

CTA use case for Provenance

(Cherenkov Telescope Array)

Mathieu Servillat

Observatoire de Paris
Paris Astronomical Data Centre

IVOA Provenance Meeting





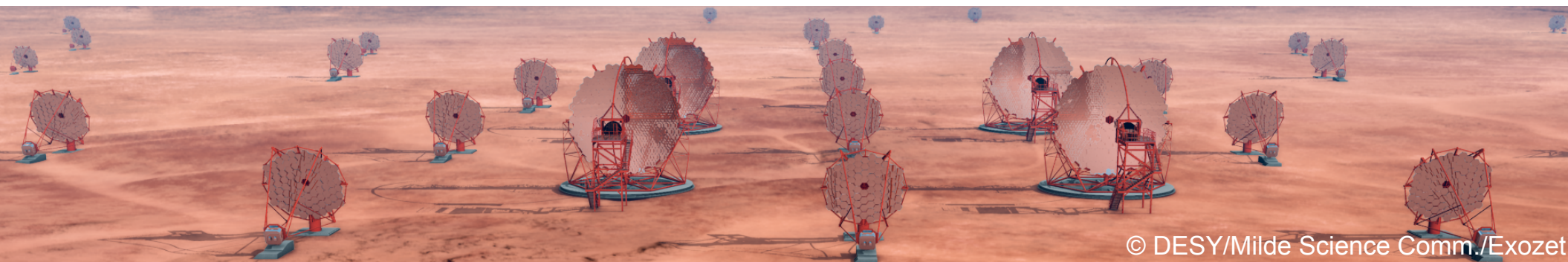
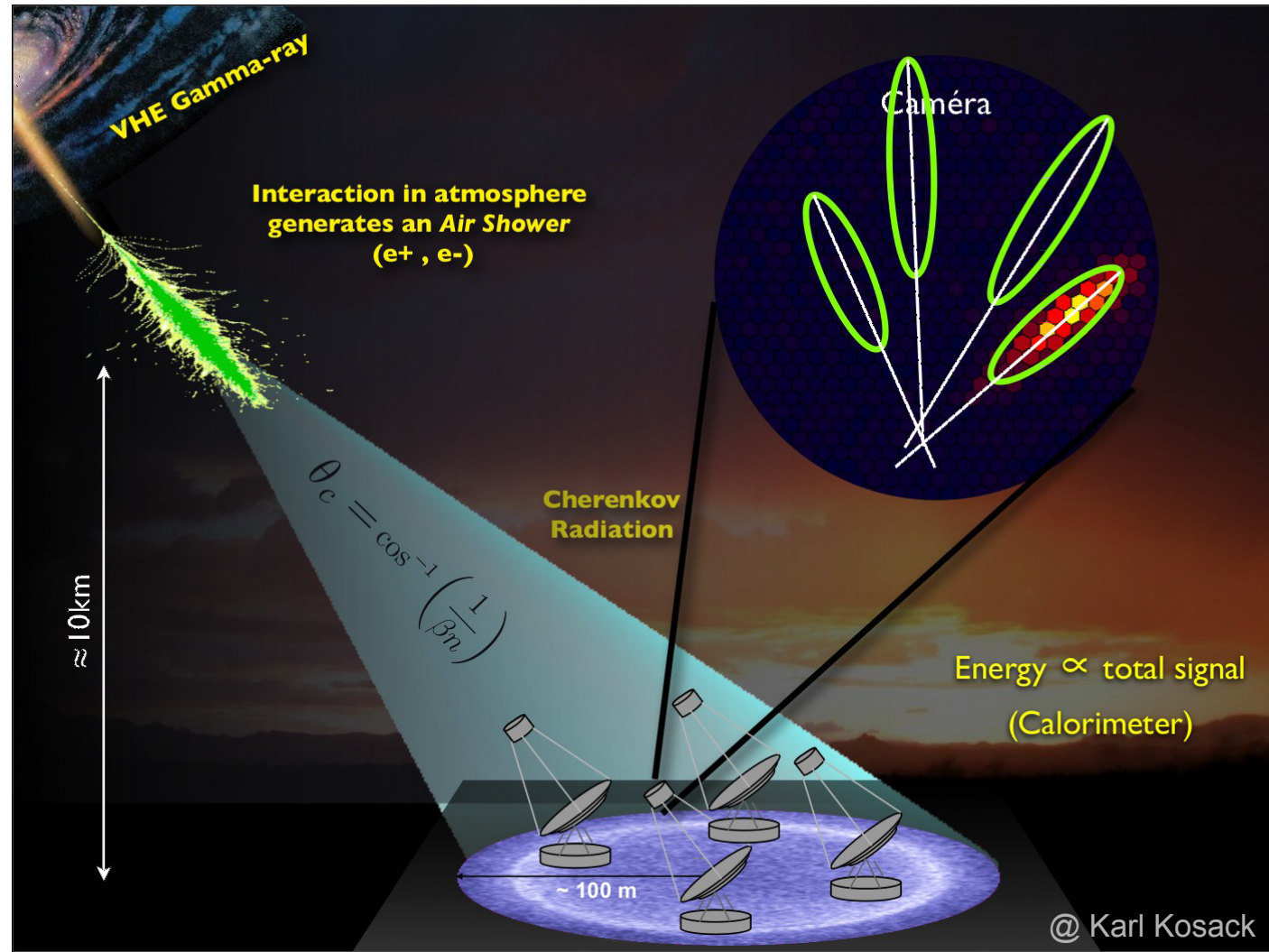
- ◆ Two arrays of 100 (South) et 20 (North) Cherenkov telescopes (4, 12 et 24 m in diameter)
- ◆ July 2015: site selection, Chile (ESO) and La Palma
- ◆ 2016: pre-production phase
- ◆ 2018-2013: production phase
- ◆ Observatory open to the Astronomy community



© DESY/Milde Science Comm./Exozet



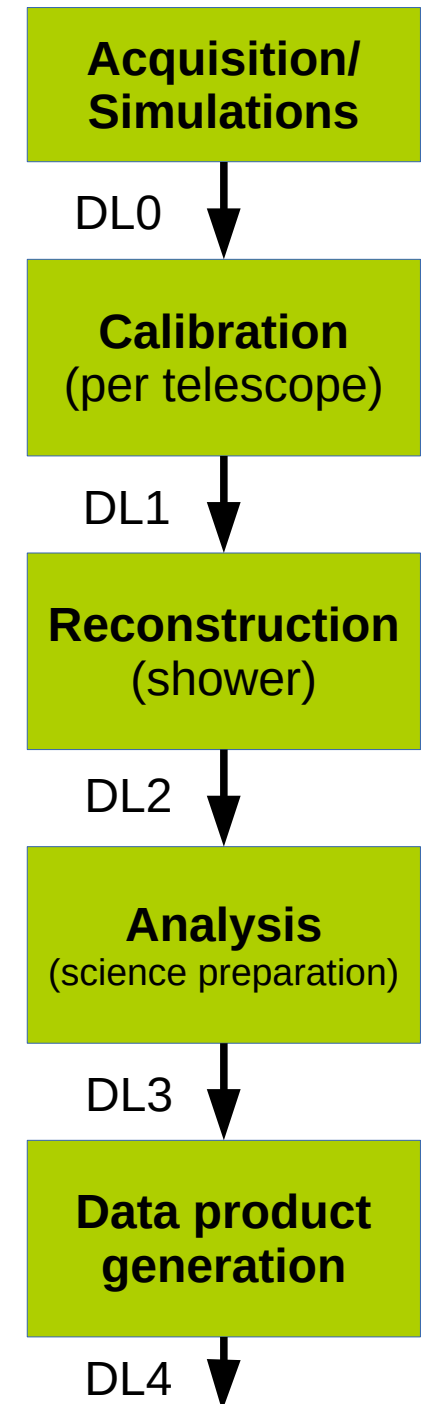
- ◆ **Dark nights** (small duty cycle)
- ◆ Field of view: 5-8 degrees
- ◆ **Event Reconstruction:**
photon, particle shower,
Cherenkov light
(faint, few nanoseconds)
- ◆ **Atmosphere** = calorimetre
Simulations, assumptions
- ◆ **Complex Metadata**,
need to be structured



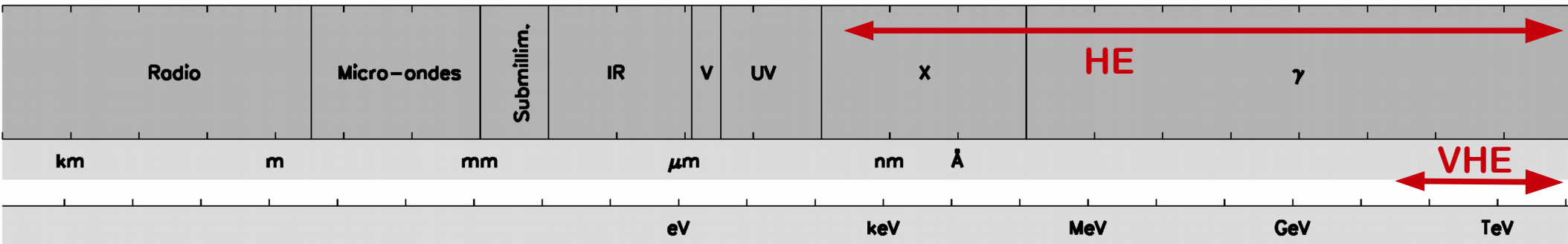
© DESY/Milde Science Comm./Exozet

Data levels and workflow

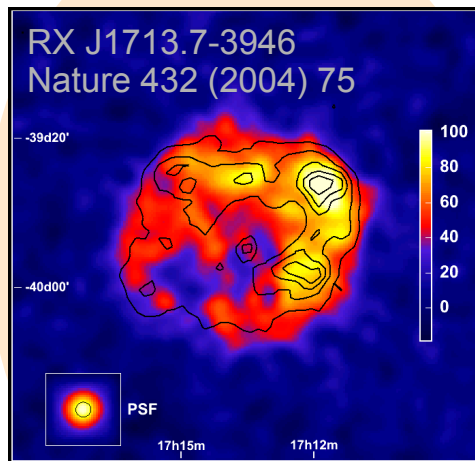
Data Level	Short Name	Description	Data reduction factor
Level 0 (DL0)	DAQ-RAW	Data from the Data Acquisition hardware/software.	
Level 1 (DL1)	CALIBRATED	Physical quantities measured in each separate camera: photons, arrival times, etc., and per-telescope parameters derived from those quantities.	1-0.2
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	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.	$10^{-5} - 10^{-3}$



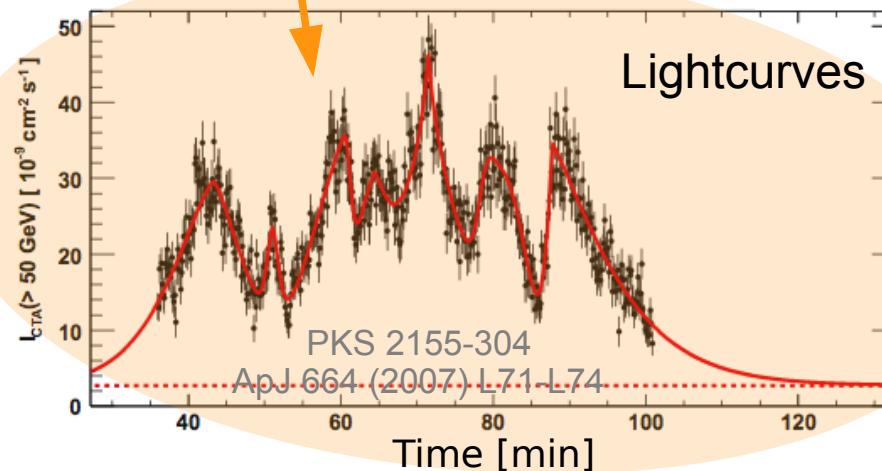
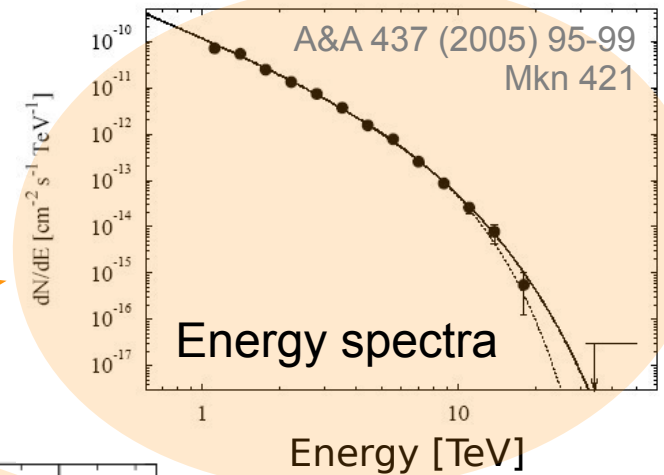
Very high energy (VHE) data



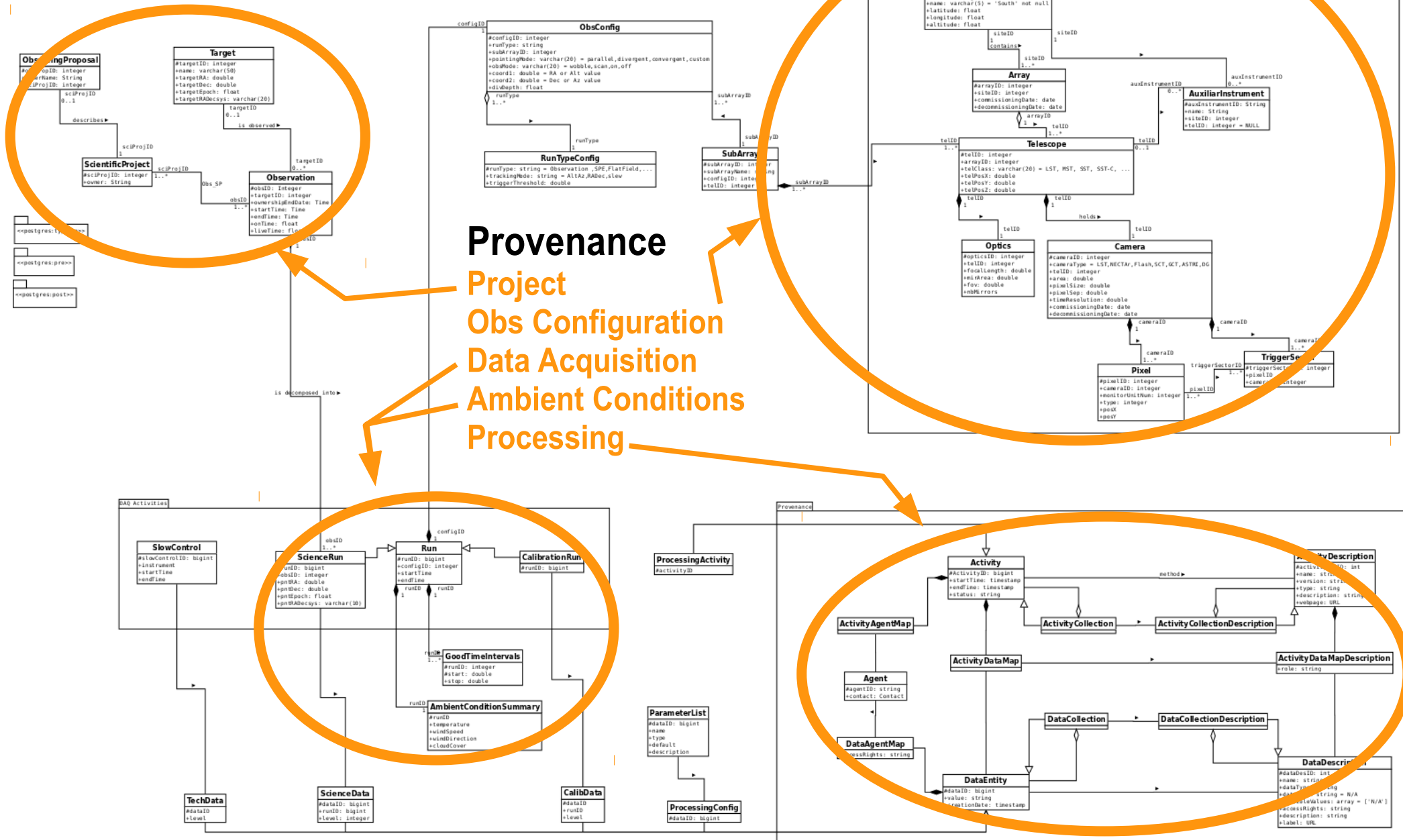
- ◆ Several orders of magnitude
- ◆ Photon counting
- ◆ Low count statistics, high background
- ◆ **Event lists**
(coordinates, time, energy)



Images

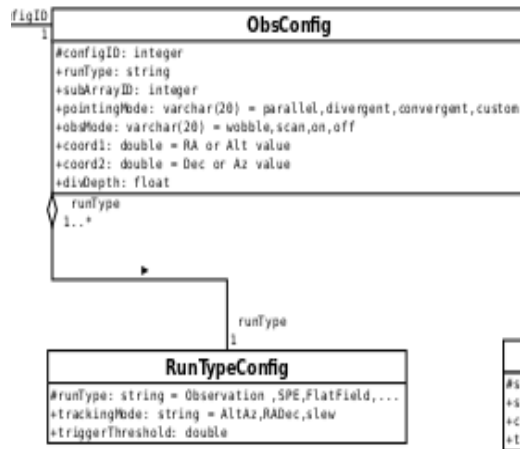


All you need is metadata!



CTA data model

ObsConfig



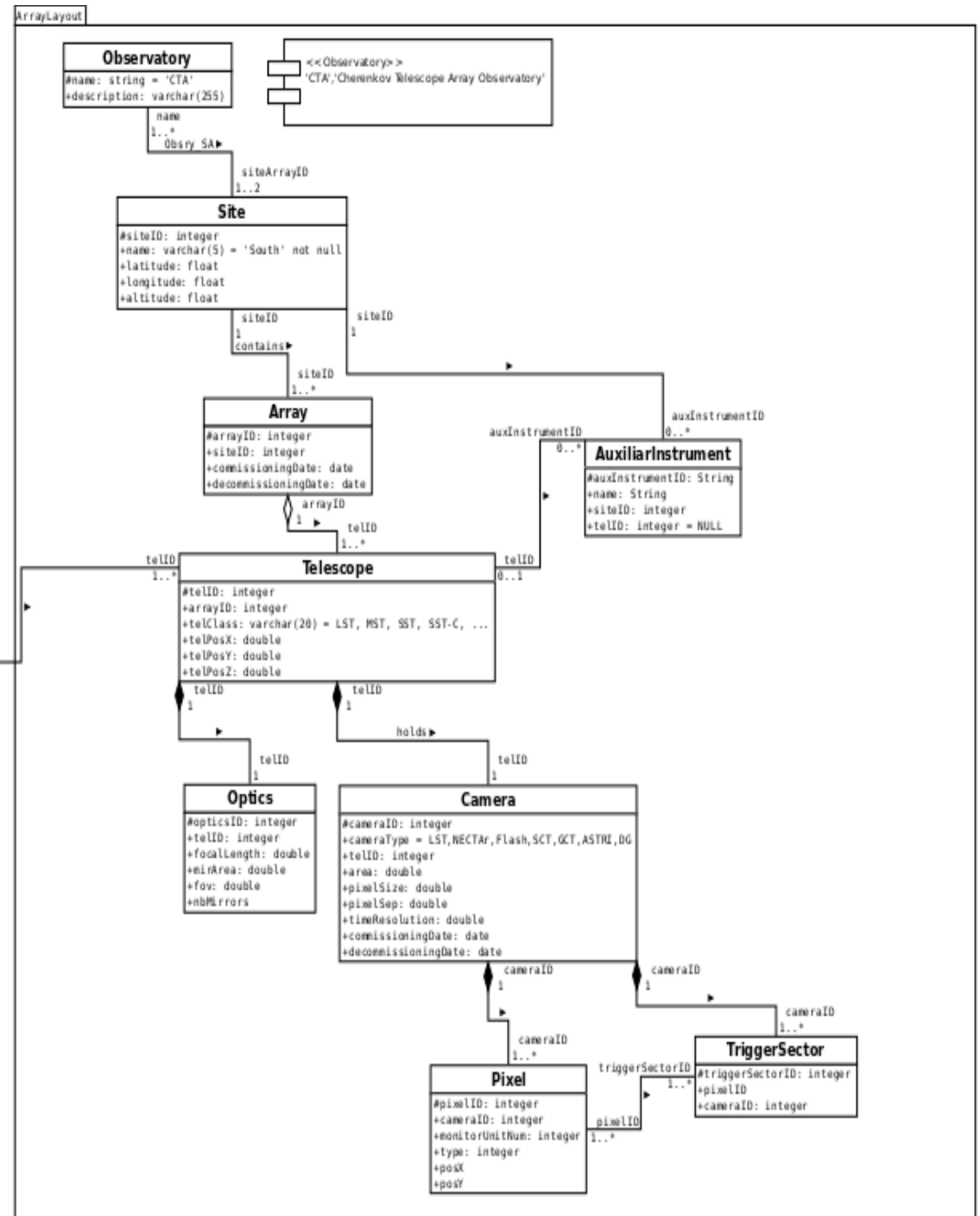
Examples :

Number of telescopes involved

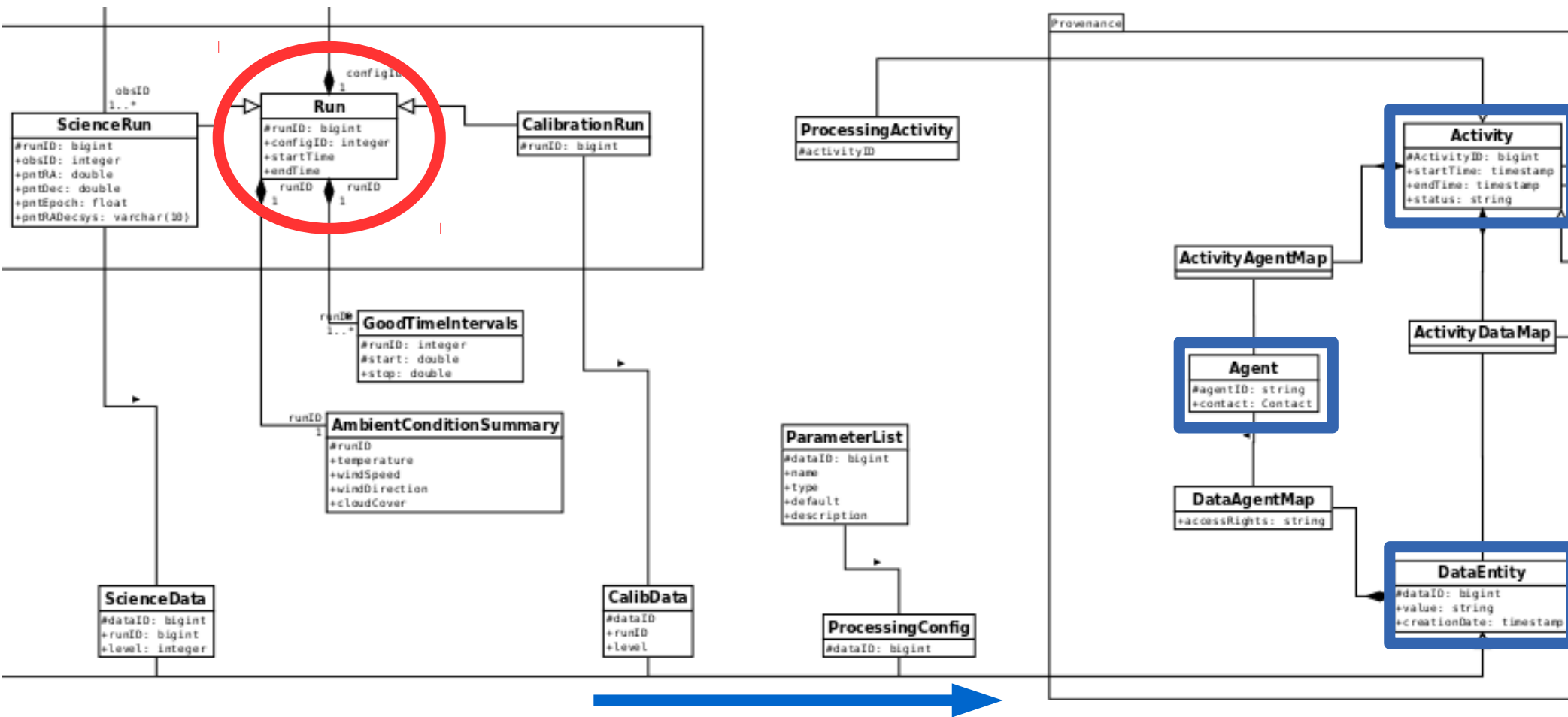
Field of view

Pointing direction

...

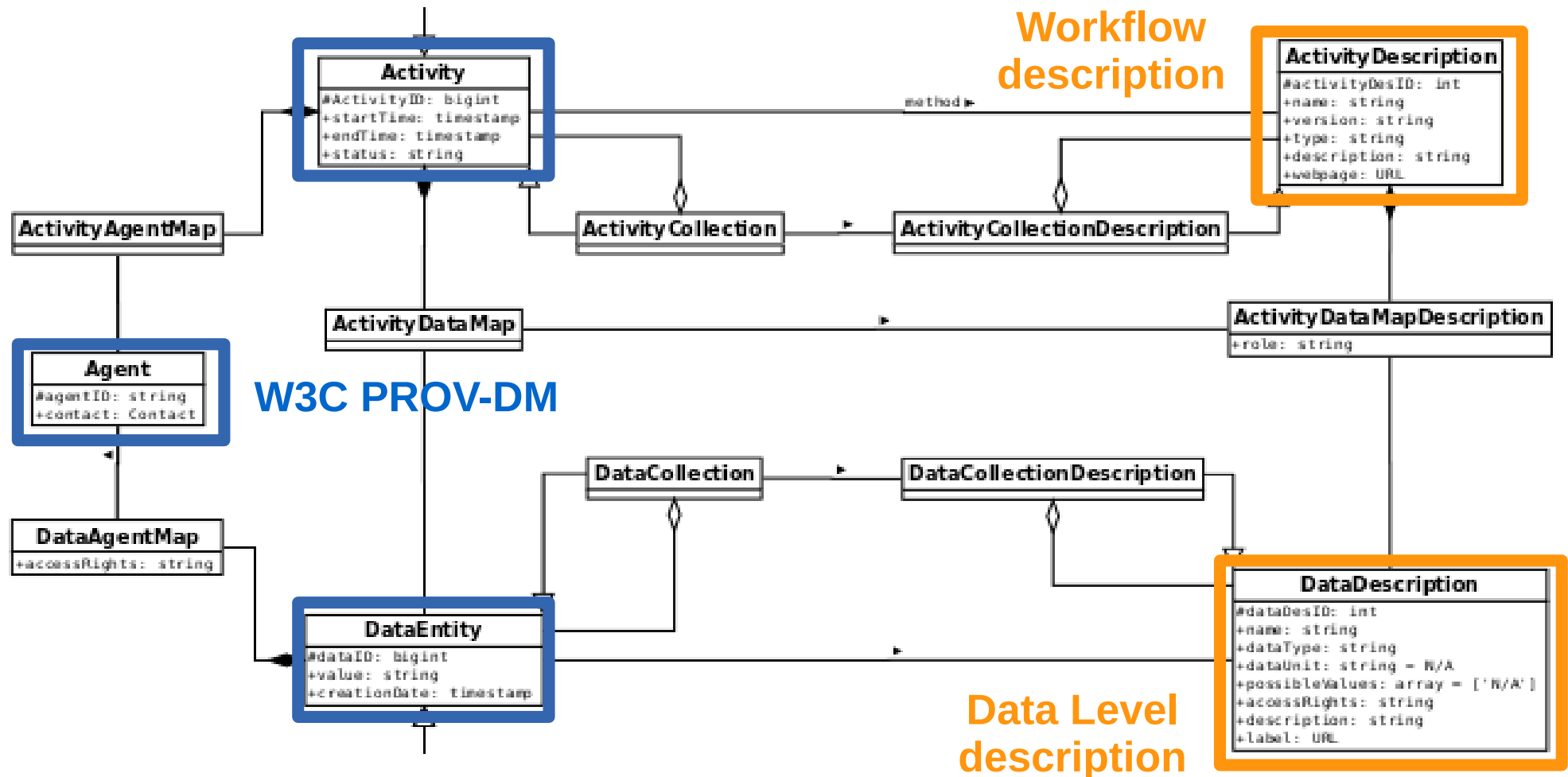


CTA data model – Data Processing



- ◆ Some **CTA specific** Activities and Entities (Run, ObsConfig, ...)
- ◆ Link to **W3C/IVOA Provenance data model** for Data Processing

Provenance at IVOA



<http://wiki.ivoa.net/twiki/bin/view/IVOA/ObservationProvenanceDataModel>

IVOA ObsCore fields

ObsCore mandatory fields:

- ◆ dataproduct_type
- ◆ calib_level
- ◆ target_name
- ◆ obs_collection / obs_id / obs_publisher_did
- ◆ access_url / access_format / access_estsize
- ◆ **Spatial:** s_ra / s_dec / s_fov / s_region / s_resolution
- ◆ **Time:** t_min / t_max / t_exptime / t_resolution
- ◆ **Wavelength:** em_min / em_max / em_res_power
- ◆ facility_name / instrument_name

Extended ObsCore fields for CTA

◆ **Optional ObsCore fields:**

- ◆ **dataprodut_subtype**: show DL0-5?
- ◆ **obs_release_date**
- ◆ **data_rights** (Public/Secure/Proprietary)
- ◆ **s_resolution min, s_resolution max** (as it is dependent on energy)
- ◆ **proposal_id**

◆ **ObsConfig:**

- ◆ **site**: North or South site.
- ◆ **sub_array_name** (or directly in **instrument_name**)
- ◆ **pointing_mode**: parallel, divergent, convergent, custom...
- ◆ **obs_mode**: wobble, scan, on, off
- ◆ **run_type**: flatfield, science, SPE...

◆ **Provenance:**

- ◆ **data_quality**: flag giving information on the data quality
- ◆ **calib_version**: version of the calibration stage of the Pipeline
- ◆ **reco_version**: version of the reconstruction stage of the Pipeline
- ◆ **reco_method**: reconstruction method used to obtain DL2 data
- ◆ **applied_cuts**: selection criteria used to obtain e.g. a DL3 photon event list
- ◆ **spectral_model**: spectral model assumed to obtain spectrum

Data mining use cases for CTA

Use case	Description
Cone Search	Search data available for a given Target
ObsCore search	Search data available 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 available 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 available corresponding to ObsConfig keywords (sub_array_name, pointing_mode, obs_mode ...), e.g.: * search data that include LST * search data for a given target and that do not include the divergent pointing mode
Provenance search	Search data available 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 Data Distiller

<http://voparis-cta-client.obspm.fr>



CTA Data Distiller

🔍 Search Form

⚙ Job List

✕ Sign out user

☒ Cone Search

Target Name

Crab Nebula

Used to query Simbad with Sesame and set RA/Dec.

Source RA (deg)

83.633

Source Dec (deg)

22.514

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

dataprodect_type

Nothing selected

Data product (file content) primary type

dataprodect_level

Nothing selected

DL0-5

CTA Data Distiller

<http://voparis-cta-client.obspm.fr>



Credits: DESY/Milde Science Comm./Exozet

CTA Data Distiller

Search Datasets

Results

Job List

Selected Job

JS9

Authentication: Sign out user

Search

Analyse

Visualisation

Results show/hide query

ADQL query

SELECT * FROM cta.vo_obscore as o WHERE 1 = intersects(o.s_region, circle('ICRS', 83.63308333, 22.0145, 0.001))

Send

ObsCore fields

UWS

	dataproduct_type	obs_collection	obs_id	target_name	s_ra (deg)	s_dec (deg)
☐	eventlist	1	23592	Crab Nebula	82.01333618164062	22.01444435119629
☐	eventlist	1	23559	Crab Nebula	85.25333404541016	22.01444435119629
☐	eventlist	1	23526	Crab Nebula	83.63333129882812	22.51444435119629
☐	eventlist	1	23523	Crab Nebula	83.63333129882812	21.51444435119629
☐	eventlist	3	5003499	CrabNebula	83.28087615966797	21.784133911132812

Showing 1 to 5 of 10 rows 5 records per page

<<

<

1

2

>

>>

SAMP

Interop (SAMP)

Send Result Table

Send Selected Data

Analysis tools

Create Count Map(s)

Extract Spectrum

Plotting tools

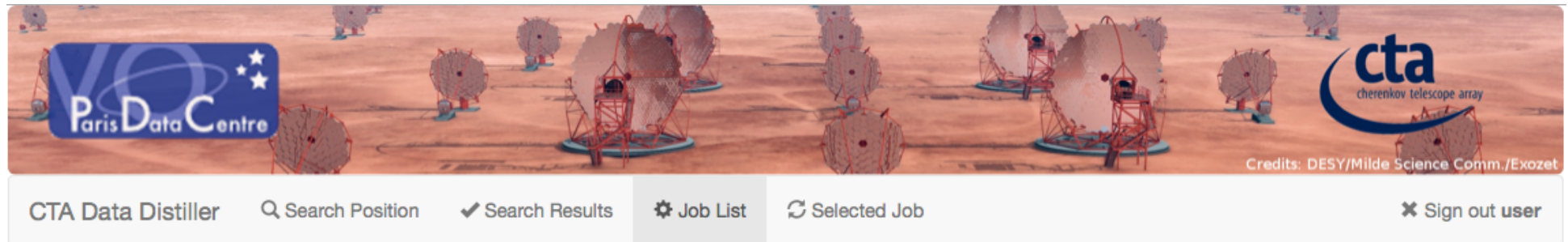
TOPCAT

Aladin

VOSpec

SPLAT

Online processing with UWS (client)



Job List

Refresh Job List Create Test Job

Job list loaded

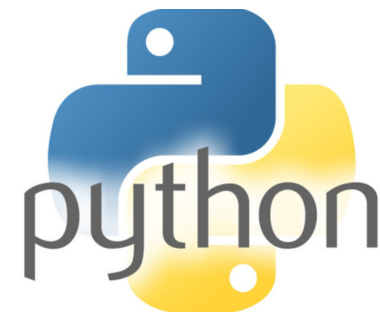
Type	Start Time	Phase	Actions			Control		
ctbin	2014-10-07 21:32:58	ABORTED	Details	Edit	Results	Start	Abort	Delete
ctbin	2014-10-06 17:12:03	COMPLETED	Details	Edit	Results	Start	Abort	Delete
ctbin	2014-10-04 14:05:12	COMPLETED	Details	Edit	Results	Start	Abort	Delete
ctbin	2014-10-03 13:22:46	ABORTED	Details	Edit	Results	Start	Abort	Delete
...								

- ◆ **Asynchronous** management of processes
- ◆ Job sent on a **work cluster**
- ◆ JavaScript library developped at PADC

Online processing with UWS (server)

Main features

- ◆ **IVOA standard**
 - ◆ Universal Worker System (UWS)
- ◆ **REST architecture**
 - ◆ Python micro-framework: bottle.py
- ◆ **Collaborative development**
 - ◆ Git server at PADC (gitolite)
 - ◆ GitHub : <https://github.com/mservillat/uws-server>
- ◆ **Tests and quality**
 - ◆ Unit tests with `unittest` and `webtest`
 - ◆ Activity history with `logging`



Prototype available

<https://voparis-uws-test.obspm.fr>

Test case for Provenance

◆ UWS server

- ◆ Used to run **activities** (jobs) on **entities** (data files)

◆ CTA data processing

- ◆ Test job: DL3 event list --> DL4 image (ctbin)

UWS Server Job Definition Job Manager ✕ Sign out user

Job Definition

Name	ctbin	<button>Load WADL</button>	<button>Get WADL</button>	Job name.																												
Description	CTOOLS command to generate a counts cube for binned maximum likelihood analysis.			Job description.																												
Parameters	<table><tr><td>✕</td><td>inobs</td><td>=</td><td>events.fits</td><td>☒</td><td>xs:anyURI</td><td>▼</td></tr><tr><td colspan="7">Input event list or observation definition file</td></tr><tr><td>✕</td><td>outcube</td><td>=</td><td>cube.fits</td><td>☐</td><td>xs:string</td><td>▼</td></tr><tr><td colspan="7">Output counts map or observation definition file</td></tr></table>			✕	inobs	=	events.fits	☒	xs:anyURI	▼	Input event list or observation definition file							✕	outcube	=	cube.fits	☐	xs:string	▼	Output counts map or observation definition file							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).
✕	inobs	=	events.fits	☒	xs:anyURI	▼																										
Input event list or observation definition file																																
✕	outcube	=	cube.fits	☐	xs:string	▼																										
Output counts map or observation definition file																																

voprov package

- ◆ Based on **prov** python package
- ◆ UWS server on job completion
 - ◆ **generates** a ProvDocument
 - ◆ Based on the **job description** (= ActivityDescription)
 - ◆ Using the job execution properties/parameters
 - ◆ **adds a result** “provenance.xml” to the jobs

Job Description

Back to job list

Type	Start Time	Destruction Time	Phase	Details			Control		
ctbin	2016-03-13 23:44:46	2016-04-12 23:43:59	COMPLETED	Properties	Parameters	Results	Start	Abort	Delete

➤ Job Properties

➤ Job Parameters

▼ Job Results

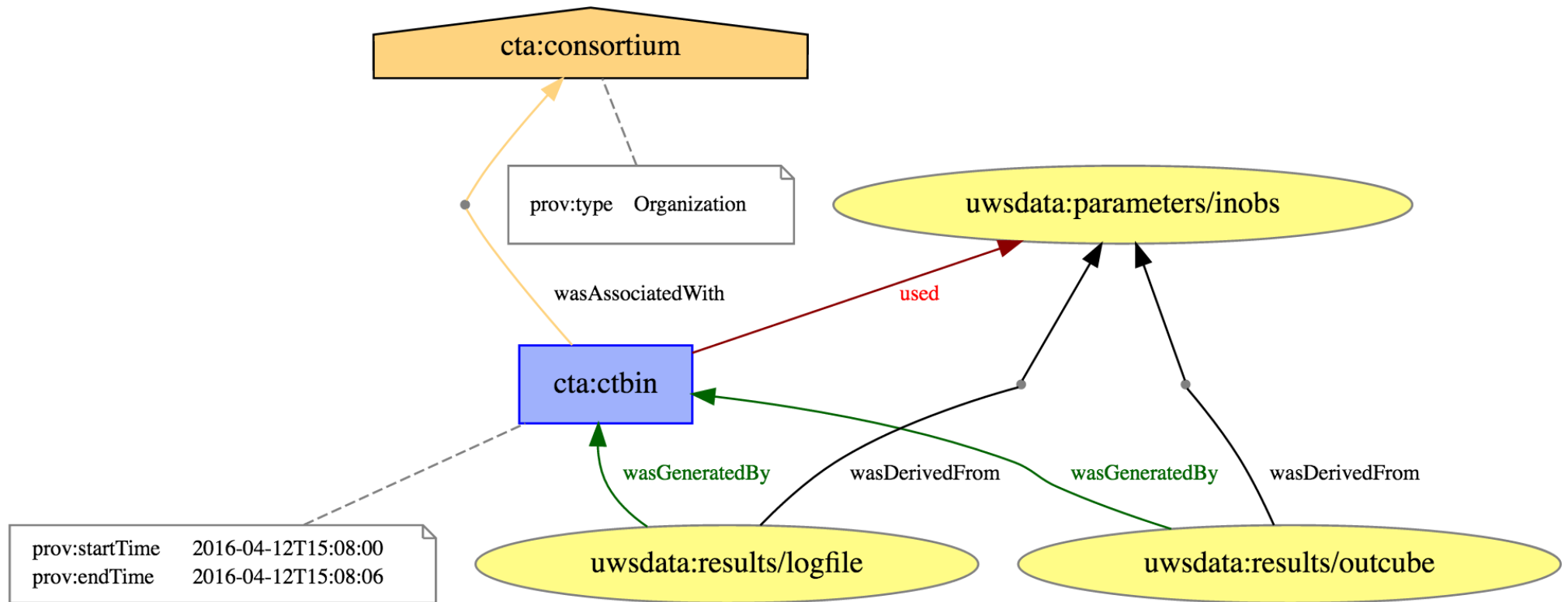
provenance.xml: https://voparis-uws-test.obspm.fr/get_result_file/bcb19a1d-4ca5-45fc-b9ad-5432632e8572/provenance/provenance.xml

PROV output (XML and JSON)

```
<prov:document xmlns:ctadata="ivo://vopdc.obspm/cta#" xmlns:ctajob
  <prov:activity prov:id="ctajobs:ctbin">
    <prov:startTime> 2016-03-13T23:44:46 </prov:startTime>
    <prov:endTime> 2016-03-13T23:44:56 </prov:endTime>
  </prov:activity>
  <prov:agent prov:id="cta:consortium">
    <prov:type xsi:type="xsd:string"> Organization </prov:type>
  </prov:agent>
  <prov:wasAssociatedWith>
    <prov:activity prov:ref="ctajobs:ctbin" />
    <prov:agent prov:ref="cta:consortium" />
  </prov:wasAssociatedWith>
  <prov:entity prov:id="uwsdata:parameters/inobs" />
  <prov:used>
    <prov:activity prov:ref="ctajobs:ctbin" />
    <prov:entity prov:ref="uwsdata:parameters/inobs" />
  </prov:used>
  <prov:entity prov:id="uwsdata:results/outcube" />
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="uwsdata:results/outcube" />
    <prov:activity prov:ref="ctajobs:ctbin" />
  </prov:wasGeneratedBy>
  <prov:wasDerivedFrom>
    <prov:generatedEntity prov:ref="uwsdata:results/outcube" />
    <prov:usedEntity prov:ref="uwsdata:parameters/inobs" />
  </prov:wasDerivedFrom>
  <prov:entity prov:id="uwsdata:results/logfile" />
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="uwsdata:results/logfile" />
    <prov:activity prov:ref="ctajobs:ctbin" />
  </prov:wasGeneratedBy>
  <prov:wasDerivedFrom>
    <prov:generatedEntity prov:ref="uwsdata:results/logfile" />
    <prov:usedEntity prov:ref="uwsdata:parameters/inobs" />
  </prov:wasDerivedFrom>
</prov:document>
```

```
{
  - wasAssociatedWith: {
    - _:id1: {
      prov:agent: "cta:consortium",
      prov:activity: "cta:anactools_v1.1"
    }
  },
  - agent: {
    - cta:consortium: {
      prov:type: "Organization"
    }
  },
  - entity: {
    uwsdata:results/fit_results: { },
    uwsdata:results/configfile: { },
    uwsdata:results/butterfly: { },
    uwsdata:results/spectrum_plot: { },
    uwsdata:results/spectrum: { }
  },
  - prefix: {
    uwsdata: "https://voparis-uws-test.obspm.fr/rest
    cta: "http://www.cta-observatory.org#",
    voprov: "http://www.ivoa.net/ns/voprov#"
  },
  - activity: {
    - cta:anactools_v1.1: {
      prov:startTime: "2016-04-07T00:26:00",
      prov:endTime: "2016-04-07T00:27:15"
    }
  },
  - wasGeneratedBy: {
    - _:id5: {
      prov:entity: "uwsdata:results/butterfly",
      prov:activity: "cta:anactools_v1.1"
    },
    - _:id4: {
      prov:entity: "uwsdata:results/fit_results",
      prov:activity: "cta:anactools_v1.1"
    },
    - ...
  },
  - ...
}
```

PROV output (SVG)



Requirements to store Provenance

- ◆ Structured **job description** (=ActivityDescription)
 - ◆ UWS pattern too limited (job description language?)
 - ◆ We use WADL, but also limited
- ◆ Locate **input entities**
 - ◆ Not specified in UWS pattern
 - ◆ Parameters “inXXX” or “input.XXX”, files, URLs
- ◆ Store **parameter list** as an input entity
 - ◆ With description (units, UCDs, min/max, choices)
 - ◆ Queryable metadata?
- ◆ Use **persistent identifiers** (obs_publisher_did?)
- ◆ Use **namespaces**

Manipulating Provenance

Storing Provenance:

- ◆ Write to files 
- ◆ Store with data product (header, ...) 
- ◆ Store in a database (using data model) 

Retrieving Provenance:

- ◆ Request Provenance path
 - ◆ From files 
 - ◆ From database (API) 
- ◆ Search data products based on Provenance 
 - ◆ A given Activity was performed (with given version)
 - ◆ A given input parameter was set to...

Provenance Access Layer

- ◆ **Specific service** to access Provenance
 - ◆ A **DataLink** could point to it from resource
- ◆ **From ObsTAP to ProvTAP?**
 - ◆ ObsCore fields are all known **a priori**
 - ◆ Provenance fields are **specific** to:
 - ◆ A **project** (CTA, ...), i.e. a given pipeline
 - ◆ A **data product** (different for CTA DL3 or DL4)
 - ◆ Would first need to request the Provenance fields and their possible values
 - ◆ Then create a query based on those fields...

Next steps

- ◆ Use the data model to define a **database**
- ◆ **I/O package** for this database
 - ◆ Using descriptions: activity/data/parameters
 - ◆ Based on prov?
- ◆ Could be included in the CTA framework
 - ◆ **ctapipe** project in Python
 - ◆ Fill the Provenance info from DL0 to DL3
- ◆ **Query** systems
 - ◆ PROV-AQ, IVOA SSA/TAP/..., files, headers...