4

Data System and Data Management in a Federation of HPC/Cloud Centers

Johannes Munke, Mohamad Hayek, Martin Golasowski, Rubén J. García-Hernández, Frédéric Donnat, Cédric Koch-Hofer, Philippe Couvee, Stephan Hachinger, and Jan Martinovič

CONTENTS

DOI: 10.1201/9781003176664-4

4.1 Introduction: Data Federation of European HPC/Cloud Centers

The Federation of European computing centers, for science and engineering to profit from the best-suited computing resources regardless of location, has been a hot topic for more than a decade. Many projects and initiatives have been trying to realize this by large-scale grid-computing infrastructures [1] like TeraGrid [2], WLCG [3], and EGI [4], to name but a few. Crucial to ongoing approaches (see, e.g. [4,5]), building on that work, are easy usability for multiple use cases and user-friendly authentication and authorization (e.g. [6,7]). To profit from the use of geographically distributed compute resources, it is necessary to access data in an efficient way independently of the location of the data store. To cover this demand, the EUropean DATa (EUDAT [5]) initiative offers several tools and services for collaborative and distributed data management.

This chapter describes our approach for the implementation of an EUDAT-based distributed data infrastructure (DDI) which is part of a data-driven computing platform enabling Large-Scale Execution for Industry and Society (LEXIS [8] – and see Chapter 2 in this volume). The platform executes orchestrated big data workflows on European high-performance-/cloud-computing resources in a multi-site setup. The LEXIS project is coordinated

by the IT4Innovations National Supercomputing Center (IT4I, Ostrava, CZ) which also represents one compute/data site. The Leibniz Supercomputing Centre (LRZ, Garching b. München, DE) is the second founding partner of the computing and data federation, with both partners providing access to classical high-performance computing (HPC) and infrastructure-as-a-service-cloud (IaaS-cloud) resources. Both centers are representative (Tier 0/1) sites within the PRACE federation of European HPC centers. The next centers connecting to the platform are the European Center for Medium-Range Weather Forecasts (ECMWF, Reading, UK) and the Irish Centre for High-End Computing (ICHEC, Dublin, IE).

Besides the DDI, LEXIS offers advanced workflow orchestration (Bull/ATOS Ystia Orchestrator [9], employing TOSCA [10] and Alien4Cloud [11]), an accounting and billing system, and a portal for workflow and data control. The different access modalities of the compute resources are overcome by the use of the high-end application execution middleware (HEAppE [12]). The infrastructure planning and development is driven by the requirements of several pilot test cases: (1) the computational modeling of a data-intensive turbo-machinery and gearbox system in aeronautics; (2) real-time data processing and simulation of earthquakes and tsunamis; and (3) weather and climate models based on massive amounts of in situ data.

LEXIS draws advantage from its multisite architecture for computing and data handling. With respect to the DDI, obvious strong points are the redundancy and data safety aspects in a geo-distributed storage/mirroring scheme. As a core component and data middleware for the DDI, the Integrated Rule-Oriented Data System (iRODS [13]) and EUDAT-B2SAFE [14] are used and every site is represented by one iRODS zone. iRODS holds file-system-like and individual metadata for data sets in a metadata catalogue (iCAT), enabling FAIR (findable, accessible, interoperable, and reusable [15]) data management. Every zone relies on an own catalogue service provider (iCAT server), which in LEXIS is set up redundantly for eventually reaching high availability (HA). iRODS facilitates a unified view on the user's data from all sites, contributing to the unified LEXIS look and feel conveyed by the LEXIS authentication and authorization infrastructure (AAI), which provides a single sign-on (SSO) with Keycloak [16]. The DDI integrates with the other components of the LEXIS platform via several REST APIs (e.g. for data transfer). It is embedded into the European data landscape by using EUDAT tools and federates easily with further EUDAT/iRODS sites.

In Sections 4.2 and 4.3, we start out describing design requirements and middleware-based implementation of the LEXIS DDI. Section 4.4 discusses the hardware backend of the DDI, with emphasis on the conceptually new usage of buffering servers (burst buffers, data nodes) in order to accelerate data transfer and conversion. Section 4.5 describes the integration of the system with the LEXIS AAI, and Section 4.6 the REST-API-based integration with other functional units of the LEXIS platform. After elaborating on the

European (EUDAT) integration and FAIR data aspects of the DDI (Section 4.7), we conclude (Section 4.8).

4.2 Requirements on the LEXIS DDI

Within the LEXIS federation, users and orchestration systems must be able to retrieve data in a secure and efficient way independent of location. In the context of the LEXIS project, this idea resulted in the following detailed requirements to the LEXIS DDI for the design process.

4.2.1 Unified Data Access

Since user and workflow data in the distributed LEXIS platform will be used in a cross-site manner, it has to be uniformly accessible from everywhere. iRODS stores files (data objects) in folders (collections) and subfolders (subcollections) and thus provides a hierarchical structure comparable to a unified file system, with data at all sites integrated into it (see [13]).

4.2.2 Usage and Federation of Diverse Data Backend Systems

Federating different HPC/data centers means bringing storage resources of different technological nature into line. As LEXIS is aiming at a growing federation, the DDI must be versatile concerning different data backend systems. In this sense, iRODS is a good choice since it acts as a middleware and supports a variety of storage types. A description of the various hardware backend systems used at IT4I and LRZ can be found in Section 4.4.

4.2.3 Reliability and Redundancy

The LEXIS project sets up reliable services in terms of availability and data safety. High-Availability iRODS Systems (HAIRS) are described in [17] and [18]. Section 4.3 describes the LEXIS implementation of a redundant, geographically distributed iRODS system for the LEXIS DDI following this approach. This offers the possibility of data mirroring and therefore fault-tolerance concerning site-specific data loss.

4.2.4 AAI Support

For a cross-site unified user access, the DDI must be compatible with a unified, LEXIS-wide AAI system, which nowadays means supporting interfaces like OpenID Connect or SAML [19, 20]. Authentication with the Keycloak-based

LEXIS AAI is enabled using an adapted iRODS-OpenID plugin, and access rights on collections and data objects are then controlled via iRODS's built-in mechanisms (see Section 4.5).

4.2.5 APIs

The immersion of the LEXIS DDI within the LEXIS platform, that is, the connection to, for example, LEXIS Portal and LEXIS Orchestrator is implemented with several RESTful APIs. The already existing iRODS APIs and client toolkits are augmented with a set of custom-designed REST APIs specific to LEXIS (Section 4.6).

4.2.6 State-of-the-art Research Data Management

To facilitate an adequate and FAIR [15] research data management and integrate with the European data management landscape and standards, LEXIS mainly makes use of EUDAT Services, as described in Section 4.7. iRODS's capability to hold basic metadata of data sets, in combination with PID assignment by EUDAT's B2HANDLE [21] is a key component for FAIR data management.

4.3 Federation via a DDI Based on iRODS

LEXIS looked into different data management solutions that could fit our requirements described in Section 4.2. As a result, the iRODS middleware [13], also used by EUDAT [5], was finally chosen to be the core of the LEXIS data system. Data ingestion, movement, and retrieval within/from the DDI are exclusively performed via REST APIs described in Section 4.6 in order to ensure sanitized usage patterns and security (restriction of entry points) within the LEXIS ecosystem. In the following sections, we summarize some basics of iRODS and then discuss the deployment of iRODS at LRZ and IT4I with one redundantly set-up iRODS zone per center, and the federation of these zones.

4.3.1 Relevant Basic Properties of iRODS

As briefly outlined in the previous sections, iRODS uses backend file systems (or also object storage systems) to provide a unified file-system-like structure of data objects and collections (similar to files and folders). To this end, an iRODS zone – the smallest usable entity of an iRODS federation – runs one iRODS server (provider server) with metadata catalogue iCAT, and zero or

more iRODS servers on machines with storage attached (consumer servers, connected to the provider). Several iRODS zones can be federated to form a large system. Beyond these federation capabilities, the middleware excels through a rule engine, running scripts on certain events (like file creation), and has the ability to store metadata with each data object or collection in an attribute–value–unit tuple (AVU – see [13]) store. Also, iRODS offers diverse client suites (command-line interface, Python bindings, etc.) which enable us to address it from our LEXIS-specific APIs (Section 4.6).

4.3.2 iRODS HA Setup

Although the iRODS zones in LEXIS are independent entities (one per computing/data center), unavailability of one of the zones could have strong consequences on workflows being executed. Such consequences can be a complete failure of the workflow, or a slower execution when the orchestrator has to get data from a different iRODS zone, far from the HPC/cloud resources used. To improve the reliability of each zone, we thus establish redundancy and a failover mechanism for the iCAT server and its database. The iRODS backend was deployed following the HAIRS concept [17,18] and the PostgreSQL database was deployed following the concept in [22], which is based on Pgpool-II [23] and repmgr [24]. Figure 4.1 depicts the setup which can reach high availability if required.

4.3.3 iRODS Zones Federation across Centers and Data Movement

An iRODS zone manages the physical storage resources at each center. It holds metadata on stored data, users, and their access rights. All this information is saved in the local iCAT database. LEXIS uses the iRODS federation mechanism to connect the iRODS zones of the centers. This allows users from a zone to access the data in a different zone while being authenticated to their home zone. Effectively, users thus have a unified access and view to data located at multiple centers. When data are only available in a remote zone, it is transparently acquired on access via the internal transfer mechanisms of iRODS. Thus, the iRODS federation covers core requirements on the LEXIS data system.

4.3.4 Storage Tiering and Underlying Data Storage

In the setup described in Section 4.3.2, the iRODS server and its iCAT database run in high-availability mode. However, if the physical storage resource is down or corrupted, data are either inaccessible or lost. While this can be avoided through geographical data mirroring via the Replication API (see Section 4.6), the iRODS storage tiering plugin [25] deployed in LEXIS provides flexible management of backend storage, including redundancy

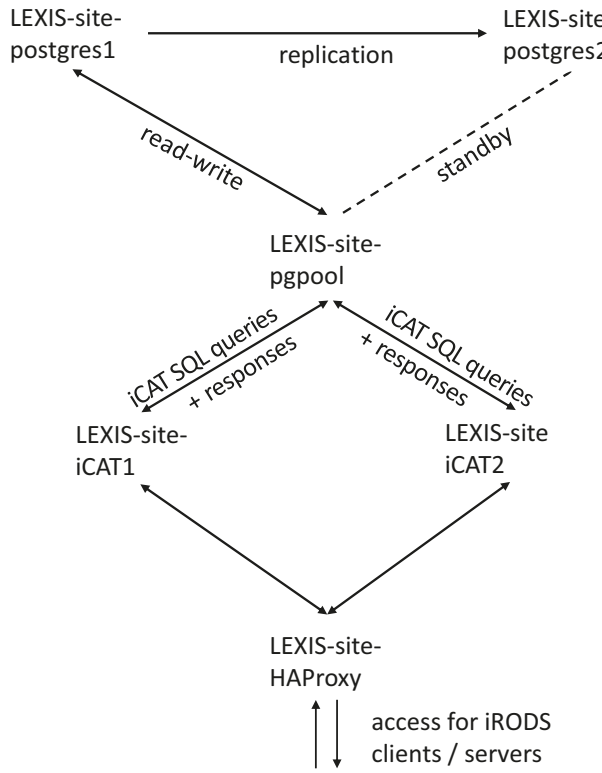

FIGURE 4.1
Redundant iCAT-PostgreSQL system following the HAIRS concept. iRODS clients and servers talk to a HAProxy which acts as a load balancer for iCAT1 and 2. The SQL queries are forwarded to the postgres1 database if it is available; otherwise failover to postgres2 is triggered. To maintain consistency, data are continuously replicated from postgres1 to 2 before failover.

when necessary. The plugin organizes multiple storage resources into so-called tiers. Once data are transferred to an iRODS zone, the data are stored automatically in the highest storage tier. The plugin sets an expiry time for the data at each tier and once this time is exceeded, the data are moved or replicated to the next tier.

4.3.5 Logical Structure of the DDI

Within the DDI, data are organized as so-called data sets (corresponding to an iRODS collection with data objects and possibly sub-collections in it). These data sets have a universally unique identifier (UUID) and can contain input and output data for a specific workflow. Internally, the DDI organizes data sets for each LEXIS computational project (see Section 4.1) and user in

three different trees, according to privacy level: Inside the */public* tree, data sets are organized by project but publicly visible. Inside the */project* tree, the data are project-private. Inside the */user* tree, data are finally private to each user (here, collections containing data are organized by project and user). This logical structure makes debugging and rights management easier.

4.4 Hardware

In this section, we discuss the storage (and computing) resources behind the LEXIS platform, as far as relevant to the DDI (see Section 4.3.4). With IT4I, LRZ, ICHEC, and ECMWF, LEXIS federates four European top-level computing and data centers with PRACE Tier-0 and Tier-1 computing resources and an aggregate peak performance of more than 30 PFlop/s. Besides more than a PByte of temporary shared-usage storage accessible by LEXIS (Section 4.4.1), resources of several 100s of TBs are exclusively available and immersed within the DDI (Section 4.4.2). In order to access file systems faster and to provide very fast volatile storage (e.g. for data encryption or conversion), data nodes (Section 4.4.3) have been installed within the LEXIS infrastructure. The use of these machines is supported by the Smart Burst Buffer (SBB) and Smart Bunch of Flash software products by ATOS, which have been adapted and evolved in the course of the LEXIS project.

4.4.1 Storage Systems for HPC and Infrastructure-as-a-Service-Cloud Clusters

HPC cluster storage usually consists of several tiers, offering different economically viable combinations of space, speed, and reliability characteristics (where usually large space and high bandwidth anti-correlate with reliability and data safety). Parallel file systems such as LUSTRE [26] (at IT4I) or IBM Spectrum Scale (ex GPFS [27], at LRZ) provide the necessary I/O bandwidth for parallel applications, as they are typically executed on HPC clusters. For security reasons, access to these large (e.g. 1.4 PB on LRZ's Linux Cluster) storage systems from outside is restricted only to a handful of protocols such as SSH (i.e. SCP, SFTP) or GridFTP [28].

For applications which are not relying on distributed-memory parallelization and fast interconnects, but rather need configurability and large run time (or uptime in case of services), IaaS-clouds are an optimum environment. These environments ideally complement the HPC ecosystem, and allow for the execution of smaller applications (pre/postprocessing) and services. They can be extended by container-orchestration frameworks such as Kubernetes as well. Consequently, LEXIS immerses OpenStack- and VMWare-based

IaaS-cloud resources provided by IT4I and LRZ (with several thousand CPU cores in total), both backed by large CEPH [29] storage clusters (100 TB–1 PB, partially SSDs). The storage resources are usually addressed from within virtual machines.

4.4.2 Storage Systems Dedicated to LEXIS

IT4I's Cloud CEPH storage cluster provides 120 TB of raw HDD backed storage and 40 TB of raw SSD backed storage. Via the POSIX compatible file system CephFS, it is used as resource for IT4I's iRODS zone. In LRZ, iRODS uses a two-tier (see Section 4.3.4) backend, with the high (fast) tier being a 50 GB partition of LRZ's Data Science Storage (DSS), based on IBM Spectrum Scale and accessible via NFS. The lower tier (with slower network connection) is a 150 TB LRZ-LEXIS Experimental Storage system (legacy IBM DS3500, being replaced and extended to 300 TB), which also hosts the LEXIS Weather and Climate Data API (WCDA).

4.4.3 HPC–Cloud-Storage Interconnect and Data Node/Burst Buffer Concept

In LEXIS, practically all systems at each site are interconnected with 10 Gbit Ethernet or better (with exception of the LRZ-LEXIS Experimental Storage). This supports the LEXIS idea of running mixed data-driven workflows using HPC and IaaS-cloud computing infrastructures, demonstrating and driving the convergence of computing paradigms. To further optimize data flows and implement in-memory data encryption/decryption, (de)compression, augmentation, or preprocessing tasks, data nodes – that is, servers with TBytes of NVMe-SSD storage and Intel Optane DC NVDIMMs – were acquired. The characteristics of the systems at IT4I and LRZ are given in Table 4.1. They can

TABLE 4.1

Characteristics of data node/burst buffer servers at IT4I and LRZ

Characteristic	IT4I node 1	IT4I node 2	LRZ node 1	LRZ node 2
CPU	2 Intel Skylake Xeon Gold 6230 (2x20 cores)	Intel Skylake Xeon Gold 6230 (2x20 cores)	Intel Skylake Xeon Gold 6230 (2x16 cores)	Intel Skylake Xeon Platinum 8260M (2x24 cores)
RAM	192 GB	192 GB	384 GB	384 GB
NVMe Capacity	12.8 TB	12.8 TB	12.8 TB	—
Optane DC NVDIMM capacity	512 GB	512 GB	1.5 TB	3.0 TB
Accelerator card	NVIDIA Quadro RTX 6000	FPGA card Bittware 520N with Intel Stratix 10	NVIDIA V100	NVIDIA V100

be used as bare-metal machines for data-intensive tasks, to host a hypervisor for virtual machines, or to run software of the project partner ATOS and be used as SBB or Smart Bunch of Flash hosts. These two concepts are set out below.

4.4.3.1 SBF (Smart Bunch of Flash)

The Smart Bunch of Flash component of the ATOS Flash Accelerator software creates a persistent NVMe volume, spread over all NVMe devices, and an XFS file system in it, which is exported to computing servers using the NVMeOF protocol [30]. The allocation of these volumes can be automated via SLURM or managed via OpenStack Cinder and the LEXIS orchestration system. These fast volumes are ideal to accelerate I/O-bound tasks (simple compression, conversion, encryption) when the pure network speed to Data Node servers is superior to the write rate of (largely HDD-based) file systems. Thus, SBF volumes will, for example, be used as a backend storage for the compression/encryption API in LEXIS (see Section 4.6.5).

4.4.3.2 SBB

In a parallel I/O, HPC setting, data nodes can be used for buffering input and output between the compute nodes of a cluster and its parallel file system (see Section 4.4.1). As illustrated in Figure 4.2, the SBB component of the ATOS Flash Accelerator suite is designed to implement a transparent cache for a parallel file system such as Lustre, GPFS, or CephFS, in such use cases. To this end, it provides:

- a client-side library intercepting I/O calls to glibc and forwarding them to a *sbbd* daemon on the server side; this library (see Figure 4.2, left part) is engaged with a simple LD_PRELOAD, and works without modifying the application – at least if linked dynamically – and without root access; and
- a server-side *sbbd* daemon (Figure 4.2, lower middle part) in charge of processing the intercepted I/O calls, managing the cache, and finally asynchronously destaging the cached data to the parallel file system (Figure 4.2, right part).

Likewise, before processing, a prefetching of data into the data node can be implemented to make data available to an application (Smart Prefetch, Figure 4.2, upper middle part). As already mentioned, within the data flow the data node can be used for pre/postprocessing tasks as well.

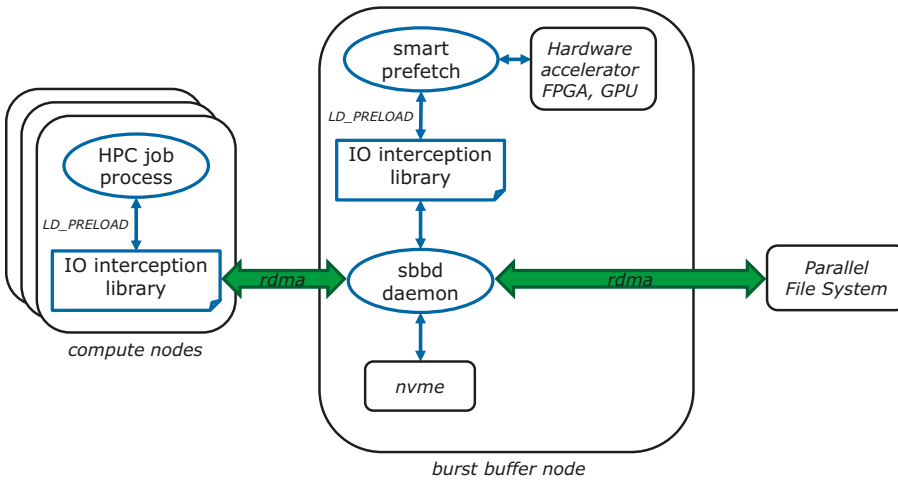

FIGURE 4.2
ATOS SBB operation concept (for an explanation of the key architectural components (I/O interception library, sbbd, prefetch mechanism), refer to the text).

4.5 Unified Access to the Platform Based on an AAI

In multiuser IT services such as the LEXIS platform and DDI, authentication of users and identity management is crucial to the service provider. The topic of authentication, authorization, and access control interplays with legal aspects (e.g. privacy, data security, and responsibilities after a hacking attempt), fundamentals of accounting (who to bill for services rendered), and informational aspects (e.g. user contact data) in a complex way. A verified identity allows the platform to identify permission classes (roles) for the users, granting them certain access rights and enabling their actions. Distributed data and computing infrastructures [1,2,3,4] have long promoted an identity management based on X.509 certificates, certification, and registration authorities. Recent approaches prefer less bureaucratic solutions based on, for example, OpenID Connect [19] or SAML [20] tokens and SSO.

4.5.1 LEXIS Identity and Access Management (IAM) Solution, SSO, and AAI

In this spirit, a LEXIS identity management and SSO system (LEXIS authorization and authentication infrastructure (AAI)) grants access to all the platforms' geographically distributed services with one login action (for a certain period

of time). Cumbersome multiple logins within our ecosystem are thus avoided. These would not only affect the user experience, but also make users tend to reuse IDs and passwords against common guidelines. With modern IAM solutions as used in LEXIS, multiple organizations can federate. The AAI system itself may be distributed, providing extra resilience to failure.

After a user is authenticated, a token is used to confirm this process, to give the user rights, and also to allow services to perform requests on behalf of the user. A token provides non-tamperable proof of identity, authentication, and authorization, with relatively short expiration time and immediate revocation possibility, minimizing the risk of stolen credentials. OpenID Connect (OIDC [19]) is a prime token and interfacing standard for the interaction between services and IAM, using REST interfaces [31] and JWT tokens [32]. To select the most suitable IAM system for the LEXIS project, we analyzed current IAM open source solutions, taking into account the following main criteria: clustering/scalability, distributed/multi-site setup capabilities, disaster recovery/backup, authentication protocol, integration capabilities, and functionalities such as user/group management with role- or attribute-based access control (RBAC/ABAC). Finally, we decided to use the product Keycloak [16] which RedHat also employs in its SSO solution.

4.5.2 Platform Services vs. AAI: Separation of Concerns

The clear separation of Keycloak-based AAI and other LEXIS services (where communication happens via defined APIs) follows the security principle: "Do one thing and do it well." The user has only one identity and different access possibilities (roles) on different systems. This also enables proper auditing, allowing traceability, which is particularly important from a security perspective.

4.5.3 LEXIS DDI and IAM/AAI System

When building the LEXIS DDI, the EUDAT-iRODS system was heavily customized with an adapted plugin in order to provide proper support of Keycloak's OpenID Connect tokens. Thus, access, data privacy and data sharing permissions can be conveniently and centrally controlled. The iRODS data management software provides its own authentication and authorization solution (see [13]), and controls access with an extension to the Unix permission model (i.e. via users and groups, and permissions to own, read, and modify data in access control lists). While it allows authentication to be delegated, the iRODS user-creation process has still to be separately executed for each user in each zone's iCAT. LEXIS users are mapped to iRODS users, while LEXIS projects (see Section 4.1) are mapped to iRODS groups. When a user is created and added to a project, in each iRODS zone of LEXIS the corresponding iRODS user is created

and added to the relevant groups, and directories for their private data within the projects are set up. This is at the moment a semi-manual process (with automated progress monitoring), handled via ticket creation in a LEXIS trouble-ticket system and administrators triggering appropriate API endpoints (Section 4.6.4).

When authenticating via OpenID Connect, there are limitations to the length of the OpenID tokens that iRODS will accept (approximately 1 kB, which is an issue with Keycloak's JWT tokens). We worked around this by modifying the iRODS-OpenID plugin architecture, such that tokens are prevalidated and only a token hash transverses the iRODS core [33]. This workaround, eventually to be superseded by an upstream solution within the iRODS framework, requires changes to the client programs (e.g. iRODS Python client).

Similar challenges have arisen when connecting the LEXIS orchestration system to the LEXIS AAI. Due to the usage of OpenID Connect and SAML in different orchestrator components (see Chapter 5), the LEXIS Keycloak system had to be configured to support SAML as well.

4.6 Data Management via APIs

To facilitate automated, controlled, and secure data handling in LEXIS, one central design principle of the LEXIS platform is that data handling, manipulation, discovery, upload, and download are always initiated via REST API calls. These calls can be initiated by users via the LEXIS web portal (including a data set-management interface), or by the LEXIS Orchestration System within a workflow. The portal or orchestrator authenticates on behalf of the user to the DDI using tokens from the LEXIS AAI. In order to provide all needed functionality, APIs for metadata-based search, (meta)data upload and download (Section 4.6.1), data staging and replication (Sections 4.6.2, 4.6.3), helper functions (Section 4.6.4) and finally data (de)compression and encryption/decryption (Section 4.6.5) have been programmed and deployed. Details on these APIs and their endpoints are laid out below. The LEXIS project intends to publish current versions of API documentation and public code on its github page [34] and/or zenodo community page [35].

4.6.1 Data Search, Upload, and Download APIs

These APIs with their group of endpoints (Table 4.2) allow for basic data management operations. Data sets can be searched for substrings according to name or metadata fields (Table 4.2, lower rows). In the backend, this search uses custom Python scripts executed on the metadata of the DDI, which is retrieved with the iRODS Python client.

TABLE 4.2

Data search, upload/download REST API endpoints

Endpoint	Method	Request body[1]	Response body	Comments
/staging/download	POST	source_system, source_path	<file contents>	Download from staging.
/dataset/download	POST	internalID, access, project, push_method, compress_ method, path	<file contents>	Download from iRODS.
/dataset	POST	file, name, internalID, access, project, push_method, compress_ method, path, metadata	internalID	Create or update a data set or sub-data set.
/dataset	DELETE	internalID, access, project, path	—	Delete data set or sub-data set.
/dataset/search/ metadata	POST	<Object with any metadata>	<list of {location, metadata, eudat}>	Return metadata of matching data sets.
/dataset/search/ metadata	DELETE	<Object with any metadata>	—	Delete matching data sets.
/dataset/listing	POST	internalID, access, project, path, recursive	name, type, size, create_ type, checksum, contents	Contents provides a list with the metadata (name, type, …) for each file in a directory.

[1] push_method may be "empty" (empty data set), "directupload" (json-encoded in file parameter), "tus" (pre-uploaded using TUS protocol, with path stored in file parameter). compress_method: "file" or "zip." access: "user," "project," "public."

Further endpoints are offered to create, upload, download, and delete data sets, and to modify these data sets by adding and removing files. In order to make uploads and downloads convenient and feasible, file decompression/compression during upload/download (i.e. upload/download of zipped files, with the contents appearing in the DDI) is supported on user request, as well as resumable downloads using the TUS protocol [36]. In some cases, the API makes use of the staging API (Section 4.6.2) as a backend. Compression/decompression functionalities are being expanded by a compression/encryption API (see Section 4.6.5). For extremely large data sets, the LEXIS portal offers the users an upload/download link based on GridFTP/B2STAGE (see Section 4.7.3).

The API group contains further endpoints which allow platform users to add, remove, and modify additional information (metadata) regarding their data sets, as a basis for FAIR research data management. This information may be authorship (creator, contributor), publication (publisher, year), copyright (license), type (e.g. audiovisual, data set, image, interactive resource, model, service, software, sound, text, workflow), identifiers (DOIs, PIDs), among others. We use standard fields defined by DataCite [37] (schema: [38]), and we allow users to define their own additional metadata and metadata schemas, performing validation.

4.6.2 Staging API

Running workflows in the LEXIS federated environment, the orchestrator has to trigger data movement between different computing and data systems across different centers. Thus, input data are made available for computing tasks, and output data are safely stored back in the DDI for later reuse. The orchestrator performs these transfers (and necessary deletion of files) through the staging REST API (endpoints: see Table 4.3).

By default (as, e.g. in the API set discussed in Section 4.6.1), REST APIs are synchronous, which means the API call blocks until the task performed is either completed successfully or has failed. However, large data transfer between the different storage facilities requires time and the API must not wait until the transfer is completed. To allow asynchronous calls to the API, a distributed data scheduler connected to a message broker was thus introduced below the REST API. When a staging API call occurs, a data

TABLE 4.3

Staging REST API endpoints

Endpoint	Method	Request body	Response body	Comments
/stage	POST	source_system, source_path, target_system, target_path, metadata, job_id(optional), task_id(optional)	requestID	Stage data between different data sources.
/stage/<request_id>	GET	—	status, target_path	Returns the status of the data transfer.
/delete	DELETE	target_system, target_path, job_id(optional), task_id(optional)	—	Delete a data set.
/delete/<request_id>	GET	—	status	Returns the status of data deletion.

transfer or deletion task is pushed to the queue and the endpoint returns an ID for the orchestrator to track the status of the operation through helper endpoints (see Table 4.3).

Django [39] is used as REST API framework due to its robustness and scalability. Celery [40] was chosen as a task queue and RabbitMQ [41] as a message broker. The resulting status of transfers is saved in a PostgreSQL database.

4.6.3 Replication and PID Assignment API

Data transfer within the DDI, that is, between different iRODS zones in the federation, is handled differently than staging API transfers from or to external source/target systems. Data replication between iRODS zones in the context of LEXIS is needed to ensure data safety and provides flexibility for the orchestrator to choose the data source closest to the cloud or HPC resources.

To this end, we developed a replication API, based on EUDAT B2HANDLE [21] and B2SAFE [14]. As described in Section 4.7, B2SAFE provides a mechanism to replicate data across iRODS zones and keeps track of replica locations by using persistent identifiers (PIDs) assigned by B2HANDLE. The replication API calls B2SAFE rules to trigger data replication between two different iRODS zones. The implementation, similar to the staging API, allows for asynchronous execution.

To adhere to FAIR [15] data principles, PIDs should be assigned to data whether the data are replicated or not. PIDs allow long-term identification of data and thus serve a crucial role in data discovery. The replication API thus offers endpoints that trigger B2HANDLE to assign PIDs and check the assignment status.

4.6.4 Helper APIs

Several helper APIs have been programmed to address mostly administrative problems, some examples of which are mentioned here. A SSHFS API helps to export data sets via SSH to virtual machines in LRZ's IaaS Compute Cloud; and endpoints in the upload/download API (Section 4.6.1) allow for the permission management of newly created users and projects within the DDI.

Direct access to the DDI via Globus/GridFTP/B2STAGE for transferring large data sets eventually uses GridFTP for data access (see Section 4.6.1), which authenticates using certificates. A Gridmap File API allows users to map and unmap their certificate to their LEXIS/DDI user when needed. Also, we set up an API returning the sizes of LEXIS data sets, which users or the LEXIS orchestrator can use to judge whether data staging is a good option or computations should rather take place on the site where the data are stored.

4.6.5 Compression/Decompression/Encryption/Decryption API

Due to corporate usage, critical data have to be stored encrypted on the LEXIS DDI, that is, on upload or staging to the DDI, encryption must take place, with decryption being triggered on download/reverse staging (e.g. to HPC systems). In order to offer this functionality, and also compression/decompression of data sets to accelerate staging of many small files, an API is being programmed with endpoints for data encryption/decryption and (de)compression. To actually execute these processes within a workflow, the data are first staged (with the staging API) on NVMe or NVDIMM volumes of the LEXIS data nodes (Section 4.4.3). Then, by calls to the compression/decompression/encryption/decryption API, encryption/decryption and (de)compression is triggered and executed in a speed-optimized manner before the data are sent on. This follows and expands on the ideas behind the compression/decompression functions in our upload/download API endpoints.

4.7 Integration with EUDAT Services

The EUDAT Collaborative Data Infrastructure (CDI) [5] is the result of a growing group of organizations providing European-scale data management services. LEXIS aims at integrating with these services. Out of the large EUDAT service portfolio, B2HANDLE [21] was chosen for PID assignment, B2SAFE [14] for data safety, and B2STAGE [42] for data staging between iRODS and HPC-cluster file systems. The usage of further services in LEXIS is being investigated, envisaging – for example – a connection to B2FIND [43], the metadata-based data discovery portal of EUDAT. Below, we go through the EUDAT services LEXIS works with in some more detail.

4.7.1 EUDAT B2HANDLE

B2HANDLE [21] is EUDAT's PID service. By assigning a unique identifier, B2HANDLE helps in long-term identification of data and helps achieving the findability in a FAIR [15] data management system.

The B2HANDLE client is deployed as a Python library on all iRODS machines in LEXIS. B2HANDLE is based on the Handle system [44] and provides an interface to access it. Correspondingly, instances of the handle server system were deployed at LRZ and IT4I. The two instances are assigned to the same handle prefix and with one another through a built-in mechanism. EUDAT offers a reverse lookup servlet [45] that enables searching against the local B2HANDLE instance. The servlet shortens the time needed

to check whether a data set has already been assigned a PID, which is a prerequisite for using B2SAFE, B2STAGE, and B2FIND with that set.

In LEXIS, iRODS collections in the federated zones are assigned PIDs. B2HANDLE assigns specific iRODS metadata to these collections, reflecting the PID assignment. The collections are then findable through their identifier.

4.7.2 EUDAT B2SAFE

B2SAFE is EUDAT's core service that ensures data safety. It provides a mechanism to replicate data between different iRODS zones deployed at different centers. B2SAFE uses core iRODS rules and provides a set of high-level iRODS rules. It also takes advantage of B2HANDLE to maintain information about the location of the replicated data. Once replication is triggered, a PID is assigned to the parent data in case of absence, and a different PID is assigned to the replicated data. B2SAFE writes iRODS metadata reflecting the replication process, such as the data set PID and the parent PID for each replicated data set. In LEXIS, B2SAFE is used as a backend to the Replication REST API (Section 4.6.3).

4.7.3 EUDAT B2STAGE

B2STAGE is EUDAT's high-performance data transfer service. It provides the ability to move data between a federated iRODS-B2SAFE infrastructure and HPC resources. B2STAGE deploys a GridFTP server on top of iRODS, allowing any client supporting GridFTP to address it. Thus, big data sets can reliably be staged to and from the LEXIS DDI using B2STAGE. On the one hand, this makes B2STAGE an ideal backend to the Staging REST API for at least part of the transfers; on the other hand, controlled data upload/download possibilities via B2STAGE can be offered to the LEXIS user for large data sets.

4.8 Conclusion

In this chapter, we presented the LEXIS DDI, based on the iRODS and EUDAT services. We laid out its significance as a data backend for the LEXIS platform, a forefront infrastructure federating European top-level HPC and cloud systems and making these landscapes converge.

The hardware backend of the DDI consists of innovative systems like CEPH, IBM Spectrum Scale and burst buffers relying on the latest Intel Optane technology. iRODS and EUDAT-B2SAFE fulfil the basic requirements to serve as a data system for LEXIS, and make for a solid federation between the data centers within the LEXIS platform. The real uniqueness of the LEXIS

DDI with respect to similar data infrastructures is however in its richness in modern interfaces.

It connects to the LEXIS ecosystem with a number of well-documented HTTP-REST APIs according to modern standards. The data search API with its rich metadata search possibilities stands out as well as the staging API, allowing for convenient, asynchronously handled data transfer requests. The APIs make interfacing and automation an easy task, enabling the LEXIS orchestration system to execute workflows in a way that optimizes data transfer times.

Seen from the outside, the LEXIS DDI uses the EUDAT kit of tools and services in order to immerse itself with pan-European data infrastructures. LEXIS data are equipped with handles from EUDAT-B2HANDLE, assigned metadata to follow FAIR standards, and replicated using EUDAT-B2SAFE. On this basis, federation with more centers and EUDAT-driven data systems has been efficiently accomplished. As a next step, we plan to expose public LEXIS data via the B2FIND data search/discovery portal.

The open and modern architecture shows its strength not only in the federation context, but also when industrial applications are executed on the LEXIS platform. The DDI has an appropriate access-rights management, interfaces to meet industry standards, and is being equipped with encryption capabilities for extra security. Service quality is ensured via the LEXIS trouble-ticket system, where also service requests (e.g. for user creation) are handled. With all these characteristics, the LEXIS DDI not only helps to accomplish the LEXIS mission of converging the HPC, cloud, and big data worlds, but also drives collaboration of industry, SMEs, and scientific research in the computing and data science sectors.

Acknowledgment

This work and all contributing authors are funded/cofunded 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).

References

[1] Foster, I. and C. Kesselman, ed. 2004. *The Grid 2: Blueprint for a New Computing Infrastructure*. San Francisco: Morgan Kaufmann. https://doi.org/10.1016/B978-1-55860-933-4.X5000-7.

[2] Pennington, R. 2002. Terascale Clusters and the TeraGrid. In *Proceedings of the 6th International Conference on High-Performance Computing in Asia-Pacific Region (HPC Asia 2002)*, Bangalore.

[3] Shiers, J. 2007. The Worldwide LHC Computing Grid (Worldwide LCG). *Computer Physics Communications* 177, no. 1–2: 219–223. https://doi.org/10.1016/j.cpc.2007.02.021.

[4] Kranzlmüller, D., J. M. de Lucas, and P. Öster. 2010. The European Grid Initiative (EGI). In *Remote Instrumentation and Virtual Laboratories*, ed. F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, 61–66. Boston: Springer. https://doi.org/10.1007/978-1-4419-5597-5_6.

[5] EUDAT Ltd. 2020. EUDAT – Collaborative Data Infrastructure. www.eudat.eu (accessed Apr. 6, 2021).

[6] Solagna, P. 2014. EGI position paper for a European identity federation for researchers. https://documents.egi.eu/document/2049 (accessed Apr. 6, 2021).

[7] EUDAT Ltd. 2020. B2ACCESS – EUDAT. www.eudat.eu/services/b2access (accessed April 6, 2021).

[8] LEXIS consortium. 2020. LEXIS Project – High Performance Computing (HPC) in Europe. https://lexis-project.eu (accessed Apr. 6, 2021).

[9] Bull Atos. 2021. Ystia Suite. https://ystia.github.io (accessed Apr. 6, 2021).

[10] OASIS Open. 2013. Topology and Orchestration Specification for Cloud Applications Version 1.0 – OASIS Standard. http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html (accessed Apr. 27, 2020).

[11] Bull Atos. 2021. Alien 4 Cloud. http://alien4cloud.github.io/ (accessed Apr. 6, 2021).

[12] Svatoň, V., J. Martinovič, J. Křenek, T. Esch, and P. Tomančák. 2019. HPCas-a-Service via HEAppE Platform. In *Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019)*, Sydney. https://doi.org/10.1007/978-3-030-22354-0_26.

[13] Xu, H., T. Russell and J. Coposky, et al. 2017. *iRODS Primer 2: Integrated Rule-Oriented Data System*. Williston: Morgan & Claypool. https://doi.org/10.2200/S00760ED1V01Y201702ICR057.

[14] EUDAT Ltd. 2020. B2SAFE – EUDAT. www.eudat.eu/services/b2safe (accessed Apr. 13, 2020).

[15] Wilkinson, M. D., M. Dumontier and I. J. Aalbersberg, et al. 2019. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3: 160018. https://doi.org/10.1038/sdata.2016.18.

[16] JBoss (Red Hat, Inc.). 2021. Keycloak. www.keycloak.org (accessed Apr. 6, 2021).

[17] Kawai Y. and A. Hasan. 2010. High-Availability iRODS System (HAIRS). In *Proceedings of the iRODS User Group Meeting 2010: Policy-Based Data Management, Sharing and Preservation*, Chapel Hill. ISBN: 978-1-452813-42–44, https://irods.org/uploads/2010/Kawai-HAIRS-paper.pdf.

[18] James, J. 2015. Configuring iRODS for High Availability. https://irods.org/2015/07/configuring-irods-for-high-availability (accessed Mar. 17, 2021).

[19] Sakimura, N., J. Bradley, M. B. Jones, B. de Medeiros, and C. Mortimore. 2014. OpenID Connect Core 1.0 incorporating errata set 1. https://openid.net/specs/openid-connect-core-1_0.html (accessed Nov. 6, 2020).

[20] Cantor, S., J. Kemp, R. Philpott, and E. Maler. 2005. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf (accessed Nov. 6, 2020).
[21] EUDAT Ltd. 2020. B2HANDLE – EUDAT. www.eudat.eu/services/b2handle (accessed Apr. 13, 2021).
[22] Depuydt, J. 2015. Jensd's I/O buffer – Setup a redundant PostgreSQL database with repmgr and pgpool. http://jensd.be/591/linux/setup-a-redundant-postgresql-database-with-repmgr-and-pgpool (accessed Oct. 1, 2019).
[23] SRA OSS, Inc. 2020. pgpool Wiki. www.pgpool.net/mediawiki/index.php/Main_Page (accessed Apr. 13, 2020).
[24] 2ndQuadrant Ltd. 2020. repmgr – Replication Manager for PostgreSQL clusters. https://repmgr.org (accessed Apr. 13, 2020).
[25] Russell T. 2017. SC17 Demo: Storage Tiering. https://irods.org/2017/12/sc17-demo-storage-tiering (accessed Dec. 1, 2019).
[26] Schwan P. 2003. Lustre: Building a File System for 1,000-node Clusters. In *Proceedings of the Linux Symposium (OLS 2003)*, Ottawa. www.kernel.org/doc/ols/2003/ols2003-pages-380-386.pdf.
[27] Schmuck, F. and R. Haskin. 2002. GPFS:A Shared-Disk File System for Large Computing Clusters. In *Proceedings of the Conference on File and Storage Technologies (FAST '02) – USENIX Association*, Monterey. www.usenix.org/legacy/events/fast02/full_papers/schmuck/schmuck.pdf.
[28] Allcock, W., J. Bresnahan, R. Kettimuthu and M. Link. 2005. The Globus Striped GridFTP Framework and Server. In *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. Seattle. https://doi.org/10.1109/SC.2005.72.
[29] Weil S. A., S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. 2006. Ceph: A Scalable, High-Performance Distributed File System. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*. Seattle. ISBN: 1-931971-47-1, www.usenix.org/legacy/events/osdi06/tech/full_papers/weil/weil.pdf.
[30] NVM Express, Inc. 2016. NVM Express over Fabrics 1.0. https://nvmexpress.org/wp-content/uploads/NVMe_over_Fabrics_1_0_Gold_20160605-1.pdf (accessed Nov. 6, 2020).
[31] Richards R. 2006. Representational State Transfer (REST). In *Pro PHP XML and Web Services*, ed. R. Richards, 633–672. Berkeley: Apress. https://doi.org/10.1007/978-1-4302-0139-7_17.
[32] Jones, M., J. Bradley and N. Sakimura. 2015. JSON Web Token (JWT) – Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc7519 (accessed Apr. 6, 2021).
[33] García-Hernández, R. J., M. Golasowski. 2020. Supporting Keycloak in iRODS systems with OpenID authentication. Presentation at *CS3 2020 – Workshop on Cloud Storage Synchronization and Sharing Services (27–29 January 2020)*, Copenhagen. https://indico.cern.ch/event/854707/contributions/3681126/.
[34] LEXIS consortium. 2021. GitHub – LEXIS: Large Scale Execution for Industry & Society. https://github.com/lexis-project (accessed Apr. 6, 2021).
[35] LEXIS consortium. 2020. Zenodo Community – LEXIS project. https://zenodo.org/communities/lexis (accessed Apr. 6, 2021).

[36] Transloadit-II GmbH. 2018. tus – Open Protocol for Resumable File Uploads. https://tus.io/ (accessed Apr. 6, 2021).
[37] Brase, J. 2009. DataCite – A Global Registration Agency for Research Data. *In Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology*, Beijing. https://doi.org/10.1109/COINFO.2009.66.
[38] DataCite – International Data Citation Initiative e.V. 2021. DataCite Metadata Schema 4.4. https://schema.datacite.org (accessed Apr. 5, 2021).
[39] Django Software Foundation. 2020. Django. www.djangoproject.com (accessed Apr. 1, 2020).
[40] Celery Project. 2020. Celery: Distributed Task Queue. www.celeryproject.org (accessed Apr. 1, 2020).
[41] Pivotal Software. 2020. Messaging that just works – RabbitMQ. www.rabbitmq.com (accessed Apr. 1, 2020).
[42] EUDAT Ltd. 2020. B2STAGE – EUDAT. www.eudat.eu/services/b2stage (accessed Apr. 13, 2020).
[43] EUDAT Ltd. 2020. B2FIND – EUDAT. www.eudat.eu/services/b2find (accessed Apr. 13, 2020).
[44] Boesch, B., S. X. Sun and L. Lannom. 2003. RFC 3650 – Handle System – Internet Engineering Task Force (IETF). https://tools.ietf.org/html/rfc3650 (accessed Apr. 27, 2020).
[45] EUDAT. 2021. B2HANDLE-HandleReverseLookupServlet. https://github.com/EUDAT-B2SAFE/B2HANDLE-HRLS (accessed Feb. 23, 2021).