescape.eu/services/data-infrastructure-open-science (2021), accessed: 2021-02-20

[6] *ESFRI Science Analysis Platform*, https://projectescape.eu/services/esfri-science-analysis-platform (2021), accessed: 2021-02-20

[7] *FAIR Principles*, https://www.go-fair.org/fair-principles (2021), accessed: 2021-02-20

[8] A.J. Peters, L. Janyst, Journal of Physics: Conference Series **331**, 052015 (2011)

[9] *EOS Website*, http://eos.web.cern.ch (2021), accessed: 2021-02-20

[10] A. Alvarez, A. Beche, F. Furano, M. Hellmich, O. Keeble, R. Rocha, Journal of Physics: Conference Series **396**, 032015 (2012)

[11] 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

[12] A. Carbone, L. dell'Agnello, A. Forti, A. Ghiselli, E. Lanciotti, L. Magnoni, M. Mazzucato, R. Santinelli, V. Sapunenko, V. Vagnoni et al., *Performance studies of the StoRM Storage Resource Manager*, in *Third International Conference on e-Science and Grid Computing, e-Science 2007, 10-13 December 2007, Bangalore, India* (IEEE Computer Society, 2007), pp. 423–430

[13] *XRootD Website*, https://xrootd.slac.stanford.edu (2021), accessed: 2021-02-20

[14] *Datalake Storages*, https://wiki.escape2020.de/index.php/WP2_-_DIOS#Datalake_Status (2021), accessed: 2021-02-20

[15] *GFAL2 Documentation*, https://dmc-docs.web.cern.ch/dmc-docs/gfal2/gfal2.html (2021), accessed: 2021-02-20

[16] *GridFTP*, https://en.wikipedia.org/wiki/GridFTP (2021), accessed: 2021-02-20

[17] *HTTP*, https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol (2021), accessed: 2021-02-20

[18] A.A. Ayllon, M. Salichos, M.K. Simon, O. Keeble, Journal of Physics: Conference Series **513**, 032081 (2014)

[19] *FTS Website*, https://fts.web.cern.ch (2021), accessed: 2021-02-20

[20] 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)

[21] *Rucio Website*, https://rucio.cern.ch (2021), accessed: 2021-02-20

[22] *QoS in ESCAPE*, https://wiki.escape2020.de/index.php/ESCAPE_QoS_Architecture (2021), accessed: 2021-02-20

[23] *QoS in Rucio*, https://indico.cern.ch/event/873367/contributions/3686567/attachments/1983605/3304257/QoS_in_Rucio_and_ATLAS.pdf (2021), accessed: 2021-02-20

[24] *Kubernetes*, https://kubernetes.io (2021), accessed: 2021-02-20

[25] *Openstack*, https://www.openstack.org (2021), accessed: 2021-02-20

[26] *CERN IT - Server Provisioning Service*, https://information-technology.web.cern.ch/services/server-provisioning (2021), accessed: 2021-02-20

[27] *Oracle DBMS*, https://en.wikipedia.org/wiki/Oracle_Database (2021), accessed: 2021-02-20

[28] A. Anisenkov, J. Andreeva, A. Di Girolamo, P. Paparrigopoulos, A. Vedaee, EPJ Web Conf. **214**, 03003. 8 p (2019)

[29] *CERN IT - File Transfer Service*, https://information-technology.web.cern.ch/services/file-transfer (2021), accessed: 2021-02-20

[30] *X.509*, https://en.wikipedia.org/wiki/X.509 (2021), accessed: 2021-02-20

[31] *Worldwide LHC Computing Grid*, https://wlcg.web.cern.ch (2021), accessed: 2021-02-20

[32] *Virtual Organization Membership Service*, https://italiangrid.github.io/voms (2021), accessed: 2021-02-20

[33] *INDIGO-IAM*, https://github.com/indigo-iam/iam (2021), accessed: 2021-02-20

[34] *INFN-CNAF*, https://www.cnaf.infn.it/en (2021), accessed: 2021-02-20

[35] *WLCG Auth WG*, https://twiki.cern.ch/twiki/bin/view/LCG/WLCGAuthorizationWG (2021), accessed: 2021-02-20

[36] M. Altunay, B. Bockelman, A. Ceccanti, L. Cornwall, M. Crawford, D. Crooks, T. Dack, D. Dykstra, D. Groep, I. Igoumenos et al., *WLCG Common JWT Profiles* (2019)

[37] *OpenID Connect*, https://openid.net/connect (2021), accessed: 2021-02-20

[38] *OAuth 2.0*, https://oauth.net/2 (2021), accessed: 2021-02-20

[39] 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)

[40] *DOMA Rucio*, https://twiki.cern.ch/twiki/bin/view/LCG/DomaRucio (2021), accessed: 2021-02-20

[41] *GFAL Testing*, https://github.com/ESCAPE-WP2/Utilities-and-Operations-Scripts/tree/master/gfal-sam-testing (2021), accessed: 2021-02-20

[42] *Elasticsearch*, https://www.elastic.co/elasticsearch (2021), accessed: 2021-02-20

[43] *CERN IT - Monitoring Service*, https://monit.web.cern.ch/monit (2021), accessed: 2021-02-20

[44] *FTS TPC Tests Toolkit*, https://github.com/ESCAPE-WP2/fts-analysis-datalake (2021), accessed: 2021-02-20

[45] *Agent for Rucio testing*, https://github.com/ESCAPE-WP2/DataLake-Crons/blob/master/scripts/rucio_produce_noise.sh (2021), accessed: 2021-02-20

[46] *Rucio Tests Toolkit*, https://github.com/ESCAPE-WP2/rucio-analysis (2021), accessed: 2021-02-20

[47] *Square Kilometre Array*, https://www.skatelescope.org (2021), accessed: 2021-02-20

[48] *perfSONAR*, https://www.perfsonar.net (2021), accessed: 2021-02-20

[49] *perfSONAR Tools*, https://docs.perfsonar.net/pscheduler_ref_tests_tools.html (2021), accessed: 2021-02-20

[50] *Open Science Grid*, https://opensciencegrid.org (2021), accessed: 2021-02-20

[51] *MaDDash*, http://docs.perfsonar.net/maddash_intro.html (2021), accessed: 2021-02-20

[52] *Grafana*, https://grafana.com (2021), accessed: 2021-02-20

[53] *Apache ActiveMQ*, http://activemq.apache.org (2021), accessed: 2021-02-20

[54] *CERN IT - Messaging Service*, https://information-technology.web.cern.ch/services/Messaging-Service (2021), accessed: 2021-02-20

[55] *STFC Cloud*, https://openstack.stfc.ac.uk (2021), accessed: 2021-02-20

[56] *University of Chicago Elasticsearch*, https://atlas-analytics.github.io/ATLAS-Analytics/infrastructure/elasticsearch (2021), accessed: 2021-02-20

[57] R.D. Maria, R. Dona, *ESCAPE Data Lake: Next-generation management of cross-discipline Exabyte-scale scientific data* (2021), CHEP2021 Proceedings, in press