Leveraging High-Performance Computing and Cloud Computing with Unified Big-Data Workflows: The LEXIS Project

Stephan Hachinger, Martin Golasowski, Jan Martinovič, Mohamad Hayek, Rubén Jesús García-Hernández, Kateřina Slaninová, Marc Levrier, Alberto Scionti, Frédéric Donnat, Giacomo Vitali, Donato Magarielli, Thierry Goubier, Antonio Parodi, Andrea Parodi, Piyush Harsh, Aaron Dees, and Olivier Terzo

S. Hachinger · M. Hayek · R. J. García-Hernández
Leibniz Supercomputing Centre (LRZ) of the BAdW, Garching b.M., Germany
e-mail: hachinger@lrz.de; hayek@lrz.de; garcia@lrz.de

M. Golasowski · J. Martinovič · K. Slaninová (✉)
IT4Innovations, VŠB—Technical University of Ostrava, Ostrava, Czechia
e-mail: martin.golasowski@vsb.cz; jan.martinovic@vsb.cz; katerina.slaninova@vsb.cz

M. Levrier
Atos, Campus Teratec, Bruyères-le-Châtel, France
e-mail: marc.levrier@atos.com

A. Scionti · G. Vitali · O. Terzo
LINKS Foundation, Torino, Italy
e-mail: alberto.scionti@linksfoundation.com; giacomo.vitali@linksfoundation.com; olivier.terzo@linksfoundation.com

F. Donnat
Outpost24, Valbonne, France
e-mail: fdo@outpost24.com

D. Magarielli
AvioAero, Rivalta di Torino (TO), Italy
e-mail: donato.magarielli@avioaero.it

T. Goubier
CEA LIST, Gif-sur-Yvette, France
e-mail: thierry.goubier@cea.fr

A. Parodi · A. Parodi
CIMA Foundation, Savona, Italy
e-mail: antonio.parodi@cimafoundation.org; andrea.parodi@cimafoundation.org

P. Harsh
Cyclops Labs GmbH, Zürich, Switzerland
e-mail: piyush@cyclops-labs.io

A. Dees
Irish Centre for High-End Computing, Technology and Enterprise Campus, Dublin, Ireland
e-mail: aaron.dees@ichec.ie

E. Curry et al. (eds.), *Technologies and Applications for Big Data Value*,
https://doi.org/10.1007/978-3-030-78307-5_8

Abstract Traditional usage models of Supercomputing centres have been extended by High-Throughput Computing (HTC), High-Performance Data Analytics (HPDA) and Cloud Computing. The complexity of current compute platforms calls for solutions to simplify usage and conveniently orchestrate computing tasks. These enable also non-expert users to efficiently execute Big Data workflows. In this context, the LEXIS project ('Large-scale EXecution for Industry and Society', H2020 GA 825532, https://lexis-project.eu) sets up an orchestration platform for compute- and data-intensive workflows. Its main objective is to implement a front-end and interfaces/APIs for distributed data management and workflow orchestration. The platform uses an open-source Identity and Access Management solution and a custom billing system. The data management API allows data ingestion and staging between various infrastructures. The orchestration API allows execution of workflows specified in extended TOSCA. LEXIS uses innovative technologies like YORC and Alien4Cloud for orchestration or iRODS/EUDAT-B2SAFE for data management, accelerated by Burst Buffers. Three pilot use cases from Aeronautics Engineering, Earthquake/Tsunami Analysis, and Weather and Climate Prediction are used to test the services. On the road towards longer-term sustainability, we are expanding this user base and aiming at the immersion of more Supercomputing centres within the platform.

Keywords High performance computing · Cloud computing · Big data · Workflows · Distributed data management · Orchestration

1 High-Performance Computing, Cloud and Big Data in Science, Research and Industry—and LEXIS

In this chapter, we present how the Horizon-2020 project 'Large-scale EXecution for Industry and Society' (LEXIS[1]) addresses the 'Big Data' theme, establishing automated data analysis and simulation workflows across world-class Supercomputing and Cloud Computing centres in Europe. We relate to the technical priorities 'Data Management' and 'Data Processing Architectures' of the European BDV Strategic Research and Innovation Agenda [1], addressing horizontal ('Cloud and High Performance Computing') and vertical concerns ('Data sharing platforms', 'Cybersecurity and Trust') of the BDV Reference Model (cf. [1]). With respect to the AI, Data and Robotics Strategic Research, Innovation and Deployment Agenda [2], we present 'Systems, Methodologies, Hardware and Tools' as cross-sectorial technology enablers.

The LEXIS collaboration is made up of two of the major European scientific Supercomputing centres (IT4I/CZ and LRZ/DE), scientific and industrial/SME partners with compute- and data-intensive use cases, and industrial/SME technology

[1] https://lexis-project.eu, H2020 Grant Agreement No. 825532.

partners. It thus aims to bring the power of scientific Supercomputing to the industrial and enterprise Big Data landscape, but also to equip scientists with industry-standard Big Data tools. In these ways, it seeds knowledge transfer between science, SMEs and industry, and addresses institutions of societal relevance like governmental agencies.

When looking at the history of large-scale computing and Big Data in the last decade, it turns out that science was leading the introduction of powerful distributed computing grids (in particular LCG [3] for CERN's Large Hadron Collider), but industry drove innovations such as Infrastructure-as-a-Service (IaaS) Clouds and Big Data ecosystems. Cloud Computing services (with, e.g. Amazon as one of the pioneers), but also famous frameworks for Big Data analytics like Hadoop (e.g. [4]), were of immediate and sometimes even almost exclusive practical relevance for implementing top-notch data services. Much of the bare compute power, however, remained with scientific High-Performance Computing (HPC), and – in countries with pronounced geopolitical interest – also with the military.

In this situation, bringing ideas from the science and industry/SME worlds together is clearly a key to further development and innovation. The LEXIS project accomplishes this by co-developing an easily-usable platform for the processing of data- and compute-intensive workflows. The LEXIS platform with its ambitious ideas is backed not only by strong computing systems and by an orchestration facility but also by a Distributed Data Infrastructure immersed with the EUDAT [5, 6] (EUropean DATa) system, which will be extensively discussed in this chapter.

From the scientific HPC centres' point of view, our agenda is certainly motivated by practical problems scientists have faced with their computing projects for the last decades. Using a Supercomputer has traditionally involved a steep learning curve and hard work on scripting workflows and submitting them to a job queue. Only researchers with very long-term experience have thus been able to efficiently run extreme-scale simulations.

Clearly, this usage pattern is not practicable for industrial/SME applications and their – often shorter – life cycles. However, it also excluded a major part of all scientists – those without focus on IT – from efficient scientific computing and data analysis. Nowadays, the average scientist appreciates modern low-threshold IT offers, such as (often commercial) IaaS or container platforms, while industry has become aware of the capabilities of supercomputers and of academic developments, for example in quantum computing. Thus, we witness a sort of 'golden age' for projects such as LEXIS which make these worlds converge in order to reach new levels of optimisation.

Below, we present the LEXIS ideas, in the light of collaborations such as the Big Data Value Association (BDVA [7]) and EUDAT, and of the European computing and data landscape in general. We will first cover the vision of HPC-Cloud-Big Data convergence in LEXIS and basics of the LEXIS concept (Sect. 2) and then the LEXIS approach to Authentication/Authorisation as a prerequisite for a secure platform (Sect. 3). Section 4 extensively discusses the European data management approach for Big Data in LEXIS. We close the chapter with a description of the LEXIS Portal as a one-stop shop for our users (Sect. 5) and our conclusions (Sect. 6).

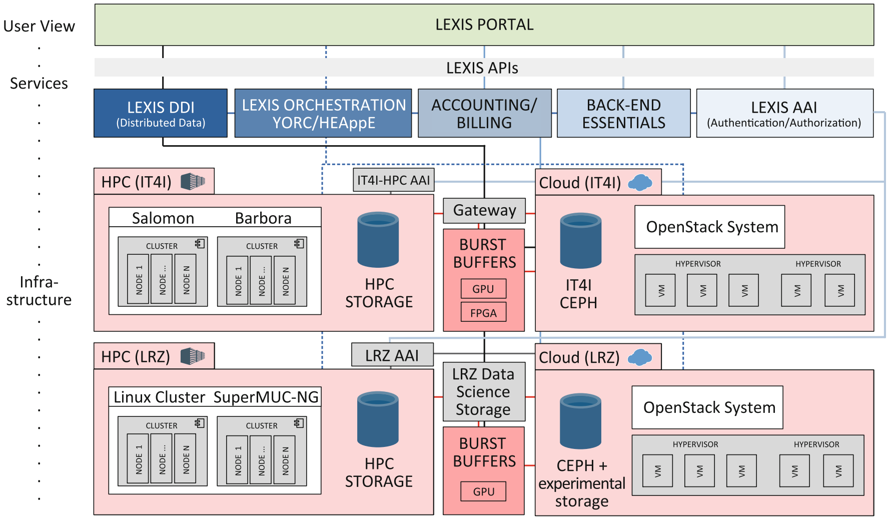


Fig. 1 Simplified scheme of the LEXIS platform: main components as of 2020 (extension to more computing and data centres is planned). The ‘back-end essentials’ box represents technical components not mentioned in this overview

2 LEXIS Basics: Integrated Data-Heavy Workflows on Cloud/HPC

2.1 LEXIS Vision

The vision of LEXIS is a distributed platform which enables industry, SMEs and scientists to leverage the most powerful European computing and data centres for their simulations, data analytics and visualisation tasks. Via a user-friendly portal with a modern REST-API[2] architecture behind it, the LEXIS Orchestration System for workflows and the LEXIS Distributed Data Infrastructure are addressed. The user uploads necessary data and software containers and specifies workflows with all computing tasks and data flows. From this point on, the Orchestration System takes care of an optimised execution on the LEXIS resources.

Figure 1 gives an overview of the federated systems within the LEXIS platform, from hardware systems (lower part), over service-layer components to APIs and the LEXIS Portal (top part). The architecture can be considered a blueprint for data processing architectures aligned with the BDVA strategy [1, Sect. 3.2]. It federates decentralised, heterogeneous resources to offer data processing services. The platform is the result of a strong initial focus on co-design, identifying available systems to be leveraged and key technologies required. Technological choices were

[2] REpresentational State Transfer Application Programming Interface.

made with a preference for state-of-the-art, open-source, extensible and sufficiently mature products. The details of this become clearer in the following parts of this book chapter, which discuss the LEXIS ecosystem from low- to high-level components.

In this section, we describe the LEXIS basics, beginning with hardware systems (Sect. 2.2), which include devices to accelerate computation (GPU and FPGA cards) and data transfer (Burst Buffers). Then, we discuss the Orchestration System (Sect. 2.3), which addresses our hybrid HPC/Cloud Computing facilities, realising a novel processing architecture for Big Data. The section is completed with a glimpse on the LEXIS Pilot use cases (Sect. 2.4), used to co-design and test the platform, and on our billing concept (Sect. 2.5) as part of a future business model.

2.2 LEXIS Hardware Resources

The LEXIS system flexibly utilises computing-time and storage grants on different back-end systems, as specified by the user.

While the LEXIS federation is planned to be constantly extended, currently the compute and data back-end resources (see Fig. 1) are contributed by two flagship Supercomputing centres: the Leibniz Supercomputing Centre (LRZ, Garching near Munich/D) and the IT4Innovations National Supercomputing Centre (IT4I, Ostrava/CZ). Systems available (cf. Fig. 1) include traditional and accelerated Supercomputing resources, on-premises compute clouds, high-end storage resources, and Burst Buffers equipped with GPUs and FPGAs.

At LRZ, 'SuperMUC-NG' (originally one of the world's top-10 HPC machines) provides 311'040 CPU cores (26.9 PFlops) and 719 TB of RAM. Compute time is granted via calls for proposals, which need to devise a promising research agenda. The smaller LRZ Linux Cluster offers (with less bureaucracy) roughly 30'000 CPU cores and 2 PFlops of compute power. This system is also used in LEXIS. Furthermore, a NVIDIA DGX-1 with eight Tesla V-100 GPUs is available for AI workloads.

At IT4I, the 'Barbora' (7'232 compute cores and 45 TB RAM) and 'Salomon' (12-core Xeon, 24'192 cores with 129 TB RAM) HPC systems are available. With some nodes accelerated by NVIDIA Tesla V100-SXM2 and Intel Xeon Phi cards, these systems altogether offer about 3 Pflops. Usage of them is subject to an open-call procedure as for SuperMUC-NG. Some millions of CPU hours have already been allocated to LEXIS in general and can readily be distributed to use cases.

The central LEXIS objective of bringing together HPC and Cloud resources in unified workflows is supported by integrating the LRZ Compute Cloud and IT4I's visualisation nodes into the platform. Altogether, these infrastructures provide more than 3'000 CPU cores for on-demand needs via on-premises OpenStack installations.

Storage resources in the ramp-up phase of LEXIS include access to 50 TB in LRZ's Data Science Storage (DSS, based on IBM Spectrum Scale, formerly known as GPFS [8]) and 150 TB in an ‘Experimental Storage System’ at LRZ, while IT4I provides 120 TB of Ceph-based [9] storage. These resources serve as back-end for the LEXIS Distributed Data Infrastructure (Sect. 4). They can be extended at any time, allocating more space in the (shared-usage) background storage of the computing centres, which is currently in the 100 PB range.

A typical issue in many data-intensive applications is the slowdown of actual data processing during in- and output of large data sets. LEXIS addresses this by flexibly inserting two ‘Burst Buffer’ systems per compute site in the data flows. Each of them offers about 10 TB of very fast NVDIMM and NVMe storage. Running the Atos ‘Smart Burst Buffer’ software, they are able to pre-fetch data or transparently cache output data. Thus, I/O is practically ‘instantaneous’ for the application, while the Burst Buffer manages the communication with the actual file system (pre-read or delayed write) in the background. In addition, the systems can reprocess data leveraging GPU and FPGA accelerator cards, and their NVDIMM/NVMe storage can be exported via NVMEoF [10] using the Atos ‘Smart Bunch of Flash’ tools.

2.3 LEXIS Orchestration

Automatising the execution of complex workflows[3] is crucial to enable more users to bring their applications onto efficient computing and data platforms. To address this challenge, the LEXIS Orchestration System uses technologies which minimise the need of users to acquire expertise outside their own domain. It provides the capabilities of composing application workflows in a simple way and of automatically running them on the most suitable set of resources (cf. Sect. 2.2). Moreover, it enables an automated, unified management of workflow steps based on different processing concepts, such as HPDA (High-Performance Data Analytics), HPC and HTC (High-Throughput Computing), fulfilling the respective infrastructural requirements.

The key difference of projects as LEXIS with respect to earlier projects on orchestration is the mixed usage of HPC and IaaS-Cloud (OpenStack) resources, and prospectively also, for example, container platforms to run tasks within one given workflow. Orchestrators that can address all this have been unavailable when designing the LEXIS architecture, and only recently, a few solutions are emerging (cf. [11]). In this context, we decided to use and co-develop Yorc (Ystia Orchestrator, [12]) to orchestrate workflows in LEXIS, and Alien4Cloud (Application LIfecycle ENablement for Cloud, [13]) as a front-end for designing workflows. This open-source system, with which LEXIS partner Atos is experienced, has since been extended to become an HPC-aware meta-orchestrator, addressing all relevant systems via plug-ins.

[3] We understand workflows – to give a simplified hint of a definition – as directed acyclic graphs with computational, pre-/post-processing, visualisation or data movement tasks as vertices here.

Alien4Cloud first helps the user to define the so-called topology for his application. This includes the hardware resources (e.g. the amount of CPU cores and RAM) and the software (e.g. frameworks or libraries) needed. Afterwards, it greatly simplifies the specification of the actual workflow (i.e. tasks and their order). Behind its drag-and-drop interface, Alien4Cloud describes all this in an extended version of TOSCA (cf. e.g. [14]). Extensible, generic application templates are provided.

Once the workflow description is generated, the back-end engine will run it on appropriate and available resources. To this end, it considers system characteristics, the user's access rights, locations where data sets reside, and custom constraints in the application template (e.g. deadlines for execution in urgent computing). LEXIS Orchestration is furthermore being equipped with dynamic scheduling capabilities, taking into account, for example, the load and availability of systems in real time.

The actual access to computing resources is mediated by the HEAppE middleware [15, 16], which serves two purposes: (i) security layer for mapping LEXIS users (cf. Sect. 3) to internal Supercomputing-centre accounts, used to actually execute jobs; (ii) sending an appropriately formatted job description (with pointers to the executables, etc.) to the workload managers of LEXIS resources. A wealth of middleware frameworks with this functionality is available on the market since grid times (cf. e.g. [17]). Clearly, HEAppE – besides providing a state-of-the-art implementation with a REST API – has the advantage that it is developed at IT4I as a project partner. Thus, it can easily be adapted, for example to provide usage data for billing (cf. Sect. 2.5).

2.4 LEXIS Pilots

Having described the basics of LEXIS orchestration, we give an impression of the first workflows ('Pilots') the platform is designed to run. These Pilot use cases have been a key for the early requirements analysis and co-design activities to create the LEXIS platform. They have been carefully selected to be sufficiently heterogeneous in order to make LEXIS generically useful.

The three initial LEXIS Pilots cover the following themes: (i) data- and compute-intensive modelling of turbo-machinery and gearbox systems in aeronautics, (ii) earthquake and tsunami data processing and simulations, which are accelerated to enable accurate real-time analysis, and (iii) weather and climate models, where massive amounts of in situ data are assimilated to improve forecasts (also predicting, e.g., flash floods). While Pilot (i) is certainly industry-centric, and (ii) is interesting for public authorities, Pilot (iii) has a broad and mixed range of applications.

Fully orchestrated workflows of the Weather and Climate Large-scale Pilot in LEXIS have already reached considerable intrinsic complexity (Fig. 2; for more details see [18]). Because of their broad range of applications, these are a prime example to elaborate somewhat upon. In the example shown in Fig. 2, the Weather and Research Forecasting (WRF) model [19] as an HPC core drives a fire risk prediction system (RISICO, cf. [18]). Likewise, hydrological risks and air quality

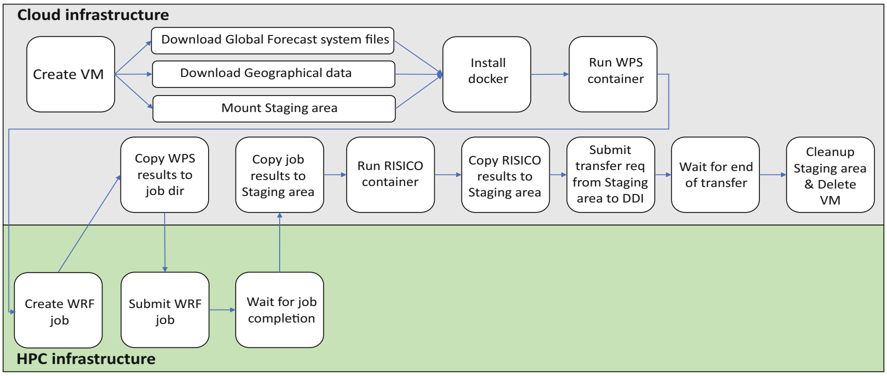

Cloud infrastructure
Create VM
Download Global Forecast system files
Download Geographical data
Mount Staging area
Install docker
Run WPS container
Copy WPS results to job dir
Copy job results to Staging area
Run RISICO container
Copy RISICO results to Staging area
Submit transfer req from Staging area to DDI
Wait for end of transfer
Cleanup Staging area & Delete VM
Create WRF job
Submit WRF job
Wait for job completion
HPC infrastructure

Fig. 2 WRF-RISICO workflow [18] from the Weather and Climate Large-scale Pilot, as presented on a SC20 conference poster (M. Hayek et al.)

can be assessed. However, we strongly target commercial applications here as well. As one example, we aim at predicting optimum sites for wind energy plants with unprecedented accuracy. Also, in collaboration with the SME NUMTECH, highly accurate modelling of agricultural conditions shall be exploited to select, for example, optimum seeding or harvesting times. With respect to previous projects (e.g. [20]), where only limited workflow automation was available, a much broader range of applications is possible without experts having to stand by or even execute steps manually. The available computing systems are optimally leveraged, as the WRF preprocessing system (WPS) and the application models run perfectly as containers on the Cloud infrastructures at LRZ and IT4I, while WRF is a classical HPC job.

Pilots (i) and (ii) will test the LEXIS platform with another variety of different application characteristics. The Earthquake and Tsunami Pilot works with additional database services and urgent computing to feed warning systems before certain deadlines. The Aeronautics Pilot boosts the performance of turbomachinery and gearbox simulations performance to make such computations part of a ‘real-time’ design process. It thus involves experiments with low-level code optimisation for GPUs attached to HPC nodes, but also quick post-processing and visualisation of simulation snapshots will play a role. Further use cases attracted through a LEXIS ‘Open Call’ complement all this and contribute to a broad validation of the platform.

2.5 *Billing for LEXIS Usage*

In order to position LEXIS as a viable innovation platform for SMEs and industry, flexible accounting and billing mechanisms are a must. The accounting process in LEXIS is designed to accommodate resources from both HPC and Cloud systems and to take into account metered consumption of data storage. Abilities to comprehensively support flat-rate or tiered pricing, as well as completely dynamic rating, charging and billing are crucial to support a sustainable LEXIS business model.

Similar to the situation in orchestration, no combined Cloud/HPC/Storage billing framework matching our requirements was known when LEXIS was initiated. Thus, the SME 'Cyclops' (CH) participates in LEXIS and enhances their successful Cyclops cloud billing system to include, for example, HPC usage (compute time) data collectors. The system with its data collectors samples resource usage in near real time. Thus, we will be able to offer paid LEXIS usage with attractive and accurate cost models.

3 Secure Identity and Access Management in LEXIS – LEXIS AAI

The Authentication and Authorisation Infrastructure (AAI) of LEXIS is the actual basis for secure access, and thus a cornerstone for the distributed computing and data platform. All LEXIS systems rely on this AAI and thus offer access control complying with industry standards.

To elaborate a bit more on this, we lay out the motivation (Sect. 3.1) for setting up a LEXIS AAI, give some technical details on our resilient solution and (Sect. 3.2) and describe the role-based access control (RBAC, Sect. 3.3) model thus implemented. The concepts follow current best practices in IT.

3.1 Why an Own LEXIS AAI?

LEXIS as a platform provides unified access to various computing and data facilities, all with their existing, operational user administration and access-rights concepts. As the European identity-provider and AAI federation landscape takes time to consolidate, LEXIS must provide access to its users in a pragmatic and secure way.

Therefore, the LEXIS partners decided, already in the early stages of co-design, to set up a simple, federated and open-standard Identity and Access Management (IAM) solution as a basis for the LEXIS AAI. Thus, a single sign on (SSO) to all parts of the LEXIS platform with the necessary convenience is provided. The LEXIS AAI integrates smoothly with the existing systems for granting access at computing/data centres: Once users authenticate via the LEXIS AAI, they use compute systems via the HEAppE middleware (cf. Sect. 2.3), which addresses the back-end systems using regular, site-specific accounts via secure mechanisms.

3.2 Realisation of LEXIS AAI Following ‘Zero-Trust’ Security Model

With its central role, the LEXIS AAI had to be built upon a reliable framework with industrial-level backing and widespread usage, but also a rich, state-of-the-art feature set. Based on a requirement analysis [21], considering also experience and prospective maintainability, the LEXIS AAI was decided to rely on the open-source IAM solution ‘Keycloak’ [22]. Keycloak constitutes the upstream of the ‘Red Hat SSO’ product. It allows for the implementation of basically any access-policy scheme (role-based or user-based access control, etc.), and for easy integration with applications using OpenID Connect [23], SAML 2.0 [24] and further authentication flows. With its further abilities, for example of delegating authentication to third-party identity providers (also Facebook, Google), it covers almost any imaginable use case.

All components of the LEXIS platform (cf. Fig. 1 in Sect. 2.1) use the central AAI via its APIs, and LEXIS users authenticate preferably via OpenID-Connect tokens. Because LEXIS has opted for a ‘zero-trust’ model, not a single service on the platform is blindly trusted by any other service. This means, each service checks the provided tokens independently against Keycloak. When tokens are forwarded as needed to back-end services, these will revalidate them.

In order to ensure high availability of the AAI service, a Keycloak cluster is run across IT4I and LRZ in ‘Cross-Datacenter-Replication’ mode. As all critical traffic between the two centres, the traffic within this cluster is encapsulated in secure channel communication (merely using an encrypted virtual private network).

Keycloak is configured to use one ‘realm’ for LEXIS identity with several ‘clients’ (in Keycloak terminology), which are the different components/services of the platform. Authorisations are configured at realm level and are then accessible at client level, allowing reusability, centralisation and simplicity of the configuration and management. If needed, additional client-level settings can be added.

Keycloak’s OpenID Connect tokens follow the JSON Web Token (JWT, [25]) standard, which consists of three parts: header, payload and signature. The verification of the signature enables a service to ensure that the token was not modified by a third party and was produced by the expected source. The content of the tokens includes the user identity and information about their granted permissions. We decided to use a unified token for all services in order to minimise load on the Keycloak service. If the need arises for the user’s permissions in one resource to be kept hidden from other resources, this concept can, however, easily be modified. Some effort (e.g. [26]) was invested to customise all systems used in LEXIS such that they support the authentication flows of Keycloak.

3.3 RBAC Matrix and Roles in LEXIS

In LEXIS, rights are granted using a role-based access control (RBAC) model. This means that all users are assigned roles, depending on their job and responsibility scope. A fundamental concept in this context is the 'LEXIS Computational Project', where each project corresponds to a group of collaborators who use a set of compute time and storage grants together. Registration of each new LEXIS user and account creation (including the role settings) are subject to an administrative verification process.

A fixed 'LEXIS RBAC matrix' defines all the access policies; that is, which role implies access to which particular LEXIS services, resources or data. When a user tries to access a service or resource, their role attribute key is checked for authorisation. The LEXIS RBAC model not only controls regular user access but also includes elevated-rights roles, for example, for project and organisation management. LEXIS systems already implementing a (more or less sophisticated) access-control mechanism were adapted such that their internal permissions consistently reflect those within the RBAC model.

This being said, all processing, visualisation, data or system-related steps on the LEXIS platform are packaged as workflows, which are being executed on behalf of the user by the LEXIS Orchestration System. These workflows are created by the user via the LEXIS portal. Already for this workflow-building process, the web portal implements a view adapted to the user's roles/rights, comprising, for example, data and systems in the user's scope. For then running the workflow in the back-end, tokens (OpenID Connect/SAML 2.0) issued to the user by the LEXIS AAI are passed through and used to log into the relevant (compute and data) services.

4 Big Data Management for Workflows – LEXIS DDI

LEXIS acts as an infrastructure provider (or 'reseller'), enabling data-heavy workflows on distributed European Supercomputing facilities. This means, it does not directly implement Big Data frameworks such as Spark (e.g. [27]) or Hadoop (e.g. [4]), but leaves it to the users to leverage optimum tools in their workflows. Thus, a prime task in LEXIS is to enable efficient data storage, access and transfer by employing a forefront distributed data management framework in a European context.

For the LEXIS 'Distributed Data Infrastructure' (DDI), we chose to adopt the EUDAT-B2SAFE solution (cf. [6]), based on the Integrated Rule-Oriented Data System (iRODS, [28]). The design is open for federation with new prospective LEXIS partners, for which installation recipes can be provided. The EUDAT ('EUropean DATa') project aims at unifying research data infrastructure, and working with their tools gives us an outstanding opportunity to transfer knowledge from academic data management to the enterprise world.

We expand on system choices, construction plan and features of the DDI below. Section 4.1 gives more details on concept and necessary interfaces of the DDI. The later subsections describe how our system matches the requirements and integrates geographically distributed storage systems (Sect. 4.2), how it handles access rights (Sect. 4.3), how it immerses in the EUDAT context (Sect. 4.4) and how the orchestrator addresses the DDI via APIs (Sect. 4.5).

4.1 Concept of Unified Data Management in LEXIS

The LEXIS DDI must enable the orchestrator and portal, and thus the authenticated users, to retrieve their LEXIS data – no matter where they are physically stored – in a unified, secure and efficient manner at all LEXIS sites. This is ensured by a system for distributed data management fulfilling the following requirements from the early co-design process:

(i) Unified access on LEXIS data in a file-system-like semantics
(ii) Reliability and redundancy
(iii) Support for diverse storage back-end systems
(iv) Support for the LEXIS AAI
(v) Support for implementing storage policies, for example selective data mirroring
(vi) Support for having metadata and persistent identifiers in the system
(vii) Support for the system to be addressed via REST APIs

With these features, the LEXIS DDI aims to be an academic/industrial data platform (IDP) facilitating data management as envisaged by the BDVA [1, Sect. 3.1]. Basic annotation with metadata shall foster semantic interoperability, and the entire data lifecycle (data taking and processing, internal re-usage with rights management, publication, etc.) is considered in the DDI design.

Within workflows, data access and transfer are to be automatically controlled by the orchestrator (cf. Sect. 2.3), by addressing the DDI via APIs. In order to save precious execution time, the orchestrator may, for example, query the physical location of input data and move compute jobs to the same computing/data centre. Likewise, it may identify and use mirror copies of the data at a proposed computing site, if the user pre-ordered his input data sets to be replicated, for example on IT4I and LRZ.

4.2 Choice of DDI System, Integration of Distributed Storage Systems

Unified access to geographically distributed storage back-ends is probably the most challenging requirement of all discussed above. The back-end systems to be used are of different technological nature and dedicated to various computing clusters, projects or purposes. Often (e.g. in the case of LRZ), the resources are operationally supported and served as a particular file system (e.g. NFS), not as bare storage.

Building the LEXIS DDI on such back-ends, leveraging a distributed file system which needs 'raw disk' access or particular back-end file systems for efficiency (e.g. Ceph [9]) is hard. Various solutions are, however, on the market to integrate existing file systems into one common data management system. Frameworks with a 'file system on file systems' approach (e.g. GlusterFS, or in a way also HDFS, cf. [29]) are usually efficient and scale well but come with a trade-off in terms of flexibility, for example in their storage policies. Also, this approach usually implies a tightly coupled system, whose behaviour in case of high WAN latencies or site failure is certainly not trivial. Thus, we rather went for a looser, middleware-based storage federation approach, whose possible performance penalties [29] are mitigated by the use of burst buffers and HPC-cluster file systems in LEXIS. Excellent open systems in this sector are, for example, iRODS, Onedata, Rucio and dCache (cf. [28, 30]). iRODS stands out through its intuitive file-system-like semantics, flexibility as for storage policies and metadata stored, and most of all through its integration in the feature-rich, European EUDAT [6] data management ecosystem and many other European projects.

Thus, we adopted for the LEXIS DDI an iRODS/EUDAT-B2SAFE based solution, which optimally matches the LEXIS requirements (cf. list/numbering in Sect. 4.1):

(i) Unified LEXIS data access: iRODS has a file-system-like view on all data, which are structured in 'data objects' (similar to files) and 'collections' ($\sim$ folders).
(ii) Reliability: can be boosted with the high-availability setup "HAIRS' [31].
(iii) Support for diverse storage back-end systems: iRODS is extremely flexible and can address any common file system, but also, for example, S3 buckets.
(iv) Support for the LEXIS AAI: here, iRODS has an iRODS-OpenID plugin which we modified to make it work with Keycloak [26].
(v) Support for implementing storage and mirroring policies: this is a traditional strength of the iRODS rule engine.
(vi) iRODS supports storing custom metadata including persistent identifiers.
(vii) The various iRODS clients available (e.g. command-line, Java & Python clients) facilitate the programming of custom 'LEXIS Data System REST APIs'.

Different geographical sites can be loosely bound in iRODS via a 'zone federation' mechanism, which enables transparent data access between the zones, while they are operated independently. Figure 3 gives an overview over the LEXIS iRODS federation, in which each major data/compute site (currently IT4I and LRZ) has its own iRODS zone. To move data between zones/centres, a simple copy command is sufficient, as the zones just appear as different top-level directories. On (transparent) data access, iRODS automatically handles the necessary data transfers with an internal SSL-secured protocol allowing multiple parallel data streams.

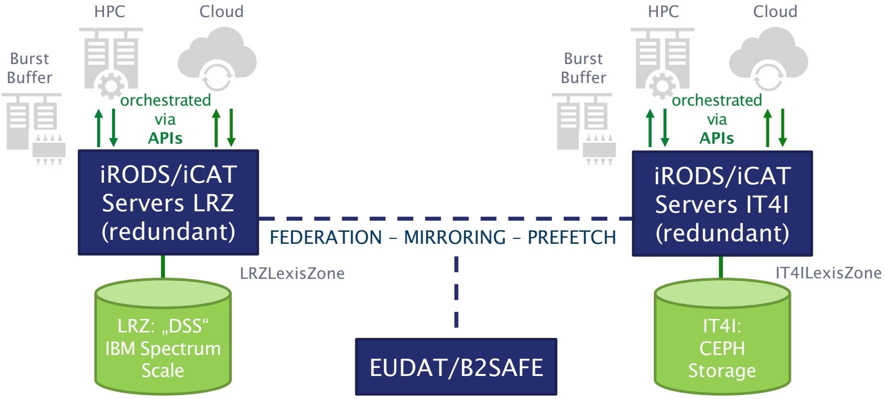


Fig. 3 LEXIS DDI federation. The zone names (IT4ILexisZone and LRZLexisZone) refer to the two computing/data centres federated in the LEXIS DDI by end of 2020. Two main operational back-end storage systems (LRZ’s ‘Data Science Storage’ or ‘DSS’, and IT4I’s Ceph system – cf. Sect. 2.2) are illustrated, as well as the transfer possibilities to Cloud and HPC infrastructure via API calls (cf. Sect. 4.5). Such transfers may leverage the LEXIS Burst Buffers (cf. Sect. 2.2)

Each zone in iRODS has a so-called ‘iCAT’ or ‘provider’ iRODS server (cf. [28]) which holds the information on the stored data, permissions, local resources and all other necessary zone-specific information in a database. In case of major problems in one zone (e.g. long-term power outage), the rest of the DDI infrastructure thus remains operational. In addition, the iRODS zones in LEXIS are each set up with a redundant iCAT (cf. [31]), using also a redundant PostgreSQL database back-end with a configuration based on repmgr and pgpool (cf. [32]).

Data mirroring between different zones, as optionally offered in LEXIS to increase resiliency and accelerate immediate data access from different centres, is implemented by the EUDAT-B2SAFE extension (cf. [6]) for iRODS. The DDI thus provides functionality to request replication on different granularity levels, for example by LEXIS Computational Project or by iRODS collection. B2SAFE then helps to set appropriate replication rules for iRODS; that is, it leverages the ability of iRODS to execute rules as a sort of ‘event handlers’ at so-called policy enforcement points (PEPs). This effectively means that, for example after storing a file in the LEXIS DDI, an arbitrary rule script (written, e.g. in Python and acting on iRODS or also at the operating-system level) can automatically be executed in order to implement data management policies (also beyond B2SAFE-related rules).

4.3 Reflecting AAI Roles in the DDI

Section 3.3 mentioned that LEXIS systems with own access-control mechanisms are set up such as to reflect the LEXIS AAI settings. The iCAT user databases on all sites are thus mirrored from the LEXIS AAI, and iRODS groups are used to implement LEXIS roles, in particular the membership or administrator role of a person in a LEXIS Computational Project (cf. Sect. 3). Actual access rights (for users/groups) are set via the iRODS access control lists.

However, also the directory (or more precisely, iRODS ‘collection’) structure itself of our DDI reflects project memberships and privacy levels of different data sets. The collection structure of the DDI starts with the iRODS zone (currently ‘/IT4ILexisZone’ and ‘/LRZLexisZone’). Three collections then exist on the next level, which are named ‘user’, ‘project’ and ‘public’, for data sets which can be accessed only by the user, by all the members of a project or by everybody. At the second level in each of these, collections for each project exist. The third level in ‘user’ and ‘project’ contains a collection for each user to write his data sets into. Each data set is automatically stored in a collection named according to a unique identifier. A project administrator can delete project data sets or publish them by moving them to the public collection hierarchy.

All this is implemented by automatically setting up the iRODS access rights (and appropriate inheritance flags) at creation of LEXIS Computational Projects and users as part of the necessary administrative process.

4.4 Immersion in the European ‘FAIR’ Research Data Landscape

Having taken care of security and privacy where needed, the next most important aspect in modern Research Data Management is controlled data sharing and reuse. In science, this reflects in the ‘FAIR’ principles [33], also cited by the BDVA [1, Sect. 2.5.2]: Data should be findable, accessible, interoperable and reusable. Even in a context of embargoed or secret enterprise data, companies can strongly profit from a basic ‘FAIR’ implementation, facilitating internal data sharing and reuse. Such an implementation relies on the assignment of persistent identifiers (PIDs) to data, the availability of a basic description of the data set (i.e. metadata) and clearly actual possibilities of data transfer.

In LEXIS, these prerequisites are largely addressed with the immersion of the DDI in the EUDAT [5, 6] ecosystem. EUDAT calls itself a collaborative data infrastructure (‘CDI’) for European research and builds its software and services along this line.

Besides EUDAT-B2SAFE (cf. Sect. 4.2), we use the EUDAT-B2HANDLE PID service (cf. [6]) in LEXIS.

Metadata, such as PIDs, but also further basic information (e.g. data set contributors, creation dates or description) are then stored directly in the Attribute-Value-Unit store for each data object and collection in iRODS. Thus, we practically cover the Dublin Core Simple and DataCite metadata schemes (cf. [34, 35]) for each LEXIS data product, enabling us to later make LEXIS data findable via research data search engines (e.g. EUDAT-B2FIND, cf. [6]) on user request.

4.5 APIs of the LEXIS DDI, and Data Transfer Within Workflows

Usage of the LEXIS DDI, be it via the LEXIS Portal (Sect. 5) or other systems of the LEXIS platform, is relying on dedicated REST APIs. These DDI APIs, with standard JSON interfaces, serve to sanitise the DDI usage pattern and thus make data manageable within an automation/orchestration context. Connection to the APIs is secured by the use of HTTPS and Keycloak Bearer tokens. Figure 4 illustrates how the DDI is thus immersed in the LEXIS ecosystem. API specifications and Swagger documentation will be released within the release cycle of the LEXIS platform.

Besides a (meta-)data search API, and many ‘smaller’ APIs (e.g. for user/rights management), the LEXIS DDI provides a REST API for data staging, which is of immediate importance for our workflows and shall thus be discussed in some detail.

Although data in the LEXIS DDI are available at all participating centres, actual compute jobs on HPC clusters normally require input and output data to be (temporarily) stored on a bare, efficient parallel file system attached to the cluster. Within a given workflow, the orchestrator thus addresses the staging API and automatically manages data movement between the different systems and the DDI as required. This includes moving input, intermediate and output files.

Performing data transfer takes time, necessitating an asynchronous and non-blocking solution for execution of staging API requests. To this purpose, the API uses – behind its front-end – a distributed task queue connected to a broker (cf. the design of [36]). At request submission, the API returns a request ID to the orchestrator. With this ID, the status of the transfer task (in progress, done, failed) can be queried via a separate API endpoint.

On the Orchestration-System side, two TOSCA components have been defined and added to the Ystia/Alien4Cloud catalogue (cf. Sect. 2.3) for transferring data to and from a computing system when executing a task (CopyToJob, CopyFromJob).

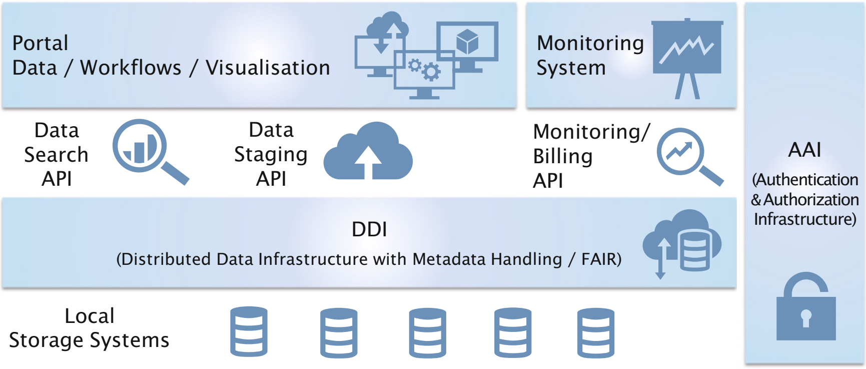

Portal
Data / Workflows / Visualisation
Monitoring System
Data Search API
Data Staging API
Monitoring/ Billing API
AAI
(Authentication & Authorization Infrastructure)
DDI
(Distributed Data Infrastructure with Metadata Handling / FAIR)
Local Storage Systems

Fig. 4 Immersion of the Distributed Data Infrastructure in the LEXIS ecosystem via its most important APIs (middle part of figure)

These components are associated to a HEAppE job component in the workflow by means of a relationship, which provides attribute values required for the necessary data transfers (e.g., source and target directories). The attribute values are then passed to the staging API to initiate the concrete transfer.

Under the hood, the back-end of the staging API is able to perform transfers using a variety of mechanisms, including those common in the world of scientific computing and HPC. It 'speaks' B2SAFE/GridFTP [6, 37], GLOBUS [38] and SCP/SFTP/SSHFS (e.g. [39]) and chooses the most efficient possibility. In the current optimisation phase of the DDI, we are beginning to regularly benchmark point-to-point speeds between all LEXIS source and target data systems. Thus, we will eliminate bottlenecks and allow the orchestrator to optimise data-transfer paths.

5 The LEXIS Portal: A User-friendly Entry Point to the 'World of HPC/Cloud/Big Data'

The ability to attract a large number of customers to the LEXIS platform mainly depends on the user-friendliness of the system. The LEXIS Portal thus plays a crucial role in lowering the barrier for SMEs to use HPC and cloud resources for solving their Big Data processing needs. It serves as a one-stop-shop and easy entry point to the entire LEXIS platform. Thus, the user does not have to deal with details of our complex infrastructure with its world-class compute and data handling capabilities.

5.1 Portal: Concept and Basic Portal Capabilities

The LEXIS Portal is highly modular by design. It integrates in a plug-and-play manner with the LEXIS DDI, Orchestration System, accounting and billing engine and AAI solution. It gives secure, role-based access to federated resources at multiple computing/data centres. The portal supports (but is not limited to) the following capabilities:

- Registration of users and organisations, and log in
- Creation and management of LEXIS Computational Projects (including addition/deletion of users)
- Requesting access to resources; view of available resources
- View of public and private data sets; data set upload
- Creation of LEXIS workflows, running of workflows as a 'LEXIS job'
- Monitoring and output retrieval for LEXIS jobs
- View of consumption of resources and billing status

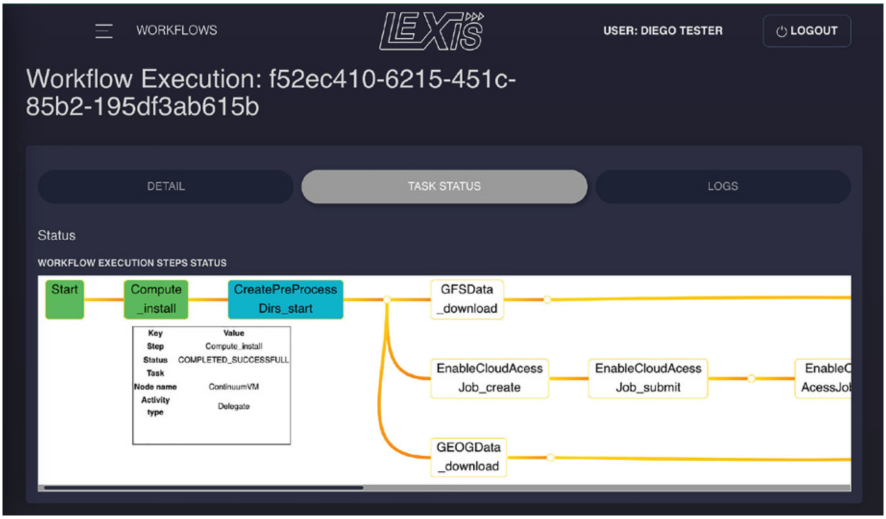
In addition, the portal back-end tracks the relationship of centre-specific HPC and cloud computing grants to the organisations and Computational Projects of LEXIS users.

The development strategy of the Portal involves an Agile methodology, however with H2020-compatible project planning for the main directions. Given the unique LEXIS requirements, we are implementing the portal from grounds up, instead of reusing existing portal frameworks. The portal front-end uses React [40], while back-end services are written entirely in Go [41], following API-driven best practices.

5.2 *Workflow and Data Management and Visualisation via the Portal*

Leveraging the experience with the Alien4Cloud UI [13] for creating workflows, the LEXIS Portal will implement an interface for workflow creation, management and monitoring. A prototype of the latter interface, focusing on part of a Weather and Climate Large-Scale Pilot workflow (cf. Sect. 2.4), is showcased in Fig. 5.

Likewise, the portal provides easy in-browser capabilities to upload new data sets (including resumable uploads based on ‘tus’ [42]), to find data sets (also by their metadata) and to modify their content. Links to high-bandwidth, out-of-browser data-transfer options for large data sets, such as GridFTP/B2STAGE endpoints of the LEXIS DDI, are provided as well. Output data of user workflows



Workflow Execution: f52ec410-6215-451c-85b2-195df3ab615b

DETAIL
TASK STATUS
LOGS
Status
WORKFLOW EXECUTION STEPS STATUS
Start
Compute _install
CreatePreProcess Dirs_start
GFSData _download
EnableCloudAcess Job_create
EnableCloudAcess Job_submit
EnableC AcessJo
GEOGData _download

Key	Value
Step	Compute_install
Status	COMPLETED_SUCCESSFULL
Task	
Node name	ContinuumVM
Activity type	Delegate

Fig. 5 Prototype of the LEXIS web portal with its workflow-monitoring interface activated. The workflow shown is at the step ‘CreatePreProcess Dirs_start’

are automatically registered – with basic metadata – as data products in the LEXIS DDI and can thus be conveniently retrieved via the Portal, or also used as an input for other workflows.

LEXIS will also provide advanced data-visualisation facilities, with the Portal as an entry point. Besides offering resource-friendly in-browser visualisation, the user can also be guided to powerful remote-visualisation systems of the LEXIS centres.

6 Conclusions

In this contribution, we presented LEXIS as a versatile and high-performance Cloud/HPC/Big Data workflow platform, focusing on its EUDAT-based Distributed Data Infrastructure and federation aspects. The LEXIS H2020 project, producing the platform, creates unique opportunities for knowledge transfer between the scientific and industrial IT communities treating Big Data. It enables industrial companies and SMEs to leverage the best European data and Supercomputing infrastructures from academia, while science can profit from applying industrial techniques, be it, for example, in the ‘Cloud-Native’ or Service Management sectors.

Right from the beginning of the project, the platform was implemented within a co-design framework. We strongly targeted the practical requirements of three representative ‘Pilot’ use cases in aeronautics engineering, earthquake/tsunami prediction and weather modelling. The pilot simulations, such as the weather models, are of concrete societal and commercial use, for example for the selection of wind energy sites. Despite this practical orientation, the project has managed to consistently follow modern, API-based and secure service design principles.

As more use cases are attracted, the platform is broadly validated, usability will be optimised, and collaboration with European users and projects can be established. This will put all components to a test, including the iRODS systems of our EUDAT-based data infrastructure, which takes care of transparent data transfers within workflows and serves the users for managing their data via the LEXIS Portal. As a result, the LEXIS platform will be optimised to reliably and efficiently execute an entire spectrum of workloads, including visualisation and GPU- or even FPGA-accelerated tasks. Orchestrated LEXIS workflows will thus efficiently combine different computing paradigms (HPC, HTC, Cloud) and analysis methods (classical modelling, AI, HPDA).

We are looking forward to extend the LEXIS federation to more computing and data sites, and software necessary to join the platform will be conveniently packaged. With this open approach, we will continue to push towards a convergence of industrial and academic data science in Europe and towards a convergence of the HPC, Cloud and Big Data ecosystems on the strongest European compute systems. Enlarging the federated LEXIS platform and making it sustainable, besides work on novel functionalities, is the key focus of the project in its second half.

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 work at IT4I is supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project 'e-Infrastructure CZ – LM2018140'. Likewise, we gratefully acknowledge the use of the computing facilities at LRZ and IT4I, as mentioned in Sect. 2.2. We thank various colleagues from EUDAT for very kind help and good collaboration.

References

1. Zillner, S., Curry, E., Metzger, A., Auer, S., Seidl, R. (Eds.) (2017). European big data value strategic research and innovation agenda. https://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf. Cited 6 Nov 2020
2. Zillner, S., et al. (Eds.) (2020). Strategic research, innovation and deployment agenda—AI, data and robotics partnership. Third Release. https://ai-data-robotics-partnership.eu/wp-content/uploads/2020/09/AI-Data-Robotics-Partnership-SRIDA-V3.0.pdf. Cited 1 Feb 2021
3. Shiers, J. (2007). The worldwide LHC computing grid (worldwide LCG). *Computer Physics Communications, 177*(1), 219–223.
4. Venner, J., Wadkar, S., & Siddalingaiah, M. (2014). Pro apache hadoop, 2nd ed. Expert's voice in big data. Apress, Berkeley, CA. doi:10.1007/978-1-4302-4864-4
5. Lecarpentier, D., Wittenburg, P., Elbers, W., Michelini, A., Kanso, R., Coveney, P. V., & Baxter, R. (2013). EUDAT: A new cross-disciplinary data infrastructure for science. *International Journal of Digital Curation, 8*(1), 279–287.
6. EUDAT Collaborative Data Infrastructure (2020). EUDAT—research data services, expertise & technology solutions. https://www.eudat.eu. Cited 6 Nov 2020
7. The Big Data Value Association (2020). BDVA. https://www.bdva.eu/. Cited 6 Nov 2020
8. Schmuck, F. B., & Haskin, R. L. (2002). GPFS: a shared-disk file system for large computing clusters. In Proceedings of the conference on file and storage technologies, FAST '02 (pp. 231–244). USENIX Association, US.
9. Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D. E., & Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system. In Proceedings of the 7th symposium on operating systems design and implementation OSDI '06 (pp. 307–320). USENIX Association, US.
10. NVM Express, Inc. (2016). NVM express over fabrics 1.0. https://nvmexpress.org/wp-content/uploads/NVMe_over_Fabrics_1_0_Gold_20160605.pdf. Cited 6 Nov 2020.
11. Colonnelli, I., Cantalupo, B., Merelli, I., & Aldinucci, M. (2020). Streamflow: Cross-breeding cloud with HPC. *IEEE Transactions on Emerging Topics in Computing*. In print. doi:10.1109/TETC.2020.3019202.
12. Bull/Atos (2020). Ystia Suite. https://ystia.github.io. Cited 6 Nov 2020.
13. FastConnect, Bull/Atos (2020). Alien 4 Cloud. http://alien4cloud.github.io. Cited 6 Nov 2020.
14. Brogi, A., Soldani, J., & Wang, P. (2014). TOSCA in a Nutshell: Promises and perspectives. In Villari, M., Zimmermann, W., Lau, K. K. (Eds.), Service-oriented and cloud computing (pp. 171–186). Springer, Heidelberg. doi:10.1007/978-3-662-44879-3_13
15. Svaton, V. (2020). Home · Wiki · ADAS/HEAppE/Middleware · GitLab. http://heappe.eu. Cited 6 Nov 2020.
16. Svaton, V., Martinovic, J., Krenek, J., Esch, T., & Tomancak, P. (2019). HPC-as-a-service via HEAppE platform. In Barolli, L., Hussain, F. K., Ikeda, M. (Eds.), CISIS 2019, advances in intelligent systems and computing, vol. 993 (pp. 280–293). Springer, Cham. doi:10.1007/978-3-030-22354-0_26

17. Imamagic, E., & Ferrari, T. (2014). EGI grid middleware and distributed computing infrastructures integration. In Proceedings of the international symposium on grids and clouds (ISGC) 2013—PoS, vol. 179 (010). doi:10.22323/1.179.0010.
18. Parodi, A., et al. (2020). LEXIS weather and climate large-scale pilot. In Barolli, L., Poniszewska-Maranda, A., Enokido, T. (Eds.), CISIS 2020, advances in intelligent systems and computing, vol. 1194 (pp. 267–277). Springer, Cham. doi:10.1007/978-3-030-50454-0_25.
19. Powers, J. G., et al. (2017). The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society, 98*(8), 1717–1737.
20. Parodi, A., et al. (2017). DRIHM (2US): An e-science environment for hydrometeorological research on high-impact weather events. *Bulletin of the American Meteorological Society, 98*(10), 2149–2166.
21. LEXIS project (2019). Deliverable 4.1: Analysis of mechanisms for securing federated infrastructure. https://cordis.europa.eu/project/id/825532/results. Cited 7 Jan 2021.
22. JBoss (Red Hat Inc.), Keycloak Community (2020). Keycloak. https://www.keycloak.org/. Cited 6 Nov 2020.
23. Sakimura, N., Bradley, J., Jones, M. B., de Medeiros, B., & Mortimore, C. (2014). OpenID connect core 1.0 incorporating errata set 1. https://openid.net/specs/openid-connect-core-1_0.html. Cited 6 Nov 2020.
24. Cantor, S., Kemp, J., Philpott, R., & Maler, E. (2005). Assertions and protocols for the OASIS security assertion markup language (SAML) V2.0. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf. Cited 6 Nov 2020.
25. Jones, M. B., Bradley, J., & Sakimura, N. (2015). RFC 7519 – JSON web token (JWT). https://tools.ietf.org/html/rfc7519. Cited 6 Nov 2020.
26. García-Hernández, R. J., & Golasowski, M. (2020). Supporting Keycloak in iRODS systems with OpenID authentication. Presented at CS3—workshop on cloud storage synchronization and sharing services. https://indico.cern.ch/event/854707/contributions/3681126. Cited 6 Nov 2020.
27. Zaharia, M., et al. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM, 59*(11), 56–65
28. Xu, H., Russell, T., Coposky, J., Rajasekar, A., Moore, R., de Torcy, A., Wan, M., Shroeder, W., & Chen, S. Y. (2017). iRODS primer 2: Integrated rule-oriented data system. In Morgan and Claypool Publishers, Williston, VT. doi:10.2200/S00760ED1V01Y201702ICR057.
29. Depardon, B., Le Mahec, G., & Séguin, C. (2013). Analysis of six distributed file systems. Report hal-00789086. https://hal.inria.fr/hal-00789086. Cited 7 Jan 2021.
30. Fuhrmann, P., Antonacci, M., Donvito, G., Keeble, O., & Millar, P. (2018). Smart policy driven data management and data federations. In Proceedings of the international symposium on grids and clouds (ISGC) 2018 in conjunction with frontiers in computational drug discovery—PoS, vol. 327 (001). doi:10.22323/1.327.0001.
31. Kawai, Y., & Hasan, A. (2010). High-availability iRODS system (HAIRS). In Proceedings of the iRODS user group meeting 2010: Policy-based data management, sharing and preservation. Chapel Hill, NC.
32. Depuydt, J. (2015). Setup a redundant PostgreSQL database with repmgr and pgpool | Jensd's I/O buffer. http://jensd.be/591/linux/setup-a-redundant-postgresql-database-with-repmgr-and-pgpool. Cited 6 Nov 2020.
33. Wilkinson, M. D., et al. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data, 3*, 1–9.
34. Apps, A., & MacIntyre, R. (2000). Dublin core metadata for electronic journals. In Borbinha, J., Baker, T. (Eds.), Research and advanced technology for digital libraries (pp. 93–102). Springer, Berlin. doi:10.1007/3-540-45268-0_9.
35. Starr, J., & Gastl, A. (2011). isCitedBy: A metadata scheme for DataCite. *D-Lib Magazine, 17*(1–2). doi:10.1045/january2011-starr.
36. Roberts, A. M., Wong, A. K., Fisk, I., & Troyanskaya, O. (2016). GIANT API: An application programming interface for functional genomics. *Nucleic Acids Research, 44*, W587–W592.

37. Allcock, W., Bresnahan, J., Kettimuthu, R., & Link, M. (2005). The globus striped GridFTP framework and server. In SC '05: Proceedings of the 2005 ACM/IEEE conference on supercomputing (pp. 54–54). doi:10.1109/SC.2005.72.
38. Foster, I. (2011). Globus online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing, 15*, 70–73.
39. Thelin, J. (2011). Accessing Remote Files Easily and Securely. Linux Journal. https://www.linuxjournal.com/content/accessing-remote-files-easy-and-secure. Cited 6 Nov 2020.
40. Facebook Inc., React contributors (2020). React—A JavaScript library for building user interfaces. http://reactjs.org. Cited 6 Nov 2020.
41. Google, Go Contributors (2020). The go programming language specification—the go programming language. https://golang.org/ref/spec. Cited 6 Nov 2020.
42. Kleidl, M., Transloadit, tus Collaboration (2020). tus—resumable file uploads. https://tus.io/. Cited 6 Nov 2020.