Implementing the Provenance data model

Mathieu Servillat

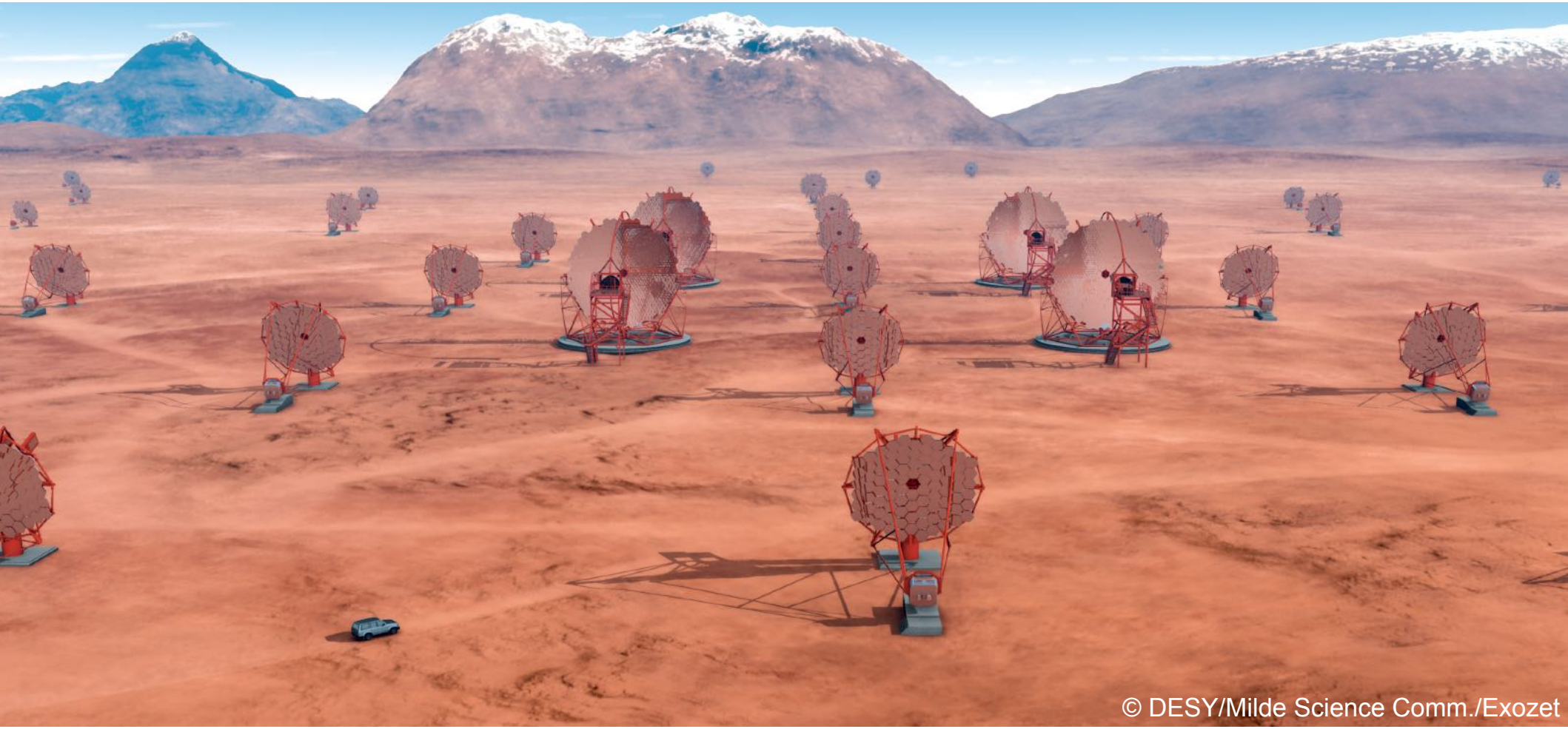
Observatoire de Paris
Paris Astronomical Data Centre

Journées Action Spécifique Observatoire Virtuel (ASOV)

cherenkov telescope array

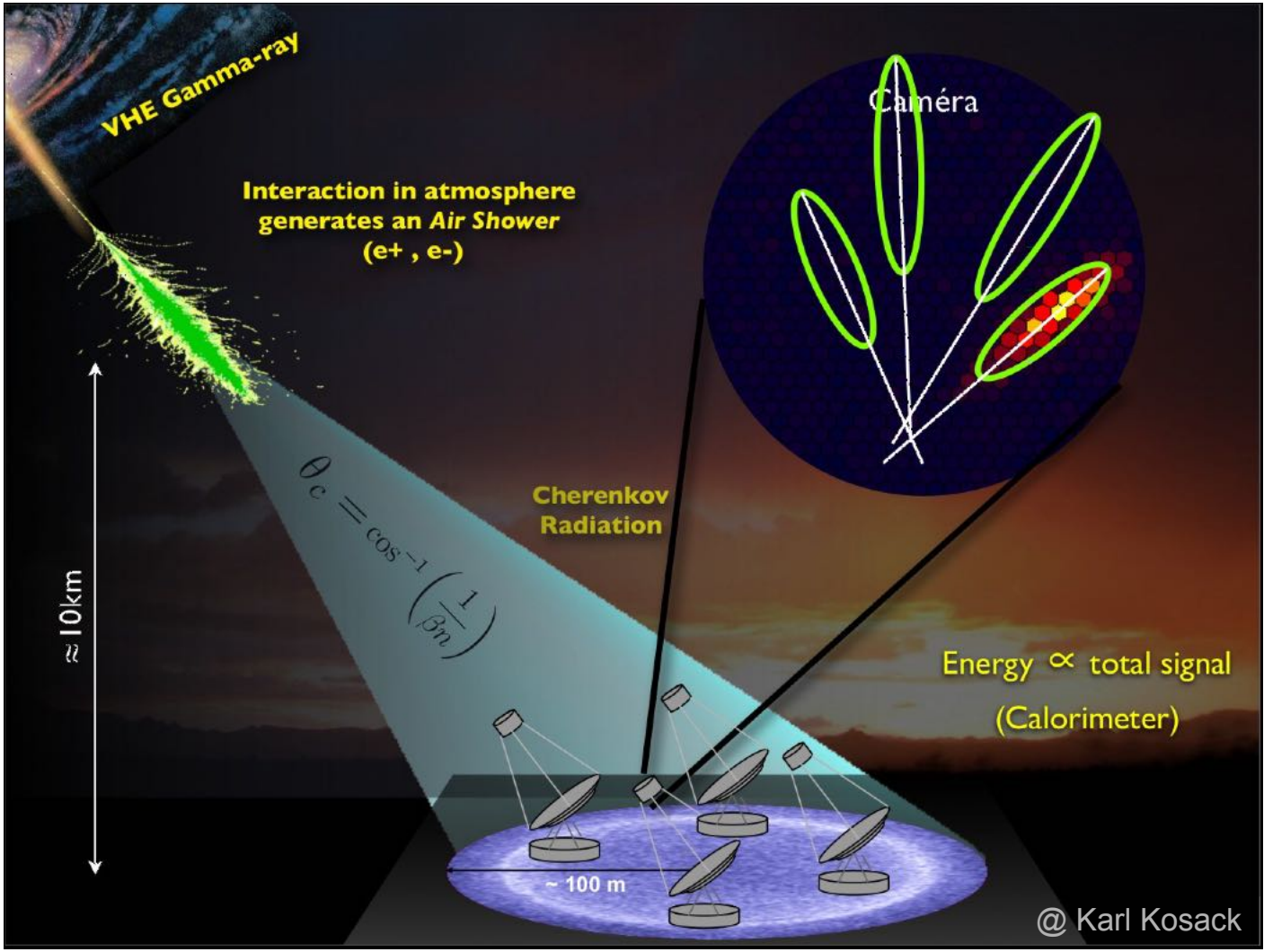
Observatory

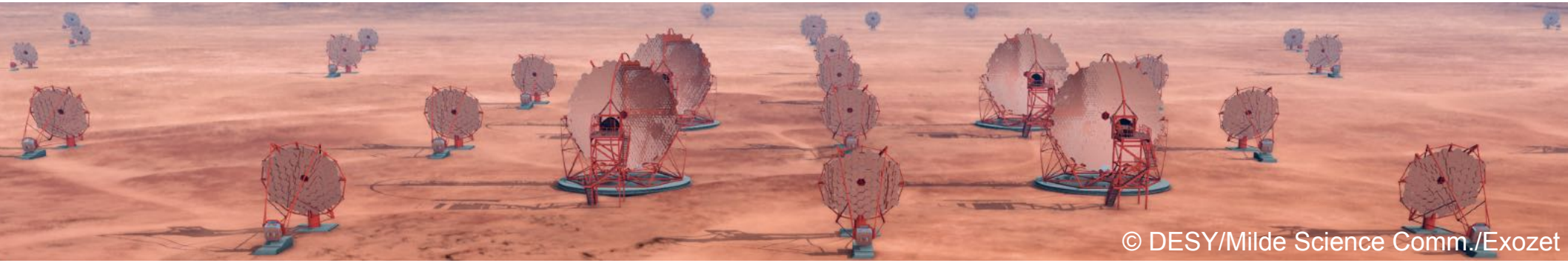
- Two arrays of **100 (South)** et **20 (North)** Cherenkov telescopes (4, 12 et 24 m in diametre)
- July 2015: site selection, Chile (ESO) and La Palma
- 2016: pre-production phase
- 2018-2013: production phase
- Observatory open to the Astronomy community



cherenkov telescope array

- **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

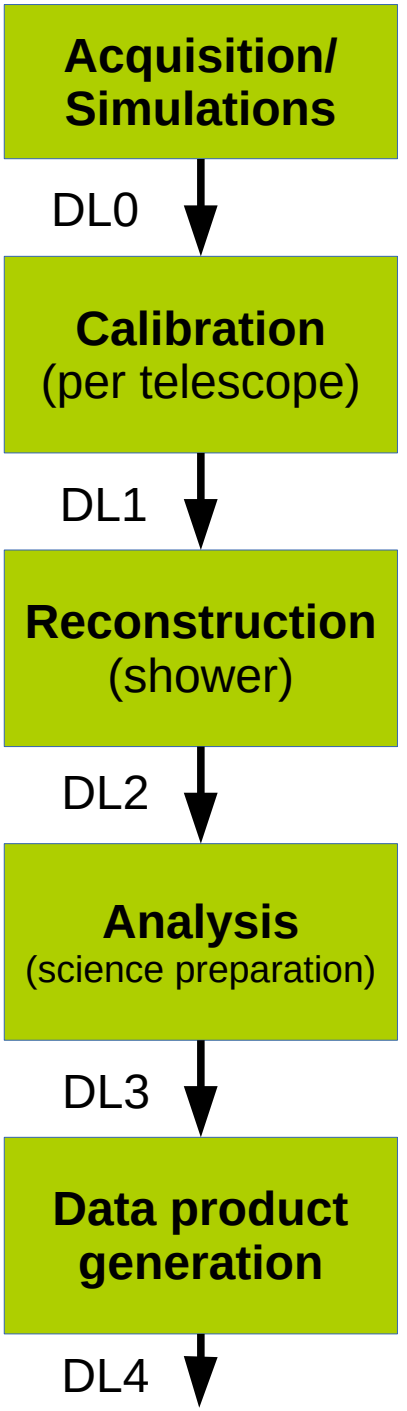




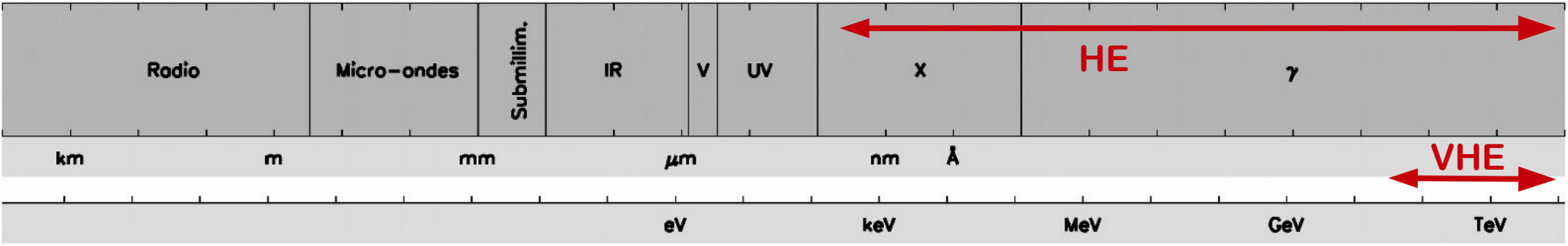
Data levels and workflow

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}

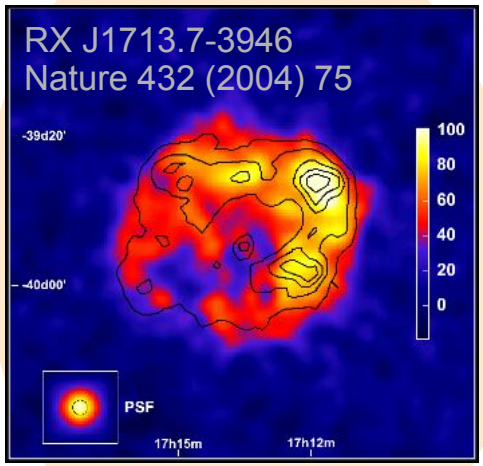
published



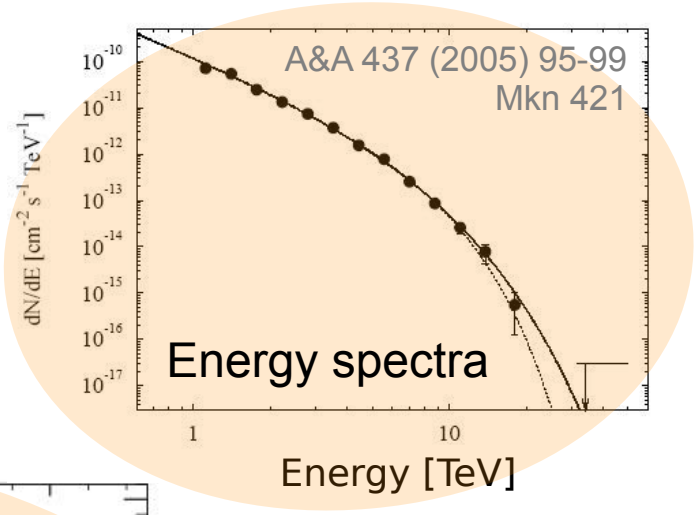
Very high energy (VHE) data

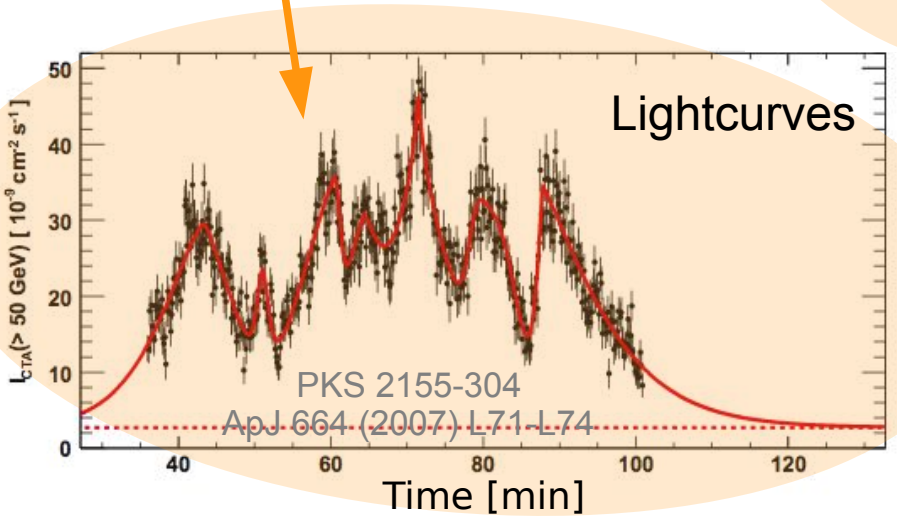


- Several orders of magnitude
- Photon counting
- Low count statistics, high background
- **Event lists**
  (coordinates, time, energy)

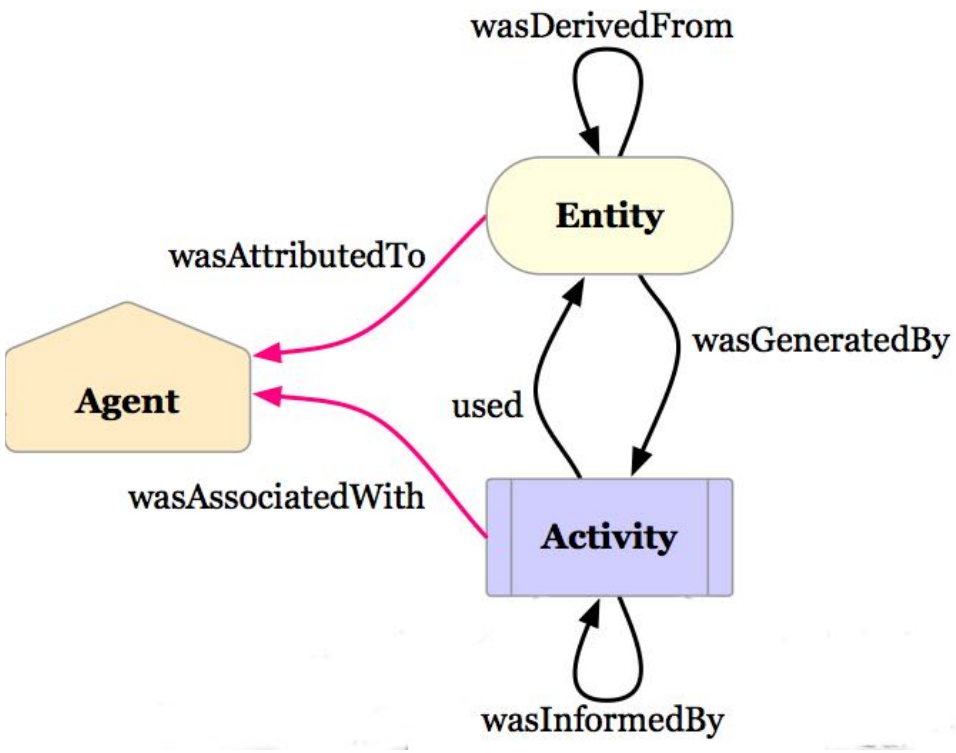


Images





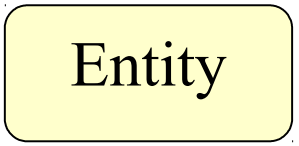
Provenance at W3C



- Makes explicit:
  - Processing steps
  - Chain of dependencies
  - Responsibilities
- Useful for all execution sequence of tasks, workflow, reduction pipeline, analysis workflow, etc.
- Applies for both acquisition and processing steps

www.w3.org/TR/prov-overview/

In the Astronomy context

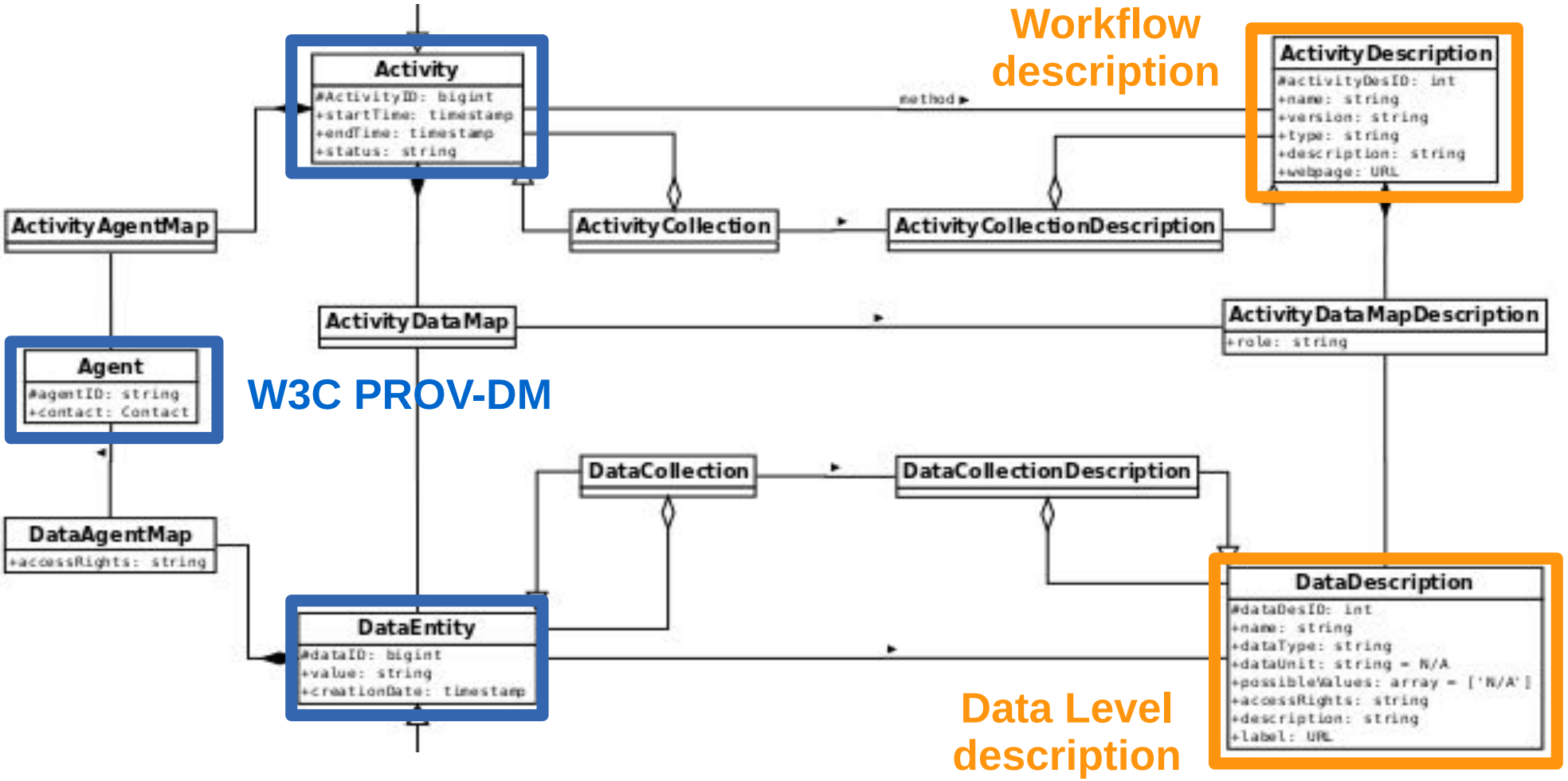


- Data products (files), ancillary data (calibration,instrumental response, etc.), processing parameter files

- Data acquisition, data processing (mosaicing, regridding, fusion, calibration, transformation, ...)

- Telescope astronomer, pipeline operator, principal investigator, Consortium, ...

Provenance at IVOA



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

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
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
dataproduct_type	Nothing selected	Data product (file content) primary type
dataproduct_level	Nothing selected	DL0-5

CTA Data Distiller http://voparis-cta-client.obspm.fr

CTA Data Distiller | Search Datasets | Results | Job List | Selected Job | JS9 | Authentication: Sign out **user**

Search | Analyse | Visualisation

SAMP

Interop (SAMP)

Send Result Table

Results

show/hide query

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

ADQL query

Send

ObsCore fields

Search

UWS

Analysis tools

Create Count Map(s)

	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 > >>

Plotting tools

TOPCAT

Aladin

VOSpec

SPLAT

Online processing with UWS (client)

CTA Data Distiller | Search Position | Search Results | Job List | Selected Job | Sign out **user**

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

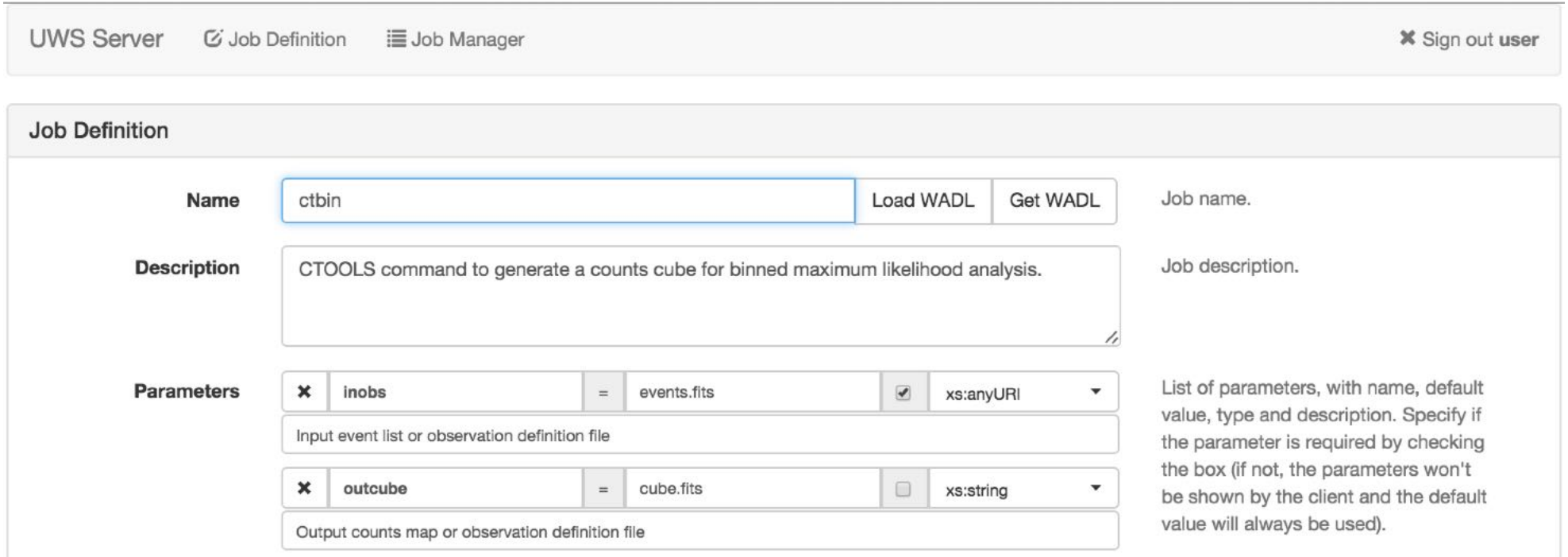
- IVOA **standard**
  - Universal Worker System (UWS)
- **REST** architecture
  - Python micro-framework: bottle.py
- **Collaborative** development
  - Git server at PADC (gitolite)
  - GitHub : https://github.com/mservillat/uws-server
- **Tests** and **quality**
  - Unit tests with `unittest` and `webtest`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

Prototype available

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

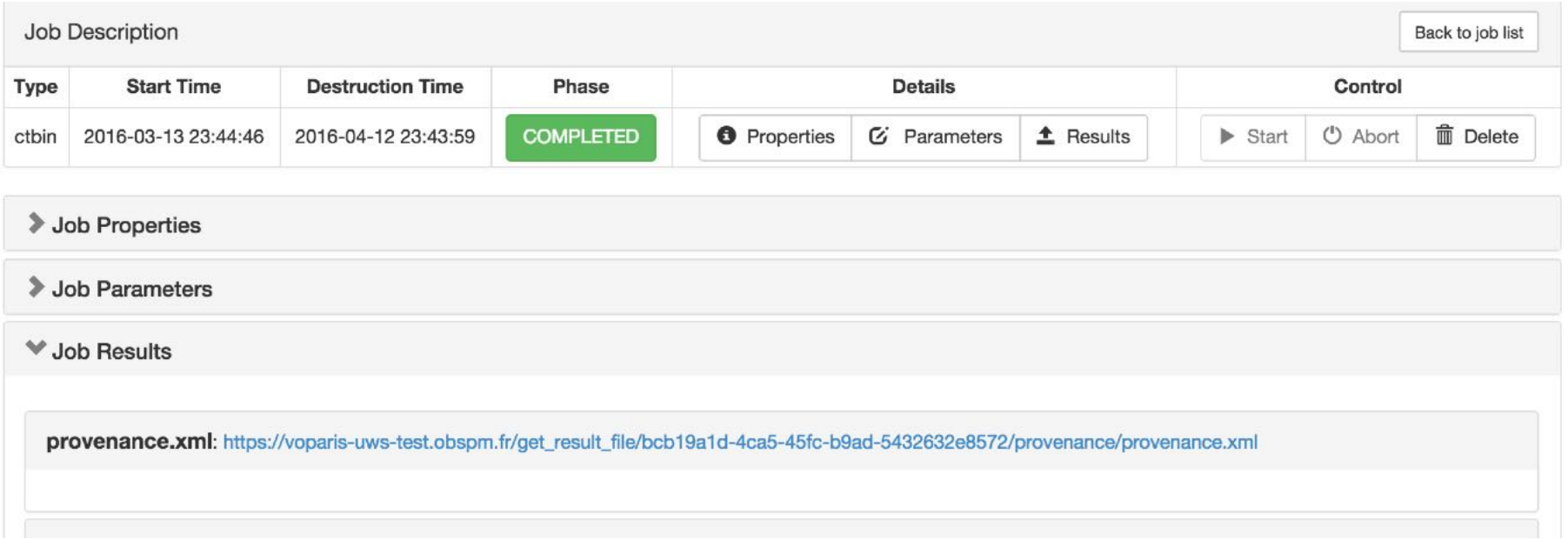
Use case for Provenance

- **UWS server**
  - Used to run **activities** (jobs) on **entities** (data files)
- **CTA data processing**
  - Test job: DL3 event list --> DL4 image (ctbin)



voprov package

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



PROV-XML output

- Standard description
- Can be translated
  - PROV-N
  - JSON
  - SVG

- Still needs:
  - prov attributes
  - VOprov attributes?
  - Relevant identifiers
  - How to store parameters?
  - UWS job description

```
<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>
```

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...