

Provenance in the context of CTA (the Cherenkov Telescope Array)

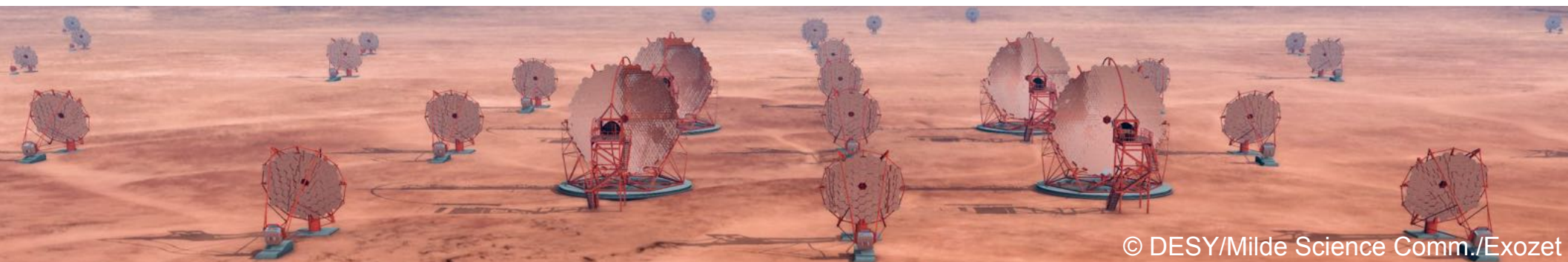
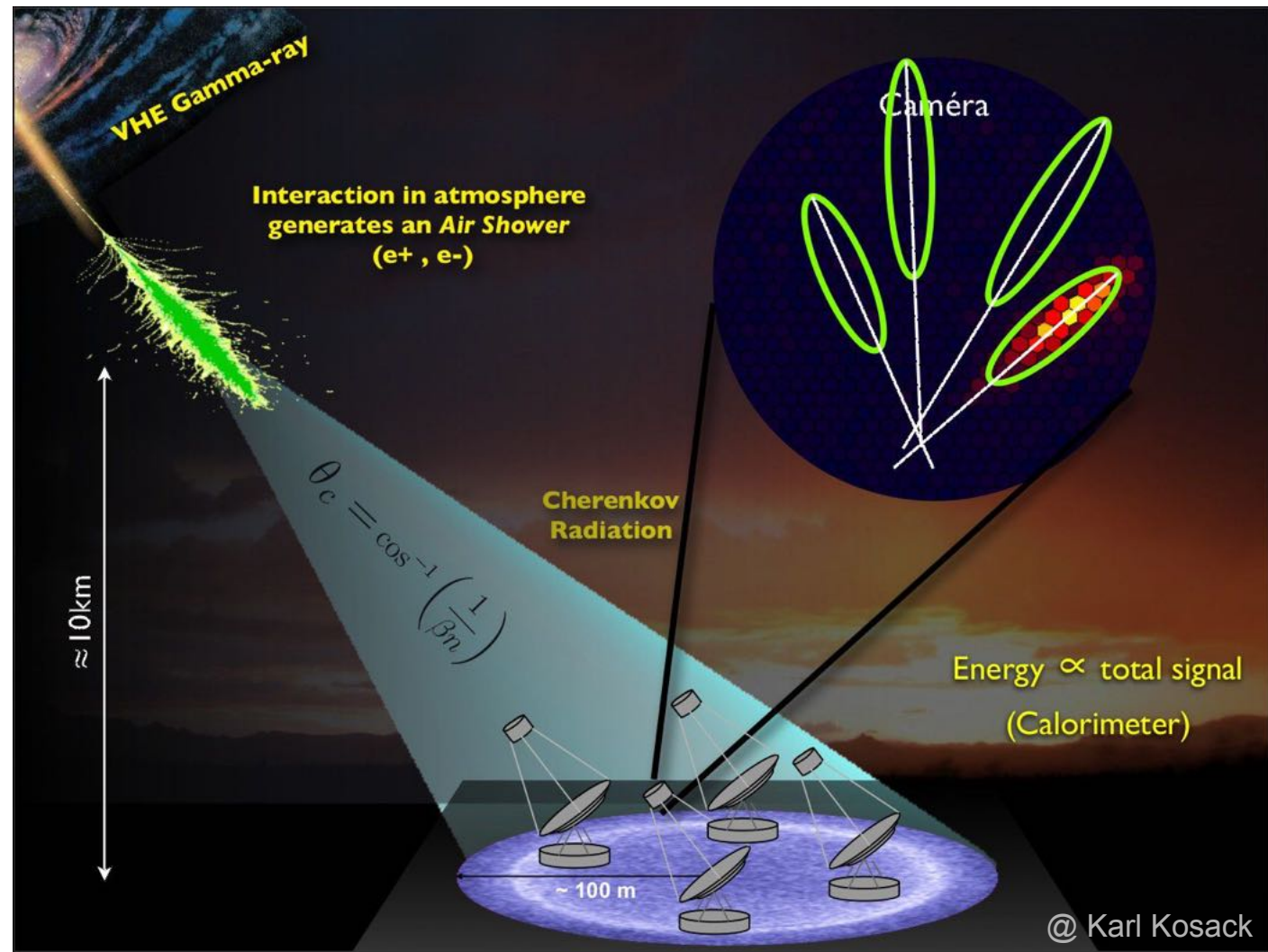
Mathieu Servillat, Catherine Boisson, Karl Kosack

August 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



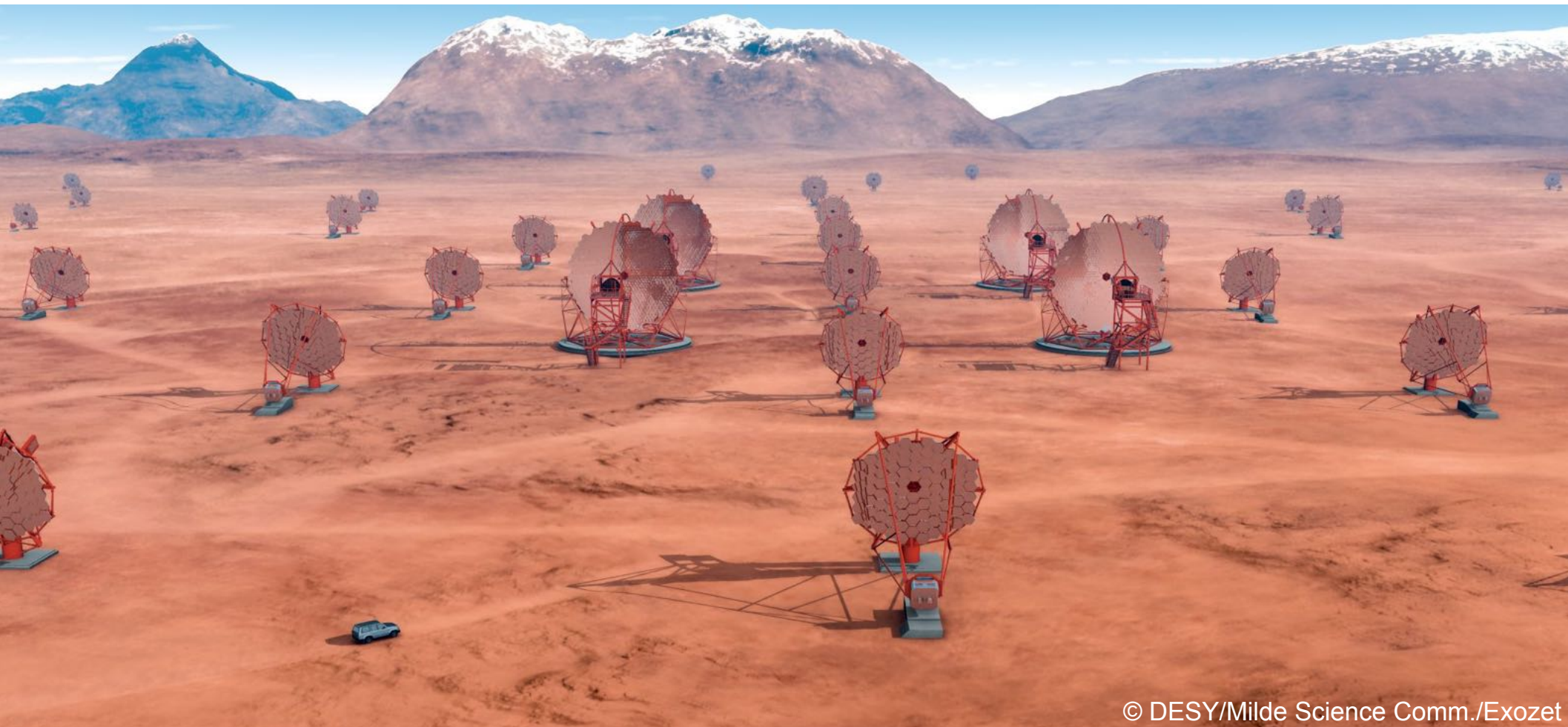
© DESY/Milde Science Comm./Exozet



cherenkov
telescope
array

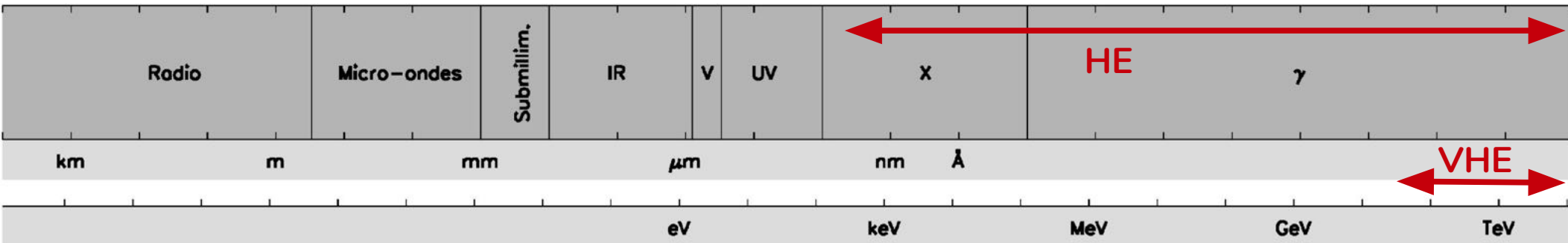
Observatory

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

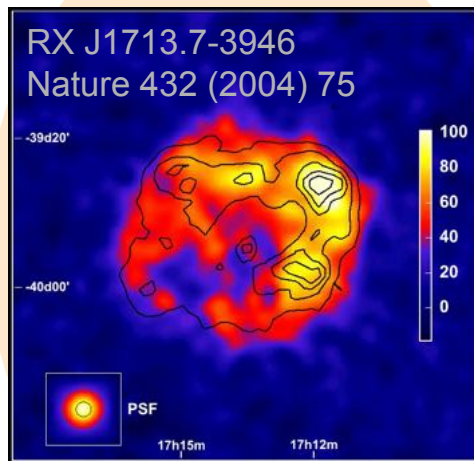


© DESY/Milde Science Comm./Exozet

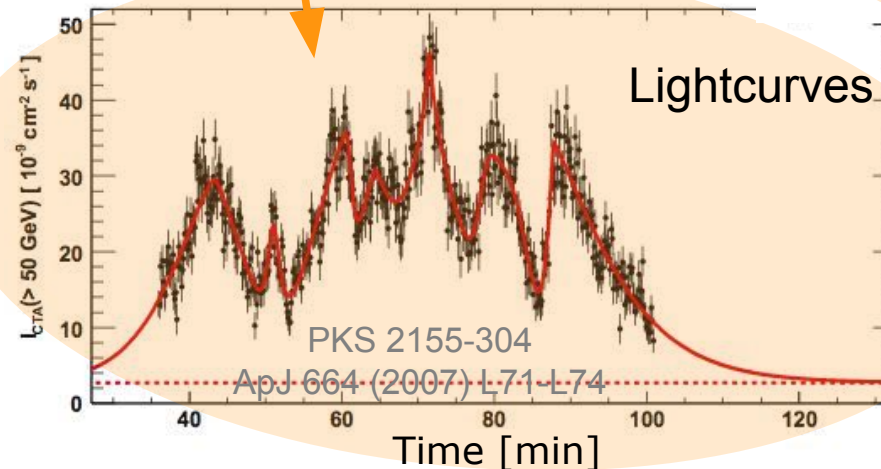
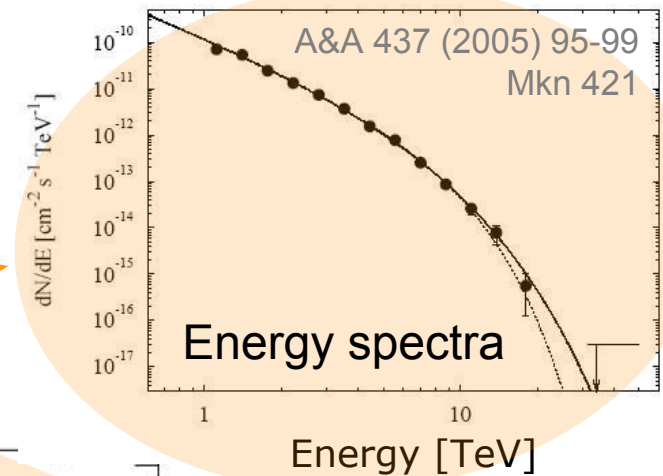
Very high energy (VHE) data



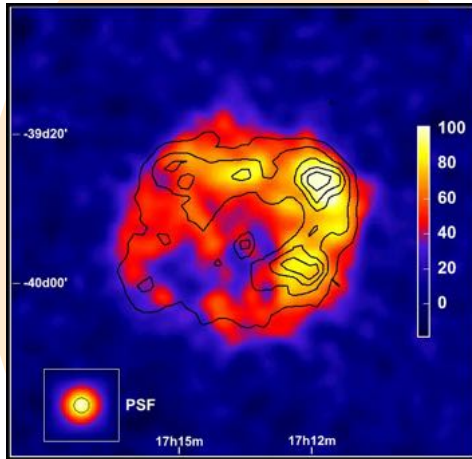
- ◆ Several orders of magnitude
- ◆ Photon **counting**
- ◆ Low count **statistics**, high background
- ◆ **Event lists**
(coordinates, time, energy)



Images

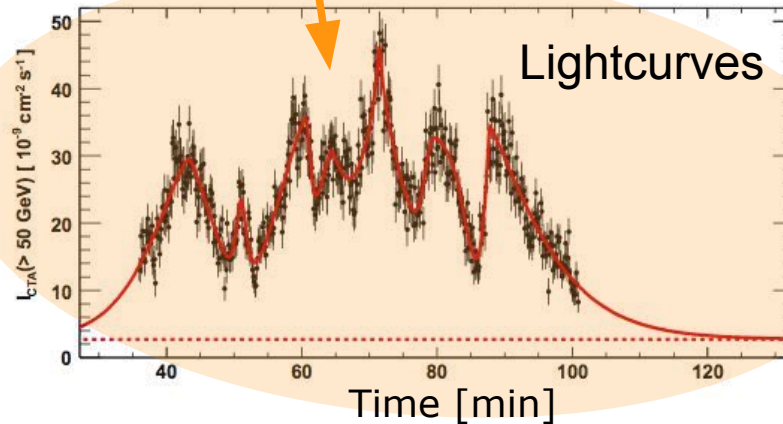


Multi-wavelength analysis

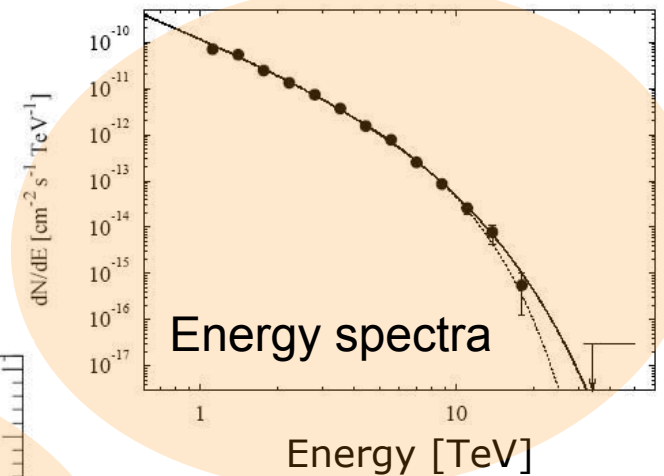


Images

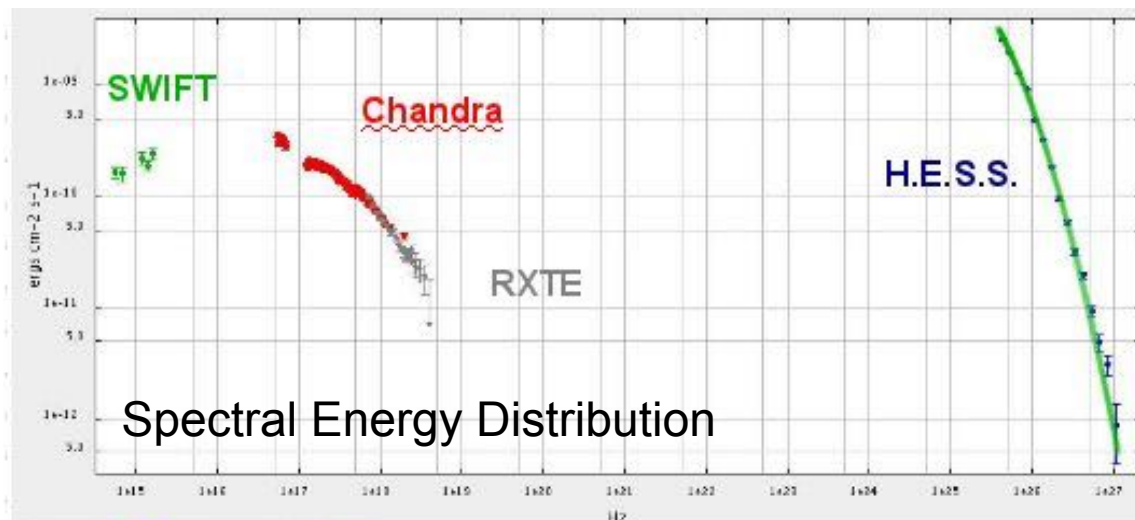
CTA Event lists
(coordinates, time, energy)



Lightcurves



Energy spectra



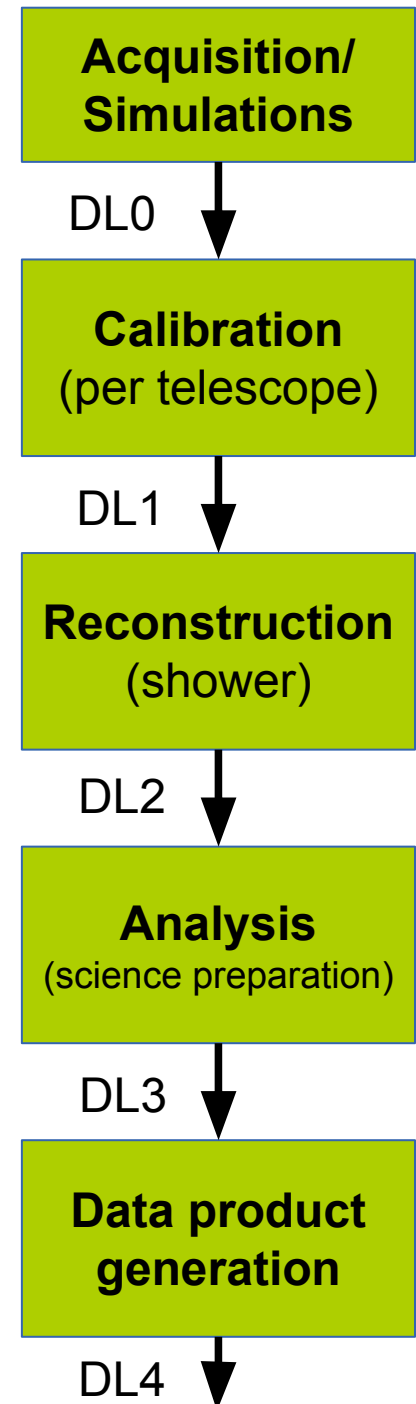
Spectral Energy Distribution

Compatible data
at other wavelength?

Simultaneous
Calibrated
Specific Processing?
Context?

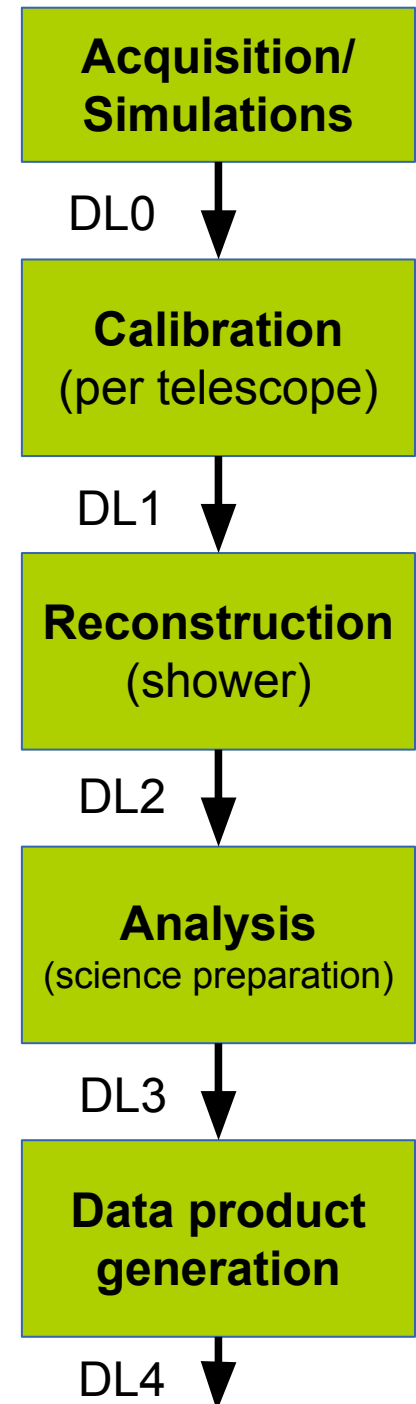
Data levels and workflow

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 Published & Archived	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.	$10^{-5} - 10^{-3}$



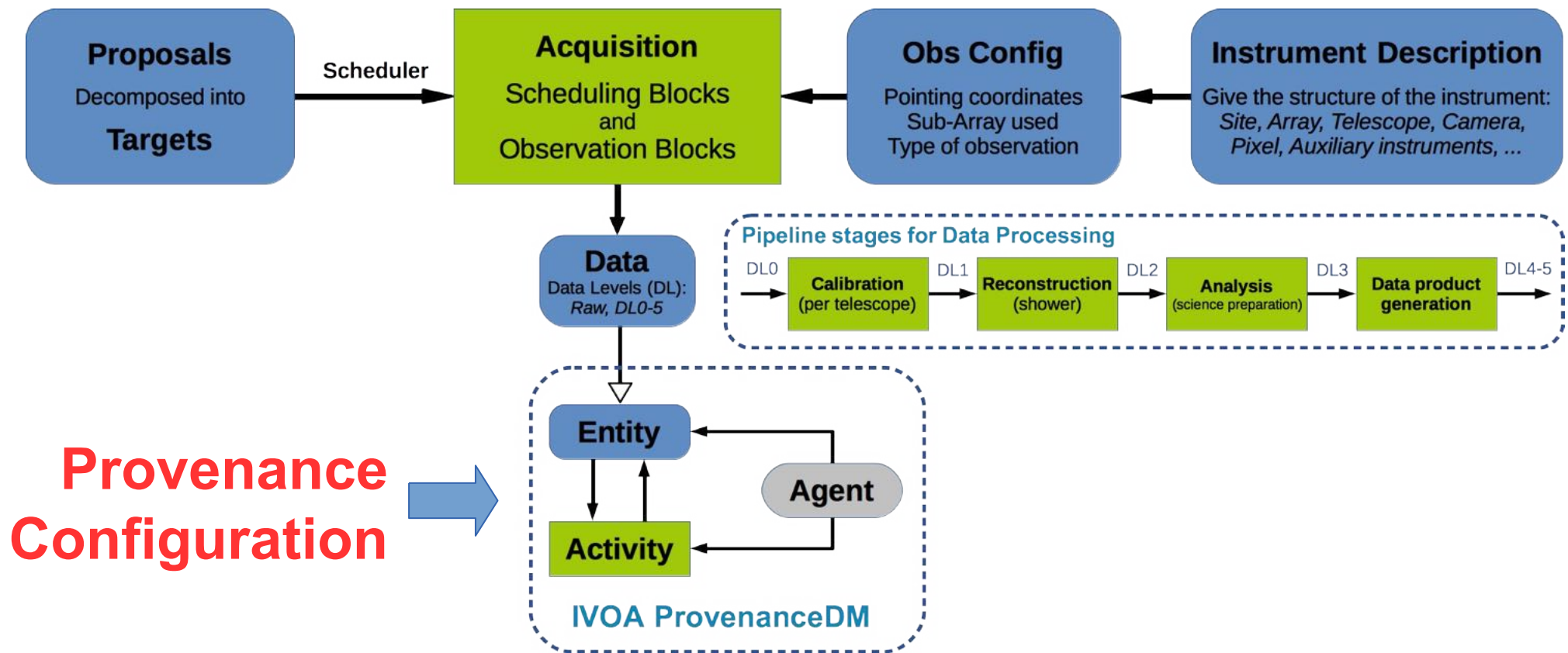
CTA Pipeline Requirements

- ◆ **Open** observatory
- ◆ 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**

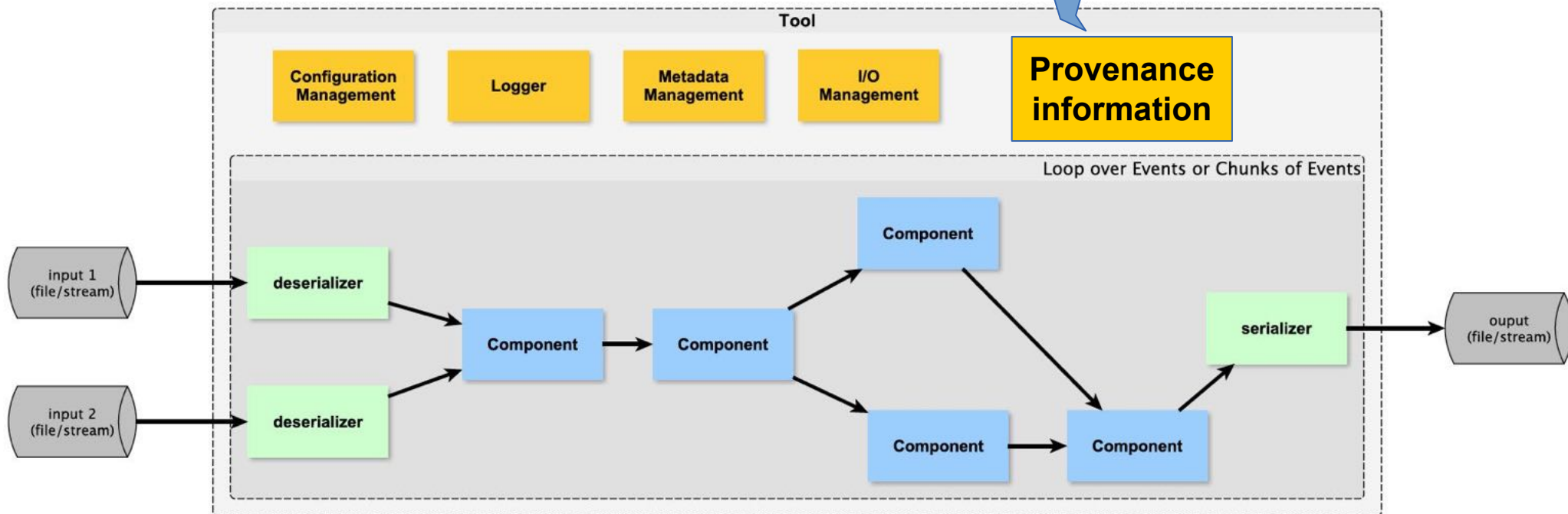


Provenance in the Pipeline



cherenkov
telescope
array

- ◆ **Ctapipe**: a CTA data processing framework (prototype, not official, not recommended for use!)
<https://github.com/cta-observatory/ctapipe>
- ◆ **Tool Python class** providing configuration, logger, I/O management... and **Provenance information**



Provenance class for ctapipe

```
from ctapipe.core import Provenance

prov = Provenance()
# prov a singleton, so this gives you the same provenance class

prov.start_activity("some_activity")

... # do things
prov.add_input_file("test.txt")
prov.add_output_file("out.txt")

prov.start_activity("some_sub_activity")

# do more things
prov.add_output_file("out2.txt")

prov.finish_activity() # finish some_activity
prov.finish_activity() # finish some_sub_activity
```

- ◆ Importance of **persistent identifiers**
- ◆ Also records **system configuration, state, and software versions** → **contextual information**

Ctapipe projects:

<https://cta-observatory.github.io/ctapipe/>

Ctapipe core classes:

<https://github.com/cta-observatory/ctapipe/tree/master/ctapipe/core>

Provenance notebook:

<https://github.com/cta-observatory/ctapipe/blob/master/examples/notebooks/provenance.ipynb>

Additional Slides

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

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

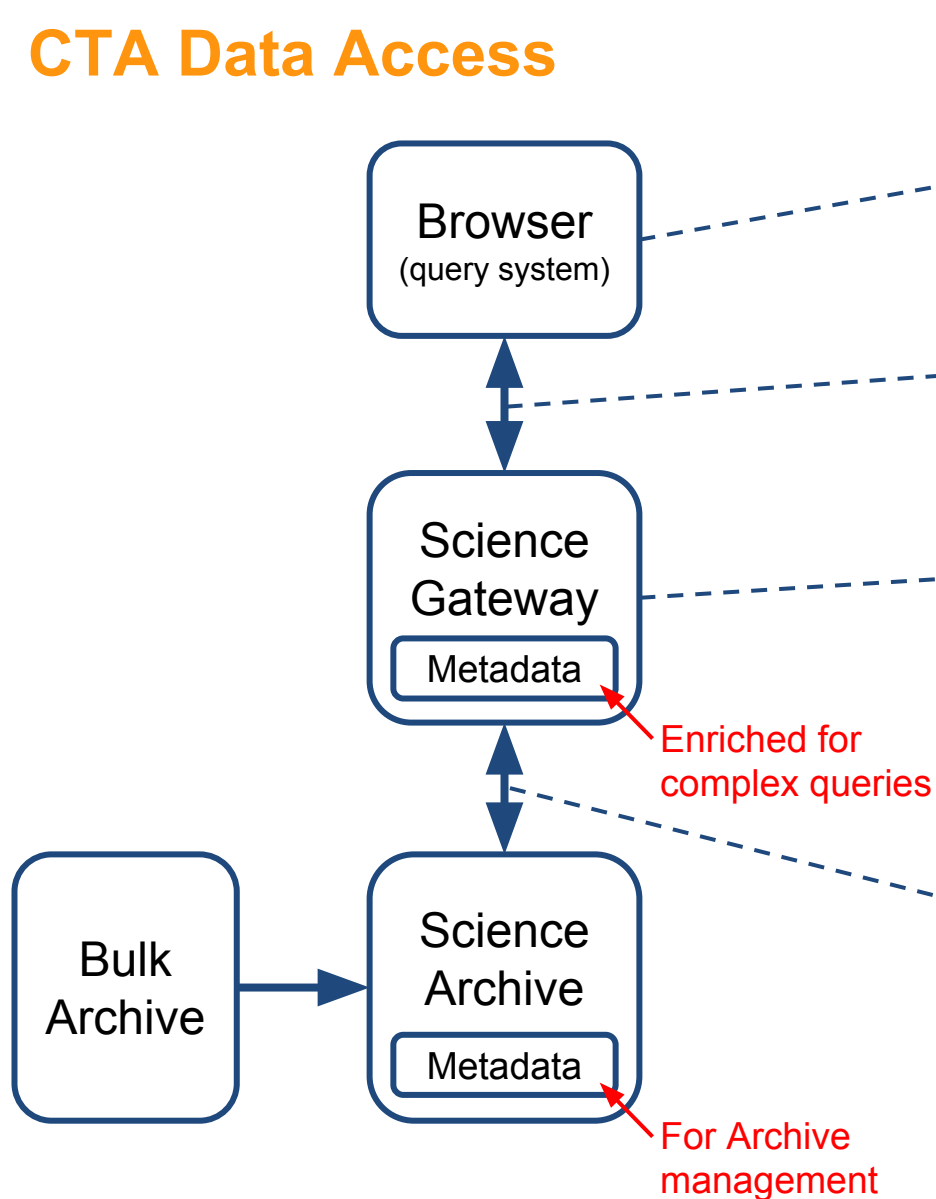
- **Asynchronous** execution of jobs
- Jobs sent to a **work cluster**
- Records **provenance** information

❖ ProvSAP and ProvTAP

- Exposes the provenance of a dataset
- Search **datasets** through their provenance

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

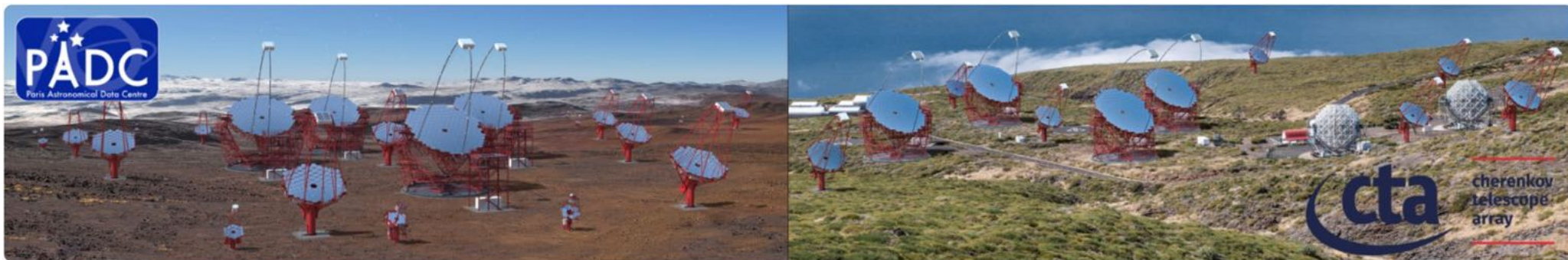
dataproduct_level

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

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

IVOA Standards

SAMP

ObsCore fields

UWS

	dataprodukt_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

Interop (SAMP)

Send Result Table

Send Selected Data

Analysis tools

Count Map(s)

Fit Spectrum

Plotting tools

TOPCAT

Aladin

VOSpec

SPLAT

Showing 1 to 5 of 6 rows

5

records per page

<< < 1 2 > >>

Definition of the gammapy_maps job

OPUS [Job Definition](#) [Job List](#) Signed in as testuser

Job Definition

Name [Load JDL](#) [Get JDL](#) Job name.

Description Job description.

URL

Contact name

Contact email

Parameters

obs_ids	=	47802 47803 47804 471	Req.? ☒	xs:string	↑	↓	×
Desc.	List of runs						
Options	List of possible choices (comma-separated values)						
Attr.	unit=... ucd=... utype=... min=... max=...						
RA	=	329.7169379	Req.? ☒	xs:double	↑	↓	×
Desc.	Target Right Ascension						
Options	List of possible choices (comma-separated values)						
Attr.	unit=deg						

List of parameters, with name, default value, type and description. Specify if the parameter is required by checking the box (if not, the parameters won't be shown by the client and the default value will always be used). A list of options can be specified (comma-separated values). Additional attributes can be defined (unit, ucd, utype, min, max).

Used

obs_ids	=	47802 47803 47804 47827 47	image/fits	↑	↓	×
Desc.	List of runs					
File ☐ or value ☐ or ID ☒ + access URL	http://url_to_the_input_file?id=\$ID					

List of input entities (e.g. files) used with their name and content type. The input is a File or an ID, possibly with a URL to resolve the ID and download the file (use \$ID in the URL).

Generated

count_map	=	count_map.fits	image/fits	↑	↓	×
Desc.	Count map					
count_preview	=	count_map.png	image/png	↑	↓	×
Desc.	Count map preview					
significance_map	=	significance_map.fits	image/fits	↑	↓	×

List of possible results with their name and content type. A default name can be provided. Note that a result can refer to a parameter (if it has the same name), e.g. the name of an output file generated by the script.

OPUS
Observatoire de Paris
UWS Server

Submission of a `gammapy_maps` job

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

OPUS

[Job Definition](#)

[Job List](#)

Signed in as **testuser** ▾

Create new `gammapy_maps` job

← Back to job list

obs_ids

47802 47803 47804 47827 47828 47829 33787 33788 33789 33790

List of runs

RA

329,7169379

Target Right Ascension

Dec

-30,2255883

Target Declination

nxpix

400

Number of pixels on the X axis

nypix

400

Number of pixels on the Y axis

binsz

0,02

Pixel size

Add control parameters

Chose parameter ▾

Submit

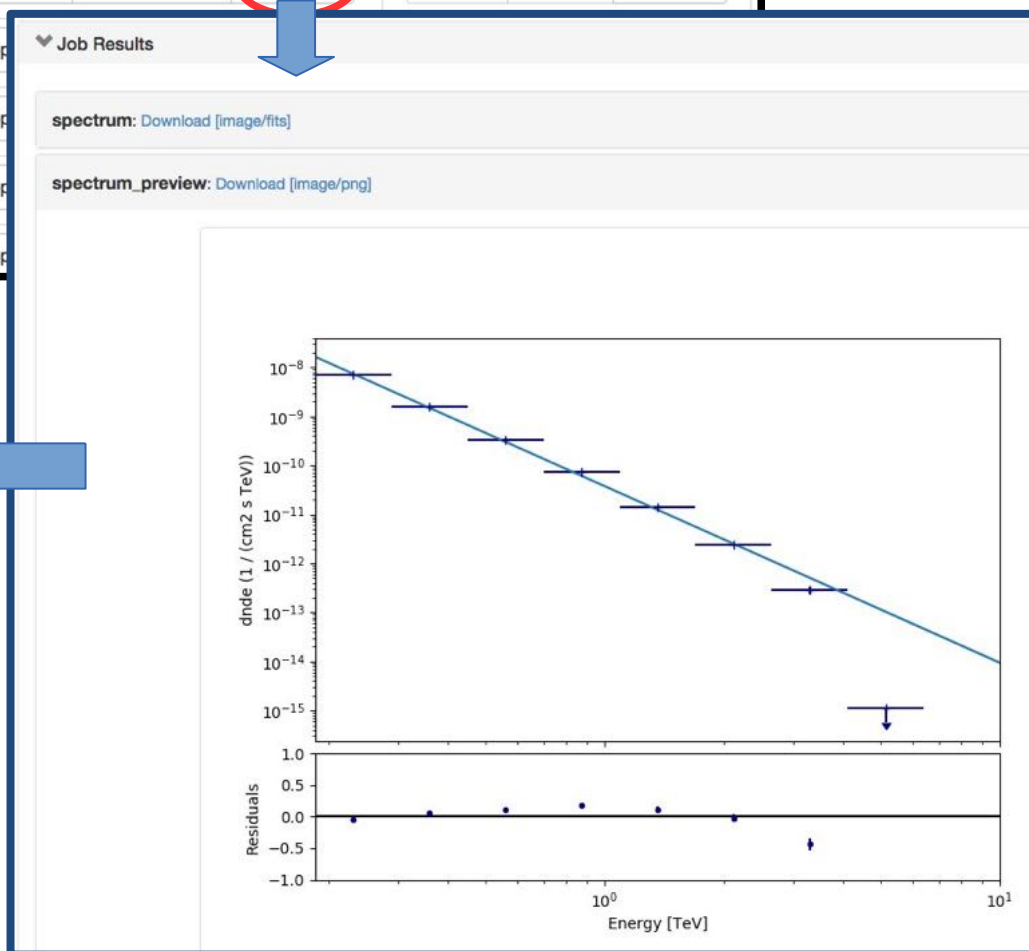
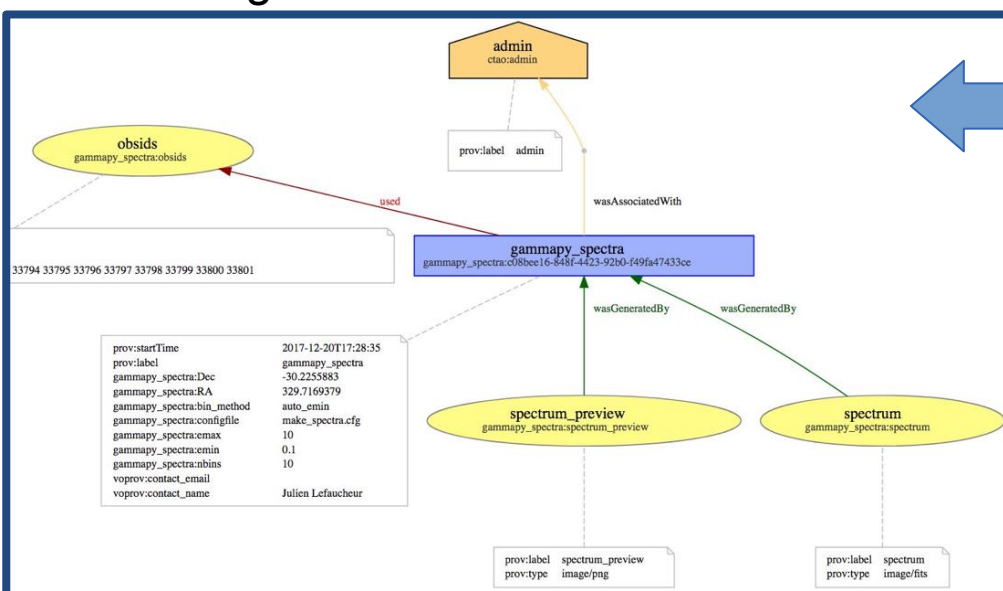
Reset

Show optional parameters

Results and Provenance

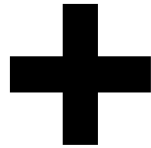
OPUS		Job Definition	Job List	Signed in as user	
Job List for gammapy_spectra		Refresh Job List		Create Test Job	Create New Job
Type	Start Time	Destruction Time	Phase	Details	Control
gammapy_spectra	2017-10-02 10:47:07	2017-11-01 10:47:05	COMPLETED	Properties Parameters Results	Start Abort Delete
gammapy_spectra		2017-11-01 10:47:03	PENDING	Properties	
gammapy_spectra	2017-09-29 15:07:52	2017-10-29 15:07:51	COMPLETED	Properties	
gammapy_spectra	2017-09-29 14:55:10	2017-10-29 14:55:09	ABORTED	Properties	
gammapy_spectra	2017-09-29 14:21:20	2017-10-29 14:21:19	COMPLETED	Properties	

Tracking of Provenance informations



Provides Provenance files

UWS
database

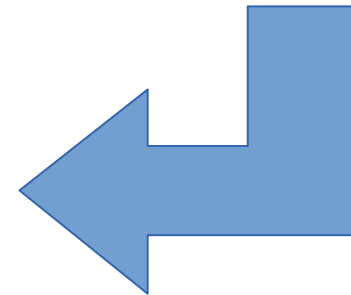
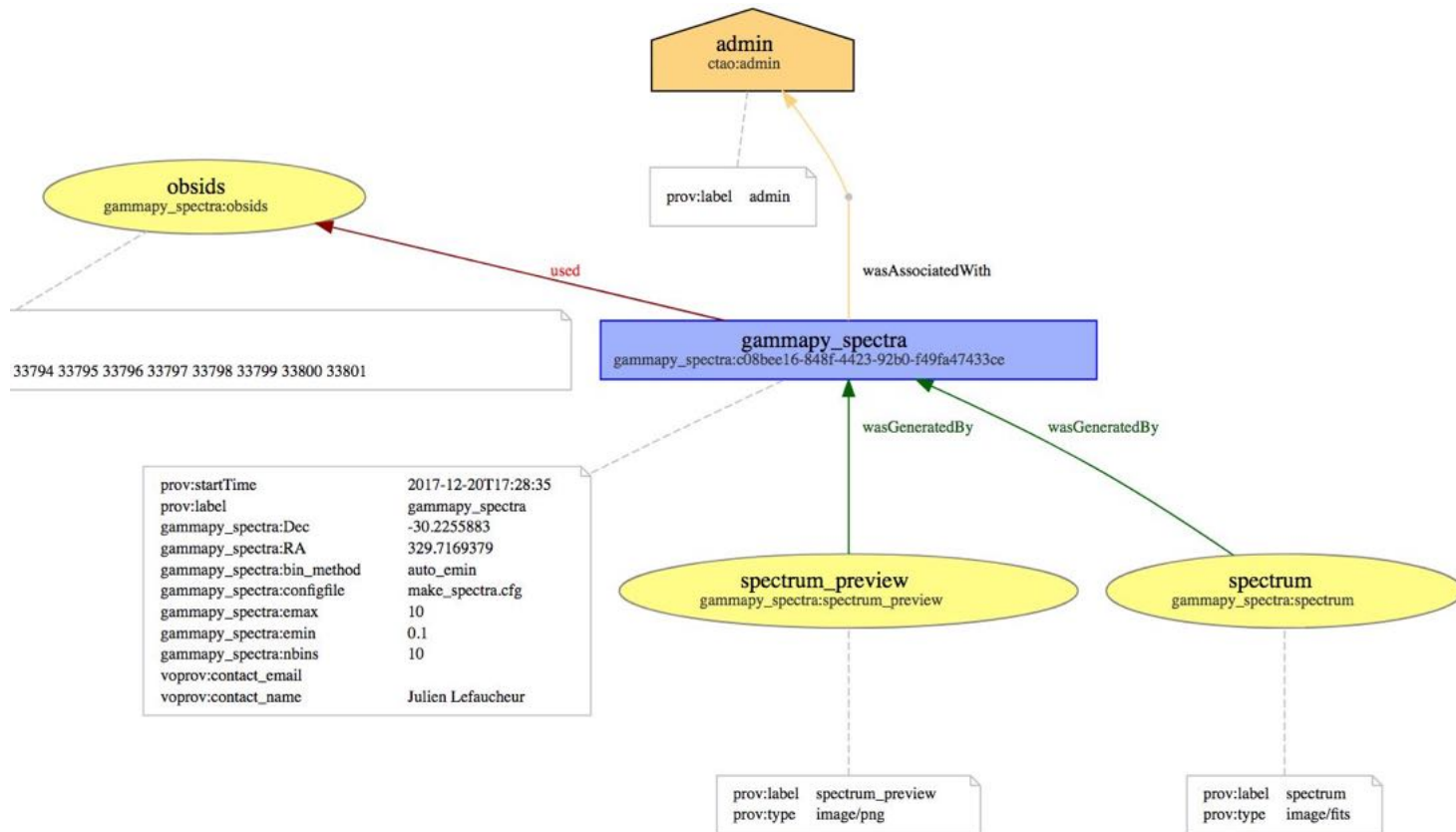


Activity
Description
(VOTable)



Provenance
Files

(VOTable, XML,
JSON, SVG)



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

CTA TAP Distiller prototype

