Provenance IVOA : modèle et collecte des informations

Mathieu Servillat

Laboratoire Univers et Théories
Observatoire de Paris
PSL Research University

Réunion annuelle de l'ASOV

IVOA Provenance

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

International
Virtual
Observatory
Alliance

IVOA Provenance Data Model

Version 1.0

IVOA Working Draft 2017-02-17

Author(s)

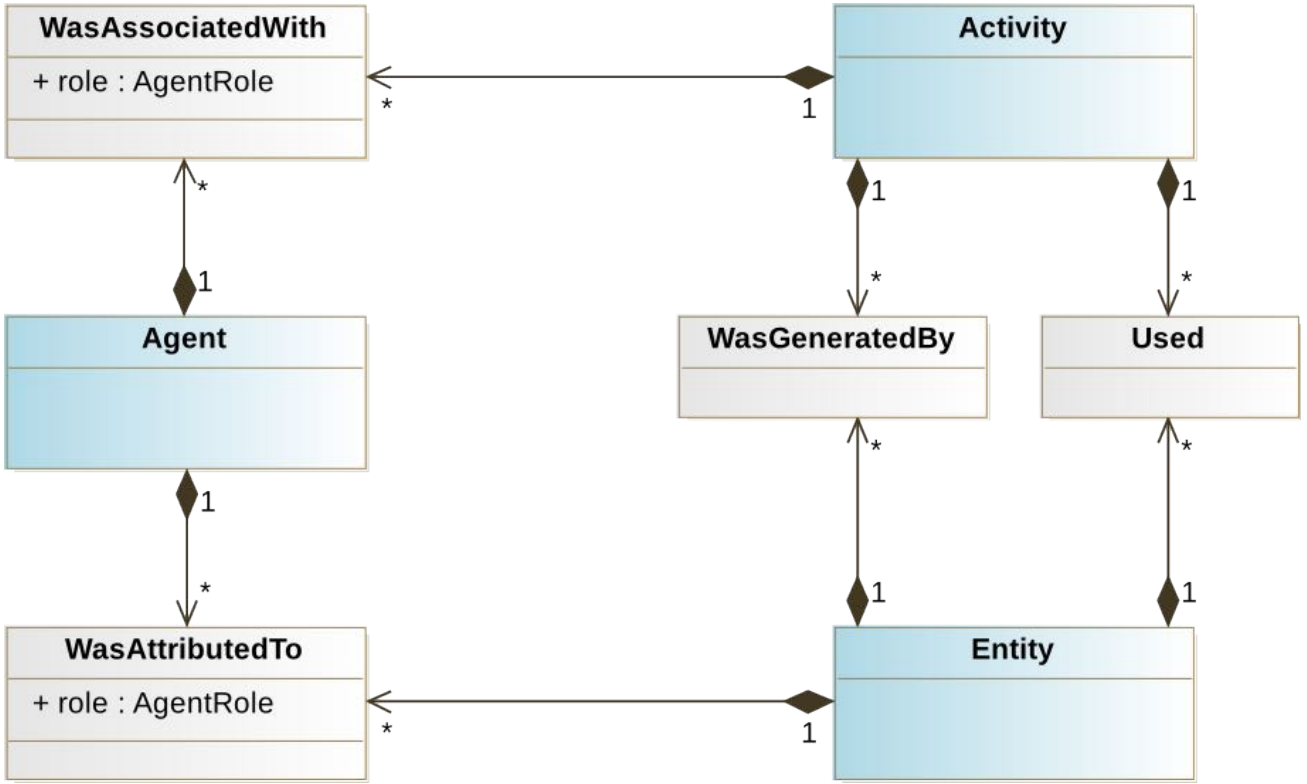
Kristin Riebe, Mathieu Servillat, François Bonnarel, Mireille Louys, Florian Rothmaier, Michèle Sanguillon, IVOA Data Model Working Group

Editor(s)

Kristin Riebe, Mathieu Servillat

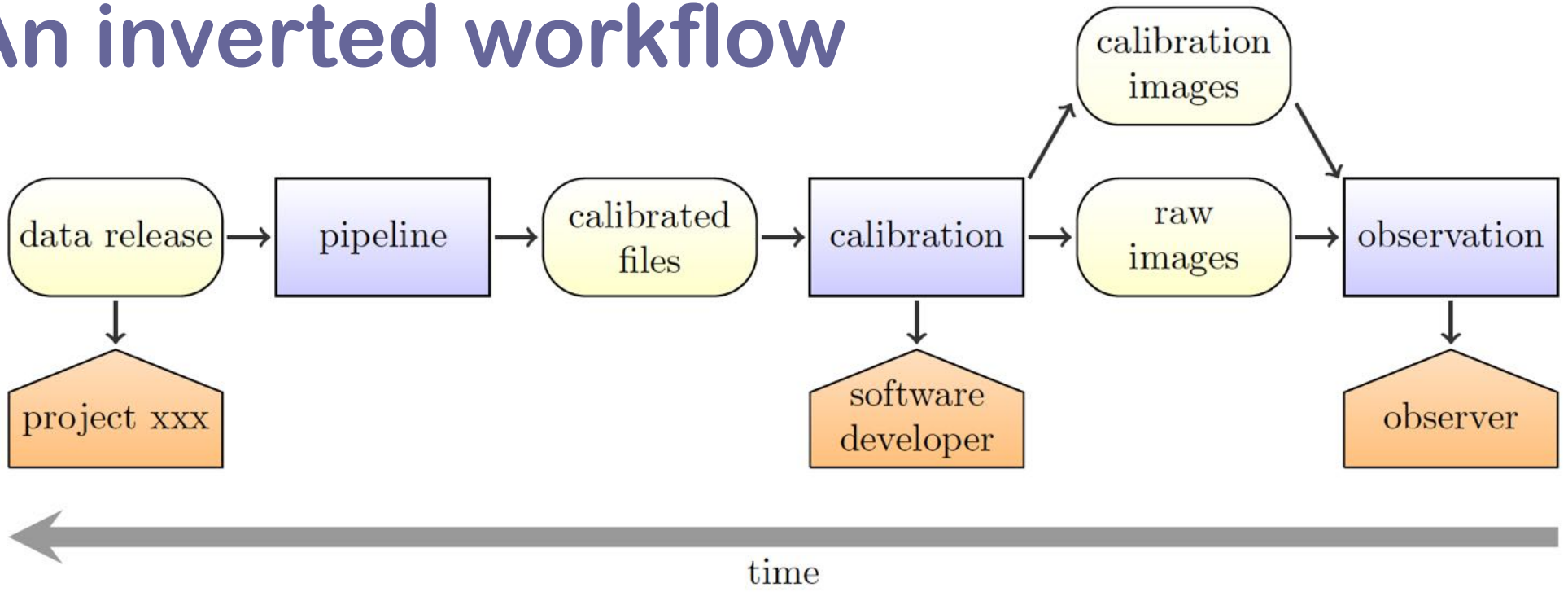
W3C PROV

Provenance is “information about **entities**, **activities**, and **people** involved in producing a piece of data or thing, which can be used to form assessments about its **quality**, **reliability** or **trustworthiness**”.



W3C PROV Ontology : https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

An inverted workflow



In Astronomy :

- **Entities** = datasets composed of VOTables, FITS files or database tables, or files containing logs, values (spectra, lightcurves), parameters, etc.
- **Activities** = an observation, a simulation, or processing steps (image stacking, object extraction, etc.).
- The **people** involved can be individual persons (observer, publisher...), groups or organisations.

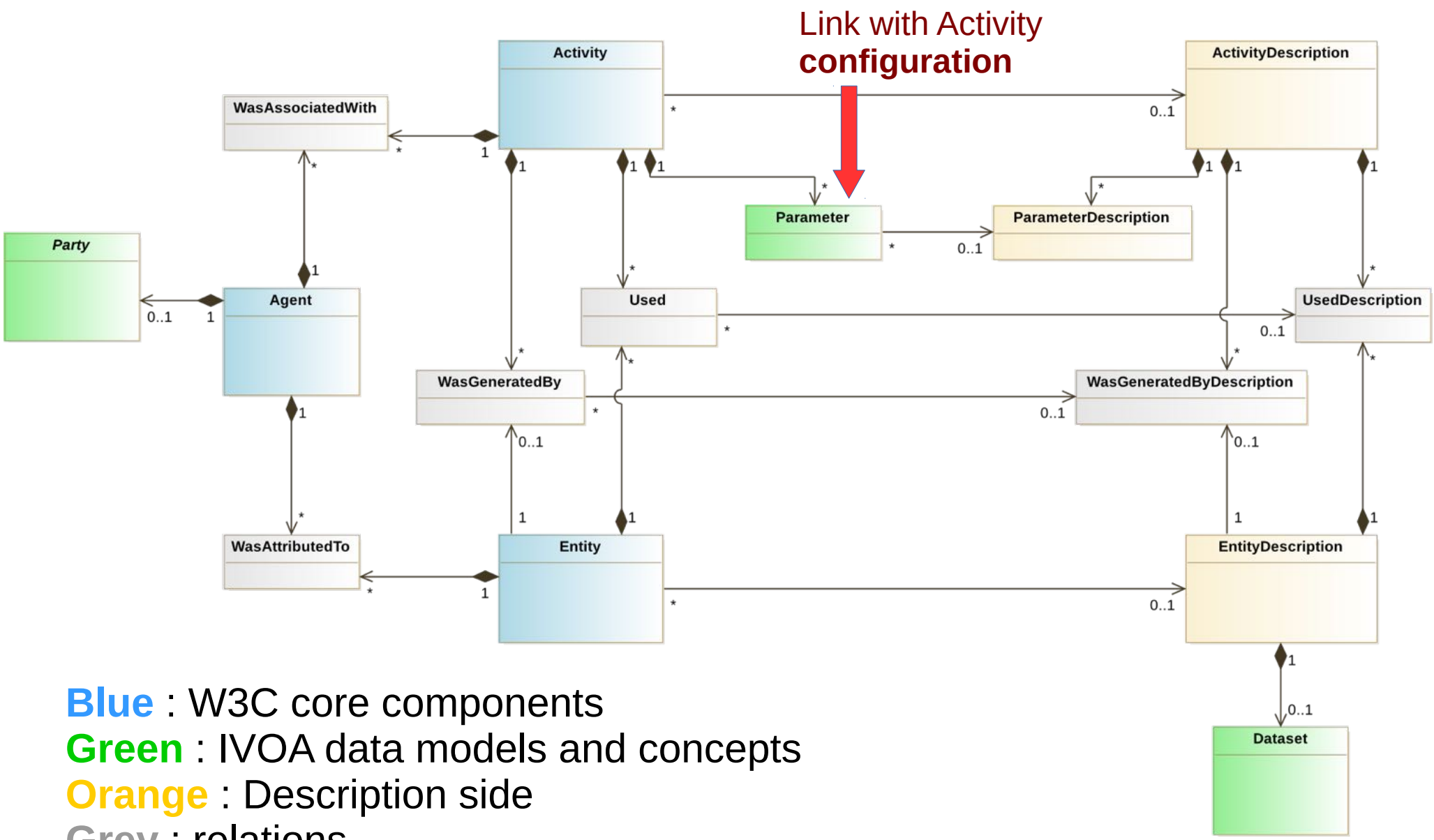
Goals of the Provenance data model

- A: Tracking the production history
  Find out which steps were taken to produce a dataset and list the methods/tools/software that was involved.
- 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: Locate error sources
  Find the location of possible error sources in the generation of a dataset.
- D: Quality assessment
  Judge the quality of an observation, production step or dataset.
- E: Search in structured provenance metadata
  This would allow one to also do a “forward search”, i.e. locate derived datasets or outputs.

Minimum requirements for Provenance

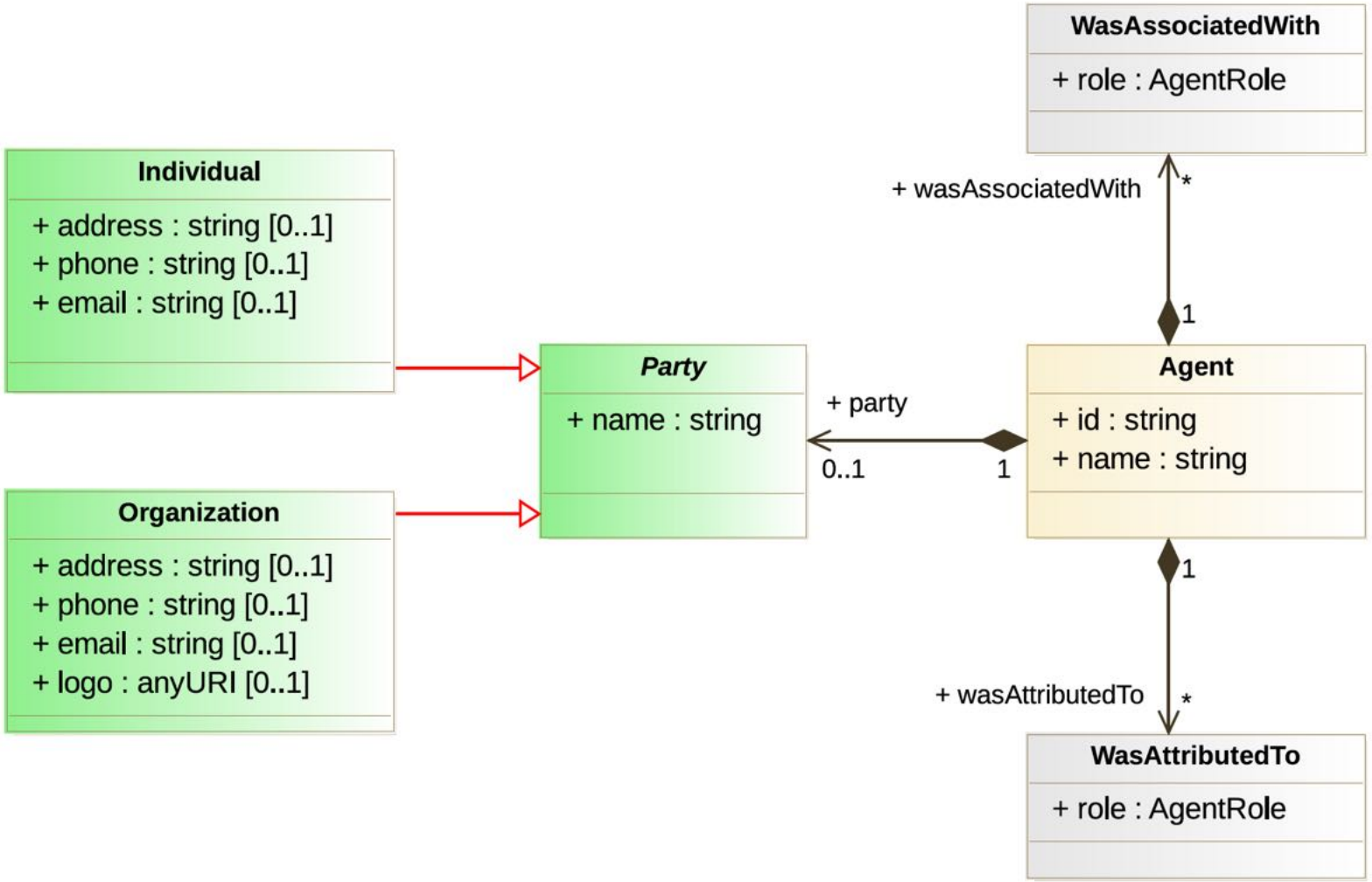
- Provenance information must be stored in a **standard model**, with **standard serialization formats**.
- Provenance information must be **machine readable**.
- Provenance data model classes and attributes should be **linked to other IVOA concepts** when relevant (DatasetDM, ObsCoreDM, SimDM, VOTable, UCDs...).
- Provenance information should be **serializable into the W3C** provenance standard formats (PROV-N, PROV-XML, PROV-JSON) with minimum information loss.
- Provenance metadata must contain information to find immediate **progenitor(s)** (if existing) for a given entity, i.e. a dataset.
- An entity must point to the activity that generated it (if the activity is recorded).
- Activities must point to input entities (if applicable).
- Activities may point to output entities.
- Provenance information should make it possible to derive the **chronological sequence** of activities.
- Provenance information can only be given for **uniquely identifiable** entities, at least inside their domain.
- Released entities should have a **main contact**.
- It is recommended that all activities and entities have contact information and contain a (short) description or link to a description.

IVOA Provenance data model

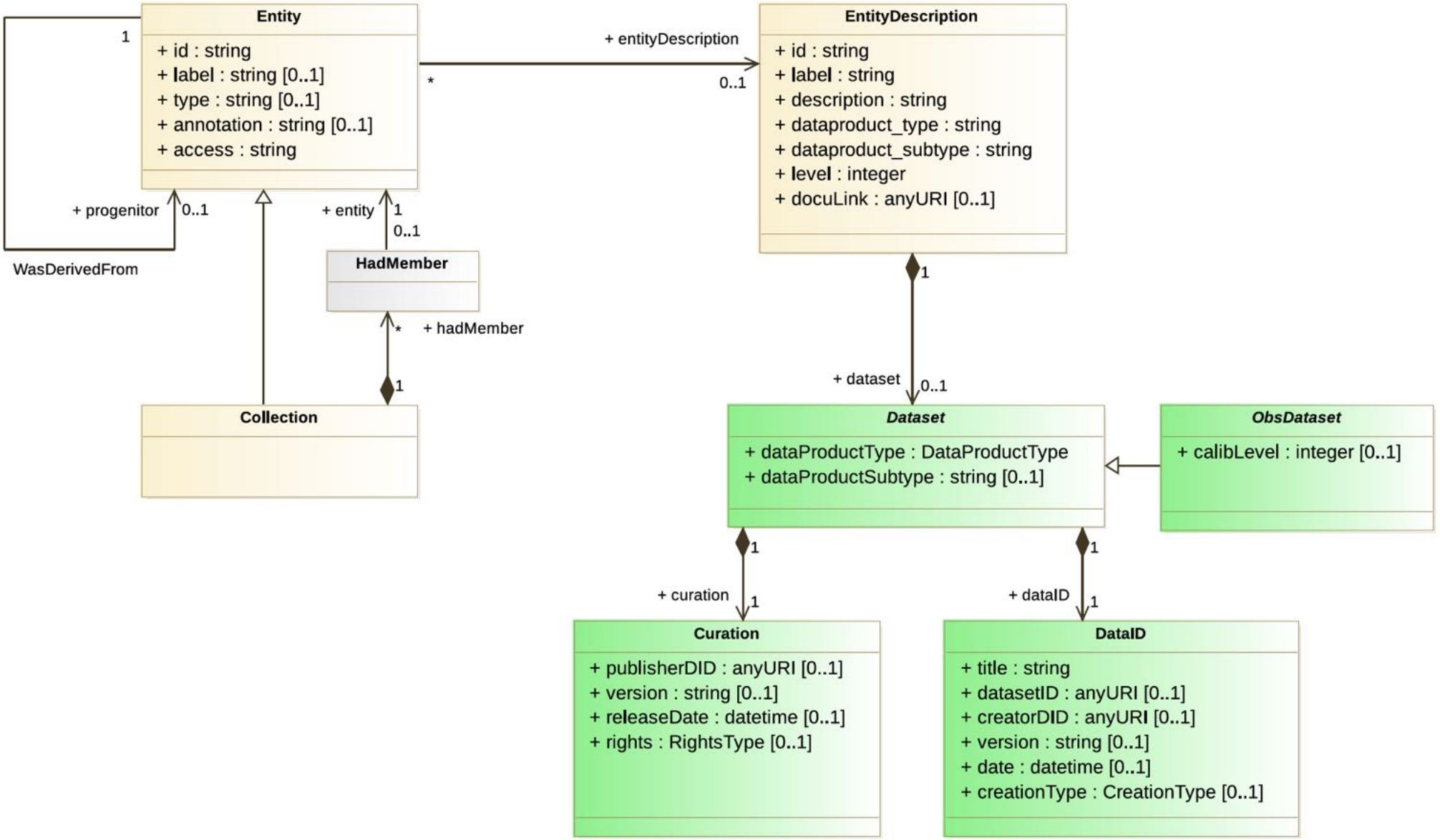


Blue : W3C core components
Green : IVOA data models and concepts
Orange : Description side
Grey : relations

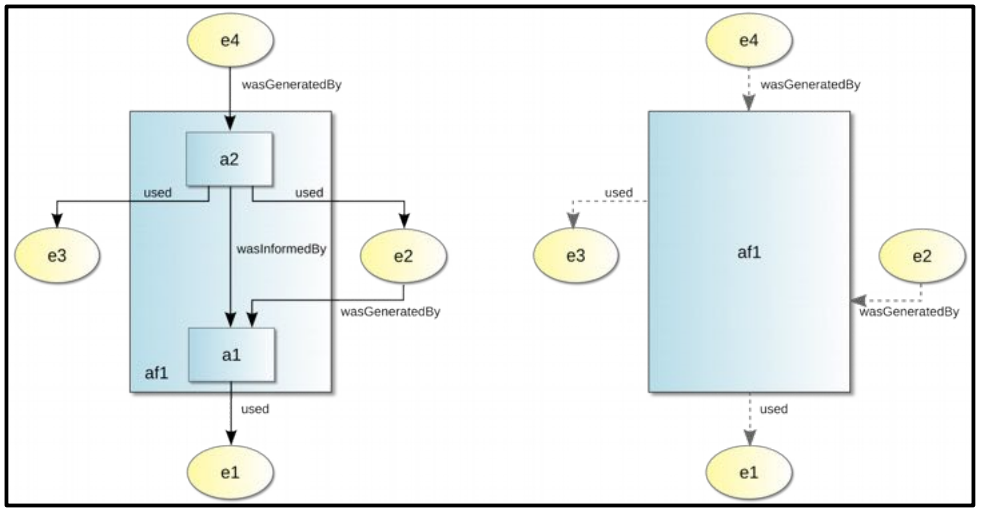
IVOA Provenance : Agent



IVOA Provenance : Entity



IVOA Provenance : Activity



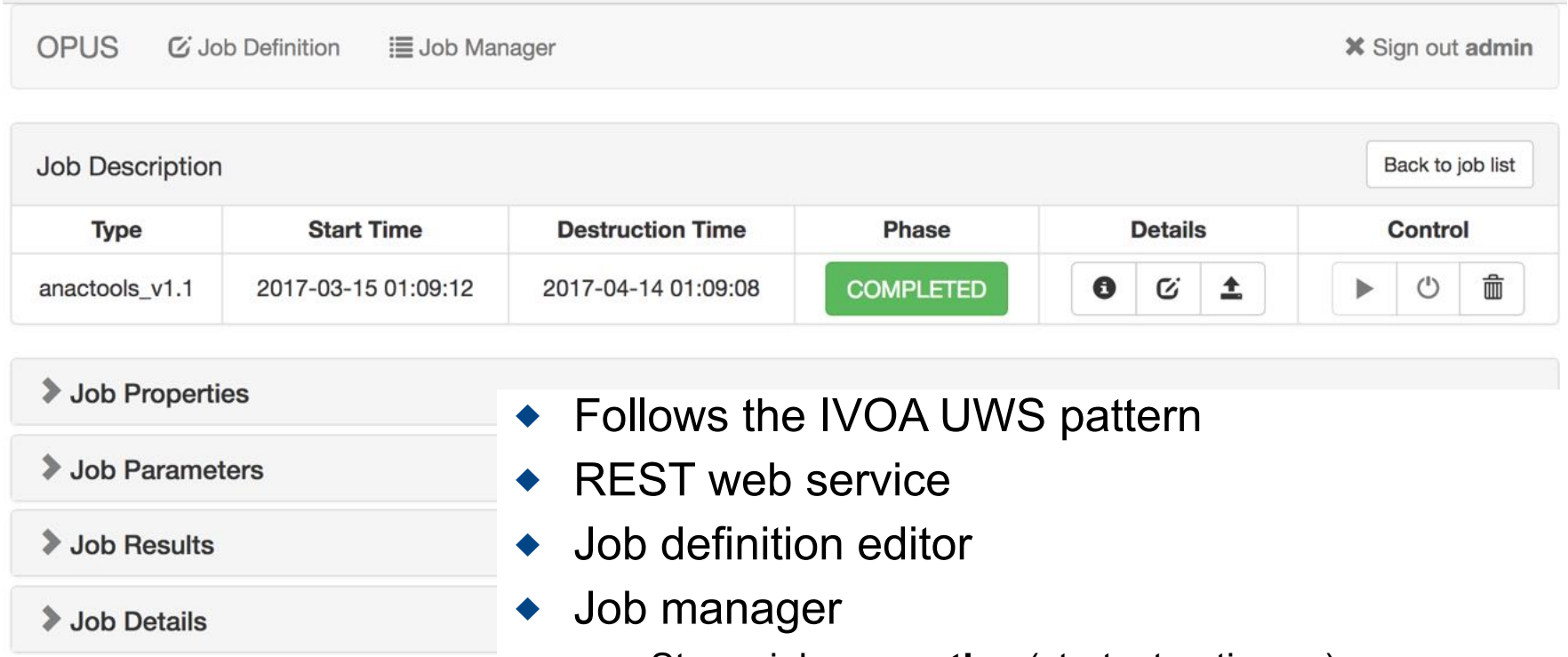
(Do not exist in W3C PROV)

VOTable serialization

```
<VOTABLE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ivoa.net/xml/VOTable/v1.2" version="1.2"
 xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/v1.2">
 <RESOURCE name="Stage1">
  <TABLE name="activities" utype="prov:activity">
    <FIELD name="name" utype="prov:activity.name" datatype="char" arraysize="*"/>
    <FIELD name="start" utype="prov:startTime" datatype="char" arraysize="*" xtype="ISO8601"/>
    <FIELD name="stop" utype="prov:endTime" datatype="char" arraysize="*" xtype="ISO8601"/>
    <FIELD name="methodname" utype="voprov:method_name" dataype="char" arraysize="*"/>
    <FIELD name="version" utype="voprov:method_version" datatype="char" arraysize="*"/>
   <DATA>
    <TABLEDATA>
     <TR>
      <TD>cta:telescope_stage_520</TD>
      <TD>2015-07-30T09:45:00</TD>
      <TD>2015-07-30T10:00:00</TD>
      <TD>Telescope_stage</TD>
      <TD>1.0</TD>
     </TR>
    </TABLEDATA>
   </DATA>
  </TABLE>
  <TABLE name="entities" utype="prov:entity">
    <FIELD name="name" utype="prov:entitity.name" datatype="char" arraysize="*"/>
    <FIELD name="label" utype="prov:label" datatype="char" arraysize="*"/>
    <FIELD name="type" utype="prov:type" datatype="char" arraysize="*"/>
    <FIELD name="run" utype="cta:runNumber" datatype="int"/>
    <FIELD name="tel" utype="cta:telescope" datatype="char" arraysize="*"/>
   <DATA>...</DATA>
  </TABLE>
  <TABLE name="usedRelationship" utype="voprov:used">
    <FIELD name="head" datatype="char" arraysize="*"/>
    <FIELD name="tail" datatype="char" arraysize="*"/>
   <DATA>...</DATA>
  </TABLE>
  <TABLE name="wasGeneratedByRelationship" utype="voprov:wasGeneratedBy">
    <FIELD name="head" datatype="char" arraysize="*"/>
    <FIELD name="tail" datatype="char" arraysize="*"/>
   <DATA>...</DATA>
  </TABLE>
 </RESOURCE>
</VOTABLE>
```

Example 1: analysis step with OPUS

- **OPUS** (Observatoire de Paris UWS Server) is a light job controller for the Paris Observatory work cluster developped in Python : https://github.com/ParisAstronomicalDataCentre/OPUS



- Follows the IVOA UWS pattern
- REST web service
- Job definition editor
- Job manager
  - Stores job **properties** (start, stop time…)
  - **Parameter** also stored
  - Access to **results**
  - Visualization of **logs** and **Provenance information**

Collecting Provenance information

OPUS

- Using **UWS**
- Database
  - Jobs
  - Parameters
  - Results
- Need a **job description** to expose provenance information

```
<uws:job xmlns:uws="http://www.ivoa.net/xml/UWS/v1.0" xmlns:xli
  <uws:jobId> 3745c408-8f39-404b-9982-d5b1116ad639 </uws:jobId>
  <uws:phase> COMPLETED </uws:phase>
  <uws:executionDuration> 300 </uws:executionDuration>
  <uws:quote> 120 </uws:quote>
  <uws:error xsi:nil="true" />
  <uws:startTime> 2017-03-15T01:09:12 </uws:startTime>
  <uws:endTime> 2017-03-15T01:10:05 </uws:endTime>
  <uws:destruction> 2017-04-14T01:09:08 </uws:destruction>
  <uws:ownerId> admin </uws:ownerId>
  <uws:parameters>
    <uws:parameter byReference="false" id="anatype"> unbinned </
    <uws:parameter byReference="false" id="run_numbers"> 23523+
    <uws:parameter byReference="false" id="edisp"> true </uws:pa
  </uws:parameters>
  <uws:results>
    <uws:result id="butterfly" xlink:href="https://voparis-uws-
    <uws:result id="stdout" xlink:href="https://voparis-uws-tes
    <uws:result id="spectrum" xlink:href="https://voparis-uws-t
    <uws:result id="fit_results" xlink:href="https://voparis-uw
    <uws:result id="configfile" xlink:href="https://voparis-uws
```

http://www.ivoa.net/documents/UWS/

ActivityDescription serialization

- VOTable based on Datalink service descriptor

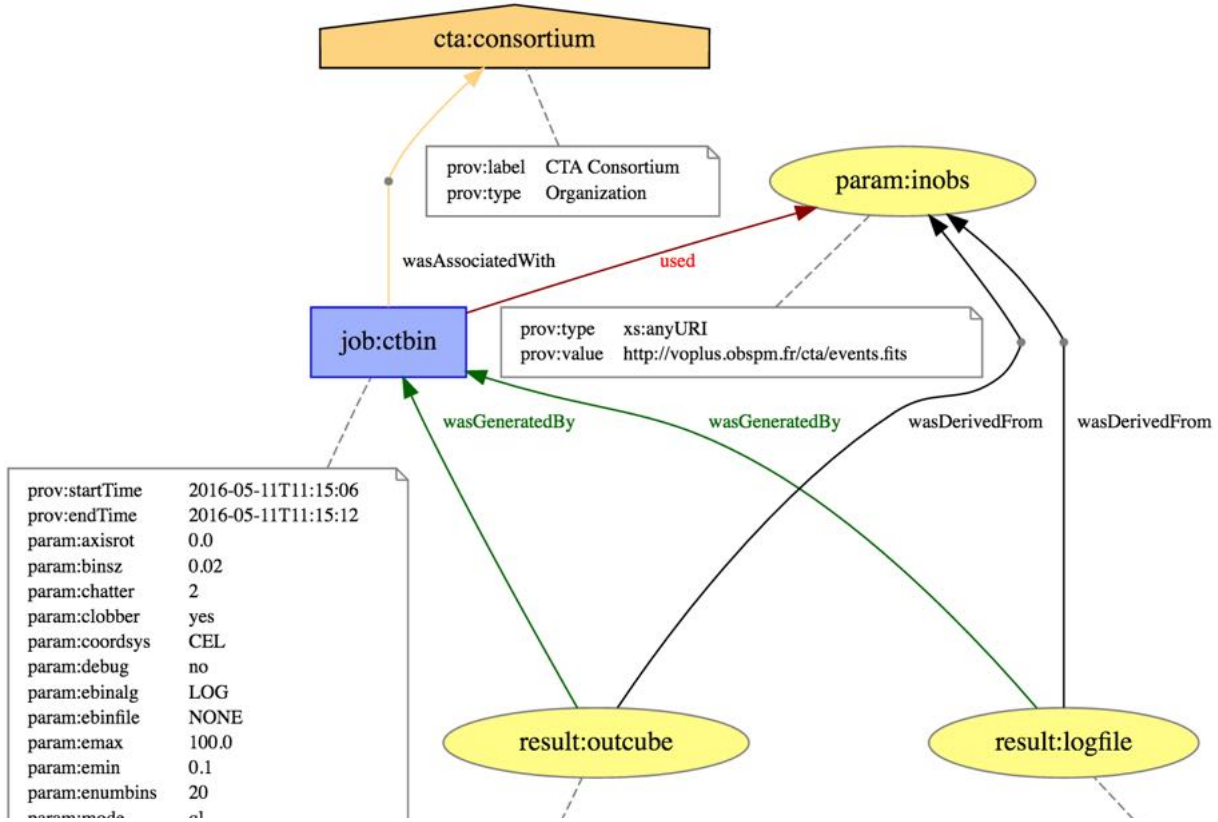
```
<VOTABLE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ivoa.net/xml/VOTable/v1.3" version="1.3"
 xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.3 http://www.ivoa.net/xml/VOTable/v1.3">
 <RESOURCE ID="ctbin" name="ctbin" type="meta" utype="voprov:ActivityDescription">
   <!--  Job description  -->
  <DESCRIPTION>...</DESCRIPTION>
   <PARAM name="label" datatype="char" arraysize="*" value="CTOOLS ctbin job" utype="voprov:ActivityDescription.label"/>
   <PARAM name="type" datatype="char" arraysize="*" value="Analysis" utype="voprov:ActivityDescription.type"/>
   <PARAM name="subtype" datatype="char" arraysize="*" value="Binning" utype="voprov:ActivityDescription.subtype"/>
   <PARAM name="version" datatype="float" value="1.0" utype="voprov:ActivityDescription.version"/>
   <PARAM name="doculink" datatype="char" arraysize="*" value="http://cta.irap.omp.eu/ctools/reference_manual/ctbin.html"
   utype="voprov:ActivityDescription.doculink"/>
   <PARAM name="contact_name" datatype="char" arraysize="*" value="CTOOLS Helpdesk" utype="voprov:Agent.name"/>
   <PARAM name="contact_email" datatype="char" arraysize="*" value="ctools@irap.omp.eu" utype="voprov:Agent.email"/>
   <PARAM name="executionduration" datatype="int" value="5" utype="uws:Job.executionduration"/>
   <PARAM name="quote" datatype="int" value="5" utype="uws:Job.quote"/>
   <!--  Job parameters  -->
  <GROUP name="InputParams" utype="voprov:Parameter">
    <!--  General parameters  -->
```

- Adding information on used/generated entities

```
   <!--  Used entities  -->
  <GROUP name="Used" utype="voprov:Used">
    <PARAM name="inobs" ref="inobs" datatype="char" arraysize="*" value="" xtype="image/fits" utype="voprc
    <PARAM name="ebinfile" ref="ebinfile" datatype="char" arraysize="*" value="" xtype="plain/text" utype=
   </GROUP>
   <!--  Generated entities / UWS results  -->
  <GROUP name="Generated" utype="voprov:WasGeneratedBy">
    <PARAM name="outcube" ref="outcube" datatype="char" arraysize="*" value="" xtype="image/fits" utype="v
    <PARAM name="logfile" ref="logfile" datatype="char" arraysize="*" value="" xtype="plain/text" utype="v
   </GROUP>
```

Provides Provenance files

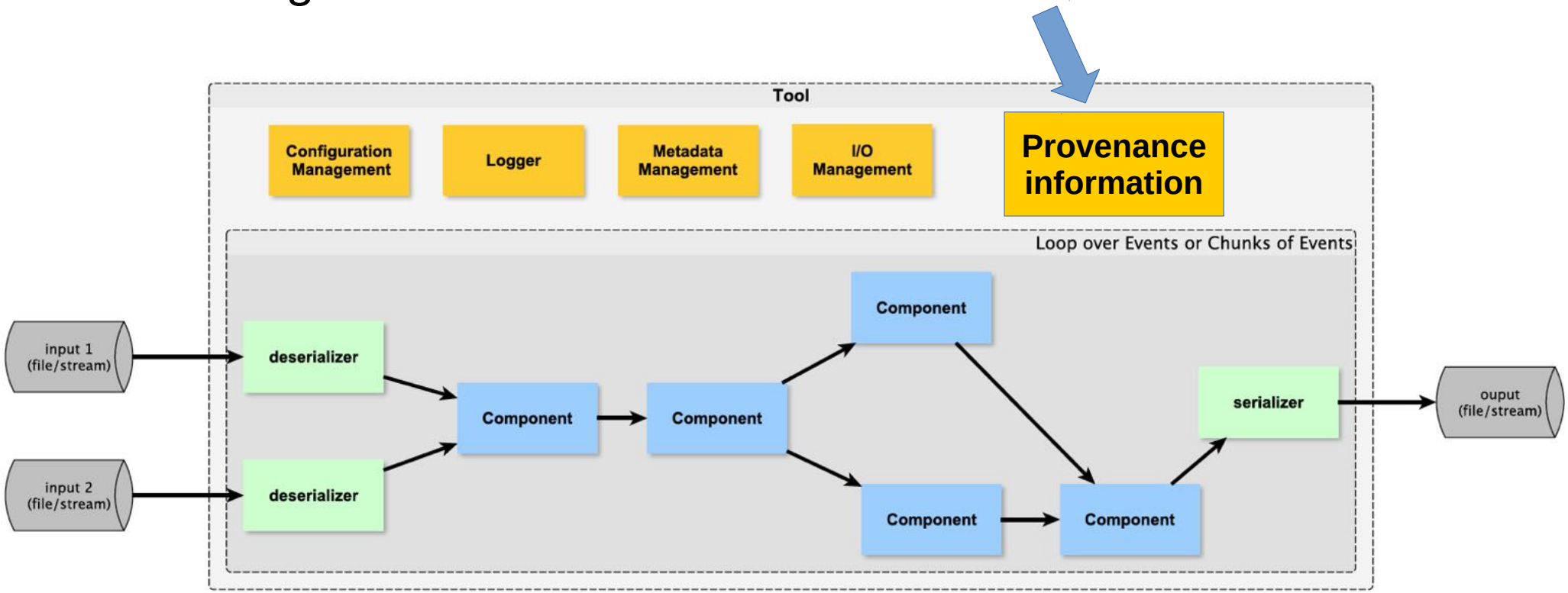
UWS database + Activity Description (VOTable) = Provenance Files (VOTable, XML, JSON, SVG)



Example 2: CTA Pipeline

cherenkov telescope array

- **Ctapipe**: a CTA data processing framework (prototype, not official, not recommended for use!) https://github.com/cta-observatory/ctapipe
- **Tool Python class** providing configuration, logger, I/O management… and **Provenance information**



Provenance class for ctapipe

```
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
```

- Importance of **persistent identifiers**
- Also records **system configuration**, **state**, and **software versions**