Large-scale EXecution for Industry & Society

Deliverable D2.4

Report of LEXIS Technology Deployment - Updated Test-Beds Infrastructure

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	D2.4 | Report of LEXIS Technology Deployment - Updated Test-Beds Infrastructure
RESPONSIBLE AUTHOR	Johannes Munke (LRZ)
WORKPACKAGE ID | TITLE	WP2 | LEXIS Requirement Definition and Architecture Design
WORKPACKAGE LEADER	Atos
DATE OF DELIVERY (CONTRACTUAL)	30/06/2021 (M30)
DATE OF DELIVERY (SUBMITTED)	31/07/2021 (M31)
VERSION | STATUS	V1.0 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	Atos, CYC, IT4I, LINKS, LRZ, O24
INTERNAL REVIEW	Giacomo Vitali (LINKS); Emanuele Danovaro (ECMWF)

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	Definition of structure and content	30/04/2021	Johannes Munke (LRZ)
0.2	Refinement structure/content	15/05/2021	Johannes Munke (LRZ)
0.3	Writing of Section 2	18/05/2021	Marc Levrier (ATOS)
0.4	Writing of Section 4.4	25/05/2021	Mohamad Hayek (LRZ)
0.5	Writing of Section 4.9	01/06/2021	Piyush Harsh (CYC)
0.6	Writing of Executive Summary, Section 4.1 and 5	01/06/2021	Stephan Hachinger (LRZ)
0.7	Writing of Section 4.5 Writing of Section 4.6 Writing of Section 4.10 Extension of Section 2 Extension of Section 4.1 Writing of Section 4.3 Writing of Section 4.2 Writing Section 1, 3 and 4, Revision – version ready for the 1st review	02/06/2021 07/06/2021 07/06/2021 08/06/2021 09/05/2021 11/06/2021 14/06/2021 15/06/2021	Frédéric Donnat (O24) Piyush Harsh, Diego Martin (CYC) Ondrej Salamon (IT4I) Marc Levrier (Atos) Jan Krenek, Vaclav Svaton (IT4I) Alberto Scionti (LINKS) Martin Golasowski (IT4I) Johannes Munke, Stephan Hachinger (LRZ)
0.8	Architecture diagram and module consoliation; Writing of Section 4.7 and 4.8, Revision – version for the 2nd review	19/07/2021	Rubén Jesús García-Hernández, Mohamad Hayek, Johannes Munke, Stephan Hachinger (LRZ)
0.9	Polishing text, minor corrections	23/07/2021	Johannes Munke, Stephan Hachinger (LRZ)
0.9b	References fixed and corrections from the 2nd review as well as further polishing implemented. Minor rephrasing and typo fixing.	25/07/2021 27/07/2021	Stephan Hachinger (LRZ) Marc Levrier (Atos), Stephan Hachinger (LRZ)
1.0	Final check of the deliverable	30/07/2021	Katerina Slaninova, Jan Martinovic (IT4I)

GLOSSARY

ACRONYM	DESCRIPTION
AAI	Authentication and Authorisation Infrastructure
ALIEN4CLOUD	Application LIfecycle ENablement for Cloud (https://alien4cloud.org)
API	Application Programming Interface
BB	Burst Buffer
CI	Continuous Integration
CD	Continuous Deployment
DDI	Distributed Data Infrastructure
FAIR	Findable, accessible, interoperable, reusable – the current state-of-the-art 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)
GPU	Graphics Processing Unit
HA	High Availability
HEAppE	High-End Application Execution Middleware https://code.it4i.cz/ADAS/HEAppE/Middleware/-/wikis/home
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
IAM	Identity and Access Management
IRODS	Integrated Rule-Oriented Data System (https://irods.org/)
M	Month
MODULE (LEXIS MODULE)	Logical/functional entity within the LEXIS platform, bundling a system or software component which is released, updated and deployed as a whole. Thus, LEXIS modules are the entities relevant for release managment and versioning. The actual moudules defined as of 2021 are laid out in the present deliverable.

MS	Milestone
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
RDMA	Remote Direct Memory Access
ROCE	RDMA over Converged Ethernet
REST	Representational State Transfer
SCP	Secure Copy
SR-IOV	Single Root I/O Virtualisation
SSD	Solid State Drive
SSH	Secure Shell
SSHFS	Secure Shell File System – a mechanism to mount filesystems of remote servers contacted via SSH (https://github.com/libfuse/sshfs)
TOSCA	Topology and Orchestration Specification for Cloud Applications (by OASIS, http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html)
VM	Virtual Machine
VPN	Virtual Private Network
V100	NVIDIA Tesla V100 Graphics Card with Volta GV100 GPU
WCDA	Weather and Climate Data API
WP	Work Package
YORC	Ystia Orchestrator (https://ystia.github.io/)

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 FIGURES

EXECUTIVE SUMMARY

This deliverable lays out the status of infrastructure and technology deployment at month 30, six months before the end of the LEXIS project. It reflects the finalisation of the platform development with respect to key technologies, while the rest of the project will be devoted to use-case support, validation, and further optimisation of the platform. The resulting architecture is discussed, focusing on the key technological elements - Distributed Compute Infrastructure, Distributed Data Infrastructure and Authentication and Authorisation Infrastructure. In the scope of Task 2.3, a quality and release management has been established for tracking the development of the LEXIS platform and for preparing its elements for distribution and installation at additional computing centres. Such an installation will be tested for the first time with ICHEC as a new LEXIS partner and computing centre. The release management works on the granularity of so-called LEXIS modules, entities of the logical LEXIS architecture, which have been assigned responsible persons for releases and for deployment at the participating centres. The foundation of the logical LEXIS architecture with its modules had already been developed with the release of Deliverable 2.3 [1], and has been further refined. In this deliverable, we consequently give an update on the infrastructure with respect to Deliverable 3.3 [2], with focus on the status of these logical modules. The infrastructure and platform thus described is the foundation for the successful, ongoing development of the LEXIS pilot test-beds, and is crucial to the success of the LEXIS Open Call use cases as well.

Position of the deliverable in the whole project context

Deliverable 2.2 [3] described the initial deployment of key technologies in LEXIS. Deliverable 2.3 [1] then gave an update on the deployment with a focus on Lessons Learned and actions taken to optimise the co-design, while Deliverable 3.3 [2] focussed on technical details of the deployment of LEXIS systems, accompanied by a demonstrator element (Deliverable 3.2 [4]). With the present deliverable, we give an update on the deployment of LEXIS platform components in the frame of a professional release management, established within WP2, and in particular within Task 2.3. We describe this release management in detail and indicate how we plan platform releases as a part of the LEXIS sustainability strategy. From a controlling point of view, the present Deliverable supports the assessment of fulfilment of the project Milestone MS6 “Final version of LEXIS technologies”.

Description of the deliverable

After an introduction (Section 1), the deliverable starts with a technological discussion of the key technological elements of the LEXIS platform and the logical module structure of the resulting system (Section 2). We continue with a description of release management and quality assurance (Section 3), before we detail on the status of the actual LEXIS modules (Section 4). After this comprehensive description of the state of the infrastructure, we conclude with a summary on the deliverable, emphasizing the usability aspects of the platform within the scope of the test-beds/pilots (Section 5).

1 INTRODUCTION

This deliverable has an important role needed for approving MS6. By MS6 of the LEXIS project, the final version of the architecture and technologies for the LEXIS platform will be delivered, and further efforts within the remaining months of the project will concentrate on optimizing the platform. The LEXIS platform has meanwhile matured with respect to the last deliverables reporting on co-design in WP2 context, and it is appropriate to give an update on LEXIS technology deployment and infrastructure status, which is the purpose of this deliverable.

This deliverable reports not only on single technologies, but also on LEXIS release management as a context for the current and future deployment efforts. Within WP2, and in particular within Task 2.3, a release management scheme has been developed coordinating future updates within the infrastructure, and the provision of deployment packages helping the inclusion of further centres in the LEXIS platform. To this purpose, the LEXIS platform, a very complex system of which an updated overview is given in Section 2, has been made into several logical modules, which are further subdivided into submodules. Individual modules are then described in Section 4, with an update on the respective development and deployment status.

The complexity of the system states a challenge for the co-design and development phase. While the modules are developed by different teams, compatibility among them must be ensured. For this purpose, the LEXIS consortium has imposed some rules regarding development, versioning and releases, which follow current software engineering best practices. These are described in Section 3.

The code developed in the project will be gradually made available to the public. To facilitate the extension of the platform to include further centres, we bundle necessary components to a software stack. Also, documentation will be provided, enabling a smooth deployment process. This will partly consist of the information already collected in the context of the release management.

2 LEXIS ARCHITECTURE

2.1 KEY COMPONENTS AND MIXED RESOURCE (CLOUD/HPC) CONVERGENCE MODEL

The LEXIS platform is a federated, user-friendly platform where various European High-Performance Computing centres and scientific data providers offer computing, visualisation and other resources together with a seamless access to distributed data and no compromise on execution times, scalability or security. To achieve this ambitious goal, the platform relies on the following layers:

- **Infrastructure layer,** including Cloud, HPC, storage and network systems available at each site; this layer includes **specialised hardware**, in particular **Data Nodes/Burst Buffers for accelerating data transfer and in-place conversion** and a **Smart Gateway for IoT / Edge** connections, first used by the WP7 Pilot,
- Cross-site orchestration layer or "**Distributed Compute Infrastructure**" (**DCI**),
- Distributed data layer with "**Distributed Data Infrastructure**" (**DDI**) and "**Weather and Climate Data API**" (**WCDA** – an infrastructure for storing and delivering curated weather/climate data sets).

Figure 1 on the next page gives an overview over the infrastructure layer, and how it federates the participating computing/data centres with connectivity via networks (GÉANT/Internet) and data layer components (DDI/WCDA).

In addition to these main layers, transversal services and technologies were required to achieve the project's objectives:

- **LEXIS AAI and LEXIS security model** (Authentication & Authorisation Infrastructure), providing single-sign-on, identity federation, LEXIS user accounts and token-based (password-less) execution flows,
- **LEXIS Web Portal** as a one stop shop for accessing the Platform with optimum end-user experience,
- **Accounting and Billing Engine**,
- **Approval Service** for managing compute/storage grants.

FEDERATED INFRASTRUCTURE

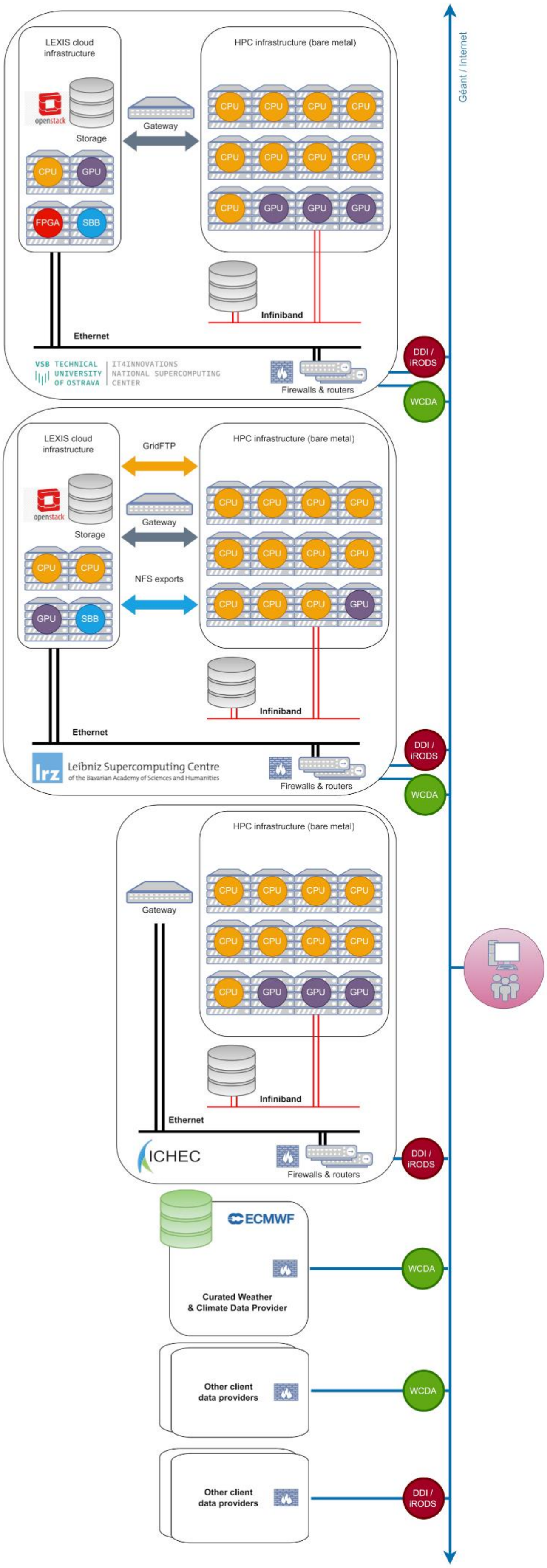

Figure 1: LEXIS Infrastructure Federation

The LEXIS platform federates resourced provided by Supercomputing centres in two modes: The "Cloud mode" offers simpler, on-demand Infrastructure-as-a-Service access to a number of compute nodes where Virtual Machines (VMs) are instantiated; in the "HPC mode", portions of supercomputers (clusters with large numbers of nodes with high-performance interconnect) are offered for parallel applications, through very strict allocation and access control schemes.

Some details of the Cloud and HPC offerings of LEXIS shall be mentioned here:

- **Cloud Computing (IT4I/LRZ):**
  - Based on OpenStack platform.
  - The LEXIS orchestration system (Yorc [5]) is directly in charge of dispatching the workloads, and provisioning the allocated OpenStack resources.
- **HPC (IT4I/LRZ/ICHEC):**
  - Highly-secured clusters with different access modes and workload scheduling systems (e.g. SLURM [6], PBS [7]), uniformly accessible through IT4I's "HEAppE" HPC-as-a-Service middleware [8]).
  - Yorc addresses HEAppE to start jobs, through a plugin developed in LEXIS.
  - Resources are only granted according to mandatory regulatory frames set up by each HPC centre.

In general, the compute nodes within these infrastructures can come with various hardware configurations, involving also specialised hardware such as GPUs or FPGAs; in an IaaS-cloud context these configurations are usually more heterogeneous (e.g. within one OpenStack-run cloud, different node types are available), leaving the user the choice of an optimum node. While on IaaS-Clouds, VMs provide access to virtualised hard disks with file systems or object storage of choice (relying e.g. on CEPH as a backend storage), HPC clusters are usually backed by specialised file systems handling massively parallel access (GPFS, Lustre).

2.2 LOGICAL MODULE ARCHITECTURE OF LEXIS

The LEXIS logical (software/infrastructure) components, as mentioned in the previous section, have been administratively and logically assigned to so-called "**LEXIS modules**". LEXIS modules have functional **submodules** with one defined person (or more) responsible for the service and one person per LEXIS computing/data centre (or more) for deployment actions.

These modules, which are further elaborated upon in Section 4, relate as follows to the main LEXIS layers and transversal services discussed in the previous Section:

- **Infrastructure layer** is reflected by the modules:
  - HPC (see Section 4.1),
  - CLOUD (see Section 4.2),

  which, with their connecting functionality, can include components already relating to the next layer.
- **Orchestration or DCI layer**, which is mostly reflected by the module:
  - ORCHESTRATOR SERVICE (see Section 4.3).
- **Data or DDI layer** is represented by the module:
  - DDI SERVICE (see Section 4.4).
- **LEXIS AAI** transversal service has its homonymous module (see Section 4.5).
- **Central Web Portal** with necessary back-end components and connectors maps to the modules:
  - LEXIS Backend Services (see Section 4.6),
  - LEXIS FrontEnd (see Section 4.7).
- **LEXIS monitoring system** is encapsulated by the module Monitoring services (see Section 4.8).
- **Accounting/billing system and the approval service** are represented by the submodules:
  - Accounting and Billing Service (see Section 4.9),
  - Approval Service (see Section 4.10).

Figure 2 gives a full overview on the LEXIS architecture and the connectivity between different modules and also submodules, as they will be further discussed in Section 4.

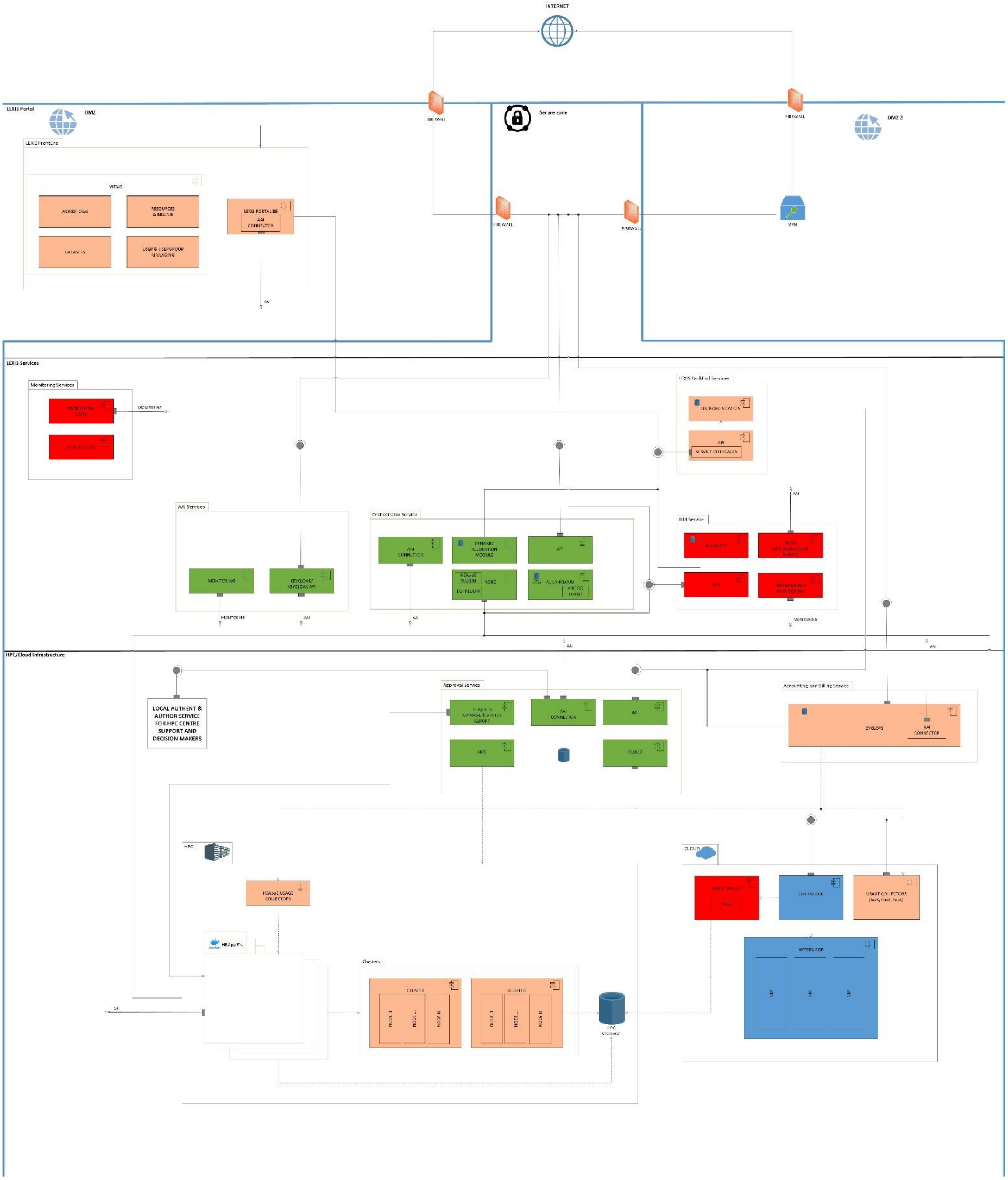

Figure 2: Diagram of the LEXIS architecture as of Milestone 6 (06/2021, Month 30). An enlarged version for better reading is available as LEXIS project public documentation https://docs.lexis.tech

3 VERSIONING, RELEASES AND CODE DISSEMINATION, AND QUALITY MANAGEMENT

Code, including installation/configuration scripts related to LEXIS modules is being bundled and made public with all relevant documentation (tutorials, API documentation, etc.) to enable compute centres to join the federation. It is intended to publish code either on LEXIS GitHub page[1] and/or its Zenodo community page[2]. During the development phase, appropriate project-internal platforms are used for collaborative development and documentation (OpenProject and GitLab hosted by IT4I). The public documentation is continuously updated at LEXIS documentation web page[3].

To ensure a decent quality of component releases within LEXIS, we developed not only development guidelines, but also guidelines for coordinated versioning and releases. This guarantees the compatibility of the LEXIS modules during the development phase and a consistent software stack. In the following, we elaborate on versioning and development/branching models and the release documentation scheme. Further technical documentation is being provided as required. We conclude the Section with general quality management standards.

3.1 VERSIONING

For assigning version numbers to the LEXIS components, Semantic Versioning [9] is used. A corresponding version number looks like:

$$n_{major}.n_{minor}.n_{patch}$$

n_{major} is only changed if there is a new version of the LEXIS component where no backward compatibility can be guaranteed (e.g. owing to API changes). For new features, n_{minor} is incremented. Bug "hotfixes" are represented by n_{patch}.

3.2 DEVELOPMENT AND BRANCHING MODEL

LEXIS components are developed with a simplified version of the Feature Branching Model [10]. Figure 3 shows a schematic diagram of the workflow.

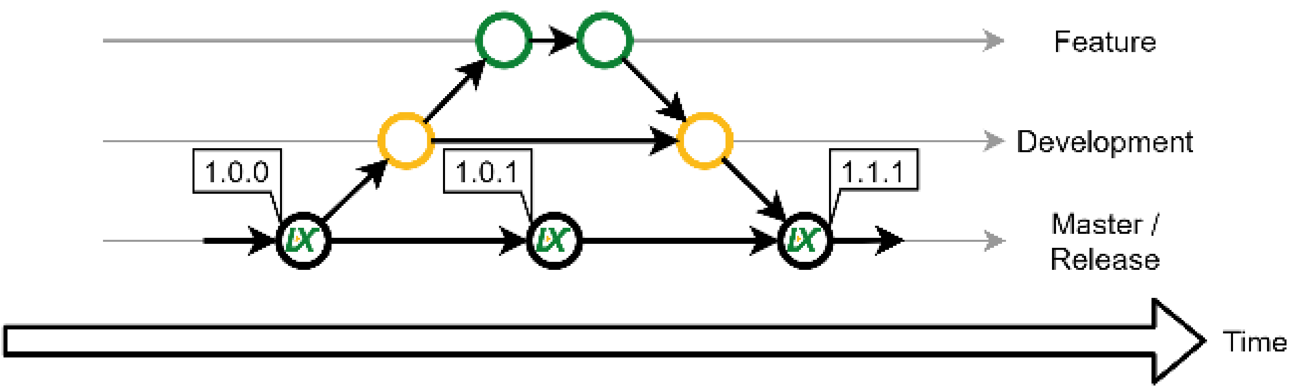

Figure 3: (Feature) Branching Model for LEXIS development

All code related to LEXIS is stored in a central GitLab repository, hosted by IT4I. The Git Version Control System (VCS) is used to keep track of the different versions and developments. The version which is deployed and/or published is always available in the master branch and is tagged with the corresponding version number (see

[1] LEXIS GitHub page: https://github.com/LEXIS-project
[2] Zenodo community page: https://zenodo.org/communities/lexis
[3] LEXIS project public documentation: https://docs.lexis.tech

Section 3.1). This opens up the possibility of using CI/CD techniques, as it is, e.g., already the case for the LEXIS Portal module. For development and testing purposes, there is a parallel development branch. For every feature request, a feature branch is created and merged into the development branch afterwards. After a final approval, the developed features from the development branch can be merged into the master. Hotfixes of bugs are usually tested, and then directly committed to the master branch.

3.3 RELEASE DOCUMENTATION

All LEXIS components and changes to them have to be documented. To this end, there is a "Module Information" page for every LEXIS component in the LEXIS wiki (OpenProject instance hosted by IT4I), which summarises the release plan as follows:

Module Information Example Module

<Description of the Module>

Overview

Software	Link to repository	Artifact type	Link to documentation
<Software (code, configuration script, ...) component>	<Link to general storage media of the code, e.g. GitLab>	<Artifact type of this component, e.g.: Python package>	<Link to the documentation of this component>

Release plan

Version	Release date	Link to release notes
<Version, e.g. 0.0.1>	<Date, e.g. 30/06/2021>	<Link to Release Notes>

The "Release Plan" table above lists the releases with version number, release date, and a link to the corresponding release notes in OpenProject, which look like this:

Release notes Example Module v.0.0.1

<Description of the Release>

Release Notes

Software	Link to Code / Software (newest version tag)
<Software (code, configuration script, ...) component>	<Link to the documentation of this component>

Changelog

- <Changes>
- <Implemented features>

This contains link to the tagged code in the `code.it4i.cz` GitLab instance and a changelog according to [11].

3.4 GENERAL QUALITY MANAGEMENT

Already early in the project, internal guidelines for managing the quality of OTHER and DEM type deliverables have been issued. Such deliverables include all LEXIS systems (hard- and software systems), which the guidelines apply to as a consequence. The guidelines include best-practice procedures for Source Code Management (on the GitLab instance `code.it4i.cz`), Documentation, DevOps practices (including CI/CD where appropriate), and automatic code sanity checks/metrics.

4 LEXIS MODULES

This chapter describes the LEXIS technology deployment status and platform/test-bed features on the granularity level of the modules and submodules. Further version-specific details are available in the LEXIS release management documentation (Section 3) internally or on request.

The modules and submodules described below correspond to the entities in Figure 2, where the communication and data flow between different platform components is reflected as well. As indicated in Section 2, we first discuss modules from the **Infrastructure layer** (HPC, CLOUD modules), then modules from the **Orchestration layer** (Orchestrator Service module) and the **Data layer** (DDI Service module), then the **LEXIS AAI** module, the modules relating to the **LEXIS Portal** (LEXIS BackEnd Services, LEXIS FrontEnd modules), and finally the modules **Accounting and Billing Service** and **Approval service**.

4.1 HPC

The HPC module of LEXIS is concerned with configuring HEAppE [8] instances for all relevant HPC cluster systems within LEXIS, allowing the LEXIS platform to integrate the HPC clusters for computation, handling LEXIS/cloud user account anonymization and mapping to HPC platforms, and providing ways for the platform to obtain relevant data (usage data, availability data, etc.) from the HPC clusters. The HPC module splits up into the submodules “USAGE COLLECTORS” (usage data collection), “HEAppE INSTANCES” (installation/configuration of HEAppE to address the relevant clusters) and “Clusters” (accessibility of clusters, cluster-internal project management, etc.).

4.1.1 Submodule “USAGE COLLECTORS”

A HPC-resource usage collector for LEXIS is deployed near the HPC infrastructure on each HPC centre. This is the data source for the Cyclops Billing system (Section 4.9), which is responsible for accounting for the usage of Cloud, HPC and storage resources on the LEXIS platform. In the backend, the usage collector forwards the queries to HEAppE instances (Section 4.1.2) for obtaining the used core hours per computational project.

4.1.2 Submodule “HEAppE INSTANCES”

Instances of the HPC-as-a-service middleware HEAppE [8] have been configured at IT4I and LRZ on VMWare-based virtual machines. These instances serve as an adaptor between the centre-specific queueing systems (SLURM, PBS) and the LEXIS Orchestration System, but also as a translation layer between internal Supercomputing-Centre accounts and LEXIS identities. Thus, LEXIS users can utilise compute time grants at the centres with their center- and grant-specific accounts, while logging in via the LEXIS single sign on. The secure mechanism for HEAppE to access the actual HPC systems is described in HEAppE and LEXIS documentation.

Multiple HEAppE instances – usually one for each supercomputing/HPC-centre project, i.e. compute-time grant – are normally installed on a single virtual machine in each HPC centre. These instances are able to address all relevant HPC clusters on these centres. HEAppE installation on IT4I and LRZ has been concluded already for a longer time; installation at ICHEC is planned.

The deployed instances of HEAppE use source code from GitHub repository[4] and the deployed version tracks upstream as much as operationally feasible. Deployed updates are documented within LEXIS release management.

4.1.3 Submodule "CLUSTERS"

Releases of the module "CLUSTERS" represent the actions to make HPC clusters in the LEXIS centres accessible via the LEXIS platform. Clearly, the clusters themselves are operationally set up and maintained by infrastructure staff not in LEXIS. Here, we only quickly enumerate the systems and their main characteristics; further details may be found, for example, in Deliverable D2.2 [3] and Deliverable D3.3 [2]. All systems at LRZ and IT4I can be addressed through the LEXIS platform and HEAppE.

IT4I. At IT4I, the Salomon supercomputer was the "flagship system" for the two years duration of LEXIS project with a performance of about 2 PFlop/s and InfiniBand FDR interconnect. This cluster will be replaced, in the course of 2021, by the Karolina system envisaged to reach a theoretical peak performance of 15.2 PFlop/s, with parts of the system operated in an IaaS-Cloud manner. The second HPC cluster available to LEXIS is the Barbora system with a theoretical peak performance of 849 TFlop/s and InfiniBand HDR interconnect. Besides these systems, a NVIDIA DGX-2 machine with V100 GPUs is installed with 2 PFlop/s of performance on AI workloads.

All IT4Innovations clusters use PBS Professional batch system for job scheduling. Users use the same HPC account across clusters for sign-in. IT4Innovations' supercomputing services (SCS) manages all installed clusters.

LRZ. The LRZ Linux Cluster, a HPC system with about 2 PFlop/s of aggregated compute power, is available to LEXIS users in a non-bureaucratic way, i.e. direct user administration by LRZ-LEXIS staff. By mid of 2020, it has been made sure that LEXIS WP5 and WP7 have access to SuperMUC-NG (LRZ's flagship machine) via compute time grants as well, based on scientifically sound proposals to the compute time committee. SuperMUC-NG provides a peak performance of almost 27 PFlop/s. In addition, LRZ provides two DGX-1 machines – model-wise the predecessor to the IT4I DGX-2 machine. SuperMUC-NG and DGX-1 machines have a somewhat separate user administration from the Linux Cluster, managed by the LRZ CXS (compute support) team.

LRZ computing clusters have been homogenised to exclusively use the SLURM batch system.

ICHEC. Access to ICHECK HPC infrastructure, with its Flagship Cluster Kay (0.6 PFlop/s) is currently being prepared.

4.2 CLOUD

4.2.1 Submodules "OPENSTACK" and "HYPERVISOR"

The CLOUD module in the architecture diagram centres on two OpenStack IaaS-Computing-Cloud instances at the time of writing of this document, represented by the submodules "OPENSTACK" and "HYPERVISOR". Both these cloud instances at IT4I and LRZ are used by the LEXIS platform to deploy VMs which are part of the workflows.

The large LRZ OpenStack instance has been in production for quite some time (therefore it tends also to be a bit more conservative in setup) and is documented under [12]. LEXIS makes use of it through a HEAppE plugin, providing an OpenStack token to the LEXIS orchestrator, and then through the orchestrator itself.

The IT4I OpenStack Cloud instance (on "LEXIS Virtualisation Nodes" at IT4I) is operated as experimental test-bed; therefore various configuration modifications can be tested. In order to provide secure access to this experimental cloud endpoint, a dedicated VPN and user identity management services have been deployed on the infrastructure. The cloud is based on OpenStack Ussuri release and is deployed using Ansible (OSA) on several virtualisation servers. Basic services like Nova (hypervisor management), Cinder (block storage) and Keystone (identity) have been

[4] HEAppE GitHub repository: https://heappe.eu

deployed along with Neutron networking with several VLANs and tenant networks based on VXLAN. A Magnum service for hosting Kubernetes clusters has been also deployed.

In terms of storage, both clouds use dedicated CEPH clusters. The KVM hypervisor is used on all nodes at IT4I and LRZ.

While the IT4I OpenStack instance may see further upgrades and releases on the mid-term, the LRZ setup will be rather static, at least for the rest of the LEXIS project.

4.2.2 Submodules “BURST BUFFER” and “FPGA”

Both at LRZ and IT4I, so-called Data Nodes or Burst Buffer Servers, have been installed. Such systems are equipped with large NVMe and NVRAM (Intel Optane DC) storage modules to accelerate data transfers by buffering. To experiment with in-place data processing (conversion, encryption etc.), GPU cards have been installed (at IT4I/LRZ) as well as one FPGA card (at IT4I).

While the hardware installation and configuration of these nodes has been completed and will be probably not changed on the mid-term, the ATOS Flash Accelerator solution deployed on the machines may see further updates. This solution provides a Smart Burst Buffer (SBB) module which can buffer input/output data for HPC or Cloud-Computing processes (taking peak load off the file system), and a Smart Bunch of Flash (SBF) module to export the Data Node storage through NVMe over Fabrics (NVMeoF) and RDMA over converged Ethernet (RoCE).

With GPU/FPGA accelerators in LEXIS, sometimes also experimentally attached to other IaaS-Cloud nodes, experiments have been conducted to expose such hardware efficiently to virtual machines, using SR-IOV and PCI passthrough.

4.2.3 Submodule “USAGE COLLECTORS”

As within the USAGE COLLECTORS submodule of the HPC submodule, endpoints are provided for the Cyclops system to evaluate system usage (first of all, CPUh consumption) and process these data for accounting and billing.

4.3 ORCHESTRATOR SERVICE

The LEXIS Orchestrator Service, as core component within the LEXIS DCI (Distributed Compute Infrastructure), provides all the necessary features to enable the execution of LEXIS workflows across the geographically distributed (computational, storage and networking) resources made available by LEXIS Supercomputing centres (IT4I, LRZ and ICHEC). It is based on an extension of the Ystia Suite for Orchestration, with its main components Yorc (Ystia Orchestrator) and Alien4Cloud (a GUI to compose TOSCA-based workflows).

Thus, the LEXIS user is provided with the capability of graphically specifying workflow components and composing a workflow, to execute it on the available resources, and to monitor the execution to collect information on the status of running jobs.

A more detailed description of the LEXIS Orchestration Service is provided in Deliverable D4.4 [13]. Here, we give an overview of the submodule characteristics.

4.3.1 Submodule “ALIEN4CLOUD” and “A4C GO CLIENT”

ALIEN4CLOUD (also simply referred to A4C) is the Ystia frontend. It provides all the functionalities to allow LEXIS users to define their application templates (based on their specific application workflows), and to execute them on a selected set of resources. It integrates a plugin for connecting to the YORC submodule, which actually executes the workflows, and it also interfaces with the LEXIS AAI for authorizing the incoming requests. It has been currently deployed in IT4I and LRZ. The ALIEN4CLOUD interface is accessible though the LEXIS portal infrastructure by an interfacing library where Go is used as a standard language. This library has been recently finalised.

4.3.2 Submodule "YORC"

YORC is the backend of the Ystia solution. It has been deployed both in LRZ and IT4I centres (HA configuration) and provides the functionalities for provisioning and executing the workflows. It also connects with the DDI and the Dynamic Allocator Module (Section 4.3.4). While stable, this component may be subject to updates in order to include new upstream functionality.

4.3.3 Submodules "YORC HEAppE PLUGIN" and "YORC DDI PLUGIN"

These plugins, developed within LEXIS, enable YORC to execute HPC jobs (via HEAppE) and to interact with the LEXIS Distributed Data Infrastructure (DDI) in order to control the data flows within single- or multi-site workflows. They are updated and enriched with features on demand; a solid status has been already reached up to this phase of the project.

4.3.4 Submodules "DYNAMIC ALLOCATION MODULE"

DYNAMIC ALOCATION MODULE (formerly known as Business Logic Unit) has been developed from in the LEXIS project to provide the capability of dynamically (i.e., during the execution of the workflow) selecting the best location set (i.e., the clusters where to run the next workflow steps), based on the value returned by an overall cost function. The module is built upon the Flask python framework, which allows to easily implement a service exposing a REST-API. The service is hosted on a virtual machine (VM) running at LRZ and consuming relatively small amount of resources (2 vCPUs, 8GB of RAM, and 20GB of storage).

The VM also runs an InfluxDB service which is used to store mainly three kinds of information: (1) the cost of all the evaluated locations (logging historical data); (2) the speed for transferring data among different locations (here, both the size and the number of files to transfer is considered); and (3) maintenance dates of the various computing systems. On this basis, the DAM can then further decide on an optimised execution location taking into account task constraints, resource availability, and provided performance. Connection with the LEXIS AAI is also integrated in the module.

At last, a Yorc plugin has been also created to connect with the exposed REST API.

4.3.5 Submodules "AAI CONNECTOR" and "API"

These submodules describe the interfaces implemented in the LEXIS Orchestration System to address the LEXIS AAI (in order to check user authentication and authorisation) as well as to be addressed by the LEXIS Portal Layer (via an appropriately-defined API). Also these interfaces, as the components mentioned in Section 4.3.1, can be regarded as stable.

4.4 DDI SERVICE

The LEXIS DDI Service module comprises all components of the Distributed Data Infrastructure (DDI) plus the WCDA (Weather and Climate Data API) component which is indispensable for storing, retrieving and managing curated, weather and climate related data within LEXIS. As mentioned in Deliverable D2.3 [1], the Distributed Data Infrastructure provides a unified access to LEXIS data without any strong focus on application domains. DDI services are fully up and running, which is a key prerequisite for approving MS6.

Below, we will begin with a description of WCDA deployment. All the core DDI services are then part of three further sub-modules – iRODS, APIs, and PERFORMANCE MONITORING.

4.4.1 Submodule "WCDA (API)"

The WCDA has been deployed at IT4I, LRZ and the ECMWF Data Centre. It serves original or modified (cut, interpolated, etc.) weather and climate data with high performance, on API requests with a domain-specific, specialised syntax. In its backend, it uses a highly-optimised, custom data stored by ECMWF, the so-called Field Data Base (FDB, see Deliverable D7.5 [14]). After our initial successful deployment, updates of the component may follow upstream developments within WP7, pushed in particular by the weather and climate data experts from ECMWF.

4.4.2 Submodule "iRODS"

The iRODS module consists of the deployment of iRODS and EUDAT services. As an initial step, core components were deployed and tested at the participating centres. Up to M15 of the project, iRODS and different EUDAT components (B2HANDLE, B2SAFE, and B2STAGE) had been deployed in a scheme capable of warranting high availability. Issues related to connecting iRODS with Keycloak were resolved, and the connection of the DDI to the LEXIS AAI was established. To support data findability, a production instance of the HANDLE system was deployed at each centre. A Persistent Identifier (PID) Prefix was acquired from the European Persistent Identifier Consortium. Each instance can assign PIDs to datasets and the PID values are synced with the international HANDLE ecosystem.

The progress since M15 has focused on resolving issues in our iRODS-B2SAFE setup. iRODS was upgraded from 4.2.3 to the latest version 4.2.8 (at the time of deployment). The upgrade solved some legacy issues we had in the previous version, but created a few other issues that were taken care of. B2SAFE was redeployed and replication of big datasets was tested.

The iRODS module is envisaged to remain at its current version for the rest of the LEXIS project.

4.4.3 Submodule "APIs"

DDI APIs provide the interfaces needed by other LEXIS components to connect to the iRODS/B2SAFE deployment, handle and move data. The principal APIs handle data search/discovery, up- and download, data harmonisation (Encryption/Compression), data staging, and monitoring.

The Data Search API allows to discover datasets according to their DataCite-like [15] metadata stored in the DDI iRODS system. The staging API is normally not addressed by the users themselves; instead, the orchestrator interfaces with it to move data between any source and target storage-system combination available, considering the participating centres. The staging API connects to the encryption and compression APIs, providing the possibility to encrypt and compress data (or the inverse processes) before moving it to its final destination.

Further auxiliary APIs developed include a SSHFS-mountpoint API, a replication API to trigger data replication, and a data size API to guide the orchestrator and its Dynamic Allocation Module (Section 4.3.4).

The LEXIS DDI APIs are programmed within the project, and new releases focus on adding functionalities needed by the LEXIS portal and orchestrator. Four releases of DDI APIs have been made (with exception of the monitoring API – one release), and new releases about every other month are planned in order to provide features for, e.g., data flow optimisation.

4.4.4 Submodule "PERFOMANCE MONITORING"

Besides the functional monitoring tests (Section 4.8), WP3 includes the development of performance monitoring methodology and tests within the scope of Task 3.5. These scripts are currently collected, archived, and set up for automated periodical runs, and will be included in future platform releases as this submodule.

4.5 LEXIS AAI

LEXIS AAI stands for LEXIS Authentication and Authorisation Infrastructure. In a few words, this is the part/component of the LEXIS platform in charge of unifying and federating identity and access for the LEXIS Platform. This component is the source of truth with respect to identity and authorisation for any other LEXIS service/component allowing all of them to work together and inter-communicate. The LEXIS AAI is the central pillar of the Zero-Trust Architecture within LEXIS, where any components do not trust each other, but refer to the LEXIS AAI to securely validate each identity and access.

The LEXIS AAI has been deployed in production mode according to the initial plan (see Deliverable D4.1 [16] and Deliverable D4.5 [17]). The deployment is based on a “cross-datacentre replication mode” installation of the Keycloak OpenSource IAM solution (used as upstream project by RedHat for their Single Sign On solution).

The LEXIS AAI deployment and its configuration will be described in more detail in Deliverable D4.7 [18] about Centralised (i.e. unified, but federated) AAI. Here, some basic overview is given. The Keycloak installation (Submodule KEYCLOAK & KEYCLOAK API) consists of five different subcomponents (Distributed Database Cluster, Cache Cluster, Keycloak Cluster, HAProxy Cluster and Keycloak Library Extension). The resulting system is well-monitored (Submodule MONITORING) to detect any outages, even if improbable. On the mid-term, the system will be kept as described below, installing only security updates with further releases.

4.5.1 Submodule “KEYCLOAK & KEYCLOAK API”

Distributed Database Cluster

The Database cluster is based on Galera cluster v25.3.33 (Galera 3 replication library) on top of MariaDB v10.3.29, as these are latest major versions mentioned in the Keycloak official documentation.

The Galera cluster is set up in active/active mode and synchronised over the LEXIS site-to-site VPN between IT4I and LRZ. As for now, the database cluster has only one node in each supercomputing centre, but this can be extended later according to needs for scalability. In case of VPN issue or any other type of issue leading to a cluster de-synchronisation, it is possible to re-synchronise the full cluster by stopping the node and restarting first the latest updated node and then the other node.

Cache Cluster

The cache cluster is based on Infinispan version 9.4.20, which was latest of the tested versions for Keycloak 12.0.4 from the Keycloak documentation. The cache cluster is set up in a replication mode allowing to store short-lived or frequently changing metadata such as user sessions. It is worth to mention that the cache does not have a persistent mode, meaning that all data in the cache are not expected to persist after cluster restart.

Similarly to the Distributed Database Cluster, the cache cluster is deployed with one node in each supercomputing centre which are synchronised over the LEXIS site-to-site VPN. In case of VPN or other issues leading to a cluster de-synchronisation, everything can be restored to normal by just restarting the cache server on each node. In order to minimise the number of servers involved in the setup, the Infinispan cache software is installed on the database server at each centre.

Keycloak Cluster

The Keycloak cluster is based on Keycloak version 12.0.4 that was latest version available when we configured the LEXIS AAI production system. It is deployed on one node in each supercomputing centre. As a matter of fact, such a Keycloak cluster can be extended by adding the same number of nodes in each data centre, and as long as the node will be properly added all the cluster will synchronise automatically.

The cluster synchronises through both the database and cache clusters. Each Keycloak installation can be updated with the latest version independently without breaking the full system, and no downtime would be expected due to the frontend proxies (see next paragraph). In order to ensure smooth updates, however, we would remove nodes from cluster, update them and then add them again to the cluster. For development purpose and continuous integration, we have planned to use a containerised version of Keycloak with standalone deployment and easy spin-up.

However, no immediate updates except for security fixes are envisaged as of writing this text.

HA Proxy Cluster

In each supercomputing centre, one HAProxy instance has been set up in order to provide connectivity to the Keycloak cluster deployed across the centres, giving priority to the local Keycloak endpoint. In IT4I, the priority is given to IT4I Keycloak, with a fall-back to the one deployed in LRZ, and the exact opposite configuration is deployed at LRZ. The two HAProxy instances are then set up with equal weight in the main Domain Name System (DNS) entry for the AAI of the LEXIS platform.

Keycloak Library Extension

In order to facilitate the integration of the LEXIS portal with Keycloak and the implementation of the Role-Based Access Control (RBAC) matrix from LEXIS, we have created an extension to the Gocloak Library (basically a Golang implementation of Keycloak API) used by the LEXIS Portal. For instance, this allows us to map high-level user permissions and access rights for the LEXIS Portal to fine-grained access rights based on the group attribute in Keycloak. The group attribute is then generally re-used by all Keycloak Clients that are the endpoints for all LEXIS Components (LEXIS DDI, LEXIS Orchestrator, etc.).

4.5.2 Submodule “MONITORING”

For monitoring purposes, the LEXIS AAI mainly relies on the Prometheus Node exporter for system metrics and then on several specific exporters such as the MySQL or Keycloak exporters from official Prometheus or Keycloak repositories. This allows the LEXIS AAI to expose metrics to a Prometheus scraper that can easily access them and store them for further exploitation by the graphical interface of the LEXIS monitoring system. These components have been deployed in a one-release manner, probably not requiring major updates on the mid-term.

4.6 LEXIS BACKEND SERVICES

LEXIS Backend Services is a pseudonym for a collection of various modules which provide capabilities to the LEXIS portal front end to connect to the orchestration, data or infrastructure layers of LEXIS. The services/submodules which make up the LEXIS Backend Services are:

- PORTAL/BackEnd Service API (with interfaces to approval system, accounting and billing),
- USERORG SERVICES.

And the SERVICE INTERFACES (as summarized in Figure 2):

- ALIEN4CLOUD (A4C) INTERFACE,
- DATASETS INTERFACE,
- APPROVAL SYSTEM INTERFACE,
- ACCOUNTING/BILLING INTERFACE.

The detailed description of each of these services can be found in Deliverable D8.1 [19]. At the time of writing, all services have been deployed at the two original LEXIS sites (IT4I/LRZ) and may see future updates for optimisation. This is made easy through a full-fledged CI/CD setup.

4.6.1 Submodule "BACKEND SERVICE API"

The BACKEND SERVICE API is the central service which captures interaction of the portal with LEXIS services. The portal UI elements only interact with the API service and all other complexities are hidden behind this. Since M15, the service was constantly improved and modified, and will probably be subject to further releases with optimisations. Better error condition differentiation was added, CI/CD integration was finalised, and authorisation checks to ensure user authentication and authorisation following the No-Trust principle were added.

4.6.2 Submodule "USERORG SERVICES"

The USERORG SERVICES submodule encapsulates database components used to store LEXIS user data and relations among the users (e.g. organisational relations for adjusting permissions accordingly). It was currently extended to, e.g., enable linking of HPC and cloud resources to LEXIS registered organisation in a more versatile manner. This service was also adapted for CI/CD automation, refactored for a more reliable interaction with the database and the AAI system, and support for the authorisation checks was integrated as well.

4.6.3 Submodule "A4C INTERFACE"

The A4C INTERFACE service provides workflow management services to the LEXIS Portal. The service encapsulates database components used to store LEXIS Workflow and Workflow Execution data. The A4C Interface service provides an interface between LEXIS Portal and Alien4Cloud, the application chosen to manage Workflow lifecycles. The service has been extended to support all required functionality to allow LEXIS Workflows and Workflow Executions to be created, ran, and managed from the LEXIS Portal, using the LEXIS Workflow Templates available. It has also been adapted to support CI/CD automation and end-point authorisation checks.

4.6.4 Submodule "DATASETS INTERFACE"

A Dataset Management GUI (Section 4.7.2) in the portal allows the user to find, manage, up- and download data in the LEXIS DDI and to use them in workflows. The corresponding interface behind this was extended since M15 to include all the new capabilities exported by the DDI Service, including all updates in the staging, replication, and data size APIs, and the addition of the new encryption API. It was also adapted to support the CI/CD automation and is in the process of getting the support for additional, fine-grained authorisation checks in place.

4.6.5 Submodule "APPROVAL SYSTEM INTERFACE"

The LEXIS Approval system, allowing users to request resources and having them approved, is described in detail in Section 4.10. The system corresponds to an interface in the portal API and to a user interface part in the portal, where the user can issue requests in an easy way. This interface was also adapted for the CI/CD automation, and it is currently being refactored to include the support for the authorisation checks.

4.6.6 Submodule "ACCOUNTING/BILLING INTERFACE"

The Cyclops ACCOUNTING/BILLING INTERFACE was extended since M15 to allow detailed data fetching and presentation within the Portal FrontEnd. The interface was extended to correlate HPC and cloud resources for registered organisation in order to enable correct data consolidation at Cyclops engine.

4.7 LEXIS FRONTEND

The main components of the LEXIS portal are represented by the modules of this group. We first describe the technical modules "LEXIS PORTAL BACKEND" and "AAI CONNECTOR", and then the modules immediately "visible" to the portal user: DATASETS view, WORFKLOW(S) management, RESOURCES AND BILLING, and USER AND USERGROUP MANAGEMENT.

All modules use WP8's CI/CD system with go.mod to ensure an interface match between APIs and frontend and allow for an agile sub-release scheme, reacting as an appropriate to user demands within the Open Call. Major releases are described in Deliverables [19, 20], according to the LEXIS Description of Action.

4.7.1 Submodule "LEXIS PORTAL BACKEND" and "AAI CONNECTOR"

The Portal BackEnd is responsible for answering all user requests via the portal API, and is also responsible for starting the initial authentication work flow via Keycloak (the "heart" of the LEXIS AAI) and for keeping the session state. To this end, it employs an "AAI CONNECTOR".

The backend service was re-factored to enable seamless CI/CD automated build support and deployment capabilities, and several bug fixes were carried out to improve functionality and reactiveness by M30.

4.7.2 Submodule "DATASETS"

This module provides the user interface for managing data in the LEXIS Portal.

The module has tracked the changes from LEXIS WP3 to provide additional functionality. Changes included the inclusion of additional metadata fields (including custom metadata and EUDAT information), upload modes (tus resumable file uploads [21]), the addressing of further APIs (GridFTP access control, SSHFS export and staging), data size query, and the offering of downloads via multipart zip.

4.7.3 Submodule "WORKFLOWS"

The Workflows module of the LEXIS Portal is used for all LEXIS Workflow and LEXIS Workflow Execution Management. It enables the creation and management of workflows from templates, as well as the creation, monitoring and management of LEXIS Workflow Executions (actual instances). Creation and input parametrisation of LEXIS Workflow Executions from LEXIS Workflows, dataset linking and handling of complex input types have been implemented. LEXIS Workflow Execution management has been extended to now include show progress in detail, as well as the execution location for applicable steps, as determined by the orchestrator.

4.7.4 Submodule "RESOURCES AND BILLING"

The RESOURCES AND BILLING module of the LEXIS Portal is a key component that interconnects the track of Projects and HPC Resources done in the USERORG SERVICES, the usage, accounting, and billing processing which Cyclops Billing Engine provides and the resources request and state that the Approval Service manages. Hence it enables the users of the Portal to create projects, request resources for them, and keep track of the usage state and credit balance.

Since M15, the initially basic module has been evolving in order to enable the user to have full control over the aforementioned capabilities.

4.7.5 Submodule "USER AND USERGROUP MANAGEMENT"

The interface of the portal identified with this module allows LEXIS administrators and persons with management roles (mostly within organisation scope) to create users and user groups. As the submodules already discussed, it is continuously being improved and released within the common CI/CD scheme of the LEXIS portal framework.

4.8 MONITORING SERVICES

The LEXIS monitoring system provides metrics and tests to check the health and performance of LEXIS systems, and an alerting system that reports malfunction and downtime in any components.

Notification channels and alerts have been set up to inform the administrators in case of consecutive failures in some of the services.

4.8.1 Submodule “MONITORING CORE“

The core of the LEXIS monitoring system is based on Elasticsearch, Logstash and Grafana, and provides center-specific and LEXIS overview dashboards. While the monitoring system has been largely consolidated, a few future releases may further optimise the usefulness for LEXIS operation.

4.8.2 Submodule “SYSTEM TESTS“

For monitoring the DDI and other systems, standardised test cases were developed using the Robot framework. The tests are executed every 20 minutes and ensure that the services are available. Furthermore, standard Prometheus exporters were deployed in all machines at the participating centres. The exporters produce time series data concerning the operation of different systems. The test suite has been consolidated as of M30 and may only have a few adjustments, but probably no major release.

4.9 ACCOUNTING AND BILLING SERVICE

This module comprises the accounting and billing engine and a small AAI CONNECTOR submodule in order to check the authorisation of requests. Its releases within LEXIS will somewhat follow upstream releases of the Cyclops billing engine. Currently, a version providing all basic functionalities needed has been deployed.

The LEXIS accounting and billing service is based on variant developed from the Cyclops Open-Source accounting and billing framework for large-scale distributed systems [22]. As described in Deliverable D8.1 [19], the accounting and billing service being integrated and extended for the LEXIS use case is cloud-native by design and developed as a set of microservices.

The progress since M15 has centred around re-factoring the Cyclops modules including plan manager, customerdb, and credit manager services to support the requirements of the LEXIS orchestration engine. Specifically, the credit manager was extended to track for core-hour credits as well as cash credits linked to an organisation’s credit account.

The remaining core services - UDR, CDR and billing also got revamped, the use of Kafka was further optimised, and OpenStack collector code dependent libraries were upgraded to improve data collection times. The Cyclops engine was adapted to LEXIS user-org service’s linking of cloud and HPC resources linked to the LEXIS organisation entities.

4.9.1 Submodule “AAI CONNECTOR”

This submodule connects the Accounting and Billing system to the LEXIS AAI, i.e. the Keycloak IAM solution. Within LEXIS, it is released with the billing system.

4.10 APPROVAL SERVICE

The Approval Service is a small service consisting of three other micro-services:

- Front-end web user interface in React.js (submodules “HEAppE´s MANAGE & STATUS REPORT” and “AAI CONNECTOR),
- Back-end API/application written in Node.js framework (with necessary emailing service included) – submodules “API”, “HPC”, and “CLOUD”, and a
- MS-SQL database (particularly Microsoft® SQL Server® 2019 Express is being used) – submodules “HPC” and “CLOUD”.

The main goal of the Approval Service is to provide a capability of accepting, on the one hand, or – on the other hand – of rejecting a request for particular computing resource by a LEXIS platform user with (in-)appropriate privileges. This depends on the organisation and grants of the users, originating from various external stakeholders: research organisations, industry, SMEs and start-ups and other third-party organisations.

All components of the Approval Service have been written from the scratch just for the purpose of LEXIS project. They are all set in separate Docker containers and use CI/CD technologies. Thus, they are ready to be deployed basically seamlessly in any other computing centre. Updates are made to fix bugs or add functionalities (which, however, in the current project state, will be hardly the case).

4.10.1 Submodules “HEAppE´s MANAGE & STATUS REPORT” and “AAI CONNECTOR”

These components allow for managing HEAppE instances and permission status of certain users. Authorisation to this service is handled via the LEXIS AAI, requiring a connector. In the front-end application itself, accepting/rejecting a request will be possible only from within of each centre and only by persons with “principal investigator” privileges or, as is being discussed, other types of elevated privileges. The status of the implementation of the AAI CONNECTOR is currently in-progress; the FrontEnd web user interface is finished and working.

4.10.2 Submodules “API”, “HPC” and “CLOUD”

These submodules encapsulate the handling of HPC and Cloud resources within the back-end application, and its API via which it connects to the frontend described in the previous subsection. The application is deployment ready except for some implementation parts regarding the approval confirmation via e-mail.

This database backend to the application is set up as well.

5 SUMMARY

With this deliverable, we summarised the LEXIS technology deployment by M30 (i.e. six months before project end). Key parts of LEXIS technology have been appreciated by the European Innovation Radar [23] and published (e.g., [24], [25]). We also would like to point out contributions on WP6 and WP7 workflows to conference papers (e.g., [26], [27]), a poster (on WP3, WP4, and WP7 collaboration) at the prestigious SC20 Supercomputing conference, and presentations in the scope of various conferences/workshops (e.g., Data Week 2021 | BDVA, EGI Conference 2020, ISGC 2021). We are confident that the Industrial and Scientific use cases within the scope of the LEXIS Open Call will lead to further dissemination and appreciation of the platform on the way to sustainably establishing the LEXIS platform.

The deliverable has laid out a professional release management as established within LEXIS WP2 and assessed the status of the Platform with respect to that management concept, focusing on so-called LEXIS modules as managed and released entities and their submodules. Release plans have been made for relevant core components of LEXIS, as a basis for sustainability of the platform and its porting to further data centres as partners.

REFERENCES

[1] LEXIS Deliverable, *D2.3 Report of LEXIS Technology Deployment – Intermediate co-design.*

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

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

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

[5] Bull Atos Technologies, "YORC," [Online]. Available: https://ystia.github.io/. [Accessed 24 July 2021].

[6] A. B. Yoo and M. A. Jette, "SLURM: Simple Linux Utility for Resource Management," in *Job Scheduling Strategies for Parallel Processing. JSSPP 2003. Lecture Notes in Computer Science, 2862*, Springer, Berlin, Heidelberg, 2003.

[7] R. L. Henderson, "Job scheduling under the Portable Batch System," in *Job Scheduling Strategies for Parallel Processing. JSSPP 1995. Lecture Notes in Computer Science, 949*, Springer, Berlin, Heidelberg, 1995.

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

[9] T. Preston-Werner, "Semantic Versioning 2.0.0," [Online]. Available: https://semver.org/spec/v2.0.0.html. [Accessed 23 July 2021].

[10] GitHub, Inc., "Understanding the GitHub flow," [Online]. Available: https://guides.github.com/introduction/flow/. [Accessed 23 July 2021].

[11] O. Lacan, "Keep a Changelog Version 1.0.0," [Online]. Available: https://keepachangelog.com/en/1.0.0/. [Accessed 23 July 2021].

[12] Leibniz Supercomputing Centre (LRZ), "Compute Cloud - Leibniz-Rechenzentrum (LRZ) Dokumentation," [Online]. Available: https://doku.lrz.de/display/PUBLIC/Compute+Cloud.

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

[14] 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.*

[15] DataCite Metadata Working Group, "DataCite Metadata Schema 4.3," DataCite - International Data Citation Initiative e.V., 16 Aug. 2019. [Online]. Available: https://schema.datacite.org/meta/kernel-4.3/. [Accessed 1 Oct. 2019].

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

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

[18] LEXIS Deliverable, *D4.7 Centralized AAI: Coverage of All Significant Systems.*

[19] LEXIS Deliverable, *D8.1 First Release of LEXIS Portal (will include report).*

[20] LEXIS Deliverable, *D8.2 Second Release of LEXIS Portal.*

[21] Transloadit-II GmbH, "tus – Open Protocol for Resumable File Uploads," 2018. [Online]. Available: https://tus.io/. [Accessed 6 Apr. 2021].

[22] Cyclops Labs GmbH, "Cyclops accounting and billing engine," [Online]. Available: https://www.cyclops-labs.io/page/it-accounting-and-billing. [Accessed 24 July 2021].

[23] European Commission, "Innovation radar > Innovation > Data Infrastructure based on iRODS/EUDAT-B2SAFE & REST APIs," [Online]. Available: https://www.innoradar.eu/innovation/38139. [Accessed 20 July 2020].

[24] S. Hachinger, M. Golasowski, J. Martinovic and et al., "Leveraging High-Performance-Computing and Cloud Computing with Unified Big-Data Worfklows: The LEXIS Project," in *Technology and Applications for Big Data Value book*, Springer Nature Switherland AG, 2021.

[25] O. Terzo and J. Martinovič, Eds., HPC, Big Data, AI Convergence Towards Exascale: Challenge and Vision, Boca Raton, FL: CRC Press, 2021 (in production, ISBN 9781032009841).

[26] T. Goubier, J. Martinovič, P. Dubrulle, L. Ganne, S. Louise, T. Martinovič and K. Slaninová, "Real-Time Model of Computation over HPC/Cloud Orchestration - The LEXIS Approach," in *Proceedings of the 14th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2020). Advances in Intelligent Systems and Computing, vol. 1194*, Łódź, Poland, 2021.

[27] A. Parodi et al., "LEXIS Weather and Climate Large-Scale Pilot," in *Proceedings of the 14th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2020). Advances in Intelligent Systems and Computing, vol. 1194*, Łódź, Poland, 2021.

[28] A. Scionti et al., "Cloud and Big-Data Convergent Architectures: the LEXIS Approach," in *Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019). Advances in Intelleigent Systems and Computing, vol. 993*, Sydney, Australia, 2020.