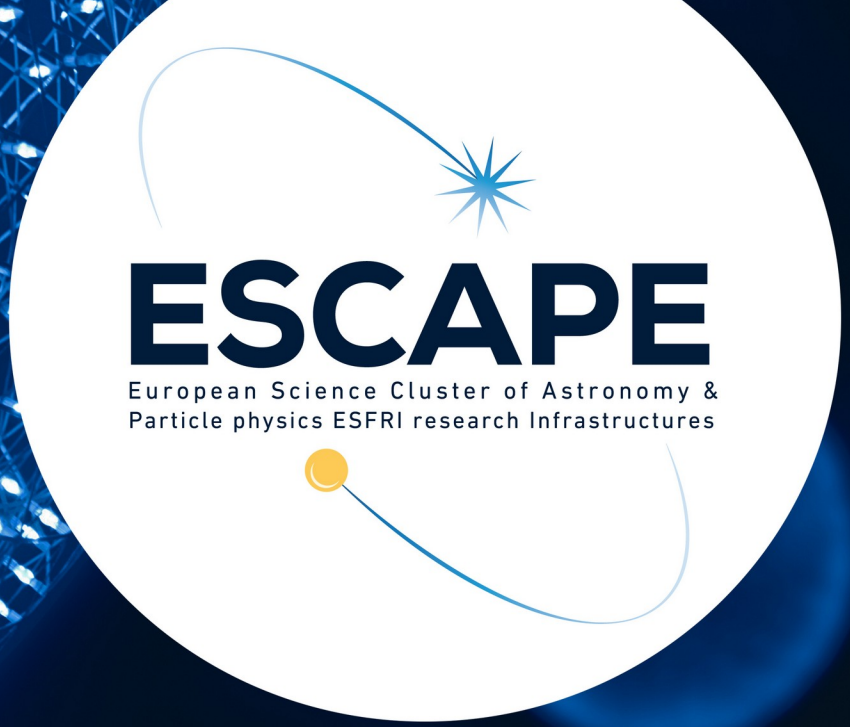


IVOA standards in ESCAPE-ESAP functionalities

S. Bertocco, K.Kliffen, D.Morris, S.Voutsinas, S.Sanchez, M.Parra, J.Swinbank

IVOA Interoperability Meeting

25-29 April 2022

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

ESCAPE EU Project

European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures

- **Data Lake (DIOS: Data Infrastructure for Open Science):**

Build a scalable, federated, data infrastructure for open science for the ESFRI
Enable connection to compute and storage resources.

- **Software Repository (OSSR: Open-source Scientific Software and Service Repository):**

Open source scientific software and service repository.

- **Virtual Observatory (CEVO: Connecting ESFRI projects to EOSC through VO framework):**

Extend FAIR standards, methods, tools of the Virtual Observatory; demonstrate EOSC ability to include existing platforms

- **Science Platforms (ESAP: ESFRI Science Analysis Platform):**

Flexible science platforms to enable the analysis of open access data

- **Citizen Science (ECO:Engagement and COmmunication):**

Open gateway for citizen science on ESCAPE data archives and ESFRI CS projects

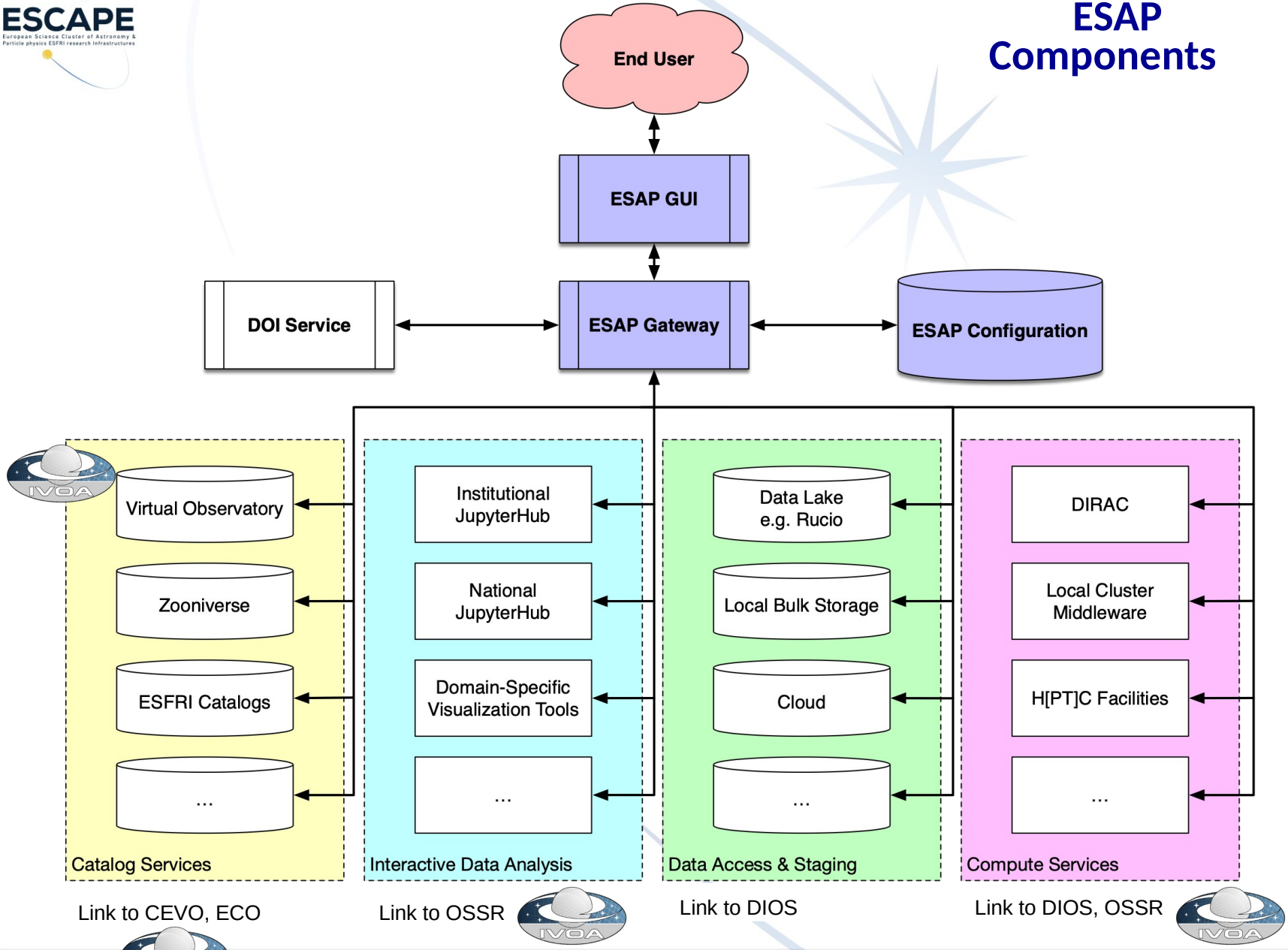
Funded by the European Union's Horizon 2020 - Grant N° 824064

Summary

- Focus on ESAP (ESFRI Science Analysis Platform)
  - components
  - features
- Where IVOA is in ESAP
  and
- where IVOA could be in ESAP
- Suggestions for discussion

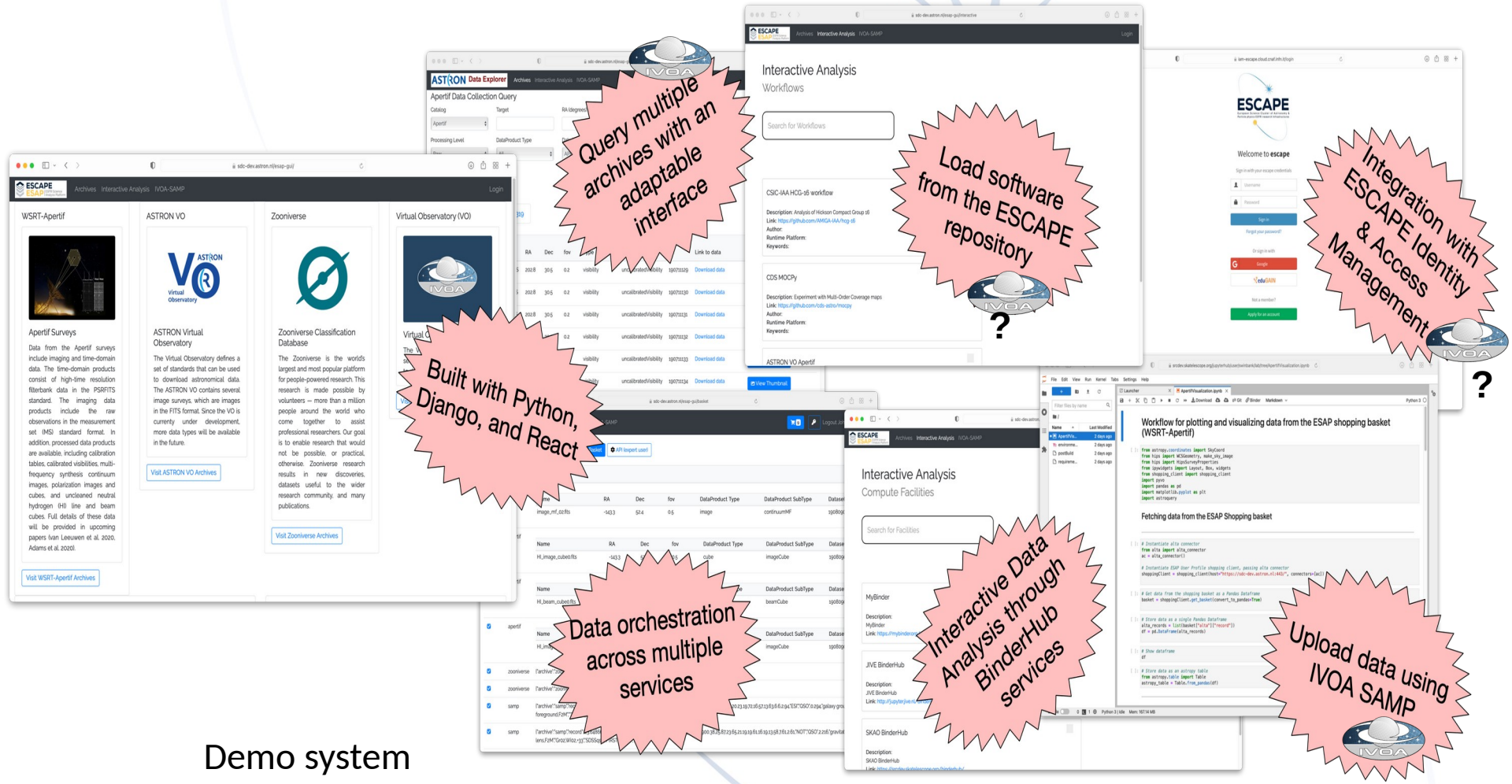
Funded by the European Union's
Horizon 2020 - Grant N° 824064

ESAP Components



Funded by the European Union's Horizon 2020 - Grant N° 824064

Current Status & Capabilities



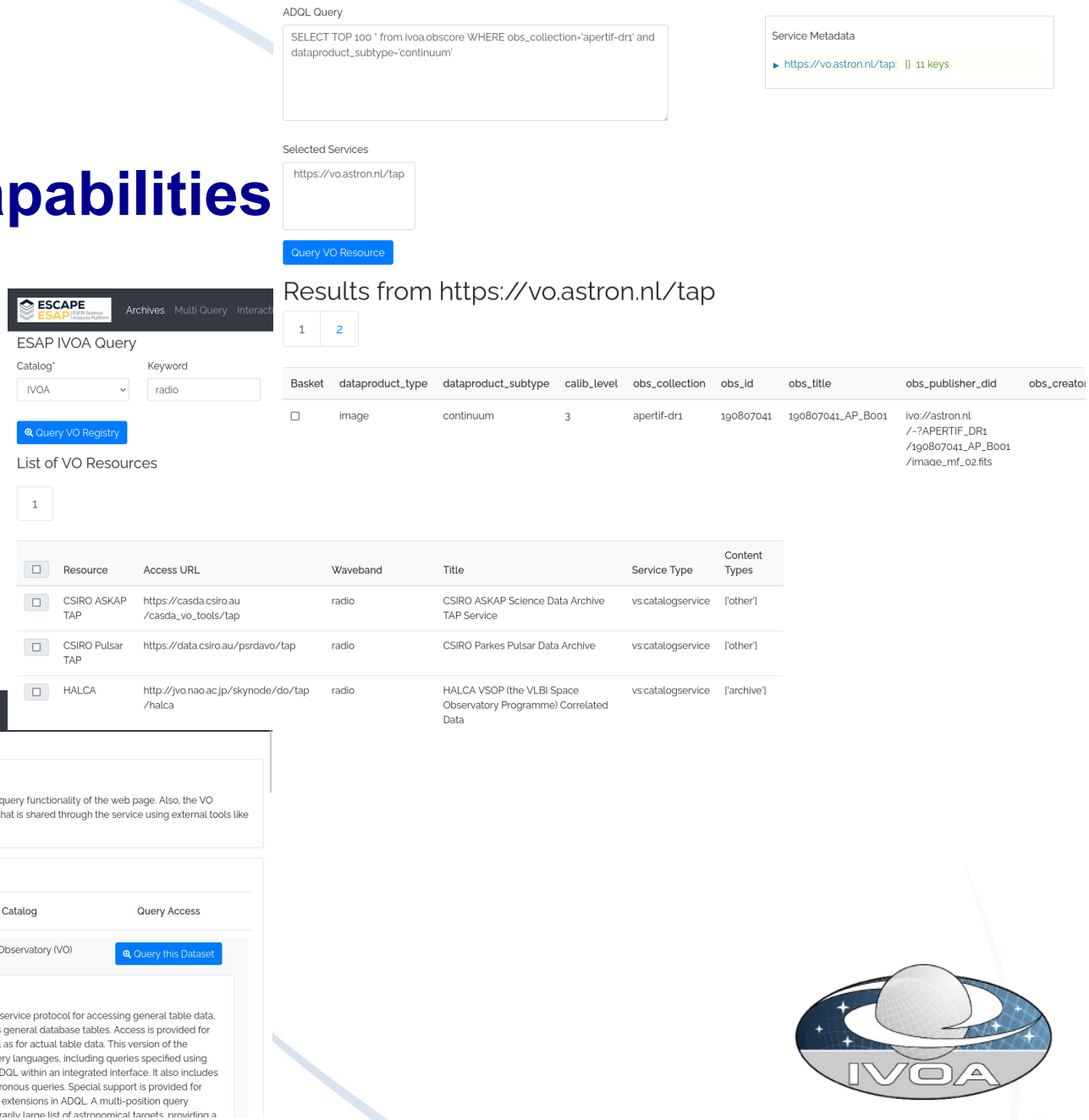
Demo system

https://sdc-dev.astron.nl/esap-gui

Funded by the European Union's Horizon 2020 - Grant N° 824064

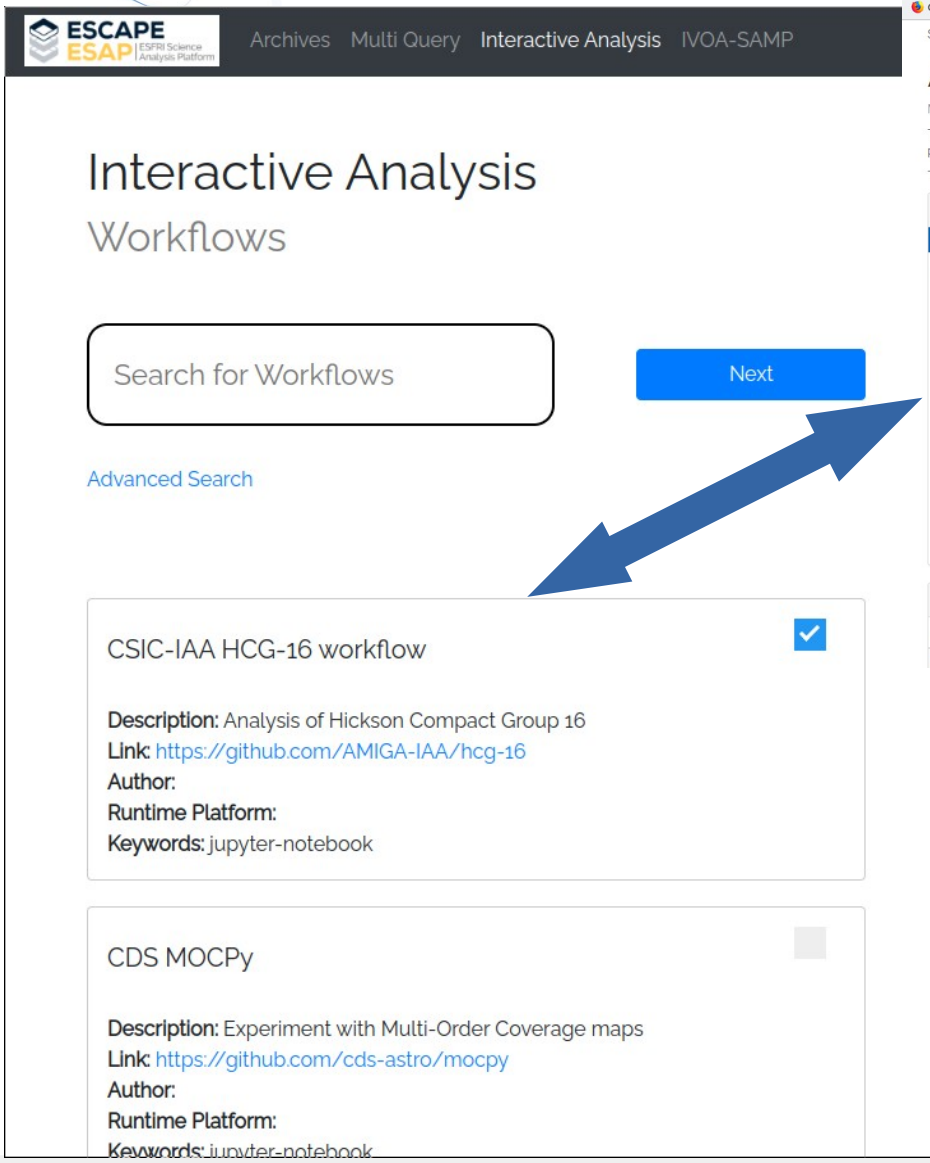
Existing Query Capabilities

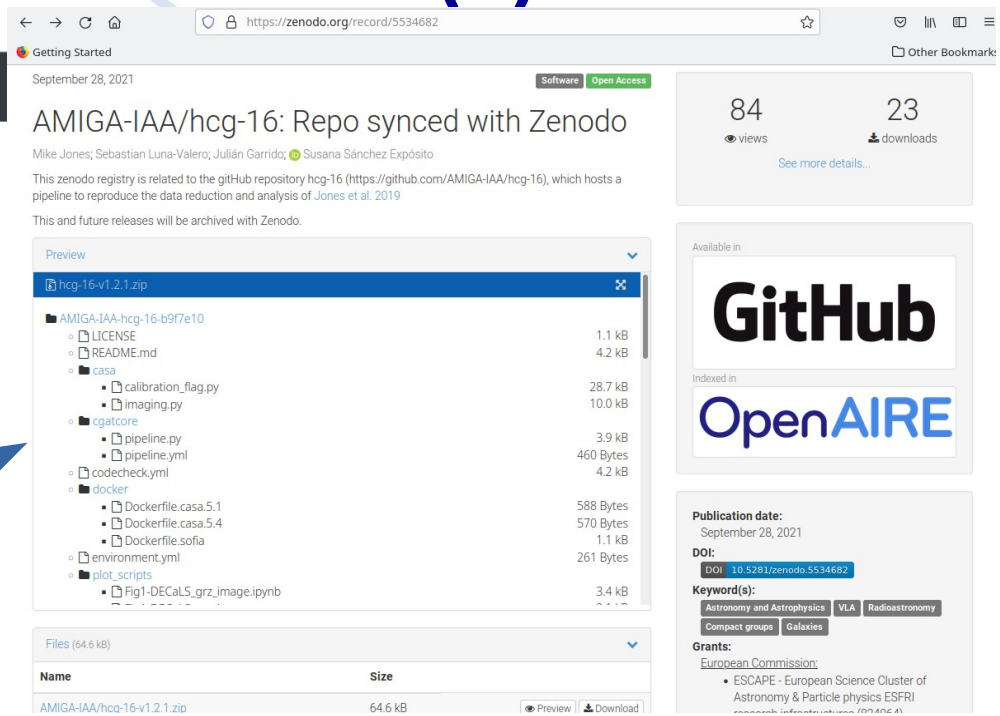
ESAP can query the registry for services and data



Funded by the European Union's Horizon 2020 - Grant N° 824064

ESAP Software Execution (1)



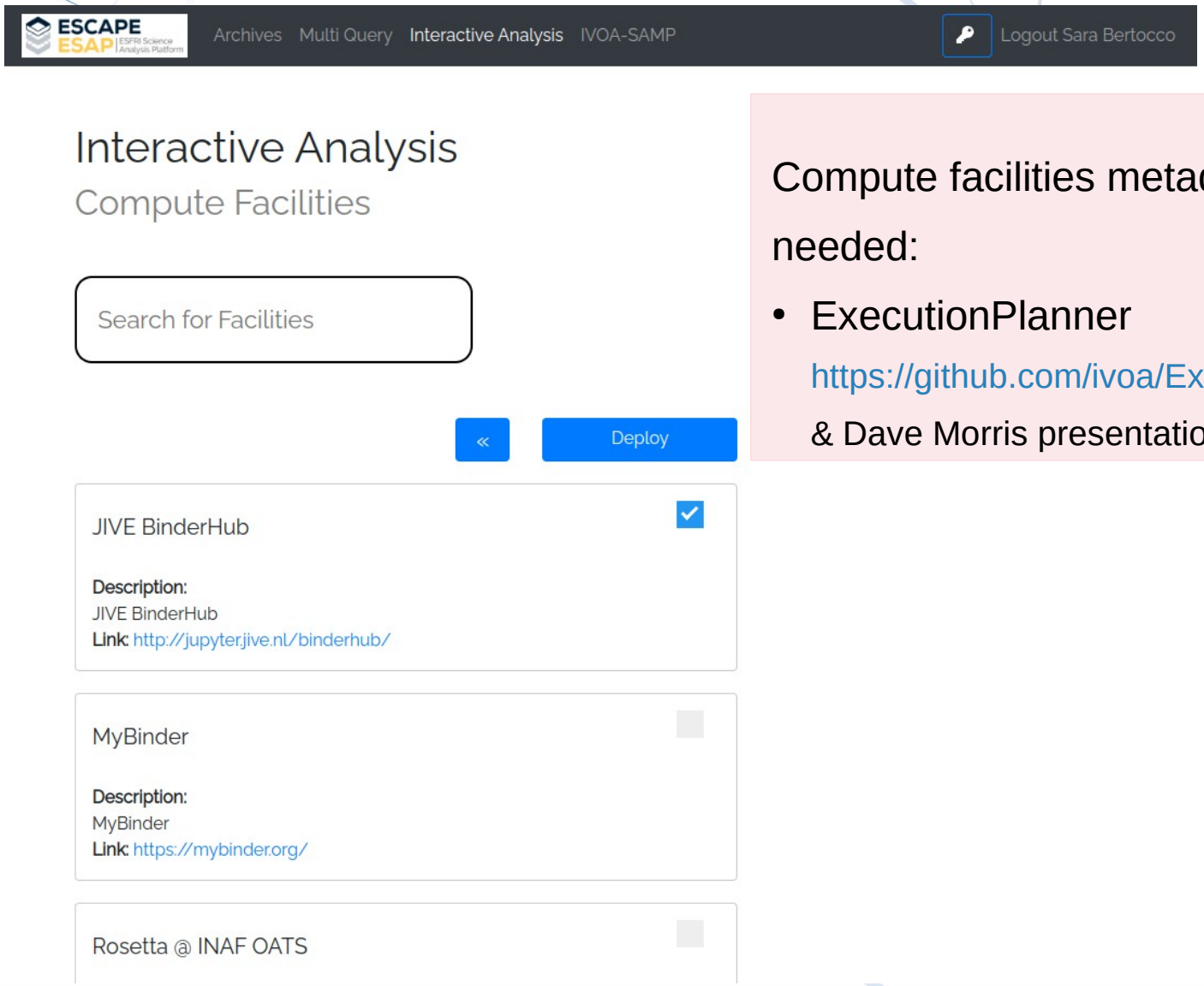


Workflow metadata description needed:

- OSSR & CEVO working on it
  codemeta.json
  https://escape2020.pages.in2p3.fr/wp3/eossr/v0.4/metadata.html

Funded by the European Union's Horizon 2020 - Grant N° 824064

ESAP Software Execution (2)



Compute facilities metadata description needed:

- ExecutionPlanner
  https://github.com/ivoa/ExecutionPlannerNote
  & Dave Morris presentation

Funded by the European Union's Horizon 2020 - Grant N° 824064

Asynchonous Tasks in ESAP

- Since now ESAP only performed synchronous queries
- VO defines a number of asynchronous protocols (e.g. async TAP) for longer running queries, wish to implement these in the ESAP Query App
- Store references to large results (1000+ rows) in Shopping basket
- Starting (and keeping track of) Batch computation jobs (e.g. simulations)
- Orchestrating Interactive Analysis platforms and data staging (e.g. jupyter hubs)

Funded by the European Union's
Horizon 2020 - Grant N° 824064

IVOA-UWS

- Provides a specification for a REST API
- Data model and control flow for Jobs
- Existing clients
- Don’t reinvent the wheel for ESAP

Universal Worker Service Pattern
Version 1.1

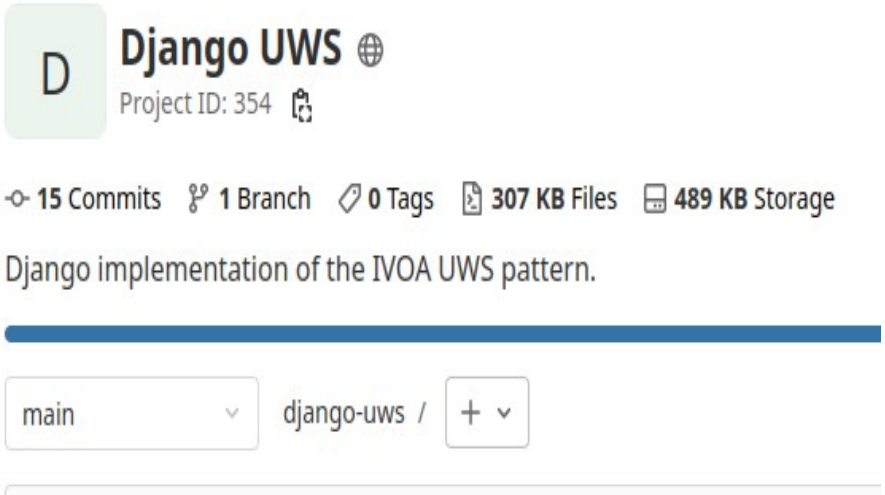
IVOA Recommendation 24 October 2016

Interest/Working Group:
http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices
Author(s):
P. A. Harrison, G. Rixon
Editor(s):
P. A. Harrison
DOI:
10.5479/ADS/bib/2016ivoa.spec.1024H

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

IVOA-UWS in ESAP

- Portable Django App (integrates easily in existing Django Projects, such as ESAP)
- Uses Django REST Framework for serialization and REST interface
- Alpha version: not fully compliant with the specification; not on PiPy yet
- Uses JSON instead of XML



https://git.astron.nl/astron-sdc/django-uws

Funded by the European Union's Horizon 2020 - Grant N° 824064

UWS Worker & Job Description Language

- Initial Job Description Language single parameter: type:

  Query.tap

  Orc.jupyter
- Job implementations should specify more parameters

```python
def init_workers() -> Dict[str, Worker]:
    tmp = {}
    for worker_name in getattr(settings, "ESAP_WORKERS", []):
        worker: Worker = import_string(worker_name)()
        tmp[worker.type] = worker
    return tmp


REGISTERED_WORKERS = init_workers()
```

```python
        try:
            worker_type = parameters["type"]
            worker = REGISTERED_WORKERS[worker_type]
            worker.run(job)

            job.phase = EXECUTION_PHASES.COMPLETED
        except Exception as err:
            # TODO: set error object
            job.phase = EXECUTION_PHASES.ERROR
```


Funded by the European Union's Horizon 2020 - Grant N° 824064

Future of UWS in ESAP

- Implement various kinds of Jobs in the workers and formalise the Job Description Language (JDL)
- Implement all parts of the REST API in the Django package
- Implement XML serialization for compatibility with existing clients
- Worker facing API replacing the Database connection and Django ORM

Suggestions:

- Extend the specification with a JSON data schema for easier integration with modern Javascript Frameworks (React, Angular etc.)
- Endpoint for registered jobs should be able to specify the parameters they allow

Funded by the European Union's Horizon 2020 - Grant N° 824064

Conclusion

- How IVOA standards “helped” ESAP features development
- How IVOA could help science platform development with new standards
- Enforce ExecutionPlanner development and standardization
- Suggest to introduce JSON in messages exchange in IVOA standards

Funded by the European Union’s
Horizon 2020 - Grant N° 824064

ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

Thank you !

Funded by the European Union's
Horizon 2020 - Grant N° 824064

Backup Slides

Funded by the European Union's
Horizon 2020 - Grant N° 824064

ESCAPE EU Project

European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures

- **Data Lake (DIOS: Data Infrastructure for Open Science):**

Build a scalable, federated, data infrastructure for open science for the ESFRI
Enable connection to compute and storage resources.

- **Software Repository (OSSR: Open-source Scientific Software and Service Repository):**

Open source scientific software and service repository.

- **Virtual Observatory (CEVO: Connecting ESFRI projects to EOSC through VO framework):**

Extend FAIR standards, methods, tools of the Virtual Observatory; demonstrate EOSC ability to include existing platforms

- **Science Platforms (ESAP: ESFRI Science Analysis Platform):**

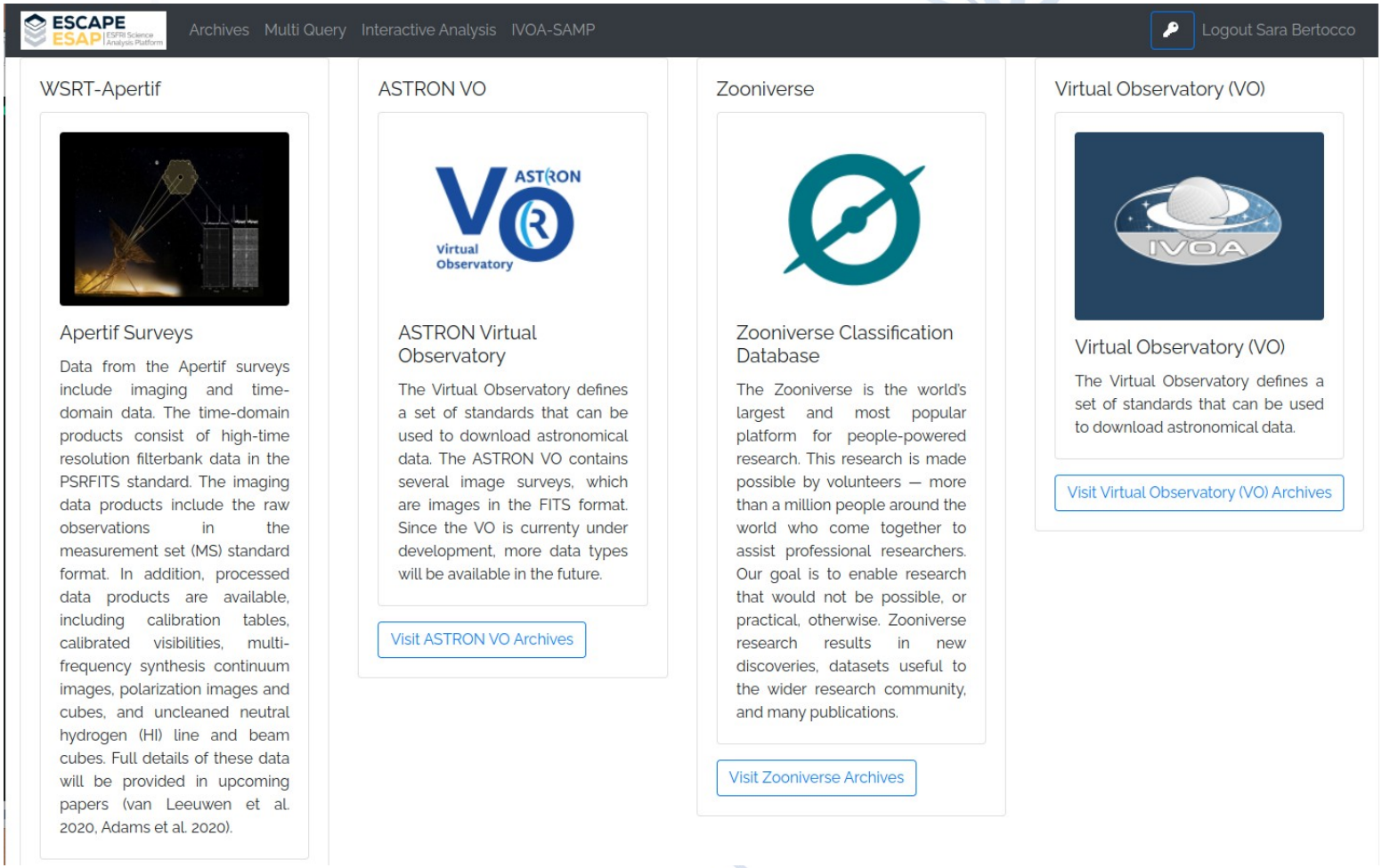
Flexible science platforms to enable the analysis of open access data

- **Citizen Science (ECO:Engagement and COmmunication):**

Open gateway for citizen science on ESCAPE data archives and ESFRI CS projects

Funded by the European Union's Horizon 2020 - Grant N° 824064

ESAP features

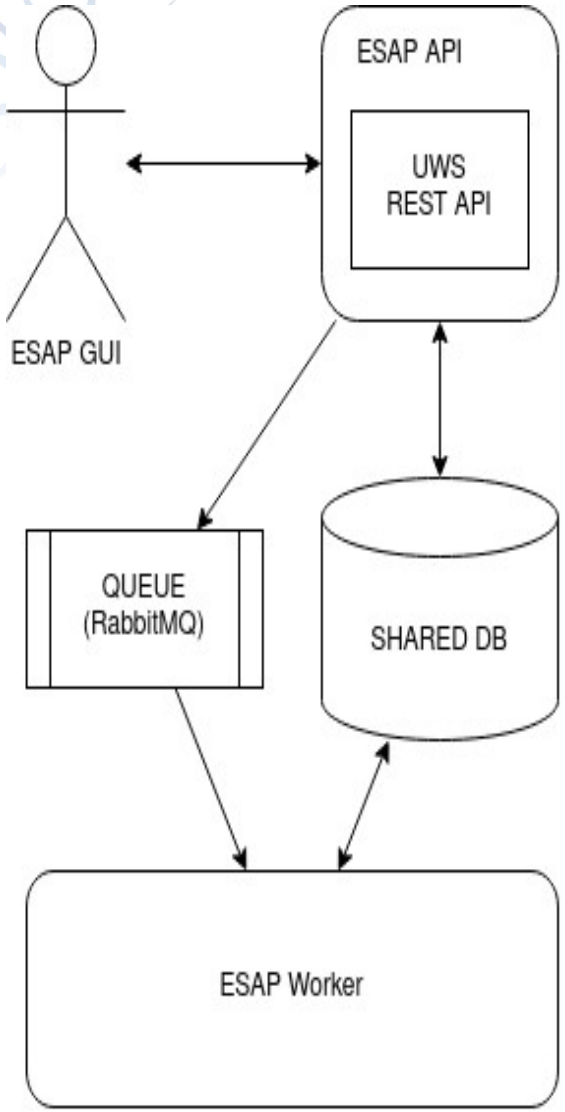


Funded by the European Union's Horizon 2020 - Grant N° 824064

Asynchronous ESAP - Architecture

- REST API
  - **UWS specification**
  - Build as Portable Django Application
- Celery Queueing system for message brokering to Worker
  - Support for various queue software:

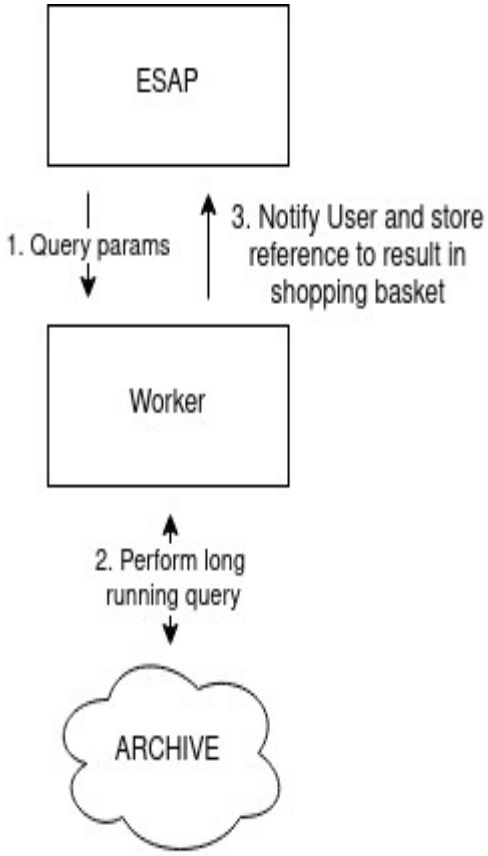
    RabbitMQ, Redis, AmazonSQS, etc.
- Celery Worker for performing tasks



Funded by the European Union's
Horizon 2020 - Grant N° 824064

Async Query Workflow

- User specifies query parameters in the ESAP GUI
- Parameters are send to the async Worker
- Worker performs the Query
- Worker stores reference to results in the shopping basket
- Worker in addition can notify the user that the query is done



Orchestration Workflow: BinderHub example

- Current focus on BinderHub: git repository -> JupyterLab interface
- User selects data in shopping basket, software workflow, target system
- Job started:
  - Clone the repository
  - Create a “postBuild” script to download data from Shopping Basket
  - Makes the repository available to BinderHub service (needs to be public)
  - Performs request to BinderHub to start a server and wait for it to be ready
  - Reports back with the URL so ESAP GUI can redirect the user
  - Cleans up the modified repository.