Large-scale EXecution for Industry & Society

Deliverable D2.5

Final assessment of the co-designed LEXIS architecture

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	D2.5 | Final assessment of the co-designed LEXIS architecture
RESPONSIBLE AUTHOR	Giacomo Vitali (LINKS)
WORKPACKAGE ID | TITLE	WP2 | LEXIS Requirement Definition and Architecture Design
WORKPACKAGE LEADER	Atos
DATE OF DELIVERY (CONTRACTUAL)	31/12/2021 (M36)
DATE OF DELIVERY (SUBMITTED)	05/01/2022
VERSION | STATUS	V1.1 | Final
TYPE OF DELIVERABLE	R (Report)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	LINKS; LRZ; O24; IT4I; CYC; EURAXENT
INTERNAL REVIEW	Ciarán O'Rourke (ICHEC); Paul Mynarsky (O24)

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

DOCUMENT VERSION

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Table of Contents and filled-in draft (first proposal) created.	20/10/2021	Giacomo Vitali (LINKS)
0.2	Defined ToC and first content for Chapter 2 and definition of contributions.	08/11/2021	Giacomo Vitali (LINKS)
0.3	All the contributions needed from partners have been received and implemented First version of the document ready for internal review.	07/12/2021	Giacomo Vitali (LINKS); Mohamad Hayek, Rubén García Hernández, Stephan Hachinger (LRZ); Frédéric Donnat (O24); Jan Křenek, Václav Svatoň, Jan Swiatowski (IT4I); Alberto Scionti (LINKS); Diego Martin (CYC); Marc Derquennes (EURAXENT)
0.4	Implemented internal reviews' comments.	22/12/2021	Giacomo Vitali (LINKS); Frédéric Donnat (O24)
1.0	Final check of the deliverable.	04/01/2022	Jan Martinovič, Kateřina Slaninová (IT4I)
1.1	Update according to the comments from the final check.	05/01/2022	Marc Derquenness (EURAXENT); Giacomo Vitali (LINKS)

GLOSSARY

ACRONYM	DESCRIPTION
AAI	Authentication & Authorization Infrastructure
A4C/ALIEN4CLOUD	Application LIfecycle ENablement for Cloud (cf. https://alien4cloud.org)
CDI	Collaborative Data Infrastructure
DDI	Distributed Data Infrastructure
EUDAT	European Data Infrastructure
HEAPPE	High-End Application Execution
HPC	High Performance Computing
IAM	Identity and Access Management
LL	Lesson(s) Learned
NVME	NVM (non-volatile Memory) Express, usually used as interface to SSDs
NVMEOF	NVMe-Over-Fabrics
NVRAM	Non Volatile Random Access Memory
OIDC	OpenID Connect protocol

POSIX	The Portable Operating System Interface
RBAC	Role-based Access Control
RDMA	Remote Direct Memory Access
SAML	Security Assertion Markup Language
SBF	Smart Bunch of Flash
SME	Small & Medium Enterprises
TSUNAWI	Tsunami software simulator; origin, propagation and physical impacts
WCDA	Weather and Climate Data API
WP	Work Package
YORC	Ystia Orchestrator (cf. https://github.com/ystia/yorc)

TABLE OF PARTNERS

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

The LEXIS project is developing solutions to integrate hardware and software components from HPC, Cloud and Big Data domains into a powerful computing and Data Management platform, including validation within real-life scenarios (Aeronautics, Earthquake & Tsunami, Weather & Climate) and providing easy access to Industries, SMEs and Academia through the LEXIS Portal.

By project month M36, the LEXIS project has deployed the final set of technical solutions which compose the platform (TRL near to 8). The evaluation of existing solutions together with existing technical constraints has been done in Deliverables D2.1 [1] and D2.2 [2], while the intermediate co-design, together with the first set of Lesson Learned (LL), have been reported in D2.3 [3]. This document complements the first set of LL, as additional issues and challenges were encountered during the implementation and integration processes. Moreover, it describes the final LEXIS Platform architecture with a focus on changes and solutions adopted to face the challenges identified in the LL and the implementation of the pilots use-cases, as well as the platform assessment within the pilots´ context, with the goal of verifying the achievement of the technical requirements.

Position of the deliverable in the whole project context

This deliverable is the final report for Task 2.2 - Co-design and Lesson Learned. It summarises the final set of key components of the LEXIS architecture, their relationships and the additional LL which were identified during the second part of the co-design processes and the platform deployment.

The main contributors of this deliverable are the “technical work packages”, namely WP2, WP3, WP4 and WP8:

- LINKS as the coordinator of co-design tasks and work package leader (WP4),
- Atos as the work package leader as well as for Burst-Buffers, Remote Visualization, Orchestration and FPGA related topics (WP2),
- IT4I as one federated HPC service provider and the LEXIS project coordinator,
- LRZ as the other federated HPC service provider and work package leader (WP3),
- O24 for security related aspects and CYC for the portal related topics and work package leader (WP8).

Besides, LINKS, Atos, LRZ and EURAXENT contributed to the section regarding LL.

Description of the deliverable

Deliverable D2.5 first provides the list of additional LL during the final deployment of the platform’s components which are described in Section 2. Then, the final LEXIS architecture is described in detail, focusing on the components and solutions which have been implemented in the platform to comply with challenges and issues identified during the deployment and validation phase of the LEXIS Platform (Section 3).

The deliverable closes with a short summary of the technology achievements that have been made and of the overall platform assessment (Section 4).

1 INTRODUCTION

About the present deliverable (D2.5)

WP2 - *LEXIS Requirements Definition and Architecture Design* evaluates the technical requirements of large-scale pilots and the existing hardware useful for the LEXIS project with the aim of enhancing the technologies to allow easy use and running of HPC/Cloud/Big Data applications. Two submitted Deliverables D2.1 [1] and D2.2 [2] from WP2 - Task 2.1 *Infrastructure Evaluation and Key Technology Evaluation* have reported the key existing and new parts of the LEXIS Platform, while Deliverable D2.4 [4], related to WP2 - Task 2.3 *Infrastructure Set-up, Roll-outs, Validation and Benchmarking*, described the LEXIS architecture as of M30, its modules and the LEXIS release management. Then, Deliverable D2.3 [3] reported, in the context of Task 2.2 *Co-design and Lesson Learned*, the key components of the LEXIS architecture and included Lessons Learned (LL) from the co-design process during the first half of the project. As the final report on this task at M36, this deliverable completes the existing set of LL in Deliverable D2.3 [3] with new ones encountered during the second part of the project, describes the final LEXIS architecture and provides its assessment in the context of the pilot test-beds. The platform's validation process as well as its results will be reported in the Deliverable D2.6 [5].

From the deeper understanding of requirements and capabilities of underlying technologies to the actual deployment, various challenges were encountered, and the corresponding solutions were proposed and have been implemented, where possible. The LL recorded some of these best practices, differences between expected objectives and achieved ones, and challenges, which served for improving the implementation of newly proposed LEXIS approaches, preventing failures, and keeping the LEXIS development on schedule. Moreover, the deployment phase of the platform and the use-cases actively involved all stakeholders in the validation process to ensure that the results met their target and were usable.

The LEXIS Platform provides a three-layer architecture which shows how hardware and software technologies interact with each other. A brief overview of our technology solutions has been reported in Deliverable D2.3 [3]. The current report contains the updated final architecture diagram showing the key components and their connections, while highlighting the solutions put in place to comply with challenges and issues identified during the deployment and validation phase of the LEXIS Platform.

Structure of the document

The document is structured as follows:

- The first part (Section 2) will complement the inventory of LL reported in Deliverable D2.3 [3] with the ones identified during the second part of the co-design and deployment phases, as well as during the LEXIS Open Call (LEXIS OC) related activities,
- The second part of the deliverable (Section 3) will describe the final LEXIS architecture and will focus on changes and solutions adopted to face the issues and challenges encountered during the project's lifetime. Moreover, the platform assessment is provided in order to verify the achievement of all the technical requirements defined at co-design phase and by the pilots´ applications.
- Finally, a summary will be provided in Section 4.

2 LESSONS LEARNED ON TECHNICAL DEVELOPMENT AND INTEGRATION ACTIVITIES

The co-design process focused on activities aiming at identifying and documenting the most significant LL resulting from the implementation and deployment activities until the official end of the project. A first set of LL, which have been identified during the co-design and the first implementation phase, have been reported in Deliverable D2.3 [3], together with the detailed description of LL motivation and methodology. However, the platform deployment and the validation process have brought to light additional challenges and issues between M18 and M36, which had been analysed and solved with the goal of minimizing the impact on the project objectives and timeline.

In fact, some key components stood out, some showing good practices, while other ones displayed differences between foreseen objectives and achieved ones, or even serious challenges during the deployment process.

Besides purely technological aspects, we also identified potential LL in organizational aspects, such as methodological approaches for interaction and cooperation between organizations and consistent process design for bringing the platform to production and accept external users in compliance with local and European regulations.

In next sections, we present the additional set of LL.

2.1 TECHNOLOGICAL RELATED LESSONS LEARNED

In this section, we present LL dealing with technology.

2.1.1 AAI – Harmonization of identity management in 2 federated HPC providers

After the implementation of the federated identity and access management system, a common user lifecycle model had to be designed and implemented in order to cope with such a system. The strategy that was adopted is described in the LL at Table 1. Moreover, the authorization strategy for the LEXIS components has been implemented so that it complies with the Zero Trust principle, as reported in Table 2.

Finally, during the deployment and maintenance of the AAI system, some issues were encountered in the synchronization of the AAI clusters in IT4I and LRZ. This produced the LL described in Table 3.

AUTHOR	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	30/07/2021	User lifecycle	Solution for ensuring consistency and proper management during user lifecycle.
QUESTIONS			**ANSWERS**
What activities led to this LL?			As partners started developing the LEXIS Platform and preparing for the Pilot workflows implementation at the same time, the platform components have not been fully integrated yet and we had to create users in different places for various LEXIS components (AAI, Portal, Distributed Data Infrastructure - DDI).
What are the challenges or issues?			The main issue with being able to create users in different places is the need for synchronization between those. Usual best practices (both software and security) recommend to only have one component in charge of IAM that all others components just refer to.

What are the envisioned solutions?	The solution envisioned and put in place consists in having a unique source of truth for IAM and all other components synchronize with it.

Table 1 LL on the User lifecycle

AUTHOR	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	**10/07/2021**	**Fine-grained authorization for all LEXIS components**	Solution to ensure proper authorization for all LEXIS components and comply with Zero Trust architecture.
QUESTIONS			**ANSWERS**
What activities led to this LL?			The authorization is handled using the OAuth 2.0 protocol and JWT tokens that are passed to different LEXIS components. In this area we needed to revoke some access from one component without revoking from all components. For instance, we needed to be able to execute some long running tasks on behalf of the user, like long term running HPC jobs or data transfer tasks without the need for the user to keep his browser open and have an active session in the LEXIS Portal.
What are the challenges or issues?			The challenges are twofold: • Revoking partial access, • Not compromising overall security.
What are the envisioned solutions?			The envisioned solution was: • To ensure least privileges by revoking grants for specific components, • To enable the feature to exchange tokens between LEXIS components so that each LEXIS component has its own token containing the permissions for itself. As a side, we also enabled the offline-token (of type refresh token) for long running tasks.

Table 2 LL on the Fine-grained authorization for all LEXIS components

AUTHOR	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	**10/07/2021**	**LEXIS AAI clustering**	LEXIS AAI is based on an active/active cluster running in both IT4I and LRZ.
QUESTIONS			**ANSWERS**
What activities led to this LL?			During development and maintenance of both IT4I and LRZ HPC centres we had some cluster de-synchronization and we manually needed to re-synchronize the database cluster and then restart the cache and Keycloak clusters.

What are the challenges or issues?	The challenges are twofold: • Maintenance in HPC centre, • Site-2-site VPN and network issue/update in each HPC centre.
What are the envisioned solutions?	The solution was to use set active/passive mode for LEXIS AAI clustering and update our HAProxy reverse proxy to redirect on active node.

Table 3 LL on the LEXIS AAI clustering

2.1.2 DDI – Delegation of DDI Security Operations to AAI

The LEXIS Distributed Data Infrastructure (DDI) provides a unified view on data stored in the LEXIS Platform for all users and from all systems in the federated computing centres. We chose the "integrated Rule-Oriented Data System" (iRODS) as the infrastructural framework for distributed data management. This system integrates well with the European Collaborative Data Infrastructure (CDI) EUDAT (as it is the basis of EUDAT B2SAFE).

During the DDI test phase, we encountered an issue related to data replication caused by conflicting EUDAT and DDI data replication policies. The detailed description of this LL is provided in Table 4.

AUTHOR	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Mohamad Hayek	**18/10/2021**	**Data replication permissions and policy**	The EUDAT B2SAFE module does not allow users who don not directly own the data to replicate. This is a problem when multiple users share a project directory in the DDI. The permissions at project level on the DDI had to be updated.
QUESTIONS			**ANSWERS**
What activities led to this LL?			Testing replication with multiple users led to this discovery.
What are the challenges or issues?			The main challenge was to adapt the permissions for all projects at LRZ and IT4I and to update the iRODS API code while trying to keep the user permissions as intended in the RBAC matrix.
What are the envisioned solutions?			Adapt the DDI permissions at project level to allow replication.

Table 4 LL on the Data replication permissions and policy

2.1.3 DDI – Deployment and Upgrade

During the DDI infrastructure's planned maintenance, some issues have been encountered when upgrading the iRODS version to the latest one, by breaking the EUDAT services and authorization systems. These LL are detailed in Table 5 and Table 6.

AUTHOR	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Mohamad Hayek	**15/02/2021**	**EUDAT services redeployment**	iRODS upgrade breaks EUDAT services. The services had to be reinstalled completely.
QUESTIONS			**ANSWERS**

What activities led to this LL?	This was caused by updating the version of iRODS from 4.2.3 to 4.2.8.
What are the challenges or issues?	Some dependencies in the EUDAT services scripts are hard-coded. We had to update the dependencies manually in the installation scripts and redeploy. The new dependencies have been pushed to LEXIS GitHub account.
What are the envisioned solutions?	EUDAT must make the installation scripts flexible by scanning through the installed iRODS external packages and choosing the latest ones.

Table 5 LL on the EUDAT services redeployment

AUTHOR	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Rubén García	**15/02/2021**	**iRODS - OpenID redeployment**	OpenID is not officially supported in iRODS 4.2, integration plugin and token service has to be modified manually for each minor release.
QUESTIONS			**ANSWERS**
What activities led to this LL?			iRODS upgrade from version 4.2.3 to 4.2.8 breaks OpenID plugin compatibility.
What are the challenges or issues?			OpenID plugin for iRODS has to be modified due to breaking changes in iRODS plugin interface.
What are the envisioned solutions?			The plugin source code was modified and rebuilt to work with iRODS 4.2.10. Compatibility with newer versions is not ensured. iRODS consortium observes changes made by the LEXIS project and is considering pulling them to the official distribution.

Table 6 LL on the iRODS - OpenID redeployment

2.1.4 Orchestrator – Workflow and Resource Management

During the pilots´ workflow implementation phase, some LL were identified in relation to specific jobs' characteristics, e.g. supporting long-term jobs whose execution could extend beyond the maximum available wall time, and requirements, such as time scheduled workflow executions. The details on the challenges and relative solutions to these LL are provided in Table 7 and Table 8.

AUTHORS	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Jan Křenek/ Václav Svatoň	**15/07/2021**	**Long-term job support**	HPC cluster walltime limits are not compatible with very long computation jobs.
QUESTIONS			**ANSWERS**
What activities led to this LL?			Some applications of the LEXIS Platform users must execute long term HPC jobs with a duration of several days or even

	weeks but typical maximum walltime for a production HPC cluster is for example[1] just two days.
What are the challenges or issues?	Exceeding the maximum walltime of the queue would cause the job to be stopped by the scheduler. One possible solution is to split this type of jobs in multiple parts and use the long-term support offered by the HEAppE Middleware. In this case, the applications need to employ a checkpointing mechanism and to use it to start the subsequent part of the job, as it will be divided into smaller tasks.
What are the envisioned solutions?	This solution is offered by HEAppE Middleware as an existing functionality that supports the execution of a long-term running jobs. Using a special attribute, the user is able to submit a job with a higher wall-time than natively supported by the selected queue. HEAppE splits this job into a number of smaller ones and sets up a dependency chain between them. The split jobs are using the same cluster system and the same queue. A possible improvement could be implemented at orchestration level such that the split jobs could utilize different systems and queues.

Table 7 LL on the Long-term job support

AUTHORS	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Jan Swiatkowski	**10/09/2021**	**Cron job support**	Some workflows must be launched periodically at specific time. An automated solution would be needed to avoid manual workflow executions.
QUESTIONS			**ANSWERS**
What activities led to this LL?			Some workflows from WP7, (e.g. Weather forecast workflow) must to be launched at specific time, to avoid the need for user intervention and to ensure quickest availability of fresh results.
What are the challenges or issues?			The most challenging part was to store the users´ authentication data and keeping them valid over the lifetime of the job.
What are the envisioned solutions?			The cron jobs info, such as parameters, users´ authentication data, etc. are stored in a database located in a separate service which then launches the executions at the right time. This service is located in LEXIS BackEnd and communicates with the A4C interface. The service is accessible as a part of the LEXIS Platform API.

Table 8 LL on the Cron job support

[1] For details on queues walltime of IT4I clusters, see https://docs.it4i.cz/general/resources-allocation-policy/

2.1.5 Analysis of Security Mechanisms for Securing Federated Infrastructure

Securing federated platform like LEXIS is very challenging. When choosing a solution, a set of requirements must be considered, such as the structure of the platform itself, its components, and their interaction, but also the implementation effort and time constraints. Various security frameworks have been taken into consideration and examined in relation to their characteristics. One example is described in Table 9.

AUTHORS	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	27/04/2020	Analysis of security mechanisms for securing federated infrastructure	Extend analysis of security mechanisms to security framework (such as SPIFFE[2]) to include service mesh technology and not only focus on Identity and Access Management solution.
QUESTIONS			**ANSWERS**
What activities led to this LL?			During M15 F2F meeting in Grenoble, EAB member Dr. Alan Sill from Texas Tech University raised a question about using service mesh for security and, in particular, SPIFFE framework instead of making an analysis of IAM solution and especially Keycloak, which triggered our interest to study the new framework.
What are the challenges or issues?			While the security framework adds more security possibilities (automatically add a communication channel between component), which seems interesting, there are potential challenges through the first supposition. One main challenge for using such security framework will be the interconnection with the LEXIS DDI, LEXIS Orchestrator and HPC centre. Such frameworks usually work fine in virtualized, containerized environments including Cloud, but do not well support existing physical components or software with high operating system constraints such as iRODS. Moreover, evaluating several security frameworks and several security IAM components would take too much time and might delay the LEXIS Project.
What are the envisioned solutions?			Take into consideration the feedback from EAB for next evolution of the LEXIS Platform.

Table 9 LL on the Analysis of security mechanisms for securing federated infrastructure

2.2 ORGANISATION RELATED LESSONS LEARNED

In this section, we present LL on organizational aspects.

2.2.1 GENERAL – federation challenges

In the context of a federated environment, it is important to put in place a solid system in order to avoid, when possible, problems related to infrastructural availabilities and failures. The envisioned solution for this challenge is

[2] SPIFFE: https://spiffe.io/

described in Table 10. The LEXIS Platform encompasses a large number of components (such as DDI, Orchestration and HPC / Cloud) increasing the complexity in respecting and implementing all the phases of software delivery life cycle (SDLC) and especially the one related to environments and testing. This complexity is increased even more with the integration of external components such as EUDAT and iRODS. Table 11 details the challenges of this LL and how we manage them.

AUTHORS	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	27/10/2021	General high-availability and resilience system	For building a highly available and resilient platform, all the requirements must be taken into account since the beginning and the design phase of the project.
QUESTIONS			**ANSWERS**
What activities led to this LL?			Some workflow executions were not completely finished during the Pilot testing due to intermittent problems with token validation, therefore we were unable to gather their results.
What are the challenges or issues?			The main challenges are twofold to build such high-available and resilient platform: • Time constraints, • System heterogeneity between HPC Centres.
What are the envisioned solutions?			Due to potential failure in the High-availability and resilience system with clusters in active/active mode, we choose to configure the HA clusters in active/passive mode. Once all the potential causes of failure will be handled, we will be able to switch back the clusters to the active/active mode. As a side note, an active/passive mode differentiates from active/active in the load balancing feature and failover phase.

Table 10 LL on the General high-availability and resilience system

AUTHORS	DATE	LL TITLE	LL BRIEF DESCRIPTION
Frédéric Donnat	27/10/2021	Development and Production environments	As far as hardware and access to HPC centre resources are concerned, it is not always possible to follow software development best practices and use several environments such as development, staging and production (with or without environment reset).
QUESTIONS			**ANSWERS**
What activities led to this LL?			Prior setting up LEXIS Platform production, we have encountered the issue that some Pilots were using the development environment for developing, testing and executing their workflows and that we did not had enough resources to deploy a new entire LEXIS Production Platform.
What are the challenges or issues?			The challenge is to secure production, testing and development environment without putting the overall project at risk by slowing down the development due to quality or

	security constraints (releasing broken feature in production or data leakage due to an open developer access).
What are the envisioned solutions?	As we could not replicate the entire LEXIS Platform, we decided to follow best practices (development, staging and production environments) for some components such as LEXIS Portal, while cleaning up and monitoring other components such as LEXIS DDI.

Table 11 LL on the Development and Production environments

2.2.2 Lesson Learned from LEXIS Open Call activities

The LEXIS OC activity gives the opportunity to test the LEXIS Platform in real conditions, as a whole, and to collect important feedbacks from the LEXIS OC Applicants. In this section, we directly report LL related to the LEXIS OC process itself and to the Applicants' on-boarding, while LL closely related to the applications implementation and to the Applicants feedbacks are reported in Deliverable D9.12 [6] in a dedicated section.

With the perspective of future exploitation post end-of-project, the adopted solutions are reported in the LL at Table 12 and Table 13, respectively.

AUTHORS	**DATE**	**LL TITLE**	**LL BRIEF DESCRIPTION**
Giacomo Vitali, Marc Derquennes	**10/01/2021**	**New cases applicants on-boarding process**	The LEXIS OC applicants' on-boarding required the definition of procedures for both organizational and technical integration into the LEXIS Platform.
QUESTIONS			**ANSWERS**
What activities led to this LL?			The LEXIS OC selection and first interactions with the applicants made clear the need for concrete on-boarding procedures.
What are the challenges or issues?			Applicants need to be properly guided for an effective implementation of their experiments in the LEXIS Platform and framework, as each of them have different technical requirements, interests and expected results.
What are the envisioned solutions?			The applicants' selection and on-boarding processes are described in detail in Deliverable D9.12 [6]. On the technical level, each applicant is assigned to an internal LEXIS referent named Project Manager, who functions as main contact point. The Project Managers organize dedicated technical TelCos with the assigned applicants, to identify the requirements (like use of specific hardware/software, needed resources, data-related requirements, etc.) and proceed with the experiment implementation. Technical activities included deployment of required code and dependencies on the clusters and interaction with specific LEXIS support. During later stages, they train the applicants on the use of the platform, collecting feedback and deriving LL during the whole process.

Table 12 LL on the Open Call applicants on-boarding process

AUTHORS	DATE	LL TITLE	LL BRIEF DESCRIPTION
Marc Derquennes	01/12/2021	Training for new applicants	The LEXIS OC applicants are very diverse and with various HPC skills. How to help them increase their skills and knowledge, improve their user experience?
QUESTIONS			**ANSWERS**
What activities led to this LL?			When evaluating the projects submitted by applicants, we had the opportunity to assess how prepared the applicants were in terms of familiarity with the HPC ecosystem, and what was potentially needed.
What are the challenges or issues?			The main challenge is in the diversity of experience the applicants have in high-end computing. The LEXIS Project developed a platform and a Portal to interact with end-users, designed to remove most barriers for non-experienced users. Nevertheless, the on-boarding process has to be facilitated at least by a minimal training, and the performance of each application experiment is also dependant on how well mastered are the arts of software development, parallel programming, code optimisation, workflow design and execution, among others.
What are the envisioned solutions?			A training programme has been designed to cover most needs from users, to help them prepare their application experiments, handle and interact the LEXIS Portal, planning their project and finding the users guides, video tutorials, the technical documentation. The training has been designed to be delivered face-to-face, or digitally (adaptation to COVID), live or as on-demand content. The training is mostly structured as following: • Overview and walkthrough, • On-boarding, administrative process, legal framework for users, • Security, • Preparing your code, • Software to be deployed for the App Experiments, • Preparing Datasets, • Preparing workflows, • Outputs: visualisation, data exports, • Using the LEXIS Portal, • Evaluating costs, project planning & running times, • On-line resources, user guides, videos, technical information.

Table 13 LL on the Training for Open Call applicants

3 LEXIS – FINAL PLATFORM ARCHITECTURE

The LEXIS consortium has built a platform which leverages large-scale geographically distributed resources from existing HPC infrastructure, employs Big Data analytics solutions and augments them with Cloud services. The LEXIS Platform relies on a three-layer architecture that has been described in detail in Deliverable D2.3 [3]. The final

architecture, whose detailed description is shown in Figure 1, is the result of the co-design process together with the modifications implemented to face the technical challenges identified in the LL.

Figure 1 Final LEXIS architecture scheme [7]

This picture updates the one in Deliverable D2.4 [4] by some changes that are described in the next Sections. Moreover, every single module is fully detailed in the same deliverable.

The presented final LEXIS architecture demonstrates the achievement of the co-design process. As it turns out from the platform validation, performed also through „real-life" workflow executions, the platform, thanks to its features, meets the requirements of the use cases. Table 14 shows the key technological features which have been

implemented and validated in relation to the specific use-cases which leverage them. The validation is described in detail in Deliverable D2.6 [5].

USE-CASE	HPC	CLOUD	LONG-TERM HEAPPE JOBS	CHECKPOINTING	FAILOVER	BATCH EXECUTION	CRON	CONCURRENT EXECUTION/URGT. COMPUTING	ENCRYPTION	REPLICATION	DDI	WCDA	VISUALISATION	FEDERATION
WP5 Aeronautics	X	X	X	X	X				X	X	X		X	X
WP6 Earthquake & Tsunami	X	X						X		x	X			X
WP7 Weather & Climate	X	X				x	x			X	X	X		X
Generic HPC	X					x	x		x	x	X			X
Generic Cloud		X					x		x	X	X		x	X

Table 14 Key technological features of the LEXIS Platform in relation to specific use-cases
Capital crosses are used to indicated fully tested features, while lowercase is used for implemented but not validated ones for specific cases.

Note that the Generic HPC and Cloud workflow templates have been successfully used by LEXIS OC applicants to execute specific workflows. Moreover, it is important to point out that the CRON and batch execution features have been developed and implemented on a later stage for specific exploitation purposes, and therefore still need final testing via the execution of complete workflows.

In the next sections, we report and describe the modifications made with respect to the architecture's structure provided in Deliverable D2.3 [3] and Deliverable D2.4 [4] for each layer of the LEXIS architecture, as well as the assessment of specific services.

3.1 LEXIS PORTAL

The LEXIS Portal is the main entry point to the LEXIS Platform: it is designed primarily for the users who do not necessarily have deep experience with working with HPC and/or Cloud environments while being provided an easy access to the most advanced capabilities and features available on the field.

Progress on design and implementation of the LEXIS Portal is mostly completed at the time of writing.

- The initial internal release -R1- was delivered in Q4/19 providing basic authentication capabilities, had initial user and organization models, and provided some basic integration with the DDI.
- The second internal release -R2- was delivered in Q2/20, with its major improvement from R1 being the ability to deploy basic workflows via the integration with the Orchestration Service; also, the CYC Accounting and Billing system was integrated.

- The third internal release -R3- was delivered right before the start of the LEXIS OC and it is by far the most complete release of the platform until now.
  The major improvements from the previous release include the implementation of request and approval of HPC resource allocations, the complete integration of authorization across all the services, and the revamping of several parts of the user interface (UI) to get a consistent user experience (UX).

The LEXIS Portal consists of the following components:

- LEXIS Portal FrontEnd: a rich web client which talks to the LEXIS Portal API,
- LEXIS Portal FrontEnd server: a lightweight process which serves the FrontEnd and supports an OpenID-Connect workflow,
- LEXIS API: a service interfaces with the FrontEnd and controls access to the other services within the system, providing the LEXIS API itself,
- LEXIS Approval System interface: a module that interacts with the HPC's HEAppE Approval System for resource request and allocation,
- LEXIS Dataset Management Interface: a module that communicates with the LEXIS DDI to obtain directory listings, file contents etc.,
- LEXIS Workflow Orchestration Interface: a module that interacts with the Alien4Cloud module of the Orchestration Service described below to support workflow deployment and monitoring,
- LEXIS UserOrg Service: a service that stores LEXIS user/organization/project mappings.

All of these components have been developed and integrated, and are considered fully functional. However, it is now during the last R4 cycle where the system is being exposed to real users to test it; hence some edge cases are expected to be discovered and fixed during this phase.

More details on the design of the LEXIS Portal components can be found in Deliverable D8.3 [8].

3.2 LEXIS SERVICES

As described in Deliverable D2.3 [3], the LEXIS Platform offers three key services:

- AAI Service, which relies on Keycloak as well as OAuth2, OpenID-Connect and SAML standard frameworks to guarantee the security of the entire LEXIS Platform,
- DDI Service, which is based on iRODS and EUDAT-B2SAFE for collecting, managing, storing, retrieving, and providing data,
- Orchestration Service, which relies on Ystia Orchestrator (YORC) to schedule the execution of tasks on the compute nodes throughout the application life cycle.

The sections below update the previous deliverables to the final status of these three services, and provide their technological assessment as well.

3.2.1 AAI Service

No architectural changes were made in LEXIS AAI since the design phase. The architecture diagram has been described first in Deliverable D4.1 [9] and then in the final version in Deliverable D4.5 [10]. Thanks to the fact that high-availability, resiliency and security have been taken into account since the beginning of the design of this component, aiming at providing the best efficiency, the architecture is flexible enough to allow for configuration changes.

Since the beginning we planned to integrate an active/active cluster mode of each single sub component (database, cache, IAM and reverse-proxy) of the LEXIS AAI based on the cross-datacentre replication mode of Keycloak. We have been able to handle the synchronization issues (due to VPN resiliency, network instability or even component configuration), by switching to an active/passive mode by reconfiguring the reverse proxy (based on HAProxy) and updating the load balancing mode.

As of now, we kept the database as an active/active cluster, the caches are synchronized between datacentres and both IAM systems are then synchronized, but only one IAM system is handling the load. As a side note, the architecture also encompasses the ability to independently grow the cluster in either datacentre.

In order to fully assess the architecture of the LEXIS AAI service from the co-design perspective, we have not only validated the deployment, but also validated the integration with all other LEXIS components. As a matter of fact, we needed to make some improvements on the LEXIS AAI configuration and add the ability to exchange tokens in order to be able to use offline token of type refresh token for long running tasks. During the co-design phase, we were not fully aware of all underlying components that would eventually be used in other services such as DDI and then we opted for an IAM solution with a very active community and industry support. This choice allowed us to enable some features (such as exchanging tokens) later on and review the configuration to better adapt the LEXIS AAI service. Last, but not least, we conducted some security auditing on LEXIS AAI deployment and configuration to ensure the proper implementation of the zero-trust architecture concept which are described in more details in Deliverable D2.6 [5].

3.2.2 DDI Service

The DDI core architecture has remained unchanged since 2019, when it was decided to be built upon iRODS and EUDAT B2SAFE (cf. Deliverable D2.2 [2]). On the API side, the staging API has undergone a major re-design after its initial deployment in 2020. This makes the staging API more flexible when additional datacentres join the LEXIS Platform. Also, APIs for encryption and compression were added to accommodate the needs of the use-cases, accelerating and securing data transfers. In addition, a synchronisation mechanism was implemented (DDI-AAI sync worker) and deployed at each centre to have the DDI structure and user/project data base synchronised to the LEXIS AAI. The architecture as of 2021, together with the requirement analysis it is based upon, is discussed in depth in a book chapter currently in print [11], and in Deliverables D3.5 [12] and D3.6 [13].

A major addition to the DDI service, challenging the architecture, was the deployment of iRODS and EUDAT components at ICHEC in 2021. The ICHEC iRODS zone was then successfully federated with the existing LRZ and IT4I zone.

Smaller works on the DDI since M16 have focused on upgrading the modules such as iRODS and redeploying EUDAT B2SAFE and B2STAGE. Furthermore, the LEXIS monitoring system was established and gradually enhanced with further metrics dashboards and alerts with automatic notifications, once LEXIS components are down. Periodic DDI tests were deployed on all the three centres. The tests ensure that all the iRODS zones and its APIs are up and functional.

The successful DDI deployment on the federated HPC centres, and the ability through the DDI API to stage a dataset to make it accessible to any compute resource in the LEXIS HPC/Cloud federated infrastructure, allows the Orchestrator Service to dynamically run workflows on any available location. In fact, the creation of a specific YSTIA DDI plugin, described in detail in Deliverable D4.6 [14] (Section 3.6), extends the orchestrator functionalities in order to proficiently use the DDI by implementing specific TOSCA components. This, in conjunction with the creation of the DAM module, resulted in the possibility to run jobs with YSTIA in any location regardless of actual data locality, which was not possible before the LEXIS project as workflows were statically instantiated. Therefore, the DDI has been successfully used and assessed as an integral part of the workflows' execution, as shown in Table 14. With all adaptations through the project time as mentioned, but with its basic architecture remaining the same, the DDI has proven successful in supporting the LEXIS Pilots as well as the first LEXIS OC use cases. Problems have been limited to implementation bugs which have been fixed, but have not affected the architecture. Clearly, this does not fully exclude a future revision of parts of the DDI in a future LEXIS extension, e.g. for easier portability where staging areas are used to buffer data (staging API, SSHFS API, etc. - cf. Deliverable D3.6 [13]).

3.2.3 Orchestration Service

No major change has been made to the Orchestration Service, the only one being the AAI connector sub-module which has been removed, as this service just relies on the APIs exposed by the AAI. In fact, all of the technical challenges encountered during the implementation phase were successfully met by implementing specific features to already existing sub-modules, as described in the LL at Section 2.1.4, so no architectural modification has been required.

The Orchestration Service has been thoroughly tested and validated during the validation by pilots and, at the time of writing, it is being further assessed by the LEXIS OC applicants. The implemented features, together with the interactions with the other LEXIS modules like DDI and the LEXIS BackEnd Services allow to meet all the technological requirements identified during the co-design phase, as described in the previous section and shown in Table 14.

3.3 INFRASTRUCTURE LAYER AND NETWORK

During 2021, no relevant modifications have been made to the LEXIS HPC and Cloud architectural infrastructure.

After M16, a new storage system was deployed at LRZ. The Hardware systems utilised are described in Deliverable D3.6 [13], Section 2.1.

The Approval Service architecture has been reworked. The HPC and Cloud modules have been removed, as their features are now implemented by the conjunction of the “Approval” and “HEAppE’s Manage and Status Report” modules, and, as envisaged, the security at the HPC level is enforced by the HPC-specific AAI instead of the LEXIS one.

In the following section we provide major updates to the FPGA accelerator usage in the context of pilot workflows.

3.3.1 FPGA Accelerator

The LEXIS infrastructure layer aims at integrating high-performance computing and storage resources to support the execution of complex application workflows (as represented by Pilots workflows) in the best way possible. To this end, innovative computing and storage systems have been acquired and made available to the LEXIS Platform. One of such advanced systems is represented by FPGAs (Field Programmable Gate Arrays). LEXIS consortium recognized that FPGA accelerators could provide speed up over industrial and scientific application use cases at a low power consumption, thanks to their flexibility and the availability of high-level synthesis tools that ease the process of writing and porting code to these systems. The LEXIS consortium agreed to investigate on the acceleration of a specific kernel in the context of the Aeronautics use case (WP5).

The acquired board belongs to the Bittware Nallatech 520 series (specifically, LEXIS opted for the 520N model which includes the QSFP connector cages and the hardened network IP), which is equipped with the Intel Stratix10-GX FPGA fabric. The Intel Stratix10 2800-GX offers excellent features to both design custom accelerators (typically programming the device through a VHDL/Verilog environments) and to accelerate (legacy) codes by developing acceleration kernels. In this latter case, the FPGA configuration and the HDL synthesis are left to the High-Level Synthesis compiler which derives the programming bitstream from the kernel description done in a high-level programming language (C/C++). In this context, Intel platforms offer the support of the OpenCL 1.X standard (which actually has been designed to be agnostic with respect to the hardware platform), providing a programming model closer to that typical of the GPUs. Usually, the main application code embeds the necessary functions that allow to interact with the board through the PCIe link (the board support package —BSP— reserves a portion of the programmable resources to implement the PCIe IP block), in order to move data in and out the main memory of the board, to initialize the computing kernels and to start them.

As a matter of showing the integration of the FPGA board with the LEXIS Platform, we chose to integrate the card into the burst buffer node available at IT4I infrastructure. Thus, the FPGA card could be explored for accelerating different types of operations, ranging from service level functions related to the data management (e.g., data compression/decompression, data encryption, data format conversion) to the actual acceleration of use cases. Here we report the status of implementation of those activities. Concerning the implementation of data management service functions, on-the-fly data encryption functionality has been correctly synthesized on the Nallatech 520N FPGA board. As an example, Table 15 summarizes the outcomes of the synthesis process (using Intel Quartus compiler toolbox) done, where ALMs are the Adaptive Logic Modules, FFs are registers, DSPs are Adders/multipliers, RAMs are blocks of embedded RAM used to map code arrays and MLABs are memory logic array blocks. In this case, we created a set of dedicated kernels, one for each type of encryption algorithm (AES128, AES256, DES, Camellia), and one for its decryption counterpart, by modifying an existing GPU library3.

On the side of simulation codes acceleration, two specific cases have been considered: one related to the Aeronautics Pilot - within the TRAF simulation code, a routine with any dependency on external library and performing SP floating point operations (summation, multiplication and division) was selected; the other one related to the Tsunami and Earthquake Pilot - acceleration of the rasterization of the mesh used in the context of WP6 workflow simulations. It is worth noting that the process of creating and optimizing FPGA kernels is iterative, i.e., it is required to synthesize and test new version of the circuits implementing the kernel multiple times. During this iterative process, large amount of space on the disk and main memory is consumed (e.g. 8 GB of storage and order of tens GB of RAM per single synthesis run for the TRAF case), thus requiring a node generally equipped with large memory and storage pools (where the FPGA is integrated) like the Burst Buffer used on the LEXIS project. Once the design is optimized, it can be moved to the Stratix10-based device for further few optimisation iterations aimed at exploiting the bigger amount of resources and the capability of synthesizing a faster circuit (i.e., clocked at a higher frequency). Results for the synthesis of the TRAF code kernel, together with the details for this case, are reported in Deliverable D5.4 [15] (Section 2.2.2).

	ALMs	FFs	RAMs	DSPs	MLABs
Full design (all kernels)	249k	337k	1.7k	3	2.1k
Total Resources	933k	3.7M	11.7k	5.76k	~23.5k

Table 15 Summary of the FPGA fabric resource usage after the synthesis process of encryption kernels completed

The last aspect that was investigated relates to the way the FPGA board can be integrated within the orchestration service. To this end, the main idea was to make the board to be transparently served through the Cloud as part of the virtual resources; i.e., allowing a virtual machine (VM) or a Docker container to get access to the board runtime module. As such two distinct paths have been tested: i) enabling PCI-passthrough and using a VM; ii) creating a Docker image configured to directly access the FPGA runtime on the burst buffer node. While the first solution provided seamless integration with the OpenStack environment, enabling PCI-passthrough in the card driver was not possible due to lack of documentation from the board supplier. To overcome this issue, we tested the second configuration, where a Docker image was set to access the FPGA runtime. One important remark from this investigation is the limitation imposed by the FPGA runtime and the driver; i.e., unlike recent GPU drivers/runtimes that allow multiple concurrent accesses to the accelerator, the Intel FPGA environment does not support such a feature. This should be translated into correct configuration of the OpenStack environment to avoid running multiple Docker containers that try to get access concurrently to the FPGA board. Such limitations should be (partially) overcome with the Intel OPAE driver architecture [16].

[3] https://github.com/heipei/engine-cuda

4 SUMMARY

The LEXIS Platform deployment phase has been concluded and all of the technological components have been put in place and properly tested by pilots while, at the time of writing, LEXIS OC applicants have started using the whole platform. The LL produced by this second part of the project have been reported in 2.1. By summarizing them, we can affirm that:

- The complexity of the LEXIS Platform, composed of a large number of modules and components, determined several unpredictable technical issues especially with regards to the maintenance and support phase.
- The federation of HPC centres, as well as the implementation of a high-availability and resilience system, requires the synchronization of HA clusters and databases, which took a lot of effort.
- The LEXIS OC activity gives the opportunity to test the LEXIS Platform as a whole through the implementation of external applications. For future exploitation, a proper on-boarding process need to be defined, as well as the relative documentation and guides.

At M36, the LEXIS Platform has been deployed, assessed and validated by its pilots. Its architecture, described in detail at Section 3, has been finalized after some modifications made to meet the challenges encountered during the whole project lifetime. The following achievements have been made:

- The R4 release of LEXIS Portal, as detailed in Deliverable D8.3 [8], supporting deployment of pilots and workflow templates, has been released.
- The AAI has been finalized and deployed, allowing uniform Role Based Access Management.
- The DDI has been deployed in both centres and now it allows full data management functionality.
- The Orchestration system has been implemented in order to meet all the requirements defined by pilots and is now under further validation by the LEXIS OC applicants.
- All the hardware systems have been installed and exposed as described in Section 3.3.

Finally, after the validation process described in Deliverable D2.6 [5], the assessment of the final LEXIS architecture confirms that all the technical requirements defined by the technical work packages and the pilots were successfully met, while ongoing LEXIS OC activity is further demonstrating the platform's capabilities.

REFERENCES

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

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

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

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

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

[6] LEXIS Deliverable, *D9.12 Open Call Framework and Stakholders Engagement on Targeted Large-Scale Pilots - final.*

[7] J. Křenek, J. Martinovič, F. Donnat, M. Golasowski, J. Munke, S. Hachinger, K. Slaninová, M. Levrier and P. Harsh, "LEXIS Platform Architecture Scheme," 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5810806.

[8] LEXIS Deliverable, *D8.3 Final Release of LEXIS Portal.*

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

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

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

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

[13] LEXIS Deliverable, *D3.6 Data Flow Optimisation and Data System Core.*

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

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

[16] "OPAE Intel FPGA Linux Device Driver Architecture," [Online]. Available: https://opae.github.io/1.0.0/docs/drv_arch/drv_arch.html.

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

A OPEN-SOURCE SOFTWARE STATUS

As reported in Deliverable D4.6 [14], in this Appendix, it is provided an update of the Open Source status of two specific software, namely the Dynamic Allocator Module (DAM) and the Orchestration Service API, which were under internal review and publication process during the submission of the dedicated Deliverable D4.6 [14].

The DAM software has completed the aforementioned procedure and is now Open Source. The Repository, documentations and other details are reported in Table 16 below.

The Orchestration Service API, instead, is under final quality and security check at the time of writing and will be Open Sourced as soon as they are completed.

The full list of the LEXIS Open Source results (relating to the LEXIS Platform, not including results from LEXIS Pilots) is a part of the Deliverable D9.10 [17].

SOURCE CODE	https://github.com/lexis-project/dynamic-allocation-module
LICENSE	MIT
DOCUMENTATION	https://github.com/lexis-project/dynamic-allocation-module/blob/main/README.md
RELEASE	version 1.0.0 https://github.com/lexis-project/dynamic-allocation-module/releases

Table 16 Dynamic Allocation Module repository