IVOA Provenance Data Model Version 1.0

IVOA Working Draft 2017-10-12

Working group

DM

This version

http://www.ivoa.net/documents/ProvenanceDM/20171012

Latest version

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

Previous versions

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

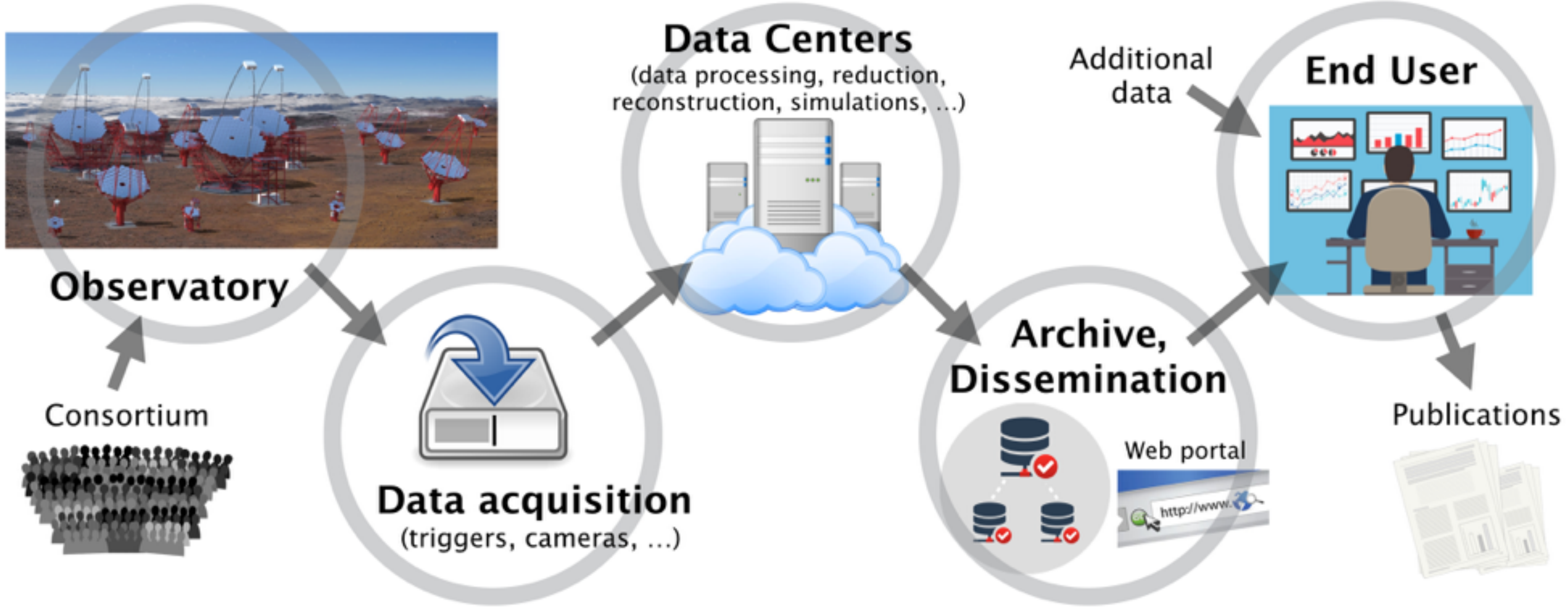
Author(s)

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

Editor(s)

Kristin Riebe, Mathieu Servillat

Objectives and context



- Data product generation **obscure** to end user
- **Quality**, **reliability**, **trustworthiness**?
- **Usefulness** of the data?

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

Goals

- **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**

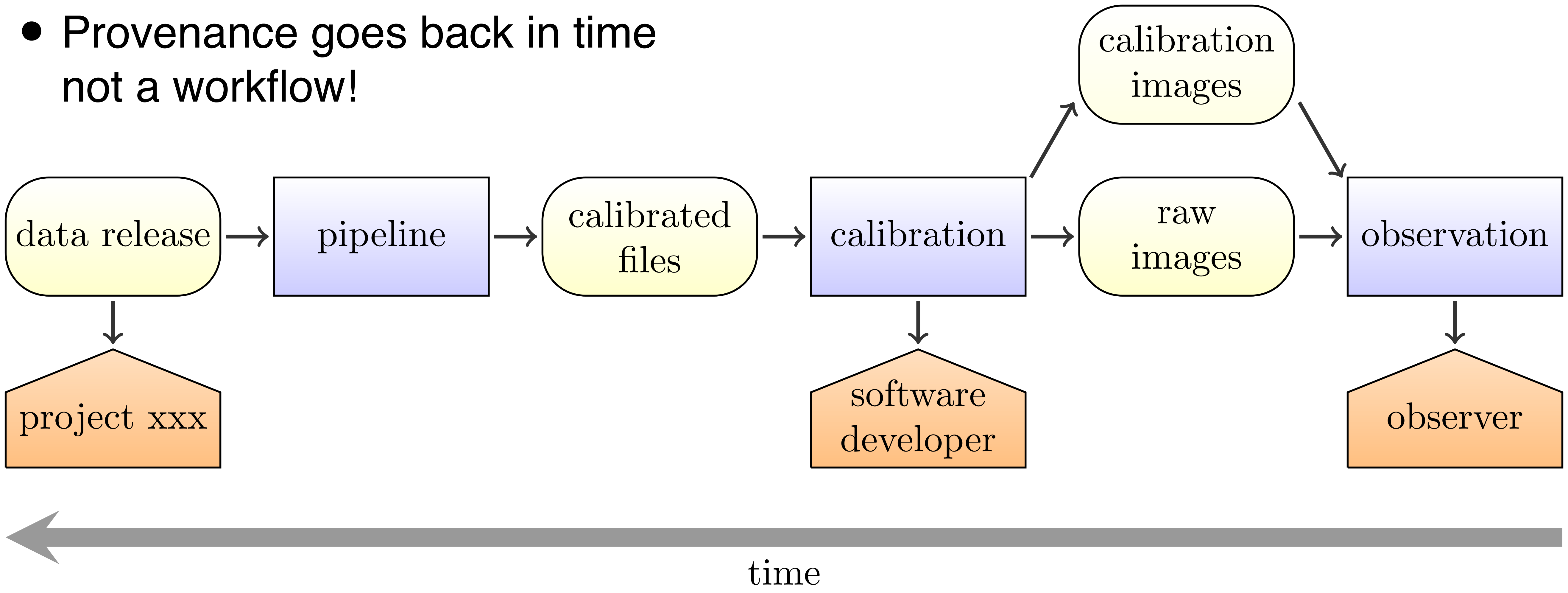
  Judge the quality of an observation, production step or dataset.

- **E: Search in structured provenance metadata**

  This would allow one to also do a “forward search”, i.e. locate derived datasets or outputs.

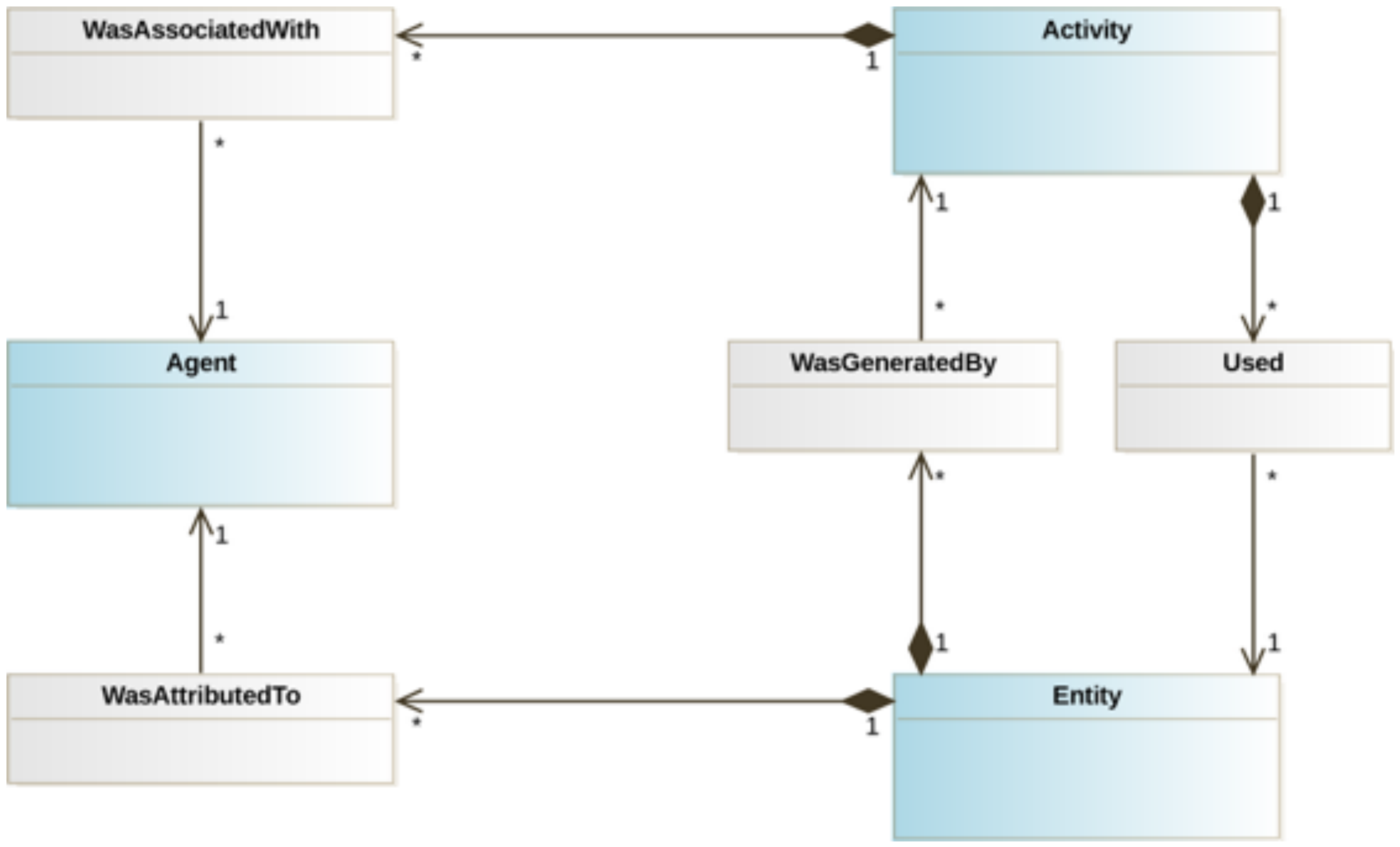
What is provenance?

- Provenance goes back in time
  not a workflow!



- **Provenance**: Identify how a data product was produced
- **Configuration**: Identify what detailed options were used
  - Instrument Configuration
- **Contextual** information:
  - Ambient Conditions
  - Software environment

Core Provenance Data Model



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

30 April 2013

Word Wide Web Consortium

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

- Core concepts from the W3C PROV recommendations
  - **Entity - Activity - Agent**
  - **Relations** and **roles** = provenance information
  - W3C PROV has many more relations
  - IVOA Provenance connected to **VO concepts** and **astronomy needs**

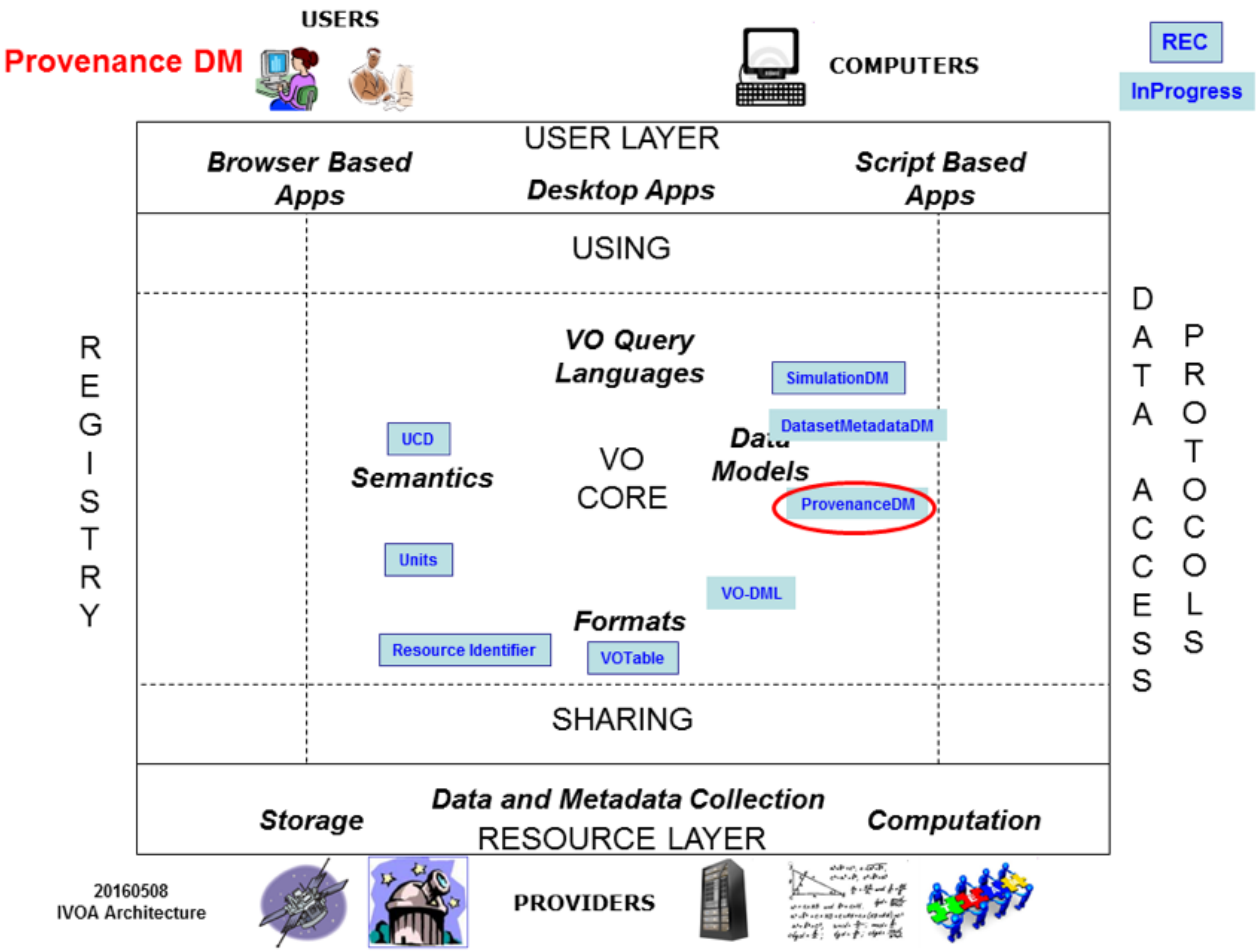
Concepts In Astronomy

- **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.).
- **Agents:** the people involved can be individual persons (observer, publisher...), groups or organisations.
- Connections to existing VO concepts
  - Entity <—> Dataset (Curation, DataID), ObsCore, SimDM DataObject
  - Activity <—> SimDM (Resource, Experiment)
  - Agent <—> Party, Contact
- Connections to external concepts (PROV, DOI, ORCID, …)

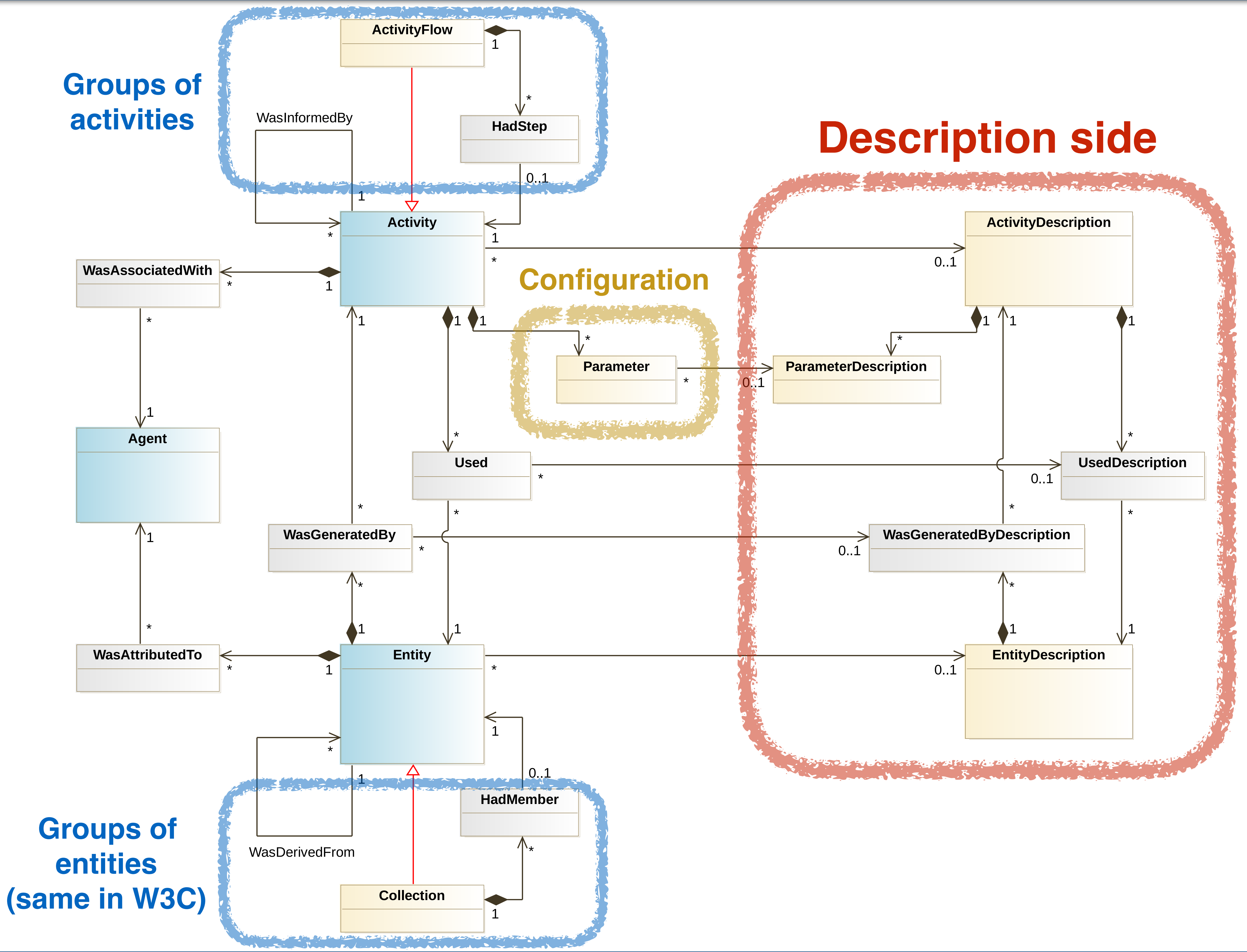
Minimum requirements

- 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.

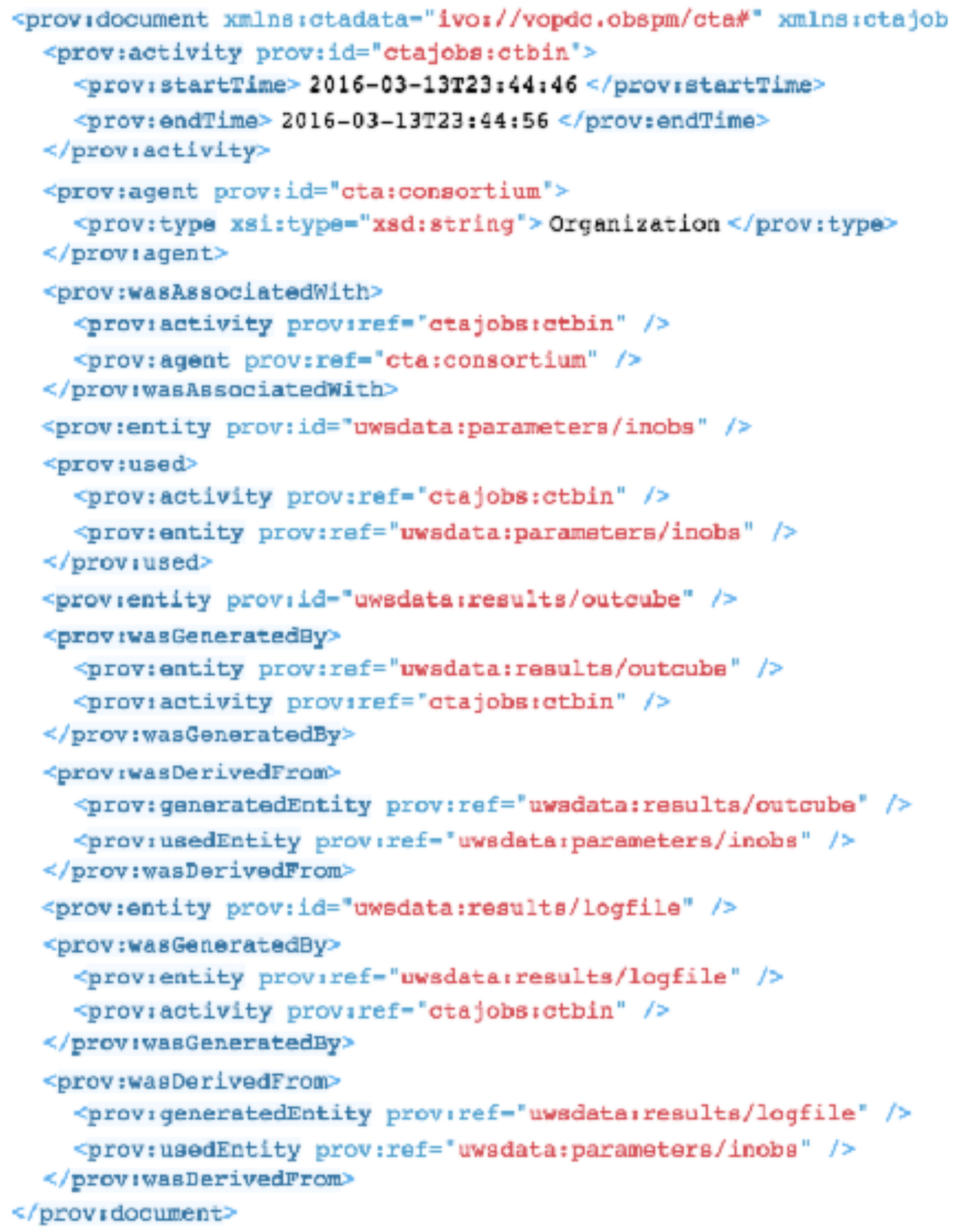
Role within the VO architecture



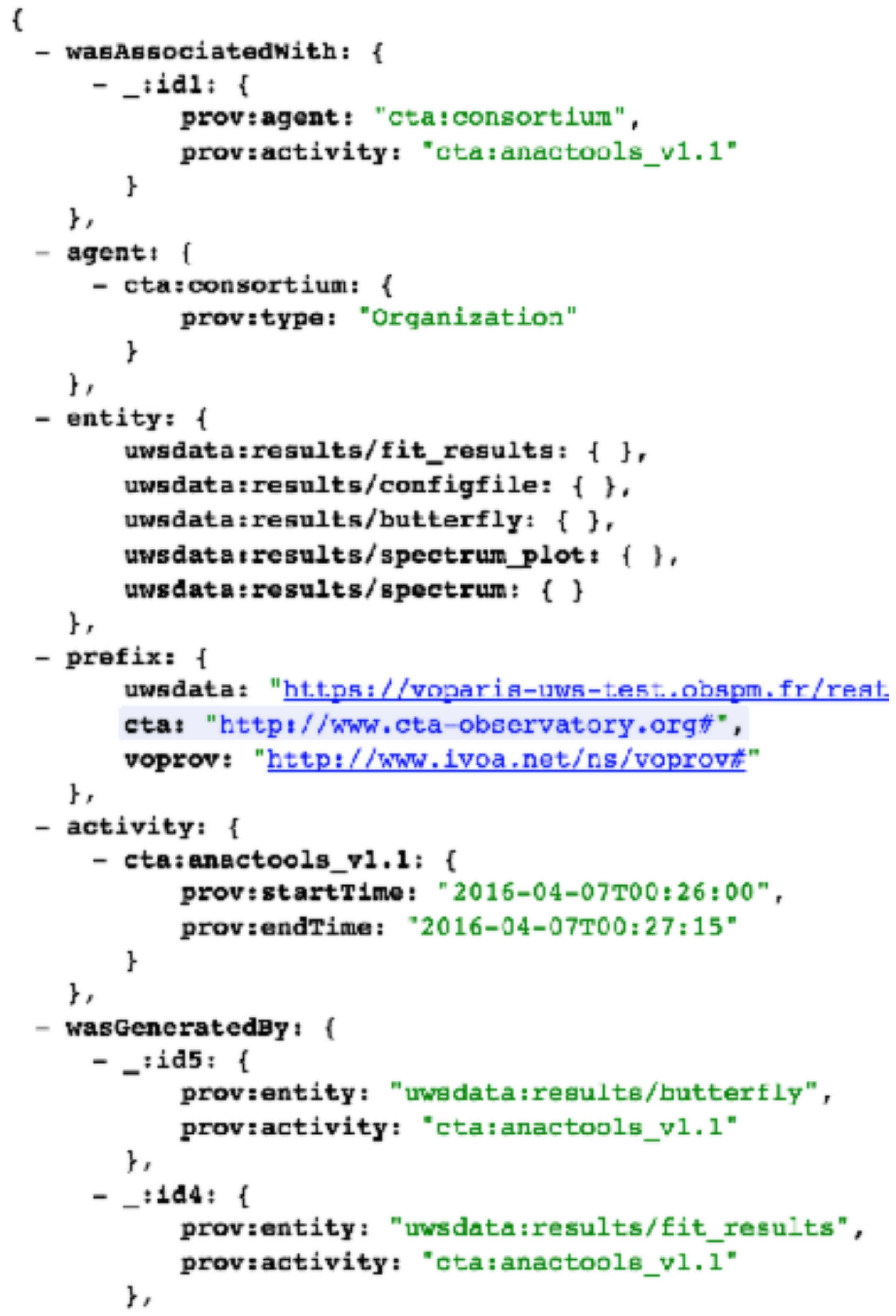
IVOA Provenance Data Model diagram



Serializations - W3C PROV formats

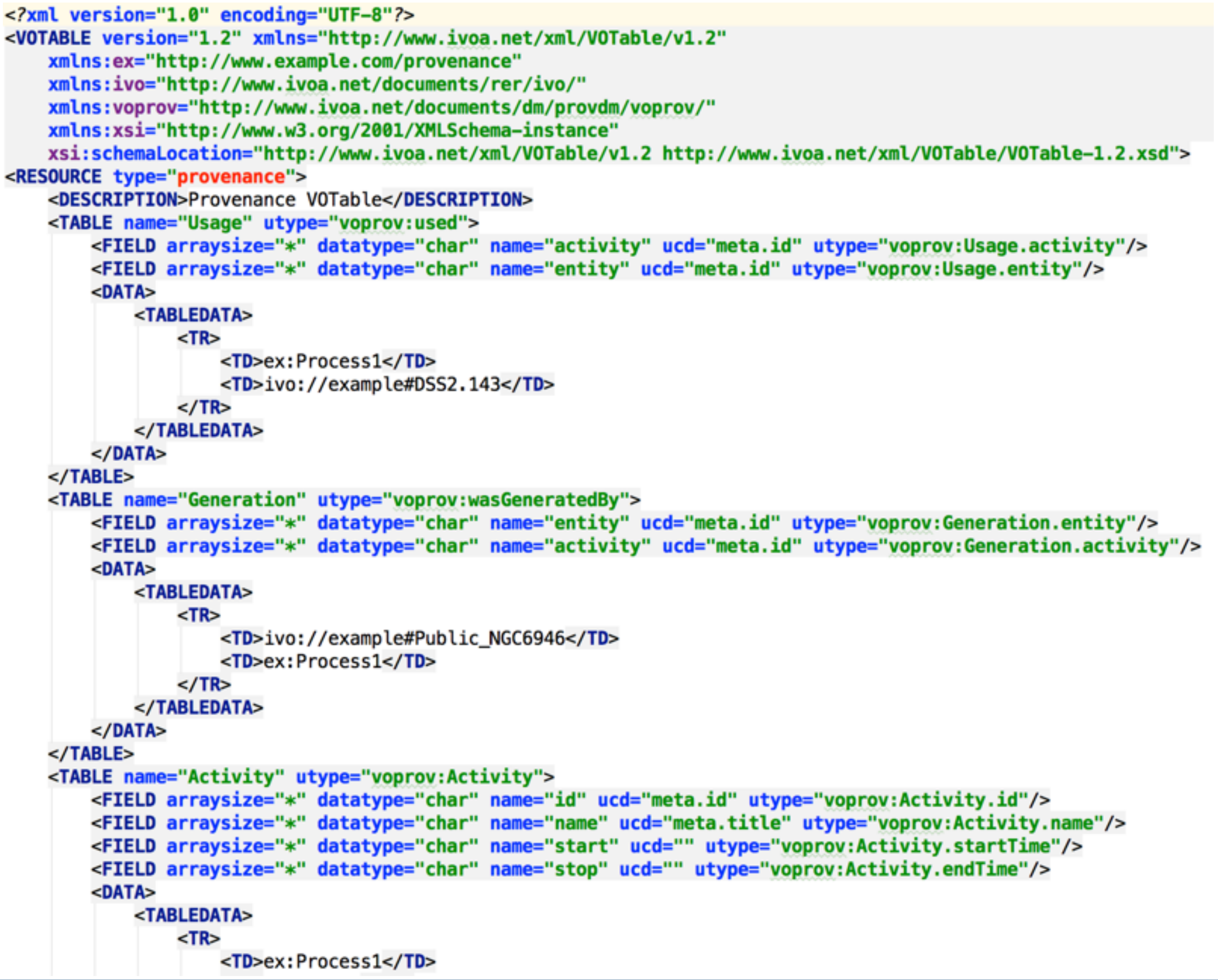


```
<prov:document xmlns:ctadata="ivo://vopdc.obspm/cta#" xmlns:ctajob
  <prov:activity prov:id="ctajobs:ctbin">
    <prov:startTime> 2016-03-13T23:44:46 </prov:startTime>
    <prov:endTime> 2016-03-13T23:44:56 </prov:endTime>
  </prov:activity>
  <prov:agent prov:id="cta:consortium">
    <prov:type 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>
```



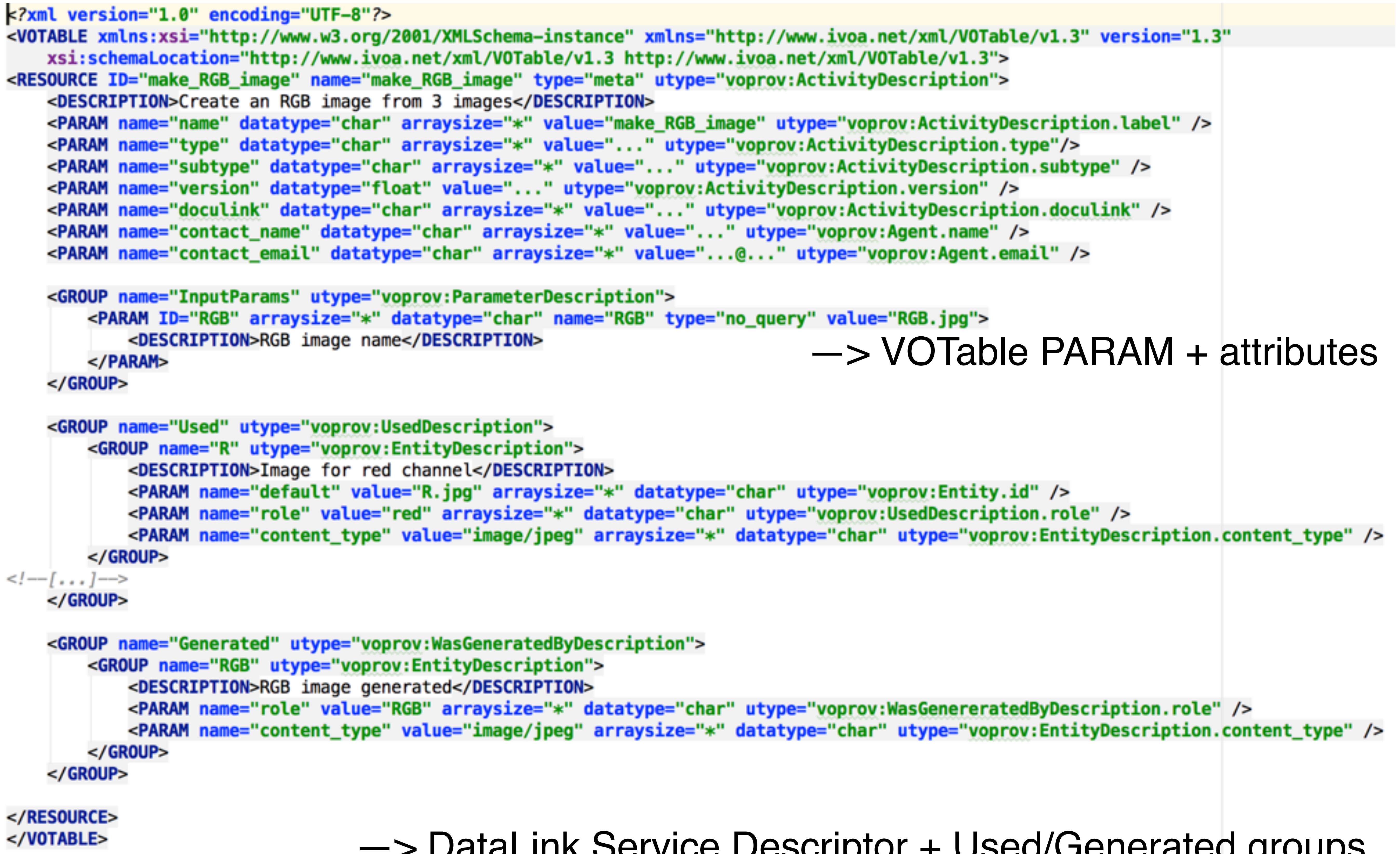
```
{
 - wasAssociatedWith: {
    - _:id1: {
          prov:agent: "cta:consortium",
          prov:activity: "cta:anactools_v1.1"
      }
   },
 - agent: {
    - cta:consortium: {
          prov:type: "Organization"
      }
   },
 - entity: {
       uwsdata:results/fit_results: { },
       uwsdata:results/configfile: { },
       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"
      },
```

Serializations - VOTable



```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
    xmlns:ex="http://www.example.com/provenance"
    xmlns:ivo="http://www.ivoa.net/documents/rer/ivo/"
    xmlns:voprov="http://www.ivoa.net/documents/dm/provdm/voprov/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd">
<RESOURCE type="provenance">
    <DESCRIPTION>Provenance VOTable</DESCRIPTION>
    <TABLE name="Usage" utype="voprov:used">
        <FIELD arraysize="*" datatype="char" name="activity" ucd="meta.id" utype="voprov:Usage.activity"/>
        <FIELD arraysize="*" datatype="char" name="entity" ucd="meta.id" utype="voprov:Usage.entity"/>
        <DATA>
            <TABLEDATA>
                <TR>
                    <TD>ex:Process1</TD>
                    <TD>ivo://example#DSS2.143</TD>
                </TR>
            </TABLEDATA>
        </DATA>
    </TABLE>
    <TABLE name="Generation" utype="voprov:wasGeneratedBy">
        <FIELD arraysize="*" datatype="char" name="entity" ucd="meta.id" utype="voprov:Generation.entity"/>
        <FIELD arraysize="*" datatype="char" name="activity" ucd="meta.id" utype="voprov:Generation.activity"/>
        <DATA>
            <TABLEDATA>
                <TR>
                    <TD>ivo://example#Public_NGC6946</TD>
                    <TD>ex:Process1</TD>
                </TR>
            </TABLEDATA>
        </DATA>
    </TABLE>
    <TABLE name="Activity" utype="voprov:Activity">
        <FIELD arraysize="*" datatype="char" name="id" ucd="meta.id" utype="voprov:Activity.id"/>
        <FIELD arraysize="*" datatype="char" name="name" ucd="meta.title" utype="voprov:Activity.name"/>
        <FIELD arraysize="*" datatype="char" name="start" ucd="" utype="voprov:Activity.startTime"/>
        <FIELD arraysize="*" datatype="char" name="stop" ucd="" utype="voprov:Activity.endTime"/>
        <DATA>
            <TABLEDATA>
                <TR>
                    <TD>ex:Process1</TD>
```

Serializations - ActivityDescription



```xml
<?xml version="1.0" encoding="UTF-8"?>
<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="make_RGB_image" name="make_RGB_image" type="meta" utype="voprov:ActivityDescription">
    <DESCRIPTION>Create an RGB image from 3 images</DESCRIPTION>
    <PARAM name="name" datatype="char" arraysize="*" value="make_RGB_image" utype="voprov:ActivityDescription.label" />
    <PARAM name="type" datatype="char" arraysize="*" value="..." utype="voprov:ActivityDescription.type"/>
    <PARAM name="subtype" datatype="char" arraysize="*" value="..." utype="voprov:ActivityDescription.subtype" />
    <PARAM name="version" datatype="float" value="..." utype="voprov:ActivityDescription.version" />
    <PARAM name="doculink" datatype="char" arraysize="*" value="..." utype="voprov:ActivityDescription.doculink" />
    <PARAM name="contact_name" datatype="char" arraysize="*" value="..." utype="voprov:Agent.name" />
    <PARAM name="contact_email" datatype="char" arraysize="*" value="...@..." utype="voprov:Agent.email" />

    <GROUP name="InputParams" utype="voprov:ParameterDescription">
        <PARAM ID="RGB" arraysize="*" datatype="char" name="RGB" type="no_query" value="RGB.jpg">
            <DESCRIPTION>RGB image name</DESCRIPTION>
        </PARAM>
    </GROUP>

    <GROUP name="Used" utype="voprov:UsedDescription">
        <GROUP name="R" utype="voprov:EntityDescription">
            <DESCRIPTION>Image for red channel</DESCRIPTION>
            <PARAM name="default" value="R.jpg" arraysize="*" datatype="char" utype="voprov:Entity.id" />
            <PARAM name="role" value="red" arraysize="*" datatype="char" utype="voprov:UsedDescription.role" />
            <PARAM name="content_type" value="image/jpeg" arraysize="*" datatype="char" utype="voprov:EntityDescription.content_type" />
        </GROUP>
<!--[...]-->
    </GROUP>

    <GROUP name="Generated" utype="voprov:WasGeneratedByDescription">
        <GROUP name="RGB" utype="voprov:EntityDescription">
            <DESCRIPTION>RGB image generated</DESCRIPTION>
            <PARAM name="role" value="RGB" arraysize="*" datatype="char" utype="voprov:WasGenereratedByDescription.role" />
            <PARAM name="content_type" value="image/jpeg" arraysize="*" datatype="char" utype="voprov:EntityDescription.content_type" />
        </GROUP>
    </GROUP>

</RESOURCE>
</VOTABLE>
```

—> VOTable PARAM + attributes

—> DataLink Service Descriptor + Used/Generated groups

Data Access

- We envision two possible access protocols, with distinct use cases :

  - **ProvDAL**: retrieve provenance information based on given ID of a data entity or activity.

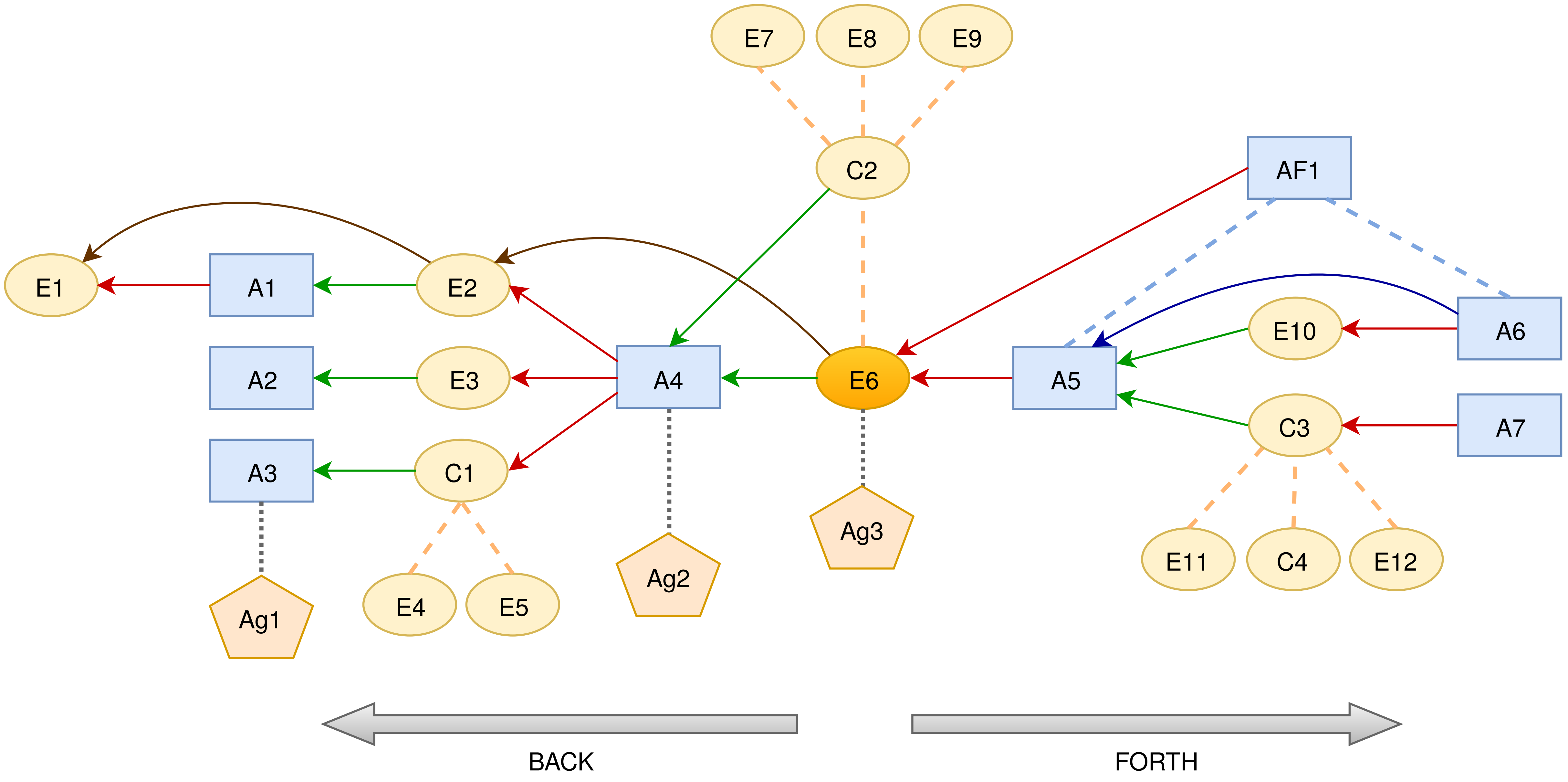
    ```
    {provdal-base-url}?ID=rave:dr4
    {provdal-base-url}?ID=rave:dr4&ID=rave:act_irafReduction
    ```

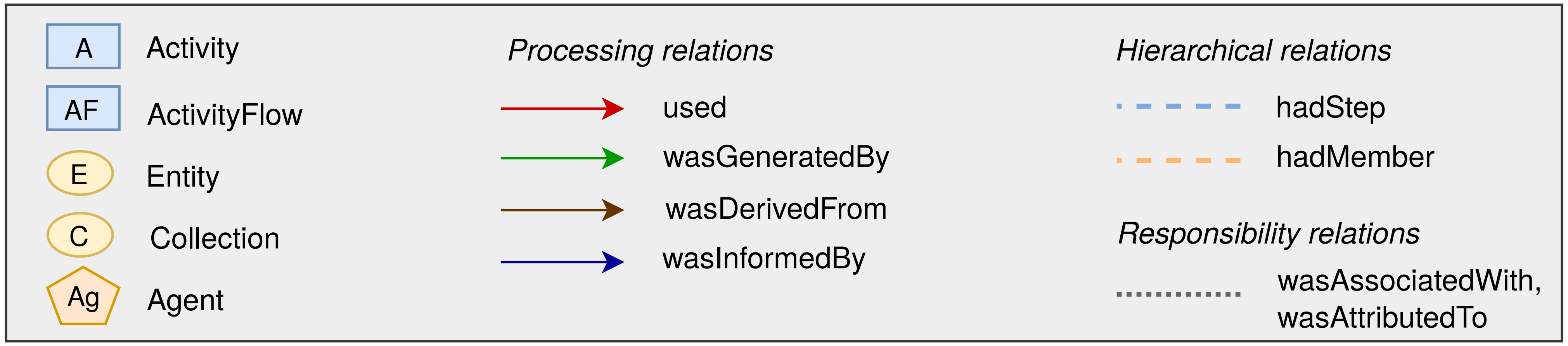
  - **ProvTAP**: allows detailed queries for provenance information, discovery of datasets based on e.g. code version.

    As for any TAP service (Dowler and Rixon et al., 2010), it is providing ADQL query responses consisting of a single table each, gathering columns selected from various tables defined in the service's TAP schema.

—> See presentations in DAL session Saturday 9-11h30

Simple Data Access



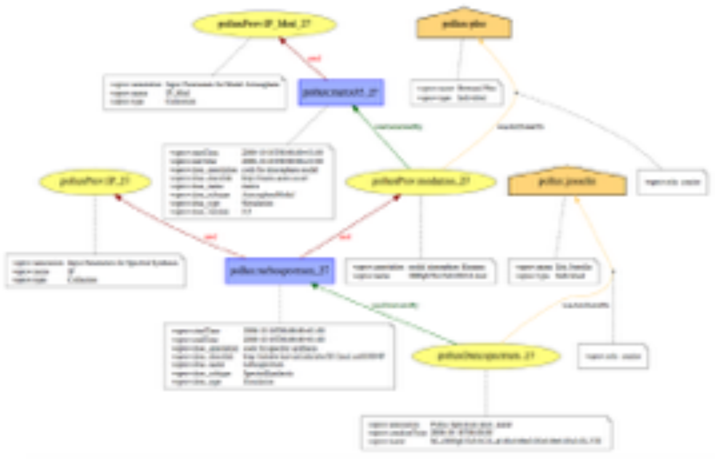


Use cases and implementations

VOPROV LIBRARY

The voprov package is an open source Python library which allows users to serialize their provenance information in different formats: PROV-N, JSON, XML, VOTable or in graphical ones: PNG, SVG,PDF.

(Cf https://github.com/sanguillon/voprov/)



This package is used in the context of Pollux.

Pollux is a stellar spectra database proposing access to high resolution synthetic spectra computed using the best available models of atmosphere and efficient spectral synthesis codes.

DJANGO PACKAGE

The Django provenance package is an open source Python package that can be reused in Django web applications for serving provenance information via a ProvDAL and a REST interface. The data model classes are directly mapped to tables in a relational database tables. It supports IVOA as well as W3C serializations into PROV-JSON and PROV-N formats.

(Cf https://github.com/kristinriebe/django-prov_vo)

This package is used in the context of RAVE.

The RAVE (RAdial Velocity Experiment) is a survey that observed the spectra of half a million stars from the southern hemisphere.

In a pipeline of several steps the data were calibrated, reduced and stellar properties were determined, which were then released in the form of star catalogues.

Prototype PostGreSQL database at CDS

In the CDS prototype we implemented a PostGreSQL database for Provenance information attached to image datasets. A database schema has been designed from the IVOA Provenance DM and implemented.

A set of images, together with their digitization and extraction steps, RGB color composition and HiPS generation activities are fed to the database. Various scenarii for querying and displaying the Provenance information have been tested. PROV-N, PROV-Json and PROV-VOTable formats for the response are provided for the query response.

A simple user interface allowing to select the main types of requests and to display the responses via W3C Prov software has been designed. It allows querying for various combinations of Provenance relationships in the database.

UWS Server at Observatoire de Paris

OPUS (Observatoire de Paris UWS System) is an open source job control system based on the IVOA UWS pattern.

It is developed in the context of the Cherenkov Telescope Array (CTA) project to test the execution of CTA data analysis tools on a work cluster.

It implements the concept of ActivityDescription files and provides the serialized provenance information as files for each executed job (see also ADASS Poster p3822).

(Cf https://github.com/mservillat/OPUS)

The CTA is the next generation ground-based very high energy gamma-ray instrument. It will serve as an open observatory providing data to a wide astrophysics community, with the requirement to propose self-described data products to users with detailed provenance information.

ADASS Poster 129 by Michèle Sanguillon + **ADASS** Poster 75 by Mathieu Servillat + **IVOA** Apps1 talk

How to use the data model?

- Before using the model
  We noticed that the simple knowledge of what is provenance information is important for the conception of all projects.
- Define unique identifiers
  Use qualified names, think about IVOID, DOI, ORCID, …
- Use of the description classes are they really needed for your project?
- Add project specific attributes to your entities, activities and agents
- Group common features for entities and/or activities
- Create ActivityDescription files
- Link to an Authentication System external link
- Adding additional metadata to an existing Entity external link

Next evolutions to be discussed

1 Introduction

1.1 Goal of the provenance model

1.2 Minimum requirements for provenance

1.3 Role within the VO architecture

1.4 Previous efforts

2 The provenance data model

2.1 Overview: Conceptional UML class diagram and introduction to core classes

2.2 Model description

2.2.1 Class diagram and VO-DML compatibility

2.2.2 Entity and EntityDescription

2.2.3 Collection

2.2.4 Activity and ActivityDescription

2.2.5 ActivityFlow

2.2.6 Entity-Activity relations

2.2.7 Parameters

2.2.8 Agent

3 Links to other data models

3.1 Links with Dataset/ObsCore Model

3.2 Links with Simulation Data Model

4 Serialization of the provenance data model

4.1 Introduction

4.2 Serialization formats: PROV-N, PROV-JSON and PROV-XML

4.3 PROV-VOTable format

4.4 Serialization of description classes in the data processing context

4.5 W3C PROV-DM compatible serializations

5 Accessing provenance information

5.1 Access protocols

5.2 ProvDAL

5.2.1 ProvDAL example use cases

5.3 ProvTAP

5.4 VOSI availability and capabilities

6 Use cases – applying the data model

6.1 How to use the data model

6.2 voprov Python package

6.3 Provenance of RAVE database tables

6.4 Provenance for CTA

6.5 Provenance for the POLLUX database

6.6 Provenance of HiPS datasets

Appendices

Next evolutions to be discussed

1 Introduction

1.1 Goal of the provenance model

1.2 Minimum requirements for provenance

1.3 Role within the VO architecture

1.4 Previous efforts

2 The provenance data model

2.1 Overview: Conceptional UML class diagram and introduction to core classes

2.2 Model description

2.2.1 Class diagram and VO-DML compatibility

2.2.2 Entity and EntityDescription

2.2.3 Collection

2.2.4 Activity and ActivityDescription

2.2.5 ActivityFlow

2.2.6 Entity-Activity relations → WasDerivedFrom? WasInformedBy?

2.2.7 Parameters

2.2.8 Agent

3 Links to other data models

3.1 Links with Dataset/ObsCore Model

3.2 Links with Simulation Data Model

Moved to the Appendices

4 Serialization of the provenance data model

4.1 Introduction

4.2 Serialization formats: PROV-N, PROV-JSON and PROV-XML

~~4.3 PROV-VOTable format~~ comes with ProvTAP

4.4 Serialization of description classes in the data processing context

4.5 W3C PROV-DM compatible serializations

5 Accessing provenance information

5.1 Access protocols

5.2 ProvDAL

5.2.1 ProvDAL example use cases

5.3 ProvTAP

5.4 VOSI availability and capabilities

—> DAL document

6 Use cases – applying the data model

6.1 How to use the data model

6.2 voprov Python package

6.3 Provenance of RAVE database tables

6.4 Provenance for CTA

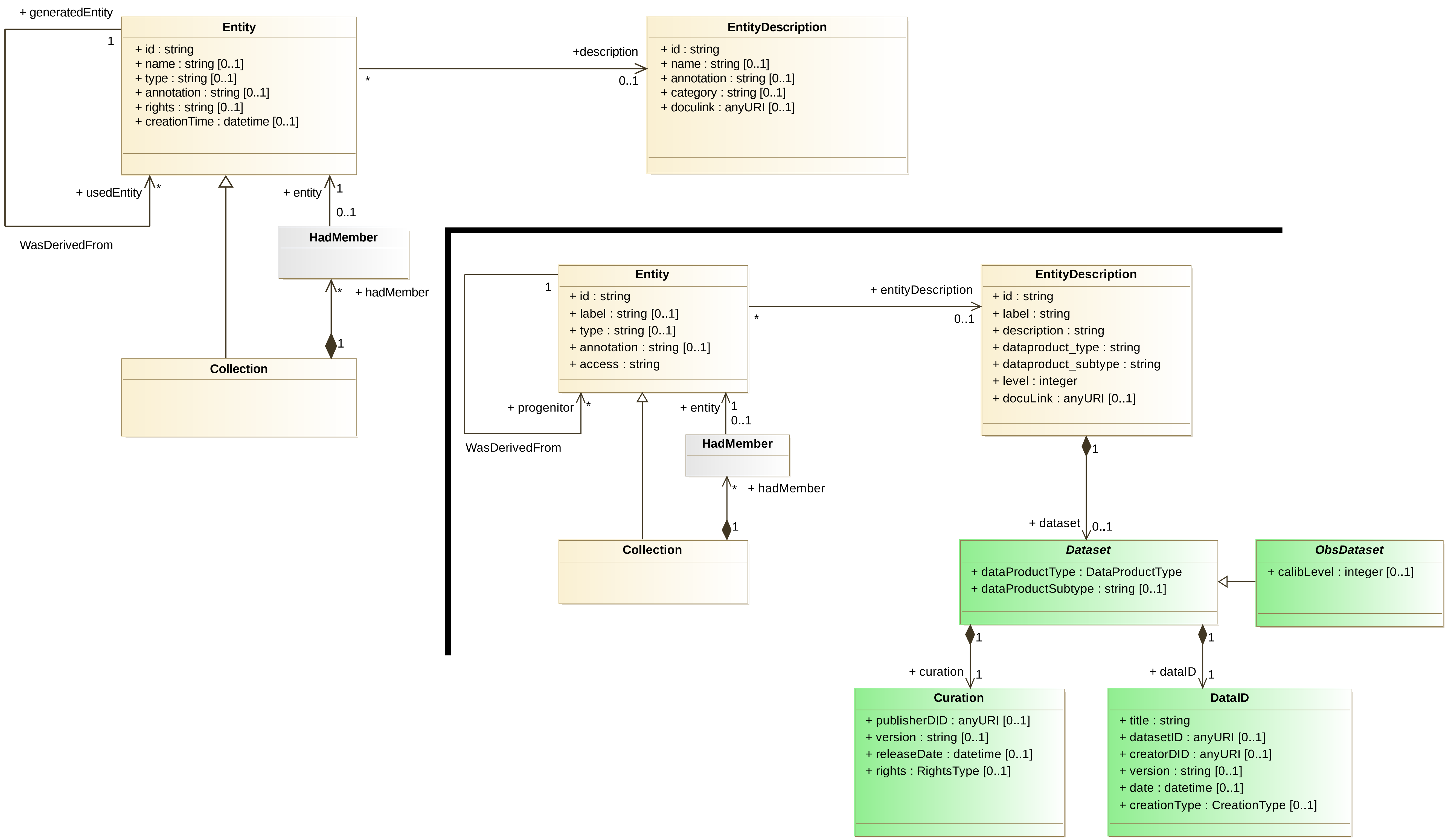
6.5 Provenance for the POLLUX database

6.6 Provenance of HiPS datasets

Appendices —> Implementation note

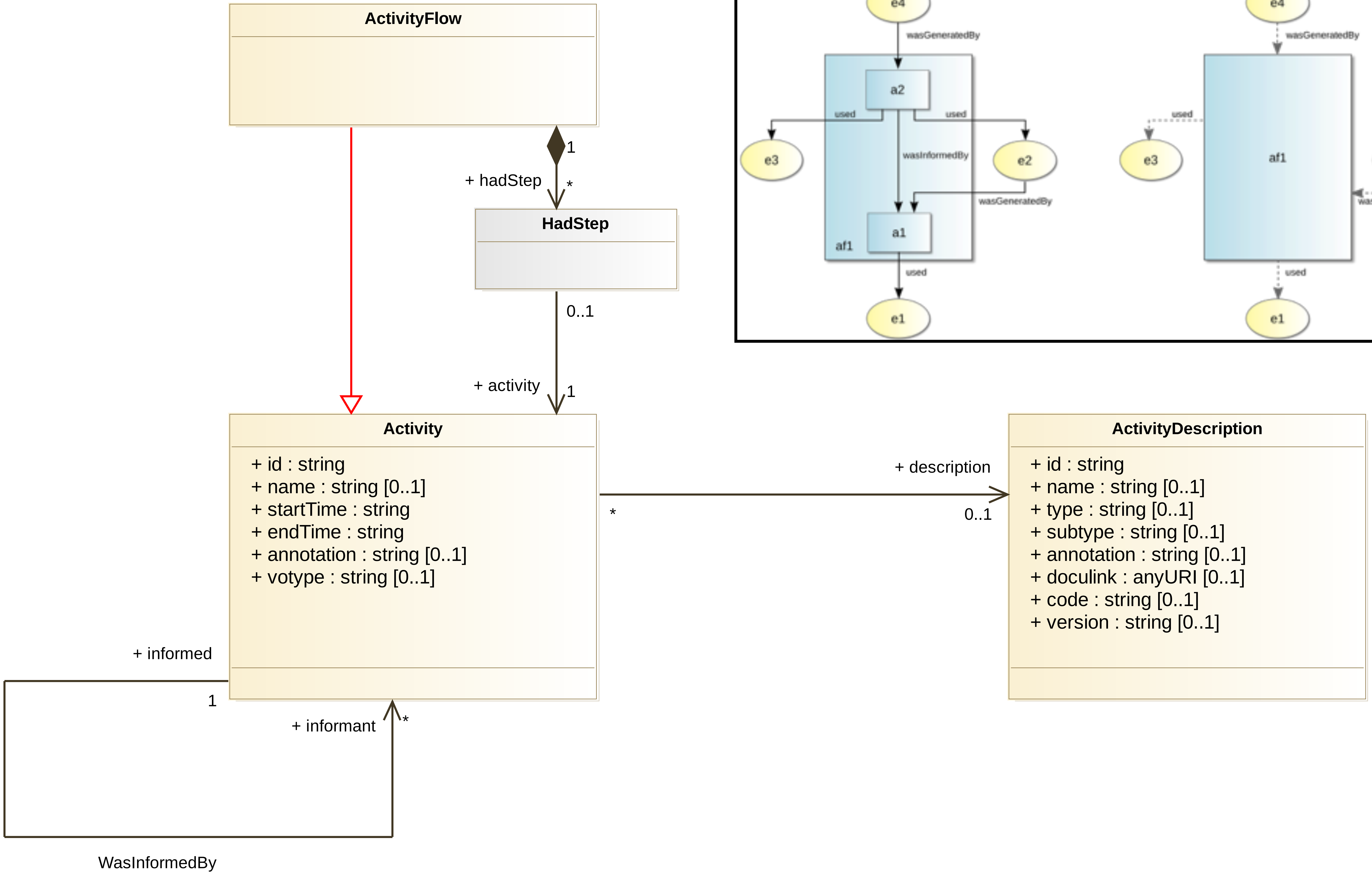
+ consistent VO vocabulary, map with external ID

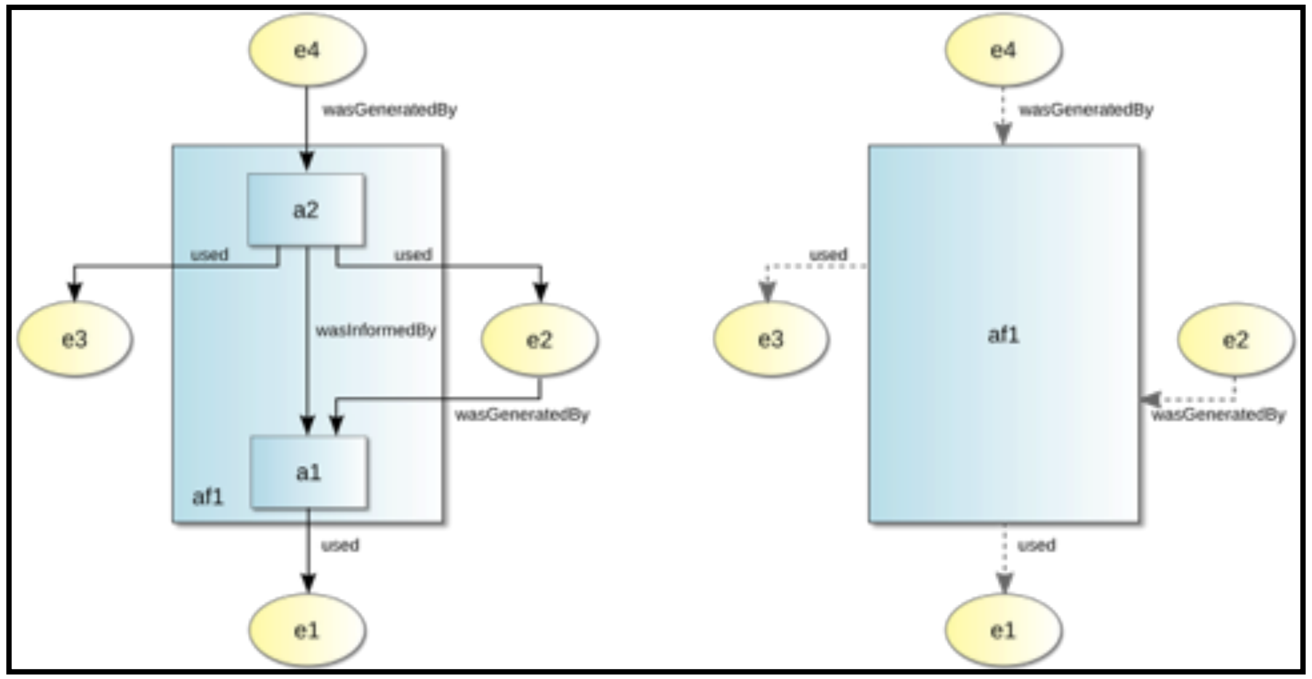
Entity



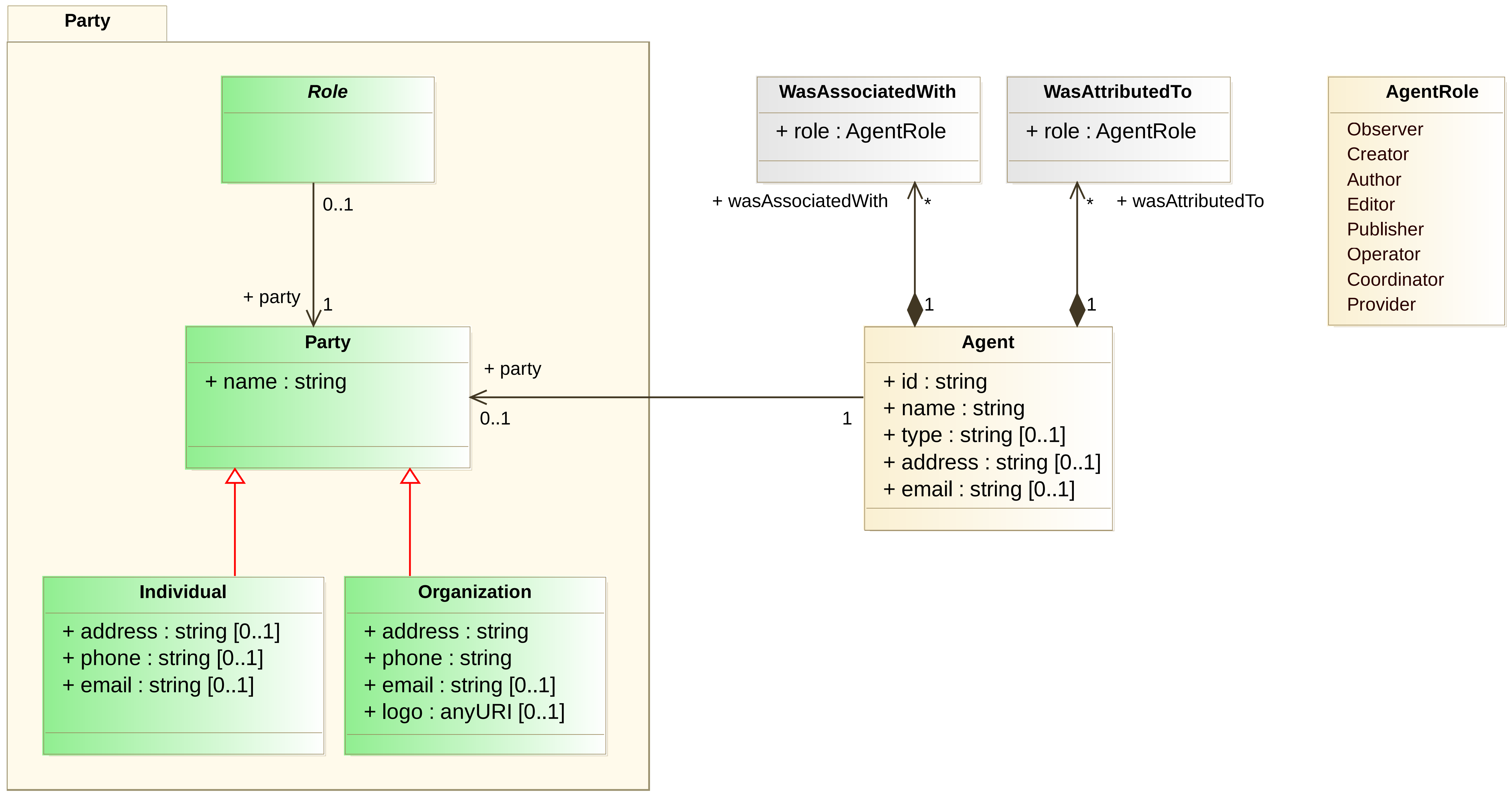
+ generatedEntity
1
Entity
+ id : string
+ name : string [0..1]
+ type : string [0..1]
+ annotation : string [0..1]
+ rights : string [0..1]
+ creationTime : datetime [0..1]
+description
*
0..1
EntityDescription
+ id : string
+ name : string [0..1]
+ annotation : string [0..1]
+ category : string [0..1]
+ doculink : anyURI [0..1]
+ usedEntity
*
+ entity
1
0..1
WasDerivedFrom
HadMember
*
+ hadMember
1
Collection
Entity
1
+ id : string
+ label : string [0..1]
+ type : string [0..1]
+ annotation : string [0..1]
+ access : string
+ entityDescription
*
0..1
EntityDescription
+ id : string
+ label : string
+ description : string
+ dataproduct_type : string
+ dataproduct_subtype : string
+ level : integer
+ docuLink : anyURI [0..1]
+ progenitor
*
+ entity
1
0..1
WasDerivedFrom
HadMember
*
+ hadMember
1
Collection
1
+ dataset
0..1
Dataset
+ dataProductType : DataProductType
+ dataProductSubtype : string [0..1]
ObsDataset
+ calibLevel : integer [0..1]
1
1
+ curation
1
+ dataID
1
Curation
+ publisherDID : anyURI [0..1]
+ version : string [0..1]
+ releaseDate : datetime [0..1]
+ rights : RightsType [0..1]
DataID
+ title : string
+ datasetID : anyURI [0..1]
+ creatorDID : anyURI [0..1]
+ version : string [0..1]
+ date : datetime [0..1]
+ creationType : CreationType [0..1]

Activity


ActivityFlow
1
+ hadStep
*
HadStep
0..1
+ activity
1
Activity
+ id : string
+ name : string [0..1]
+ startTime : string
+ endTime : string
+ annotation : string [0..1]
+ votype : string [0..1]
+ description
*
0..1
ActivityDescription
+ id : string
+ name : string [0..1]
+ type : string [0..1]
+ subtype : string [0..1]
+ annotation : string [0..1]
+ doculink : anyURI [0..1]
+ code : string [0..1]
+ version : string [0..1]
+ informed
1
+ informant
*
WasInformedBy


e4
wasGeneratedBy
a2
used
used
e3
wasInformedBy
e2
wasGeneratedBy
a1
af1
used
e1
e4
wasGeneratedBy
used
e3
af1
e2
wasGeneratedBy
used
e1

Agent


Party
Role
0..1
+ party
1
Party
+ name : string
Individual
+ address : string [0..1]
+ phone : string [0..1]
+ email : string [0..1]
Organization
+ address : string
+ phone : string
+ email : string [0..1]
+ logo : anyURI [0..1]
+ party
0..1
1
WasAssociatedWith
+ role : AgentRole
WasAttributedTo
+ role : AgentRole
+ wasAssociatedWith
*
1
*
1
+ wasAttributedTo
Agent
+ id : string
+ name : string
+ type : string [0..1]
+ address : string [0..1]
+ email : string [0..1]
AgentRole
Observer
Creator
Author
Editor
Publisher
Operator
Coordinator
Provider