The LEXIS Platform

Easy access to heterogenous computational workflows execution

Mohamad Hayek (LRZ)
Martin Golasowski (IT4I)

iRODS User Group Meeting 2022
• The LEXIS platform
• LEXIS Distributed Data Infrastructure
• Authentication with OpenID
• iRODS-Keycloak fine grained access control
• Staging API
• Automated Robot tests
• Bandwidth measurements
THE LEXIS PLATFORM: Timeline

2018 – LEXIS Platform concept
• Technical Concept
• Creation of a Consortium – 17 members, from Germany, Czech Republic, France, Ireland, UK, Italy, Switzerland
• Submission of the LEXIS Project to the EU Commission – Programme H2020
• Approval by the EU Commission - total budget circa 14 millions € - Grant agreement 825532
• Coordinator: IT4Innovations – National Supercomputing Centre of Czech Republic (Dr. Jan Martinovic)

2019 – Start of the project – 3 years timeline
• Project ending December 2021
• Completion validated by the EU Commission June 2022
• Exploitation post-end of project: starting now

2022 – Onward
• Technical tuning & development
• Enrolling new partners (Infrastructures, Data, Services)
• Structuration of legal entity
• Funding
From inception LEXIS has been built by an ecosystem made of representants of:

- Industries, (4)
- Research Organisations (6)
- Super computing centres (4)
- Service companies (4)
- SMEs and Start-ups (5)

The LEXIS Platform will further develop by capitalising on:

- Existing partners,
- Developing the number of infrastructures as members of the federation,
- Strongly reinforcing services by welcoming new Service partners,
- Increase technical ways for integration, interconnections and cooperation with the addition of new components to the federation via development of APIs,
- Digital Sovereignty European framework (GAIA-X or else).
Key Points of LEXIS Platform

• Dynamic, complex Cloud- & High-Performance-Computing / Big Data workflows
  • Orchestration in geographical federation with YORC, HEAppE
  • Real-time deadline-aware workflows, etc.

• Cross-site (meta-)data federation
  • Distributed data management and data discovery with EUDAT/iRODS
  • Data transfers accelerated by Burst Buffer nodes; FPGAs/GPUs for on-line processing

• Web portal and interfaces for workflow set-up / execution
  • Unified access to all services via Keycloak-based LEXIS AAI

• Easy HPC/Cloud access for SMEs/Industry – Big Data for everyone
  • HPC-as-a-Service approach
  • Control over resource usage
  • Fine-grained accounting and billing for multiple HPC centres with CYCLOPS
Platform architecture overview

- Federation of European computing centres
- HPC & Cloud service providers, Data providers
- Unified & distributed data management
- Orchestration
- Federated Authentication & Authorization Infrastructure (AAI)
- Masking of technical and operational differences across organizations
Orchestration service & workflow management

- Execution on geographically distributed HPC and Cloud resources
  - **Cloud**: via OpenStack built-in interface
  - **HPC**: job execution is mediated by HEAppE middleware
- Data management and orchestration policies
  - Leverage the LEXIS DDI service for an effective **data transfer** between systems
  - Placement of **workflow tasks** on the most suitable resource

https://github.com/alien4cloud/alien4cloud
https://github.com/ystia
http://heappe.eu
LEXIS Security requirements

- Custom AAI solution with trusted access to HPC with PI approval
- Security-by-design
  - Zero trust, minimal attack surface, separation of concerns
- Modern frameworks
- HPC infrastructures are protected
  - Isolated by the HEAppE middleware (developed in IT4I)
  - Deployed in both IT4I and LRZ
- Flexible
  - Blurs differences between HPC centres
  - Provides SSO across the LEXIS federation
Distributed data infrastructure

**iRODS Servers LRZ**
- iRODS
- LRZ: „DSS“
- IBM Spectrum Scale/GPFS

**iRODS Servers IT4I**
- iRODS
- IT4I: CEPH Storage

**STAGING – REPLICATION – MONITORING**
- LRZLexisZone
- IT4ILexisZone

**Cloud**
- Burst buffer orchestrated via staging REST API

**HPC**
- Burst buffer orchestrated via staging REST API

**EUDAT/B2SAFE**
Zone structure organization

One zone per federated center

Hashed project name

The dataset

$ ils /IT4ILexisZone
/IT4ILexisZone:
  C- /IT4ILexisZone/home
  C- /IT4ILexisZone/project
  C- /IT4ILexisZone/public
  C- /IT4ILexisZone/trash
  C- /IT4ILexisZone/user

iRODS user home
Project level datasets
Public datasets
The trash
User datasets per project
• Current state
  • auth_microservice broker (https://github.com/lexis-project/auth_microservice)
  • irods_openid_plugin (https://github.com/lexis-project/irods_auth_plugin_openid)
  • patched Python irodsclient (https://github.com/lexis-project/python-irodsclient)

• Obsolete?
• How about iRODS 4.3.0?
Users belong to groups according to RBAC matrix
- **DAT_READ / DAT_WRITE** – R/W access to datasets
- **DAT_LIST** – Listing datasets incl. metadata
- **DAT_PUBLISH** - Access to B2SHARE, write permissions to /public collection

Mapping to iRODS groups
- Two groups with R/W access and _mgr with publish
- User sync script runs every 3 minutes and performs a sync
• Django based RESTful API
• Scope: LEXIS orchestrator can move data by simple HTTP request
  • between iRODS,
  • Cloud, and
  • HPC resources at all LEXIS centers.
• Uses LEXIS AAI and the HEAppE middleware to authenticate the requests and the access to the resources
• Deploys a queuing system using Celery and RabbitMQ to allow asynchronous requests.
LEXIS Staging API - Architecture

Staging API server

- Apache HTTP Server
- PostgreSQL Server
- Django Staging API REST Framework
- RabbitMQ Server

Staging API client

- Celery worker
- Task queue
- RabbitMQ

API call flow:
- API call from the client
- Request to Apache HTTP Server
- WSGI to Django Staging API REST Framework
- Task processing via Celery worker
- Response back to client
The source/target combinations are covered by 10 staging classes and served by 7 celery tasks.

Each center has their separate deployment of the celery worker.

Based on the source/target combination, the task is pushed to a specific queue.

<table>
<thead>
<tr>
<th>Class</th>
<th>Source</th>
<th>Target</th>
<th>Celery task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>iRODS</td>
<td>Local cloud/staging area</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Local cloud/staging area</td>
<td>iRODS</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Local cloud/staging area</td>
<td>Local cloud/staging area</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>iRODS</td>
<td>local HPC</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>local HPC</td>
<td>iRODS</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Local cloud/staging area</td>
<td>local HPC</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>local HPC</td>
<td>Local cloud/staging area</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Local cloud/staging area</td>
<td>remote HPC</td>
<td>2 &amp; 4</td>
</tr>
<tr>
<td>9</td>
<td>local HPC</td>
<td>Remote cloud/staging area</td>
<td>5 &amp; 1</td>
</tr>
<tr>
<td>10</td>
<td>Local cloud/staging area</td>
<td>Remote cloud/staging area</td>
<td>2 &amp; 1</td>
</tr>
</tbody>
</table>
Automated Robot tests

- Uses Python and the Robot framework to run basic checks
- The suite is executed every 10 or 20 minutes
- Tests include:
  - Local iCAT test: obtains session with password and OpenID token and performs a file transfer
  - Federation test: tests file transfer between federated zones with OpenID token
  - Authentication test: obtains and validates OpenID token from LEXIS AAI
  - APIs test: verifies function of DDI APIs
- Generates an HTML report with the results
- Sends alert via Slack supported API if a test fails
Automated Robot tests

HTML Reports

Mattermost chatbot for alerting

Irzbot | 5:20 PM
Tests failed
irods_local robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_irods_local_20220602172007.html

Irzbot | 5:20 PM
Tests failed
irods_federation.robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_irods_federation_20220602172007.html

Irzbot | 5:23 PM
Tests failed
auth.robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_auth_20220602173326.html

Irzbot | 5:23 PM
Tests failed
irods_local robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_irods_local_20220602173326.html

Irzbot | 5:23 PM
Tests failed
irods_federation.robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_irods_federation_20220602173326.html

Irzbot | 5:25 PM
Tests failed
irods_local robot Log: https://skpilrz-lexis-elasticsearch.srv.mwn.de/log_irods_local_20220602173524.html
A series of bandwidth tests were executed between LRZ and IT4I to help estimate the data transfer time between the 2 centers.

The tests were executed 20 times in each direction and with different data sizes.

The tests included:

- iRODS to iRODS via icp
- iRODS to iRODS via EUDAT B2SAFE
- iget from local zone via python client
- iput to local zone via python client
- iget from remote zone via python client
- iput to remote zone via python client
• The tests show increased performance with larger files.
• Datasets with a big number of files suffer from low transfer rate

**SOLUTION:**
• Compress the data before moving it to iRODS
• Uses a dedicated machine with 64 VCPUs and NVME disk to perform the compression
• Sustain the platform for commercial use
• Extend Staging APIs for more use-cases (object storage)
• Upgrade to iRODS 4.3.0 and rework OpenID?
• Prepare a DDI node deployment Docker image
  • iRODS iCAT
  • Staging worker
  • HEAppE
  • User sync

Workflow in the LEXIS portal
THANK YOU!

CONTACTS

Mohamad Hayek (LRZ)
hayek@lrz.de

Martin Golasowski (IT4I)
martin.golasowski@vsb.cz

Our GitHub organization
https://github.com/lexis-project