Large-scale EXecution for Industry & Society

Deliverable D3.6

Data Flow Optimisation and Data System Core

Co-funded by the Horizon 2020 Framework Programme of the European Union
Grant Agreement Number 825532
ICT-11-2018-2019 (IA - Innovation Action)

DELIVERABLE ID | TITLE	D3.6 | Data Flow Optimisation and Data System Core
RESPONSIBLE AUTHOR	Emanuele Danovaro (ECMWF)
WORKPACKAGE ID | TITLE	WP3 | LEXIS Data System
WORKPACKAGE LEADER	LRZ
DATE OF DELIVERY (CONTRACTUAL)	31/12/2021 (M36)
DATE OF DELIVERY (SUBMITTED)	30/12/2021 (M36)
VERSION | STATUS	V1.0 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	ECMWF; LRZ
INTERNAL REVIEW	Lukáš Vojáček (IT4I); Natalja Rakowsky (AWI)

Project Coordinator: Dr. Jan Martinovič – IT4Innovations, VSB – Technical University of Ostrava
E-mail: jan.martinovic@vsb.cz, **Phone:** +420 597 329 598, **Web:** https://lexis-project.eu

DOCUMENT VERSION

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Table of content Structure and assignment of subtasks LRZ	15/10/2021 15/10/2021	Emanuele Danovaro (ECMWF), Stephan Hachinger (LRZ) Stephan Hachinger (LRZ)
0.2	Inserted glossary, various LRZ and ECMWF contributions Polishing	24/11/2021-10/12/2021 13/12/2021	Nicolau Manubens (ECMWF), LRZ LEXIS Team Emanuele Danovaro (ECMWF), Stephan Hachinger (LRZ)
0.3	Response to review and polishing for final-check version.	20/12/2021-28/12/2021	Emanuele Danovaro (ECMWF), Stephan Hachinger (LRZ)
1.0	Final check of the deliverable	30/12/2021	Jan Martinovič, Kateřina Slaninová (IT4I)

GLOSSARY

ACRONYM	DESCRIPTION
AAI	Authentication and Authorization Infrastructure
API	Application Programming Interface
AWS	Amazon Web Services (for Virtual Machine hosting and more)
BB	Burst Buffer
DDI	Distributed Data Infrastructure
EUDAT	EURopean DATa Collaborative Data Infrastructure (cf. [2] and https://www.eudat.eu)
FAIR	Findable, accessible, interoperable, reusable – the currently most popular paradigm for Research Data Management.
FPGA	Field-Programmable Grid Array (in the context of LEXIS, we mean accelerator cards, possibly with network connectivity, equipped with FPGAs)
GPFS	General Parallel File System (by IBM)
HEAPPE	High-End Application Execution Middleware (https://heappe.eu)
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
IAAS	Infrastructure-as-a-Service, usual denomination for a cloud-service on which entire Virtual Machines can be deployed by the user
IRODS	Integrated Rule-Oriented Data System (https://irods.org/)
JSON	JavaScript Object Notation
NFS	Network File System

NVME	NVM (non-volatile Memory) Express, usually used as interface to SSDs
NVMEOF	NVMe-Over-Fabrics
NVRAM	Non-Volatile Random Access Memory
PID	Persistent Identifier
REST	Representational State Transfer
SAS	Serial attached SCSI (see also SCSI)
SBB	Smart Burst Buffer (Smart BB), Atos product
SBF	Smart Bunch of Flash, Atos product based on NVMeoF
SCSI	Small Computer System Interface (standard for connecting storage and other devices)
SSD	Solid State Drive
SSH	Secure Shell
VM	Virtual Machine
VPN	Virtual Private Network
V100	NVIDIA Tesla V100 Graphics Card with Volta GV100 GPU
WCDA	Weather and Climate Data API
WRF	Weather Research and Forecasting (a popular weather/climate simulation code)

TABLE OF PARTNERS

ACRONYM	PARTNER
Avio Aero	GE AVIO SRL
Atos	BULL SAS
AWI	ALFRED WEGENER INSTITUT HELMHOLTZ ZENTRUM FUR POLAR UND MEERESFORSCHUNG
BLABS	BAYNCORE LABS LIMITED
CEA	COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES
CIMA	CENTRO INTERNAZIONALE IN MONITORAGGIO AMBIENTALE - FONDAZIONE CIMA
CYC	CYCLOPS LABS GMBH
ECMWF	EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS
EURAXENT	MARC DERQUENNES
GFZ	HELMHOLTZ ZENTRUM POTSDAM DEUTSCHESGEOFORSCHUNGSZENTRUM GFZ
ICHEC	NATIONAL UNIVERSITY OF IRELAND GALWAY / Irish Centre for High-End Computing
IT4I	VYSOKA SKOLA BANSKA - TECHNICKA UNIVERZITA OSTRAVA / IT4Innovations National Supercomputing Centre
ITHACA	ASSOCIAZIONE ITHACA
LINKS	FONDAZIONE LINKS / ISTITUTO SUPERIORE MARIO BOELLA ISMB
LRZ	BAYERISCHE AKADEMIE DER WISSENSCHAFTEN / Leibniz Rechenzentrum der BAdW
NUM	NUMTECH
O24	OUTPOST 24 FRANCE
TESEO	TESEO SPA TECNOLOGIE E SISTEMI ELETTRONICI ED OTTICI

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Deliverable D3.6 provides an update to earlier descriptions of the infrastructure core of the LEXIS data system, as developed in Tasks 3.2 and 3.3. This core consists of the iRODS/EUDAT [1, 2] based Data Management system used to build the LEXIS Distributed Data Infrastructure (DDI), as well as the DDI Application Programming Interfaces (APIs) exclusively used for handling data and addressing the system. Although developed in the scope of WP7, the LEXIS Data System also includes the WCDA (Weather and Climate Data API) as a very important component, which is reflected in this deliverable as well.

Position of the deliverable in the whole project context

Targeted as discussed above, Deliverable D3.6 discusses the systems of the LEXIS Distributed Data Infrastructure (DDI) in their final status, and updates Deliverables D3.1 [3], D3.2 [4] and D3.3 [5]. Deployment locations and the relation of LEXIS Data System components to LEXIS modules (cf. Deliverable D2.6 [6]) are discussed in the short D3.5 report [7], which presents the pure infrastructure, while the present deliverable covers system functionality.

Besides an updated system description, Deliverable D3.6 however covers a point even more interesting from a Research & Development point of view: data flow benchmarking and optimisation, which mostly took place in the later parts of the LEXIS project within the scope of Task 3.5. Optimisations on the DDI but also on other parts of the data flow have enabled us to efficiently support the LEXIS pilots. Thus, this deliverable is an important outcome of Task 3.5 within WP3, even if it summarizes the effort spent within Tasks 3.1, 3.2 and 3.3 as well. The monitoring system (Task 3.4) has been a somewhat separate component (fundamental also for Task 3.5, though), and has been discussed separately in Deliverable D3.4 [8].

Together with Deliverable D3.5 [7], this deliverable confirms the conclusion of the works required for reaching Milestones MS6 “Final version of LEXIS technologies” and MS8 “Final integration and LEXIS technologies validation” with regards to the DDI and WCDA: The LEXIS Data System is completely developed and integrated with the Orchestration System. The DDI systems have been successfully deployed at the IT4I, LRZ and ICHEC data centres.

Description of the deliverable

Besides Introduction and Summary, this deliverable is composed by two main Sections:

- Section 2 describes the LEXIS System Data Core with Section 2.1 focusing on the hardware/infrastructure components and Section 2.2 discussing the software components used by the LEXIS DDI, including custom components developed by LEXIS partners. Moreover, in Section 2.2.4 we describe the authentication mechanism and the integration with LEXIS Authentication and Authorization Infrastructure (AAI).
- Section 3 describes optimisations in the data flow handled by LEXIS DDI. It is organized in two Subsections: Section 3.1 analyses the modules for performance monitoring at iRODS level and by using high-level APIs, while Section 3.2 focuses on general-purpose optimization (i.e. data compression) and pilot-specific optimization strategies.

1 INTRODUCTION

The LEXIS Data System has been built in order to present a unified view on distributed data in LEXIS [9]. While the underlying iRODS/EUDAT [1, 2] system facilitates the unified view on data and the storage of metadata, Application Programming Interfaces (APIs) on top of this system allow for an integration with the LEXIS platform. The data system thus allows for automatised data handling and data transfer handling within LEXIS workflows, and for the longer-term management of input and output data according to state-of-the-art principles. Details of this concept and the research on it can be found in [10, 11] and are illustrated in this deliverable, where the final status of the LEXIS Data System is discussed. Besides a generic data management component, the LEXIS Distributed Data Infrastructure (DDI), the LEXIS Data System comprises also the LEXIS WCDA (Weather and Climate Data API) as a high-performance data store for curated weather and climate data within WP7 workflows and other use cases on this topic.

The development of the systems mentioned above in terms of functional features has been concluded mid of 2021, and optimisation has been performed afterwards. Thus, the LEXIS Pilots and the LEXIS Open Call users are supplied with a reliable data back-end. The LEXIS Data System is exclusively addressed via its well-defined APIs in the scope of our approach to manage generic data in a controllable (and machine-actionable) manner.

Below, we first discuss (Section 2) the LEXIS Data System Core (and underlying hardware); we have aligned the structure of this discussion with the LEXIS module and release scheme detailed in Deliverables D2.4 [9] and D2.6 [6]. Afterwards, we describe our efforts to benchmark and optimise the system with focus on data flows (Section 3), before we conclude (Section 4).

2 LEXIS STORAGE INFRASTRUCTURE AND DATA SYSTEM CORE

Within this Section, we discuss the core of the LEXIS Data System - as the result of Tasks 3.2 and 3.3 - in its final state as of Q4/2021. We first give a quick overview of the underlying storage infrastructure components (Section 2.1) and then focus on the entire Data System Core and the WCDA (Section 2.2). The discussion builds upon our earlier description within Deliverable D3.3 [5], which, however, covered the distributed computing and orchestration infrastructure, the portal, AAI and application-oriented systems (e.g. modelling frameworks) of LEXIS as well. The following Sections thus represent an update on the data-system-oriented part of Deliverable D3.3 ([5], Sections 2.3 and 3.2 therein).

2.1 STORAGE INFRASTRUCTURE COMPONENENTS RELEVANT FOR DDI

The hardware components relevant for the DDI comprise project-specific systems and parts of operational storage systems (Section 2.1.1), as well as the Data Nodes / Burst Buffer Nodes (i.e. buffering servers with large amounts of NVDIMMs, Section 2.1.2). The storage systems serve as back-end for the LEXIS Data System (described later in Section 2.2), while the Data Nodes have been purchased in the course of the project to accelerate data flows (with their usage further described in Sections 2.2.6, 3.2.1 and 3.2.3.1).

2.1.1 Back-end Storage Systems

The main storage systems used by the LEXIS DDI at the end of the project are listed in Table 1:. The table lists only storage resources directly used as iRODS resources at each federated site. Storage resources used for HPC jobs such as local Lustre or GPFS systems are not listed here, as those are not used exclusively by the LEXIS DDI. Further details on the resources can be found in Deliverables D3.3 [5] and D3.5 [7].

DATA CENTRE	STORAGE DEPLOYED	DESCRIPTION	TOTAL CAPACITY AVAIL.
ICHEC	• Amazon AWS instance storage (NVMe, 100 GB)	ICHEC uses an Amazon AWS VM instance to provide an iCAT server for its zone; attached storage is used as iRODS resource	0.1 TB
IT4I	• CEPH cluster in the Lexis experimental infrastructure (CephFS, 120 TB raw and 60 TB effective capacity)	Data are stored in erasure coded pool with sufficient amount of redundancy available	60 TB (120 TB raw)
LRZ	• Data Science Storage (DSS) partition (50 TB) • Experimental storage A (IBM SAS, 150 TB) • Experimental Storage B (IBM SAS, 300 TB)	All systems are used as individual resources in iRODS tiering plugin to ensure redundancy. In addition, DSS is used for staging areas (e.g. to have data available in Compute-Cloud nodes), and the Experimental Storage is used as a back-end for the WCDA.	500 TB

Table 1: Short overview of storage systems used by the LEXIS DDI (iRODS resources)

While at IT4I the Ceph system is used for universal purposes, the different systems at LRZ have been assigned to dedicated purposes: The DSS provides the necessary first tier storage for iRODS and the staging area while the Experimental Storage provides second tier storage for iRODS to ensure data safety, and storage for the WCDA.

2.1.2 Data Nodes

In order to accelerate data transfers within the LEXIS Data System by buffering and on-the-fly processing of the data, servers complying to the Data Node concept of Atos (containing large amounts of NVMe and NVRAM storage) have been purchased. These nodes can run the Atos Smart Burst Buffer (SBB) and Smart Bunch of Flash (SBF) solutions for data buffering and flash-drive export; both solutions have been applied within LEXIS (cf. Section 3.2.3.1 and SBF volume usage in IT4I – Deliverable D3.5 [7], Section 2.2.6 in there). In addition, direct usage is made of a Data Node for running the en-/decryption and compression/decompression APIs (Section 2.2.6 of this deliverable).

Table 2 briefly lists all of the Data Nodes deployed at IT4I at LRZ as part of the LEXIS physical infrastructure; a detailed description of their hardware configuration is available in Deliverables D3.3 [5] and D3.5 [7].

DATA CENTRE	DATA NODES DEPLOYED	USAGE	TOTAL CAPACITY AVAIL.
IT4I	Two servers with: • 12.8 TB NVMe • 512 GB NVDIMM • 100 GE NICs + IB NICs for direct connection with HPC fabric	Staging area for the LEXIS DDI, hypervisor nodes for LEXIS OpenStack, provider of NVMe-OF volumes for cloud instances. One server hosts NVIDIA Quadro RTX6000 GPU, the second one hosts Intel Stratix 10 FPGA.	25.6 TB NVMe, 1 TB NVDIMM

LRZ	Two servers deployed, each with slightly different configuration: Server / Data Node A: • 12.8 TB NVMe • 1.5 TB NVDIMM Server / Data Node B: • 3 TB NVDIMM	Server A provides additional staging area for encryption and compression API. Server B is used to assess the acceleration provided by the burst buffer in WP6 workflows. Both Servers are equipped with NVIDIA V100 GPU cards in order to accelerate on-the-fly data processing.	12.8 TB NVMe, 4.5 TB NVDIMM

Table 2: Short overview of LEXIS Data Nodes at IT4I and LRZ

2.2 DATA SYSTEM CORE

The LEXIS Data System Core comprises the iRODS/EUDAT [1, 2] based Data Management system used to build the LEXIS Distributed Data Infrastructure (DDI), as well as the http-REST (REpresentational State Transfer) APIs exclusively used for handling data and addressing the system. It is the result of Task 3.2 (iRODS/EUDAT deployment, Data Search and Up-/Download APIs, Data Harmonisation APIs) and Task 3.3 (Monitoring, Staging and Orchestration APIs). The description of these Tasks from the project proposal was fully implemented; the harmonisation API, however, does not harmonise data formats as was envisaged in an original idea. Instead, it was implemented as a metadata-upload API that adds descriptive metadata to any dataset in LEXIS and thus simplifies data re-usage, and as a compression/encryption API set allowing to compress and encrypt any data in LEXIS. These sub-APIs address concrete use cases and unify data handling, while there is no point in further data format unification in the scope of a multi-purpose and multi-domain-research infrastructure.

The description of the DDI below is - for the first time - oriented at the LEXIS module architecture, as laid out in Deliverable D2.6 [6] in its latest version. This architecture packages systems, in particular software systems, into "release scopes" such that for every LEXIS module a software release plan and release management can be established - i.e., all software within one module is released together. Also, it makes sure that basic prerequisites regarding documentation and source-code/software management are met for deploying the LEXIS systems and extending the platform to further sites. This has paid out when the LEXIS DDI was extended to ICHEC (cf. Section 3.1.1 and Deliverable D3.5 [7]).

The Sections below remain compact in their descriptions, mention updates within the last six months, and refer to more elaborate discussions where appropriate. After an introduction to basics of the DDI (Section 2.2.1), the technical building blocks of the DDI are explained - including the Weather and Climate Data API (WCDA) built within WP7 (Section 2.2.6), a peripheral part to the DDI, but an integral, important part of the LEXIS Data System.

2.2.1 DDI Basics

The basic idea of the LEXIS DDI has been recently described by us in a conference publication [11] (and as well in a book currently in print [10] and in Deliverable D3.3 [5]). Here, we follow this description with a few adaptations.

In order to run cross-datacentre workflows and safeguard LEXIS data, the LEXIS DDI was established, based on iRODS [1] and EUDAT [2] modules for a distributed data management system. This infrastructure stores input, intermediate and output data of LEXIS workflows and provides unified access from all computing centres. It is accessed via REST-APIs (cf. Section 2.2.6) built within the LEXIS project; thus, data handling patterns are sanitized and machine-actionable data handling is possible. One such API triggers data staging between the DDI and production computing infrastructures of the participating Supercomputing centres (and their attached cluster file systems) for running computations, which usually requires centre-specific transfer mechanisms and configurations.

Encapsulating these mechanisms, our DDI-based data management concept for LEXIS workflows decouples from the specialities and characteristics of data centres joining the LEXIS federation. iRODS allows for transparent cross-centre data access (and a unified directory-tree-like view), and – with the EUDAT-B2SAFE module [2, 12] – also for policy-based data replication and mirroring to reduce access times.

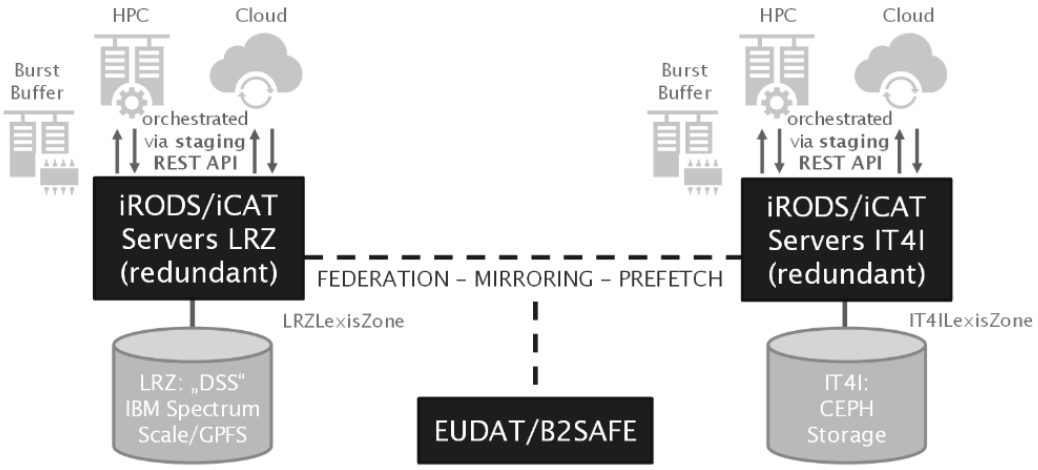

Figure 1 Federation of IT4I and LRZ within the LEXIS DDI (2019/20; now extended to ICHEC), based on iRODS and EUDAT-B2SAFE
The back-end storage systems have been discussed in Section 2.1.1. Each centre runs a so-called iRODS zone (cf. Section 2.2.2) with its independent metadata catalogue (iCAT/iRODS "provider server"). Data are transparently accessible via the inter-zone federation. Staging to Cloud/HPC systems is controlled via the staging API of the DDI (cf. Section 2.2.6), with Burst Buffers providing temporary, fast storage.

Figure 1 illustrates the LEXIS DDI federation as it was built in 2019/20 (as of 2021, this has been extended to ICHEC). The iRODS data-system middleware provides cross-centre access to data via its internal data fetching and exchange mechanisms. Manually or with EUDAT-B2SAFE, data prefetching and mirroring policies can be implemented for data within workflows or at rest. The prime example to this is a possibility to request data geo-replication in the LEXIS Portal, in order to make data available faster and significantly increase data safety. In addition, the LEXIS DDI actively supports FAIR Research Data Management [13], as it is indispensable nowadays. For confidential corporate data, the DDI features a sophisticated rights-management concept, going hand in hand with the possibility of preparing a FAIR data publication at a later time.

2.2.2 DDI Submodule IRODS: iCat PostgreSQL Database and iRODS Server Instances

The iRODS servers of the LEXIS DDI are distributed between LRZ, IT4I and ICHEC. At every site ("iRODS zone"), besides one iCAT server as mentioned in the previous Subsection, a PostgreSQL database back-end for storing the iCAT is needed, as well as a storage server (iRODS "consumer server" or the iCAT server in a double role) which has access to the storage back-end for the data. This basic DDI system, as well as its design (and design choices) are extensively discussed in a book chapter [10].

At LRZ, the iRODS infrastructure is set up resembling the HAIRS (High-Availability iRODS System) concept (see [10]). Two iRODS machines mirroring the same iRODS zone are deployed behind a HAProxy load balancer. Coupled with a redundant PostgreSQL database with fail-overs controlled by Pgpool, the availability of the LRZ LEXIS zone ("LRZLexisZone") is ensured. As physical storage back-ends for the LRZ LEXIS zone (cf. Section 2.1.1), the LRZ-DSS (GPFS/NFS system) and the LEXIS experimental storage are used. The DSS is the first choice; when full, data are automatically transferred to the experimental storage by an iRODS plugin.

IT4I deployed an iRODS zone “it4iLexisZone”, which consists of one iCAT server and one consumer server. Both servers are virtual machines deployed in a VMware cluster with high availability enabled. The servers are connected to a CephFS back-end resource with approximately 60 TB of space available (cf. Section 2.1.1). The connection is instantiated by the Linux kernel driver, which mounts the CephFS directly on both servers.

Finally, an ICHEC virtual machine was used to integrate ICHEC within the DDI, deploying an iCAT server for the “ICHECLexisZone” with database, as well as a first (relatively small) storage resource to begin with. Although ICHEC has “rented” that virtual machine from AWS, we have recorded high data-transfer rates both from the LRZ and IT4I to the machine. For redundancy and reliability as in the other two zones, this deployment may be extended in the future.

All in all, the deployment described consistently allows for inter-site data-transfer rates of more than 100 MB/s with some optimisation (cf. Section 3.1.1). Also in other respects, it has been found to well meet the requirements from the use cases as well as the surrounding LEXIS ecosystem (cf. Section 3 and [10]), as perceived up to now.

2.2.3 DDI Submodule IRODS: OpenID Plugin and Broker, Relation to AAI

In order to integrate well with the LEXIS platform, the DDI has to utilize the LEXIS authorisation and authentication infrastructure. This part of the iRODS submodule contains the software packages needed to interface the LEXIS AAI system to the iRODS storage back-end service.

IRODS has been connected to the LEXIS AAI by the authentication plugin irods-contrib/irods_auth_plugin_openid and its associated broker heliumdatacommons/auth_microservice, as described in Deliverable D3.3 [5]. The latter connection of iRODS to Keycloak had required major adaptations to the authentication plugin due to an incompatibility with the large-size tokens issued by Keycloak (cf. Deliverable D3.3 [5]). These adaptations within the LEXIS project were one of our major early technological break-throughs, and were subject of presentations e.g. at an international storage-system workshop [14]. They are being fed back to the upstream iRODS community.

In the current AAI flow, DDI-API servers dealing with user requests (below the surface, see also Section 2.2.6) receive a token from the user, and exchange it for a long-term offline token. This offline token is sent to the broker microservice for storing, and the python iRODS client used on the DDI-API servers performs iRODS calls using a hash of the offline token (see [14]). After the process is finished, the offline token is invalidated.

During the latest months of the project, we have updated all these components to support the latest versions of iRODS (4.2.8, 4.2.9 and 4.2.10). The modified libraries are available in the LEXIS GitHub repository.

2.2.4 DDI Submodule IRODS: EUDAT Components

This module contains the software packages (in particular EUDAT B2SAFE and B2HANDLE [10]) needed to implement EUDAT data management policies (cross-site replication and persistent identifier assignment) on top of iRODS. Thus, it makes the LEXIS-DDI iRODS system compatible with the EUDAT ecosystem and allows us to connect to that system and interoperate. A first data federation bridge has thus been established to EUDAT site SURFSARA, and further integration between EUDAT systems (as a part of EOSC) and the LEXIS DDI is envisaged.

A further important EUDAT service we are using is B2STAGE (again see [10]). It connects iRODS to compute resources via a GridFTP interface.

Further details on all the EUDAT components deployed in LEXIS are described in [10] but also in our older Deliverable 3.3 [5]. During the last months of the project, EUDAT B2HANDLE and B2SAFE were redeployed after upgrading iRODS to version 4.2.8. A newer version of B2STAGE (now iRODS Globus connector) was deployed. The current deployment status is reflected in Deliverable 3.5 [7].

2.2.5 DDI Submodule WCDA: Weather and Climate Data API (WCDA)

As mentioned in Section 2.2.1 and detailed here (and in Section 2.2.6), the access to data in LEXIS is facilitated via well-defined REST APIs, resulting in sanitized, convenient and well-controllable data handling patterns. In the sector of WP7 and weather forecast / climate data, the Weather and Climate Data API (WCDA) provides the respective services. It is a set of web service endpoints, enabling programmatic and consistent access to large-volume weather forecast and climate data. The outer API layer is served by a specialised database back-end for fast and versatile data access.

The WCDA has been designed as a RESTful API, following industry norms. HTTP requests can be sent to the end points defined in the API. The request layout is required to follow a specialised, domain-specific input protocol. If the request parameters match the defined protocol, information is obtained in response, or actions are mandated to the web service - for instance to prepare a data set for retrieval. A user of the API or a user application can thus send a series of HTTP requests and process the obtained responses, until the required data or information is obtained as response.

User authentication parameters have been included in the API definition so that users can provide different types of credentials and be granted access to certain dataset collections in function of their role. This allows the LEXIS WCDA services to connect to various authentication systems, such as the LEXIS AAI based on Keycloak.

The API definition will be extended in future iterations to accept “feature extraction” (i.e. sub-setting) parameters in data retrieval endpoints, such that an implementing service will have the possibility to respond only with the requested subset of data, thus reducing the amount of data transferred and enabling optimised distributed workflows.

A software system named Polytope has been developed as part of LEXIS to operate as weather and climate data service complying with the WCDA API. Emphasis has been put into developing it to be efficient, scalable, and portable, so that it can serve Big Data workflows across multiple HPC infrastructures.

The Polytope/WCDA software system has been released multiple times over the course of the project and has been deployed at three different locations: ECMWF, LRZ, and IT4I. These deployments can be accessed via the following public URLs:

- https://polytope.ecmwf.int/api/v1/ ,
- https://wcda.it4i.lexis.tech/api/v1/ ,
- https://wcda.lrz.lexis.tech/api/v1/ .

LEXIS’ Polytope/WCDA instances can be called from data workflows running on diverse parts of the LEXIS infrastructure such as Cloud-Computing systems, if there is network access to the web services implementing WCDA, and if LEXIS’ Keycloak authentication credentials are provided in the HTTP requests.

The Polytope/WCDA deployment at ECMWF provides direct access to ECMWF’s internal FDB (Fields Database), with real-time meteorological data from ECMWF’s operational forecasts, and to ECMWF’s Meteorological Archival and Retrieval System (MARS). MARS is the world largest meteorological archive, storing more than 300PB of meteorological observations and forecasts. It currently covers several past decades and is growing by 200TB/day.

The Polytope/WCDA instance at LRZ has been configured to operate in a federated mode with the instance at ECMWF. Whenever a LEXIS authenticated user requests data from the “lexis-mars” collection at the LRZ instance, data is retrieved from the instance at ECMWF and forwarded to the user.

The latest Polytope/WCDA version deployed was v0.7.5, released on 16th September 2021. The Polytope software system and python client have been published open source on GitHub, PyPi and ReadTheDocs (cf. Deliverables D2.6 [6] and D9.10 [15]).

Both the WCDA API and the Polytope software system have been described with more detail in Deliverable D7.5 [16].

2.2.6 DDI Submodule APIs: DDI APIs

The DDI APIs are the connection point between the LEXIS portal (or also an advanced user) and the data storage, including metadata handling for search and discovery, upload and download, data staging, and monitoring. Encryption and compression can also be requested where appropriate. In our earlier deliverables and publications (e.g. [10]), it has been made clear that endorsing data access strictly via APIs is important for keeping data in the LEXIS DDI machine actionable, i.e. for handling LEXIS data by orchestrated workflows.

The DDI APIs have been further subdivided thematically (also allowing the use of specialized hardware for specific tasks) into the following APIs:

- **Data search, upload, and download**. This API provides metadata search capabilities, and upload and download of datasets or specific files or directories within a dataset. Details, including endpoints can be found in [10], Section 4.6.1.
- **Staging.** The Staging API provides the necessary endpoints for the LEXIS orchestrator to stage data between different data sources at LRZ, IT4I, and ICHEC. It supports different data transfer techniques and authenticates the requests with the LEXIS AAI. It provides different options to the user to perform intermediate steps on the data such as compression and encryption before staging the data to its final destination. Details about the endpoints can be found in [10], Section 4.6.2. The staging API architecture described in Deliverable D3.2 [4] was restructured by adopting the Celery [17] distributed task queue solution and with the deployment of two Celery workers (one for staging and another for compression/encryption) at each centre. The four workers are shared between the two instances of the API. When a staging request is received, the API sends tasks to the workers taking into consideration the optimal worker to execute the task on.
- **Data size.** The Data size API provides an endpoint to calculate the size of the dataset, the total number of files, and the number of small files (less than 32MB). The results of this API calls support the LEXIS Orchestrator in making a decision whether to compress the dataset or not.
- **Replication.** The replication API provides an interface to EUDAT B2HANDLE and B2SAFE. The endpoints allow users to assign persistent identifiers (PID) to datasets and replicate datasets across the different LEXIS zones.
- **GridFTP permission handling**. Large datasets should be transferred using high-performance protocols rather than the JSON over HTTPS used in REST APIs. This API allows users to enable and disable direct access to the iRODS back-end via GridFTP.
- **SSHFS export handling.** Secure access to LEXIS staged datasets is provided by allowing the export of specific datasets to specific cloud instances, via SSHFS. This API provides the creation and deletion of the export, using SSH keys.
- **Encryption and compression.** The motivation for this API was the increased data rate provided by iRODS when staging large files. Staging a dataset with large amount of small files performs poorly compared to that of a single large file. Moreover, some use-cases require increased security which lead to the development of the encryption API. The architecture of the encryption and compression API mimics that of the staging API and deploys a celery worker on the burst buffer node at each centre. This allows for a fast compression and encryption on the dataset and decreases the time needed to stage a dataset by up to a factor of 10. The encryption uses the aes-256-ctr algorithm. The API runs on one Data Node at LRZ, while at IT4I a VM is used with Data-Node storage mounted using the Atos SBF solution. Both these solutions show very high performance, utilising NVMe or NVDIMM storage.

The specific endpoints of these APIs are found under a unified address. Further information on the deployment locations, documentation and ongoing publication of the source codes of these APIs is given in Deliverables D3.5 [7] and D9.10 [15].

3 DATA FLOW OPTIMISATION

Within Task 3.5, the data flow in LEXIS has been benchmarked with low- to high-level tests, and fundamental aspects of data flows have been optimised. As stated in Deliverable D2.6 [6], network tests and data flow benchmarks as presented here usually uncover the largest potential for optimisation within a distributed data-processing platform such as LEXIS. Thus, the tests presented here and the consequential optimisation have been essential to push the platform from its technological completeness (Milestone M6) to its final, verified and optimised status (Milestone M8). Section 3.1 presents our benchmarking efforts, and Section 3.2 the resulting actions for optimisation.

3.1 PERFORMANCE MONITORING AND BENCHMARKING

Our benchmarking concept has been oriented at the fundamental layers of the DDI - basic iRODS/EUDAT-B2SAFE system, API layer, and monitoring layer. It first tests raw iRODS performance between different sites with files of different sizes to be transferred. Section 3.1.1. shows one full test result of this group of tests - and probably the most important one for our first optimisations (cf. Section 3.2) - as an example: the speeds measured between IT4I and LRZ for a transfer of one file of different sizes using the iRODS (i) command “icp”. In a scientific publication and a bachelor thesis currently being finalised, results from the complete test suite are to be discussed as well as a comparison of iRODS transfer speeds to speeds within other distributed data management systems (GlusterFS, cf. [18]; MinIO [19]). The key result from all these tests is that iRODS shows decent performance if files are large enough.

3.1.1 Raw DDI/iRODS Performance Measurements

To gain insight into the data transfer capabilities between two iRODS zones, many factors must be considered including the size of the test dataset, the number of files and the methods used for data transfer. Since iRODS uses single threading when transferring files less than 32MB and multithreading otherwise (up to 16 threads), it was expected that the data transfer rate would increase with bigger files.

To benchmark the results between two iRODS zones, multiple tests were executed between LRZ and IT4I, LRZ and ICHEC, and IT4I and ICHEC. The following tests were performed:

1. iRODS to iRODS via icp,
2. iRODS to iRODS via EUDAT B2SAFE,
3. iget from local zone via python script,
4. iput to local zone via python script,
5. iget from remote zone via python script,
6. iput to remote zone via python script.

20 runs were executed for each case in each direction and different dataset sizes. Table 3 shows the datasets used.

DATASET	SIZE	NUMBER OF FILES	AVERAGE SIZE PER FILE
Tiny	5 MB	1	5 MB
Small	100 MB	1	100 MB
Medium	1 GB	1	1 GB
Large	10 GB	1	10 GB

Table 3: Datasets used in performance tests

For each case, the low value, high value, average, and median speed were calculated. The results show an increase in the data transfer rate by a factor of up to 25 times in some directions depending on file size. As a prime example, Figure 2 shows the data transfer rate between LRZ and IT4I using icp.

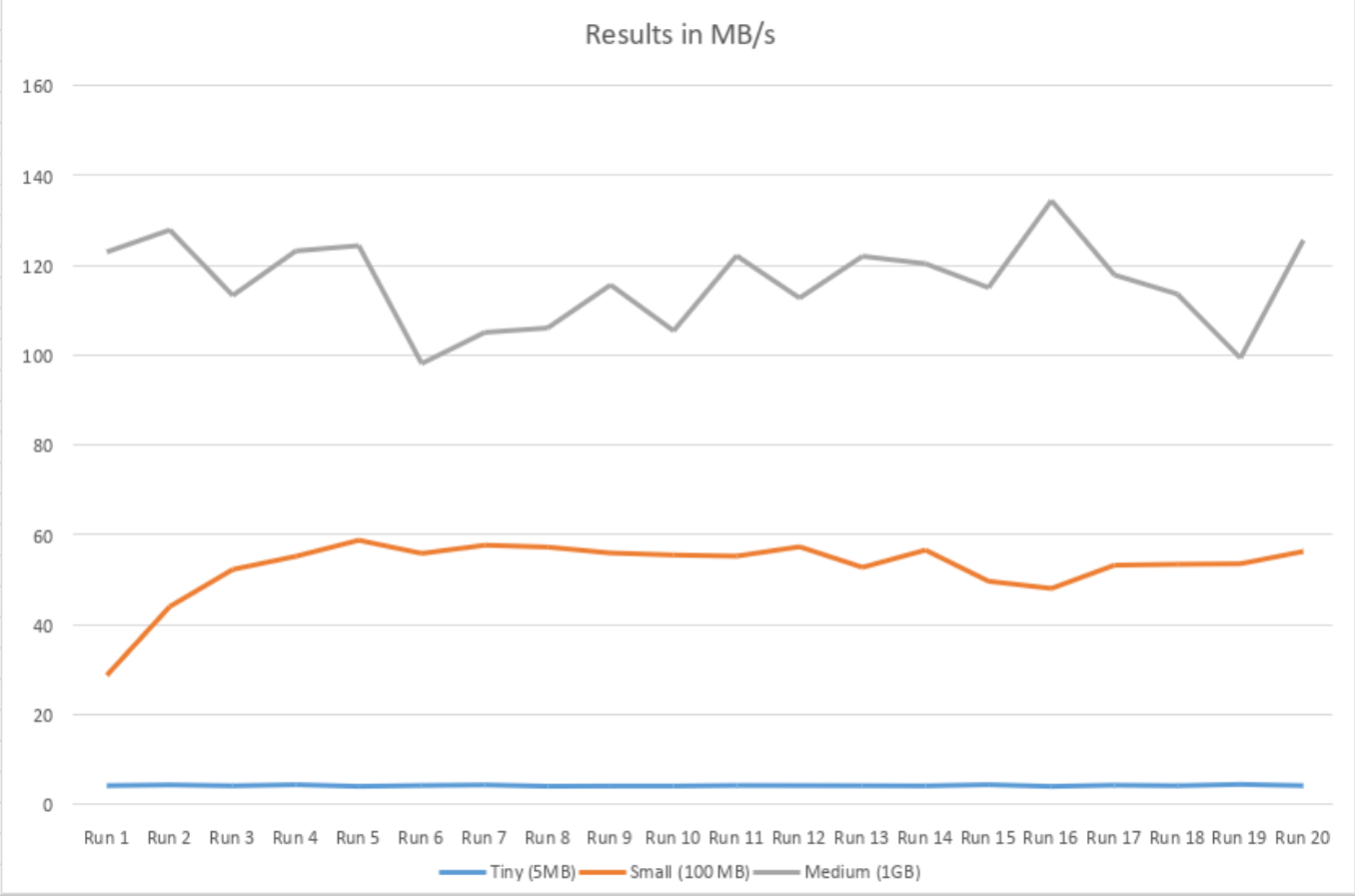

Figure 2: Data transfer rate between LRZ and IT4I using icp

These results were the main motivation to introduce the compression API described in Section 2.2.6. Further results are the subject of a bachelor thesis (LMU Munich, G. Lindner) being submitted and a publication currently being prepared.

3.1.2 DDI API Performance Measurements

Besides the low-level performance measurements described in the previous Subsection, we also implemented a golang application exercising the high-level DDI APIs and measuring the resulting data-transfer speeds. This tool enables point-to-point measurements of the several endpoints supported by LEXIS DDI and is used to measure the overhead introduced by the Python implementation of the APIs.

In particular, we test two different scenarios: transmission of a collection of small files as well as transmission of a large dataset, to assess the latency and the bandwidth. The test datasets are:

- Small-files-set: 342 files and 48 folders, 12MB in total, a mix of ASCII text file and png/jpeg images, meteorological domain definitions generated by WRF Preprocessing System,
- Large-files-set: 1 file, 8.1 GB compressed meteorological data.

The DDI APIs are designed to handle relatively big datasets, so an asynchronous behaviour has been adopted. Main operations, such as staging, replicate, duplicate and compress are triggered by REST API calls, returning a JSON message containing a `request_id`. The `request_id` is used can be used in subsequent API calls to monitor the status of the operation. The accuracy of performance measurement is limited by the frequency at which we

request updates on the progress of each request. To balance load by requests and accuracy, a polling time of 20 seconds has been selected for the small test dataset and 60 seconds for the large test dataset. DDI-side compression and encryption are not enabled in these tests.

The DDI API response time is usually in the 1.5-2 sec range. For the large-files-set, the outcome of our API tests is in line with the performance of the underlying iRODS, with a small overhead, usually in the same order of magnitude of the polling procedure, thus negligible (< 1%). For the small-files-set test case, we have experienced much lower performance, in line with the iRODS tests described in the previous Subsection. The additional overhead introduced by the high-level Python API is limited to a few seconds but adds a non-negligible latency with significant variability over time. Accurate measurement of the overhead is troublesome due to the sampling frequency, but would not gain much additional insight here anyway.

The performance assessment of the high-level DDI API suggests that the overhead is limited and justified by the added convenience. On the other side, data transfer of small files is not handled efficiently and thus we strongly suggest using the DDI facilities for data compression for every collection of small files, as a simple and effective data aggregation approach avoiding speed problems (cf. Section 3.2.1).

3.1.3 DDI Submodule Performance Monitoring: Continuous Monitoring

LEXIS has been enriched with a Dynamic Allocation Module (DAM, see Deliverable [20], Section 3.9) to dynamically determine the best location for the execution of a given task within a workflow running on the LEXIS platform. One of the parameters adopted by the DAM is the data transfer time among different computing centres. This is clearly a crucial parameter, as - depending on the task placement - data transfers can be a determining constraint on the overall execution speed of a workflow. Thus, we prepared a DDI Performance Monitoring submodule that provides the DAM with current data-transfer speeds among the LEXIS locations based on “real-life” tests, avoiding a performance assessment just based on static assumptions or one-time benchmarking.

Data collected by the Performance Monitoring module are sent to the InfluxDB database (cf. [21]) adopted by the DAM through an API provided by WP4. Each measurement is recorded as a tuple

$$< center\_src, center\_dst, file\_size, performance, timestamp >,$$

where $centre\_src$ and $center\_dst$ can be one of IT4I/LRZ/ICHEC, $file\_size$ is the size of the test dataset (in MiB), $performance$ is the measured bandwidth in MiB/sec, and the $timestamp$ can be used to retrieve the most recent measurement.

Only measurements involving two different data centres are sent to the InfluxDB, while data staging within a computing centre is not affecting the dynamic allocation, since we assume rather stable behaviour on the high-performance internal network of the LEXIS supercomputing centres.

The DDI continuous monitoring put in place may in the future be extended to have finer granularity. Furthermore, monitoring of actual data transfers within LEXIS workflows may be used to provide speed data, and thus to replace some measurements or further enhance precision.

3.2 OPTIMISED DATA MANAGEMENT

In this Section, we discuss the consequences of the findings above - general measures for performance enhancement and bottleneck avoidance within the LEXIS DDI (Section 3.2.1). Furthermore, we discuss specific results accelerating the workflows/data-transfers within the LEXIS Pilot use cases WP5-7 (Sections 3.2.2-3.2.4). The optimisations described in Section 3.2.1 and 3.2.3 have made use of the Data Nodes deployed in the LEXIS Infrastructure (cf. Section 2.1.2).

3.2.1 General Measures for Enhanced Data Flow Performance

Avoidance of ineffective file transfers

From the measurements presented in Section 3.1.1 and those discussed in Section 3.1.2, it is evident that the DDI can show performance an order of magnitude lower when confronted with the transfer of many small files, as opposed to one large file. This can be traced to the fact that iRODS writes iCAT metadata for every data object (file) written to the system, which is a relatively slow process involving multiple servers and consumes a fixed time, independent of the data size. The first, most important optimisation measure on the DDI was thus the introduction of a compression (and encryption) API (cf. Section 2.2.6). Via this API, data sets on the DDI can be ordered to be compressed into one archive. The compression process takes place in NVMe or NVDIMM-based memory on one of the Data Nodes (cf. Section 2.1.2), and thus takes little time (likewise, datasets can be ordered to be encrypted via the same API for security reasons). Transfers of the data set, from this point on, will clearly use the full iRODS transfer bandwidth, as overheads due to many iCAT read/write operations are avoided. Thus, data staging operations within typical LEXIS workflows are speeded up significantly.

Before data transfers, we have modified the response of the DDI APIs such that the orchestrator receives information on the number of small files (below a typical threshold in the order of 32MB) to be transferred. Transfers of such files are relatively ineffective not only due to iCAT writes but also because iRODS does not leverage parallel streams to transfer them. Thus, the orchestrator DAM can - if compression is impossible - decide to avoid data transfers by placing subsequent computing steps at the same site.

Optimisation of chunk sizes in file transfers

The iRODS Python client, which is used as a back-end to the staging APIs of the DDI, has turned out to transfer data in chunks of a size not really optimised. An LRZ/LMU bachelor student (G. Lindner) has been figuring out transfer speeds with different chunk sizes, and we have set the chunk size in the LEXIS production environment to a better value in the meantime, while full optimisation will require further measurements.

Optimisation of I/O conditions for HPC applications

In cases where the I/O of an HPC application is slow due to the underlying file system, the Burst Buffers concept supported by the ATOS Smart Burst Buffer solution can be applied. First, successful proofs of concept have been built for the HPC code “TsunAWI” [22] used within WP6, as described in Section 3.2.3.

Direct optimisation of applications

Clearly, also the direct optimisation of applications has been performed in LEXIS. As an example, the PostGIS [23] databases involved in WP6’s workflows have been optimised (cf. Section 3.2.3.2), and datasets transferred within WP7 workflows (cf. Section 3.2.3) are stripped in real-time to only contain necessary data.

3.2.2 WP5-specific Results - Data Compression

Within the LEXIS co-design, it turned out that the codes used by WP5 (cf. Deliverable D5.4 [24]; e.g. TRAF [25]) have a relatively strong demand for snapshotting. Simulation state snapshots (checkpoints) repeatedly generated by these codes (to track the progress of the computation and to allow for a restart after problems) were predicted to produce data rates higher than the maximum DDI data-transfer speeds. After the introduction of the compression API, the performance of the DDI in daily usage should be well sufficient to deal with the snapshotting-related transfers in WP5 workflows. The DDI can then be used to store checkpoint files produced by the TRAF computation. For the duration of the HPC computation, a specific application task monitors the files created by TRAF (getting these listed by querying a HEAppE API endpoint). After a given elapsed time without any modification to a checkpoint file (five minutes in the Turbomachinery case), this component considers that the checkpoint file is complete and calls the DDI API to store this checkpoint file in a dataset in DDI.

3.2.3 WP6-specific Results - I/O and Database Acceleration

In WP6, much effort has been put into I/O acceleration for the HPC workflow components, with tests based on the Smart Burst Buffer concept and into an acceleration of queries to the PostGIS [23] database crucial to the time-critical workflows. Further details can be found in LEXIS deliverables related to WP6 (e.g. Deliverables D6.3 [26] and D6.4 [27]) and were presented at the LEXIS Final F2F meeting in September 2021 in Prague.

3.2.3.1 Acceleration Of I/O with Help of Smart Burst Buffer

The SBB is an Atos software product that uses one or more Data Nodes (cf. Section 2.1.2) to cache writes and reads from clients to a (slower) shared filesystem (cf. [10]). This product was tested within LEXIS on a step of the WP6 workflow, which is quite suited to be accelerated by a burst buffer (of any kind): TsunAWI [22] runs usually involve a final interpolation of the results onto a differently-shaped output grid. To this end, results written to the file system as NetCDF [28] are read in again by a postprocessing library, interpolated, and written out as the final result as GeoTIFF [29]. This step can, e.g., save some storage space with respect to the raw, first output by selecting only areas of interest to remain in the final data set.

To see how much this process accelerates by buffering (given good theoretical prospects), TsunAWI was run on an SBB-cached file system at Atos facilities and on LRZ-internal experimental HPC infrastructure (using LRZ Data Node B, see Section 2.1.2). With both Atos and LRZ clusters providing extremely fast file systems (Lustre; GPFS/NFS), the file-system speed was unfortunately not saturated by the (somewhat slower) post-processing application, and thus no speed gain was visible with the real-world application. We proceeded with modified experimental scenarios, showcasing how TsunAWI can profit from a burst buffer in different HPC settings.

Experiments on Atos test cluster and results

- When the back-end file system was artificially made somewhat slower at the Atos site, a notable performance improvement was observed with the burst buffer, and the total volume of reads and writes to the back-end filesystem was reduced by 100% and 85%, respectively.
- With a simulated higher number of simple post-processing processes, saturating the file-system I/O-wise, clear benefits of the buffering are evident as well.

Experiments on LRZ experimental HPC infrastructure and results

The following experiments have involved writing at burst speed, with an I/O (file size, etc.) profile similar to the real-life tasks described above, from an LRZ experimental HPC machine over the LRZ Data Node B (Section 2.1.2) to the underlying file system. As the real-life application involves a write-read-write sequence, it may (if data processing could be accelerated or it was executed with a slower back-end file system) see additional acceleration from a burst buffer compared to the values given below.

- When 192GB of data were written, fitting into the fast SBB storage, the following write speeds were obtained:
  - Without SBB: 192GB of data written in 3m20s; at **960MB/s**,
  - With SBB: 192GB of data written in 38s; at **5000MB/s**.
- When 640GB of data were written, and the SBB cache was artificially reduced to 140GB to simulate the writing of data not fitting in the SBB, we could still gain the following write speed:
  - With SBB: 640GB of data written in 9m; at **1,200MB/s**, with a best-case behaviour that clocked in at **11GB/s** for a few seconds at the start of the experiment (while without SBB here a similar speed to above, ca. **900MB/s** was observed). This lower number here relates also to the fact that the performance of the flash cache hardware for SBB (NVDIMMs in Data Node B) is at around 1,500MB/s for long operations.

To summarize, I/O-wise the interpolation within the post-processing for TsunAWI could substantially profit from SBB usage.

3.2.3.2 PostGIS Database Optimisation

The PostGIS [23] databases involved in the WP6 workflow in order to process building- and city-related data relevant for damage prediction have been strongly optimised within the LEXIS project. While the placement of these databases on NVMeoF volumes (Atos SBF solution) promised considerable speed gains, an optimisation of the database core has accelerated WP6 workflows the most at the end.

3.2.4 WP7-specific Results - Data Stripping / Optimised Usage of WCDA and DDI

Numerical Weather Prediction (NWP), as performed within WP7 workflows (cf. [30]), generates a multi-dimensional representation of the atmosphere, of the soil, of the ocean, of Earth surface. More in general, we consider the output of an NWP model as a six-dimensional dataset including time, space (latitude, longitude, elevation), and the physical variables (e.g. temperature, relative humidity, geopotential, etc.), with a sixth dimension to represent ensemble simulations (not currently adopted in the context of the LEXIS workflow).

An NWP usually generates and stores 2D slices of the 6D model. Each slice represents a physical variable at a given time-frame for a specific ensemble member and at a specific elevation (for fields describing the atmosphere). In the case of global-scale NWP, data are usually encoded as GRIB messages, which are a compact, self-describing data format defined by WMO (cf. [31]). It is the standard adopted both by most common global-scale NWP systems including NOAA's GFS (cf. [32]) and ECMWF's IFS (cf. [33]).

In the case of IFS, ECMWF produces four simulations per day at a resolution of 9 km for the deterministic run and 51 ensemble members at 18 km resolution. This generates roughly **70 TB** of data every day. Only a small subset of these is required for a regional down-scaling performed by meso-scale NWP models like WRF, currently adopted in the context of WP7 workflows. In particular, in the LEXIS workflow, we adopt a subset of GFS NWP output as initial and boundary conditions for the regional down-scaling over Europe and Italy and a subset of IFS NWP output for regional down-scaling over Europe and France. Initial and boundary conditions for a two-day forecast of such size are usually in the order of **4-5 GB**. In the context of LEXIS, we have developed the WCDA to efficiently select and crop the required global-scale NWP and minimize data movement from global-scale NWP to regional-scale WRF runs and enable access to global-scale NWP through a RESTful API.

A similar effort has been done to handle WRF output. In this case, data are encoded in a sequence of 4D files in NetCDF [28] data format. Each NetCDF represents a time-step generated by the WRF model and includes a full 3D description of the atmosphere, earth surface and earth soil for many physical variables. A 2-day high-resolution simulation (2.5 km over France or Italy) generates 49 time-steps of about 2 GB each, for a total of roughly **100 GB per model run**. Downstream applications (hydrological simulations performed by Continuum, forest-fire risk computations by RISICO or extreme rainfall forecast computed by ERDS, cf. [30]) require only a handful of physical variables, mainly at Earth surface level. In most cases, we can thus strongly reduce the required bandwidth for data movement in the WP7 workflows (cf. Deliverable D7.6 [34]) by selecting the relevant variables for each model. We developed a data stripping procedure that inspects the WRF output and produces a stripped-down version containing only required fields, such as 2-meter temperature, 10m wind (u and v component), or relative humidity.

Stripped-down files are kept in NetCDF data format, with the same structure but much smaller, in the order of 10-12 MB per time step (as opposed to 2 GB). In our test cases, on average, data stripping thus **reduces the required bandwidth by a factor of 170**, so the benefit of the reduced data movement greatly surpasses the cost of the additional post-processing step required.

4 CONCLUSIONS

In this report, the optimisation of the LEXIS Data System in the course of Task 3.5, and the final status of the components of the LEXIS Distributed Data Infrastructure (DDI) and Weather and Climate Data API (WCDA) have been discussed. While the bare systems and installation details are reflected in Deliverable D3.5 [7], here we have described technical concepts, and our actions to benchmark the DDI system and remove bottlenecks. The transfer of large numbers of small files has turn out the most common “show-stopper” for LEXIS workflows. This problem was conceptually understood in terms of writes to the iRODS catalogue (iCAT) and mitigated by compression/bundling of files before transfer.

With the systems as described in Deliverable D3.5 [7] and the present deliverable, LEXIS-WP3‘s initial aim of a flexible and comprehensive deployment of the LEXIS Data System on two computing and one data centre (IT4I, LRZ, ECMWF) has been fulfilled, as well as the ambition to extend the LEXIS Data System to ICHEC as computing centre which has joined LEXIS in December 2020. The DDI and WCDA are successfully supporting our broad range of LEXIS use cases within the LEXIS Pilot and the LEXIS Open Call projects, and are subject of a growing number of publications (e.g. [10, 11]). Besides serving LEXIS internally, the LEXIS Data System embeds LEXIS data and the LEXIS Platform in the European data management landscape and in particular in EUDAT. The usage of EUDAT services and tools, most prominently B2SAFE and B2HANDLE, renders LEXIS compatible with next-generation data solutions in the scope of the European Open Science Cloud (EOSC) and other European infrastructure frameworks.

REFERENCES

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

[2] D. Lecarpentier, P. Wittenburg, W. Elbers, A. Michelini, R. Kanso, P. V. Coveney and R. Baxter, "EUDAT: A new cross-disciplinary data infrastructure for science," *International Journal of Digital Curation,* vol. 8, no. 1, p. 279–287, 2013.

[3] LEXIS Deliverable, *D3.1 Local Storage Solutions Report.*

[4] LEXIS Deliverable, *D3.2 Mid-Term infrastructure.*

[5] LEXIS Deliverable, *D3.3 Mid-Term Infrastructure (Deployed System Hard/Software).*

[6] LEXIS Deliverable, *D2.6 Infrastructure Validation and Assessment Report.*

[7] LEXIS Deliverable, *D3.5 LEXIS Data System Core (Infrastructure).*

[8] LEXIS Deliverable, *D3.4 Monitoring System.*

[9] LEXIS Deliverable, *D2.4 Report of LEXIS Technology Deployment - Updated Test-Beds Infrastructure.*

[10] J. Munke and et al., "Data System and Data Management in a Federation of HPC/Cloud Centres," in *HPC, Big Data, and AI Convergence Towards Exascale*, Boca Raton FL (USA), 2021.

[11] S. Hachinger and et. al, "HPC-Cloud-Big Data Convergent Architectures and Reserach Data Managment: the LEXIS Approach," in *The International Symposium on Grids and Clouds (ISGC)*, Taipei, Taiwan, 2020.

[12] „EUDAT Collaborative Data Infrastructure: B2SAFE-EUDAT," 2020. [Online]. Available: https://www.eudat.eu/services/b2safe. [Přístup získán 6 Nov 2020].

[13] M. Wilkinson, M. Dumontier, I. Aalbersberg and et al., "The FAIR Guiding Principles for scientific data management and stewardship," *Sci Data,* vol. 3, no. 160018, 15 May 2016.

[14] R. J. García-Hernández and M. Golasowski, "Supporting Keycloak in iRODS systems with OpenID authentication," in *CS3 2020 - Workshop on Cloud Storage Synchronization and Sharing Services*, 2020.

[15] LEXIS Deliverable, *D9.10 Impact KPI and Metrics Achievements Report and Plan - final version.*

[16] LEXIS Deliverable, *D7.5 First Release and Test-bed Deployment of Weather and Climate Data Interchange for both In-situ Unstructured Observations and Model Data Output.*

[17] "Ask Solem & contributors (2021): Celery - Distributed Task Queue," 2021. [Online]. Available: https://docs.celeryproject.org.

[18] B. Depardon, G. Le Mahec and C. Séguin, "Analysis of Six Distributed File Systems, Report hal-00789086," 2013. [Online]. Available: https://hal.inria.fr/hal-00789086.

[19] "MinIO | Kubernetes Native, High Performance Object Storage," 2021. [Online]. Available: https://min.io.

[20] LEXIS Deliverable, *D4.6 Design and Implementation of the HPC-Federated Orchestration System – Final.*

[21] F. Lardinois, “Comprehensive Performance Monitoring And Alert Service For Web Apps,” 2013. [Online]. Available: https://techcrunch.com.

[22] J. Behrens, “TsunAWI - Unstructured Mesh Finite Element Model for the Computation of Tsunami Scenarios with Inundation,,” in *5. NAFEMS CFD-Seminar: "Simulation komplexer Strömungsvorgänge (CFD) - Anwendungen und Trends"*, 2008.

[23] C. Strobl, “PostGIS,” in *Encyclopedia of GIS*, Boston MA (USA), Springer, 2008.

[24] LEXIS Deliverable, *D5.4 Avio Aero use cases: Critical Review to Highlight Benefits and Limits by Operating on Advanced HPC Solutions.*

[25] A. Arnone, “Viscous Analysis of Three-Dimensional Rotor Flow Using a Multigrid Method,” *J. Turbomach.,* p. 435–445, 1994.

[26] LEXIS Deliverable, *D6.3 Pilots Improvemens: Evaluation of Software Development.*

[27] LEXIS Deliverable, *D6.4 Pilots Results: Improved Scenarios Measurements and Evaluation.*

[28] UCAR/Unidata, Unidata | NetCDF, Boulder CO (USA): NetCDF, 2021.

[29] Open Geospatial Consortium, “OGC GeoTIFF Standard, Version: 1.1,” 14 09 2019. [Online]. Available: https://docs.opengeospatial.org/is/19-008r4/19-008r4.html.

[30] A. Parodi and et al., “Exploitation of Multiple Model Layers Within LEXIS Weather and Climate Pilot: An HPC-Based Approach,” in *HPC, Big Data, and AI Convergence Towards Exascale*, CRC Press / Taylor & Francis, 2021.

[31] WMO, “FM 92 GRIB Edition 2, Manual on Codes No. 306,” 2003. [Online]. Available: https://library.wmo.int/?lvl=notice_display&id=10684#.Yc3LGS_ypiM.

[32] L. Harris, L. Zhou, X. Chen and J.-H. Chen, “The GFDL Finite-Volume Cubed-Sphere Dynamical Core: Release 201912,” in *NOAA Technical Memorandum OAR GFDL*, Princeton NJ (USA), 2020.

[33] N. Wedi and et al., “The modeling infrastructure of the integrated forecasting system: Recent advances and future challenges,” in *ECMWF Technical Memoranda*, Reading (UK), 2015, p. 760.

[34] LEXIS Deliverable, *D7.6 Deployment of Test-bed Infrastructure Components and Adoption of Weather and Climate Data Interchange for Model Layer Interoperability.*