# Data Access (and Provenance) in the context of CTA

## *(the Cherenkov Telescope Array)*

Mathieu Servillat, Catherine Boisson

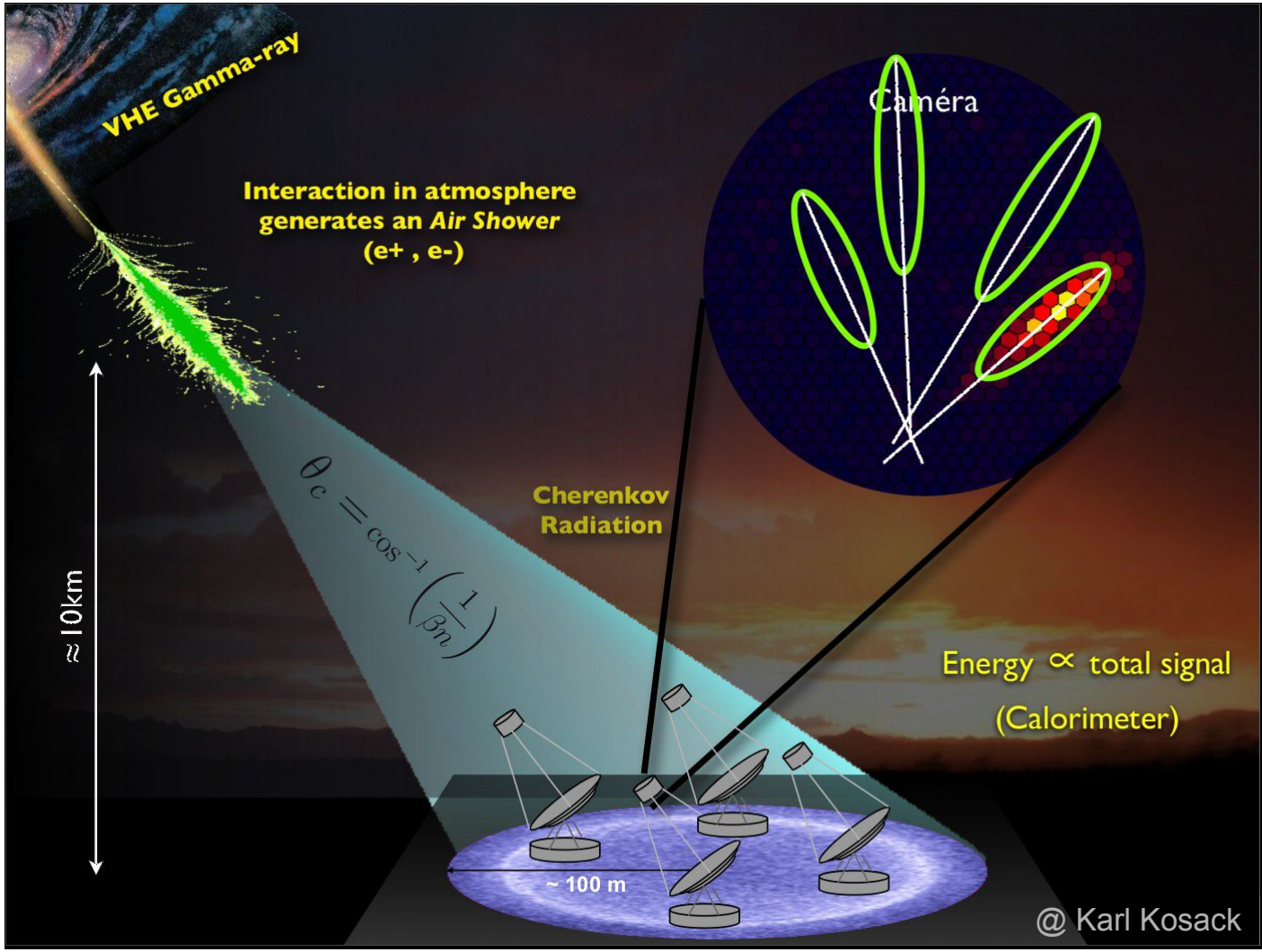
IVOA Victoria
May 2018

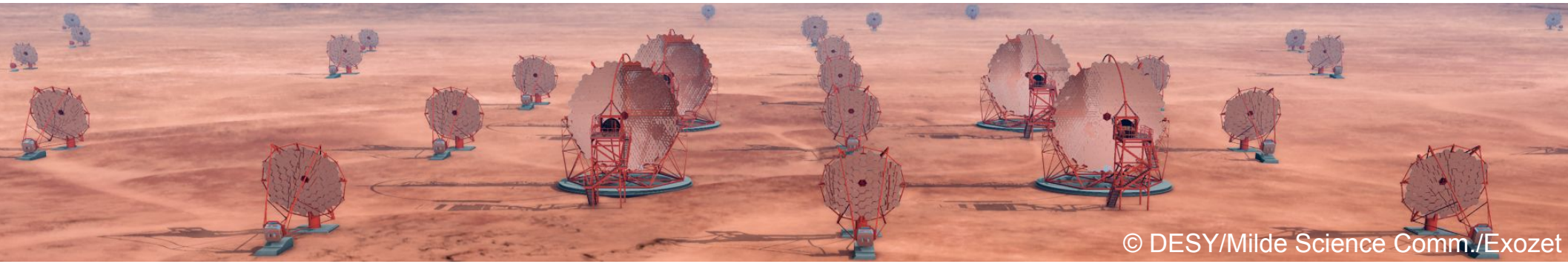





# Cherenkov Imaging

- **Dark nights** (small duty cycle)
- Field of view: 5-8 degrees
- **Event Reconstruction**: photon, particle shower, Cherenkov light (faint, few nanoseconds)
- **Atmosphere** = calorimetre Simulations, assumptions
- **Complex Metadata**, need to be structured

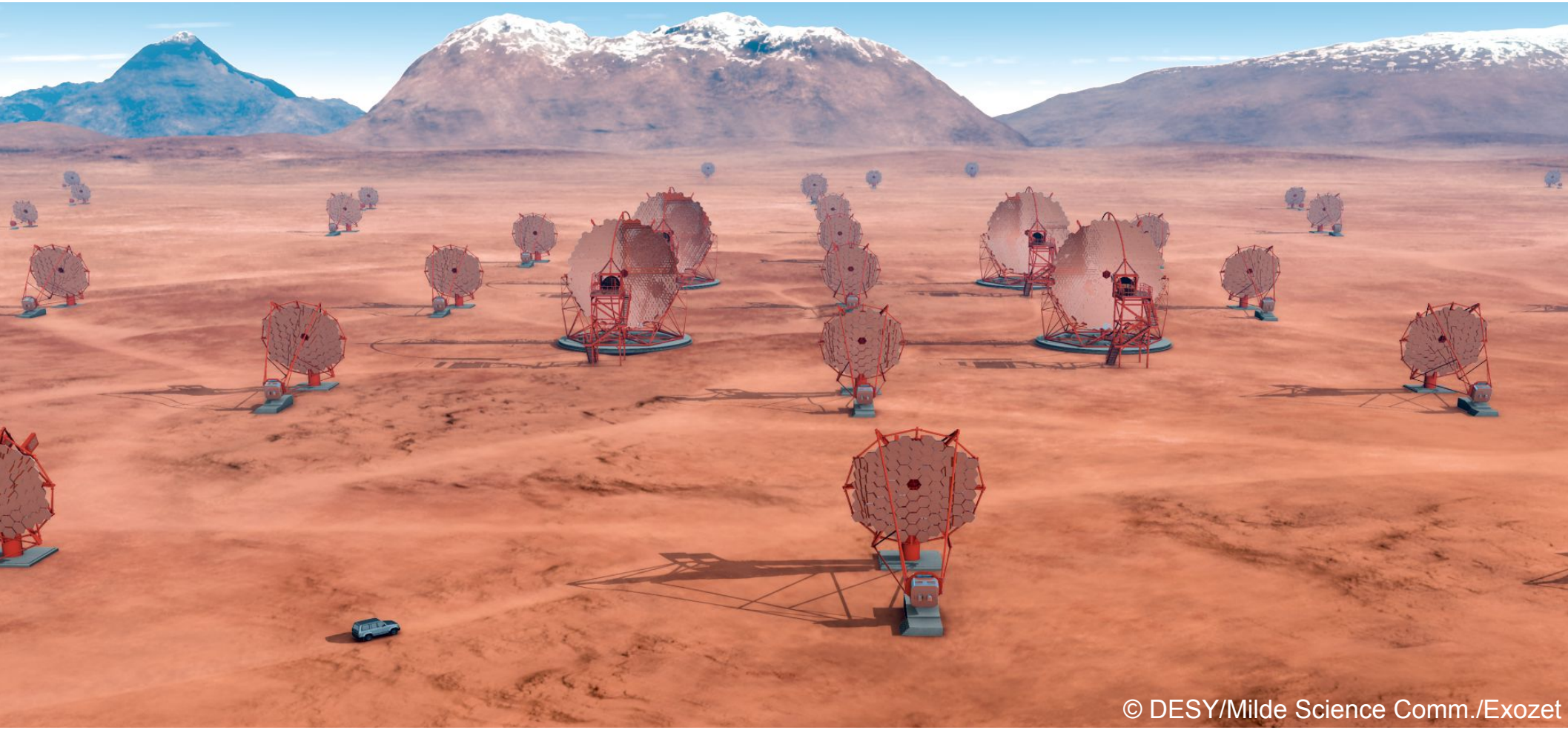


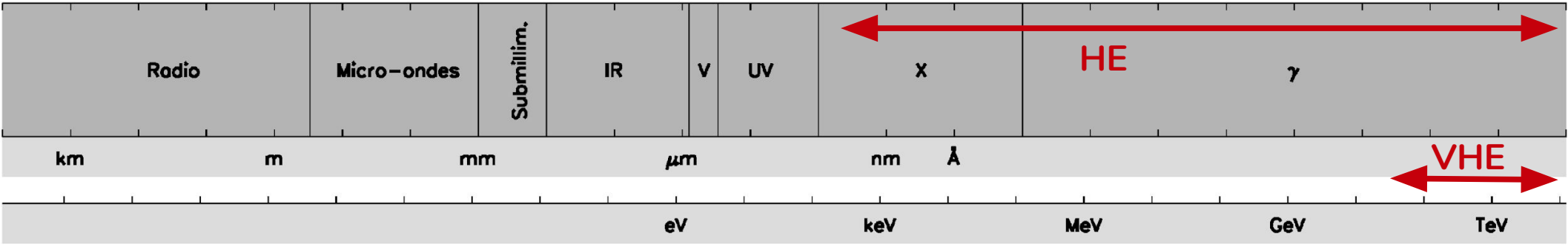
cherenkov telescope array

# Observatory

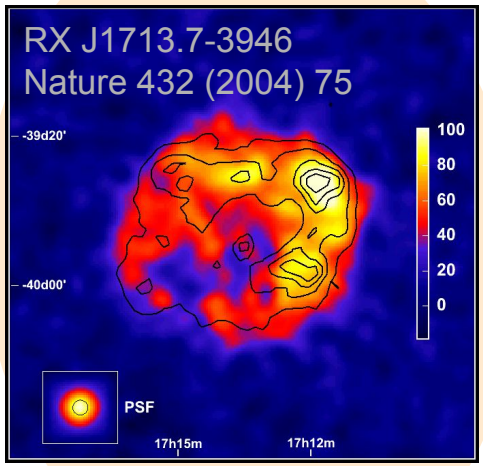
- Two arrays of **100 (South)** et **20 (North)** Cherenkov telescopes (4, 12 et 24 m in diametre)
- July 2015: site selection, Chile (ESO) and La Palma
- 2016: pre-production phase
- 2018-2013: production phase
- Observatory **open** to the Astronomy community



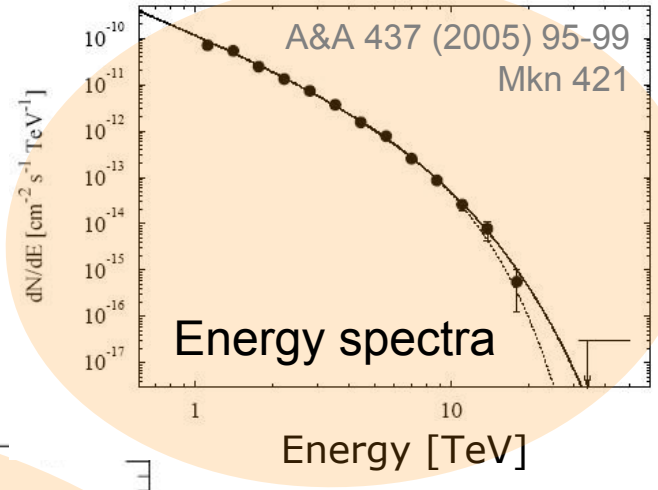
# Very high energy (VHE) data

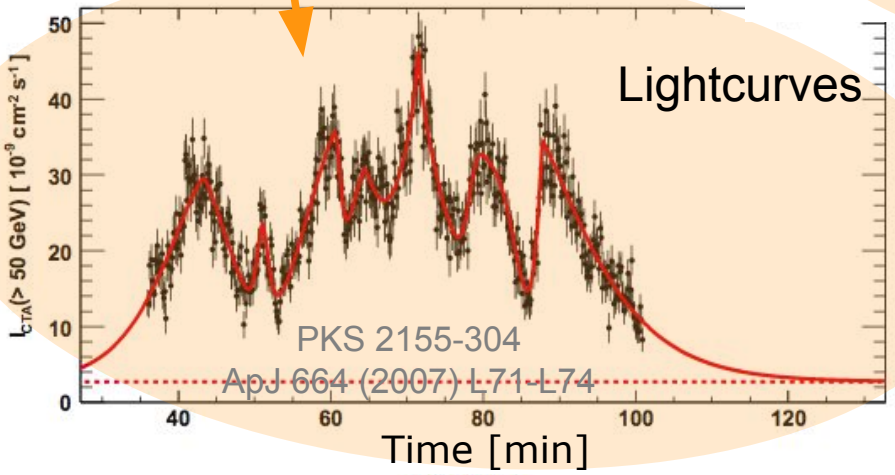



- Several orders of magnitude
- Photon **counting**
- Low count **statistics**, high background
- **Event lists**
  (coordinates, time, energy)



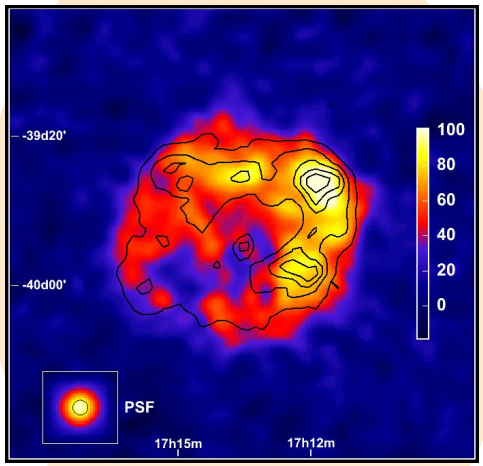

Images



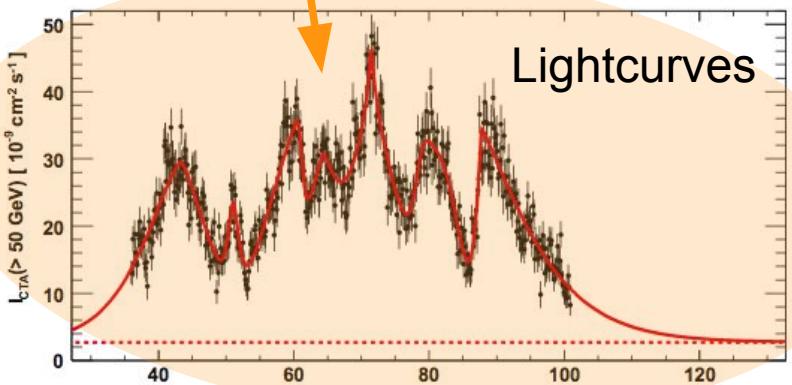



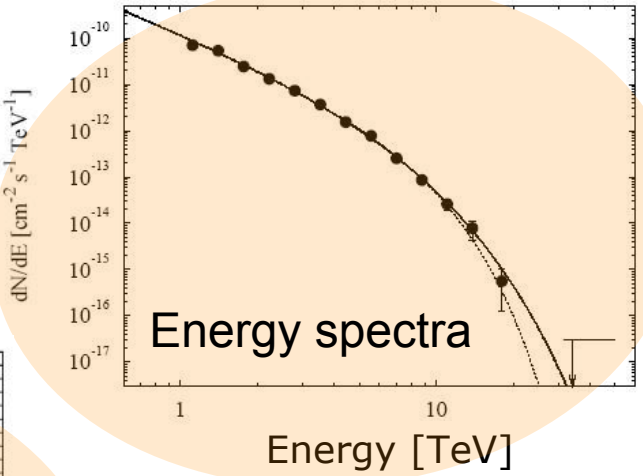
# Multi-wavelength analysis

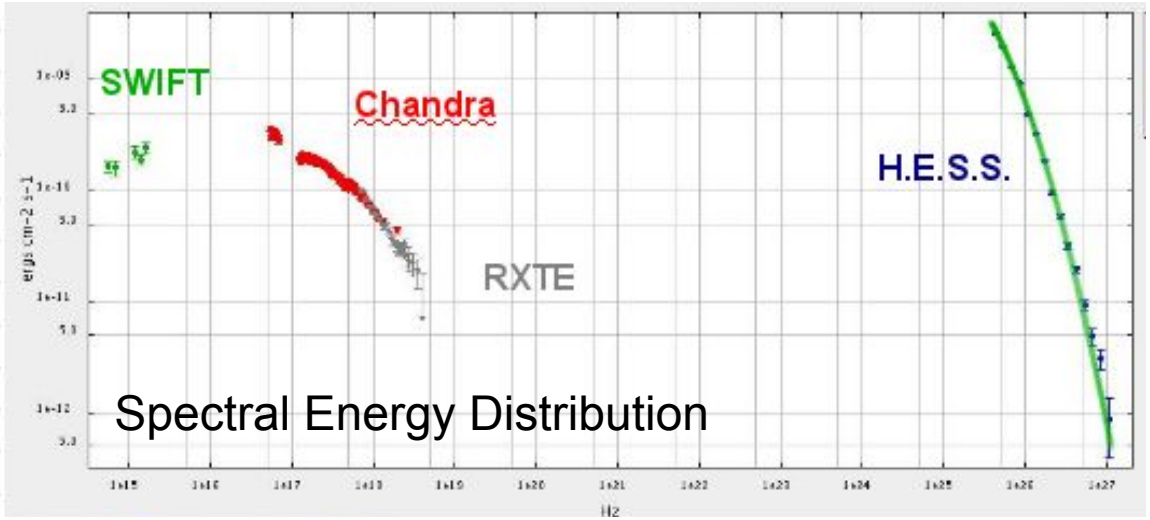
**CTA Event lists**
(coordinates, time, energy)




Images



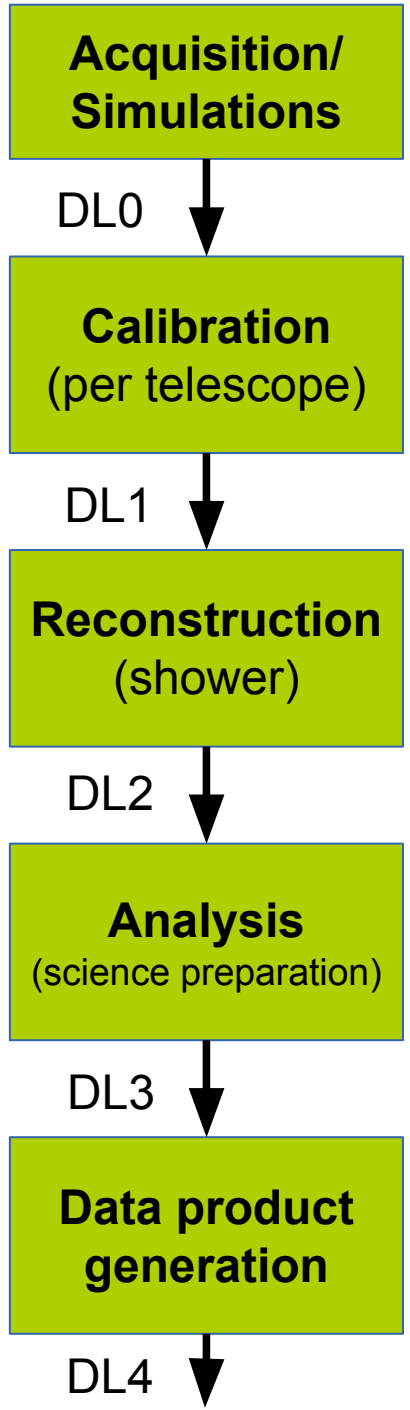

**Compatible data at other wavelength?**

**Simultaneous**
**Calibrated**
**Specific Processing?**
**Context?**

# Data levels and workflow

| Data Level | Short Name | Description | Data reduction factor |
|---|---|---|---|
| Level 0 (DL0) | DAQ-RAW | Data from the Data Acquisition hardware/software. | |
| Level 1 (DL1) | CALIBRATED | Physical quantities measured in each separate camera: photons, arrival times, etc., and per-telescope parameters derived from those quantities. | 1-0.2 |
| Level 2 (DL2) | RECONSTRUCTED | Reconstructed shower parameters (per event, no longer per-telescope) such as energy, direction, particle ID, and related signal discrimination parameters. | $10^{-1}$ |
| Level 3 (DL3) | REDUCED | Sets of selected (e.g. gamma-ray-candidate) events, along with associated instrumental response characterizations and any technical data needed for science analysis. | $10^{-2}$ |
| Level 4 (DL4) | SCIENCE | High Level binned data products like spectra, sky maps, or light curves. | $10^{-3}$ |
| Level 5 (DL5) | OBSERVATORY | Legacy observatory data, such as CTA survey sky maps or the CTA source catalog. | $10^{-5}$ - $10^{-3}$ |

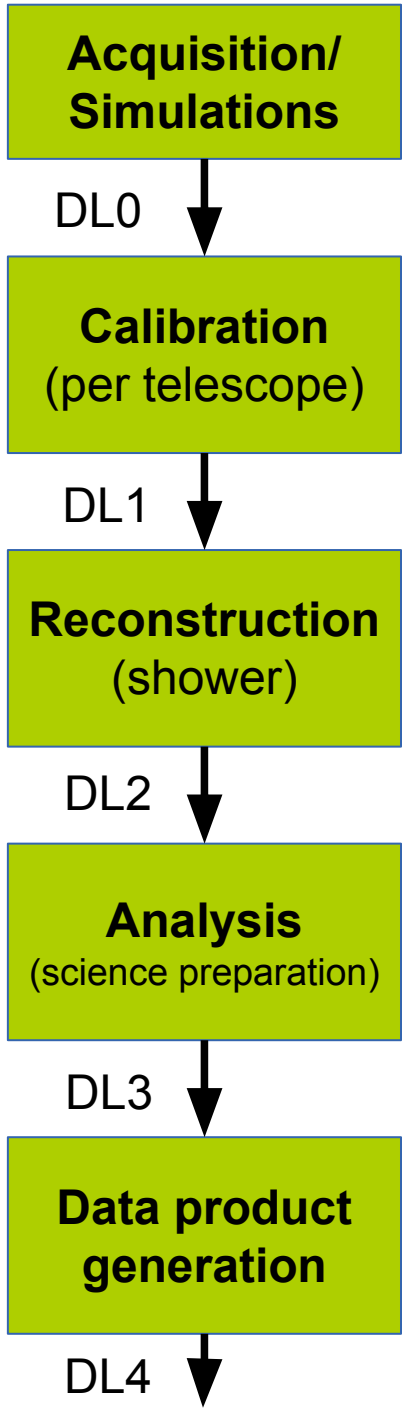
Published & Archived



# CTA Pipeline Requirements

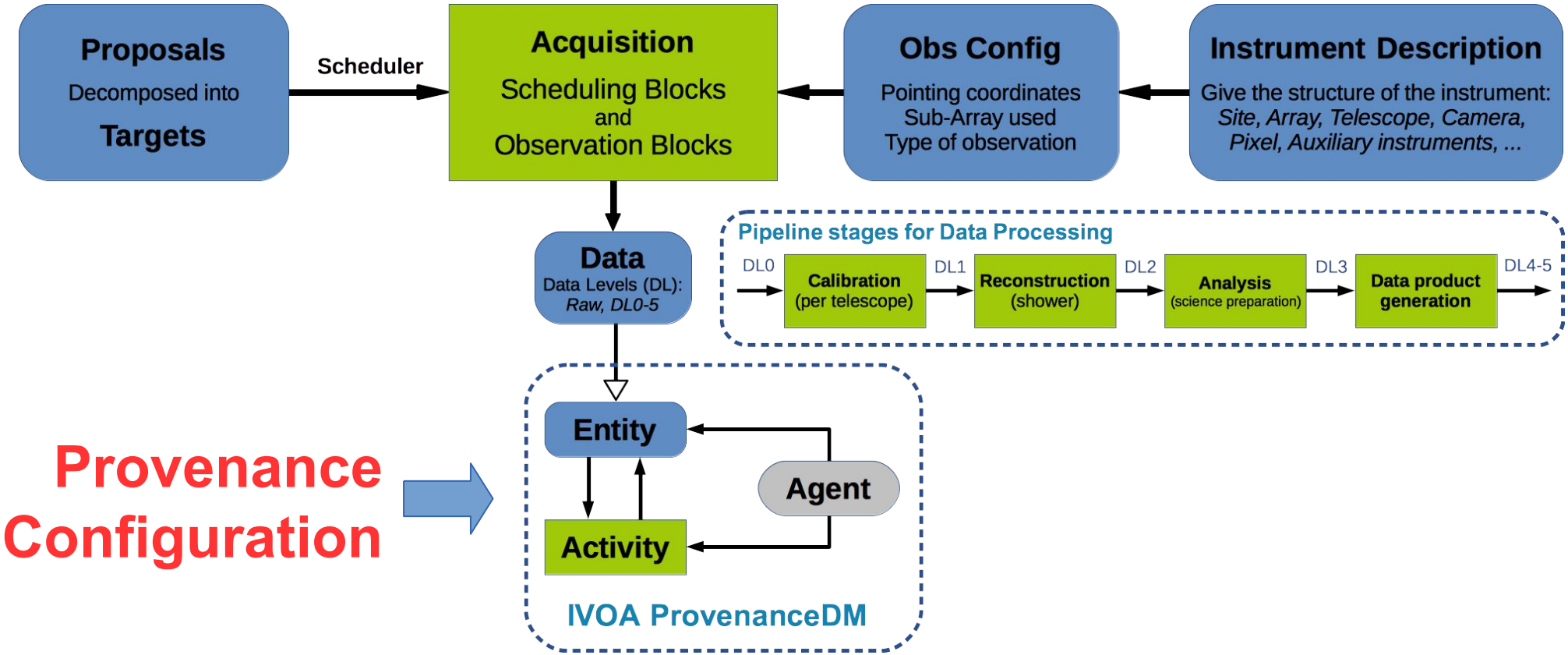
- **Open** observatory
- A-USER-0110 : must ensure that data processing is **traceable** and **reproducible**
- **Inform** user on processing steps performed
- **Link to progenitor** to regenerate data (DL3 to DL4)

- Identify how a data product was produced
  ⇒ **Provenance**
- Identify what detailed options were used
  ⇒ **Configuration**



# Master Configuration Data Model

- Defines **structure** of services, content and context of data
- Can be seen as a **global interface**



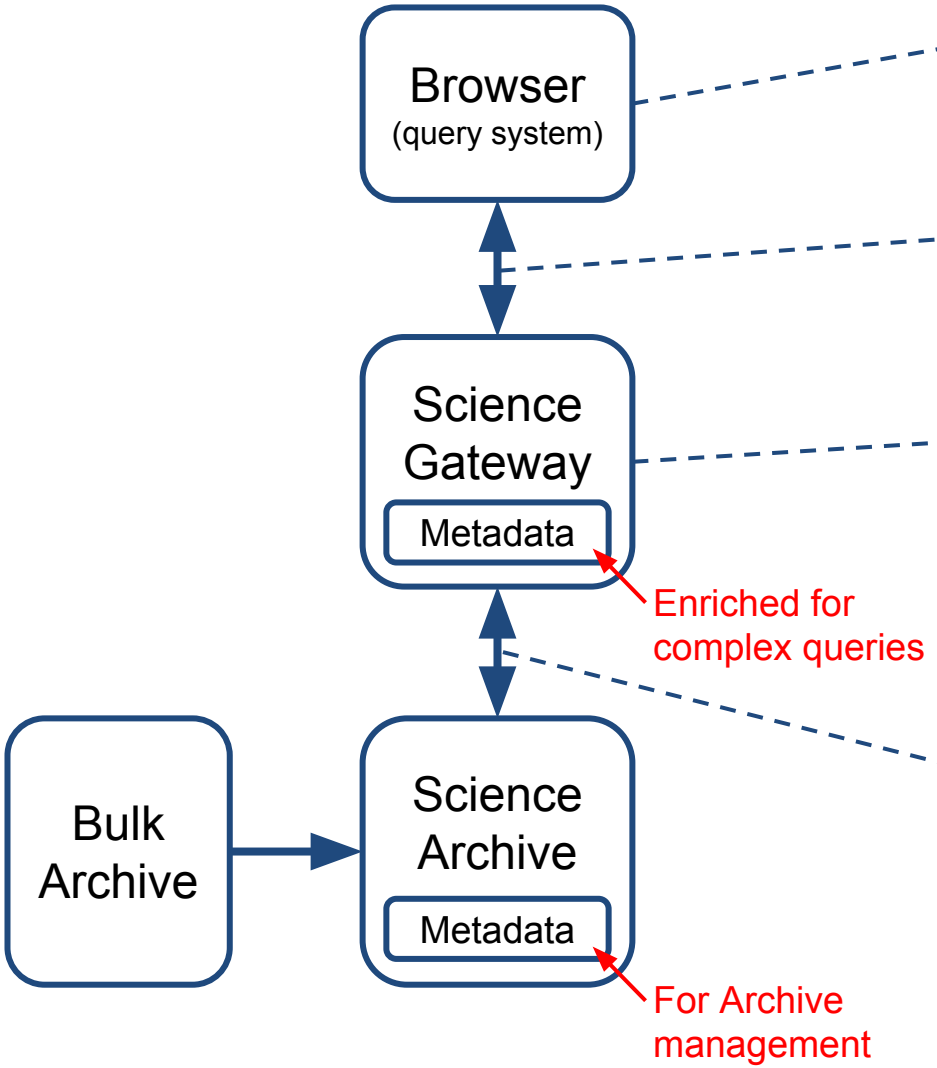
# CTA Data Access Use cases

- The **PI** of a successful proposal wants to retrieve the data
  - **Simple query** by obs_id (or PI name, or direct link sent to the PI)
  - Need user authentication and authorization
- A CTA Science User wants to find a **specific data set**
  - **Complex query**
  - Using Cone Search (RA, Dec) and/or other information (time range, spectral range, instrument configuration, nature of the target, keywords in the proposal, data processing details, …)
- A Science User wants to gather more information on a source **detected at other wavelengths**
  - No knowledge about CTA a priori
  - **Query limited to "generic" information** sent to several archives

⇒ The Virtual Observatory (VO) framework is useful for all those use cases

# Science Gateway in the VO framework

## CTA Data Access




## In the Virtual Observatory Framework

**Client:** submits a query
- **VO tools** (Topcat, Aladin, scripts…)
- Dedicated **Web Client**

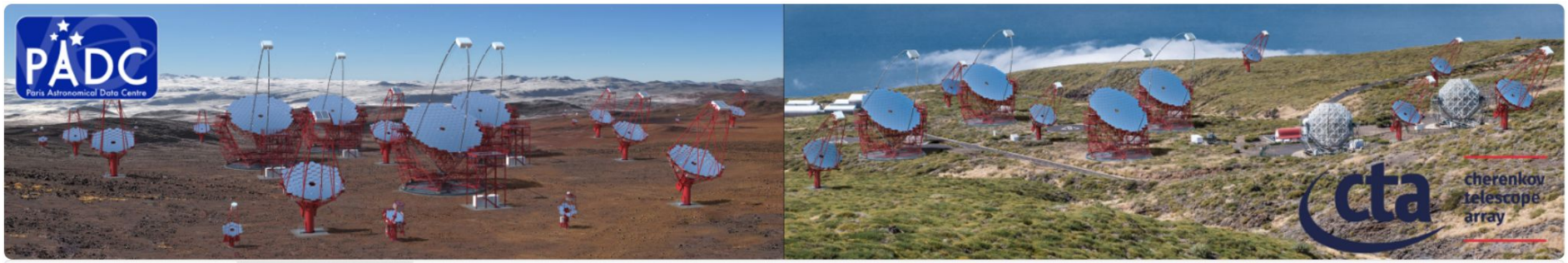
**Protocol:** standard for query exchange
- **ADQL** (Astronomical Data Query Language)
- **TAP** (Table Access Protocol)

**Server:** computes query results
- **TAP** Service
- VO **Data Models** (ObsCore, DataSet, ...)
  - RA → s_ra
  - Dec → s_dec
  - obs_id, t_min, t_max, access_url, …
- ⇒ **ObsTAP** Service

**Retrieval System:**
- VO ObsCore **access_url** + **DataLink**
- Any service at the **access_url**
  - FTP, HTTP server
  - VO Space
- e.g. https://archive.cta.org/retrieve?id=###

# CTA TAP Distiller

https://voparis-cta-test.obspm.fr




CTA Data Distiller | Search Form | Results | Sign in

**Cone Search**

| | | |
|---|---|---|
| **Target Name** | PKS 2155-304 | Used to query Simbad with Sesame and set RA/Dec. |
| **Source RA (deg)** | 329.717 | |
| **Source Dec (deg)** | -30.226 | |
| **Search radius (deg)** | 0.001 | |

Submit | Reset

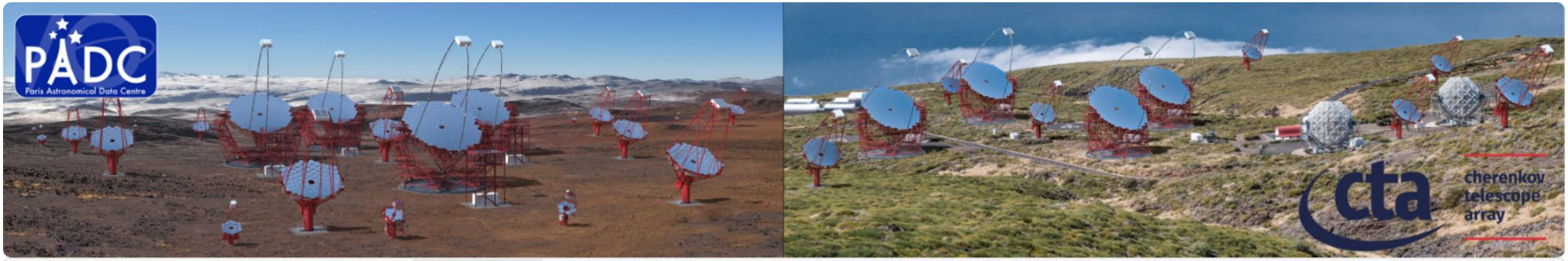
- Django, jQuery, BootStrap3
- Name resolver (Simbad through Sesame)
- Builds and Sends the ADQL query

**ObsCore Search**

| | | |
|---|---|---|
| **proposal_id** | | Proposal ID |
| **dataproduct_type** | Nothing selected | Data product (file content) primary type |
| **dataproduct_level** | Nothing selected | DL0-5 |

# CTA TAP Distiller

https://voparis-cta-test.obspm.fr




CTA Data Distiller | Search Form | Results | Job List | Selected Job | Authentication: | Sign out user

Search | Analyse

IVOA Standards → SAMP

## Results

show/hide query

SELECT * FROM cta.vo_obscore as o WHERE 1 = intersects(o.s_region, circle('ICRS', 329.717000, -30.226000, 0.001000)) — Send

ADQL query

ObsCore fields

Search

| | dataproduct_type | obs_collection | obs_id | target_name | s_ra (deg) | s_dec (deg) |
|---|---|---|---|---|---|---|
| ☐ | eventlist | 2 | 47802 | PKS 2155-304 | 330.295 | -30.2256 |
| ☐ | eventlist | 2 | 47803 | PKS 2155-304 | 329.138 | -30.2256 |
| ☐ | eventlist | 2 | 47804 | PKS 2155-304 | 329.717 | -29.7256 |
| ☐ | eventlist | 2 | 47827 | PKS 2155-304 | 330.295 | -30.2256 |
| ☐ | eventlist | 2 | 47828 | PKS 2155-304 | 329.138 | -30.2256 |

Showing 1 to 5 of 6 rows 5 records per page

<< < 1 2 > >>

UWS

**Interop** (SAMP)

- Send Result Table
- Send Selected Data

**Analysis tools**

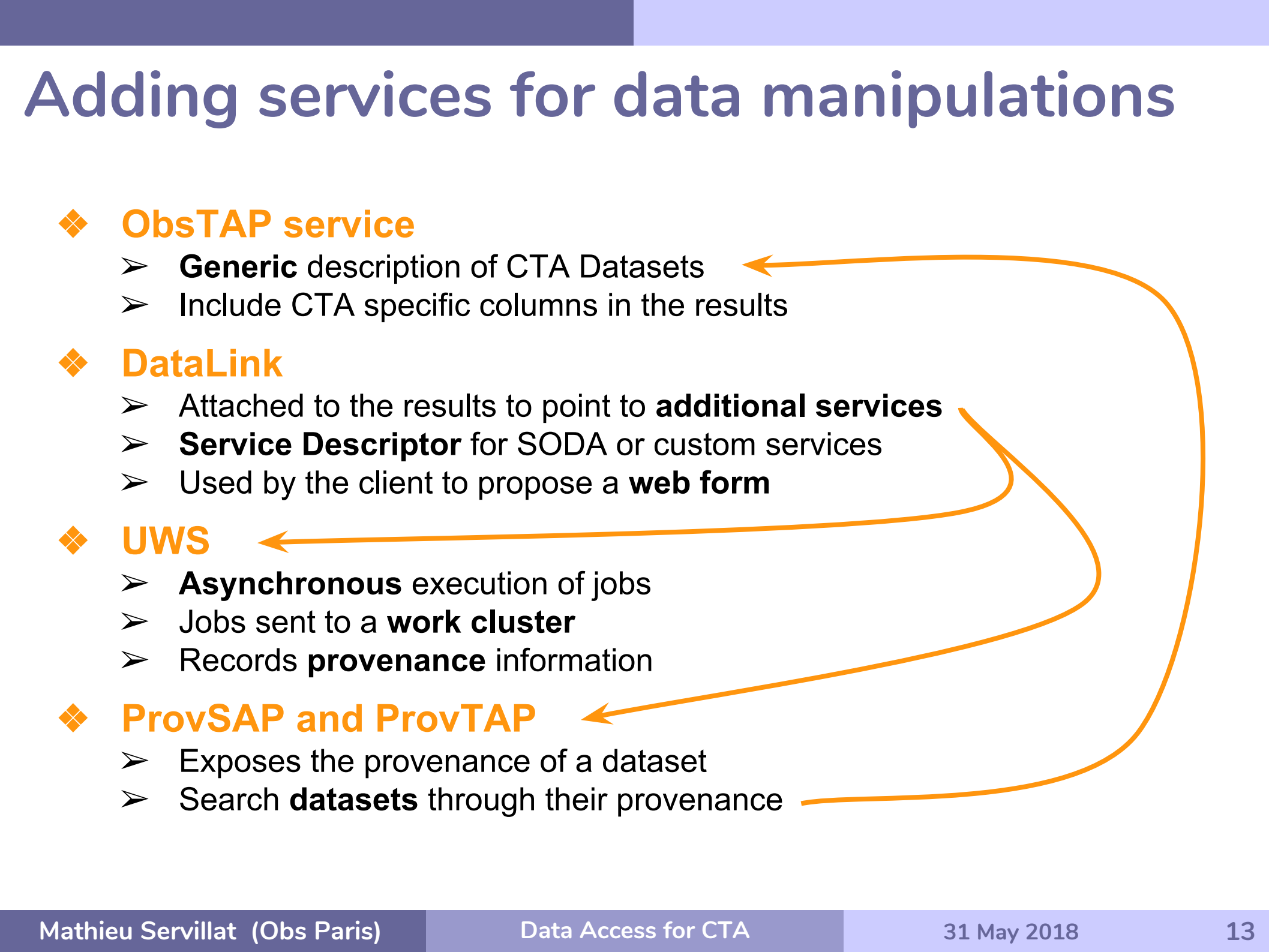
- Count Map(s)
- Fit Spectrum

**Plotting tools**

- TOPCAT
- Aladin
- VOSpec
- SPLAT

# Adding services for data manipulations

- **ObsTAP service**
  - **Generic** description of CTA Datasets
  - Include CTA specific columns in the results
- **DataLink**
  - Attached to the results to point to **additional services**
  - **Service Descriptor** for SODA or custom services
  - Used by the client to propose a **web form**
- **UWS**
  - **Asynchronous** execution of jobs
  - Jobs sent to a **work cluster**
  - Records **provenance** information
- **ProvSAP and ProvTAP**
  - Exposes the provenance of a dataset
  - Search **datasets** through their provenance



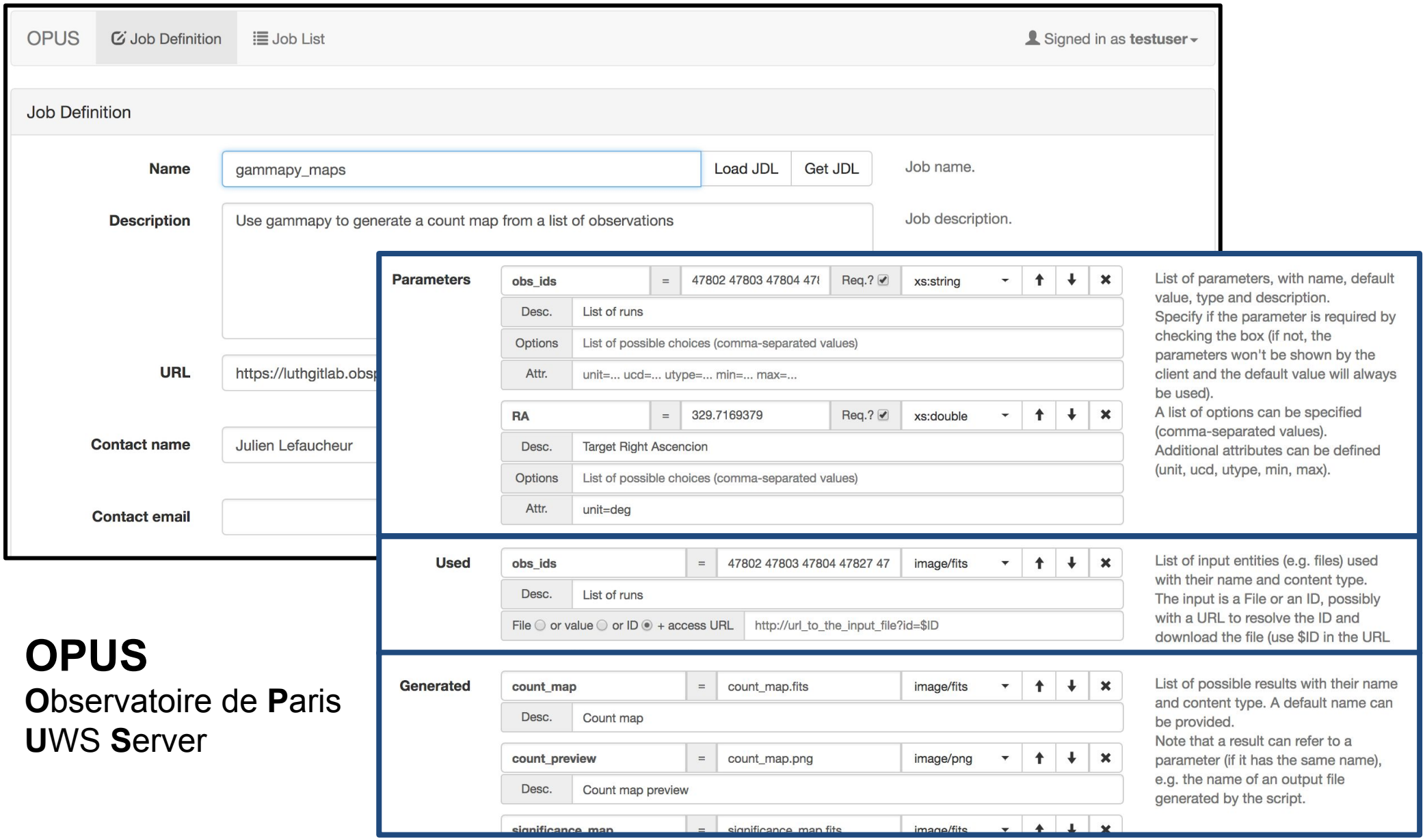
# Provenance ActivityDescription serialization

```
<RESOURCE ID="gammapy_maps" name="gammapy_maps" type="meta" utype="voprov:ActivityDescription">

    <DESCRIPTION>Use gammapy to generate a count map from a list of observations</DESCRIPTION>
    <!-- Service Descriptor -->
    <PARAM name="accessURL" datatype="char" arraysize="*" value="https://voparis-uws-test/rest/gammapy_maps" />
    <PARAM name="standardID" datatype="char" arraysize="*" value="ivo://ivoa.net/std/SODA#1.0" />
    <!-- Activity Description -->
    <PARAM name="type" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.type"/>
    <PARAM name="subtype" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.subtype"/>
    <PARAM name="annotation" datatype="char" arraysize="*" value="Use gammapy to generate a count map from a list of
    <PARAM name="version" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.version"/>
    <PARAM name="doculink" datatype="char" arraysize="*" value="https://luthgitlab.obspm.fr/jlefaucheur/hess_release
    <PARAM name="contact_name" datatype="char" arraysize="*" value="Julien Lefaucheur" utype="voprov:Agent.name"/>
    <PARAM name="contact_email" datatype="char" arraysize="*" value="" utype="voprov:Agent.email"/>
    <!-- UWS job attributes -->
    <PARAM name="executionDuration" datatype="int" value="600" utype="uws:Job.executionDuration"/
    <PARAM name="quote" datatype="int" value="120" utype="uws:Job.quote"/>

    <!-- UWS parameters (Provenance Entities or Parameters) -->
    <GROUP name="InputParams">
        <PARAM ID="obs_ids" arraysize="*" datatype="char" name="obs_ids" value="47802 47803 4780
            <DESCRIPTION>List of runs</DESCRIPTION>
        </PARAM>
        <PARAM ID="RA" datatype="double" name="RA" value="329.7169379" unit="deg"...>
        <PARAM ID="Dec" datatype="double"
        <PARAM ID="nxpix" arraysize="*" da
            <DESCRIPTION>Number of pixels
            <VALUES>
                <MIN value="0"/>
                <MAX value="1000"/>
            </VALUES>
        </PARAM>
        <PARAM ID="nypix" arraysize="*" da
        <PARAM ID="binsz" datatype="float"
    </GROUP>
```

**VOTable**
**DataLink** Service Descriptor
**UWS** Job Description Language
**Provenance** ActivityDescription

```
<!-- Used Entities -->
<GROUP name="Used">
    <GROUP name="obs_ids" utype="voprov:UsedDescription" ref="obs_ids">
        <PARAM arraysize="*" datatype="char" name="role" utype="voprov:UsedDescription.role" value="DL3"/>
        <PARAM arraysize="*" datatype="char" name="location" utype="voprov:EntityDescription.location" value=""/
        <PARAM arraysize="*" datatype="char" name="content_type" utype="voprov:EntityDescription.content_type" v
    </GROUP>
</GROUP>

<!-- Generated Entities / UWS results -->
<GROUP name="Generated" utype="voprov:WasGeneratedBy">
    <GROUP name="count_map" utype="voprov:EntityDescription">
        <DESCRIPTION>Count map</DESCRIPTION>
        <PARAM arraysize="*" datatype="char" name="role" utype="voprov:UsedDescription.role" value="DL4 image"/>
        <PARAM arraysize="*" datatype="char" name="default" utype="voprov:Entity.id" value="count_map.fits"/>
        <PARAM arraysize="*" datatype="char" name="content_type" utype="voprov:EntityDescription.content_type" v
    </GROUP>
    <GROUP name="count_preview" utype="voprov:EntityDescription">
        <DESCRIPTION>Count map preview</DESCRIPTION>
```

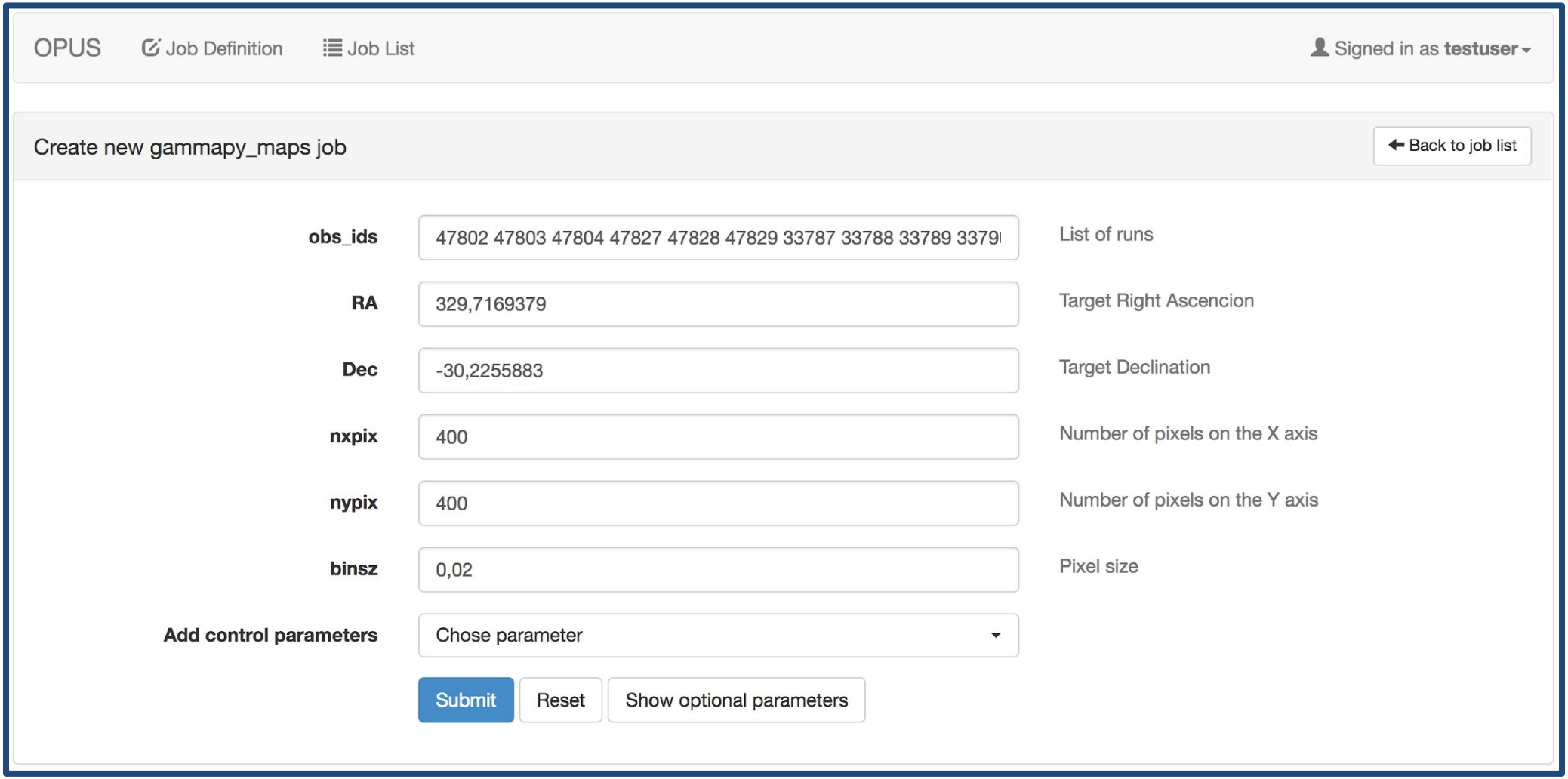
# Definition of the gammapy_maps job




**OPUS**
**O**bservatoire de **P**aris
**U**WS **S**erver

# Submission of a gammapy_maps job

- OPUS reads the **ActivityDescription** file to generate a form
- This form also carries the Datasets **Obscore description**



# Results and Provenance

OPUS | Job Definition | Job List | Signed in as **user**

Job List for gammapy_spectra

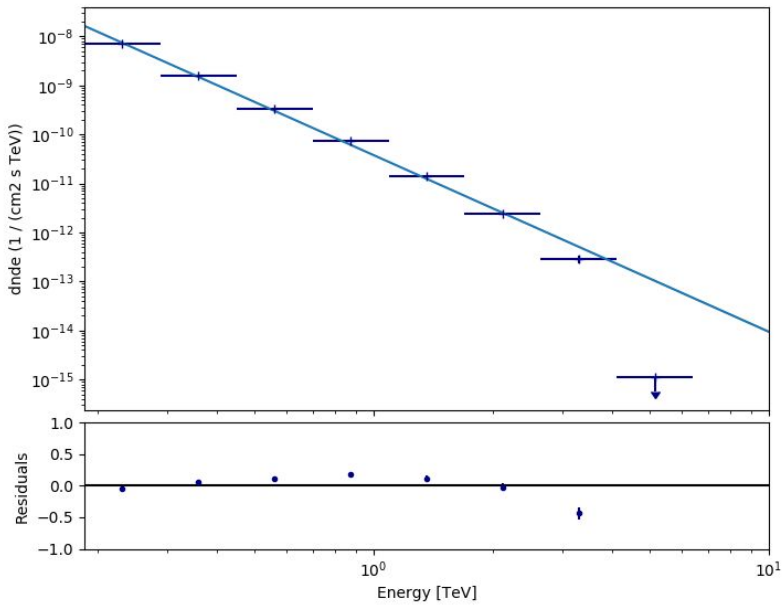
Refresh Job List | Create Test Job | Create New Job

| Type | Start Time | Destruction Time | Phase | Details | Control |
|---|---|---|---|---|---|
| gammapy_spectra | 2017-10-02 10:47:07 | 2017-11-01 10:47:05 | COMPLETED | Properties \| Parameters \| Results | Start \| Abort \| Delete |
| gammapy_spectra | | 2017-11-01 10:47:03 | PENDING | Prop | |
| gammapy_spectra | 2017-09-29 15:07:52 | 2017-10-29 15:07:51 | COMPLETED | Prop | |
| gammapy_spectra | 2017-09-29 14:55:10 | 2017-10-29 14:55:09 | ABORTED | Prop | |
| gammapy_spectra | 2017-09-29 14:21:20 | 2017-10-29 14:21:19 | COMPLETED | Prop | |

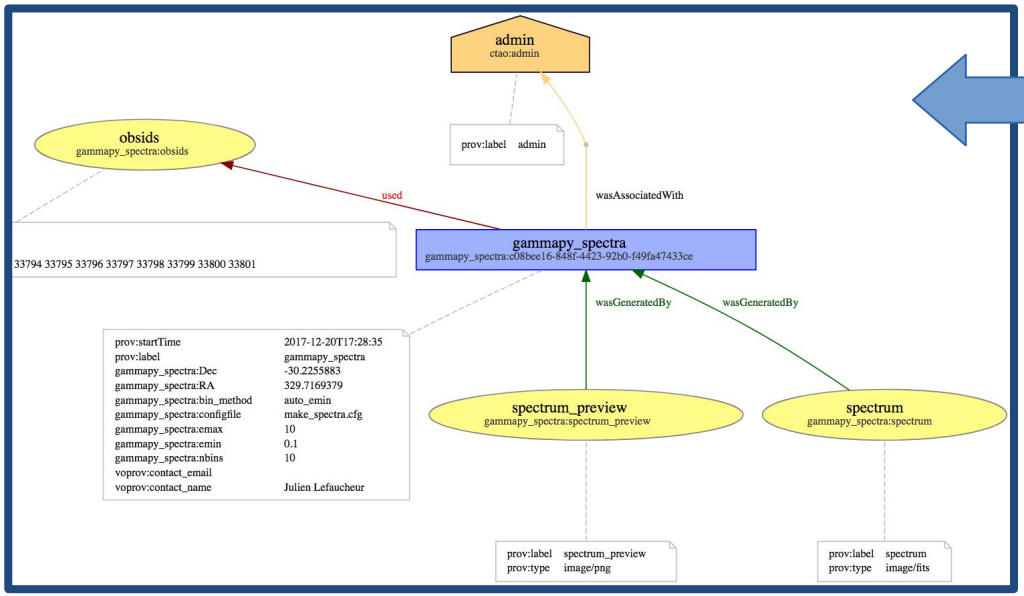
Job Results

**spectrum**: Download [image/fits]

**spectrum_preview**: Download [image/png]

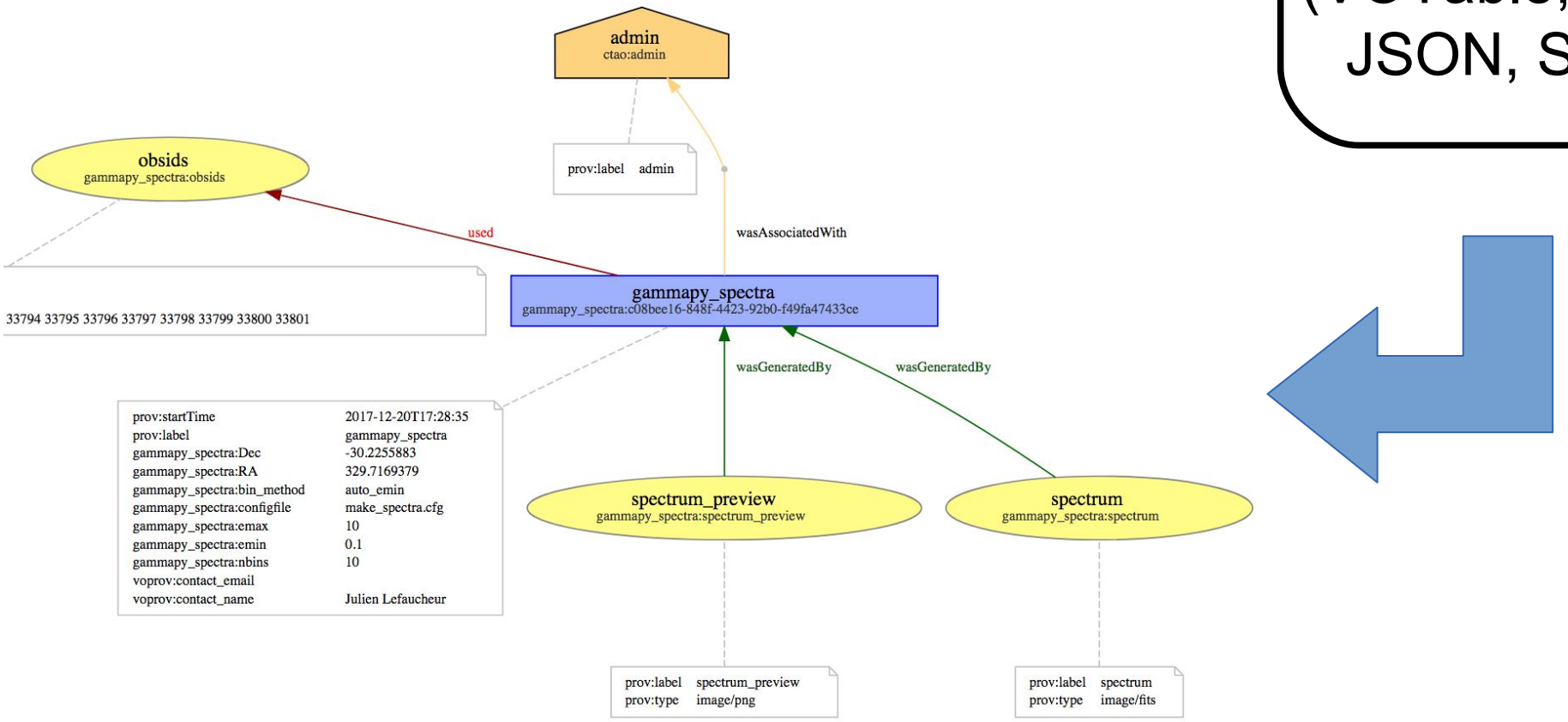



Tracking of Provenance informations



# Provides Provenance files

UWS database + Activity Description (VOTable) = Provenance Files (VOTable, XML, JSON, SVG)



# Conclusions

- **The CTA Data Access use case is a challenge for the VO**
  - **ObsTAP** with additional fields
  - **DataLink**: {links} but also SODA and custom services
  - **UWS** for asynchronous job execution
  - **ProvSAP** and **ProvTAP**
- **Prototypes in development**
  - **TAP Distiller**
    - Project specific search form to submit an **ADQL** query
    - Access to a **TAP** server with **ObsCore** fields (and project specific fields)
  - **OPUS**
    - Light job controller based on **UWS**
    - Enables the tracking of **Provenance** information

# What kind of queries for CTA?

| Use case | Description |
|---|---|
| Cone Search | **Search data for a given Target** |
| ObsCore search | **Search data corresponding to ObsCore keywords**<br>(target_name, time interval, ...), e.g.:<br>● search data for a given **target** at a given **time**<br>● search data in a given **region of the sky**<br>● search data that contain events at **energy** higher than 50 TeV |
| ObsCore optional search | **Search data corresponding to ObsCore optional keywords**<br>(target_class, data_rights, ...), e.g.:<br>● search **public** data for all **blazars**<br>● search data for a given **proposal_id** |
| ObsConfig search | **Search data corresponding to ObsConfig keywords**<br>(sub_array_name, pointing_mode, obs_mode ...), e.g.:<br>● search data that include the **Large Size Telescopes** (LSTs)<br>● search data for a given target, that do not include the divergent **pointing mode** |
| Provenance search | **Search data corresponding to Provenance keywords**<br>(calib_version, creation_date ...), e.g.:<br>● search data produced by a given **version of the pipeline** and for a given target<br>● search data produced using a given **reconstruction method**<br>● search data for a given target produced with **loose cuts** |

# CTA TAP Distiller prototype