Large-scale EXecution for Industry & Society

Deliverable D4.3

Definition of Data Priority, Analytics Policies, and Security Assessment

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	D4.3 | Definition of Data Priority, Analytics Policies, and Security Assessment
RESPONSIBLE AUTHOR	Alberto Scionti (LINKS)
WORKPACKAGE ID | TITLE	WP4 | Orchestration and Secure Cloud/HPC Services Provisioning
WORKPACKAGE LEADER	LINKS
DATE OF DELIVERY (CONTRACTUAL)	31/03/2020 (M15)
DATE OF DELIVERY (SUBMITTED)	16/06/2020 (M18)
VERSION | STATUS	V1.2 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	Alberto Scionti (LINKS)
INTERNAL REVIEW	Stanislav Bohm (IT4I), Vytautas Jancauskas (LRZ), Stephan Hachinger (LRZ)

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	First draft of the document	23/01/2020	Alberto Scionti (LINKS)
0.2	Filled in all LRZ-specific/DDI-centric sections	27/01/2020	Mohamad Hayek, Stephan Hachinger (LRZ)
0.3	Contribution for section on OBM optimization added, check for Section 2 and Section 3 done (LRZ, CEA, CIMA)	28/02/2020	Alberto Scionti (LINKS), Thierry Goubier (CEA), Antonio Parodi (CIMA), Stephan Hachinger (LRZ), Mohammad Hayek (LRZ)
0.4	Addressing comments on some sections of the document	15/03/2020	Alberto Scionti (LINKS), Sean Murphy (CYC), Frederic Donnat (O24)
0.5	Addressing minor issues on the content	16/03/2020	Alberto Scionti (LINKS)
0.6	Changing images on DDI infrastructure and updating text in Section 2.1; re-written Section 4	27/03/2020	Alberto Scionti (LINKS), Stephan Hachinger (LRZ), Mohammad Hayek (LRZ), Martin Golasowski (IT4I)
0.7	Addressing Big Data explanation in Section 3, adjustment of figures	29/03/2020	Alberto Scionti (LINKS), Thierry Goubier (CEA), Emanuelle Danovaro (ECMWF)
0.8	Finalization for internal review process	31/03/2020	Alberto Scionti (LINKS)
0.8.2	Addressing internal review no. 1	06/04/2020	Alberto Scionti (LINKS)
0.9	Addressing internal review no. 2	15/04/2020	Alberto Scionti (LINKS)
1.0	Language corrections (2nd review)	29/04/2020	Vytautas Jancauskas (LRZ)
1.0.1	Review of Section 1 and alignment with Deliverable 4.5; High-level structure changed/improved	06/05/2020-11/05/2020	Frederic Donnat (O24), Alberto Scionti (LINKS)
1.1	According to the 2nd LRZ review (VJ/SH), revised content of sections and reorganized into smaller number (more coherent)	12/05/2020	Alberto Scionti (LINKS), Stephan Hachinger (LRZ)
1.1.2	Minor corrections	19/05/2020	Alberto Scionti (LINKS), Stephan Hachinger (LRZ)
1.1.3	Final changes	04/06/2020	Alberto Scionti (LINKS)
1.2	Synchronization with D4.4, final check	16/06/2020	Katerina Slaninova (IT4I)

GLOSSARY

ACRONYM	DESCRIPTION
AAI	Authentication & Authorization Infrastructure
ACL	Access Control List
API	Application Program Interface
DAG	Direct Acyclic Graph
DDI	Distributed Data Infrastructure
DOI	Digital Object Identifier
FPGA	Field-Programmable Gate Array
HEAPPE	High-End Application Execution (Middleware)
HPC	High-Performance Computing
IAAS	Infrastructure as a Service
IFS	Integrated Forecast System
IRODS	Integrated Rule-Oriented Data System
NRT	Near Real Time
NVME	Non-Volatile Memory Express
OBM	Open Building Map
PB	Petabyte
RBAC	Role Based Access Control
SSD	Solid State Disk
TB	Terabyte
TOSCA	Topology and Orchestration Specification for Cloud Applications
UI	User Interface
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
WCDA	Weather and Climate Data API
WPS	WRF Pre-processing System
WRF	Weather Research and Forecasting (model)

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
GFZ	HELMHOLTZ ZENTRUM POTSDAM DEUTSCHES GEOFORSCHUNGSZENTRUM GFZ
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

The goal of the LEXIS (Large-scale Execution for Industry & Society) project is to design and implement a platform for executing complex workflows, where the High-Performance Computing (HPC), Big Data and Cloud domains will converge. Such a platform, in its final form, will take advantage of the large-scale, geographically distributed resources provided by supercomputing centres through their respective infrastructures. The aim of the co-design activity is to ensure the integration of all needed technological elements. Workflows that require HPC and Cloud resources and need to process large amount of data (Big Data) will be effectively executed on the LEXIS platform. The LEXIS platform will integrate orchestration, distributed data management, accounting/billing, and security services; it will also allow to access to federated resources through a dedicated portal.

Orchestration and security play a key role, as a basic prerequisite for the efficient execution of the users' applications on the available resources, with convenient and secure access to computing power and storage. WP4 is responsible for developing and integrating these elements, ensuring their interoperability with the remainder of the LEXIS platform. The way data analytics and data priority are defined in the LEXIS context, the solutions implemented to address data movement and urgent computing-oriented requirements, as well as user roles (which relate to how users are granted to access data) are discussed in this deliverable.

Position of the deliverable in the whole project context

Deliverable 4.3 is a product of the WP4 (Orchestration and Secure Cloud/HPC Services Provisioning), and it is related to the activities of Task 4.1.

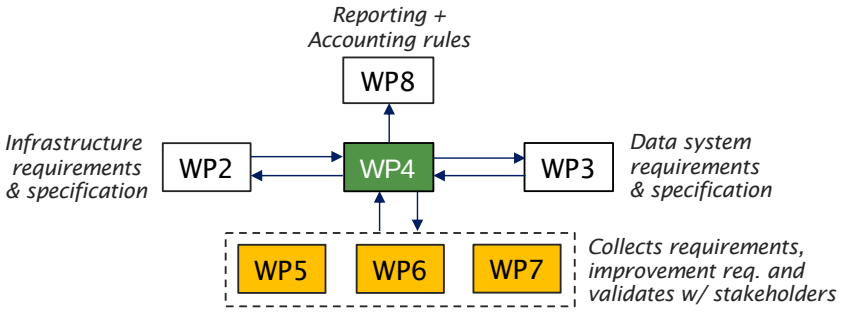

Figure 1 Position of WP4 in the LEXIS project.

As depicted in Figure 1, this work package (WP4) sets the foundations for LEXIS workflows, taking into account their entire lifecycle (from the definition, to the execution and completion), security aspects and monitoring. WP4 needs to interact with other technical work packages. WP2 provides input concerning requirements and specifications and ensures that developments in WP4 are aligned with the general requirements for the LEXIS platform. WP3 is in charge of developing LEXIS Data System, which LEXIS workflows and orchestrator rely on for storing data. WP8 is focused on creation of the portal and on using monitoring data for billing purposes. WP4 provides mechanisms for monitoring resources and the integration of workflow management at the portal level. LEXIS Pilot use cases (WP5, WP6 and WP7) will test the capabilities of the LEXIS platform, and of the orchestrator in particular.

Description of the deliverable

This deliverable focuses on the following data management aspects:

- Data analytics and data priority definition in the LEXIS context,
- Review of the LEXIS Pilot use cases to highlight (big) data and urgent computing-oriented requirements,
- Analysis of the solutions that allow to address (big) data and urgent computing-oriented requirements,
- Definition of data management policies,
- Summary of the security aspects connected to the data management.

Contributors to the deliverable are:

- LINKS as the leader of WP4, responsible for preparing this document,
- O24 as the leader of the activities concerning security aspects (WP4),
- LRZ and IT4I as partners involved in WP3, thus in the creation of the link between the LEXIS Data System and the orchestrator,
- Atos as the party responsible for the integration of orchestration technology in the LEXIS architecture,
- CEA and CIMA as the leading partners of WP6 and WP7, with responsibility for the related use cases,
- GFZ as the partner involved in the improvement of the OpenBuildingMap database, which is part of WP6,
- ECMWF which is mainly involved in the integration of WCDA.

1 DATA PRIORITY AND ANALYTICS IN THE LEXIS CONTEXT

The LEXIS platform supports the execution of very complex application workflows, with the LEXIS Pilot use cases (WP5-7, see also their deliverables [1, 2, 3, 4]) being instructive examples. Unlike pure, traditional HPC applications, the LEXIS Pilots involve computing on HPC and Cloud infrastructures, assembling and generating, moving, and analysing large amounts of data. LEXIS distributed execution environment, the "Big Data" aspect in particular necessitates the identification of respective use-case specific and generalized requirements, as well as the identification of approaches to fulfil them ("data priority"). These are one basic ingredient of the LEXIS workflow control concepts and ensure a smooth execution of the LEXIS workflows associated with the Pilot use cases on the federated LEXIS platform. The WP6 and WP7 use cases, from which corresponding LEXIS workflow templates (the reader can refer to Deliverable 4.4 [5] for the definition of "templates" and other notions specific to LEXIS workflows) are derived, are of particular importance here, as they pose the most pronounced specific requirements. Within both Pilots, execution on dynamically selected suitable infrastructure resources (at the participating computing centres IT4I and LRZ) is envisaged, also in a cross-site pattern. This avoids issues with systems being unavailable or occupied at one of the sites. Furthermore, WP7 but even more so WP6 involves "urgent computing" patterns, where certain parts of the use case execution flow (e.g., simulation components) have to be executed immediately or within some deadline.

In the following subsections, we first recap general aspects of "Big Data" relevant to the LEXIS context (Section 1.1). We proceed with an analysis of the Pilot use cases (Section 1.2) and close the section with the identification of requirements/priority aspects (Section 1.3). Section 2 will later develop this into data management concepts and policies in the multi-component and multi-site LEXIS infrastructure.

1.1 BIG DATA ANALYTICS AND LEXIS

LEXIS workflow templates are (similar to probably most definitions of a "workflow") equivalent to direct acyclic graphs (DAGs) of processing steps. Traditional HPC workflows have the main requirement of appropriate scheduling, i.e., the simultaneous allocation of multiple processing elements for each task on a system. In LEXIS, multiple heterogeneous computing systems, including IaaS Compute Clouds, are involved in the execution of a workflow. With the LEXIS Pilots involving large amounts of data, a "Big Data" handling and analytics aspect is added to this setting. The LEXIS platform and use cases are thus a prime example for the convergence of HPC-, Cloud- and Big Data-centric problems and solutions.

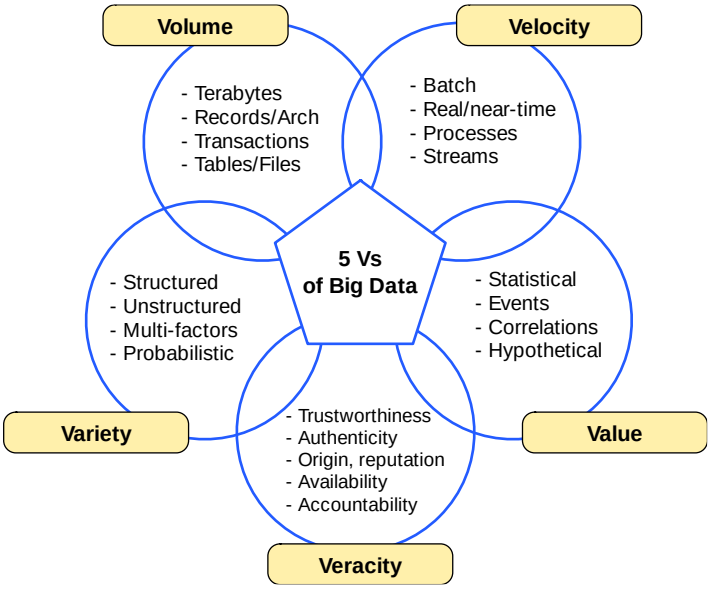

Figure 2 Big Data characteristics ("5 V's").

As illustrated in Figure 2, the handling of Big Data, and the problems and solutions appearing in this context revolve around five main aspects, known as the "5 V's" [6, 7]:

- *Volume:* large datasets must be processed, ranging from TBs to PBs,
- *Velocity:* data is generated, moved, processed, and stored at fast pace,
- *Variety:* unstructured data or data with complex structure is processed, which cannot simply be handled, e.g., by inserting it into a relational database,
- *Value:* the data is a business asset of scientific or analytic value, as it describes reality (events) and allows to find correlations or derive models and predictive hypotheses,
- *Veracity*: the correctness and accuracy of the data collected and generated is a critical point, with a large (hardly foreseeable) amount of data often requiring checks or quality control.

Looking at the LEXIS Pilot use cases, these characteristics show up on multiple occasions. The Weather and Climate Large-Scale Pilot (WP7), as an example, uses large amounts of meteorological input data. To this point, it accesses the largest European meteorological archive (MARS) with more than 300 PB of data (*Volume*). The involved workflow processes 3-D input from a coarse weather/climate model (initial/boundary conditions), as well as meteorological station data and satellite data of very different shape, which are assimilated (*Variety*). It generates output of all kinds (from forest-fire danger assessment to flood-prediction data), some of which is clearly highest of *Value* (e.g., to governmental agencies) when being delivered fast or even in a near-real-time (NRT) pattern (*Velocity*). During the complete process, data corruption will mostly result in a partial workflow reset (re-calculation of a result and all following steps), i.e., severe delays, which might necessitate dedicated data quality control/quality assurance and data-integrity checks (*Veracity*).

Similarly, in the WP6 Pilot, Earthquake and Tsunami related damage prediction requires fast data collection, processing and transfer practically under real-time constraints. The LEXIS workflow template associated to this Pilot has the particular characteristic that certain processing branches have to deliver results within a very sharp deadline. If this does not happen, the respective result will lose its value (e.g., a tsunami prediction after the tsunami has arrived), and alternative data which have been generated faster (with a more simplistic model, or on resources more easily available) have to be used.

In the following section (Section 1.2) and its subsections, we discuss **data analytics in the LEXIS context**, focusing on aspects related to the usage of the LEXIS platform. We treat the WP6 and WP7 Pilots in detail, where Big Data problems play an extremely prominent role, although clearly also WP5, the Aeronautics Pilot, touches similar aspects. According to [8], data analytics include operations applied on data such as (pre-)processing, storage, manipulation, transfer, and visualization. The WP6 discussion thus includes a section on the OpenBuildingMap database optimisation done in LEXIS, since this database is a key element of the WP6 data management chain. General requirements on LEXIS data management in the context of Big Data/data analytics and approaches to fulfil them (**LEXIS data priority**) are derived afterwards (Section 1.3).

1.2 DATA ANALYTICS IN THE LEXIS PILOTS

The LEXIS platform architecture and its technological solutions will be tested through complex applications coming from the LEXIS Pilot use cases. Among WP5, WP6 and WP7, the latter two also present requirements in terms of urgent computing on dynamically selected resources and coarse-grain stream processing (i.e., here: NRT data-stream processing with deadlines and "late data" being discarded). From the analysis of WP6 and WP7 Pilots, specific requirements for the orchestrator (see Deliverable 4.4 [5]) and data-management facilities emerged. In the following, these Pilots are further analysed with a focus on data handling.

1.2.1 WP6 Pilot

An in-depth analysis of the entire WP6 Pilot was presented in Deliverable 6.1 [9] and completed in Deliverable 6.2 [3]. It was shown that the processing steps belonging to use case execution flow (see Figure 3) can be grouped to

form processing “macro-steps” with common requirements (see coloured boxes in Figure 3). The WP6 workflow templates bear specific orchestration requirements: interaction with the user, deadlines, and scheduled interactions with a continuously running OpenBuildingMap (OBM) database.

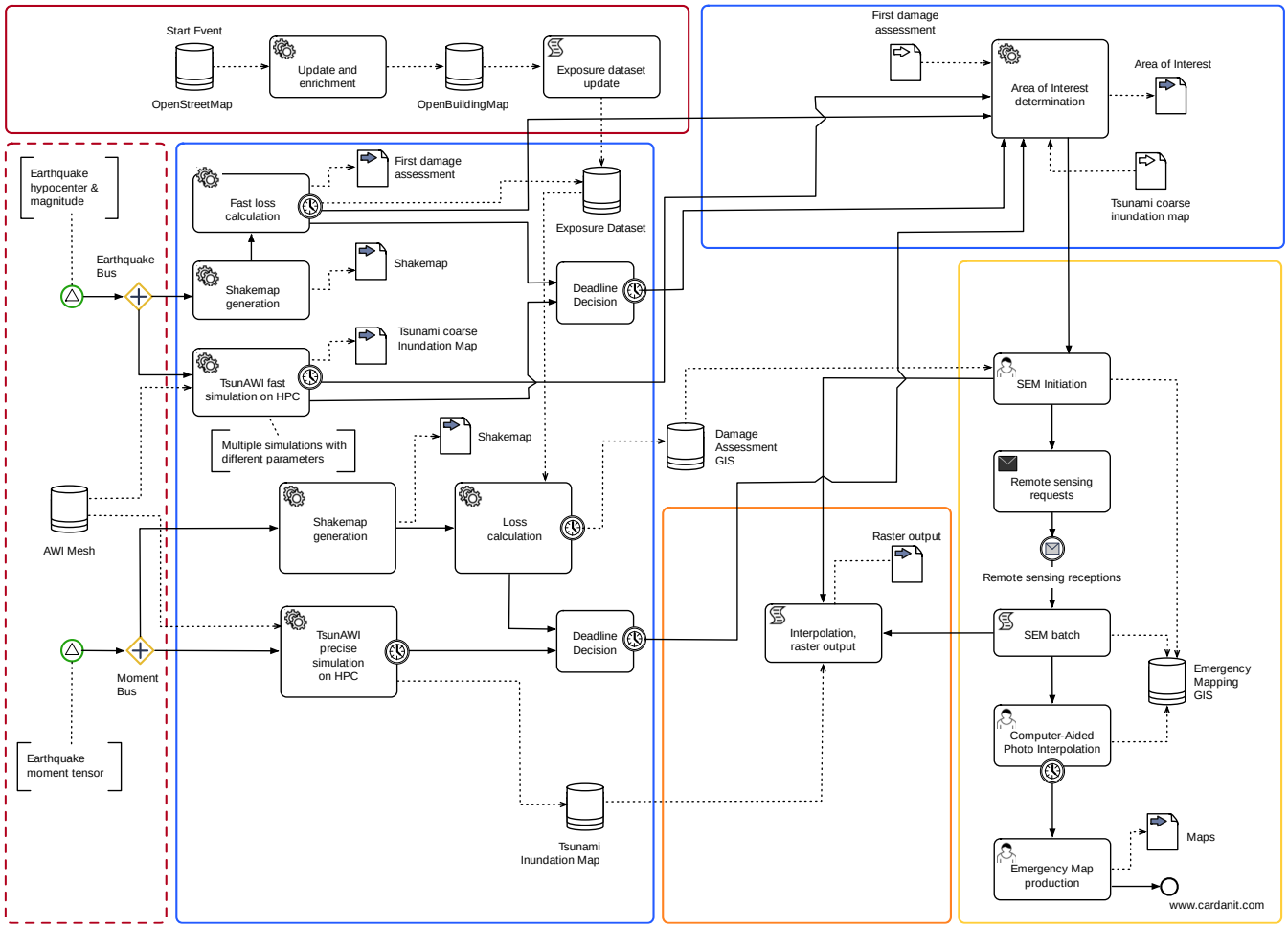

Figure 3 WP6 Pilot main execution flow highlighting macro-steps of execution (red, red-dashed, blue, orange and yellow boxes).

The OpenBuildingMap and exposure dataset are updated by continuously-triggered tasks (Figure 3, red box), fetching new data every minute from the master OpenStreetMap database. The red-dotted box in Figure 3 contains further long-running tasks for polling earthquake events. All these tasks thus require check-pointing for a workflow recovery mechanism to be put in place. The main part of the execution flow is event-triggered, and the connection to the external events’ source (the GEOFON seismic network from GFZ) is ensured by the long-running event polling task already mentioned (red-dotted box in Figure 3). After an event is received, tasks contained in the blue box can start (normally once per earthquake event). Their execution produces the following outputs: shakemaps, areas of interest, loss assessments (both coarse and detailed) and a tsunami inundation map. Those outputs are produced on the systems where the tasks have a suitable implementation to run: shakemap generation and loss assessment tasks are more Cloud-oriented, and tsunami simulations are HPC tasks, with possible execution on Cloud resources. So, tsunami arrival time and inundation maps will be produced either on an HPC or on a Cloud system (or on both), and shakemaps and loss assessments will be produced on Cloud resources. Deadline management in this workflow is ultimately dependent on data availability at the point of decision, and not task termination; available data are evaluated at the decision point which is planned to run on Cloud resources and input-producing processes are terminated. In particular, the tsunami inundation map will have to be moved from the HPC resources to the Cloud decision point.

These outputs are used to trigger the execution of tasks contained in the yellow box, with the aim of producing emergency maps (Figure 3). This is less demanding in terms of workflow management since most of the involved processing steps require intervention of operators for processing data offline (interactively). The execution is expected to typically last between three and seven days following an event. These tasks are discussed here and included in Figure 3 merely because they are relevant for whole WP6 Pilot use case and its data control: in certain scenarios, additional "delayed" requests will be made by the operators to the geographical loss assessment database or to the pipeline producing the inundation map (orange box). Intermediate results of the workflow have thus to be kept for the relevant period, and accessibility of necessary data(-bases) have to be ensured during the process.

The analysis of the execution of this LEXIS Pilot use case highlights the need of immediately triggering the execution of tasks once input data become available or events are fired. Therefore, the orchestration system must be extended with an "urgent-computing" mechanism to provide computing resources and necessary input data (at the computing site) for starting urgent tasks in the shortest possible amount of time.

The second requirement results from deadlines having to be met with the generation of certain data products. Looking at Figure 3, this already reflects in the generation of multiple alternative data products (e.g., a high- and lower-resolution version of the same data, produced on different systems). Even a concurrent execution of certain tasks (producing similar data products) on LEXIS HPC and Cloud resources must be considered if the value of the output data, or the damage by not producing it fast enough, is sufficiently large. The orchestrator must handle deadlines for execution, including those dynamically arising, for instance, from the successful fast production of a high-resolution data product: when a low-resolution production is still queued, it has to be discarded. Besides being a challenge for execution control, this poses requirements to data management: input data has to be mirrored - possibly in advance - to all resources possibly involved. Also, output data of all concurrent production tasks for a certain functional data product has to be uniformly available in the LEXIS platform. The orchestrator has to be able to estimate and minimise the time for completion of a task, from trigger to result availability, including (if appropriate) the time needed to make the data (input or output) available on the resources involved. To begin addressing all these requirements, specific TOSCA components have been designed and added to those available in YSTIA (A4C) catalogue (i.e., it contains components used to construct YSTIA Application Templates which in turn describe operatively how and what to execute on the selected computational and storage resources – see Deliverable 4.4 [5]). For instance, as reported in Subsection 2.3, the YSTIA (A4C) catalogue has been enriched with two additional ones that provide capability of controlling data movement.

An optimisation of the WP6 Pilot data flow not directly connected to orchestration policies shall not remain unmentioned here: considerable effort has been put into improving the performance of the OBM database. The main OBM database is continuously updated according to changes from the global OpenStreetMap (OSM) database. This is an intensive procedure, involving semantic enrichment of the building data in OSM, e.g., with parameters and geometries of buildings. The OBM database, based on PostgreSQL and PostGIS and held on about 12TB of flash storage, could accommodate 1 million building updates per 24h, while OSM contains about 400 million buildings. In order to accelerate this in LEXIS (see also Deliverable 6.2 [3]) leveraging the partners' experience, running the database on exported storage of the Burst Buffer (see Deliverable 3.3 for more details on the Burst Buffer technology in "Smart Bunch of Flash" mode [10]) has been considered, as well as a multi-node database setup. Currently, the updates have also been accelerated by an optimisation of the programmed process itself.

The WP6 workflows, and in particular the OBM and loss-assessment components are envisaged to make ample use of the Burst-Buffer/Data Node systems ("Smart Bunch of Flash", cf. Deliverable 3.3 [10]) in order to optimise overall speed. The necessary data transfers and reservation of the burst buffer has to be managed by the LEXIS orchestration system.

1.2.2 WP7 Pilot

The analysis on the WP7 Pilot use case (for further information see Deliverable 7.1 [11]) reveals less long-running-service/database-oriented task patterns than WP6. Instead, WP7 focuses on orchestration of multiple computational models. Also, here the triggering of models when input data become available is a central aspect, as are urgent-computing requirements, e.g., in the case of flash-flood prediction.

Figure 4 depicts the partitioning of a complete WP7 LEXIS Pilot run for hydro-meteorological forecasting. The input data (Figure 4, orange boxes) used to feed these models can partially be pre-fetched (by cron-like jobs or other mechanisms, e.g., to download the GFS global circulation data from NCEP[1]). Other parts (e.g., personal weather stations data and Italian weather data at CIMA) have to be retrieved on the short term. The data utilized will be accessed using the LEXIS Weather and Climate Data API (WCDA) besides the LEXIS Distributed Data Infrastructure (DDI) and institutional data servers. As detailed in Deliverable 7.6 [4], the execution of the hydro-meteorological forecasting model (whose main parts are represented in the blue and green boxes, Figure 4) starts with running a "WRF Pre-processing System" ("WPS") docker container on Cloud-Computing resources. Afterwards, the WRF model is executed on HPC resources (requiring up to 1,500-1,600 cores, green box). The necessary data transfers have to be controlled and optimised by the orchestrator, in particular when a (possibly flexible) multi-site allocation of the different Cloud and HPC tasks is implemented. Here, burst buffers may be used to pre-fetch input data or also to cache, e.g., the results of WPS, which often is I/O-dominated.

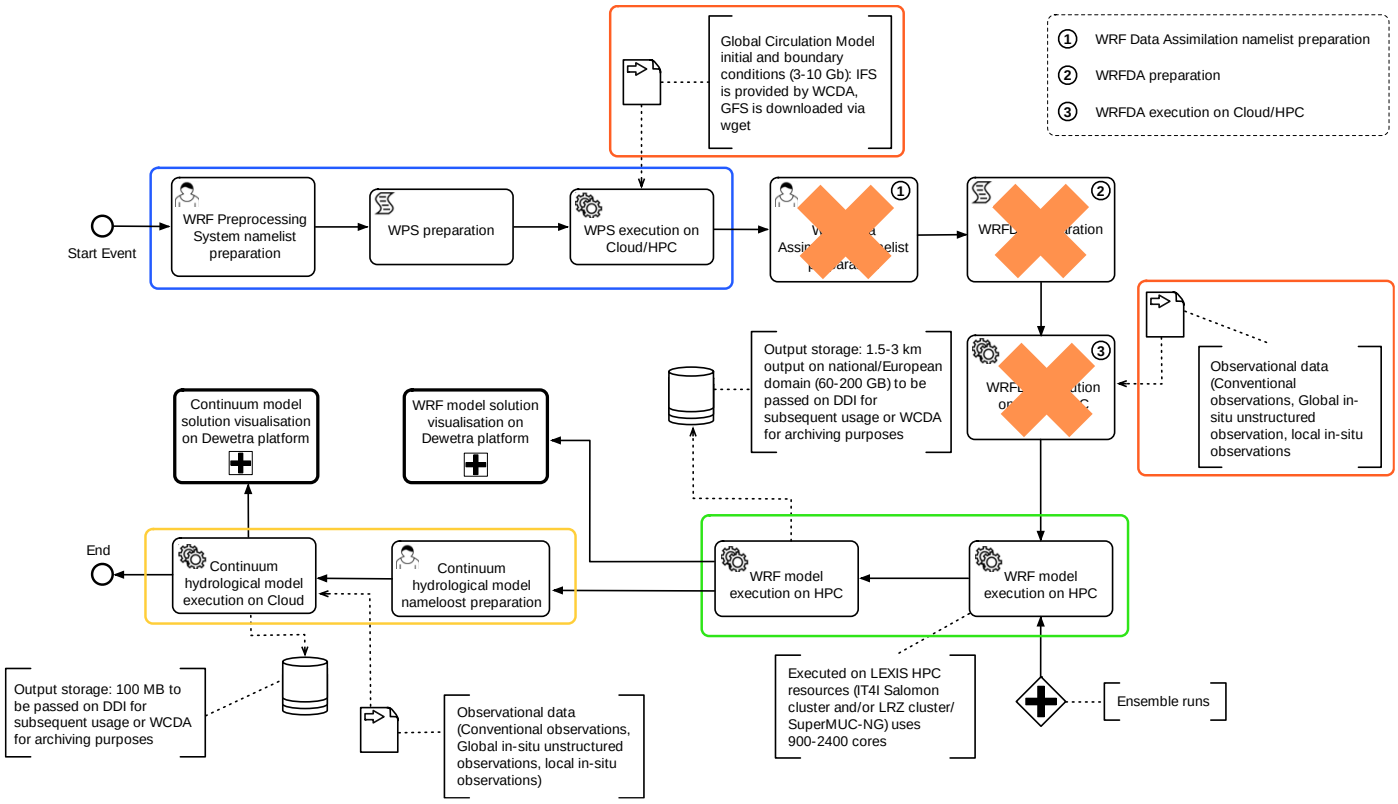

Figure 4 WP7 Pilot main execution flow highlighting macro-steps of execution (blue, orange, green, yellow, and black boxes). Tasks not yet implemented are crossed out with orange crosses.

The requirements of optimising execution time (including data transfers) and making urgent computing possible will make complex considerations necessary. Execution times for the Cloud and HPC jobs in LEXIS workflows can mostly be estimated. However, considering the case for the hydro-meteorological forecasting, the workflow involves a plethora of models and will include data-heavy assimilation tasks in its final version (see crossed-out

[1] National Centers for Environmental Prediction

parts depicted in Figure 4). The most computationally heavy part - the execution of WRF - will then split into several data-assimilation runs requiring the orchestrator to efficiently bring additional data into the running workflow (see Figure 5). The number of data-assimilation runs should be known a priori. Yet, making the data available such that processing does not stall, checking the data integrity, and triggering the necessary execution steps will be a demanding orchestration task.

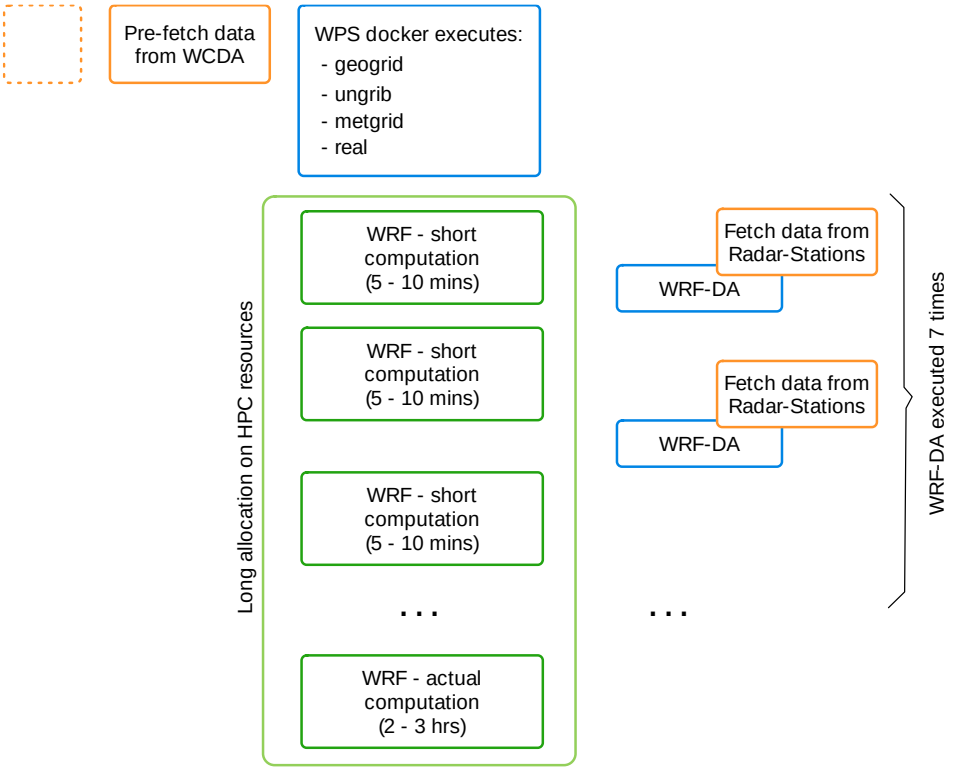

Figure 5 WP7 LEXIS Pilot use case — unrolling of the WRF execution.

The WRF output data are then fed into application models (Figure 4, yellow box). These comprise hydrological models (e.g. HMC – Continuum), a fire risk protection model (RISICO), and an air quality model (ADMS). All these models, partially being containerised, are preferably executed on Cloud-Computing infrastructures (for details the reader is referred to Deliverable 7.6 [4]). Thus, the orchestrator has to handle the necessary data transfers from HPC to Cloud systems in the LEXIS platform. As already mentioned in Section 1.2.1, in order to begin addressing these requirements, additional components have been added to the YSTIA (A4C) catalogue. Specifically, all the data-transfer tasks mentioned above, are handled by YSTIA (A4C) components, as reported in Section 2.3. Finally, in the case of hydro-meteorological forecasting use case, the output of the WRF runs is not only used to feed the "yellow box" models, but also the Dewetra[2] visualization web platform.

1.3 USE-CASE REQUIREMENTS AND DATA/ANALYTICS PRIORITY

This section discusses the requirements and priorities in data management derived from the discussion of the use cases presented above.

1.3.1 Inclusion of use-case specific databases and storage systems in workflows

For certain tasks, optimised data-management approaches have been developed in the various domains of science and research. The invaluable experience with these approaches has to be leveraged, even if this goes somewhat at the expense of the unified approach of managing LEXIS data in the Distributed Data Infrastructure (DDI). In this

[2] Dewetra visualization web platform: http://mydewetratest.cimafoundation.org

regard, the optimisation of the OpenBuildingMap database in the WP6 Pilot or the access to data sources through WCDA in the WP7 Pilot are good examples of flexible and efficient use of optimised data-management approaches. Indeed, relying on such high-performance databases and APIs (in their specific domains), as well as data fetching from legacy or institutional data repositories (e.g., institutional NFS drives) should be supported where needed.

1.3.2 Continuous writing of checkpointing / recovery files

All the LEXIS workflows associated to the Pilot use cases have to write out checkpoint data and intermediate results. These data have to be continuously saved in the LEXIS DDI or other appropriate storage systems. Checkpointing data and intermediate results retention are profitable for all LEXIS workflows in general, since they make possible partial re-executions. In case of WP5, as an additional requirement, a live analysis of the simulation progress through the checkpoints is envisaged. In this regard, data has to be written as far as possible to avoid stalling of the respective workflow (possibly, data retention time should be customisable). Thus, the orchestrator has to ensure that the proper set of resources (i.e., those providing the best execution performance, also including fast access to the storage resources) is always chosen.

1.3.3 Combined control of data prefetch and computing allocation to optimise computing time and meet urgent-computing requirements

In an urgent-computing (WP6, WP7) or time-critical setting (possibly most applications), the orchestrator has to choose timely-available and appropriate computing resources for tasks, but also make the necessary input or intermediate data available at the respective computing site. Thus, it has to optimise the combined execution time for data transfer/staging and actual computation. Failure of data transfer has to be handled, e.g., by choosing an alternative computing site.

In various circumstances (e.g., for input data with low update frequency), a longer-term prefetching may be in place. Such data may also be mirrored over several parts of the LEXIS data infrastructure (see next subsection). Data prefetched and not needed immediately shall mostly be retrieved or transferred in low-load periods of the LEXIS DDI. When many tasks are active, high loads on the LEXIS computing resources and the data infrastructure may be unavoidable. In these cases, the orchestrator should prioritize the execution of urgent-computing tasks, by selecting always the resources that provide the best estimated execution performance, possibly taking into account both computing and I/O performance for each possible resource target (e.g., the HPC computing resources and I/O subsystem in a given HPC centre).

1.3.4 Data accessibility/mirroring

In case where the site for execution of certain tasks is dynamically selected (e.g., in an urgent-computing setting), certain input data should possibly be mirrored (in advance) to all possible sites involved. Likewise, the output data must be made uniformly available on the LEXIS platform. If possible, availability (i.e., access time) for the next task has to be optimised by mirroring or physically/geographically transferring the output data on the fly. Disturbances or, in general, effects on other tasks running on the LEXIS platform at the same time have to be possibly considered.

1.3.5 Staging of data into a running workflow

In some cases, there would be a need for fetching input data made available (or generated) just during the execution of the workflow; for example, WP7's WRF, in its final version, will run with data assimilation tasks (which assimilate data from, e. g., weather-station or radar systems). Data assimilation should be handled in such a way that avoids reserved computing resources getting idle.

To this end, a proper data assimilation strategy leveraging on TOSCA components available on the YSTIA (A4C) catalogue (or newly added to the catalogue for this purpose) and the LEXIS data system must be devised and put in place.

2 DATA MANAGEMENT IN THE LEXIS CONTEXT

Data management comprises the infrastructural resources, orchestration mechanisms and policy aspects related to an efficient access to the data. This allows to fulfil the LEXIS data priority and use-case requirements, as discussed in Section 1.3.

The basic infrastructure for Data Management in LEXIS is described in Deliverable 2.1 [12] and 2.3 [13] (both public) and comprehensively documented in Deliverable 3.3 [10] (confidential). Its main components are:

- Unified, federated "Distributed Data Infrastructure" (DDI) with a common view of the LEXIS data from all participating computing/data centres; this system is based on iRODS (integrated Rule-Oriented Data System) and integrated with EUDAT's European research data services[3],
- Specialised and local storage libraries, in particular the Weather and Climate Data API (WCDA) with its back-end database for the storage of climate-related data in WP7,
- Staging and buffering resources, in particular the LEXIS Burst Buffers allowing for I/O prefetch and buffering in order to accelerate workflows.

The orchestration system uses the infrastructure via transfer components in the YSTIA (A4C) catalogue that can trigger and control LEXIS workflows' data transfers to and from the DDI and between systems. The communication of the orchestrator with the infrastructure normally takes place via http(s)-based API calls.

These aspects are discussed in the subsections below. We first focus on policy aspects for the platform (Section 2.1), which allow for fulfilling the requirements laid out in Section 1.3. These policies aim at a clean, modern, secure, and efficient data management using the best possible subset of LEXIS resources as an optimum basis for every analytics task. Sections 2.2-2.4 then focus on implementation aspects concerning the LEXIS platform, taking into account the policies defined: The usage of the LEXIS data system via APIs is described in Section 2.2. Section 2.3 deals with data system / API usage by the orchestrator. Section 2.4 discusses the interaction of LEXIS platform users with their data, as supported by the LEXIS Web portal and its connectors to the data system APIs.

2.1 POLICY ASPECTS FOR DATA ANALYTICS/MANAGEMENT IN WORKFLOWS

Data Analytics and Management in LEXIS follow a few general policies:

- *Automatization and restricted direct access:* Data transfers within a LEXIS workflow are automatically controlled by the orchestrator and do not require or allow direct user interaction. Thus, interactions with the LEXIS data system are restricted (see also Section 2.2) to APIs and defined points for user access (Portal; endpoints for staging of big datasets based on B2STAGE, pure GridFTP, or GLOBUS).
- *Cross-Site data availability:* The DDI and WCDA are used to ensure data availability at the other computing/data centres for LEXIS workflows that involve multi-site resources. All data can either be mirrored or transparently accessed from any LEXIS computing/data centre (if mirroring is too costly).
- *Data to Compute / Compute to Data:* The data system shall make the orchestrator aware of the physical data location (also on the DDI with its site-independent logical view on data). Thus, the orchestrator can access the closest data copy for staging it onto Cloud or HPC systems, and it can store output to the closest DDI back-end by default. In order to avoid slow data transfers in case data are not mirrored by default, the orchestrator may choose a different computing site.

[3] EUDAT: https://eudat.eu

From the requirements and priorities in Section 1.3, we can derive the following further policies:

- *Optimisation of execution time including estimate for data transfer:* The orchestration system shall be able to optimise the overall execution time, covering as many as possible of the LEXIS systems and possibly taking into account networking and I/O performance (through a sufficiently accurate estimation).
- *Local/Specialised storage libraries:* Specialised and high-performance storage libraries available at LEXIS participant sites shall be leveraged within LEXIS workflows as far as technically possible. With respect to general-purpose legacy storage systems, the LEXIS DDI should be preferred.
- *Checkpointing and live data staging with high priority:* Continuous checkpointing is needed by most Pilots and has to be allowed at high enough I/O priority in order to not make computing processes stall. Turning this process around, some Pilots require a live staging of data into their workflow, requiring a high-priority I/O process as well. Burst Buffers may be of help in these processes (see below).
- *Urgent Computing Priority:* In order to allow for an immediate execution, critical tasks can be executed multiple times in parallel (discarding those which finish last). They shall be executed on the most appropriate and available resources, possibly pausing, or even discarding other compute- and data-transfer tasks.
- *Data prefetching and buffering:* In particular for urgent-computing tasks, as much as possible of the input data shall be prefetched and mirrored to every computing site (in low-system-load periods) where the tasks are possibly executed. For short-term pre-fetching and output buffering, the LEXIS Burst Buffers shall be used where possible.
- *Resilience:* Failing data transfers, as far as possible, should be compensated by choosing another computing site.

These general policies concerning data management in analytics workflows shall - with time - reflect more and more in the implementation of the LEXIS orchestration system. For more concrete view of how the orchestrator can approach a dynamic management of computing and storage resource allocation, the reader is referred to Deliverable 4.4 [5].

2.2 ROLE OF DDI APIS AND WCDA

Data is brought into the LEXIS platform mainly via the LEXIS portal and semi-automatic upload facilities for extremely big datasets, which are currently being prepared (e.g., B2STAGE). The LEXIS portal (at the time of writing this document) communicates with the LEXIS DDI via a data listing API, a user/rights management API and the upload/staging API.

It is through the execution of LEXIS workflows that data gets moved to the right places and intermediate results as well as outputs get stored in the DDI as far as appropriate. To this end, the orchestrator interacts with the DDI's data-staging API. For weather and climate applications, WCDA is used instead or in addition (see Deliverable 7.1 [11]).

The interaction via APIs serves to sanitize usage patterns and thus make data manageable within an automatization/orchestration context. It also increases security by limiting the number of entry points to the system and restricting its usage. The frontends to the APIs (Orchestrator, LEXIS Portal) then serve to guarantee a user-friendly system despite the deliberate usage limitations.

As an example, we discuss the staging API in the following text, also because it is the crucial component allowing the orchestrator to move data between LEXIS data system, institutional staging/scratch spaces, and computing systems immersed in the LEXIS platform. The API provides endpoints for copying, deleting, or moving (copy-deleting) data. Addressing these endpoints (see Table 1), requests are triggered to be asynchronously fulfilled (in the background), and the completion status can be queried.

Endpoint	Method	Request body	Response body	Comment

/stage	POST	{
		"source_system":
		"lrz_iRODS",
		"source_path":
		"public/testruben/dataset-16168",
		"target_system":
		"lrz_staging_area",
		"target_path":
		"DDIStaging/dataset-161684"
		}
			{
			"request_id":
			"cc19e4a8-e4cf-4bca-bf7a-2bc9a27c44d6"
			}
				Stage a dataset from a source system to a target system

/stage/
<request_id>
	GET	-	{
			"status": "Task still in the queue, or task does not exist"
			} **or**
			{
			"status": "Task Failed, reason: <specific reason>"
			} **or**
			{
			"status": "Transfer completed"
			} **or**
			{
			"status": "In progress"
			}
				Check the status of staging call

/delete	DELETE	{
		"target_system":
		"lrz_staging_area",
		"target_path":
		"DDIStaging/dataset-161683"
		}
			{
			"request_id": "cc19e4a8-e4cf-4bca-bf7a-2bc9a27c44d6"
			}
				Delete a dataset on the target machine

/delete/
<request_id>
	GET	-	{
			"status": "Task still in the queue, or task does not exist"
			} **or**
			{
			"status": "Task Failed, reason: <specific reason>"
			} **or**
			{
			"status": "Data deleted"
			} **or**
			{
			"status": "In progress"
			}
				Check the status of the deletion call

Table 1 Main functionalities offered by the DDI's Data Staging/Transfer API relevant to the orchestrator.

Technically, the API is implemented on one (or later more) virtual machine ("data transfer steering machine") for the frontend and the backend. The actual execution of the transfers is handled by direct access to the relevant file systems, or by delegating the task to efficient lower-level staging mechanisms (B2STAGE, GridFTP, GLOBUS). Figure

6 shows the connectivity of the steering machine for full staging functionality, larger parts of which are already implemented.

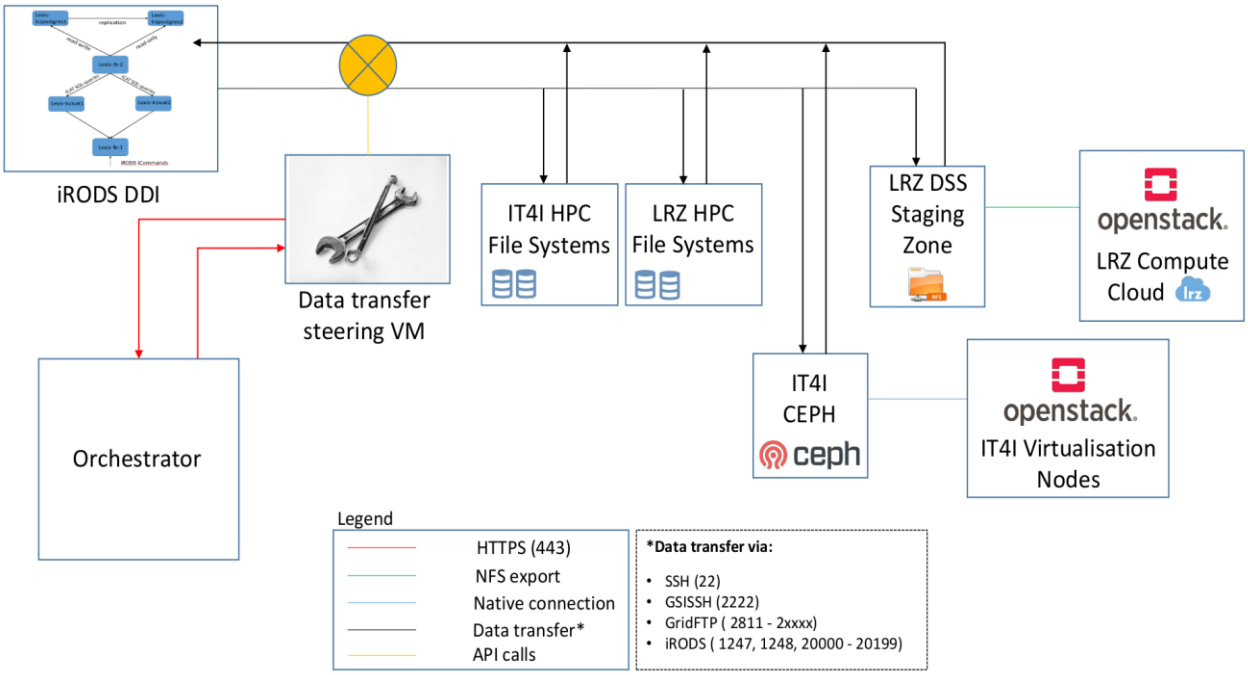

Figure 6 Schematic view of the data transfer backend of the DDI's Data Staging/Transfer API to which the orchestrator connects (the schematic of the DDI itself - used as a small icon in the upper/left corner - is shown/discussed, e.g., in Deliverable 4.5 [14]).

2.3 CONTROL OF DATA MOVEMENT BY THE ORCHESTRATOR VIA DDI APIS; ACCELERATION WITH BURST BUFFERS

Enabling the orchestrator to manage data transfers starts from the implementation of components in the YSTIA (A4C) catalogue that serve this purpose. These TOSCA software components should allow getting control of the data transfers during the execution of the workflows (i.e., LEXIS workflows are mapped on YSTIA (A4C) applications – see Deliverable 4.4 [5]). Specifically, two TOSCA components have been defined and added to the YSTIA (A4C) catalogue, having in mind transferring data from Cloud to HPC resources and vice versa:

- *CopyToJob* allows to transfer data from an OpenStack VM to the input folder of a HEAppE job (e.g., after a pre-processing phase and before the computation will take place),
- *CopyFromJob* allows to transfer data from an (output) directory of a HEAppE job to the input directory of an OpenStack VM (e.g., after the computation complete and before a post-processing phase).

These components are extended to allow for:

- Copying data from the Cloud to the DDI or the other way around, and
- Copying data from HPC systems to the DDI and the other way around.

These components are associated to a HEAppE job component in the YSTIA (YORC) workflow, by means of a relationship. Through the relationship, the components are able to retrieve the attribute values that are required to correctly implement the data transfer (e.g., source and target directories). These values are then passed to the DDI API to initiate the transfer.

As reported also in Deliverables 2.1 [12], 2.2 [15], 3.3 [10], and 4.4 [5], Burst Buffer technology provides infrastructural resources to optimize data management and transfers. The orchestrator has to take this into account (cf. Section 2.1). Accordingly, with the project progressing, the workflow management in LEXIS is envisaged to allow

for the automated inclusion of the Burst Buffers in the data flow. This allows LEXIS workflows to make use of the fast storage devices (i.e., NVDIMMS, NVMe SSDs, etc.) and the resulting high I/O performance in the buffer machines. Part of the four Burst Buffer systems (two at IT4I, two at LRZ) deployed for LEXIS contains FPGA or GPU acceleration cards. These can be used to manage (in-hardware) data compression, conversion, and encryption, thus further improving performance, flexibility, and security of data transfers.

2.4 LEXIS USER DATASET MANAGEMENT VIA THE LEXIS PORTAL

In this section we focus our attention on the interaction between the LEXIS users and the platform with the purpose of managing input and output data. The LEXIS Portal is the central component here, as it allows for dataset upload (or gives instructions to use big-data upload mechanisms such as B2STAGE, GridFTP or GLOBUS), and for making LEXIS workflows aware of the datasets uploaded. So, datasets can be "connected" to a LEXIS workflow template.

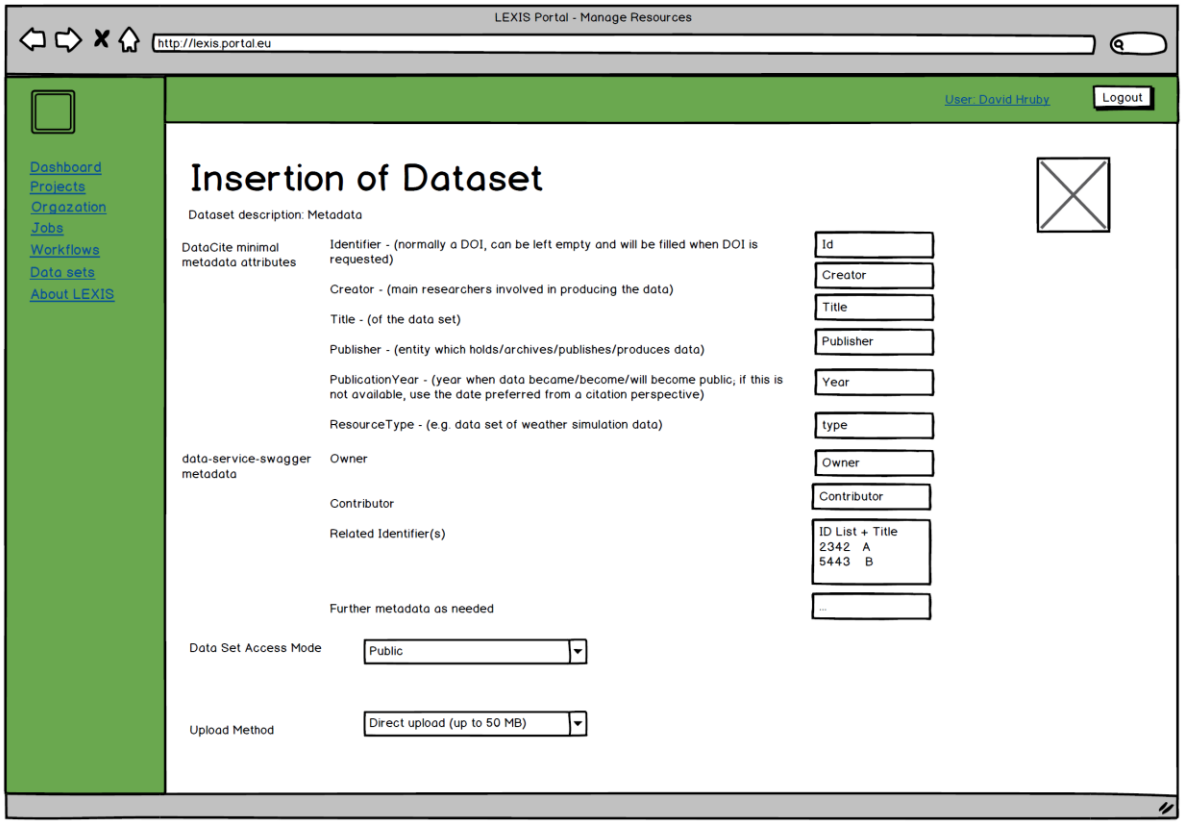

Figure 7 Wireframe UI: example view of the LEXIS portal — inputs and datasets definition

Within the LEXIS platform, a catalogue of available LEXIS workflow templates (see also Deliverable 4.4 [5]) is made accessible to the LEXIS users, as well as the listing of LEXIS datasets that can be processed through the selected LEXIS workflow templates. Input and output data staging/retrieval is to be fully handled by the orchestrator, calling the data staging/transfer API of the DDI which executes the transfers in the background. From the portal viewpoint, once a LEXIS workflow template is selected, the datasets which it operates on can also be selected. The logic in the portal ensures that datasets can be correctly bound to specific tasks of the LEXIS workflow template. Data ingest and retrieval from/to the LEXIS user (or between the DDI and other storage systems in general) is based on standard protocols including http, scp, GridFTP, GLOBUS. In the light of this, the LEXIS portal is designed in such a way that these operations are possible directly via the portal and also leveraging APIs (i.e., the data-system APIs - cf. also Deliverable 3.3 - and orchestration API [10]).

Datasets may also include metadata to support, for example, basic search queries. While the base DDI technology supports the definition of arbitrary metadata, a specific (small) set of metadata tags has been chosen for use within LEXIS. A wireframe example of the portal UI for uploading a dataset and adding metadata is shown in Figure 7. As

visible in the figure, the metadata includes basic information such as creator, title of the dataset, entity which holds/archives/publishes/produces the data, publication year, and owner. This corresponds well to the generally accepted Data Cite metadata fields needed for the retrieval of Digital Object Identifiers (DOIs), and also modes for a simple approach to "FAIR data" [16] in LEXIS. Beside this information, the user has to choose the access mode (public, private, etc.) and the type of uploading mechanism:

- Direct Upload: this mechanism is considered usable for uploading files with a maximum size of 50MB,
- B2STAGE, GridFTP or GLOBUS transfer.

3 LEXIS USER ACCESS AND SECURITY ASPECTS IN DATA MANAGEMENT

Beside the architectural aspects discussed in the previous two sections and in other deliverables already mentioned, data management is also concerned with the additional aspect of security: the capability of the LEXIS platform of performing authentication and authorization checks, which is of primary importance for granting access to the data. Realizing that security aspects are equally important as those of executing workflows in the most efficient way (i.e., with the highest performance), they have been part of the co-design activity since the beginning of the project. In the Deliverable 4.1 [17], technological solutions used to design and implement the AAI service in LEXIS were presented and analysed. Deliverable 4.5 [14] still goes in that direction, by reviewing the LEXIS architecture from a security standpoint, and highlighting the main principles adopted to ensure security.

In this section we focus our attention on those security aspects that, more than others, influence the data management, such as roles assigned to the LEXIS users (RBAC matrix). These reflect in the specific DDI/iRODS structure. We also briefly summarise the broadest context of security best practices put in place to ensure security of the LEXIS platform.

3.1 LEXIS ROLES IN DDI/IRODS

WP2 and WP4 have collaboratively devised a Role-Based Access Control (RBAC) matrix model, resulting in the definition of various roles and their rights (see Deliverable D4.5 [14]).

The RBAC matrix model yielded a certain DDI directory-structure, rights model, and rules automated rights setting on certain events (e.g., a new user joining LEXIS platform), which are implemented in the DDI subsystem. In the following, we briefly describe the general concept, which may be extended, e.g., to take into account more organisational roles in the course of the project.

3.1.1 DDI Directory Structure

Inside the root directory, three directory hierarchies are built (*user*, *project* and *public*). The first hierarchy (i.e., *user*) contains the users' directories. In these directories, the respective LEXIS users have exclusive read/write access to their data. The second directory hierarchy begins with the *project* directory (referring to "project" as scope of a LEXIS compute-time allocation/computing permission, i.e., a LEXIS Computational Project). In this directory, subdirectories and access control lists (ACLs) are such that all datasets are shared among the members of a project. The third directory hierarchy in the root is built within the *public* directory. In this directory, data is available to everyone.

3.1.2 Relation between RBAC Roles and DDI permissions

For the DDI, besides the membership of a LEXIS user in a certain project (referring to "project" again as scope of a LEXIS compute-time allocation/computing permission, i.e., a LEXIS Computational Project), the following RBAC roles are relevant for setting the rights: *lex_adm*, *lex_sup*, and *project manager*.

In general, RBAC-roles are implemented in iRODS as irods-groups; if the roles refer to a certain scope (e.g., a project manager), an irods-group is made for every scope (e.g., 'projectX_prj_adm' for a project manager which has elevated rights within a project). For support and administration purposes, the LEXIS support (*lex_sup*) and LEXIS admin (*lex_adm*) groups were created, respectively. Thus, project managers can request support and a handful amount of LEXIS support developers (*lex_sup* members) can be given access certain project directories. These rights can be also revoked by the project admins once their issue is resolved. On the other hand, elevated rights can be given to general LEXIS admins (*lex_adm*). Details on all this are given in Deliverable D4.5 [14].

Furthermore, a general LEXIS iRODS group has been created. Every member of the LEXIS platform is a member of this LEXIS group, which has, e.g., read rights on the complete *public* directory tree, as well as sufficient rights for the *project* and *user* root directories, such that all LEXIS users can pass through this directory to the specific subdirectories they are allowed to.

3.2 SECURITY BEST PRACTICES

As reported in Deliverable 4.5 [14], the LEXIS platform can be regarded to consist of three main layers interacting with each other:

- Front-end layer represented by the LEXIS Portal front-end and a reverse proxy endpoint,
- Functional service layer, which comprises of the LEXIS DDI, LEXIS Orchestration, LEXIS Portal back-end, and LEXIS AAI services,
- Infrastructural layer (HPC and Cloud infrastructures), which provides the back-end computing, storage and networking resources, and where data is loaded and processed.

By default, all LEXIS Services do not trust each other, thus they are required to authenticate against the LEXIS AAI (i.e., a user authentication token is provided, and checked by the service against the AAI system before performing any operation in the platform). In the infrastructure layer, authentication and authorization against the AAI system of the supercomputing centres is necessary. The HEAppE middleware is thus used for mapping the LEXIS AAI identity with the identity stored in HPC centres. Forcing each LEXIS service to authenticate and validate authentication and authorization against the LEXIS AAI every time is the LEXIS implementation of the principle "do not trust services". A single authentication and authorization reference system (i.e., the LEXIS AAI) is adopted instead.

The second security principle that is reflected in the LEXIS platform is the one of "least privileges". It is followed by the LEXIS AAI. The Role Based Access Control (RBAC) matrix, which contains all roles and access rights to the LEXIS platform, is built to provide the minimum set of access rights required for a task to be performed by an identity. The third security principle reflected by the LEXIS platform is that of the "separation of duties".

To address all these three security principles, appropriate architectural decisions have been made from the co-design phase to the implementation phase, which also influenced the way data management is addressed in LEXIS. The following subsection shortly summarises how we ensured that these security principles are followed by the LEXIS platform. We note that the security solutions implemented in LEXIS to adhere with the above-mentioned security principles, complement those already in place in each HPC centre.

3.3 ASSESSMENT OF BEST PRACTICES IMPLEMENTATION

Based on the architectural decisions made during co-design phase and the analysis for IAM systems (cf. Deliverable 4.1 [17]), the LEXIS AAI is based on the open source Keycloak solution. The LEXIS AAI enables token-based (OpenID Connect and SAML 2.0 protocol) authentication and authorization, and the HEAppE middleware allows to map the LEXIS identities to supercomputing centre identities following the centres' operational security models. The LEXIS AAI service is optimised to manage the authentication and authorization process in a federated environment. All other services (comprising the LEXIS DDI services) use the token provided by Keycloak (directly through the "Client" in Keycloak terminology) to perform their transactions/operations.

To ensure the proper rights to grant the access to data and computing resources, the RBAC matrix model has been adopted. The RBAC matrix provides the definition of roles inside the LEXIS platform and also allows to implement the principle of “least privileges”. Roles are assigned to the LEXIS user in such a way as to grant them the least level of privileges that allows to perform the specific operations. As explained in Subsection 3.1.2, the RBAC matrix model is reflected in the iRODS folder structure used by LEXIS users to gain access to the data.

The third principle (“separation of duties”) is addressed by ensuring that architectural elements are isolated enough to offer the smallest attack surface as possible. From this viewpoint, critical architectural components are on networks that are protected from the outside (see Deliverable 4.5 [14]). For instance, VPN and VLANs are widely used in IT4I to ensure network separations among HPC production clusters and LEXIS experimental infrastructure. The site-to-site VPN between IT4I and LRZ provides a secure communication channel for all LEXIS Services and especially LEXIS DDI (containing sensitive datasets or results such as ones for WP5).

4 CONCLUSION

This deliverable is concerned with the analysis of "Big Data" aspects in connection with the LEXIS Pilot use cases, and the identification of use-case specific and generalized requirements (referred to as "data priority") that must be addressed to enable data management and smooth execution of LEXIS workflows. Going in this direction, the document discusses the technical approaches and policies employed to fulfil these requirements. Thus, this report is closely related to other deliverables (D2.1 [12], D2.2 [15], D2.3 [13], D3.3 [10], D4.1 [17], D4.2 [18], D4.4 [5], D4.5 [14], D5.1 [1], D5.2 [2], D6.1 [9], D6.2 [3], D7.1 [11], and D7.6 [4]) that the reader can refer to for more details on the specific topics.

More specifically, this document analyses data management from different perspectives. First, it thoroughly reviews the Pilot use cases in the light of the 5 V's that define the "Big Data" context and of "urgent computing" needs. Secondly, the outcome of this analysis is a set of specific requirements which are matched by the data management approach exposed in Section 2, including mechanisms for the LEXIS users to upload and connect their datasets with the LEXIS workflow templates. The presented approach will be fully reflected in the final implementation of the DDI and orchestrator systems. Finally, the security aspects that affect data management have been analysed.

All these elements provide a comprehensive view of how the LEXIS project addresses the data management in LEXIS workflows, by providing details on the technical and architectural solutions put in place.

REFERENCES

[1] LEXIS Deliverable, *D5.1 Turbomachinery Use Case: Analysis of Results Run on State-of-Art HPC System.*

[2] LEXIS Deliverable, *D5.2 Rotating Parts Use Case: Analysis of Results Run on State-of-Art HPC System.*

[3] LEXIS Deliverable, *D6.2 Pilots Improvements: Solutions Adopted.*

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

[5] LEXIS Deliverable, *D4.4 Definition of workload management policies in federated cloud/HPC environments.*

[6] J. Anuradha, "A brief introduction on Big Data 5Vs characteristics and Hadoop technology," *Procedia computer science,* vol. 48, pp. 319-324, 2015.

[7] N. Kaur and S. K. Sood, "Kaur, Navroop, and Sandeep K. Sood. "Dynamic resource allocation for big data streams based on data characteristics (5 V s)." 27.4 (2017): e1978.," *International Journal of Network Management,* vol. 27, no. 4, 12 May 2017.

[8] A. L'Heureux, K. Grolinger, H. F. Elyamany and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access,* vol. 5, pp. 7776-7797, 7 June 2017.

[9] LEXIS Deliverable, *D6.1 Baseline scenarios and requirements.*

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

[11] LEXIS Deliverable, *D7.1 Design for Interchange of Weather & Climate Model Output between HPC and Cloud Environments.*

[12] LEXIS Deliverable, *D2.1 Pilots needs / Infrastructure Evaluation Report.*

[13] LEXIS Deliverable, *D2.3 Report of LEXIS Technology Deployment - Intermediate Co-Design.*

[14] LEXIS Deliverable, *D4.5 Definition of Mechanisms for Securing Federated Infrastructures.*

[15] LEXIS Deliverable, *D2.2 Key parts LEXIS Technology Deployed on Existing Infrastructure and Key Technologies Specification.*

[16] 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.

[17] LEXIS Deliverable, *D4.1 Analysis of Mechanism for Securing Federate Infrastructure.*