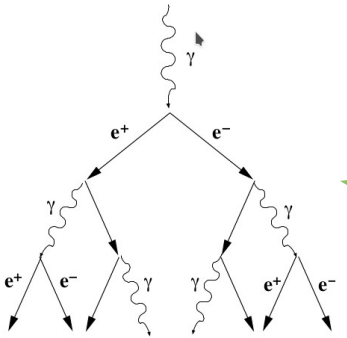
Archiving data from a software telescope

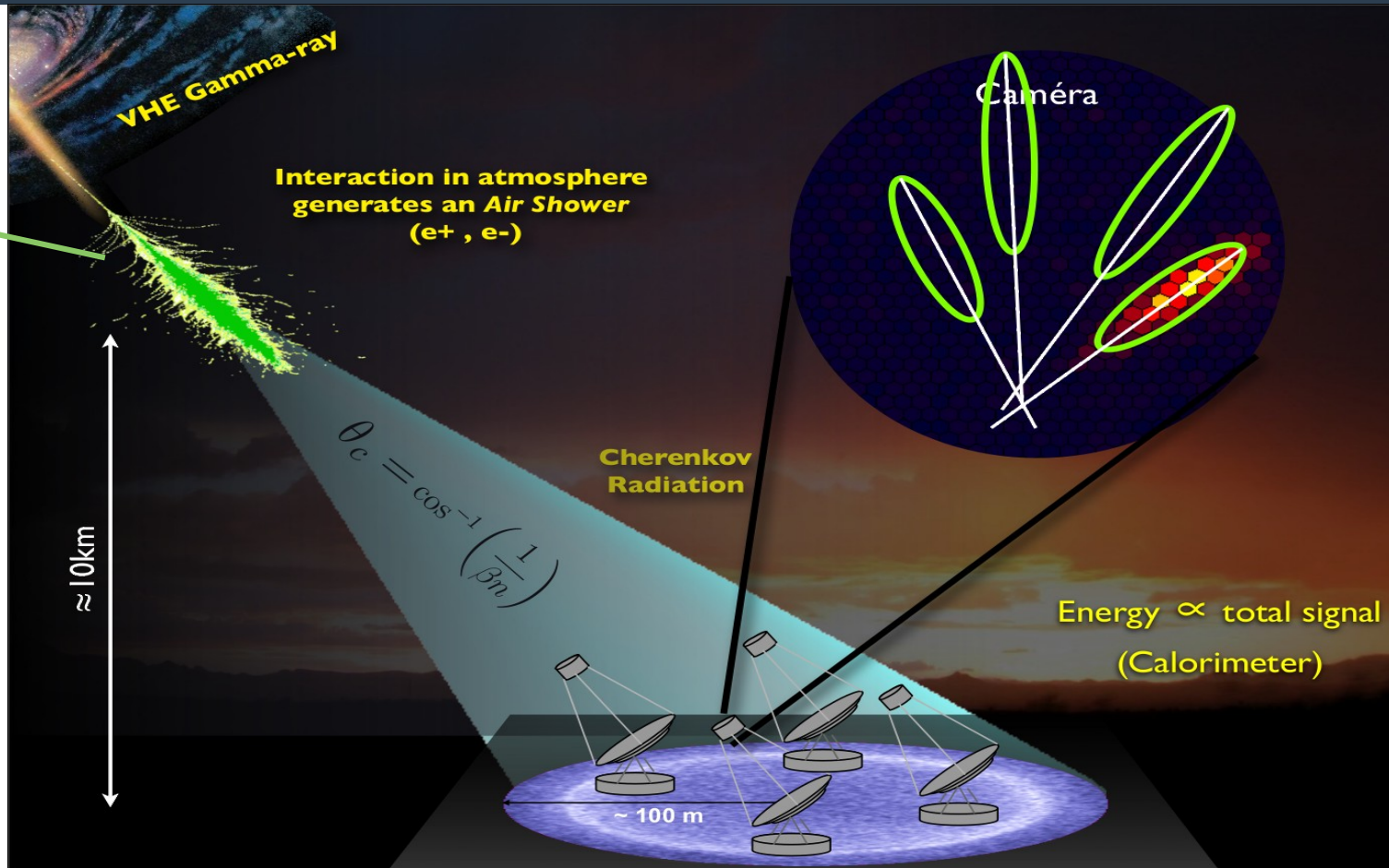
Catherine Boisson (Obs. de Paris)
with **M. Servillat**, K. Kosack, M. Louys , F. Bonnarel, M. Sanguillon

The New Era of Multi-Messenger Astrophysics, Groningen 2019

Cherenkov astronomy principles



Dark nights : small duty cycle
Event reconstruction :
photon, particle shower, Cherenkov light (faint, few nanoseconds)
Atmosphere = **calorimeter**
Simulations, assumptions
Complex metadata :
need to be structured



Challenge : complex instrument

Instrument's response changes with:

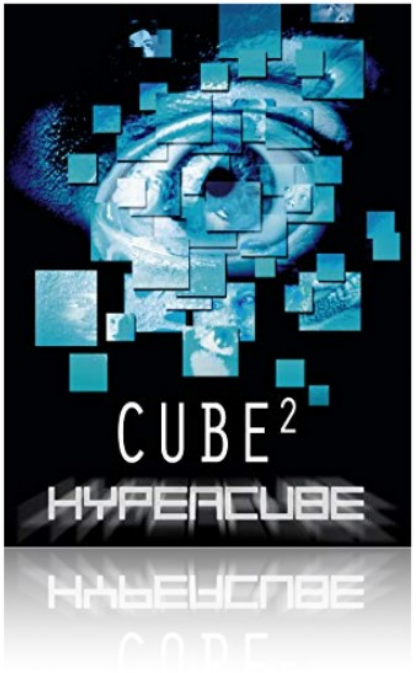
Change during an observation

- Gamma-ray **Energy**
- **Position** in Field-of-View
- **Zenith Angle** (elevation): atmosphere thickness
- **Azimuth**: Earth magnetic field orientation
- **Ground position** of shower in the array / Number of telescopes of each type that trigger / exactly which telescopes trigger!
- **Subarray Choice**

Change between observations

- **Atmosphere Density** profile
- Optical **Night-Sky-Background** light level (Moon, Zodiacal light, Light pollution)
- Atmosphere **Aerosol** content profile
- Detector Configuration (high voltage gain, etc)
- Analysis Configuration (reconstruction algorithm, discrimination strength, …)

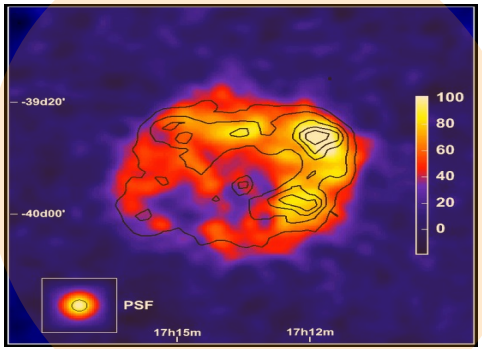
Potentially *very high-dimensional* **Instrumental Response Functions**!



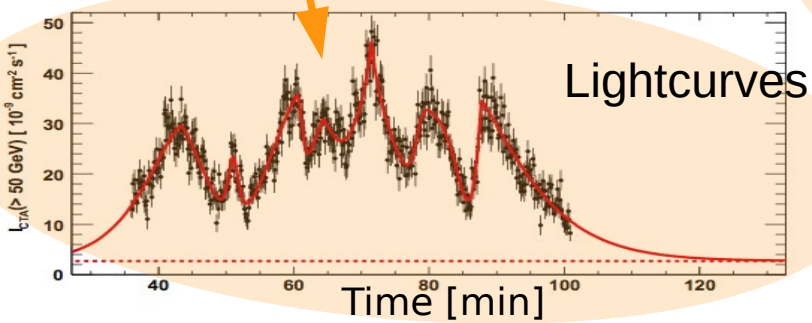
Or lots of custom simulations…

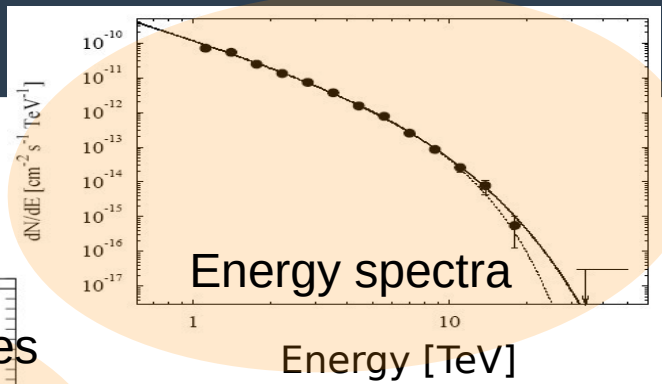
Multi-wavelength analysis

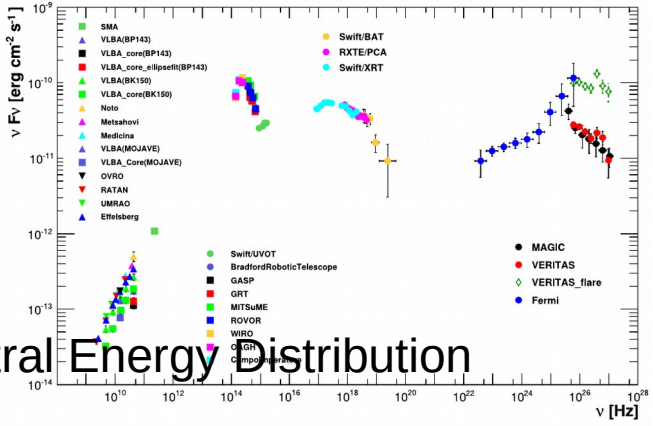
Event lists
(coordinates, time, energy)



Images





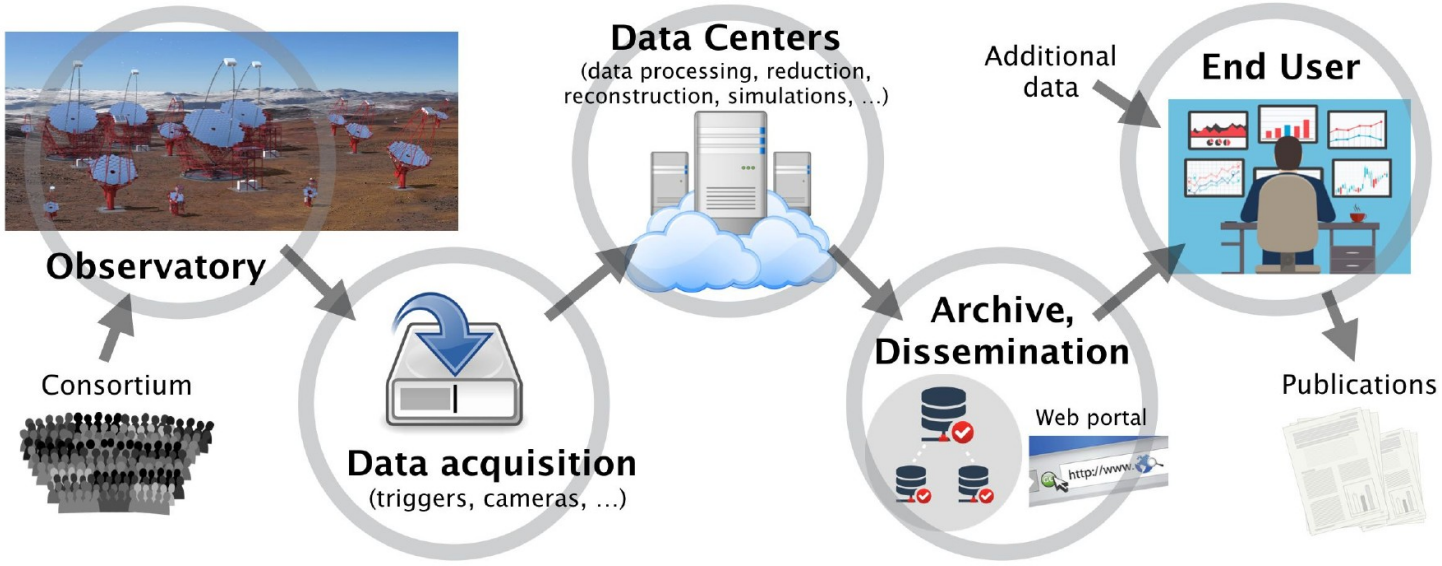


Spectral Energy Distribution

Compatible with data at other wavelength?

Simultaneous ?
Calibrated ?
Specific Processing ?
Context ?

Objectives and context



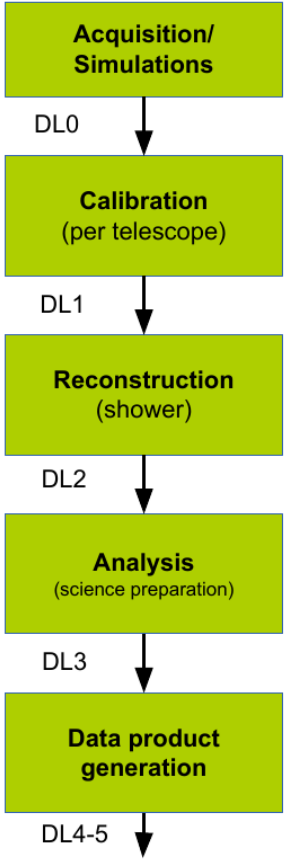
@M. Servillat

Data product generation **not obvious** to end user
Quality, **reliability**, **trustworthiness**?
Usefulness, **pertinence** of the data?

Need **structured** and **detailed** provenance information

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 (e.g. DL3 to DL4-5)



➢ Identify **how** a data product was produced

→ **Provenance**

➢ Identify what **detailed options** were used

→ **Configuration**

IVOA Goals for Provenance

- **Describe how data sets were produced:**
  - Observing process and observing conditions (summary only)
  - Data reduction, selection and extraction methods applied to raw measurements to build up science-ready data products (source lists, event lists, spectra, light curves, images, …)
  - Worflows executed to build theorical data (spectra, images, …)
- **Help VO users to:**
  - Derive selection criteria to filter out suitable data for her/his scientific needs
  - Run her/his own reduction method on intermediate data products in order to refine data analysis
  - Apply customized / dedicated workflows
  - Estimate better which data release fits their needs best

IVOA Design for Provenance

- **Not only history**
  - More than a log file
  - Attached to each science dataset distributed by a project archive if needed
- **Bring together commonalities**
  - General view for different archives / projects
  - Make it accessible via IVOA data access protocols
  - Enhance existing dataset distribution services by linking the provenance description
    - Not a new « full picture » model
- **Benefit from outside experience and framework**
  - W3C Provenance model
  - Existing serialisation formats

W3C Provenance definition

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

W3C PROV (PROV-DM, 2013)

Provenance refers to « the **information about entities, activities and people** involved in **producing, influencing**, or **delivering** a **piece of data or a thing**, which can be used **to form assessments** about its **quality, reliability** or **trusworthiness** »

Word Wide Web Consortium

Core concepts from the W3C PROV recommendations:

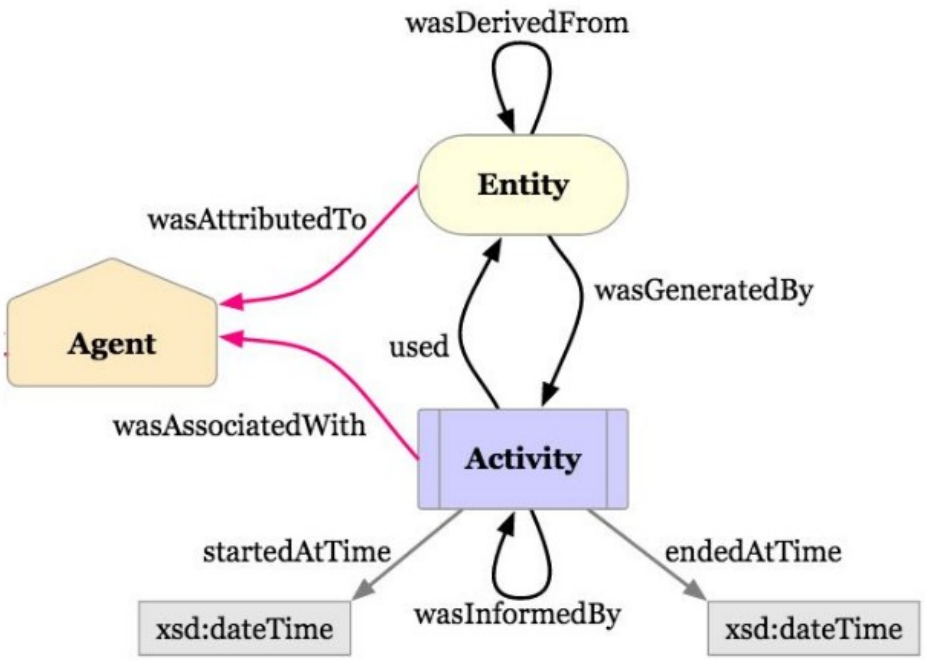
- **Entity - Activity - Agent**
- **Relations** and **roles**: e.g. generation, usage, influence, association, attribution, derivation, information

but W3C PROV has more relations

- IVOA Provenance connected to **VO concepts** and **astronomy needs**

Core Provenance data Model

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



In our context

- Entity: data products (files), ancillary data (calibration, instrumental response, etc.), processing parameter files
- Activity: data acquisition, mosaicing, regridding, fusion, calibration, …, transformation
- Agent: Telescope astronomer, pipeline operator, principal investigator, etc.

Word Wide Web Consortium

Existing Implementations:

- Java library
- Python package
- ProvStore web service

Goals for an IVOA Provenance DM

A: Tracking the production history

B: Attribution and contact information

- Find the people involved in the production of a dataset, that need to be cited or can be asked for more information.

C: Quality assessment

- Judge the quality of an observation, production step or dataset.
- Get detailed information on the methods/tools/software that were involved
- Check if the processing steps (including data acquisition) went "well"
- Extract the ambient conditions during data acquisition (cloud coverage? wind? temperature?)

Goals for an IVOA Provenance DM

D: Locate error sources

- Check the version of the code/pipeline (ctapipe) that was used, the versions of the dependencies (OS, python, packages)
- Check if the workflow was correctly followed, look for missing steps in the processing, for proper configuration, proper execution environment

E: Search in structured provenance metadata

- Find all the DL0/1/2/3 data acquired with a given configuration/version of the DAQ and/or processed with a given version/configuration of the processing pipeline, and/or for which a given version of the calibration of the instrument was used
- Find all generated DL1 data files that used incorrectly generated file X as input, so that they can be marked for re-processing

Chain of entities, activities, agents

- Provenance goes back in time (not a workflow!)
- Relations do help to track history of an entity

But the W3C core model is too generic

⇒ additional relations, entities and attributes are needed

⇒ **enrich** the W3C classes by adding new classes : ActivityDescription, EntityDescription, ActivityFlow, etc

Relevant provenance information

- **C: Quality and Reliability:**

  Which steps, which algorithms involved in the calibration?

  ⇒ **Descriptions**: information on the expected content of an activity and on the expected structure of an entity, i.e what is known before any activity or entity instance is created

- **D: Locate errors**

  What was the detailed configuration of the pipeline?

  ⇒ **Configuration:** information passed to an activity in order to initialize its inner working (parameter, config file)

- **C + D tasks**

  ⇒ **Context:** information on the context that influences the development of an activity, but for which there is no or little control at the moment of its execution (e.g. Ambient Conditions, Instrumental Context, Execution Environment).

IVOA Provenance Data Model

Proposed recommendation:

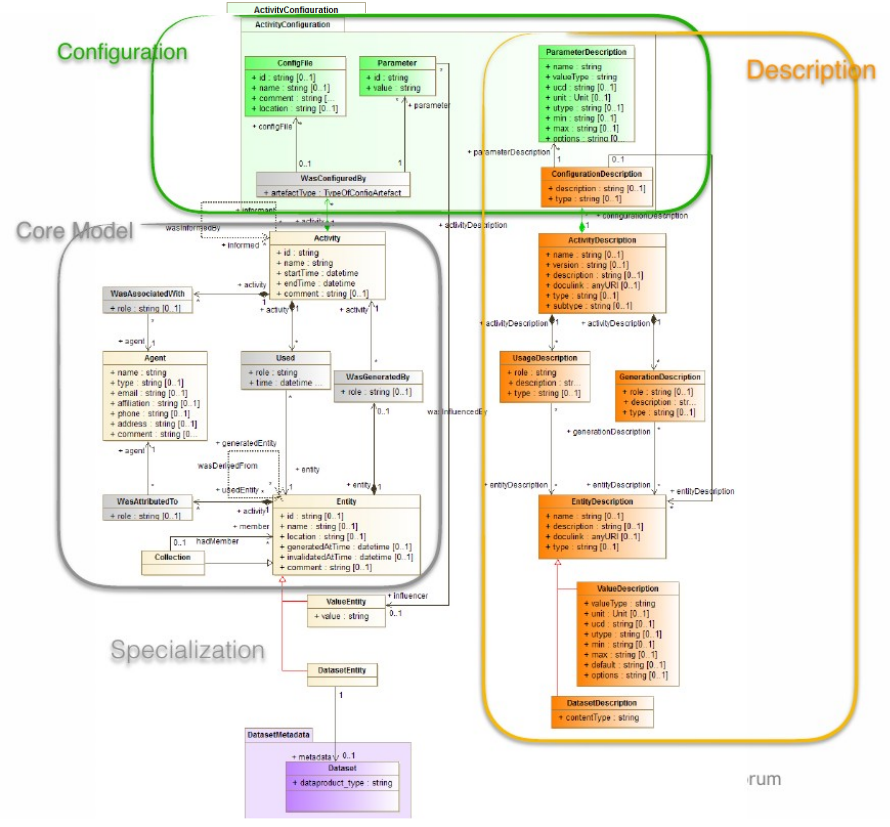
- Core W3C model
- Activity **Descriptions**
- Activity **Configuration**

+ Detailed granularity

⇒ **Reproducibility**

+ Relevant information attached to datasets

⇒ **Pertinence** of dataset for Science



IVOA Provenance Data Model

Version 1.0

IVOA Proposed Recommendation 2018-10-15

Working group
DM

This version
http://www.ivoa.net/documents/ProvenanceDM/20181015

Latest version
http://www.ivoa.net/documents/ProvenanceDM

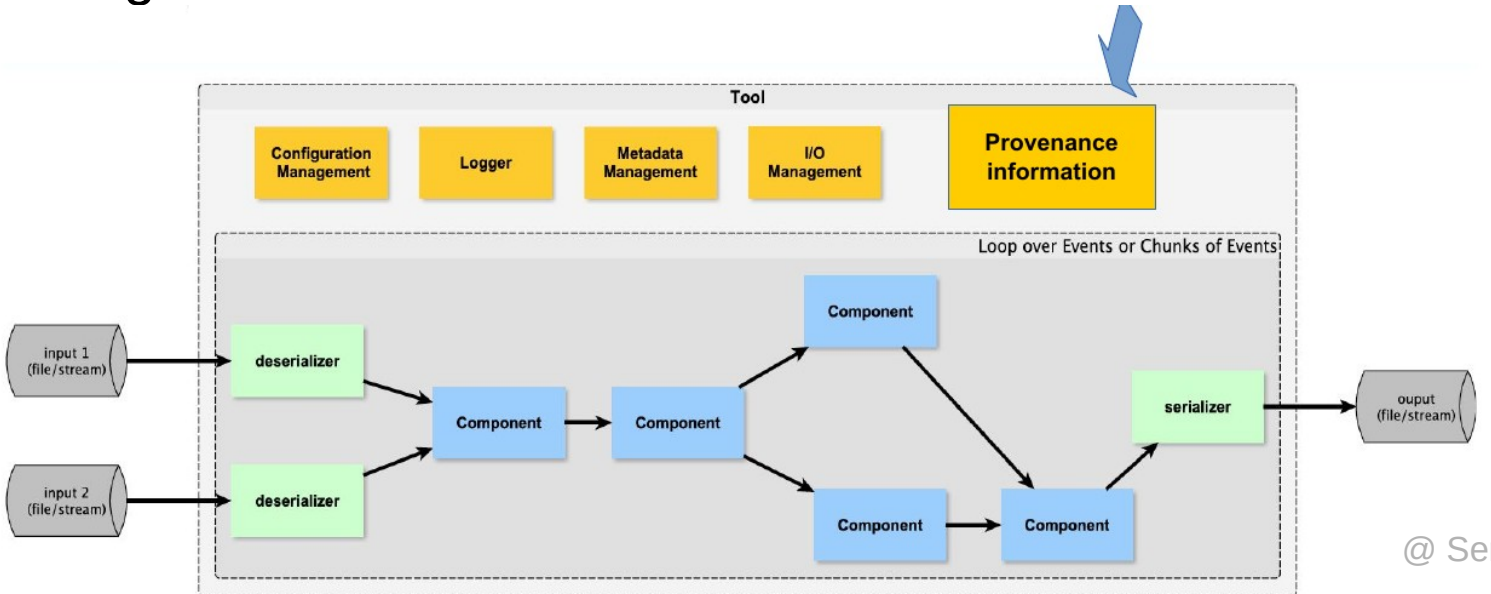
Previous versions
WD-ProvenanceDM-1.0-20180530.pdf
WD-ProvenanceDM-1.0-20170921.pdf
WD-ProvenanceDM-1.0-20161121.pdf
ProvDM-0.2-20160428.pdf
ProvDM-0.1-20141008.pdf

Author(s)
Mathieu Servillat, Kristin Riebe, François Bonnarel, Anastasia Galkin, Mireille Louys, Markus Nullmeier, Michèle Sanguillon, Ole Streicher, and the IVOA Data Model Working Group

Editor(s)
Mathieu Servillat

Provenance in ctapipe

- **Ctapipe** : the CTA data processing framework https://github.com/cta-observatory/ctapipe
- **Tool Python class** providing configuration, logger, I/O management and **Provenance information**



Provenance inside OPUS

OPUS (**O**bservatoire de **P**aris **U**WS **S**ystem) is a job control system developed using the Python micro-framework bottle.py.

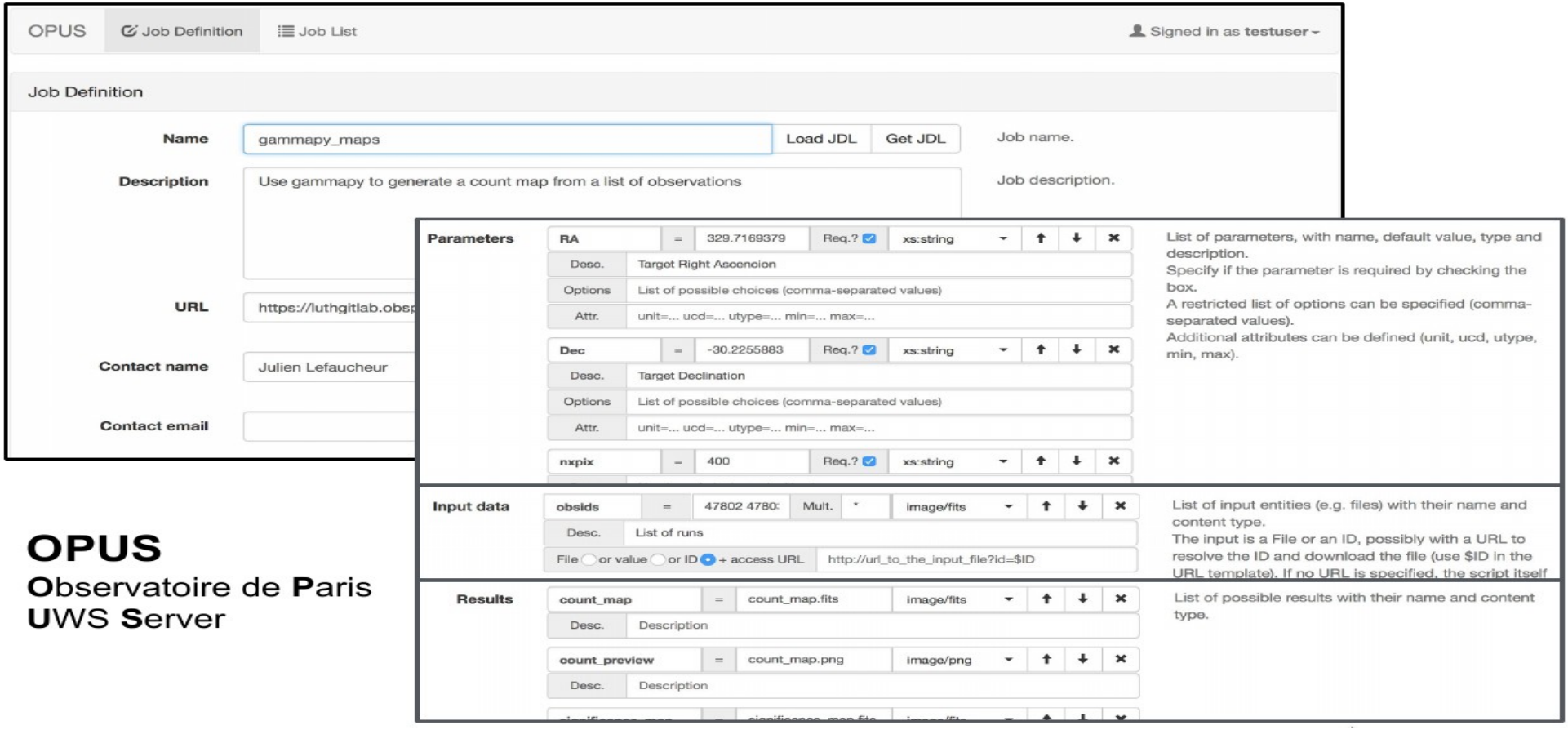
The **Universal Worker System pattern v1.1 (UWS)** as defined by the IVOA is implemented as a REST service to control job execution on a work cluster.

OPUS also follows the proposed **IVOA Provenance Data Model** to capture and expose the provenance information of jobs and results.

https://github.com/ParisAstronomicalDataCentre/OPUS

https://opus-job-manager.readthedocs.io

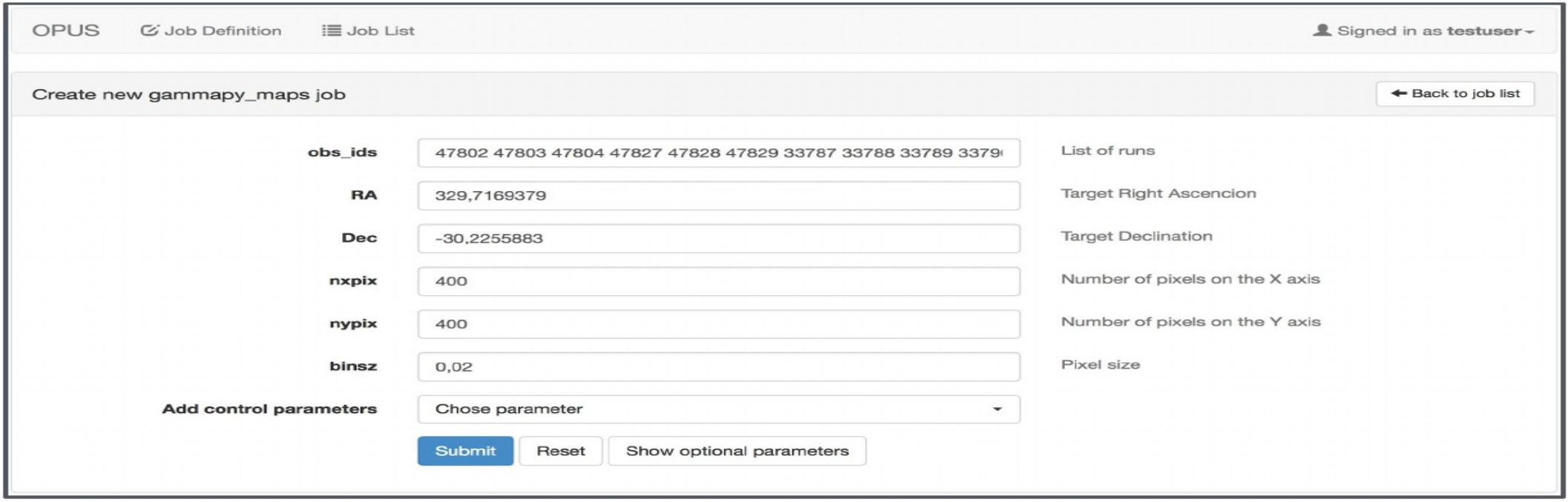
Definition of the gammapy_maps job



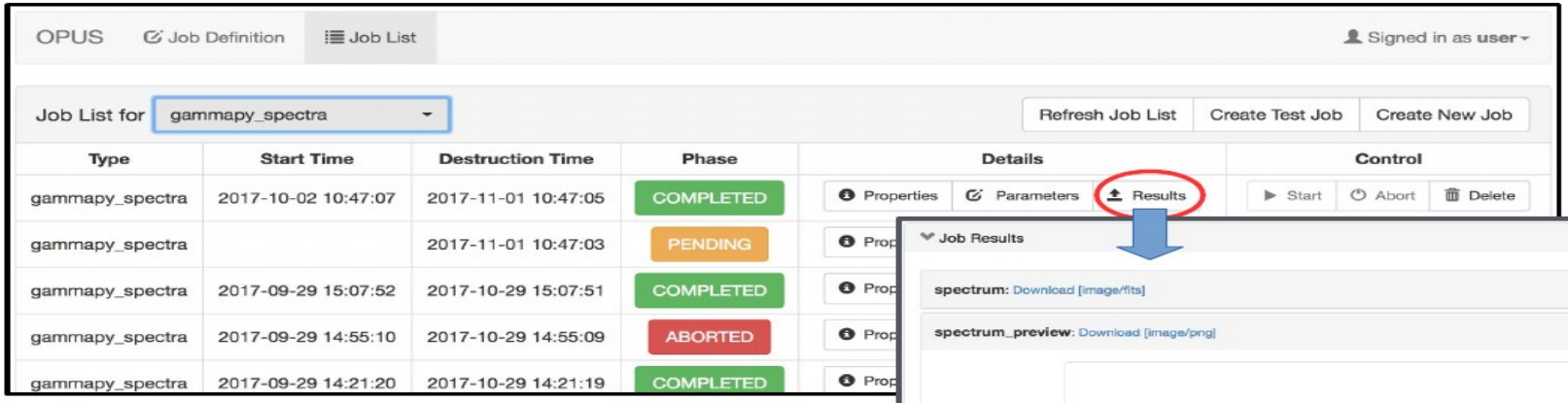
OPUS
Observatoire de **P**aris
UWS **S**erver

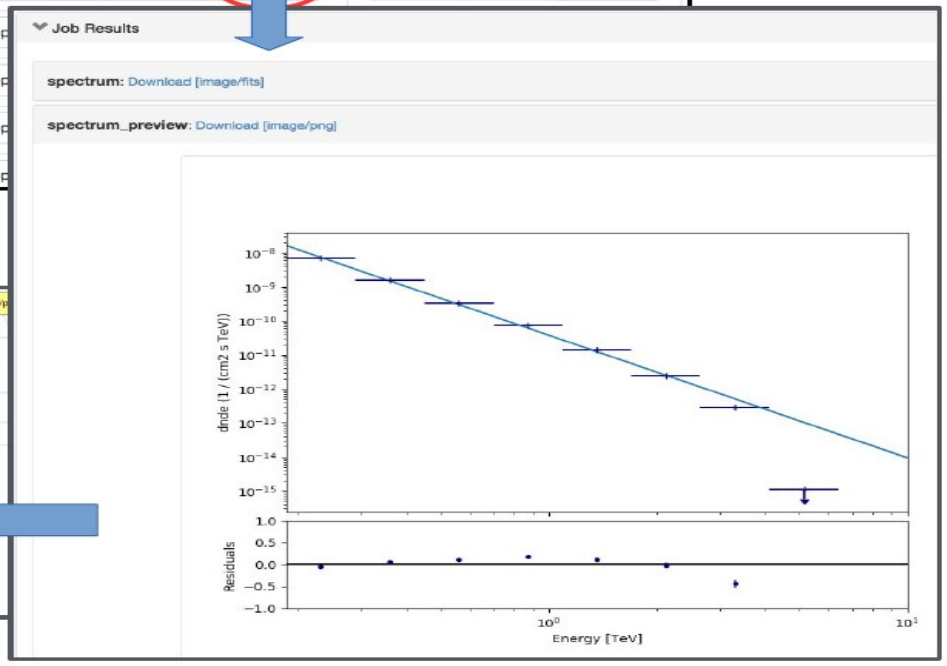
Submission of a gammapy_maps job

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

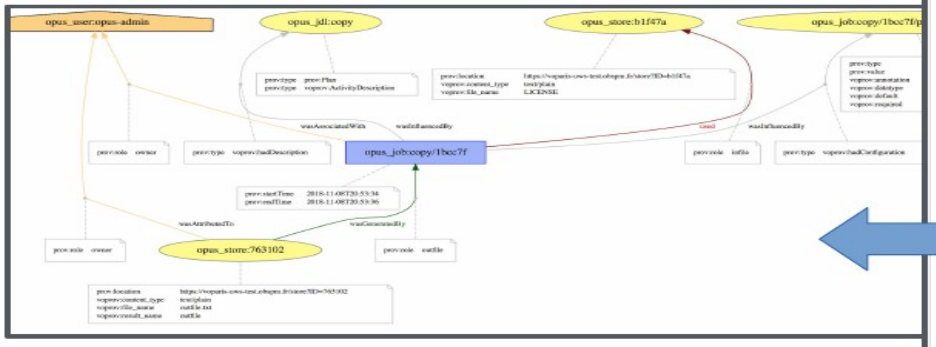


Results and Provenance

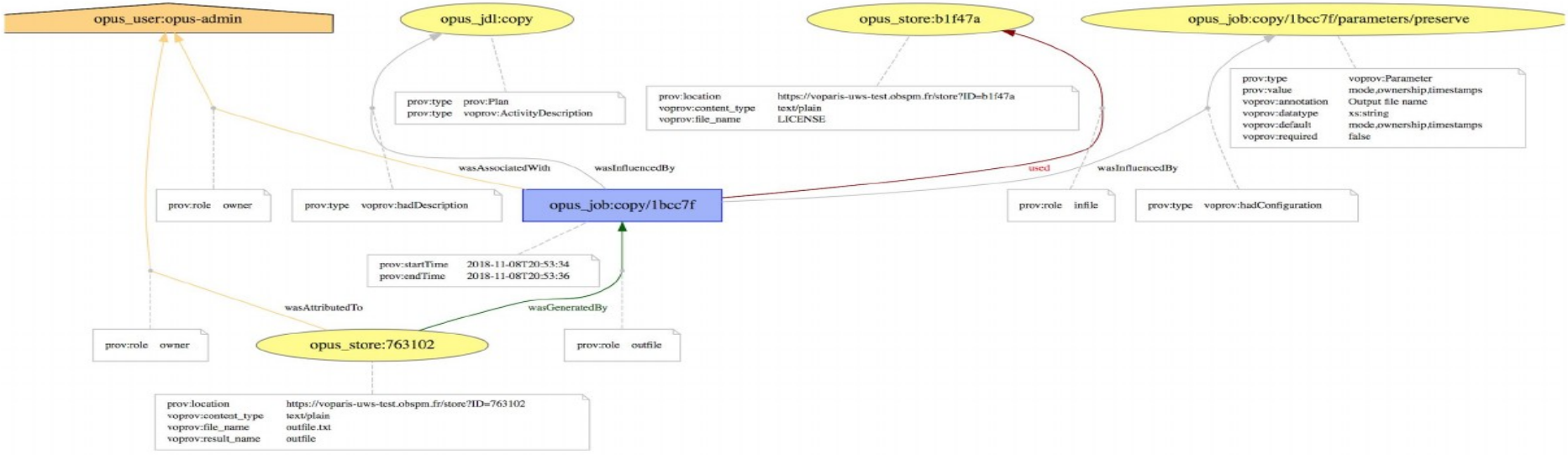




Tracking of Provenance informations



Provides Provenance files



Complex searches in provenance databases

- **Use existing ADQL (Virtual Observatory query langage) infrastructure**
  - GUI clients: Aladin Sky Atlas (desktop), TOPCAT
  - Scripting clients: `pyvo.readthedocs.io`, STILTS, tapsh
  - Various software packages for implementing servers

- **Implemented via custom ADQL functions**
  - *Example: retrieve all precursors of a given set of entities / activities:*
  - ```
    SELECT
      find_prov_precursors(max_depth, entities, activities);
    ```
    → *returns subgraph of all specified nodes' precursors*

- **Portability for various actually used database back ends**
  - recursive CTEs (common table expressions) of SQL as implementations langage

adapted from Markus Nullmeier (Heidelberg)
    ```

Docker & Provenance - KM3NeT

Docker Images for Software Development

- Integrated into GitLab CI Pipelines in KM3NeT

Singularity Images used in the production pipeline, built from Docker images as containers to run reproducible analysis chains in strictly controlled environments.

- Seamless integration into the (file) system of the host computer
- Completely independent base environment
- Perfect fit for Grid computing with heterogeneous systems
- Already used in ANTARES production pipelines and currently tested in KM3NeT

Data Provenance

- File history is part of the data preservation
- Provides historical records of the data and its origins, like
  - container IDs (e.g. Singularity Registry)
  - specific software and library versions
  - and additional parameters (e.g. command line arguments or configuration files)
- Data and Software stored together in a single image file
- Reproducibility of results through containerised analysis chain

adapted from Tamas Gal (KM3NeT, ECAP)

Implementations

Based on four use-cases

- CTA
- RAVE – the Radial Velocity Experiment
- POLLUX (synthetic stellar spectra service)
- SVOM gamma ray burst /transients
- Prototype TAP-based API for images in an archive (@CDS)

Summary

- Provenance DM mature
- The various profiles the IVOA DM can offer
  - Workflow : Activity focused
  - Data flow / archive : Dataset focused
  - Credits/responsibility views

Bring your experiment use cases !

Implementations

Ready :

- CTA : OPUS job submission and execution (LUTH). VOTable, JSON
- Image database prototype in Triplestore (CDS) RDF
- HiPS Image database (CDS) PROV-TAP (Table access protocol) VOTable
- RAVE implementation (AIP, Postdam)
  - Simple access (Prov-SAP) prototype Prov-N, PROV-JSON
- Provenance for Pollux DB & voprov library (LUPM) VOTable, Prov-N, JSON

Under study :

- CTA pipe implementation for raw data DL0 - DL1
- SVOM pipeline execution tracking JSON