A transnational data system for HPC/Cloud-Computing Workflows based on iRODS/EUDAT

Martin Golasowski
IT4Innovations, VŠB – Technical University of Ostrava
Ostrava, Czechia
martin.golasowski@vsb.cz

Mohamad Hayek
Leibniz Supercomputing Centre (LRZ)
Garching b. Munich, Germany
hayek@lrz.de

Rubén J. García-Hernández
Leibniz Supercomputing Centre (LRZ)
Garching b. Munich, Germany
garcia@lrz.de

ABSTRACT

In this contribution, we present a transnational iRODS federation as a backend for distributed computational/Big Data workflows from science and industry. This system, the "LEXIS Distributed Data Infrastructure (DDI)", has been built in the project "Large-Scale EXecution for Industry and Society" (LEXIS, H 2020 GA 825532). It makes use of EUDAT B2SAFE, B2HANDLE, and B2STAGE on top of iRODS, to support a variety of use cases, starting with the three LEXIS Pilots: Simulations in Aeronautics Engineering, Earthquake/Tsunami Analysis, and Weather and Climate Prediction. Our presentation covers different aspects of setting up this system, from system to high-level concepts. We layout our experience in setting up a version of HAIRS (High-Availability iRODS System, cf. contributions of Kawai et al. to this meeting series), where we adapted the concept, and of installing EUDAT modules on top of it to provide the functionalities required within LEXIS. Afterward, we lay out our approach to integrating the iRODS system with the LEXIS platform, based on providing REST APIs to control and address the data system. These REST APIs are mostly custom LEXIS developments, based on conventional programming interfaces (e.g. python client) available for iRODS. Finally, we give a short status and outlook on the application of the system, and further aspects of our project interesting for the iRODS community (e.g., authentication via OpenID Connect with adaptation to Keycloak, collection structure of the iRODS system and backend systems, as also described in LEXIS publications).

Keywords

iRODS, EUDAT, High Performance Computing, Workflows

INTRODUCTION

The LEXIS project takes on a challenge of orchestrating complex HPC, Cloud and Big Data workflows on the resources of competitive European Supercomputing centers (LRZ/DE, IT4I/CZ, ICHECH/IL). It is a European Horizon 2020 project which runs between 2019-2021. The challenge has three main parts. The first one is the orchestration of workflows running on geographically distributed, federated Cloud and HPC resources. The second challenge is to provide a unified way for storing complex data and transparently move it between the compute resources, which led us to design the iRODS-based LEXIS Distributed Data Infrastructure (DDI). The last challenge is to provide a robust and secure Authentication and Authorization Infrastructure (AAI) for the whole platform.

The orchestration system in the LEXIS Platform is based on the advanced distributed orchestration framework YORC [1] which uses an extension of the TOSCA [2] standard to describe the workflows. The orchestration system then uses the LEXIS DDI to manage the data used by the individual workflow tasks [3] [4].

iRODS UGM 2021, June 8-11, 2021, Virtual

This paper focuses on integration of iRODS [5] in the LEXIS platform, where it is used as core of the DDI. The platform is implemented by a set of services which communicate by REST-based APIs with zero-trust architecture based on JWT [6] tokens and the OpenID Connect [7] authentication protocol. In this work, we also briefly describe the REST APIs we built on top of the iRODS to provide data management and staging capabilities for the platform.

LEXIS DISTRIBUTED DATA INFRASTRUCTURE

The LEXIS DDI must be able to handle large data sets (TBs) following the FAIR[1] principles, must move the data seamlessly between various locations and expose REST based APIs for the management. A schema showing the distributed infrastructure context and with various compute and storage resources connected to the LEXIS DDI is given in Figure 1

The iRODS solution has been selected as a base of the LEXIS DDI since it provides sufficient abstraction of different storage resources available at each center. Individual zones run by the centers are federated which allows seamless transfer of the data between locations while maintaining strict access policy. Data annotation and publication capabilities of the LEXIS DDI are implemented using the iRODS metadata available for each dataset and collection. The EUDAT [8] modules B2HANDLE [9] and B2SAFE [10] are used to obtain unique identifiers (PIDs) for the datasets stored in the iRODS. These add-ons leverage the internal iRODS rule engine and Python scripts.

The LEXIS DDI is formed by federated iRODS zones, which host data in a strictly defined structure, a set of REST APIs and various support services for token exchange, monitoring, auditing and publishing data.

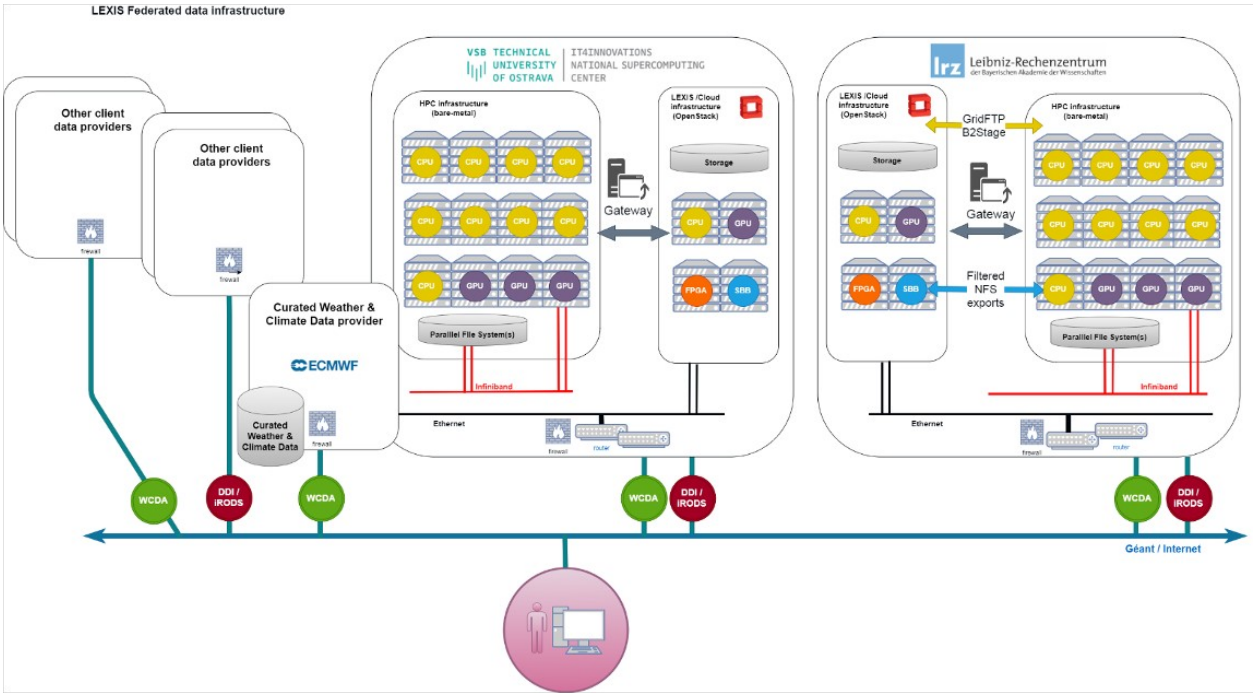

Figure 1 LEXIS Distributed Data Infrastructure connecting several compute and data providers

[1] FAIR data are data which meet principles of findability, accessibility, interoperability, and reusability cf. e.g. www.force11.org/group/fairgroup/fairprinciples

IRODS OPENID INTEGRATION

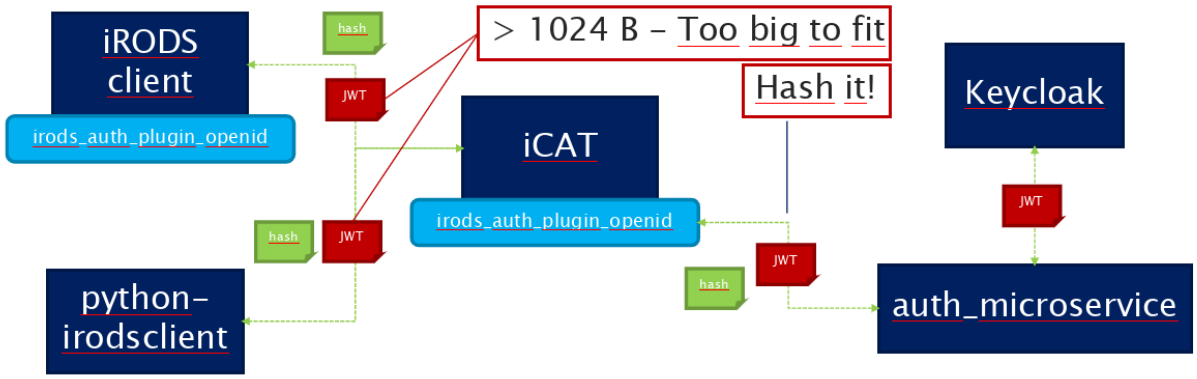

Figure 2 iRODS - OpenID authentication flow in LEXIS. The three JWT transfers (marked red) are substituted by transfers of the token hash (green)

The core iRODS software provides basic authentication mechanisms such as username / password combinations, and more advanced authentication is handled via plugins. An OpenID authentication plugin was developed by Kyle Ferriter in 2018 [11], and the iRODS python client was forked to support OpenID access, although the changes were not ported upstream, and it did not see widespread use. The current iRODS plan, to the authors' knowledge, is the integration of OpenID functionality directly in iRODS in an upcoming version.

We used these plugins as a starting point to integrate the DDI with LEXIS AAI, but some changes were needed. The most significant one relates to the fact that Keycloak [12] (the Identity Managment solution which LEXIS uses) creates quite large JWT [6] tokens, larger than the space allocated for them in the iRODS plugin, which uses the 1024 bytes long username field to transfer tokens. Our solution checks if the token is too large, and hashes it before sending it to iRODS (Figure 2). On the iRODS plugin side, the changes include an additional check in the database to see if a token exists with the proper hash. This necessitates that the webportals using iRODS as a back-end pre-authorize the token by calling the iRODS broker (the plugin component taking care of interfacing with the Keycloak server.

Further changes included bugfixes and merging of upstream changes to ensure compatibility with the latest iRODS versions. We have published the changes, although once a new iRODS version including native OpenID Connect support is published, they will be rendered obsolete.

SETTING UP IRODS IN HIGH AVAILABILITY

Since LEXIS interconnects many different components, any failure in one of the components can have huge consequences on the entire system. To avoid a single point of failure scenario, we aimed to setup iRODS in high availability mode in all centers involved in LEXIS.

A version of HAIRS [13] (High-Availability iRODS System, cf. contributions of Kawai et al. to this meeting series) was deployed. The setup consists of two iCAT servers based on the iCAT database. In front of the two servers is an instance of HA Proxy. All three instances refer to themselves with the FQDN of the iRODS server. A request to iRODS will be routed through the HA Proxy to one of the iCAT servers.

The weak remaining then is the iCAT database. If the database is down, the two iCAT servers cannot respond appropriately to requests. To secure the database setup, we opted for a deployment based on repmgr [14] and pgpool [15]. The setup consists of two PostgreSQL instances. On both instances repmgr is deployed to manage the replication between the two instances. At any point in time, only one instance is the primary instance and read and

write access is allowed. The second instance is in secondary mode. All updates to the primary database are propagated to the secondary via repmgr. In front of the PostgreSQL instances is an instance of pgpool. The main role of pgpool is to manage and trigger the failover once the primary database is down. In that scenario, pgpool declares the secondary database as primary and allows read and write access to it. The LEXIS high availability setup is depicted in Figure 3.

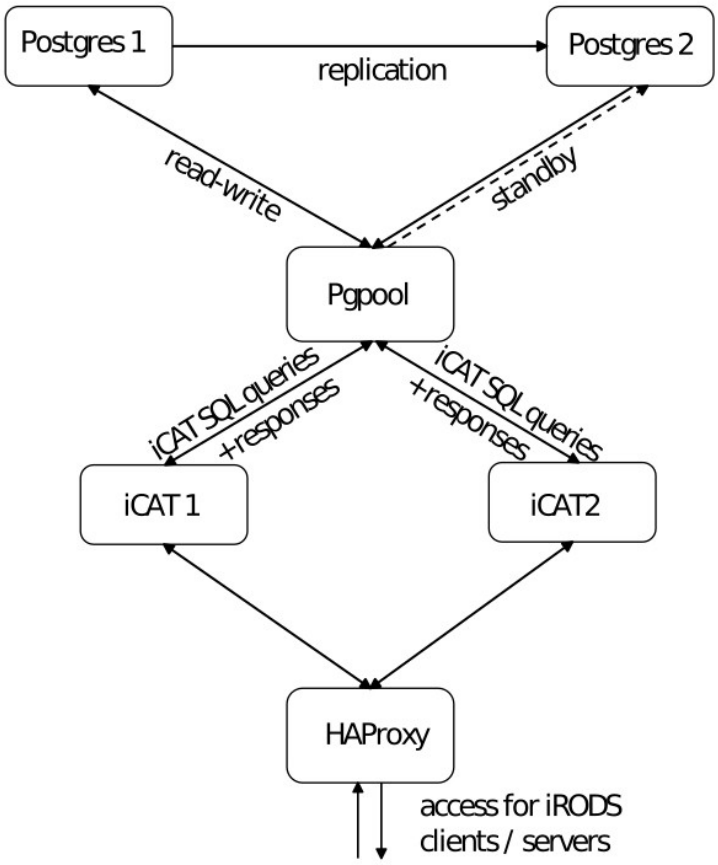

Figure 3 LEXIS redundant iRODS setup per zone

INTEGRATION WITH EUDAT

The EUDAT [8] Collaborative Data Infrastructure (CDI) offers different data management solutions for European research. Since LEXIS and EUDAT aim at achieving the FAIR data principles, we decided to deploy some of their components such as B2HANDLE [9], B2SAFE [10] which relies on iRODS, and B2STAGE [16] in LEXIS. B2HANDLE guarantees long term references to data by assigning persistent identifiers (PID). B2HANDLE is based on the HANDLE [17] system.

Another important EUDAT component is B2SAFE. B2SAFE is employed in LEXIS to manage the replication and long-term preservation of data between the different iRODS zones. This helps in optimizing the access for users on the different computing systems available in LEXIS.

To move data from a non iRODS environment to iRODS, B2STAGE was deployed. B2STAGE employs a GridFTP server on top of iRODS and allows user to move data in or out of iRODS via basic GridFTP commands,

CUSTOM APIS AND THE USE OF THE IRODS PYTHON CLIENT

The LEXIS DDI provides custom APIs to communicate with the webportal of the LEXIS platform and the LEXIS orchestrator. This helps in automating the execution of workflows. The APIs are divided into three categories: LEXIS iRODS API, Staging API, and encryption/compression API.

The LEXIS iRODS API manages the lifecycle of users and projects. The API provides endpoints to create and delete users across the federated iRODS zones, create projects and collections across the federated iRODS zones, set user's ACLs based on roles in LEXIS, and to obtain tokens that are used to connect to iRODS.

The Staging API provides the possibility to move data between the DDI and our cloud computing and HPC systems. The API is based on Django [18] and uses LEXIS Authentication and Authorization Infrastructure (AAI) to authenticate users' requests. Since data transfer is a time-consuming task, a queuing system using Celery [19] and RabbitMQ [20] is deployed. The user receives a request ID when initiating a request. The ID can be used to track the status of the data transfer, which proceeds asynchronously.

To improve data transfer rate and to secure sensitive data in iRODS, the encryption/compression API is used. The API uses a similar queuing system and performs compression on datasets with large numbers of small files for transfers. The encryption works asynchronously: Data are encrypted when moving to iRODS and then decrypted when staged to the target. The API uses aes-256-ctr encryption mode and runs on a dedicated LEXIS Burst Buffer/Data Node with 64 CPU cores, and large NVMe disk.

CONCLUSION

The distributed data infrastructure showcased in this paper provides a fault-tolerant back-end for the data tasks in the LEXIS European Cloud-HPC Workflow Platform (H2020), including efficient data transfer across sites across Europe. We leverage iRODS capabilities to federate geographically distributed data sources, and EUDAT services (especially B2SAFE) to ensure DATA FAIR principles.

We are currently at the stage of benchmarking and optimizing the performance of the system. We tested different iRODS setups, and finally selected the HAIRS deployment with redundant PostgreSQL setup due to its high-availability properties. We have also made extensive use of the iRODS Python client as part of the interfaces to other LEXIS components. For further information, the reader is invited to have a look at our book publication [21].

Setting up iRODS-OpenID Connect authentication in order to provide a single-login interface across all the LEXIS components was a challenging endeavor, due to some iRODS assumptions (maximum token size) which were not satisfied by our OpenID solution (Keycloak), and which required modifications to the iRODS authentication plugin. We therefore look forward to the finalization of the iRODS work on native implementation of OpenID Connect authentication.

ACKNOWLEDGMENTS

This work and all contributing authors are funded/co-funded by the EU's Horizon 2020 research and innovation programme (2014-2020) under grant agreement no. 825532 (Project LEXIS – "Large-scale EXecution for Industry and Society"). The authors would like to sincerely thank the colleagues from the iRODS consortium and EUDAT for their help and collaboration.

REFERENCES

[1] Atos BDS R&D, "Yorc 4.0.2," 2020. [Online]. Available: http://yorc.readthedocs.io/. [Accessed 29 Jul. 2020].

[2] OASIS TC, "Topology and Orchestration Specification for Cloud Applications Version 1.0," OASIS Standard, 25 Nov. 2013. [Online]. Available:

http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html. [Accessed 27 Apr. 2020].

[3] V. Svatoň, J. Martinovič, J. Křenek, T. Esch and P. Tomančák, "HPC-as-a-Service via HEAppE Platform," in *Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019). Advances in Intellegent Systems and Computing, 993*, Sydney, Australia, 2019.

[4] LEXIS Project, "Deliverable 4.2: Design and Implementation of the HPC-Federated Orchestration System - Intermediate," [Online]. Available: https://lexis-project.eu/web/wp-content/uploads/2020/08/LEXIS_Deliverable_D4.2.pdf. [Accessed 2 Apr. 2021].

[5] H. Xu, T. Russell, J. Coposky, A. Rajasekar, R. Moore, A. de Torey, M. Wan, W. Shroeder und S.-Y. Chen, iRODS Primer 2: Integrated Rule-Oriented Data System, Williston, VT: Morgan & Claypool Publishers, 2017.

[6] M. Jones, B. Campbell and C. Mortimore, "RFC 7523 - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants," Internet Engineering Task Force (IETF), May 2015. [Online]. Available: https://tools.ietf.org/html/rfc7523. [Accessed 1 Apr. 2020].

[7] N. Sakimura, J. Bradley, M. B. Jones, B. de Medeiros and C. Mortimore, "OpenID Connect Core 1.0 incorporating errata set 1," The OpenID Foundation, 8 Nov 2014. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html. [Accessed 27 04 2020].

[8] EUDAT Collaborative Data Infrastructure, "EUDAT," EUDAT Ltd, 2020. [Online]. Available: https://www.eudat.eu. [Accessed 27 Apr. 2020].

[9] EUDAT Collaborative Data Infrastructure, "B2HANDLE - EUDAT," EUDAT Ltd, 2020. [Online]. Available: https://www.eudat.eu/services/b2handle. [Accessed 13 Apr. 2020].

[10] EUDAT Collaborative Data Infrastructure, "B2SAFE - EUDAT," EUDAT Ltd, 2020. [Online]. Available: https://www.eudat.eu/services/b2safe. [Accessed 13 Apr. 2020].

[11] K. Ferriter, „ OpenID Connect Authentification in iRODS,“ in *iRODS User Group Meeting*, Durham, 2018.

[12] JBoss (Red Hat Inc.), Keycloak Community, "Keycloak," [Online]. Available: https://www.keycloak.org/. [Accessed 10 Apr. 2020].

[13] Y. Kawai und A. Hasan, „High-Availability iRODS System (HAIRS),“ in *Proceedings of the iRODS User Group Meeting 2010: Policy-Based Data Management, Sharing and Preservation*, Chapel Hill, NC, 2010.

[14] 2ndQuadrant Ltd., "repmgr - Replication Manager for PostgreSQL clusters," 2020. [Online]. Available: https://repmgr.org/. [Accessed 13 Apr. 2020].

[15] SRA OSS, Inc. et al., "pgpool Wiki," 2020. [Online]. Available: https://www.pgpool.net/mediawiki/index.php/Main_Page. [Accessed 13 04 2020].

[16] EUDAT Collaborative Data Infrastructure, "B2STAGE - EUDAT," EUDAT Ltd, 2020. [Online]. Available: https://www.eudat.eu/services/b2stage. [Accessed 13 Apr. 2020].

[17] B. Boesch, S. X. Sun and L. Lannom, "RFC 3650 - Handle System Overview," Internet Engineering Task Force (IETF), Nov. 2003. [Online]. Available: https://tools.ietf.org/html/rfc3650. [Accessed 27 Apr. 2020].

[18] Django Software Foundation, "Django," 2020. [Online]. Available: https://www.djangoproject.com. [Accessed 2020 Apr. 1].

[19] Celery, "Celery: Distributed Task Queue," 2020. [Online]. Available: http://www.celeryproject.org/. [Accessed 1 Apr. 2020].

[20] Pivotal Software, "Messaging that just works — RabbitMQ," 2020. [Online]. Available: https://www.rabbitmq.com/. [Accessed 1 Apr. 2020].

[21] J. M. O. Terzo, HPC, Big Data, AI Convergence Towards Exascale. Challenge and Vision, Taylor & Francis Ltd (Verlag), 2021.