matchmaker

Mining astronomical catalogues in Python

Dany Vohl

E-OSSR Onboarding Presentation

2023/06/15

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.

Introduction

ESCAPE — Extreme Universe science case

- Test science project on **compact objects**

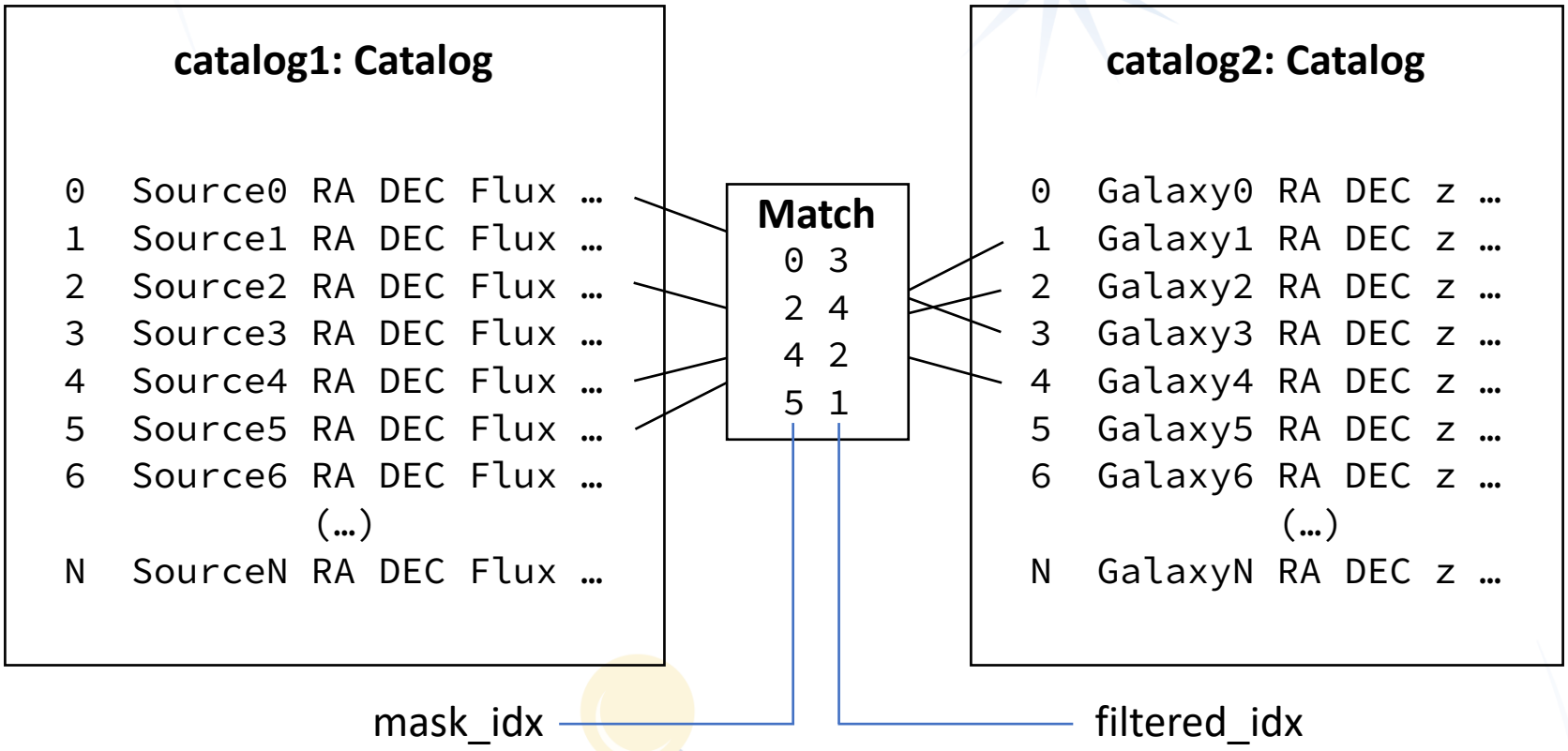
matchmaker

- **Purpose**: Intuitive and self-contained data mining of astronomical catalogues in Python

- **Use case**: Search for *Persistent Radio Sources* associated to *Fast Radio Bursts* in large radio survey

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

matchmaker



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

matchmaker

Cross-matching

catalog1: Catalog

df: pandas.DataFrame

0	Source0	RA	DEC	Flux	…
1	Source1	RA	DEC	Flux	…
2	Source2	RA	DEC	Flux	…
3	Source3	RA	DEC	Flux	…
4	Source4	RA	DEC	Flux	…
5	Source5	RA	DEC	Flux	…
6	Source6	RA	DEC	Flux	…
		(…)			
N	SourceN	RA	DEC	Flux	…

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

matchmaker

Cross-matching

catalog1: Catalog

df: pandas.DataFrame

0	Source0	RA	DEC	Flux	…
1	Source1	RA	DEC	Flux	…
2	Source2	RA	DEC	Flux	…
3	Source3	RA	DEC	Flux	…
4	Source4	RA	DEC	Flux	…
5	Source5	RA	DEC	Flux	…
6	Source6	RA	DEC	Flux	…
		(…)			
N	SourceN	RA	DEC	Flux	…

Properties

```
cols: Columns
use_distance: bool
matches: {key:Match}
[images: {key:Image}; ...]
```

```
label: str
unit: astropy.unit
description: str
```

functions

```
load_data(…)
as_SkyCoord(…)
[...]
```


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

matchmaker

Cross-matching

matchmaker.dataset.__init__

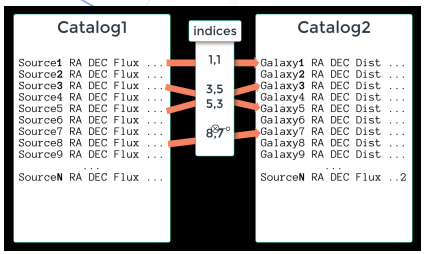
```
class Catalog:
    df = None
    name = None

    def __init__(self, ra=None, dec=None, use_distance=False):
        self.cols = Columns(ra=ra, dec=dec)
        self.use_distance = use_distance
        self.matches = {}

    def add_match(self, catalog_name, from_to=0, ...

    def get_mask_to_third_catalog(self, cat1, cat2...

    def load_data(self):
        pass

    def filter_box(self, boxes):...

    def as_SkyCoord(self, mask=None, distance=None...

    def prop(self, prop, default=None):...

    def prop_to_unit(self, prop, unit, mask=None, ...
```

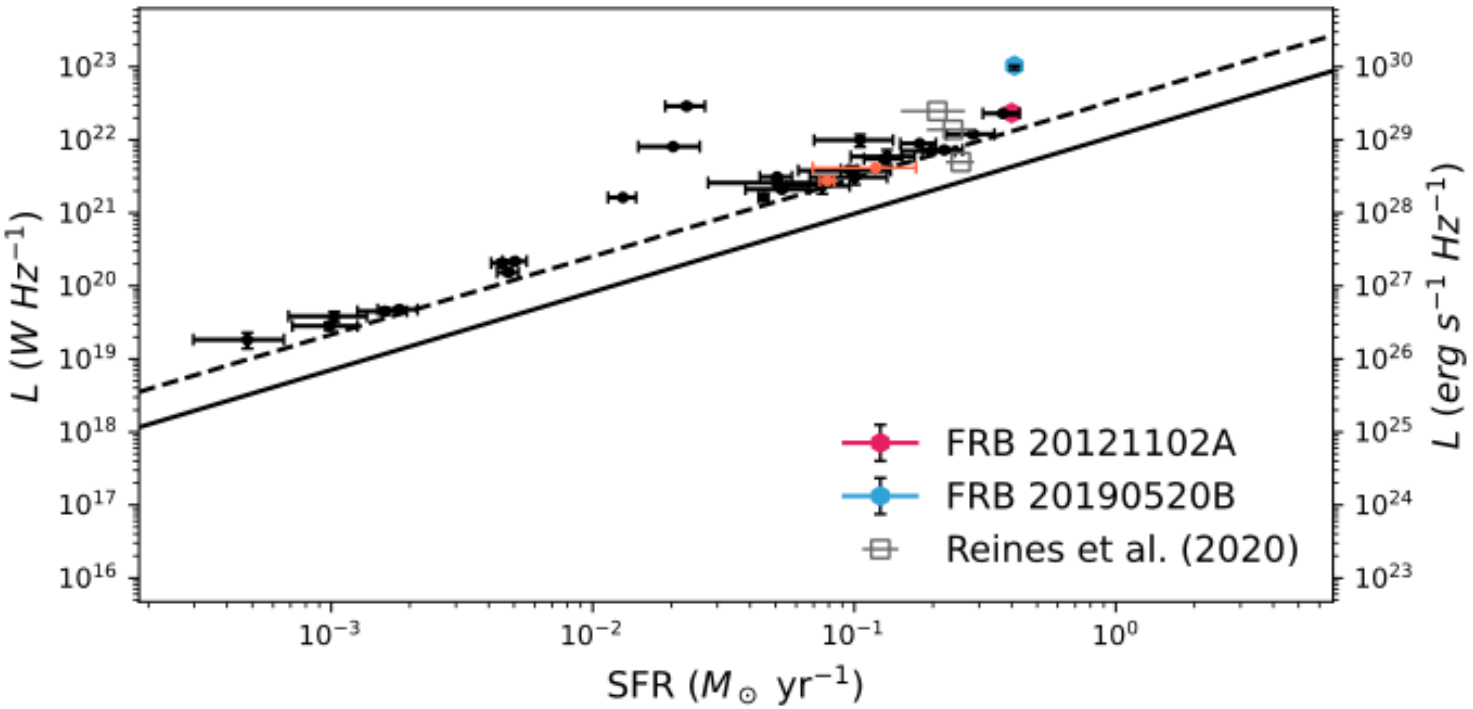
matchmaker.dataset.lotss

```
import numpy as np
import astropy.units as u
import pandas as pd

from . import (Catalog, Column, DATA_BASE_PATH)
from ..utils import load_fits_as_dataframe, load, get_matched
from ..measures import distance_atomic, luminosity_distance as _luminosity_distance, powerlaw_scale
from ..models.lofar import power_law, sigma as lum_sfr_sigma

class Lotss(Catalog):
    file_location = DATA_BASE_PATH + 'lotss/LoTSS_DR2_v110_masked.srl.fits'
    name = 'lotss'
    boxes = None

    field_file_location = DATA_BASE_PATH + 'lotss/obslist.csv'

    def __init__(self, load_data=True, constrain=False, smin=None, compact_only=False, single_only=False):...

    def load_data(self, constrain=False, smin=None, compact_only=False, single_only=False):...

    def semi_major(self, mask=None, with_unit=False, to_unit=None):...

    def semi_minor(self, mask=None, with_unit=False, to_unit=None):...

    def set_distance(self, source:Catalog):...

    def luminosity_distance(self, mask=None, output_units='W_Hz', with_unit=False, with_error=False, log=False):...

    def radio_loudness(self, obj1, mask_lotss=None, mask1=None, alphas=None, verbose=False):...

    def distance_to_lum_sfr_relation(self, obj1, mask_lotss=None, mask1=None):...

    def select_lum_sfr_outliers(self, obj1, mask1=None, mask2=None, n_sigma=1.5, filter_on_sfr=None, store=True, verbos
```

matchmaker

Aggregate / candidate selection

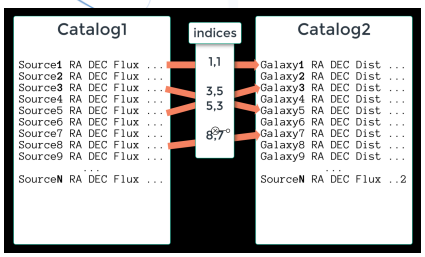


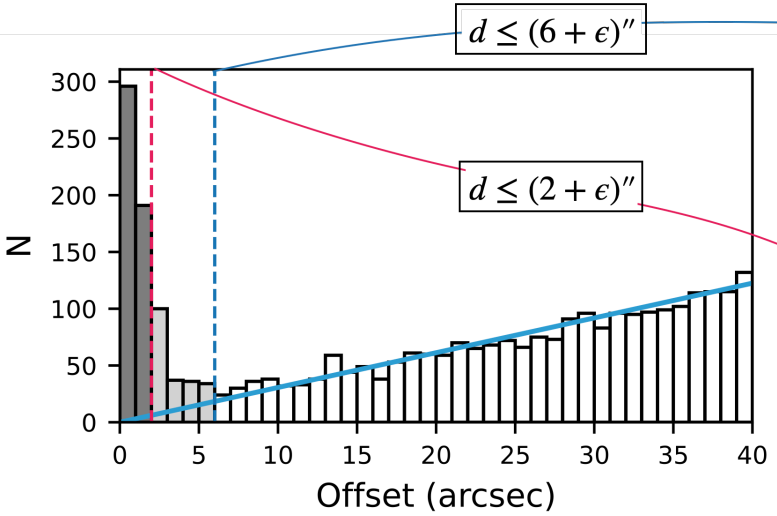


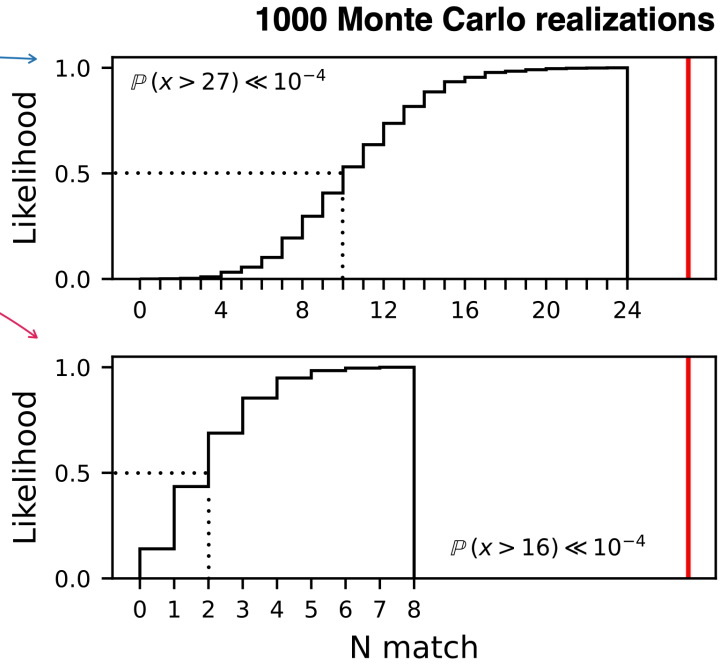
Funded by the European Union's Horizon 2020 - Grant N° 824064

matchmaker

Statistical analysis

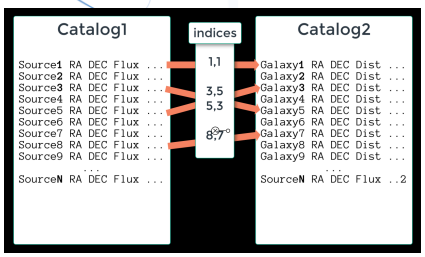


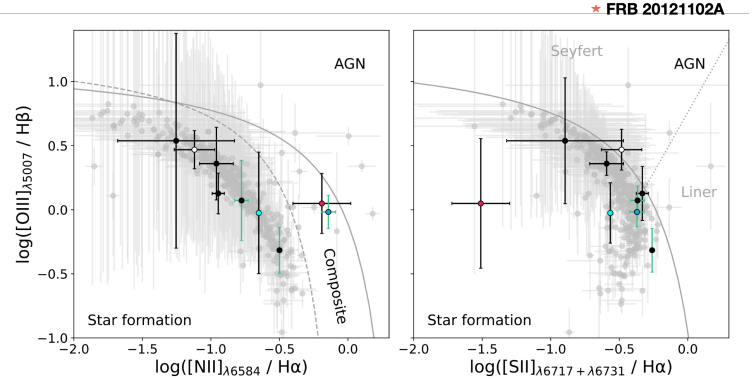


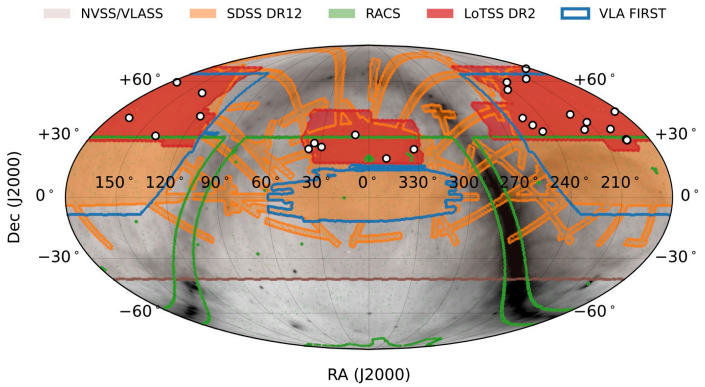


Funded by the European Union's Horizon 2020 - Grant N° 824064

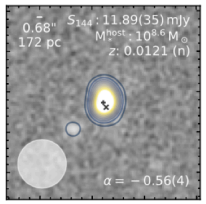
matchmaker

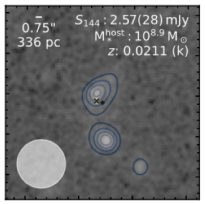


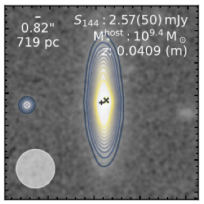


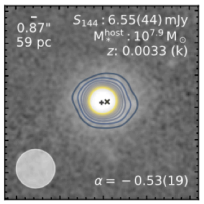


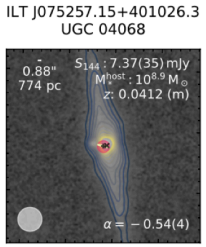
Augment via ancillary data (e.g. VO)

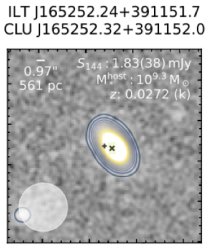


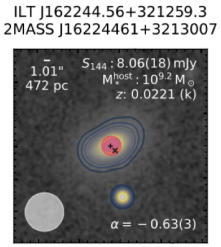


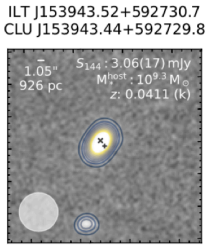


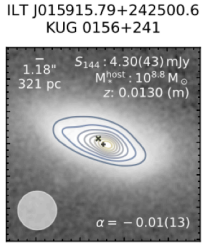


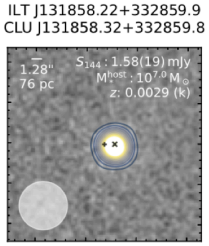


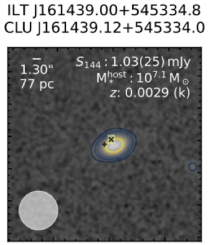


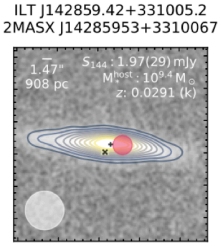


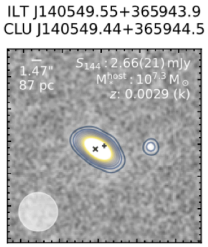


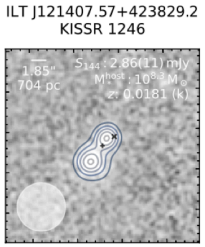


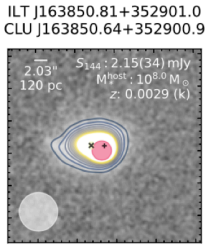


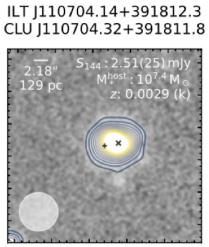












Funded by the European Union's Horizon 2020 - Grant N° 824064

Software/Service Requirements

- Development currently on escape2020 gitlab
- Pull request, release, versioning…
- CI: Travis and unit tests: In prep.
- Documentation: doc file, readthedocs in prep.
- BSD-3 license

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

Software/Service Requirements

Python 3

requirements.txt (numpy, scipy, astropy, pandas, …)

[Most of the code is standalone with minimal dependance on other packages]

Requires RAM (hold multiple datasets into memory)

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

OSSR Integration – 2 slide

- What is available: source code
- What will be onboarded: source code, example notebook

- What is the ”user story” of a EOSC user taking on the software/service?
  - From the data side (what data can be analysed and how)
    - Tabular data (csv, fits tables, …)
  - From the OSSR side (how to find data and easy use demos, tutorials, documentation, …)
    - Escape 2022 meeting, EOSC M24 review demo…

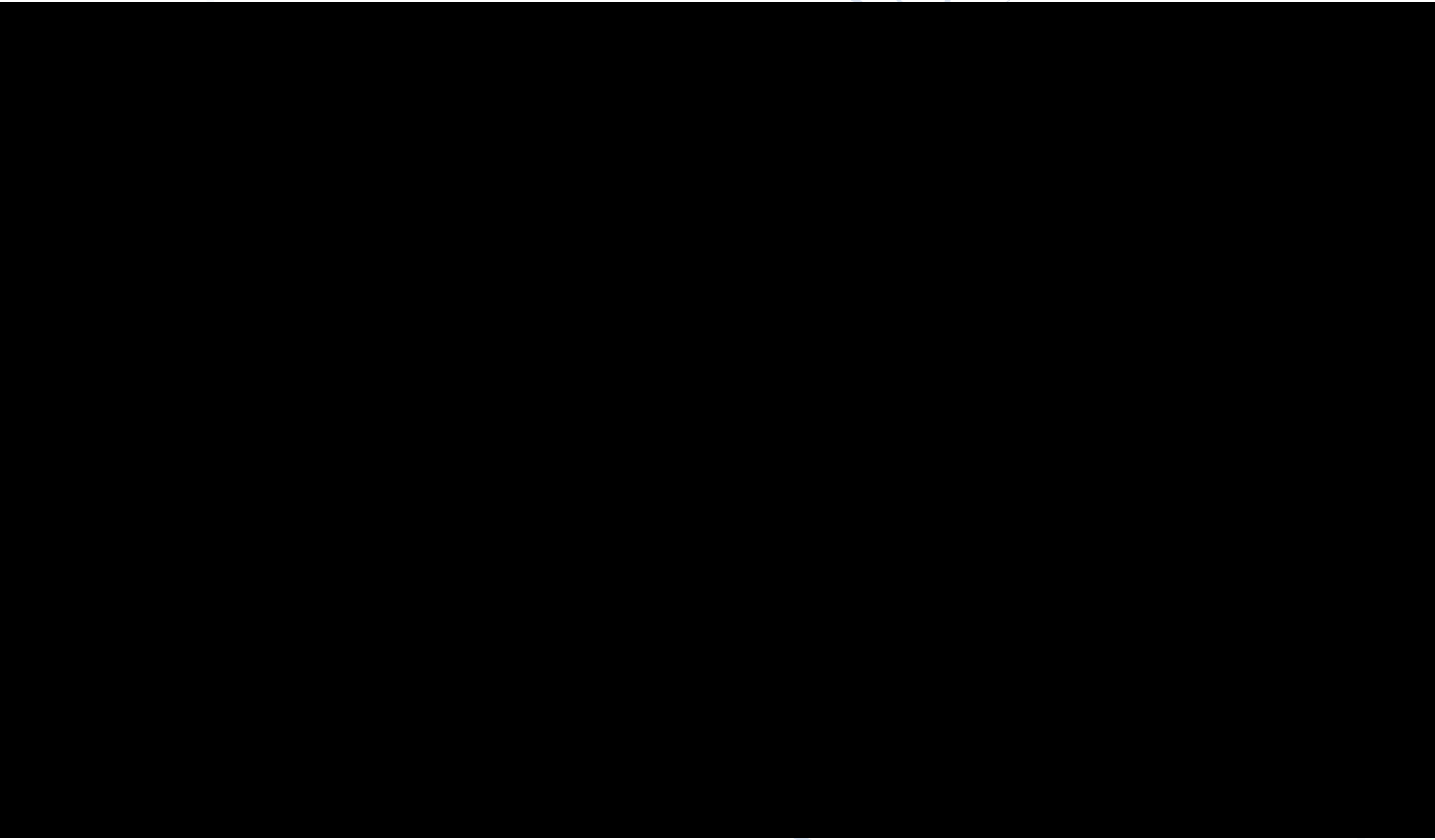
Funded by the European Union’s Horizon 2020 - Grant N° 824064

Time for a short demo (~10 min)

- Show how the software is used and what is the outcome
- What should and can a EOSC user do with the software?

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

Time for a short demo (~10 min)



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

Open Points and Discussion Time

- Which of your questions have not been covered so far?
- What do you want to discuss?

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

TOC of Tech Report

- Introduction
  - ESFRI/RI and Partner, Science Case
  - Software and Service Name
- Software/Service Development Strategy
- Software/Service Requirements
- OSSR Integration
  - Status
  - Content
  - User Story

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