Spoke 3 Archive Infrastructure

Massimo Costantini - INAF - OATs

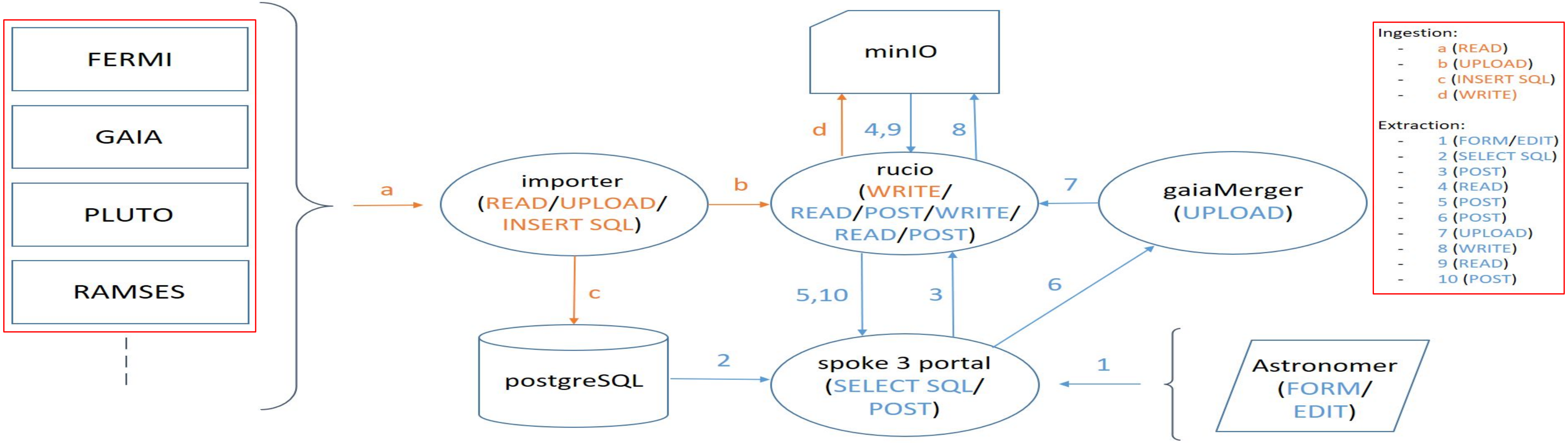
IVOA Interoperability Meeting, Edward St. John Learning and Teaching Center (ESJ)
Jun 2 – 6, 2025

The High-Performance Computing, Big Data, and Quantum Computing Research Center, established and managed by ICSC, is one of five National Centers created under the Italian National Recovery and Resilience Plan (PNRR) and funded by the European Union.

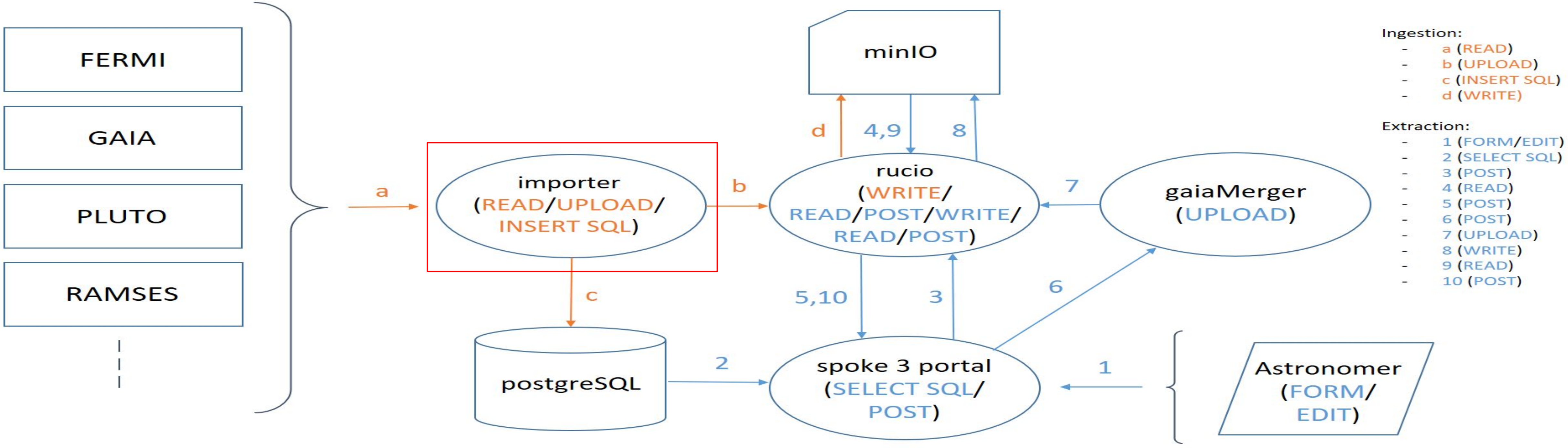
The main goals of the Spoke 3 project are to leverage state-of-the-art solutions in High-Performance Computing (HPC) and Big Data processing and analysis, to address challenges in the fields of Astrophysics and Cosmic Observation.

Spoke 3 is structured into several Work Packages (WPs) and this presentation focuses on WP4, which is dedicated to challenges related to Big Data Management, Storage, and Archiving.

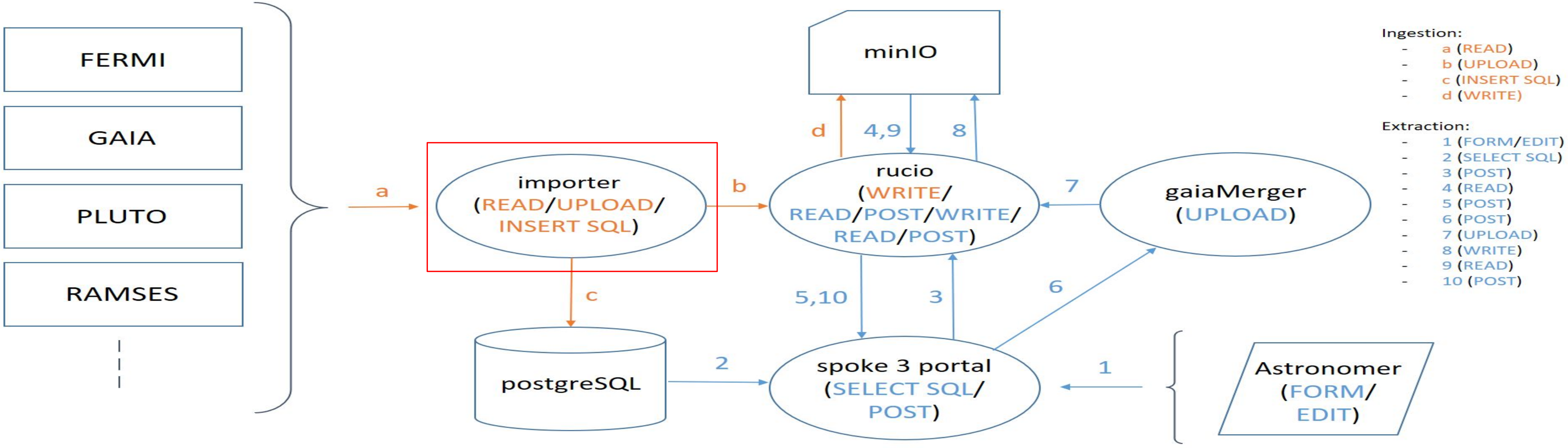
This "actors/actions" schema illustrates the aggregation of multiple data sources, including both observational and simulation-based data.
There are two main data workflows: Ingestion and Extraction.



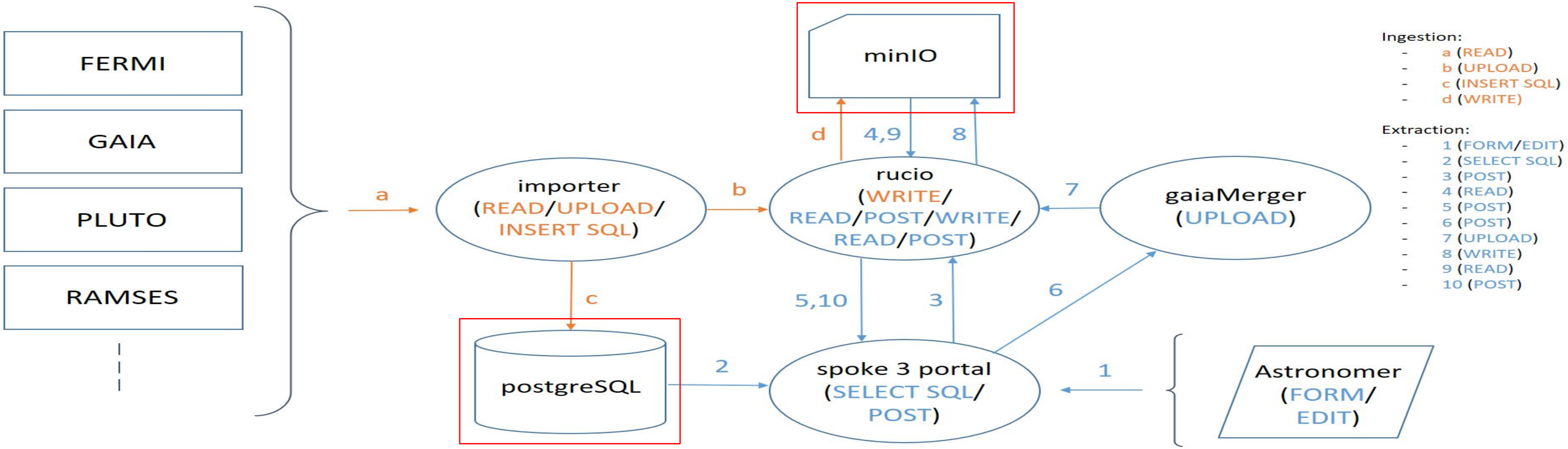
The Ingestion workflow starts by separating metadata from raw data by the "Importer", a software component implemented as a device and also monitored within the TANGO Controls distributed control system framework.



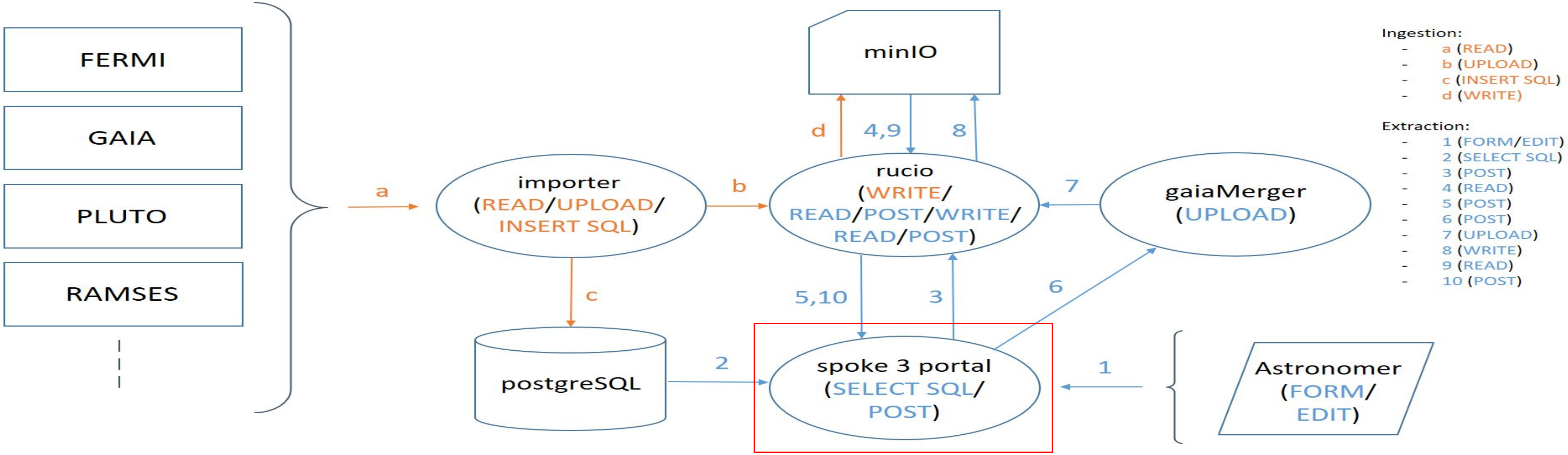
The initialized "Importer" instance retrieves the required data description model from a local MariaDB database and, using mapping tables, extracts key-value pairs through functions specifically developed for the various data formats used in our use cases.



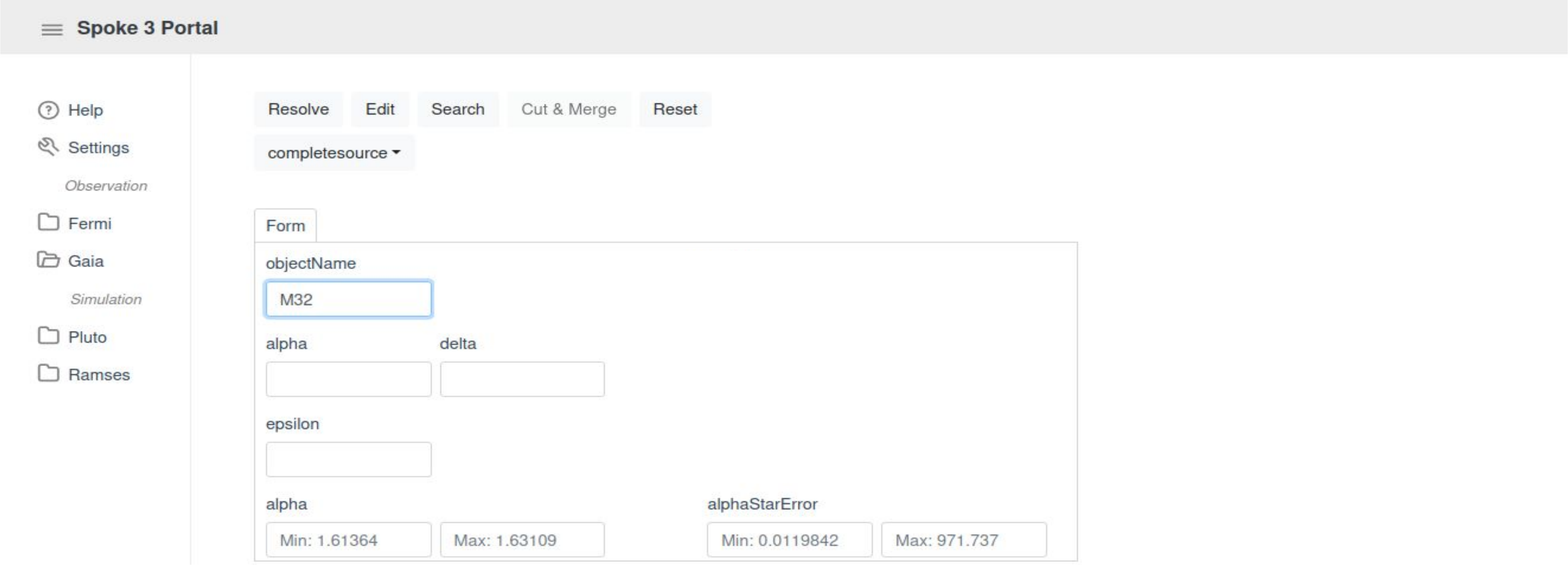
Metadata is written to a PostgreSQL database while the raw data is managed by a Rucio infrastructure for data storage and replication using a MinIO-based S3 object storage backend.
At this point, the Extraction workflow can begin.



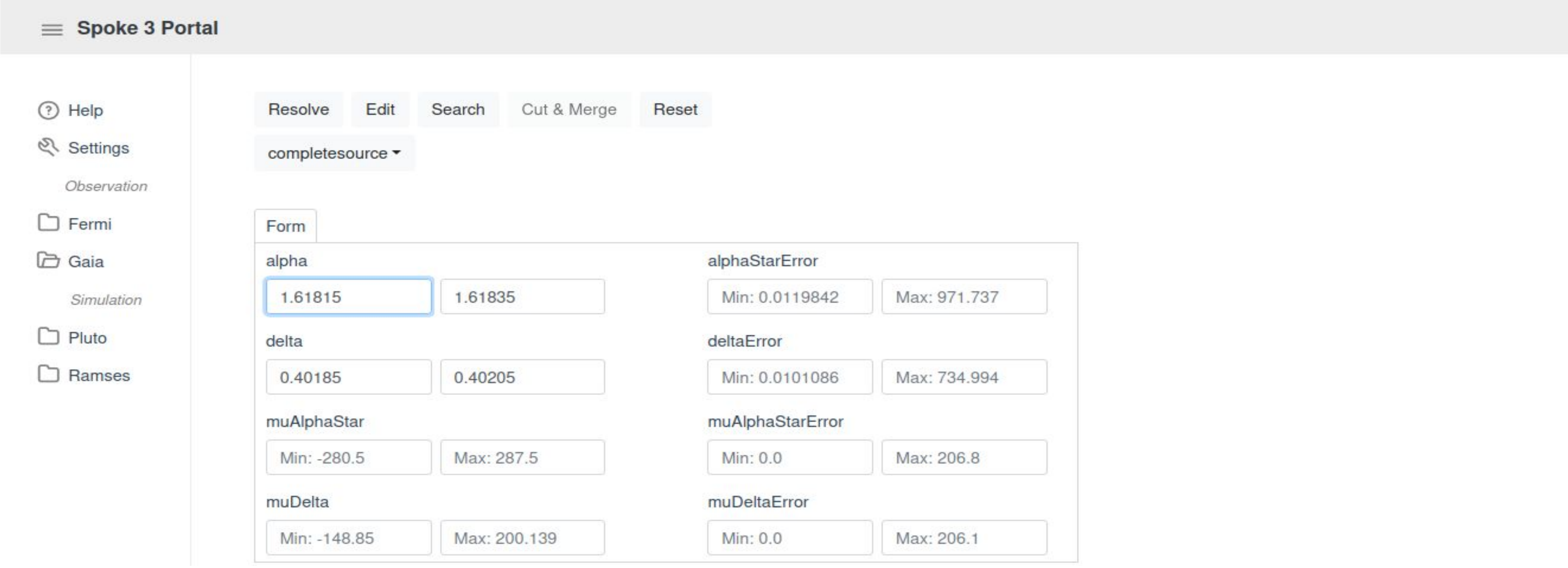
A custom portal allows astronomers to create queries to search for specific metadata and download only the files they are interested in. The portal also supports operations like "Cut & Merge" (in the Gaia use case) and provides access to Fermi Tools (in the Fermi use case).



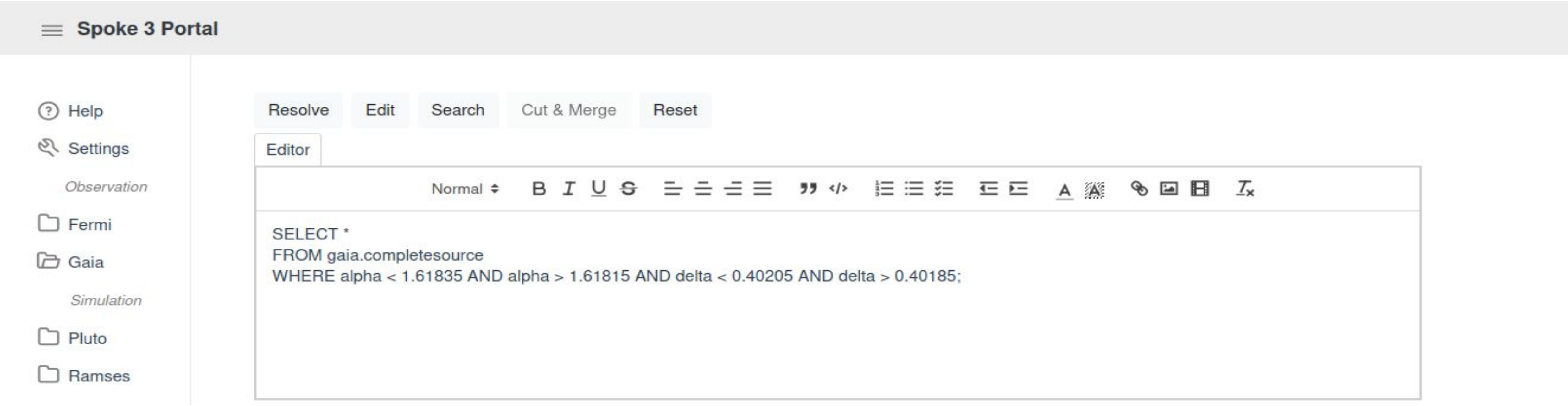
Through a form-based interface, users can, for example, select an objectName and retrieve the corresponding alpha, delta, and epsilon values, or select other metadata to generate a query.
An integrated editor also allows users to write and modify queries. [1]



Through a form-based interface, users can, for example, select an objectName and retrieve the corresponding alpha, delta, and epsilon values, or select other metadata to generate a query.
An integrated editor also allows users to write and modify queries. [2]



Through a form-based interface, users can, for example, select an objectName and retrieve the corresponding alpha, delta, and epsilon values, or select other metadata to generate a query.
An integrated editor also allows users to write and modify queries. [3]



In the Gaia use case, the query returns two tables. The first (Sources) is actually the result of merging two queries: one to PostgreSQL to retrieve metadata, including the Rucio Data Identifier (DID), and one to Rucio to locate the corresponding data file. The second table contains Transits.

Sources

	file_name	source_id	alpha	alpha_star_error	delta
☐	CompleteSource_130097_0003	3425114617586286848	1.618320372472243	0.0252838789312494	0.40193937806911
☐	CompleteSource_130097_0003	3425114617586288128	1.6182774194347807	0.08808795660406121	0.4018892840683867
☐	CompleteSource_130097_0003	3425114617587515008	1.6182973429850582	2.864232984901267	0.4019187701892775
☐	CompleteSource_130097_0003	3425114823742974720	1.6182802384704114	20.517780727768972	0.4020303469978408
☐	CompleteSource_130097_0003	3425114823745120512	1.6182794166233065	0.534499046224391	0.4020499999301908

Transits

	file_name	source_id	transit_id	ac_win_coord	transit_time
☐	AstroElementary_134755_0006	3425114617586286848	71044382629193047	[6008, 10765, 7849, 23375, 5877, 2960, 16268, 27836, 28160, 4277]	2017-05-05 15:27:27.522425
☐	AstroElementary_134756_0006	3425114617586286848	71048474888300196	[9139, 14738, 21180, 27859, 3839, 32384, 20020, 18213, 13261, 19935]	2017-05-05 17:14:01.677219
☐	AstroElementary_134757_0006	3425114617586286848	71058215738283403	[32345, 14328, 19099, 11576, 21558, 31161,	2017-05-05

Messages

Search started: 9:16:48 AM
Total PostgreSQL time: 145.00 ms
Total Rucio time: 580.00 ms
Merge time: 514.00 ms
Render time: 4.00 ms
Total time: 1147.00 ms

At this point, users can select one or more rows and perform a “Cut & Merge” operation, producing a significantly smaller file to download. [1]

Sources

☐	file_name	source_id	alpha	alpha_star_error	delta
☑	CompleteSource_130097_0003	3425114617586286848	1.618320372472243	0.0252838789312494	0.4019393780691 1
☑	CompleteSource_130097_0003	3425114617586288128	1.6182774194347807	0.0880879566040612 1	0.4018892840683867
☐	CompleteSource_130097_0003	3425114617587515008	1.6182973429850582	2.864232984901267	0.4019187701892775
☐	CompleteSource_130097_0003	3425114823742974720	1.6182802384704114	20.517780727768972	0.4020303469978408
☐	CompleteSource_130097_0003	3425114823745120512	1.6182794166233065	0.534499046224391	0.4020499999301908

Transits

☐	file_name	source_id	transit_id	ac_win_coord	transit_time
☑	AstroElementary_134755_0006	3425114617586286848	71044382629193047	[6008, 10765, 7849, 23375, 5877, 2960, 16268, 27836, 28160, 4277]	2017-05-05 15:27:27.522425
☐	AstroElementary_134756_0006	3425114617586286848	71048474888300196	[9139, 14738, 21180, 27859, 3839, 32384, 20020, 18213, 13261, 19935]	2017-05-05 17:14:01.677219
☐	AstroElementary_134757_0006	3425114617586286848	71058215738283403	[32345, 14328, 19099, 11576, 21558, 31161,	2017-05-05

Messages

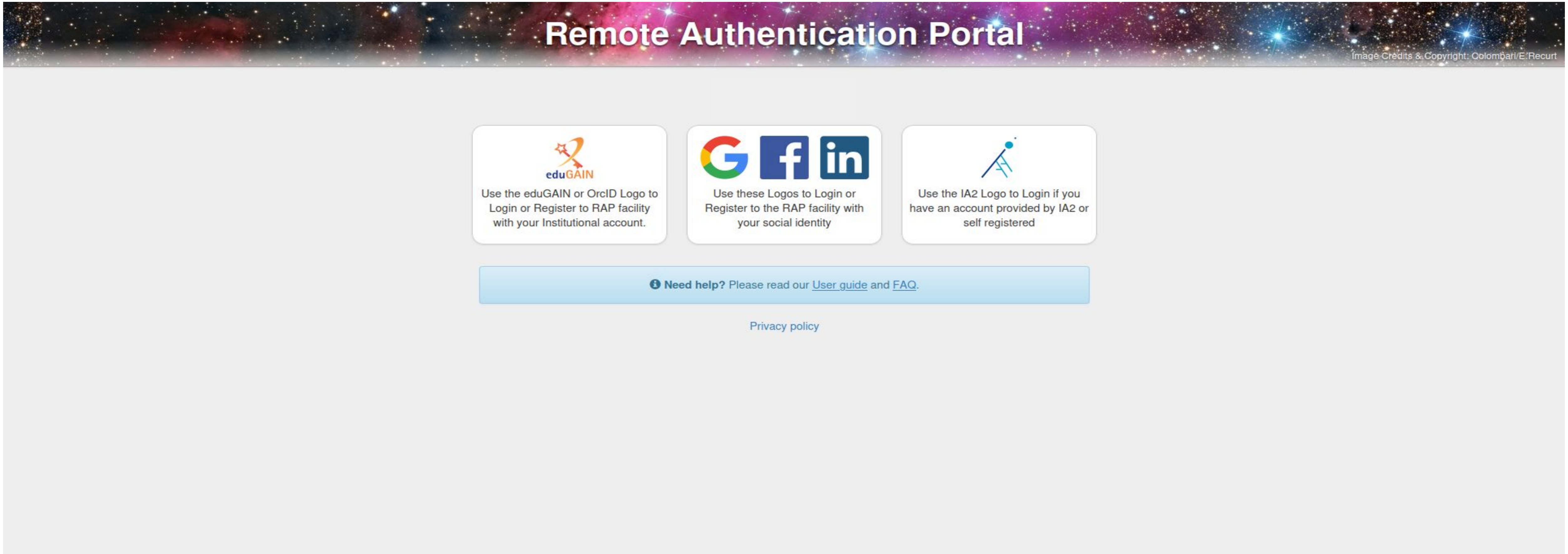
Search started: 9:16:48 AM
Total PostgreSQL time: 145.00 ms
Total Rucio time: 580.00 ms
Merge time: 514.00 ms
Render time: 4.00 ms
Total time: 1147.00 ms

At this point, users can select one or more rows and perform a “Cut & Merge” operation, producing a significantly smaller file to download. [2]

Cut & Merged

file_name
gaia_merger_4390e96bc45c191c225133de0a2e7792_1747639922.27298

Gaia data is not publicly accessible, so users must log into the Spoke 3 Portal via the RAP (Remote Authentication Portal) to obtain the necessary authentication and authorization credentials.



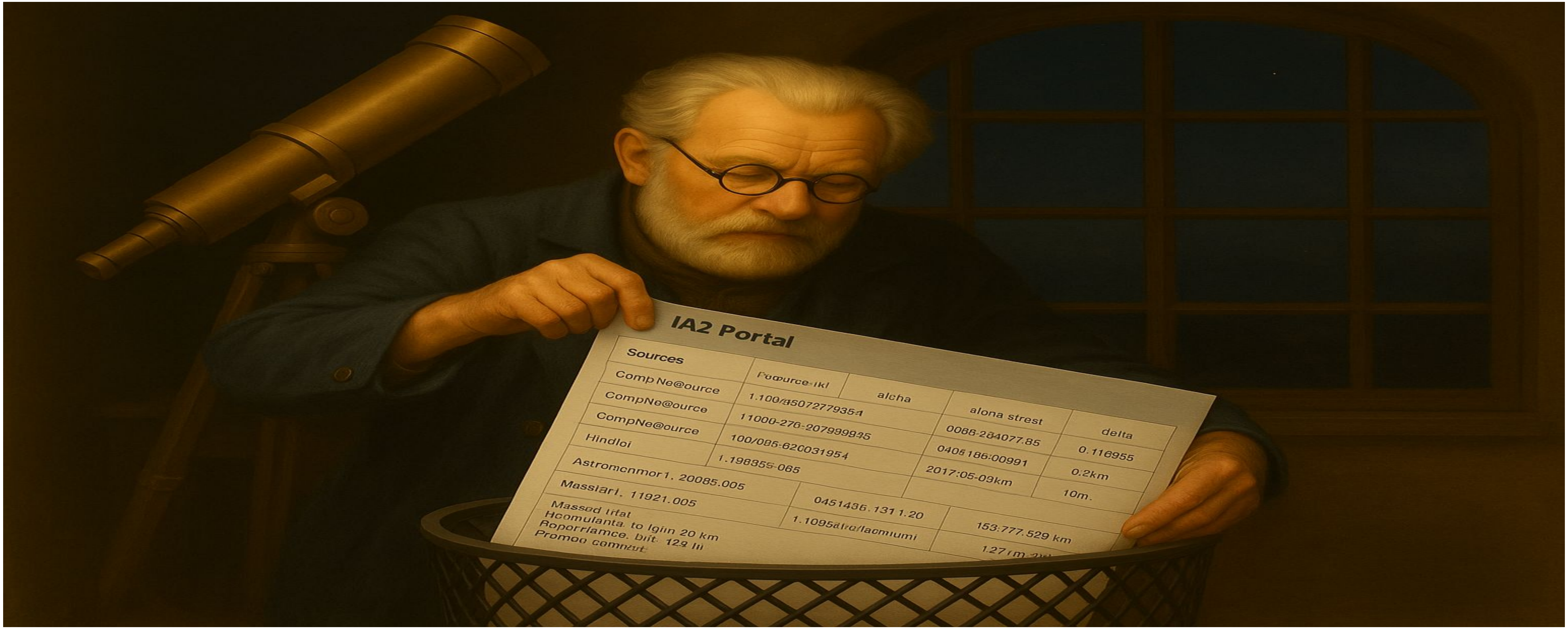
Additional features of the Spoke 3 Portal include:

- the ability to automatically correct the selection of transits that do not match any source;
- comprehensive logging of all operations (queries and downloads), collected using Elasticsearch and visualized through Kibana.

Key Points So Far:

- aggregations of both observational and simulation-based data through Ingestion and Extraction workflows;
- metadata is stored in PostgreSQL; raw data is managed by Rucio over a MinIO-based S3 object storage backend;
- a custom portal allows querying metadata and downloading only relevant files, supporting operations like “Cut & Merge” (for Gaia) or Fermi Tools access;
- users interact with the Spoke 3 Portal via forms or query editor;
- authentication via RAP is required to access private Gaia data;
- features include automatic correction of unmatched transits and full operation logging with Elasticsearch/Kibana.

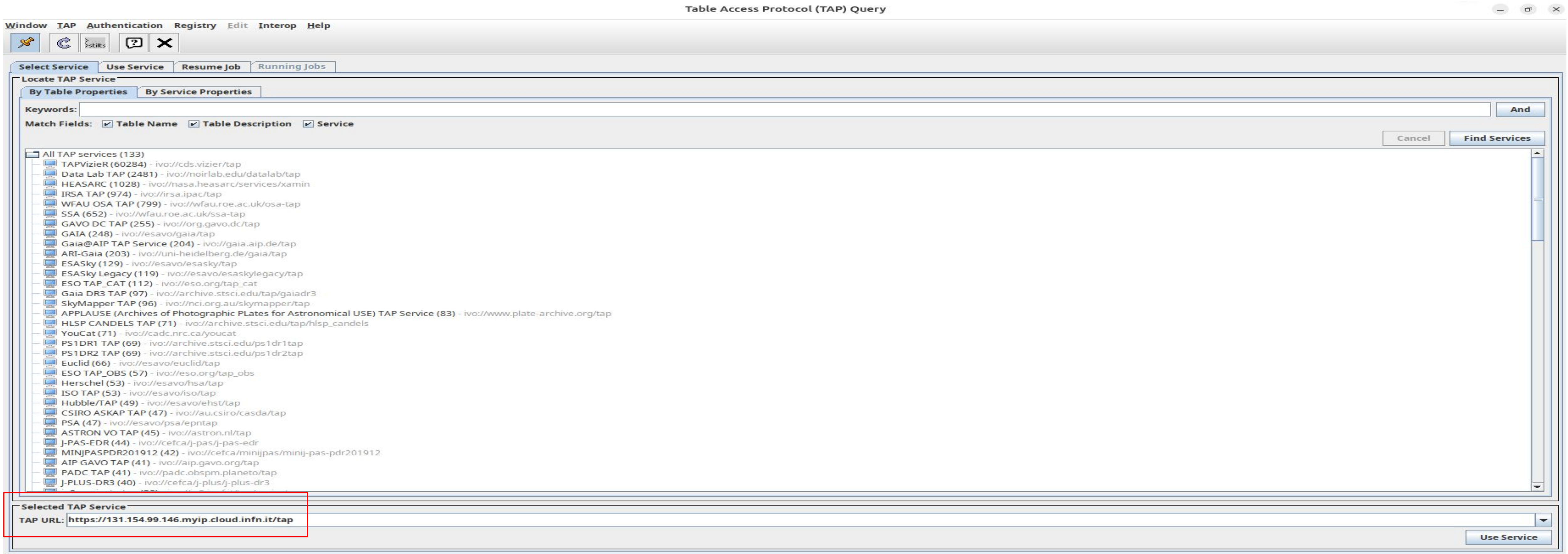
But what if the astronomer doesn't like my portal...?



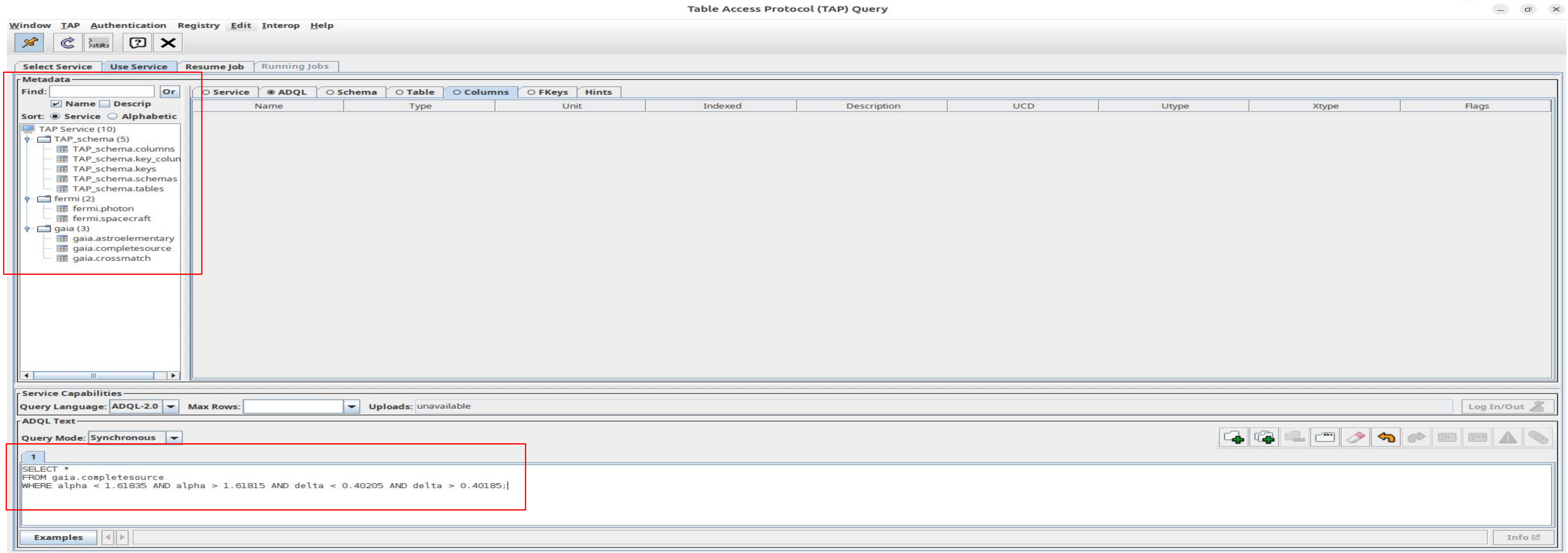
IVOA (International Virtual Observatory Alliance):

- **one of the microservices used by the Spoke 3 Portal is a technically compliant TAP (Table Access Protocol) exposing a DataLink table to enable all the custom services available for the query results;**
- **this allows users to perform the same operations previously described, but using other clients, like TOPCAT.**

By appending the /tap endpoint to the Spoke 3 Portal URL, users can access the TAP schema and the PostgreSQL tables directly from TOPCAT.
The integrated editor also allows query writing and editing. [1]



By appending the /tap endpoint to the Spoke 3 Portal URL, users can access the TAP schema and the PostgreSQL tables directly from TOPCAT.
The integrated editor also allows query writing and editing. [2]



In the result table, it is possible to select one or more rows, exactly as previously described with the Spoke 3 Portal, and create a subset.

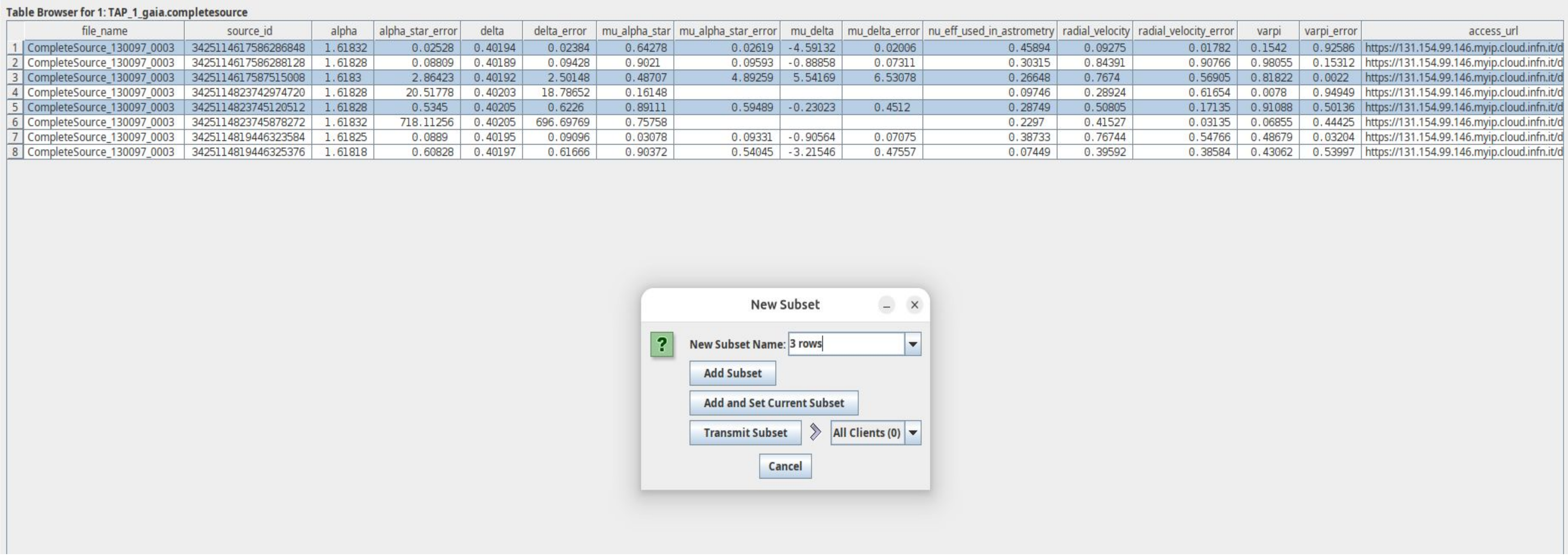


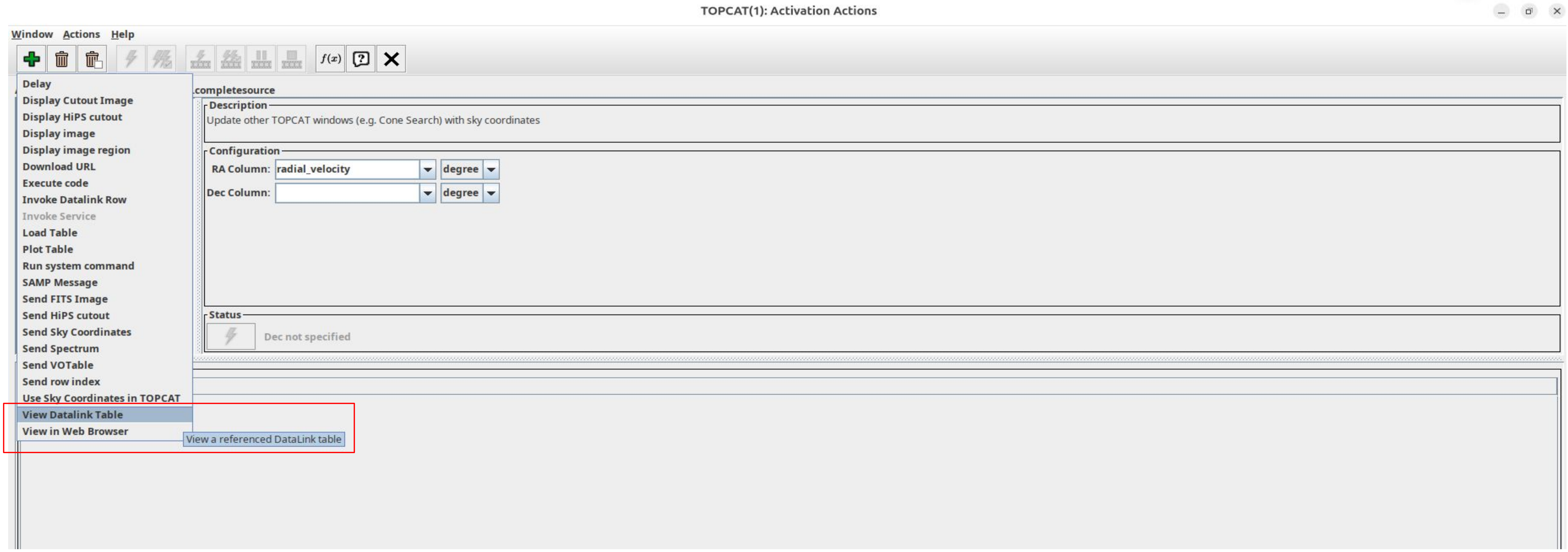
Table Browser for 1: TAP_1_gaia.completesource

	file_name	source_id	alpha	alpha_star_error	delta	delta_error	mu_alpha_star	mu_alpha_star_error	mu_delta	mu_delta_error	nu_eff_used_in_astrometry	radial_velocity	radial_velocity_error	varpi	varpi_error	access_url
1	CompleteSource_130097_0003	3425114617586286848	1.61832	0.02528	0.40194	0.02384	0.64278	0.02619	-4.59132	0.02006	0.45894	0.09275	0.01782	0.1542	0.92586	https://131.154.99.146.myip.cloud.infn.it/d
2	CompleteSource_130097_0003	3425114617586288128	1.61828	0.08809	0.40189	0.09428	0.9021	0.09593	-0.88858	0.07311	0.30315	0.84391	0.90766	0.98055	0.15312	https://131.154.99.146.myip.cloud.infn.it/d
3	CompleteSource_130097_0003	3425114617587515008	1.6183	2.86423	0.40192	2.50148	0.48707	4.89259	5.54169	6.53078	0.26648	0.7674	0.56905	0.81822	0.0022	https://131.154.99.146.myip.cloud.infn.it/d
4	CompleteSource_130097_0003	3425114823742974720	1.61828	20.51778	0.40203	18.78652	0.16148				0.09746	0.28924	0.61654	0.0078	0.94949	https://131.154.99.146.myip.cloud.infn.it/d
5	CompleteSource_130097_0003	3425114823745120512	1.61828	0.5345	0.40205	0.6226	0.89111	0.59489	-0.23023	0.4512	0.28749	0.50805	0.17135	0.91088	0.50136	https://131.154.99.146.myip.cloud.infn.it/d
6	CompleteSource_130097_0003	3425114823745878272	1.61832	718.11256	0.40205	696.69769	0.75758				0.2297	0.41527	0.03135	0.06855	0.44425	https://131.154.99.146.myip.cloud.infn.it/d
7	CompleteSource_130097_0003	3425114819446323584	1.61825	0.0889	0.40195	0.09096	0.03078	0.09331	-0.90564	0.07075	0.38733	0.76744	0.54766	0.48679	0.03204	https://131.154.99.146.myip.cloud.infn.it/d
8	CompleteSource_130097_0003	3425114819446325376	1.61818	0.60828	0.40197	0.61666	0.90372	0.54045	-3.21546	0.47557	0.07449	0.39592	0.38584	0.43062	0.53997	https://131.154.99.146.myip.cloud.infn.it/d

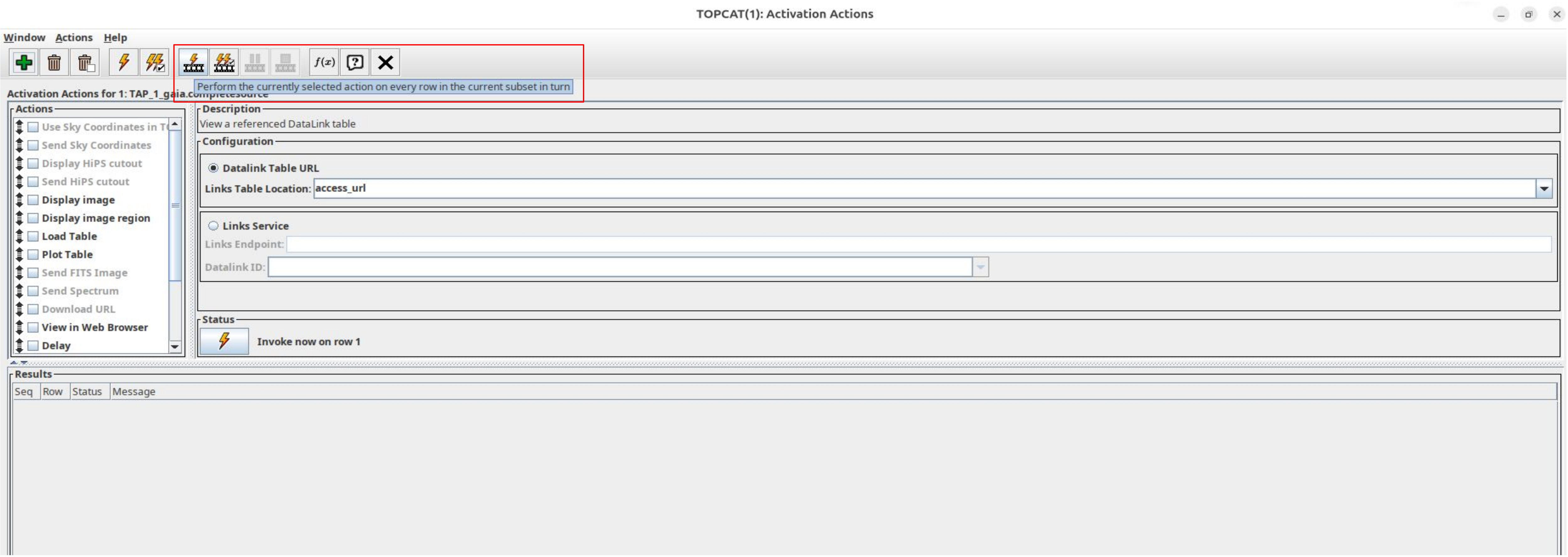
It is worth noting the column named access_url, which is required by TOPCAT to invoke the DataLink. This column embeds two UUIDs (Universally Unique Identifiers): the first is used to store the selected rows and queries, the second to create a personal cache for each user accessing TOPCAT.
Both operations are managed by Redis (the NoSQL in-memory database).

access_url
https://131.154.99.146.myip.cloud.infn.it/datalink?id=e7d8f16d-b7dc-4fc2-bd22-df0b3d1efd83&clientId=d8a25848-e8a8-468c-a1db-c5c0ae356802
https://131.154.99.146.myip.cloud.infn.it/datalink?id=cb372fee-d99d-4f24-be5c-2b314fc2e595&clientId=d8a25848-e8a8-468c-a1db-c5c0ae356802
https://131.154.99.146.myip.cloud.infn.it/datalink?id=1e8b4111-a35b-495d-88d0-5ba65992e4ab&clientId=d8a25848-e8a8-468c-a1db-c5c0ae356802

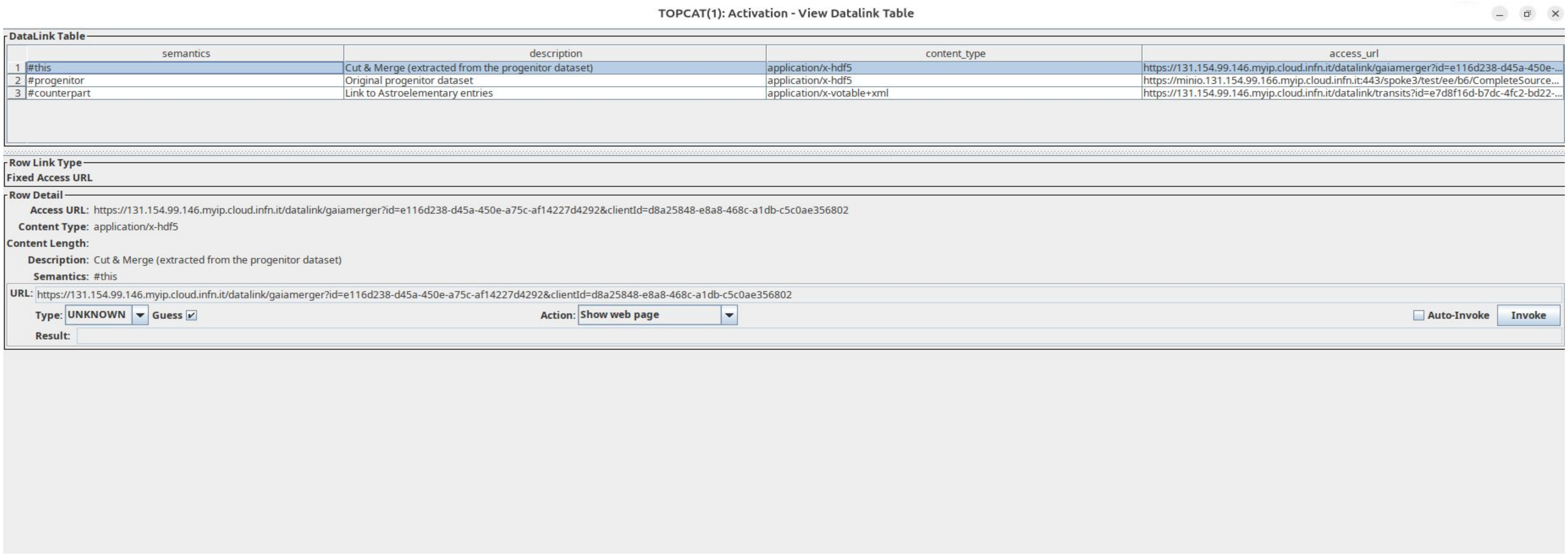
In the Activation Actions menu, it is necessary to select "View Datalink Table"...



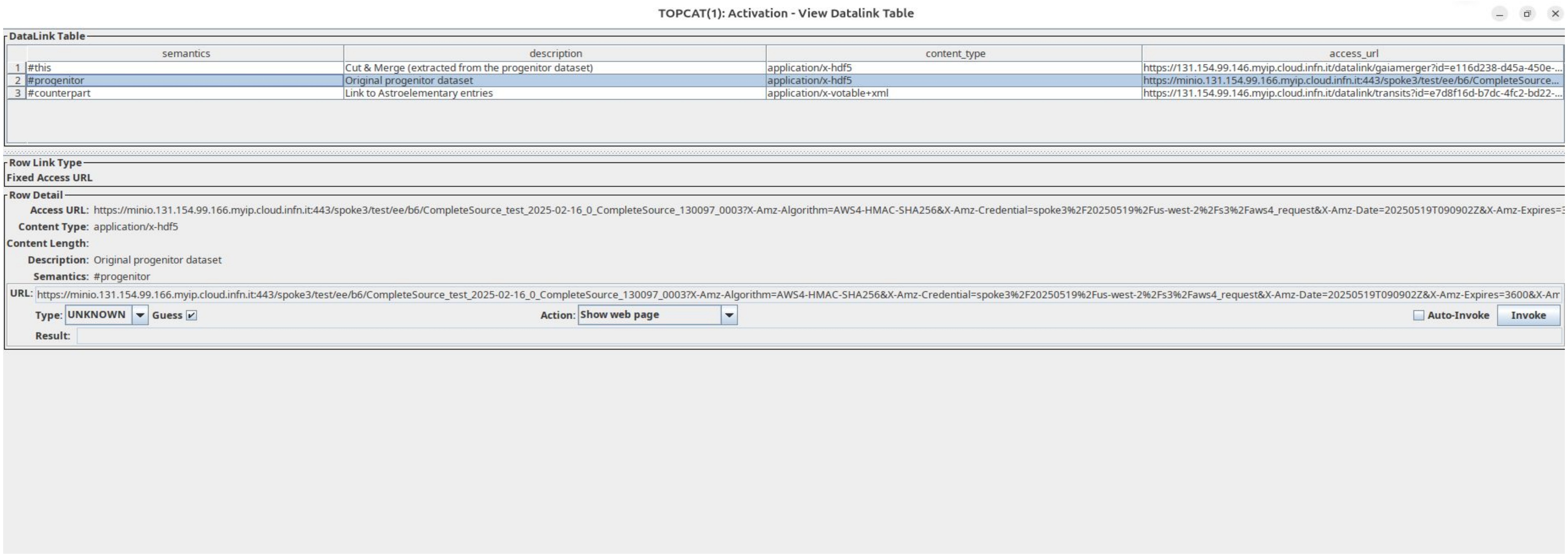
…and finally, invoke the DataLink.



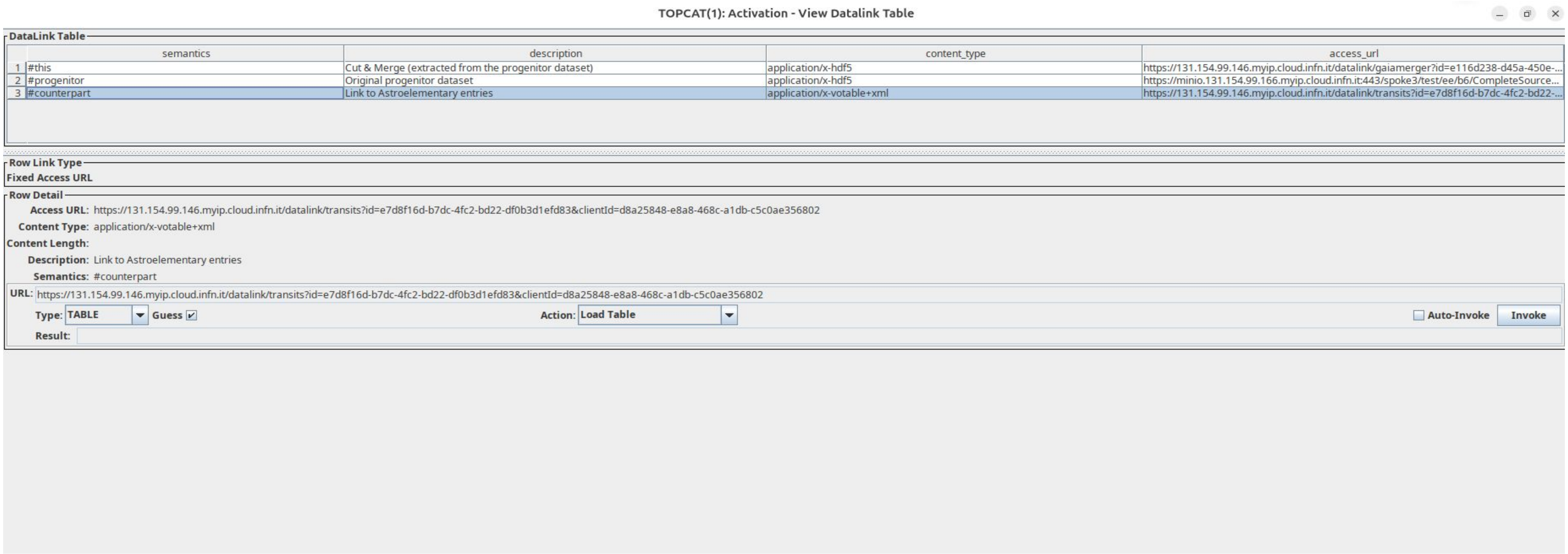
By selecting #this, a "Cut & Merge" operation can be performed, exactly as previously described with the Spoke 3 Portal. Since the content_type is HDF5, the file will be downloaded directly through the browser....



…if #progenitor is selected, the original file containing all sources or all transits will be downloaded…



…and finally, if #counterpart is selected, the TAP is accessed again to retrieve the Transits table, and then the entire process can be repeated: selecting one or more rows, creating a subset, and invoking the DataLink to either download the original file or perform a new "Cut & Merge" on the selected rows.



Key Points So Far:

- a separated microservice exposes TAP and a DataLink table, usable also via other clients like TOPCAT;
- the /tap endpoint is used to access the TAP schema and PostgreSQL tables;
- the queries return the Sources table with an access_url column containing two UUIDs, for Redis storage and user cache;
- users can select rows, create subsets, and invoke DataLink directly from TOPCAT;
- #this (to trigger a "Cut & Merge"), #progenitor (to download the full file), and #counterpart (to access the Transits table), are vocabulary terms that allow automated identification of the type of action to perform on the described query result;
- the Transits table provides a recursive DataLink solution, enabling the same workflow again.

And what if the astronomer doesn't like TOPCAT either...?

This cell initializes the Python client by importing required libraries, setting the base URL of the TAP server, and defining the query. It is equivalent to setting the /tap endpoint and compose a query in TOPCAT's ADQL panel.

[1]:

```python
# 1. Import libraries, define base URL, and set up the query

# Import libraries
from astropy.table import Table
import pandas as pd
import pyvo
from pyvo.dal.adhoc import DatalinkResults
from urllib.parse import urlparse, parse_qs

# Base URL of the TAP & DataLink service
BASE_URL = "http://localhost:3000"

# Query to run
query = """
SELECT *
FROM gaia.completesource
WHERE alpha < 1.61835 AND alpha > 1.61815
  AND delta < 0.40205 AND delta > 0.40185
"""
```

This cell queries the /tap endpoint to retrieve available schemas, tables, and columns. The XML response is parsed into a Pandas DataFrame. It is equivalent to TOPCAT's metadata loading when connecting to the TAP service. [1]

[2]:
```
# 2. Fetch available tables using PyVO

# Fetch tables
tap_service = pyvo.dal.TAPService(f"{BASE_URL}/tap")

# Iterate through all available tables
rows = []
for table_id, table_info in tap_service.tables.items():

    # Extract schema and table name
    if "." in table_info.name:
        schema, table = table_info.name.split(".", 1)
    else:
        schema = ""
        table = table_info.name

    # Iterate over columns and collect metadata
    for col in table_info.columns:
        rows.append({
            "schema": schema,
            "table": table,
            "column": col.name,
            "datatype": str(col.datatype)
        })

# Convert to a DataFrame and display
tables = pd.DataFrame(rows)
display(tables)
```

This cell queries the /tap endpoint to retrieve available schemas, tables, and columns.
The XML response is parsed into a Pandas DataFrame.
It is equivalent to TOPCAT's metadata loading when connecting to the TAP service. [2]

	schema	table	column	datatype
0	TAP_schema	TAP_schema.columns	arraysize	char
1	TAP_schema	TAP_schema.columns	datatype	char
2	TAP_schema	TAP_schema.columns	column_name	char
3	TAP_schema	TAP_schema.columns	table_name	char
4	TAP_schema	TAP_schema.columns	ucd	char
...	...	...	...	...
211	gaia	gaia.crossmatch	did_rse	char
212	gaia	gaia.crossmatch	id	int
213	gaia	gaia.crossmatch	file_version	int
214	gaia	gaia.crossmatch	flags	int
215	gaia	gaia.crossmatch	heal_pix_fov	int

216 rows × 4 columns

This cell sends the query to the /sync endpoint and retrieves the result as a VOTable. The response is parsed by Astropy and displayed as a Pandas DataFrame for inspection in the Jupyter notebook. This mirrors executing a query and viewing results in TOPCAT's table viewer. [1]

[3]:
```
# 3. Run query using PyVO and convert to Pandas via Astropy Table

# Execute the query
results = tap_service.search(query)

# Convert to Astropy Table
table = results.to_table()

# Convert to Pandas DataFrame for ease of use
df = table.to_pandas()
display(df)
```

This cell sends the query to the /sync endpoint and retrieves the result as a VOTable. The response is parsed by Astropy and displayed as a Pandas DataFrame for inspection in the Jupyter notebook. This mirrors executing a query and viewing results in TOPCAT's table viewer. [2]

	id	did_rse	did_scope	did_name	checksum	file_version	file_name	file_extension	key_name	
0	39432	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5369	34251
1	39438	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5370	34251
2	39441	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5371	34251
3	39635	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5430	34251
4	39640	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5431	34251
5	39642	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5432	34251
6	40112	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5573	34251
7	40118	TEST_USERDISK	test	CompleteSource_test_2025-02-16_0_CompleteSourc...	df04ee8881a1be3d3762ed41fe3551ca	0	CompleteSource_130097_0003	.h5	CompleteSourceImpl_5574	34251

This cell extracts the access URL from the TAP result row and queries the /datalink endpoint. It retrieves links to #this, #progenitor, and #counterpart replicating TOPCAT's behavior when invoking DataLink on selected table rows. [1]

[4]:
```
# 4. Use PyVO to access DataLink services and parse with Astropy Table

# Extract the access URL from the first result row
access_url = df.iloc[0]["access_url"]

# Create a DatalinkResults object from the URL
datalink = DatalinkResults.from_result_url(access_url)

# Convert to Astropy Table, then to Pandas DataFrame
datalink_table = datalink.to_table()
datalink_df = datalink_table.to_pandas()

# Reorder columns for consistency with TOPCAT
ordered_columns = ["semantics", "description", "content_type", "access_url"]
datalink_df = datalink_df[[col for col in ordered_columns if col in datalink_df.columns]]

# Display the DataFrame
display(datalink_df)
```

This cell extracts the access URL from the TAP result row and queries the /datalink endpoint. It retrieves links to #this, #progenitor, and #counterpart replicating TOPCAT's behavior when invoking DataLink on selected table rows. [2]

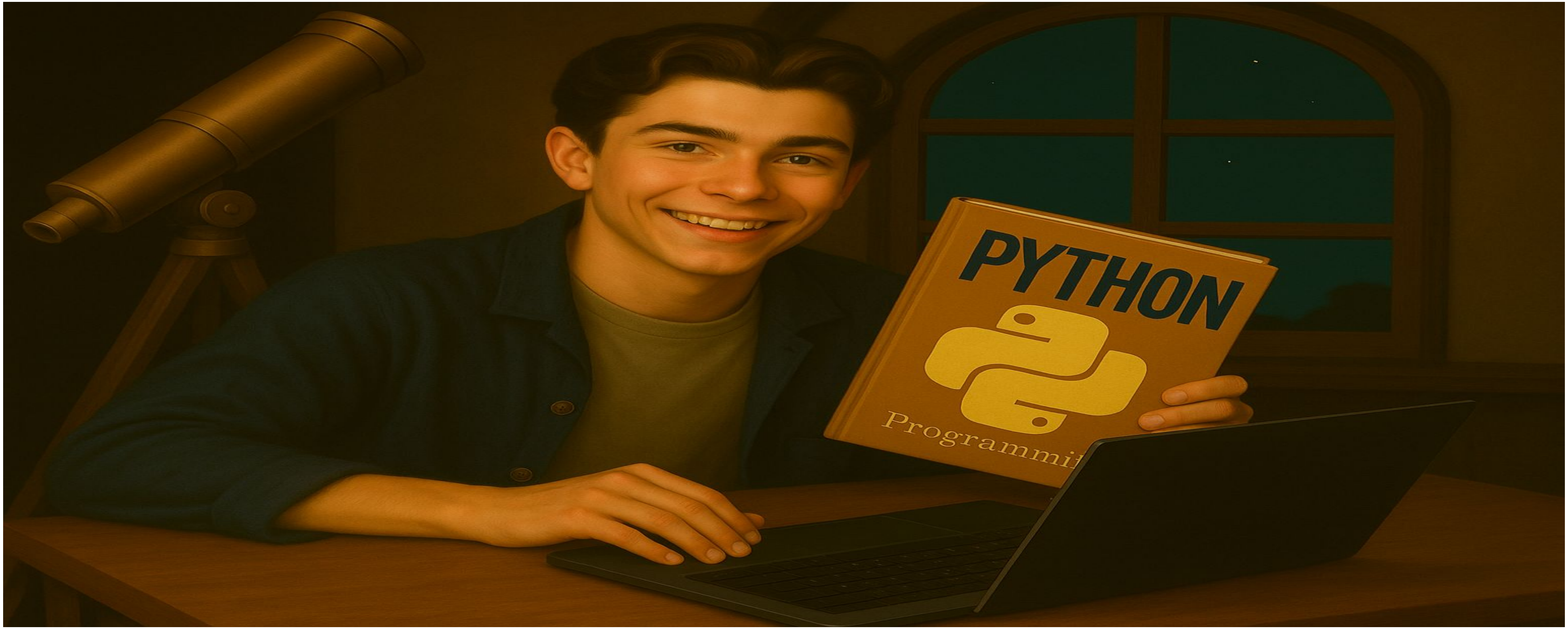
	semantics	description	content_type	access_url
0	#this	Cut & Merge (from progenitor)	application/x-hdf5	http://localhost:3000/datalink/gaiamerger?id=7...
1	#progenitor	Original progenitor dataset	application/x-hdf5	http://localhost:3000/api/files?name=CompleteS...
2	#counterpart	Link to transits	application/x-votable+xml	http://localhost:3000/datalink/transits?id=799...

[ ]:

Key Points So Far:

- As the entire workflow can be performed with TOPCAT, the same goes for Python with Astropy/PyVO, which is simply another example of VO-aware client.

Modern tools. Younger astronomers...!



Summary:

- the Spoke 3 Archive Infrastructure project ingests and extracts both observational and simulation-based data;
- metadata is stored in PostgreSQL, while raw data is managed using Rucio on a MinIO-based S3 storage backend;
- a custom portal enables advanced querying and custom operations like "Cut & Merge" (for Gaia) or access to Fermi Tools;
- the same features are exposed via TAP/DataLink microservices, compatible with other clients like TOPCAT;
- Python with Astropy/PyVO is just another example of a VO-aware client.

What's Next:

- comparative performance testing between Oracle (for Gaia) and PostgreSQL, including high-load scenarios;
- evaluation of sharding strategies for distributing data across multiple databases or servers, improving performance and scalability;
- future extension of the portal to support new astrophysical datasets and use cases;
- furthermore, resources and services described are not registered, data concerning Pluto and Ramses are still being analyzed, ADQL features are not yet implemented.

Any questions…?