Provenance as a requirement for **large-scale complex astronomical instruments**

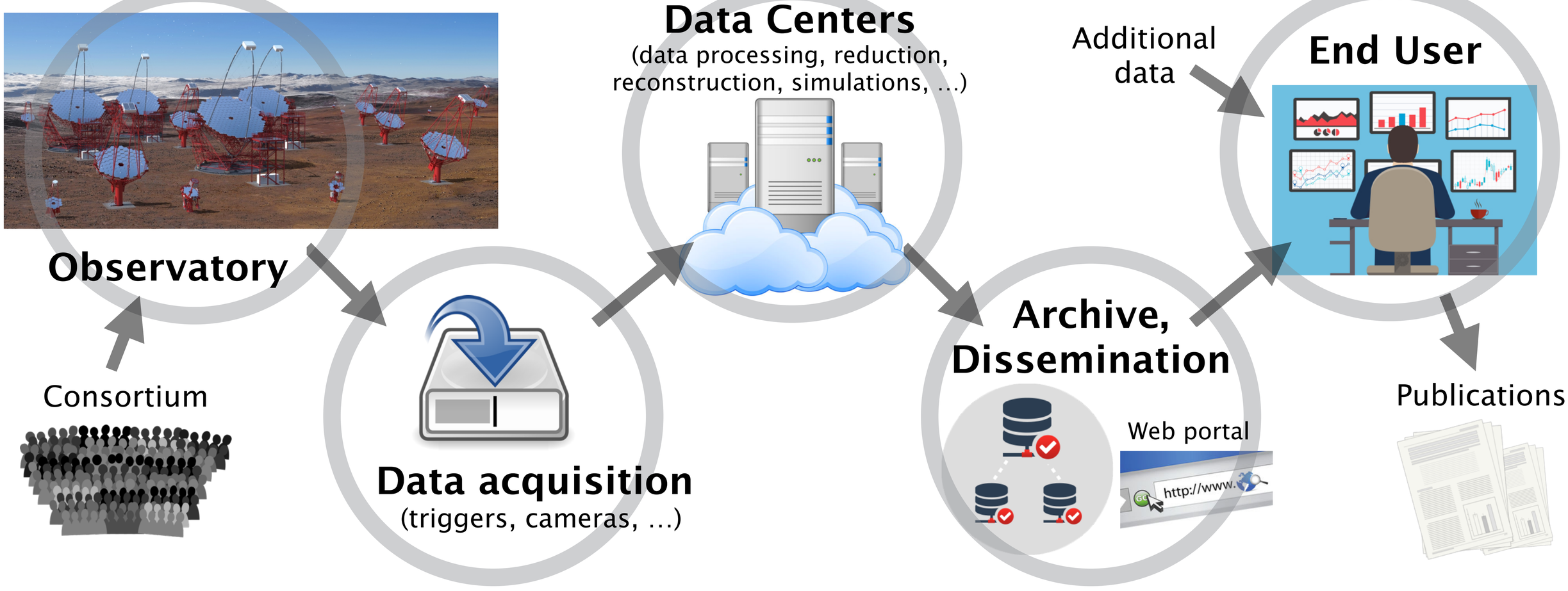
Mathieu Servillat,[1] Catherine Boisson,[1] Julien Lefaucheur,[1] Karl Kosack,[2] Michèle Sanguillon,[3] Mireille Louys,[4,5] François Bonnarel [4]

[1] LUTH, [2] CEA Saclay, [3] LUPM, [4] CDS, [5] ICube

mathieu.servillat@obspm.fr

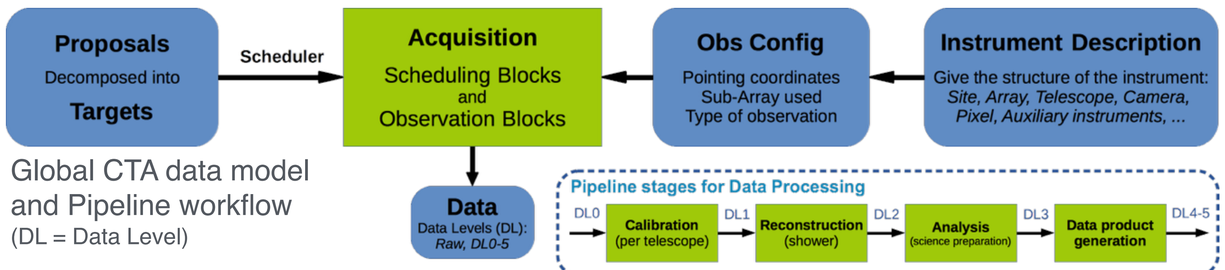
Context: State of the art observations are now performed by large-scale complex astronomical instruments. A **consortium** of specialists is generally responsible for the development and the operation of large **observatories**, as it is the case for example for the Cherenkov Telescope Array (CTA). The path of the data production from **acquisition** to **dissemination**, through e.g. **data centers**, **archives** and **web portals,** can be extremely obscure to the **end user**.

Provenance: to assess the **usefulness** and the **quality** of the data for their own scientific work, end users need a flowchart explaining the large number of steps and complexity involved in the data preparation. This can be done by collecting **provenance information** at each step of the data preparation. We followed the **IVOA Provenance data model** (see Poster 129) to develop solutions for CTA.



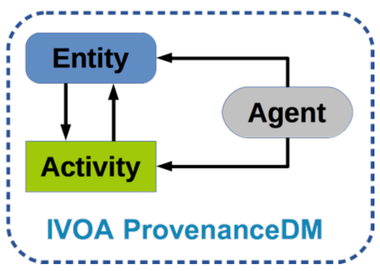
1/ How to collect provenance information during CTA data production?

- Include the relevant **metadata** in a structured CTA data model



Global CTA data model and Pipeline workflow
(DL = Data Level)

- Follow the **IVOA Provenance data model** for the generated **Data**



- **Collect** provenance information at each step of the data processing:
  - ➡ Use **unique identifiers** for entities, activities and agents
  - ➡ **Describe** each activity
  - ➡ Keep a list of all **used** and **generated** entities during the execution of an activity
- A **Provenance Python class** has been developed for the CTA Pipeline framework `ctapipe`

```python
from ctapipe.core import Provenance

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

prov.start_activitity("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
```

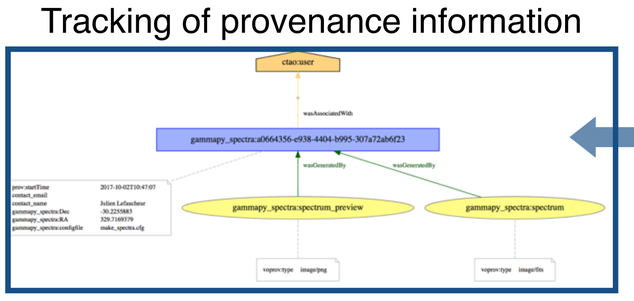
https://github.com/cta-observatory/ctapipe

2/ How to store and expose the provenance information in a standard format?

We developed a **job control system** that stores provenance information following the IVOA UWS pattern and Provenance data model. The following features have been implemented:

- ➡ **Edit and fill** Activity Descriptions
- ➡ Run jobs **asynchronously** on a **work cluster**
- ➡ **Generate** and return Provenance files after job completion

Tracking of provenance information



https://github.com/mservillat/OPUS

Acknowledgements: This work was partially funded by the Federal Ministry of Education and Research in Germany and by the ASTERICS project (http://www.asterics2020.eu/). Additional funding was provided by the INSU (Action Spécifique Observatoire Virtuel, ASOV) and the Paris Astronomical Data Centre at the Observatoire de Paris.