



Martin Golasowski, Tobias Janča, Jan Martinovič

IT4Innovations, Czechia

Update on iRODS distributed data management
in the LEXIS Platform

Easy and Safe access to supercomputing

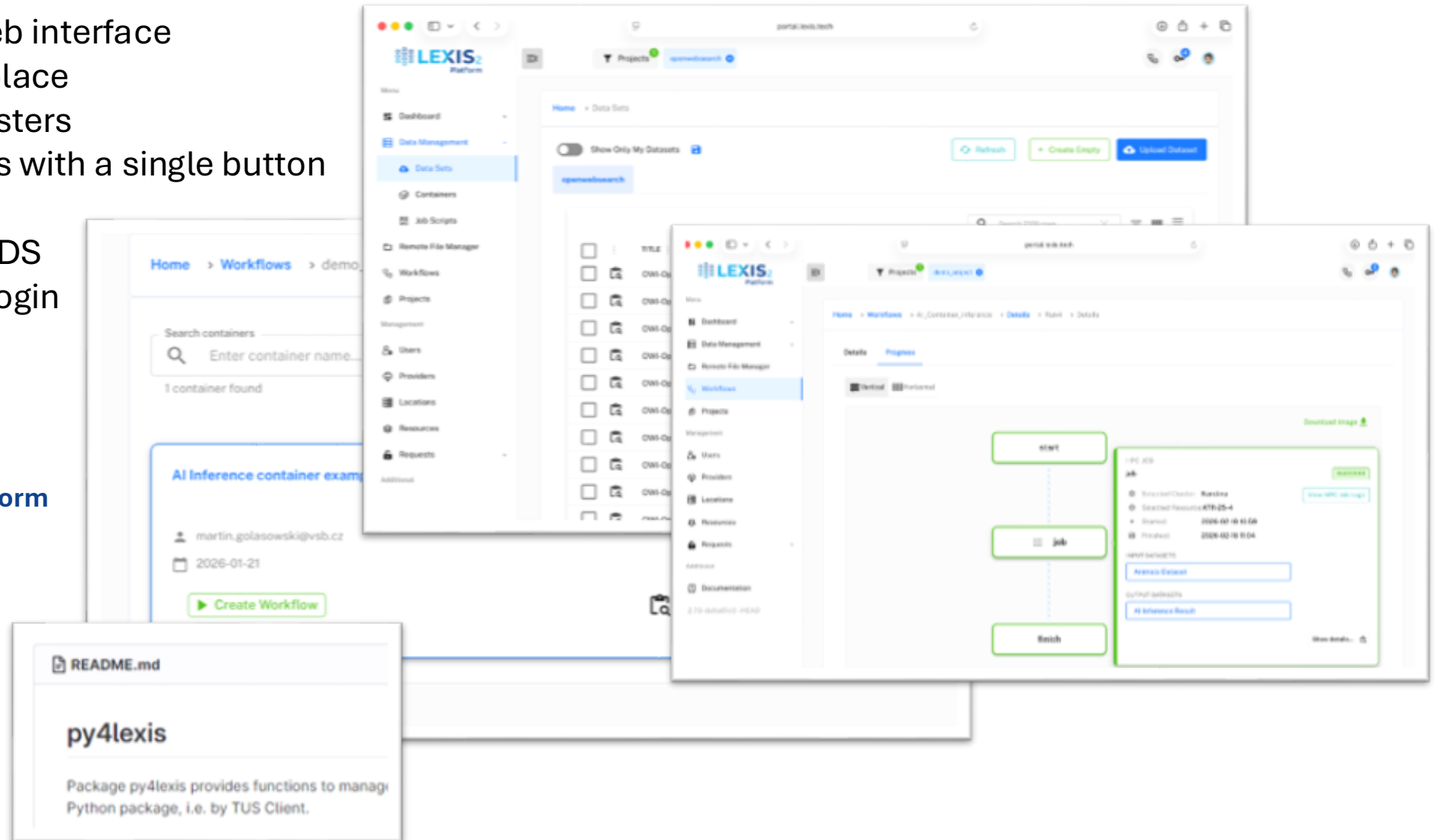


- Let's have a nice web interface
- Allocations at one place
- Across multiple clusters
- Launch applications with a single button
- Get HPC job logs
- Manage data in iRODS
- Use common web login
- ... and many more

Visit for more:

www.lexis.tech

opencode.it4i.eu/lexis-platform



How to run a simple HPC workflow



1.Upload your input data

2.Create your jobscript or upload a container

3.Create workflow

4.Run it

5.View results and logs

Prepare job
directory

Stage input data

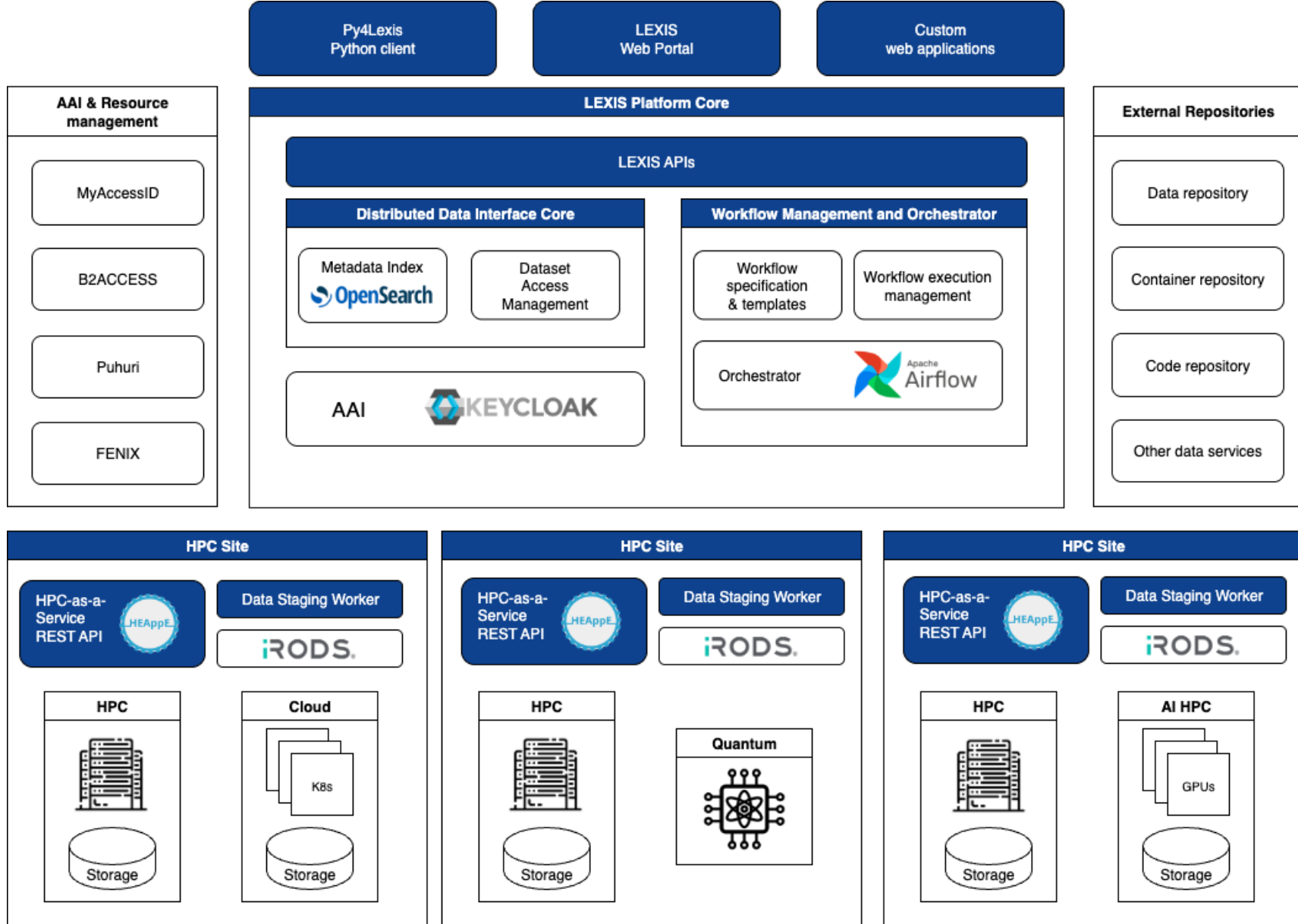
Submit an HPC
job

Wait until
finished

Download result

Simple HPC workflow steps

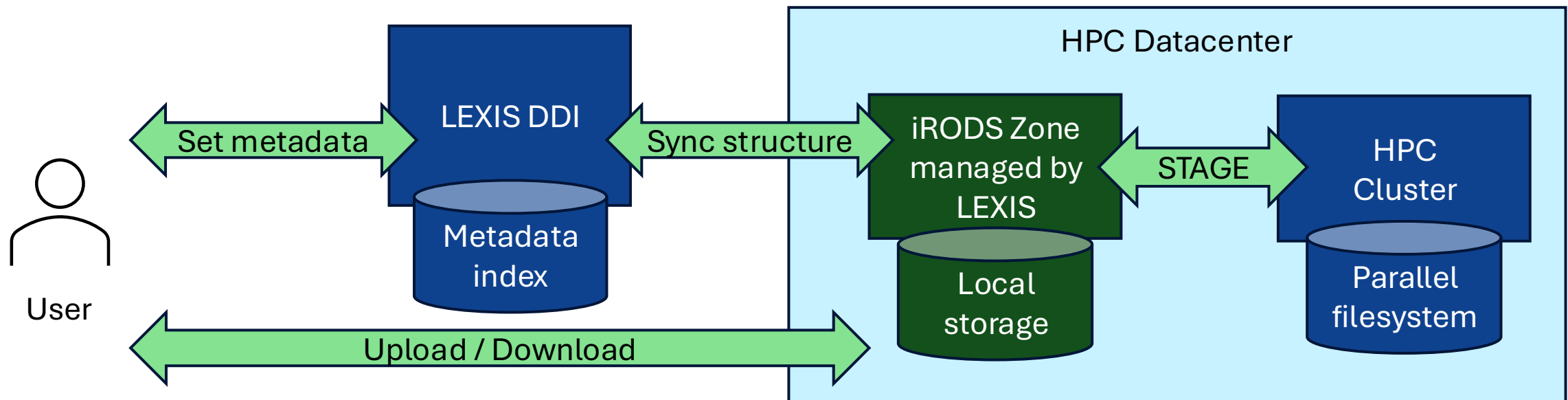
LEXIS Platform Architecture



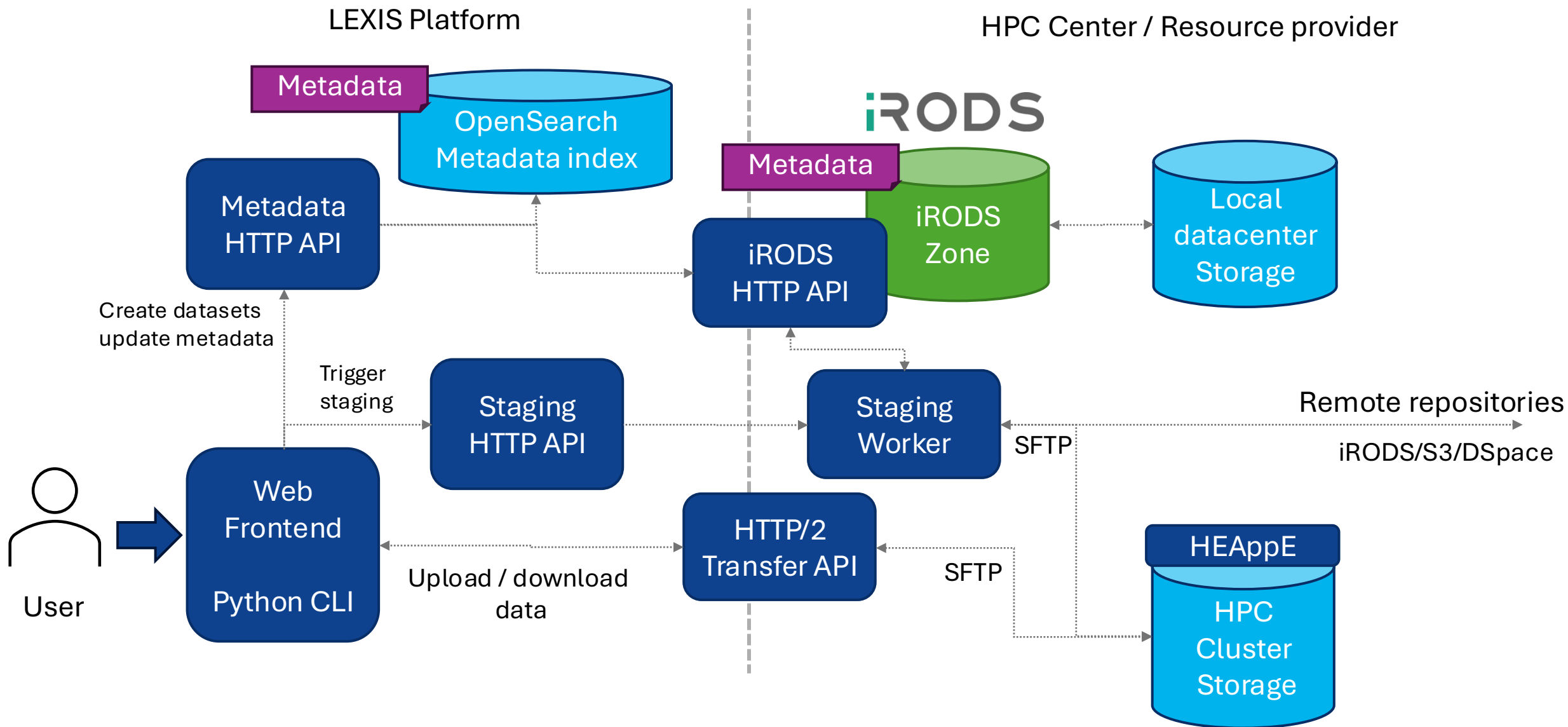
- **Hiding** of technical and **operational differences** across organizations
- **HPC & Cloud** service providers, Data providers
- Unified & distributed **data management**
- Part of **EuroHPC Federation Platform**
- **Connected to federated Authentication & Authorization Infrastructure** (MyAccessID)

iRODS Role in LEXIS Platform

- Unified representation of various storage backends
- Federated AAI through OpenID and HTTP API
- Allocation based access - suitable for HPC allocation model
- Ensures data locality – caching of remote datasets near compute



LEXIS Distributed Data Interface – DDI



Improvements in the LEXIS DDI

- Upgrade to iRODS 5.0.2 & HTTP API 0.6.0
 - iRODS 5.1 & HTTP API 0.7.0 in progress
 - Moved completely to GenQuery2
 - iRODS HTTP API improves usability significantly
- Code refactoring
 - Strongly typed code
 - Better CI/CD
- HTTP/2 Remote file management API
 - Direct SFTP transfers to HPC
 - Can be reused also as direct API to iRODS

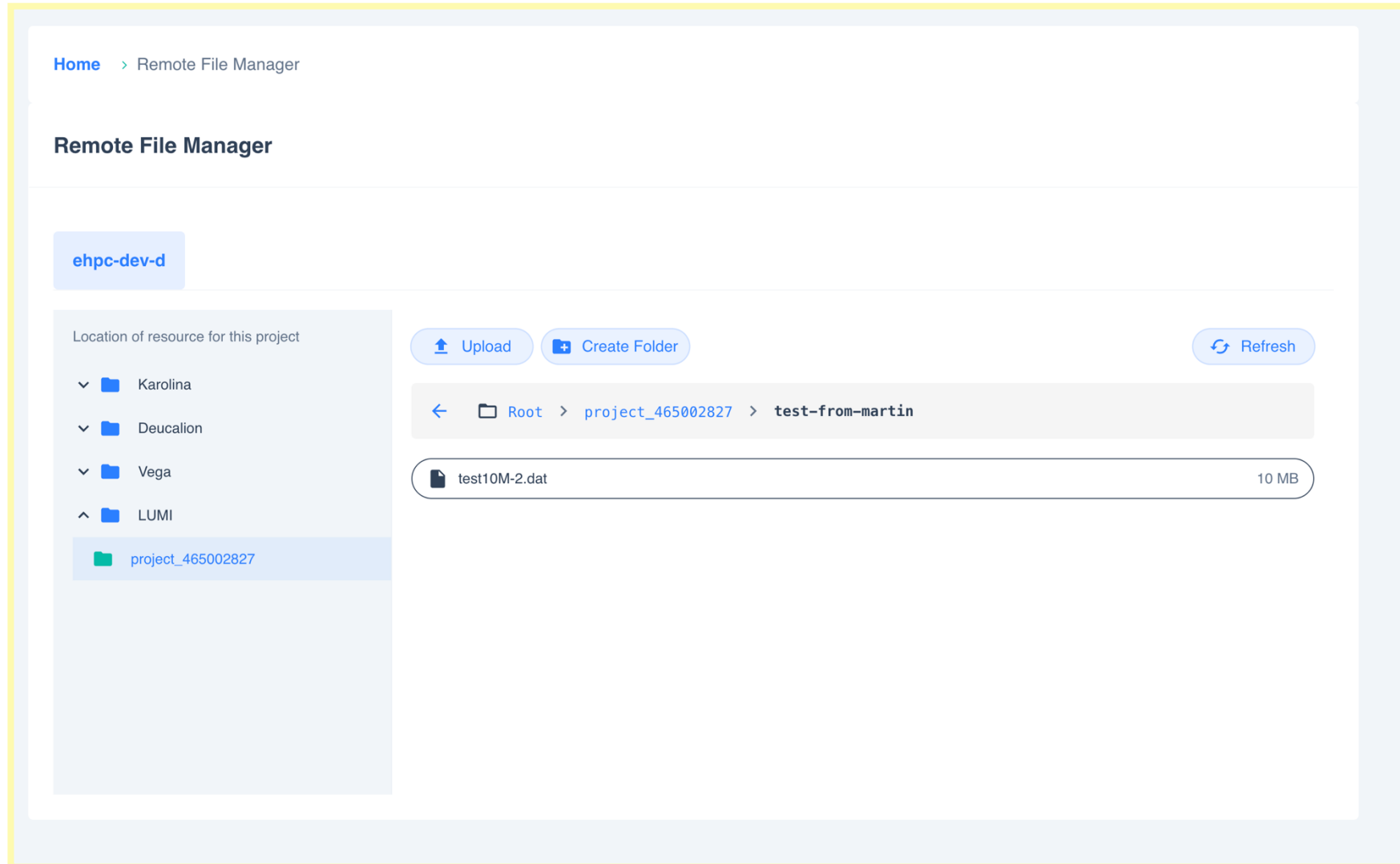
A screenshot of a code editor showing the file 'irods_dataset.py' with a size of 27.20 KIB. The code is Python and defines a function 'get_sizes_bulk'. It includes docstrings, SQL queries for GenQuery2, and logic for batching paths and executing queries. The code is line-numbered from 617 to 646.

```
617
618 def get_sizes_bulk(irods_client: IrodsHttpClient, absolute_paths: list[str]
619
620 """
621 Returns {path: size} for the given dataset collections (summed recurs
622 'get_sizes_bulk' SpecificQuery. Empty collection -> 0, missing path -
623 Long path lists are batched to stay under the iRODS GET-URL limit.
624
625 SpecificQuery to register per zone (iadmin asq, no trailing ';'):
626 WITH p AS (SELECT path FROM unnest(string_to_array(?, '|')) AS t(
627 dedup AS (SELECT DISTINCT ON (data_id) data_id, coll_id, dat
628 SELECT p.path, COALESCE(SUM(d.data_size), 0) AS size
629 FROM p JOIN r_coll_main c ON (c.coll_name = p.path OR c.coll_name
630 LEFT JOIN dedup d ON d.coll_id = c.coll_id
631 GROUP BY p.path
632
633 """
634 sizes = {}
635 for batch in _chunk_paths_by_length(absolute_paths, IRODS_QUERY_ARGS_
636 # paths joined with '|' into ONE bind param; '\x01' args-delimit
637 args = "|".join(batch)
638 result = irods_client.queries.execute_specific_query(
639     name=BULK_SIZE_QUERY_NAME,
640     args=args,
641     args_delimiter="\x01",
642     count=len(batch) # avoid the default row limit truncating re
643 )
644 for row in result.get("data", {}).get("rows", []):
645     if len(row) >= 2:
646         path, size = row[0], row[1]
647         sizes[path] = int(size) if size else 0
648
649 return sizes
650
651 """
```

Lessons learned on iRODS transfers

- Native iRODS protocol
 - Great performance, parallel TCP, TLS
 - Lot of ports to be open (1247, 20000 – 20199) – complex setup in corporate networks
 - Unable to integrate with web logins natively
- HTTP API Frontend
 - Universal HTTP(S)
 - OpenID support – enables cloud-like access
 - Caveat with S3 storage resource – no parallel streaming, full file must be cached
- New HTTP/2 API frontend
 - Currently written in Python, will explore Go or Rust to improve perf.
 - Offloads stream and chunk handling to well known web servers and clients

Remote file management API



OpenWebSearch.eu datasets



Dashboard

SERVICES

- Websites
- OURRS Search
- Crawling

EXPERIMENTAL

- Deep Research

UTILITIES

- Parse Preview
- YOARS EXTERNAL
- Impressum
- Privacy Statement
- Terms of Use
- Open Web Search Book



openwebindex.eu

Filter datasets... All Collections All Datacenters All Resources Public Sort by: Collection D

OWI - Open Web Index 8.2 GB

01/06/2026 → 01/06/2026 559 files

CURLIE_FULL IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at IT4I

owilix remote pull all/id=d1b020e8-5f3b-11f1-b1e0-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OWI - Open Web Index 39.8 MB

01/06/2026 → 01/06/2026 135 files

LICENSES IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at IT4I

owilix remote pull all/id=92a01354-5f3b-11f1-bcbf-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OWI - Open Web Index 96.2 GB

01/06/2026 → 01/06/2026 2,528 files

MAIN IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at IT4I

owilix remote pull all/id=67f076a8-5f3b-11f1-8a15-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OWI - Open Web Index 862.5 MB

01/06/2026 → 01/06/2026 380 files

LEGAL IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at IT4I

owilix remote pull all/id=67ea13d0-5f3b-11f1-ae03-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OWI - Open Web Index 4.2 GB

01/06/2026 → 01/06/2026 425 files

CURLIE_FULL IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at CSC2

owilix remote pull all/id=29b624e4-5ffa-11f1-b1e0-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OWI - Open Web Index 614.1 MB

01/06/2026 → 01/06/2026 315 files

LEGAL IT4I PUBLIC

All OWI data crawled between 2026-06-01 and 2026-06-01 at CSC2

owilix remote pull all/id=29b7e18a-5ffa-11f1-b01c-0e7110c447b2

VIEW STATISTICS

View Dataset License Open in LEXIS Portal

OpenWebSearch.eu - Public web indexes



The image displays a screenshot of the LEXIS2 Platform interface, specifically the 'Data Sets' section. The main content area shows details for the 'OWI-Open Web Index-curlie_full.owi@it4i-2026-06-28-2026-06-28' dataset. The interface includes a sidebar with navigation options like Dashboard, Data Management, Data Sets, Containers, Job Scripts, Remote File Manager, Workflows, Projects, Users, Providers, Locations, Resources, and Requests. The main content area displays a table with columns for Absolute Path, Last Modified Date, Creation Date, Zone, Resource Name, Resource Hash, Location Name, Access, and Project Shortname. A diagram is overlaid on the bottom left, illustrating the data distribution layer architecture. The diagram shows a flow from 'Crawlers' through a 'Common Component: Logging and Monitoring' and a 'Crawling queue::Frontier Apps' to a 'Copy-from-Warc' step. This leads to an 'Apache Storm Cluster' and a 'Cluster Manager', which then connect to 'Cloud Infrastructure'. The 'Cloud Infrastructure' feeds into the 'OWS Data Distribution Layer - Data Center specific bucket', which includes an 'iRODS API', an 'iRODS Zone', and an 'Object Store as a cache (S3)'. The 'Object Store as a cache (S3)' is connected to the 'iRODS Zone' via a bidirectional arrow. The diagram also shows a list of language folders on the right side of the interface, including 'language=cmn', 'language=lue', 'language=rnd', 'language=gaa', 'language=guw', 'language=kur', 'language=nlz', 'language=nlz', 'language=nlz', 'language=nlz', and 'language=nlz'. A 'Refresh' button is visible above the list. A 'Drag 'n' drop some files here' message is at the bottom right.

Diagram illustrating the data distribution layer architecture:

- 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)



EXA4MIND



LEXIS₂
Platform

AI Training app

- Web application focused on AI models lifecycle management, inference and visual workflows

The image displays two overlapping screenshots of the AI Training app interface. The left screenshot shows the main dashboard, and the right screenshot shows the workflow editor.

Dashboard Screenshot (Left):

- Header:** "AI TRAINING" logo and "Inference Service" status.
- Welcome Banner:** "Welcome to AI Training Portal" with a "Create New Operational" button.
- Summary Cards:**
 - 1 Workflow (In Inference)
 - 2 Datasets (Upload + Workflow)
 - 15 Projects (Active)
 - 4 Access Requests (Waiting Review)
- Recent Workflows:** List of workflows including "Student_project" (inference - 5 days ago).
- Active Projects:** List of active projects including "ATR-26-2: LEXIS Platform feature development and validation" (ends Jun 2027) and "EXA4MIND W4 Scientific" (ends Dec 2027).
- Quick Actions:**
 - New Workflow: Design a DAG-based training pipeline
 - View Executions: Monitor active and past runs
 - Upload Dataset: Add training data to your project
 - Workspace: Manage projects, users & resources
- Footer:** User profile "User martin.golasowski@v..."

Workflow Editor Screenshot (Right):

- Header:** "Active Project AI testovani" and "Create Workflow" button.
- Workflow Diagram:** A visual workflow showing inputs "Input: DATA_INPUT_SOURCE_1" and "Input: DATA_INPUT_SOURCE_2" feeding into a "job1" node (COMPUTE), which then outputs to "Output: DATA_OUTPUT_SOURCE_DIR".
- Components Panel:** A sidebar titled "Components" with a search bar and a list of available components:
 - Computation (5 items):
 - AI_Efficientnet: Execute Efficientnet model finetuning or...
 - AI_Unsloth: Execute Unsloth fine-tuning for Llama...
 - AI_Whisper: Execute Whisper model fine-tuning
 - AI_Yolo: Execute YOLO model
 - HPC_Job: Execute high-performance computing tasks
 - Inputs (4 items)
- Footer:** User profile "User martin.golasowski@v..."

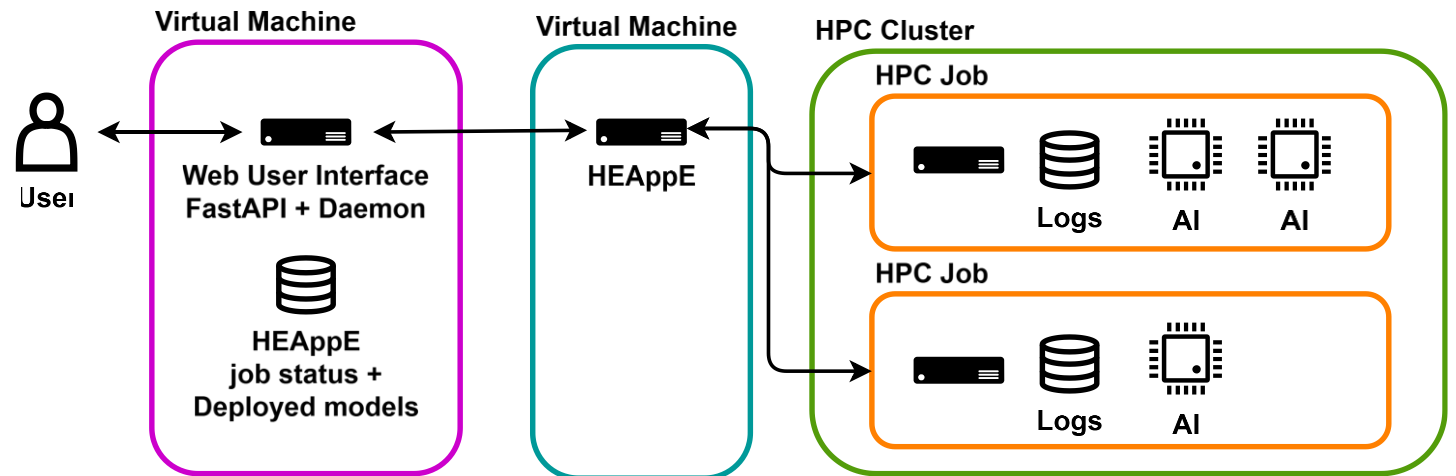
EXA4MIND AI HPC Inference service



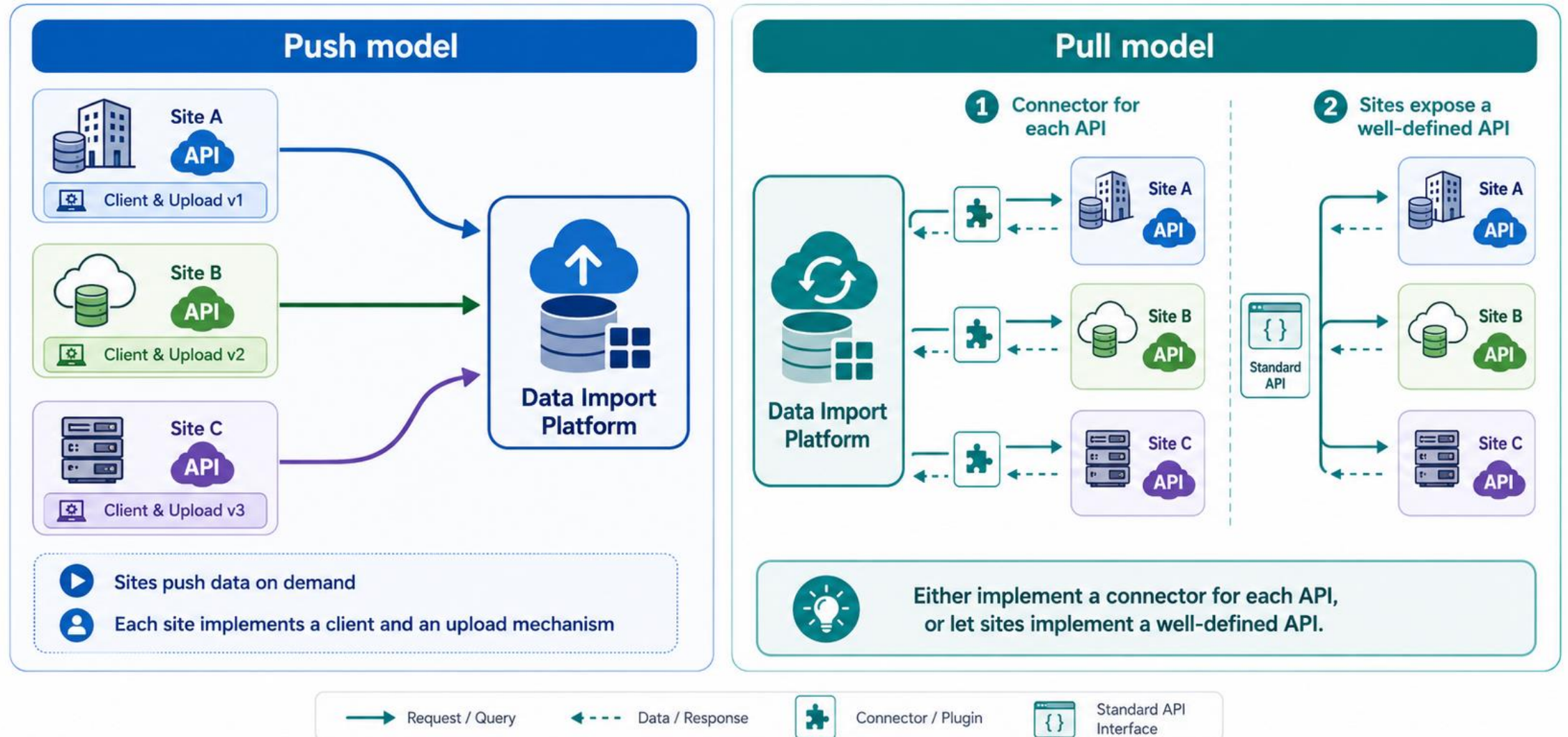
- Enables AI/LLM inference on supercomputers in HPC environments
- Interface to select models from Hugging Face
- Hides complexity of HPC job orchestration and resource management
- Uses HEAppE to run and monitor inference jobs across clusters

Source code:

<https://opencode.it4i.eu/exa4mind/platform/aqis/query-inference/ai-inference-service>



Dataset importing capabilities



Summary & Next steps

- Upgrade to latest iRODS version
- Streamline the data transfer APIs
- Implement the dataset importing capability
- Focus on AI oriented applications and integrations

Thank you!

Contact us at: support@lexis.tech



LEXIS Websites

www.lexis.tech

docs.lexis.tech

Multi-cluster
workflow execution

