



Martin Golasowski & Marek Nieslanik

IT4Innovations, Czechia

Efficient data staging with iRODS HTTP API
in the LEXIS Platform 2

Easy and secure access to supercomputing



- Modern web interface
- Allocations at one place
- Multiple clusters in one workflow
- Launching applications with a single button
- Easy access to job logs
- iRODS data management
- Common web login
- ... and many more

Visit for more:

<https://lexis.tech>

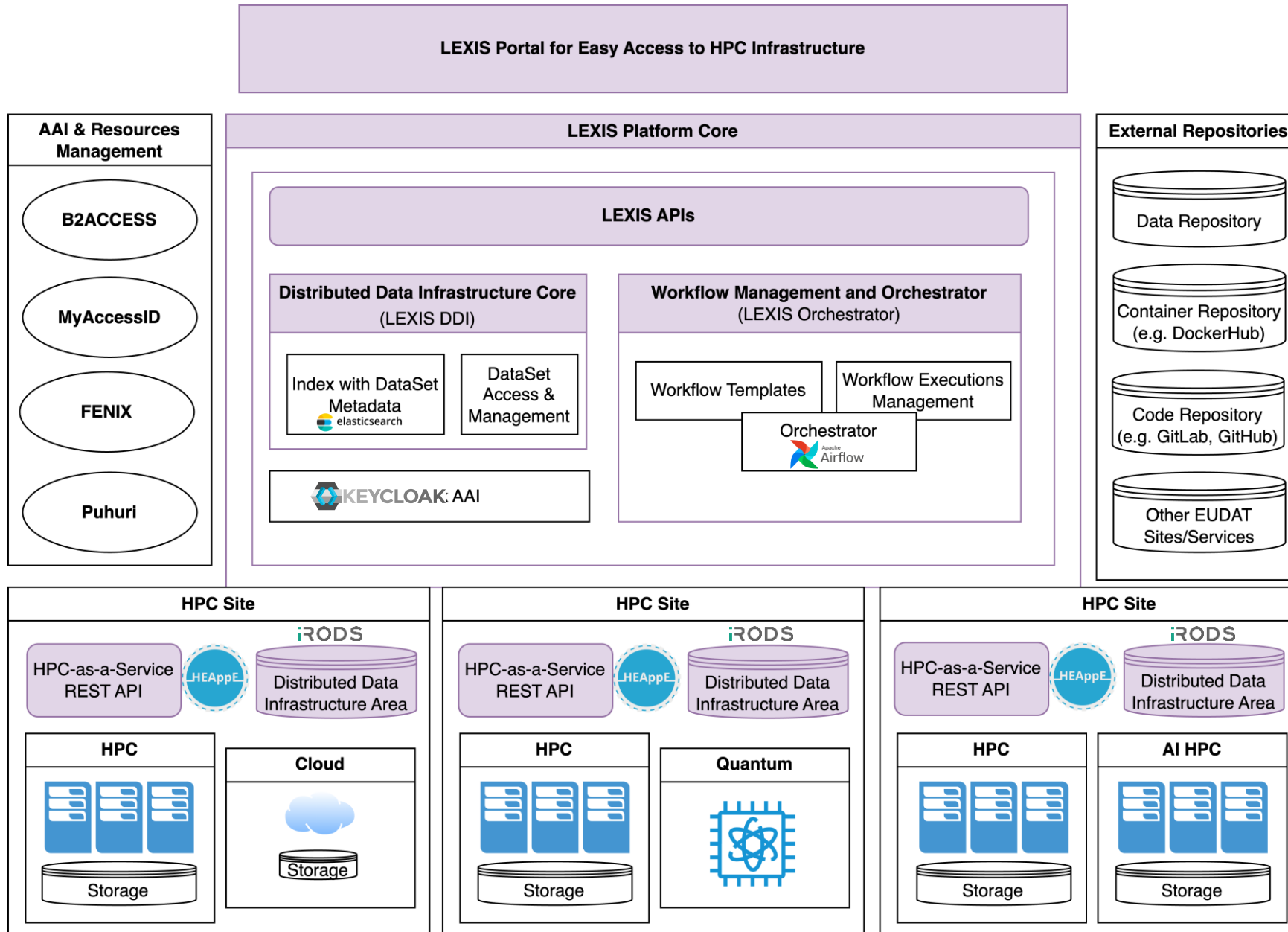
The image displays three overlapping screenshots of the LEXIS2 Platform. The largest screenshot in the background shows the 'Containers' page with a search bar, a 'Refresh' button, and a 'Create Container' button. Below these are three container cards: 'Test Container' (7th Of Mar 2024), 'Quantum AI Test Container' (21st Of Mar 2024), and 'Python Animals AI Example' (12th Of Mar 2024). A smaller screenshot in the middle shows the 'Locations' page with a table of computing resources. The third screenshot, in the foreground bottom left, shows the 'py4lexis' package documentation, including a 'README.md' file and a description of the package's purpose.

NAME	DESCRIPTION
iRODS LRZ	iRODS at LRZ
Barbora	HPC Cluster Barbora
iRODS LRZ OWSeu	iRODS Zone OWSLRZZ
Cirrus	HPC Cluster Cirrus
Leonardo	HPC Cluster Leonardo
LocalCluster	Tartu testing instance
ZYC-2	HPC Cluster Alveo
LRZ Linux cluster 2	HPC Linux Cluster CoolMUC-2 lxlogin4.lrz.de
LUMI	HPC Cluster LUMI lumi.csc.fi
IT4i Staging Area	IT4i Staging Area staging

py4lexis

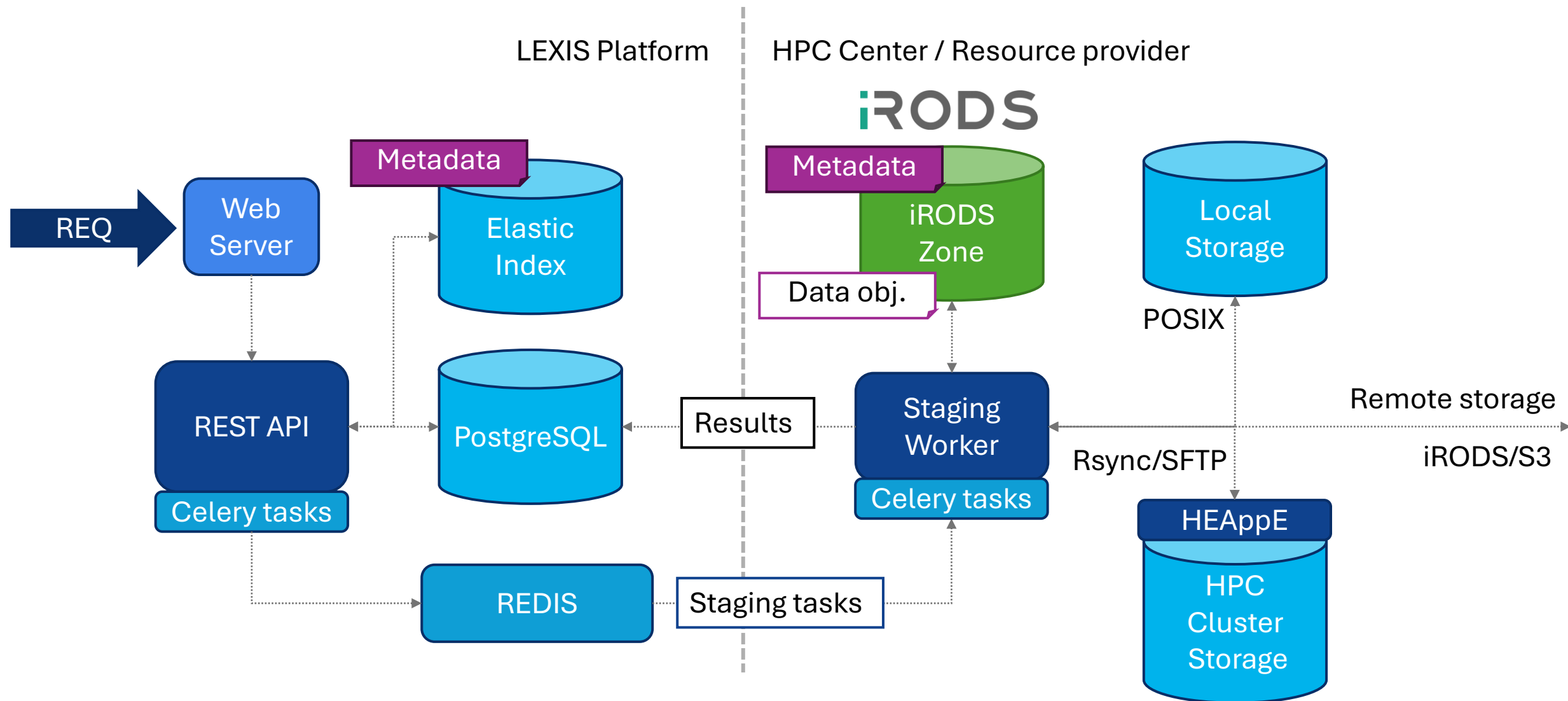
Package py4lexis provides functions to manage Python package, i.e. by TUS Client.

Platform Architecture

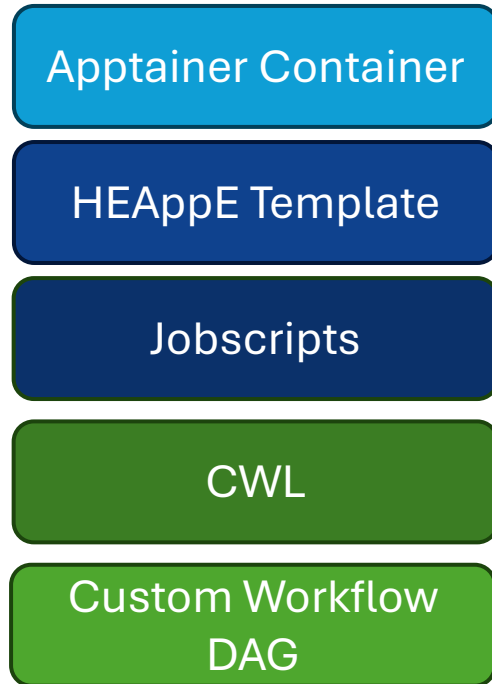


- Part of the **EuroHPC JU Federation platform**
- **Hiding** of technical and **operational differences** across organizations
- **HPC & Cloud** computing providers
- Unified & distributed **data management based on iRODS**
- Workflow orchestration
- **Federated Authentication & Authorization Infrastructure (AAI)**

LEXIS Distributed Data Interface - DDI

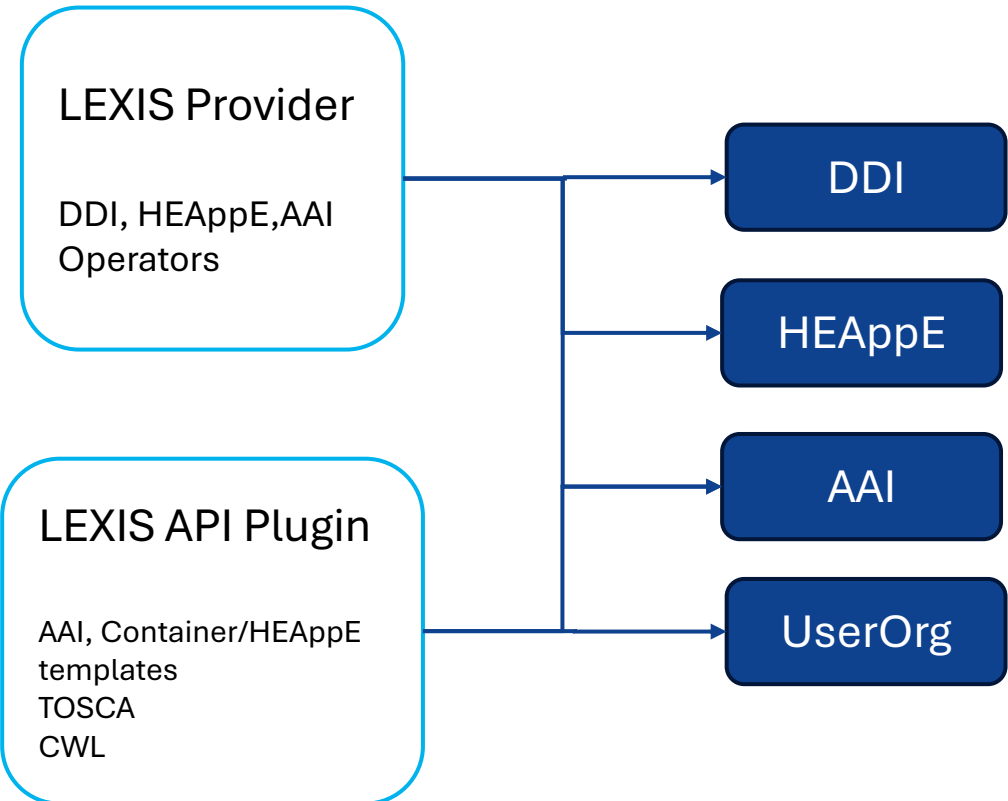


Apache Airflow integration in LEXIS



Workflow specification

Integration with LEXIS APIs



Selected LEXIS use-cases



- LIGATE Project

- Application for molecular docking simulation – private IP by DOMPÉ
- LEXIS provides access to workflows with this application running on HPC
- *Without direct access* to the binary or source code



- OpenWebSearch.eu

- European open web index processed through LEXIS on several HPC locations (LRZ, IT4I, CSC, DLR)
- Public indices made available through the LEXIS Portal

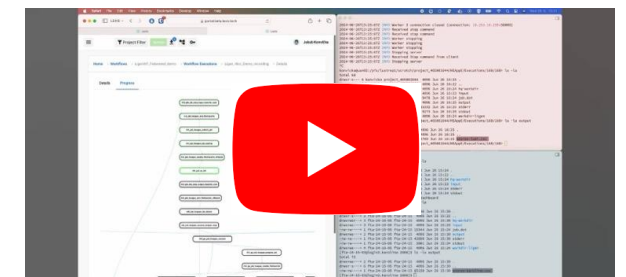


Federated execution on 4 HPC clusters at once

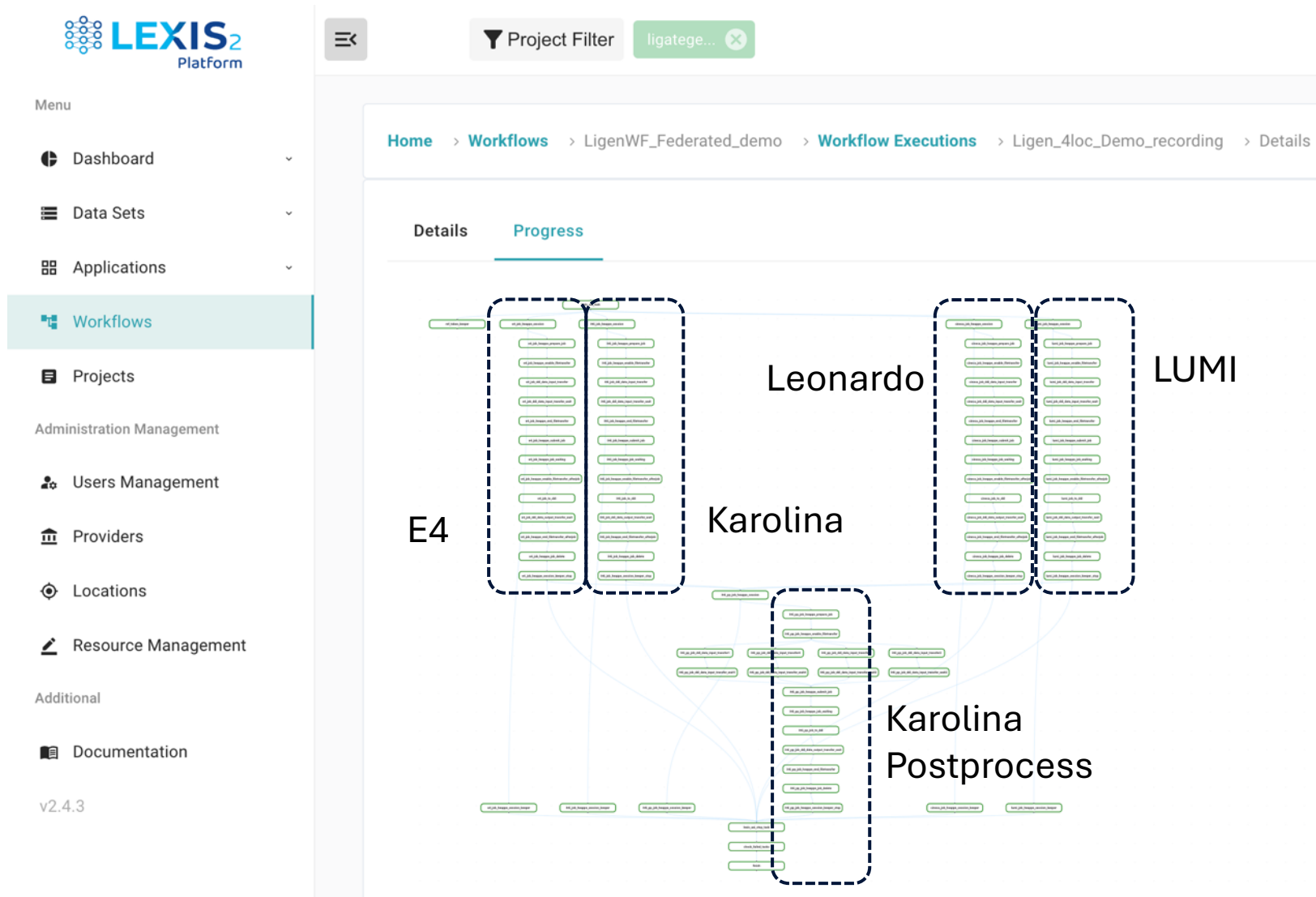


Executed on 4 locations

- LUMI
- Leonardo (CINECA)
- Karolina (IT4I)
- E4 Private Cluster



[See it in action](#)



OpenWebSearch.eu - Public web indexes



The screenshot displays the LEXIS2 Platform interface. The left sidebar contains a menu with options: Dashboard, Data Sets, Uploads, Workflows, Public (selected), Applications, Workflows, Projects, Administration Management, Users Management, Providers, Locations, Resource Management, Additional, and Documentation. The main content area shows a breadcrumb trail: Home > Public > OWI-Open Web Index-main.owi@lrz-2023-12-07 > Details > File List. Below this, there's a folder structure: OWI-Open Web Index-Main.Owi@Lrz-2023-12-07 > 2023 > 12 > 07. A 'Refresh' button is visible. The interface also shows a list of folders with language codes: language=aar, language=abc, language=abk, language=afr, language=amh, language=arc, language=ase, and language=ave. An inset diagram illustrates the system architecture, showing layers from Crawlers at the top to OWS Data Distribution Layer at the bottom. The OWS Data Distribution Layer includes iRODS API, iRODS Zone, and Object Store as a cache (S3). Another inset shows the details of a specific index: OWI-OPEN WEB INDEX-MAIN.OWI@IT4I-2025-06-16:2025-06-16. This details view includes a metadata table with columns for ID, Path, File List, Update Metadata, Delete, and Download. The metadata table lists the index ID, its path, and its public status.

portal.beta.lexis.tech/publicDataSets/e2f40f27-9e47-4464-a7a8-447eb3d7d388/details/fileList

Project Filter EverestG...

Home > Public > OWI-Open Web Index-main.owi@lrz-2023-12-07 > Details > File List

OWI-Open Web Index-Main.Owi@Lrz-2023-12-07 > 2023 > 12 > 07

Refresh

language=aar language=abc language=abk language=afr language=amh language=arc language=ase language=ave

Crawlers

Common Component: Logging and Monitoring

Crawling queue::Frontier Apps

Copy-from-Warc Sitemap-Crawl Exploratory Crawl

Apache Storm Cluster Cluster Manager

Cloud Infrastructure

OWS Data Distribution Layer - Data Center specific bucket

iRODS API iRODS Zone Object Store as a cache (S3)

Home > Public > OWI-Open Web Index-main.owi@it4i-2025-06-16:2025-06-16 > Details

Copy Id Copy Path File List Update Metadata Delete Download

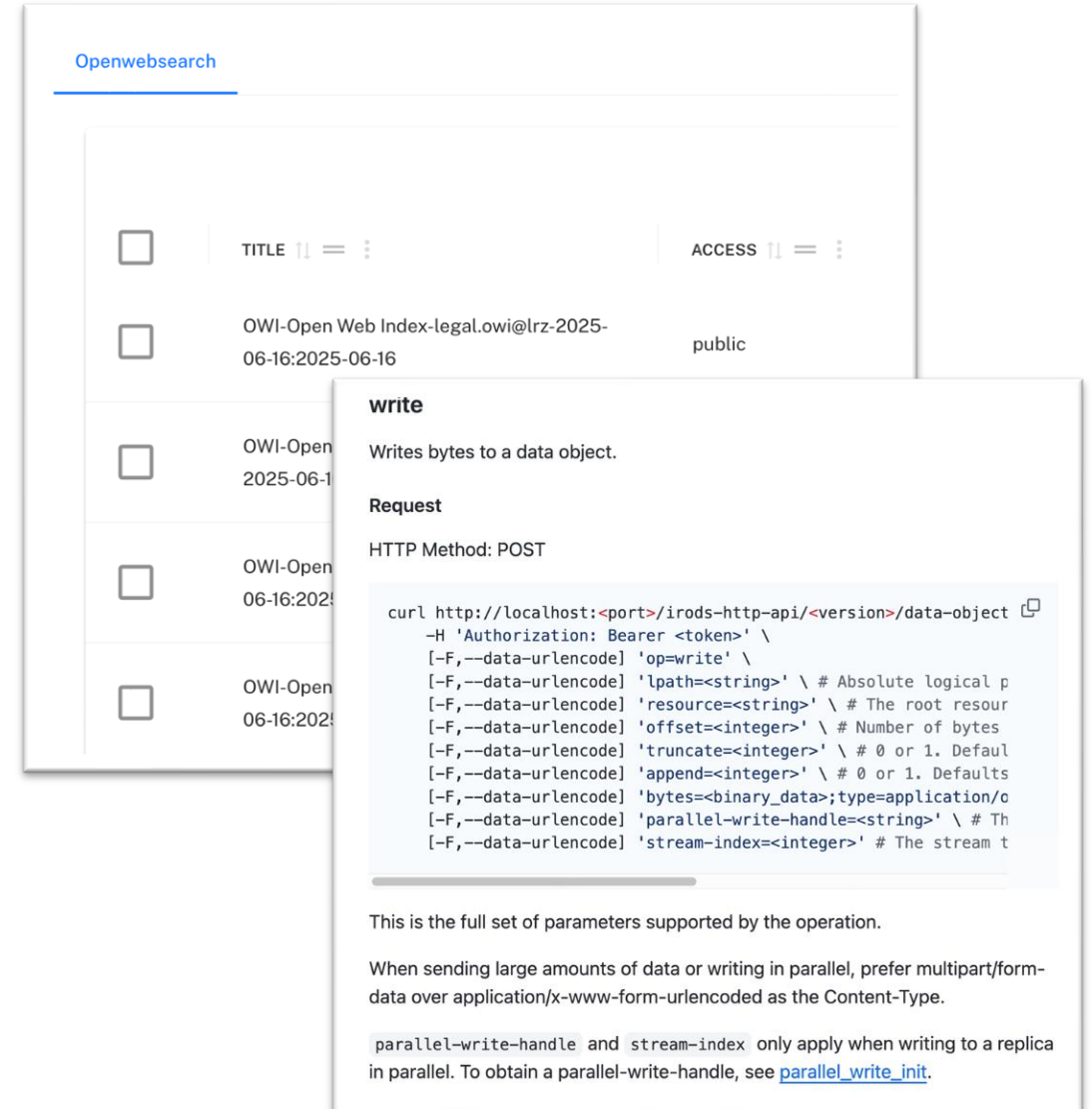
OWI-OPEN WEB INDEX-MAIN.OWI@IT4I-2025-06-16:2025-06-16

Metadata

/IT4ILexisV2/public/proj862c5962623246664c1fda27b7afb108/c9f53082-4b46-11f0-91af-528c047b29ff	
public	
OWI-Open Web Index-main.owi@it4i-2025-06-16:2025-06-16	

Improvements in the LEXIS DDI

- Upgrade to iRODS 4.3
 - Direct integration with OpenID
 - iRODS HTTP API improves usability significantly
- Chunked data transfers
 - RAM buffers improve scalability of the remote staging workers
- iRODS zone restructuring
- Data access analytics
- Performance & scalability tests

The image shows a screenshot of the LEXIS DDI Openwebsearch interface. The interface has a header "Openwebsearch" and a table with columns for checkboxes, "TITLE", and "ACCESS". The table contains several rows of data, including "OWI-Open Web Index-legal.owi@lrz-2025-06-16:2025-06-16" and "OWI-Open 2025-06-16:2025-06-16". Overlaid on the right side of the screenshot is a detailed documentation box for the "write" operation. This box includes a description "Writes bytes to a data object.", a "Request" section stating "HTTP Method: POST", and a curl command:

```
curl http://localhost:<port>/irods-http-api/<version>/data-object -H 'Authorization: Bearer <token>' \ [-F,--data-urlencode] 'op=write' \ [-F,--data-urlencode] 'lpath=<string>' \ # Absolute logical p [-F,--data-urlencode] 'resource=<string>' \ # The root resour [-F,--data-urlencode] 'offset=<integer>' \ # Number of bytes [-F,--data-urlencode] 'truncate=<integer>' \ # 0 or 1. Default [-F,--data-urlencode] 'append=<integer>' \ # 0 or 1. Defaults [-F,--data-urlencode] 'bytes=<binary_data>;type=application/o [-F,--data-urlencode] 'parallel-write-handle=<string>' \ # Th [-F,--data-urlencode] 'stream-index=<integer>' \ # The stream t
```

 Below the curl command, it states: "This is the full set of parameters supported by the operation." and "When sending large amounts of data or writing in parallel, prefer multipart/form-data over application/x-www-form-urlencoded as the Content-Type." At the bottom, it notes: "parallel-write-handle and stream-index only apply when writing to a replica in parallel. To obtain a parallel-write-handle, see [parallel_write_init](#)."

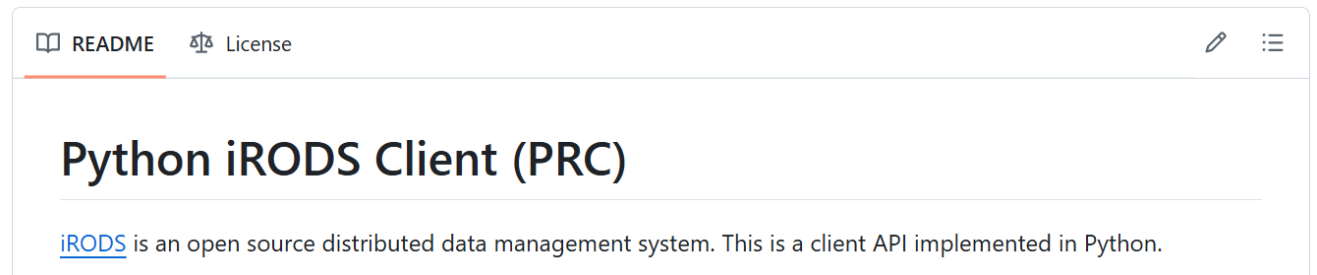
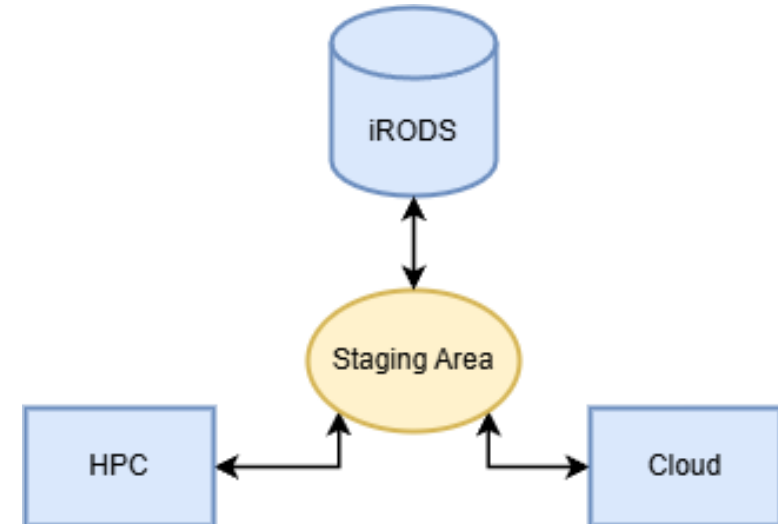
Previous data staging method

Data are moved in whole files in two steps with Staging Area:

- HPC → Staging Area → iRODS
- iRODS → Staging Area → HPC

Technologies used:

- **SCP** for HPC communication
- **Python iRODS Client** for iRODS communication

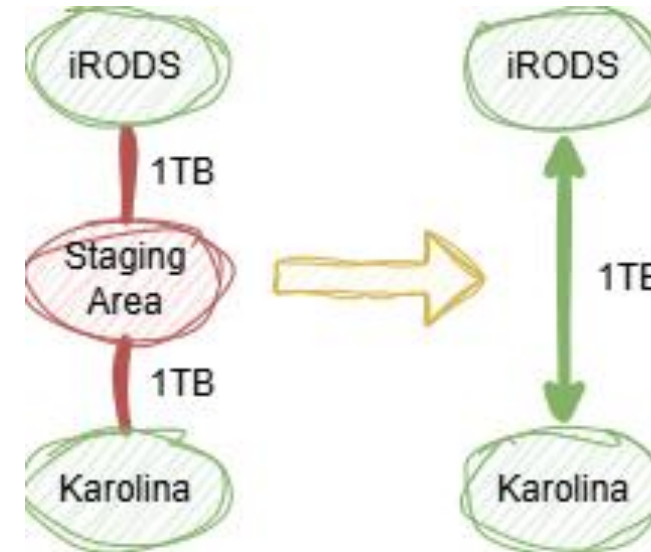


Chunked Transfer – Key Idea

- Transfer data chunk by chunk
- Avoid using the staging area

Two technological changes:

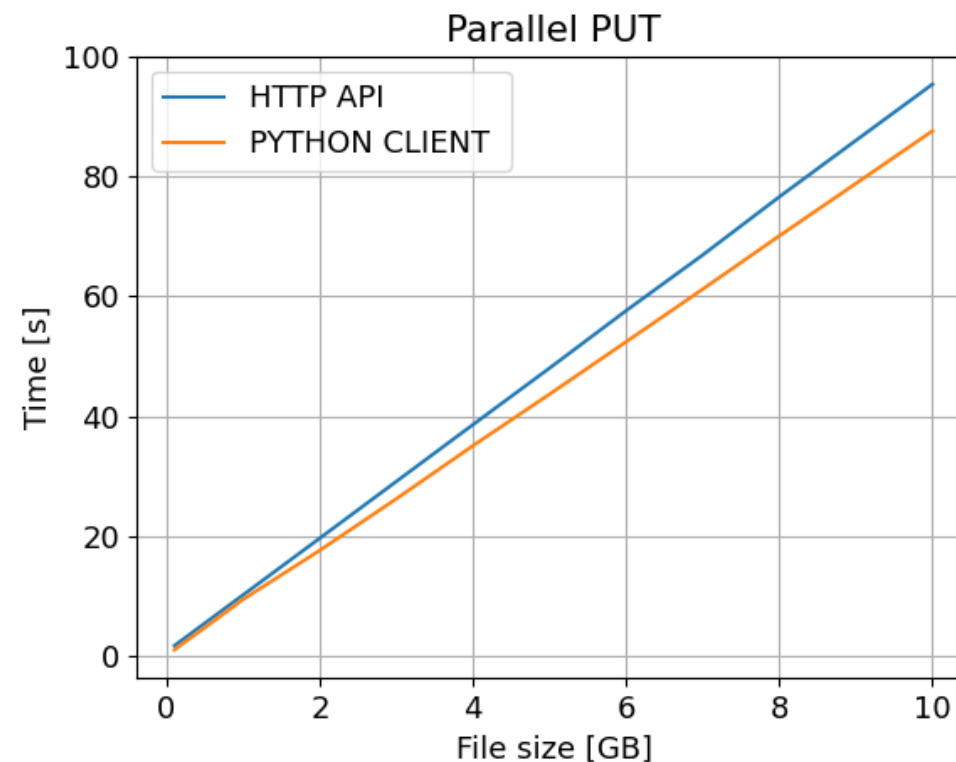
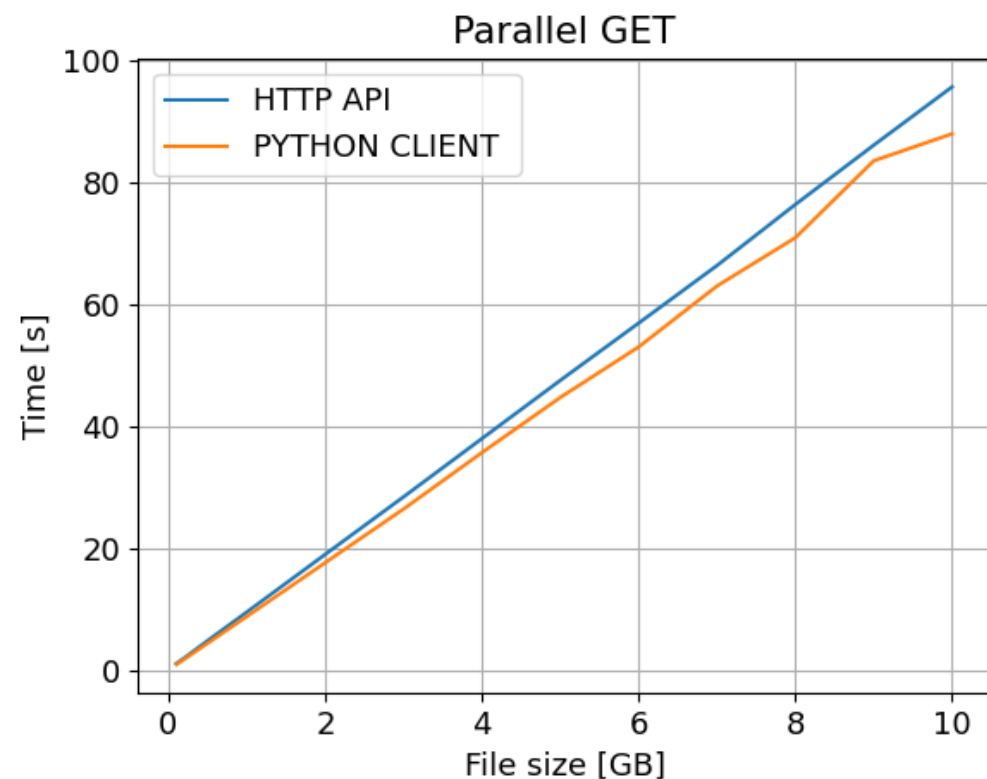
- **SFTP** instead of the SCP
- **Python iRODS HTTP API Client**



```
irods_fd = session.data_objects.open_async_writer(irods_file_path, workers_count=5)
with sftp_client.open(hpc_file_path, "r") as sftp_reader:
    while True:
        data = sftp_reader.read(chunk_size)
        if len(data) == 0:
            break
        irods_fd.write(data) # asynchronous write - parallel implementation
```

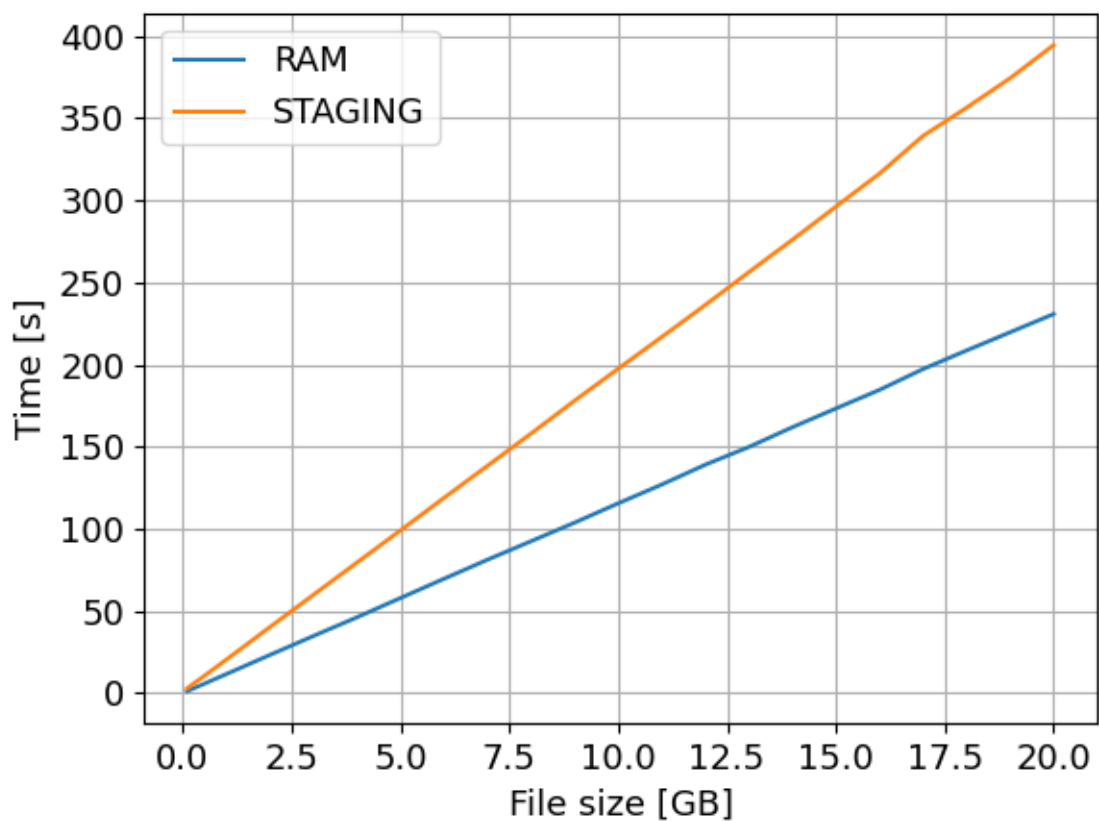
Performance tests

- Parallel GET / PUT
- **10 %** slowdown

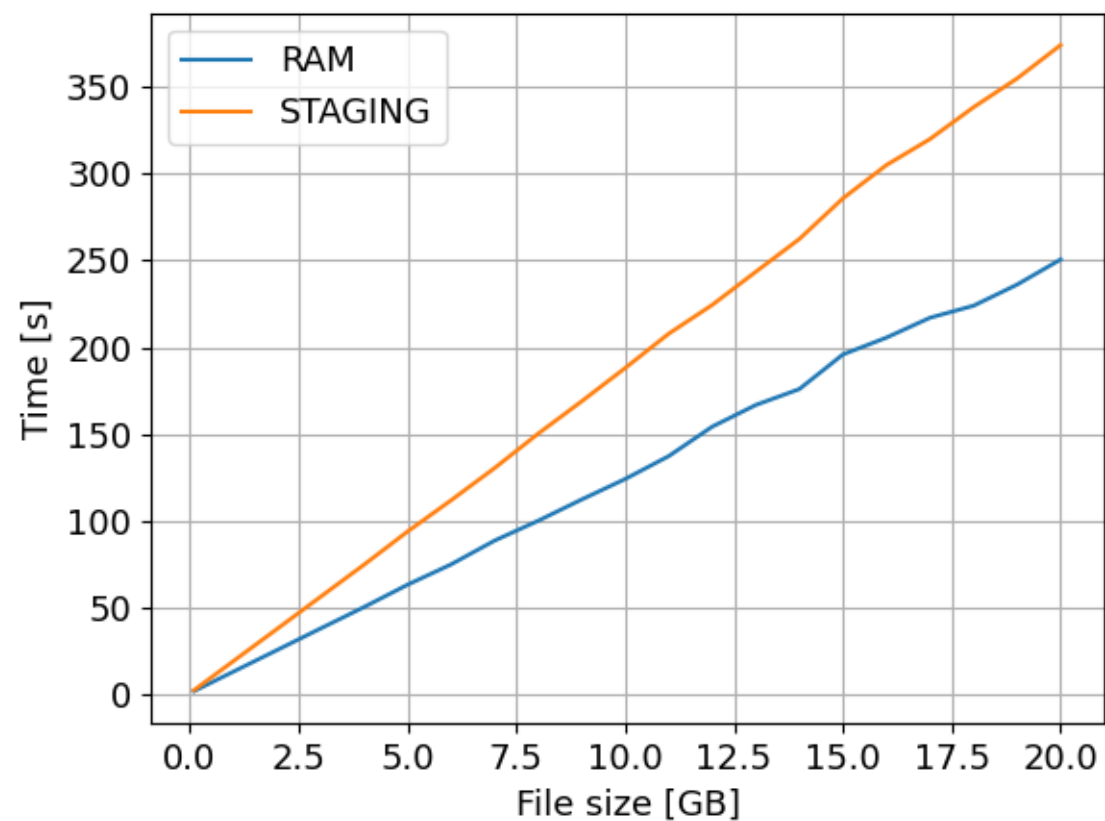


Chunked Transfers

Data transfer from HPC to iRODS



Data transfer from iRODS to HPC




Python HTTP API Client


Source code: <https://opencode.it4i.eu/lexis-platform/data/python-http-irods-client>

- OpenID support
- Parallel get / put
- Asynchronous file reader / writer

P **python-http-irods-client** 🌐

🔗 main ▾ python-http-irods-client / + ▾ Find file Edit ▾ Code ▾ ⋮

 **Reauthenticate fix**
Marek Nieslanik authored 1 week ago

4a054f54  History

```
obj = session.data_objects.parallel_write(file, irods_path, workers_count=4, chunk_size=128 * 1024)
```

```
afd = session.data_objects.open_async_writer("/tempZone/home/rods/result.txt", workers_count=2)

afd.write("test1\n") # thread 1
afd.write("test2\n") # thread 2
afd.write("test3\n") # thread 1
afd.write("test4\n") # thread 2

afd.close()
```

Airflow iRODS provider

Source code: <https://opencode.it4i.eu/exa4mind>



- Native iRODS integration with Apache Airflow
- Direct access to iRODS operations from Airflow DAGs
- Built as a standalone library

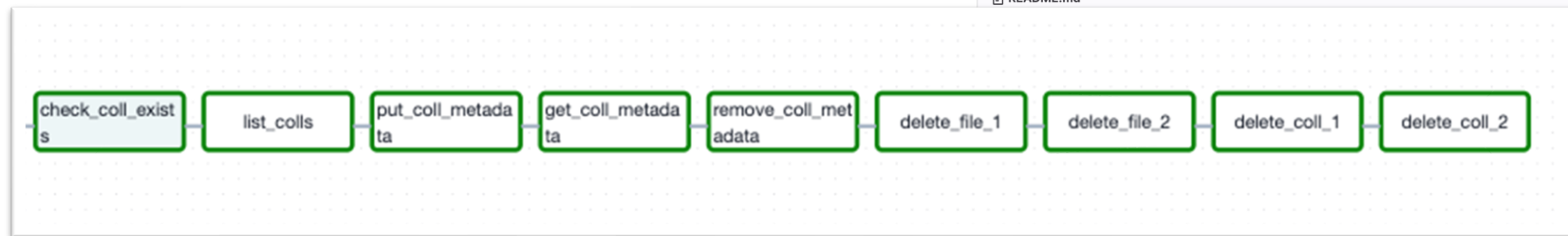
The screenshot shows the GitHub repository page for `airflow-provider-irods`. At the top, the repository name is displayed with a lock icon. Below it, there are tabs for `main` and `airflow-provider-irods`. A commit by `made address in pic neutral` is highlighted. Below the commit, a table lists the files in the repository:

Name	Last commit	Last update
images	made address in pic neutral	2 weeks ago
irods_provider	Update imports	5 months ago
tests	Adds skeleton	11 months ago
LICENSE	correct readmes, licenses, other glitches	2 weeks ago
README.md	Edit README.md	2 weeks ago
pyproject.toml	Adds working provider	10 months ago

Below the table, the `README.md` file is selected. The content of the README is displayed, showing a sequence of tasks in an Airflow DAG:

```
graph LR; A[check_coll_exists] --> B[list_colls]; B --> C[put_coll_metadata]; C --> D[get_coll_metadata]; D --> E[remove_coll_metadata]; E --> F[delete_file_1]; F --> G[delete_file_2]; G --> H[delete_coll_1]; H --> I[delete_coll_2];
```

At the bottom of the README, there is a disclaimer: "The EXA4MIND platform is currently under significant development. When using our modules, please consider security aspects. We are happy to receive feedback from you."



Example Airflow DAG

Main contributor: Mohamad Hayek (LRZ)

Summary & Next steps

- Improved DDI data staging infrastructure
- iRODS 4.3 roll-out in next month
- Expose more iRODS features in the web GUI (tickets)
- Scale the DDI to handle more users

Thank you!

Contact us at: support@lexis.tech



LEXIS Websites

www.lexis.tech
docs.lexis.tech

Multi-cluster
workflow execution

