CTA use case for Provenance
(Cherenkov Telescope Array)

Mathieu Servillat

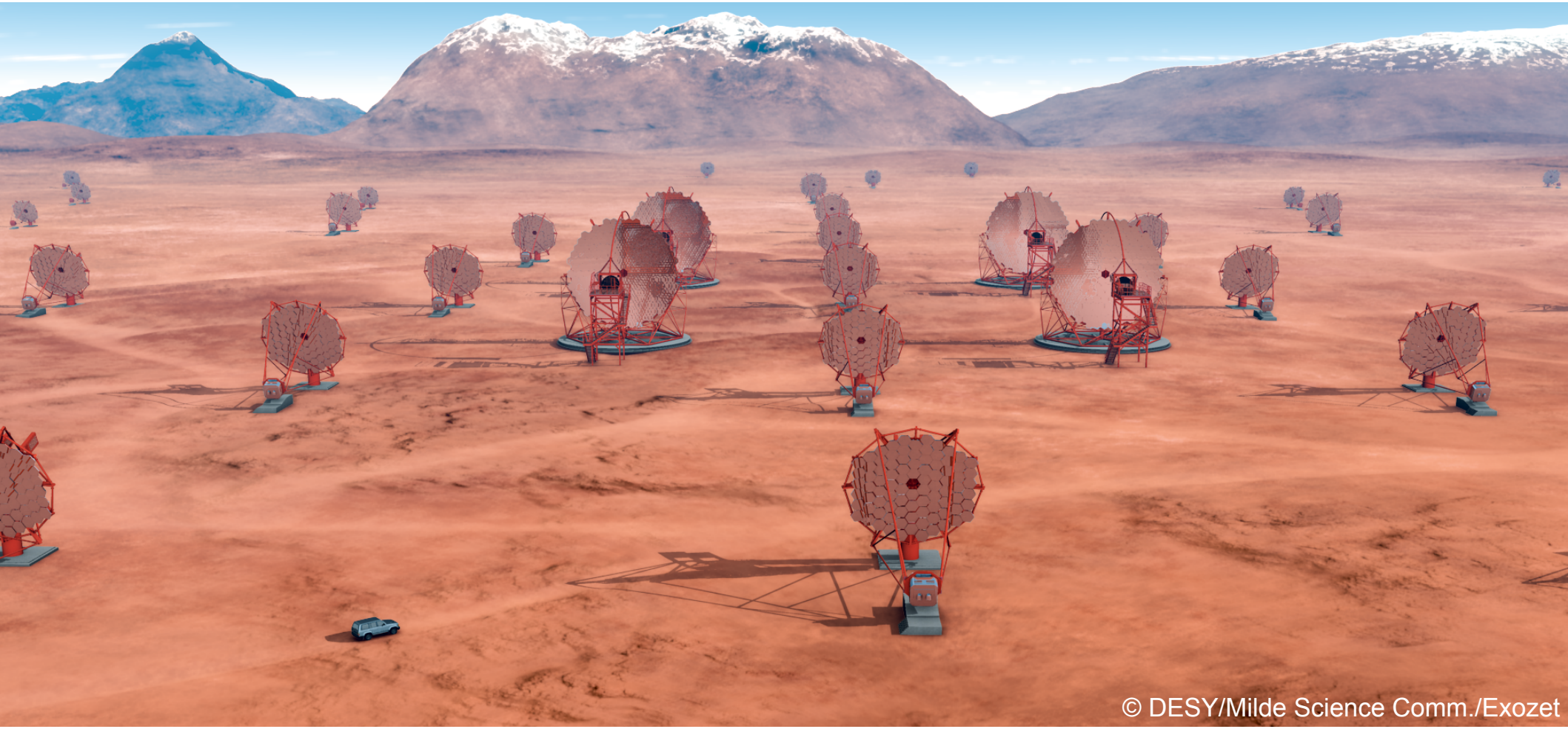
Laboratoire Univers et Théories
Observatoire de Paris
PSL Research University

IVOA Cape Town meeting

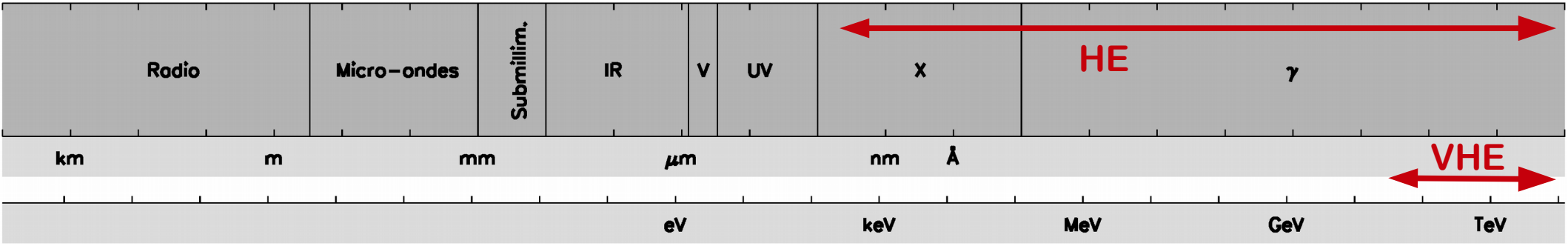
cherenkov telescope array

Observatory

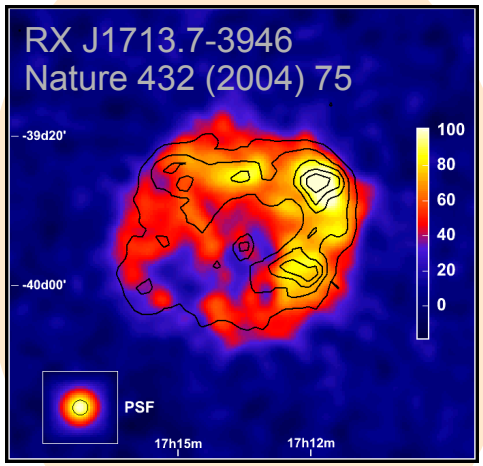
- Two arrays of **100 (South)** et **20 (North)** Cherenkov telescopes (4, 12 et 24 m in diametre)
- July 2015: site selection, Chile (ESO) and La Palma
- 2016: pre-production phase
- 2018-2013: production phase
- Observatory **open** to the Astronomy community



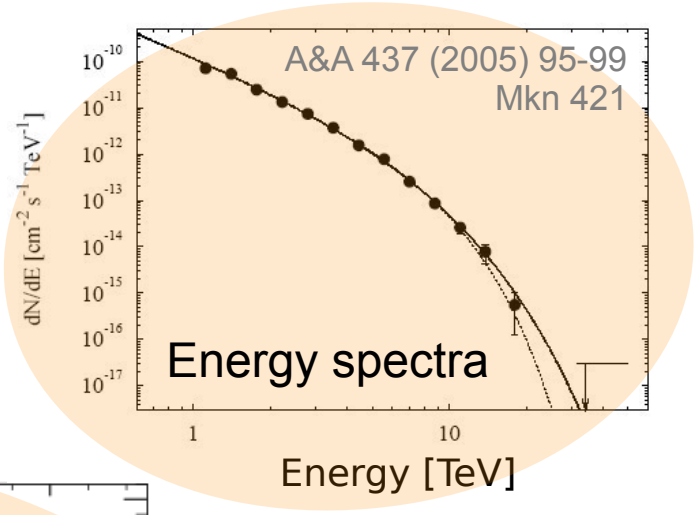
Very high energy (VHE) data

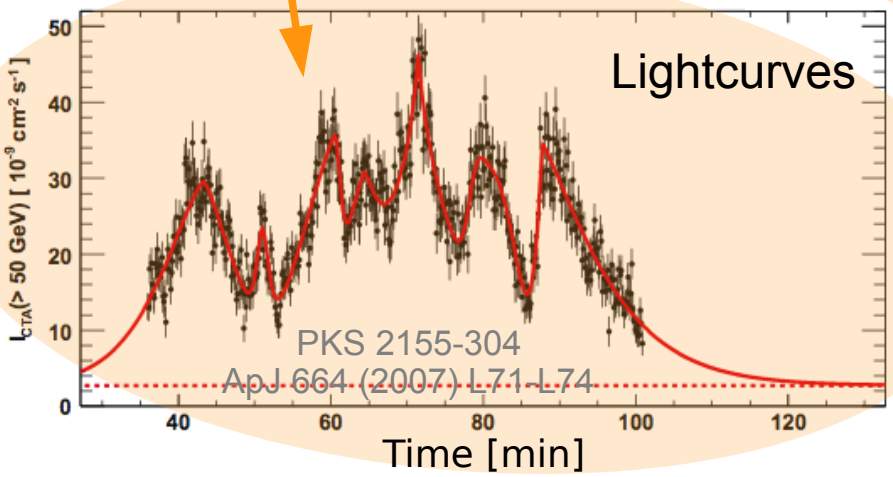


- Several orders of magnitude
- Photon counting
- Low count statistics, high background
- **Event lists** (coordinates, time, energy)



Images

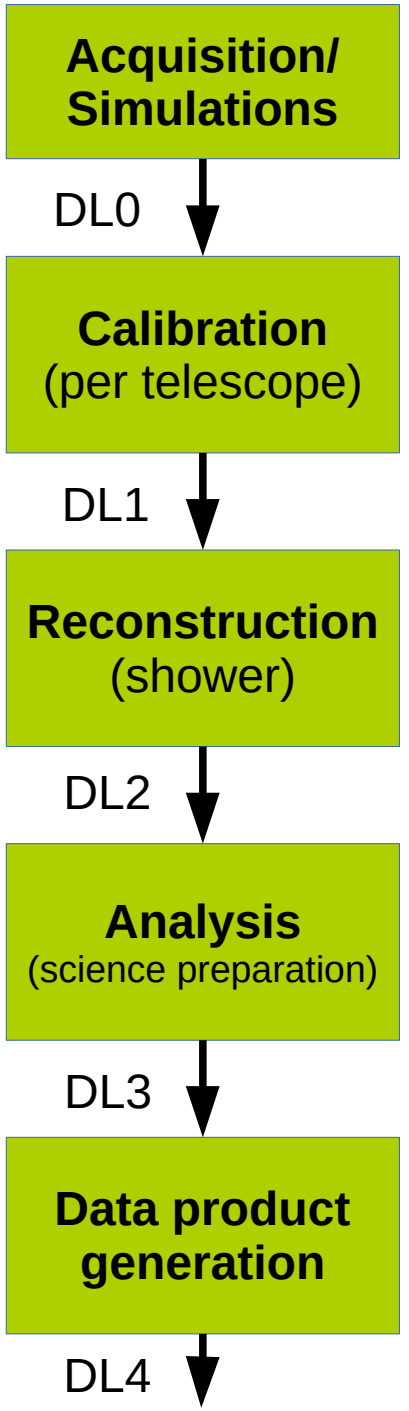




Data levels and workflow

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



Mapping IVOA ObsCore fields

ObsCore mandatory fields:

- dataproduct_type (event, image, spectrum, timeseries)
- calib_level (1: instrument data, 2: calibrated data **vs**. DL0-5)
- target_name
- obs_collection / obs_id / obs_publisher_did (granularity?)
- access_url / access_format / access_estsize
- facility_name / instrument_name (array layout, sub-arrays?)
- **Spatial:**
  s_ra / s_dec / s_fov / s_region / s_resolution (energy dependent!)
- **Time:**
  t_min / t_max / t_exptime / t_resolution
- **Wavelength:**
  em_min / em_max / em_res_power (from TeV to meters...)

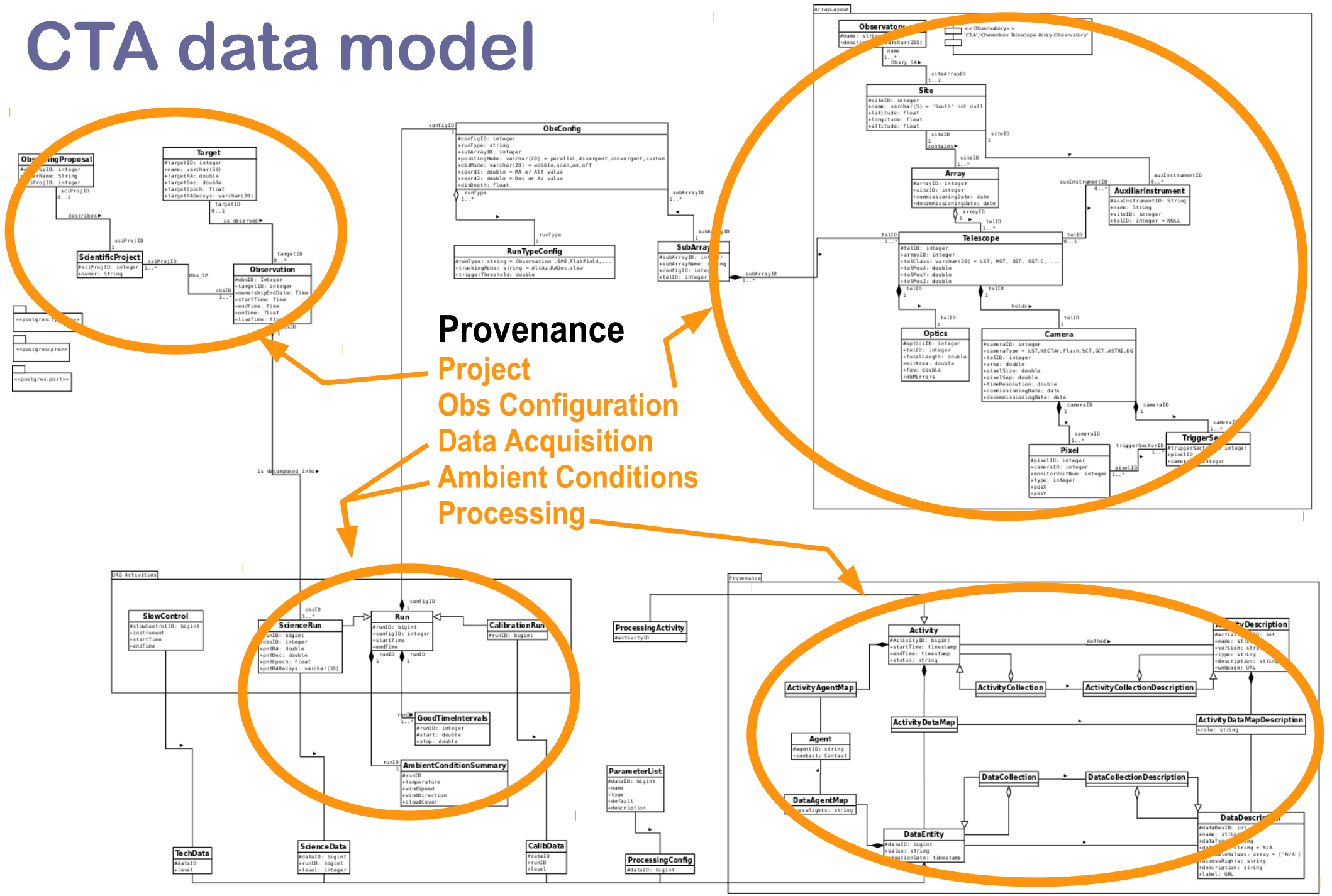
Extended ObsCore fields for CTA

- **Optional ObsCore fields:**
  - **dataproduct_subtype**: show DL0-5?
  - **data_rights** (Public/Secure/Proprietary)
  - **obs_release_date**
  - **s_resolution min**, **s_resolution max** (as it is energy dependent)
  - **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

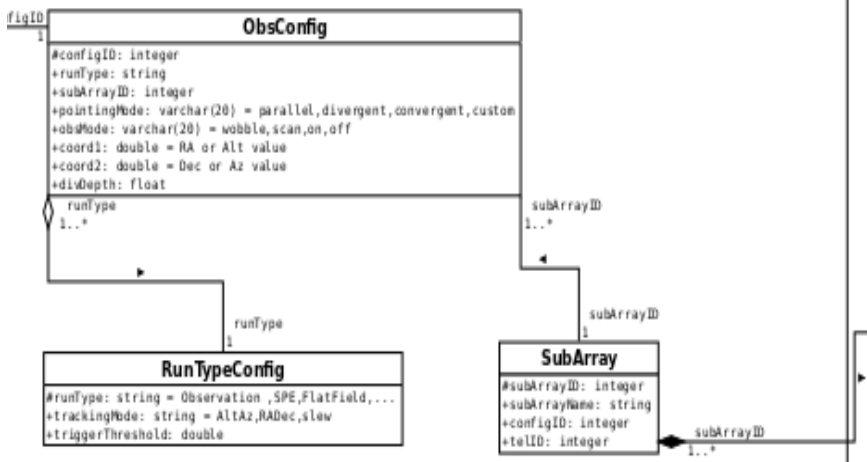
Data mining use cases for CTA

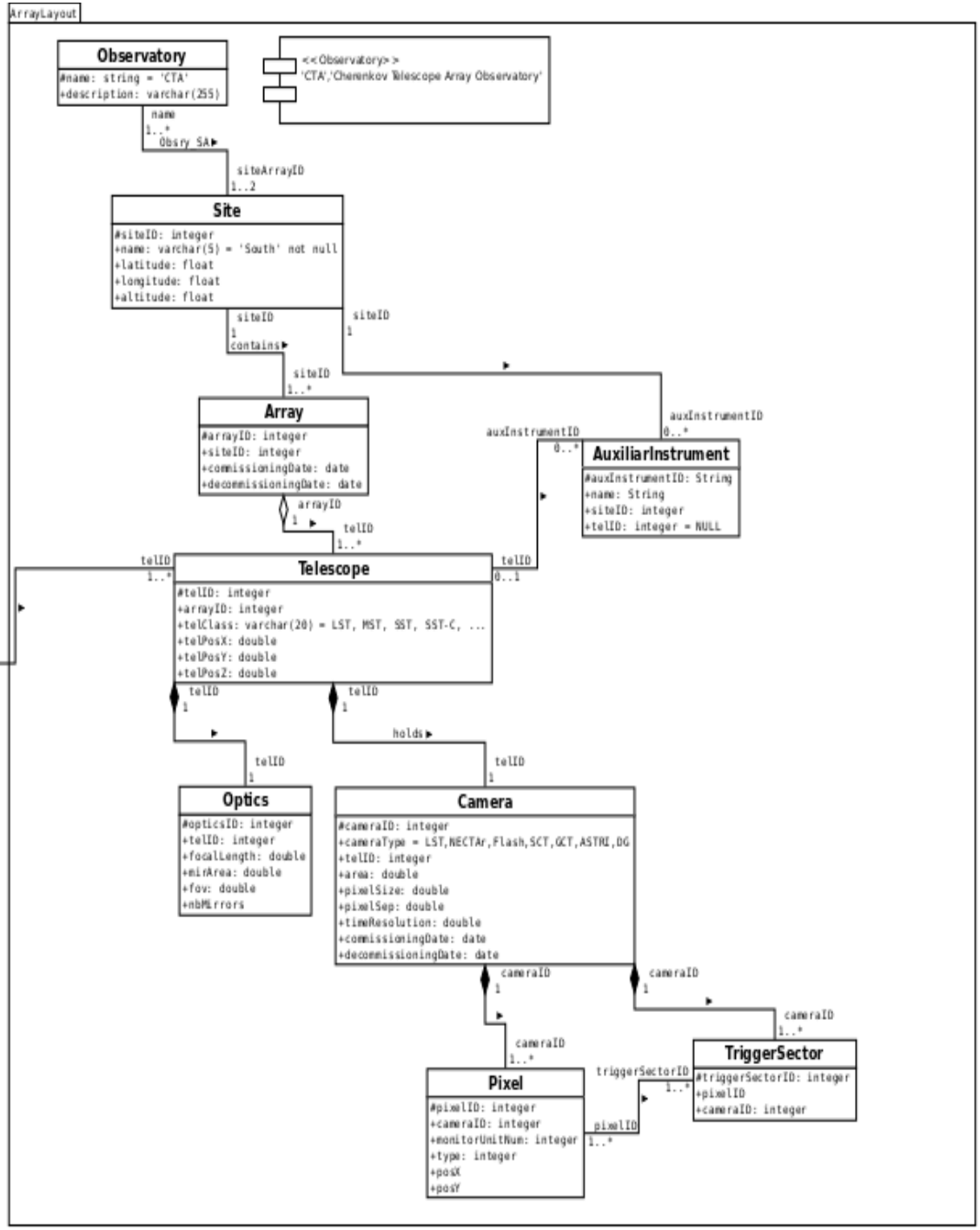
Use case	Description
Cone Search	**Search data available for a given Target**
ObsCore search	**Search data available corresponding to ObsCore keywords** (target_name, time interval, ...), e.g.: • search data for a given **target** at a given **time** • search data in a given **region of the sky** • search 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 the **Large Size Telescopes** (LSTs) • search data for a given target, 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 model



CTA data model ObsConfig





Examples :

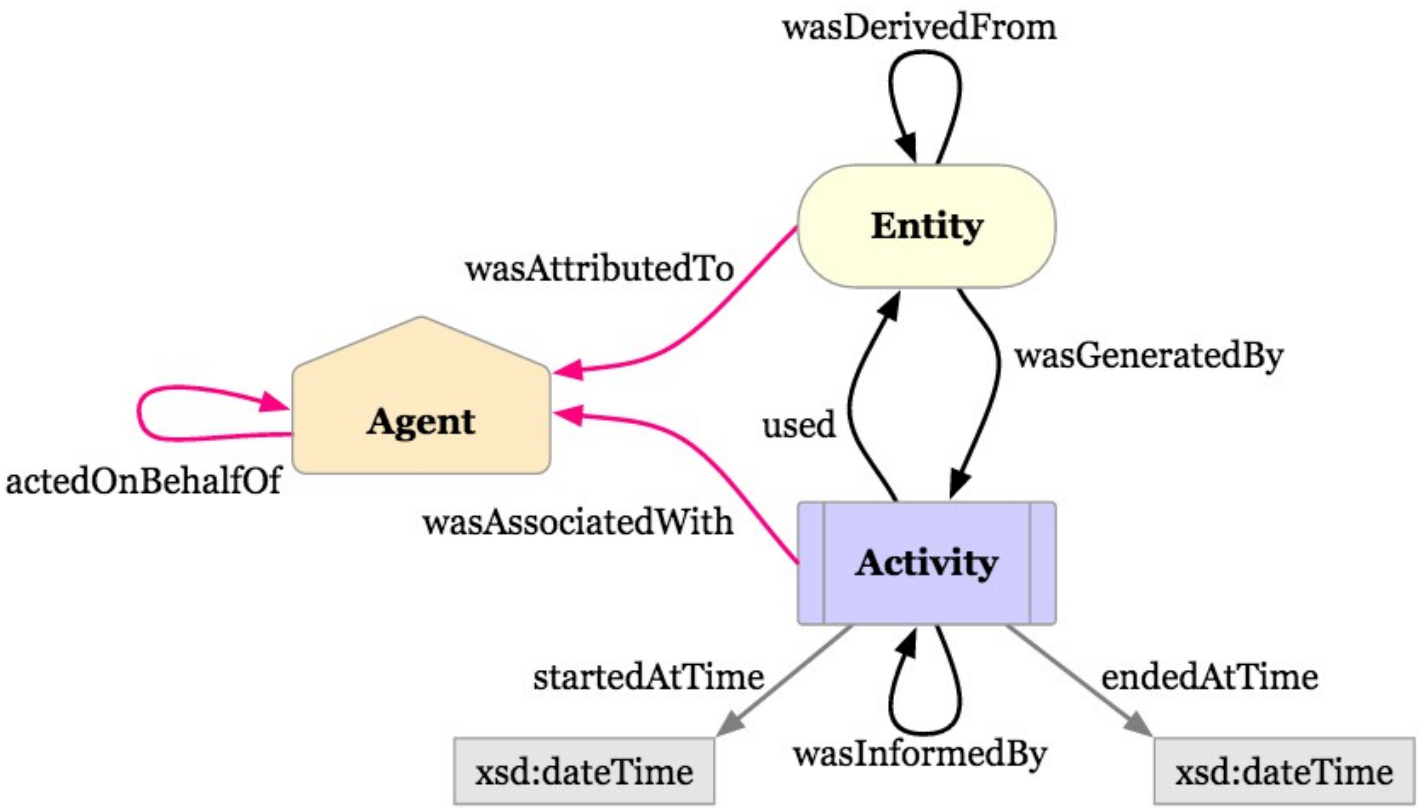
Number of telescopes involved

Field of view

Pointing direction

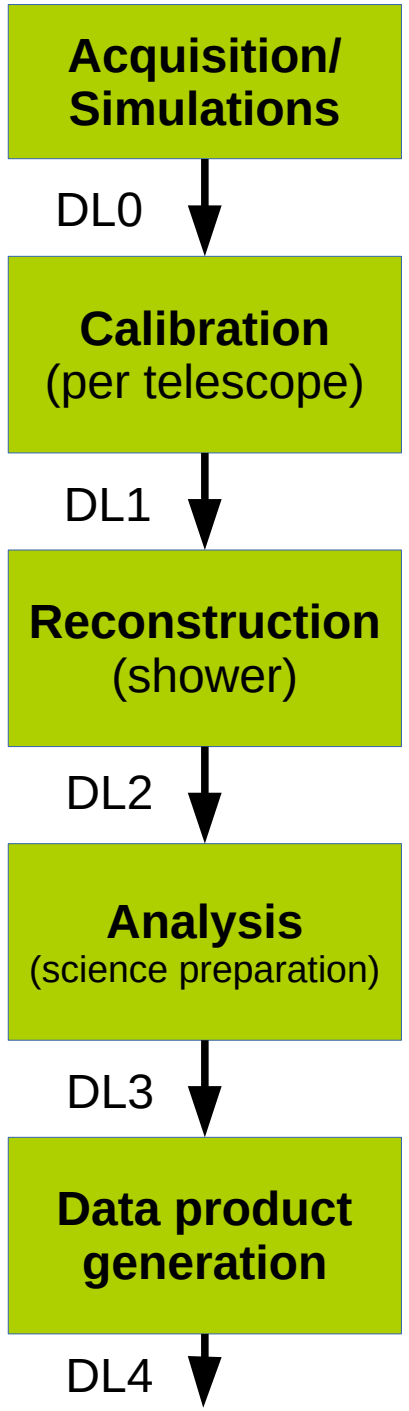
...

CTA data model – Pipeline

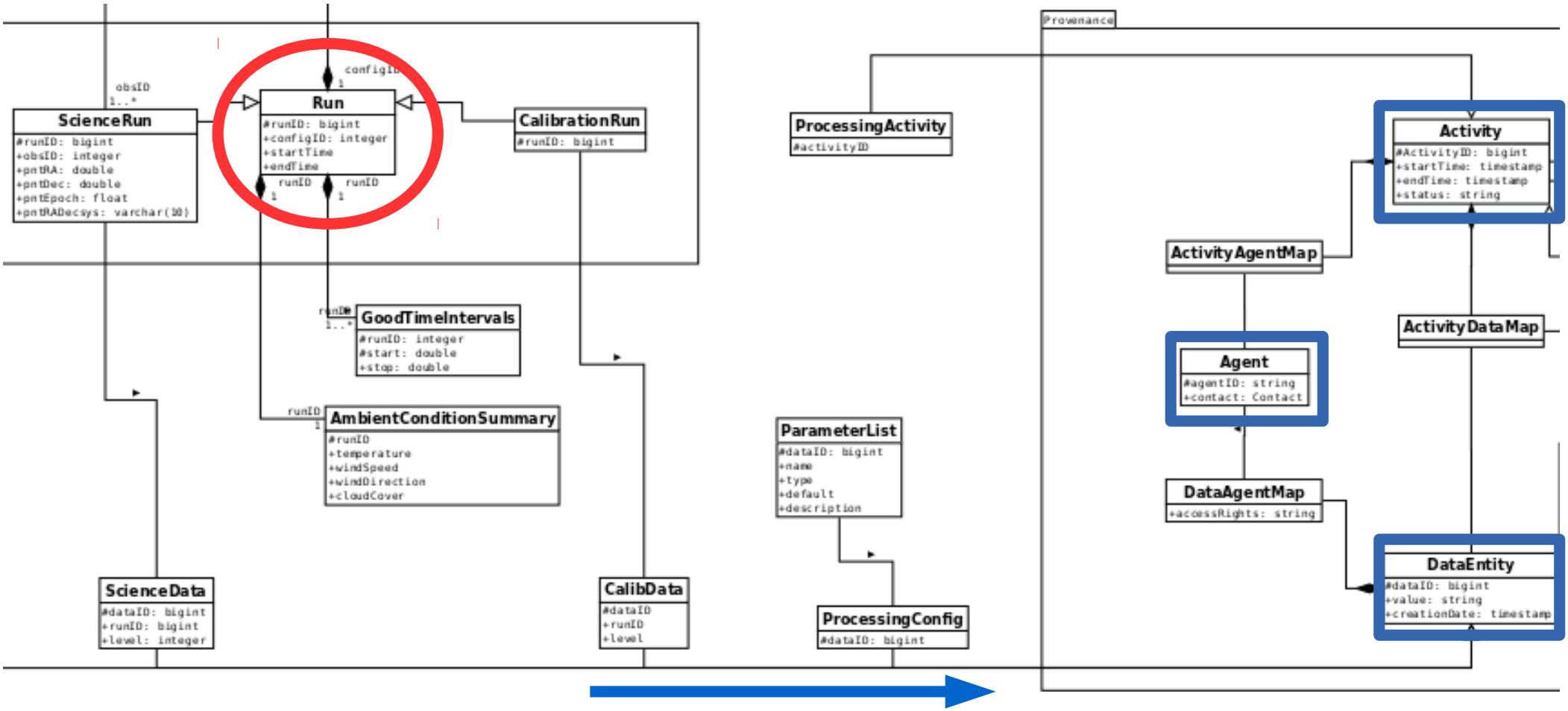


W3C PROV Ontology

https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/



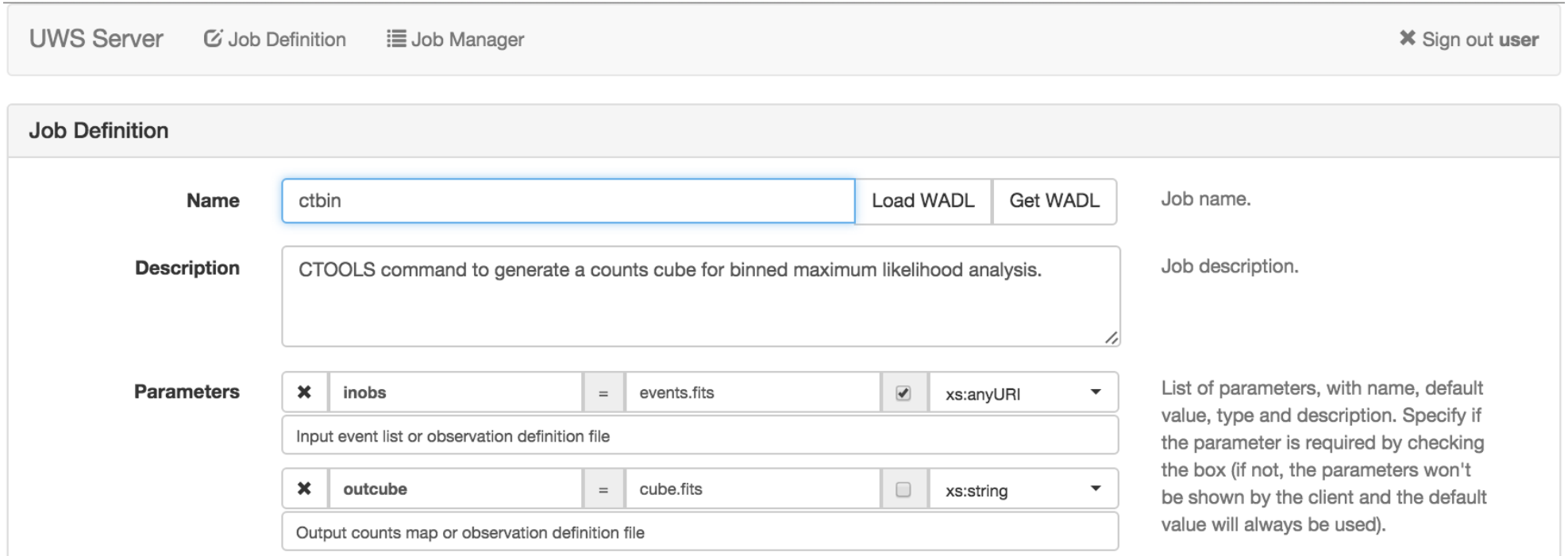
CTA data model – Data Processing



- Some CTA specific Activities and Entities (Run, ObsConfig, ...)
- Link to W3C/IVOA Provenance data model for Data Processing
- ProcessingConfig? List of parameters?

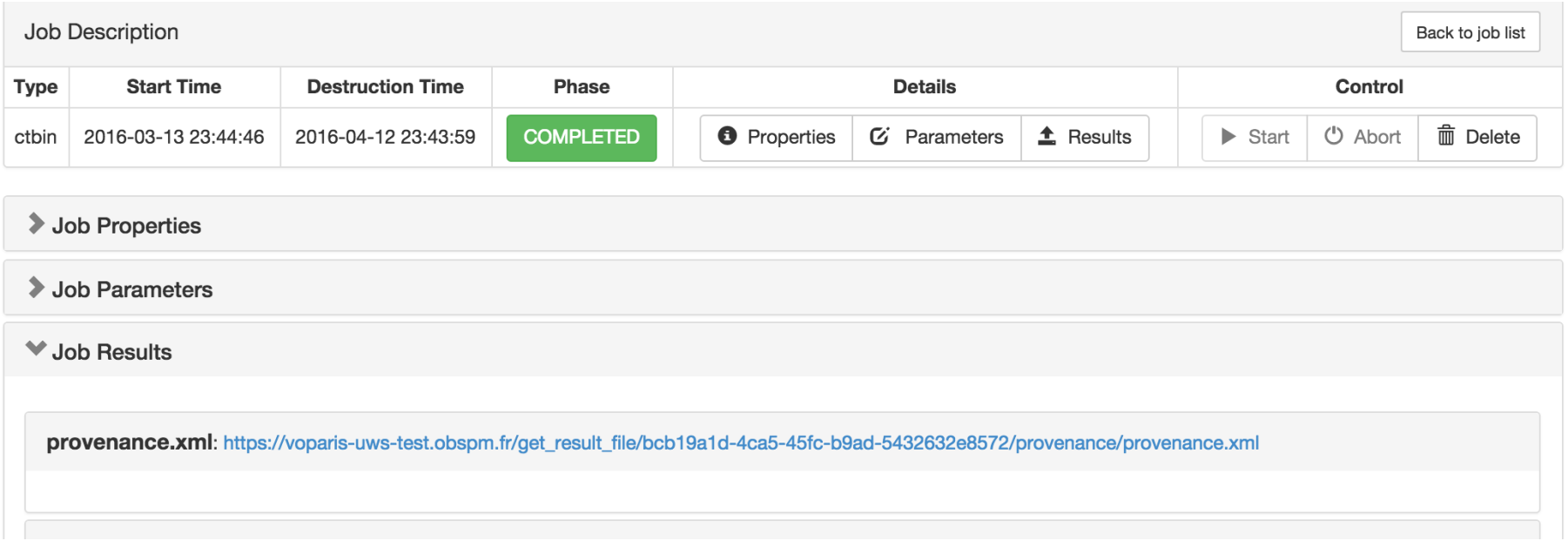
Test case for Provenance

- **UWS server**
  - OPUS: Observatoire de Paris UWS Server (→ GWS2 session)
    https://github.com/ParisAstronomicalDataCentre/OPUS
  - 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
  https://github.com/trungdong/prov
- On UWS job completion, the server will:
  - **generate** a ProvDocument
    - Based on the **job description** (= ActivityDescription)
    - Using the job execution properties/parameters
  - **add results** provxml/provjson/provsvg to the job

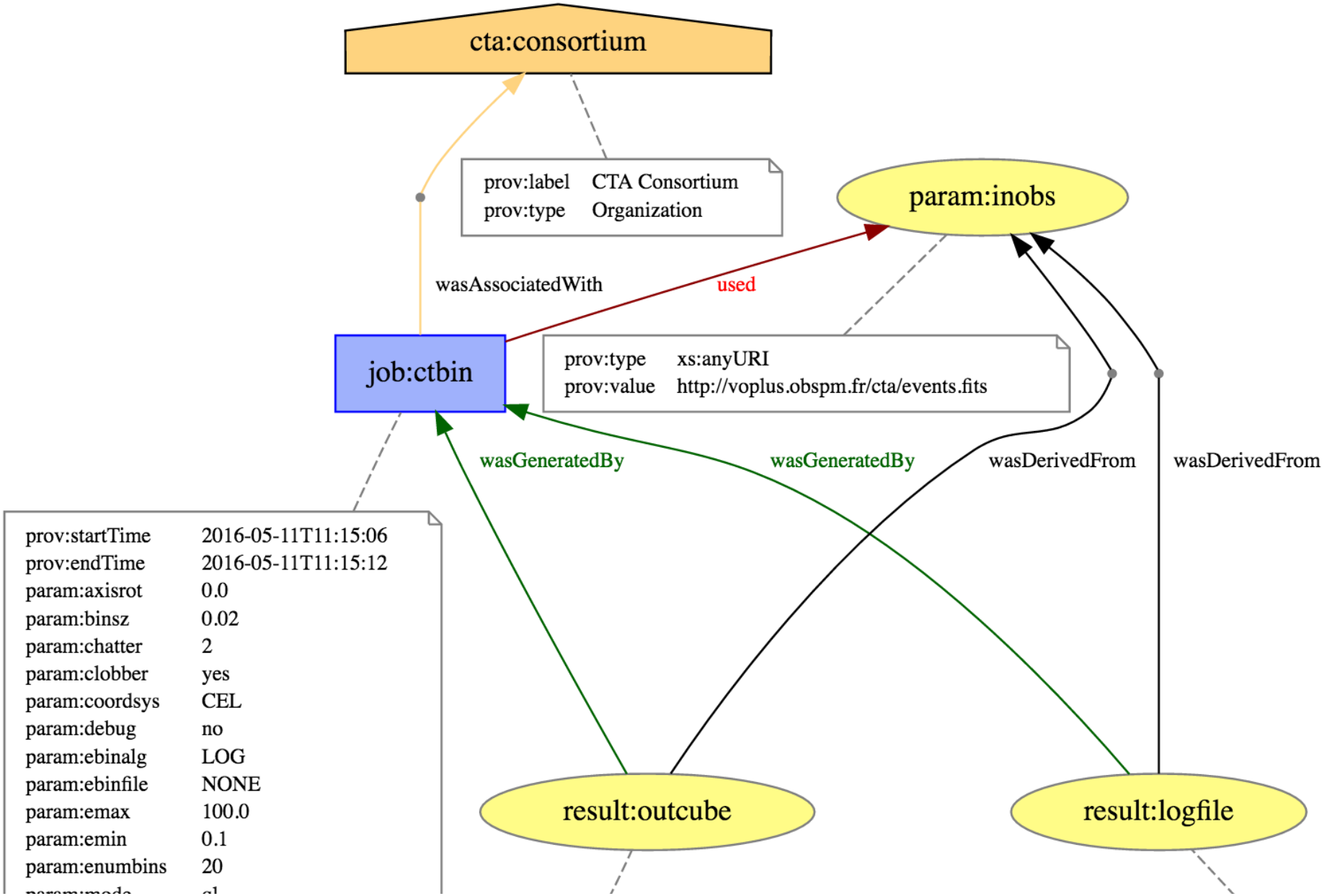


PROV output (XML and JSON)

```xml
<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"
        },
```

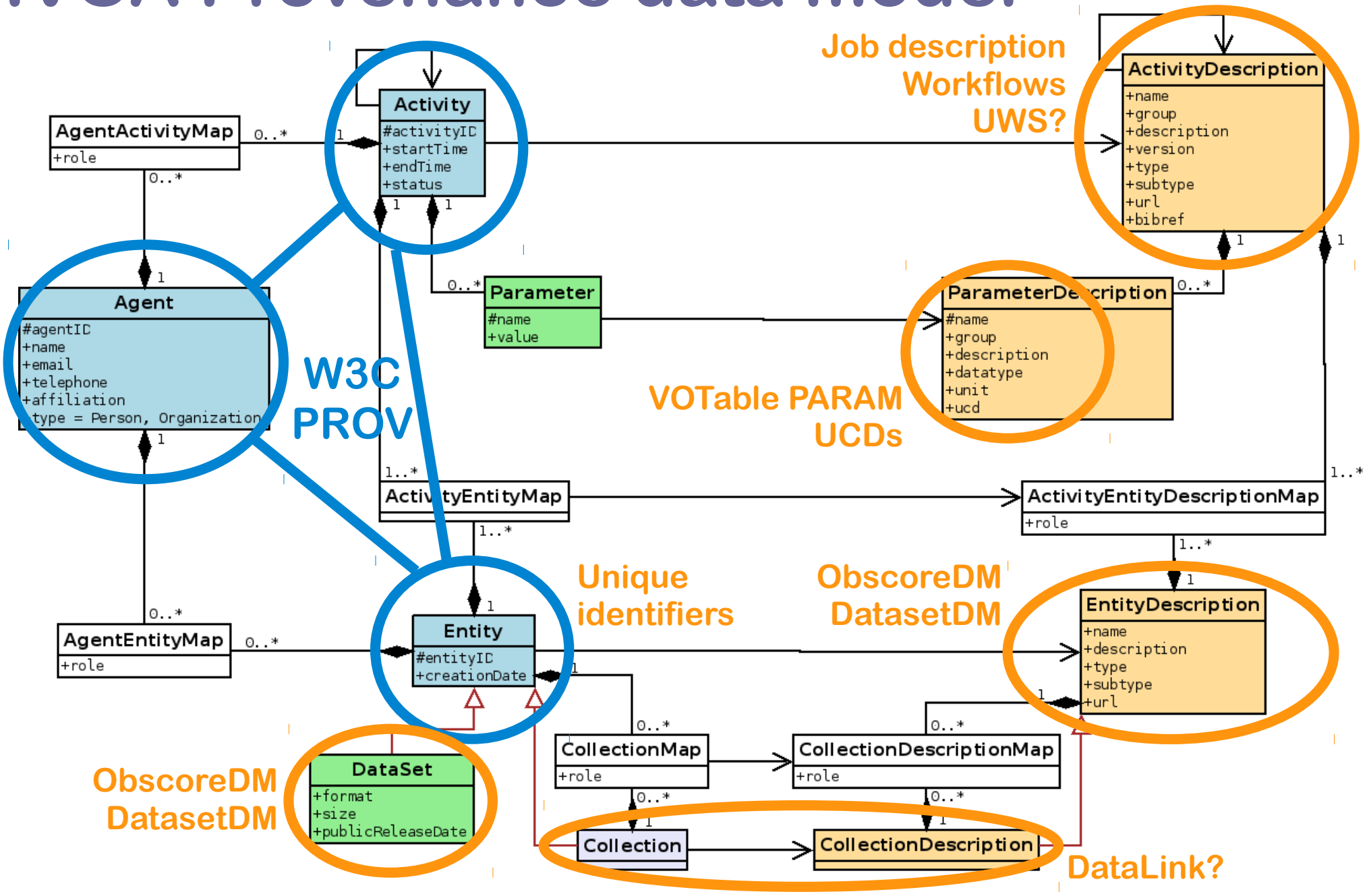
PROV output (SVG)



Requirements to store Provenance

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

IVOA Provenance data model



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/values 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 to get the ObsCore metadata (use unique identifiers)

Next steps

- Data Model definition:
  - Prototype side in diagram (as in SimDM)?
  - Parameter class?
  - Link to Dataset, Collections?
- Use the data model to define a database
- I/O package for this database
- 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...