# ProvTAP : restart and WorkingDraft

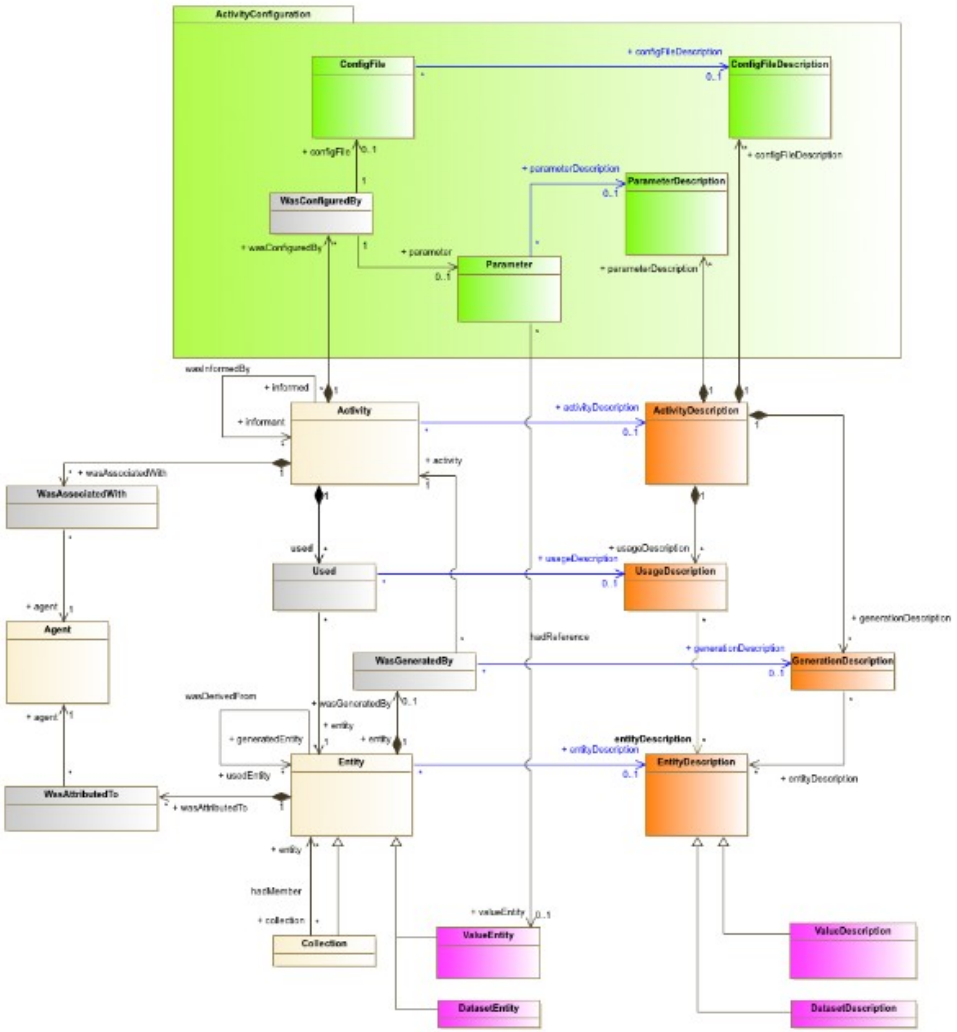
F.Bonnarel (CDS/ObAS), M.Servillat (Luth/ObsPM) L.Michel (Galhecos/ObAS)

CENTRE DE DONNÉES
ASTRONOMIQUES DE STRASBOURG

ESCAPE
European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

IVOA

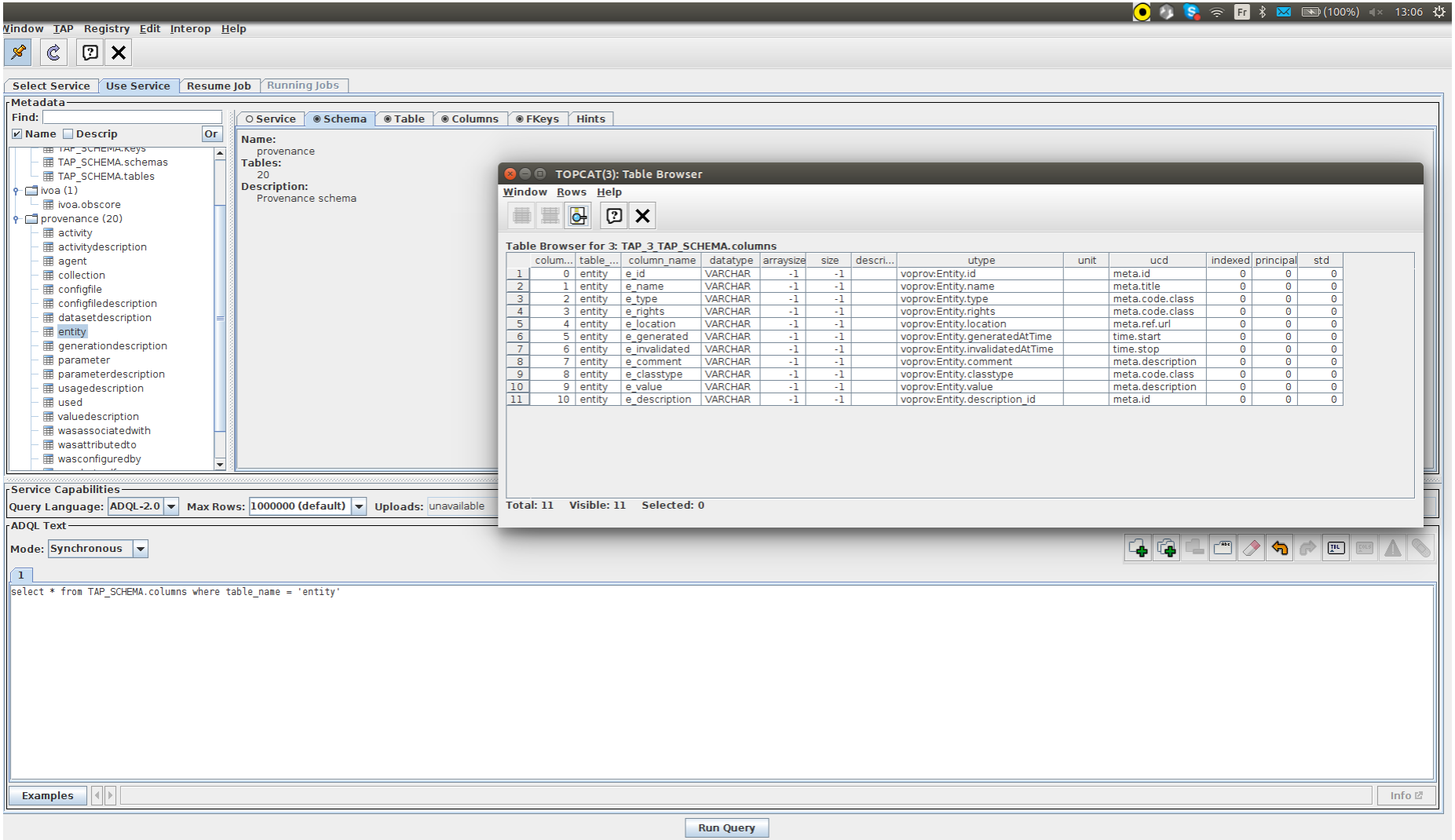
# IVOA provenance datamodel:
# - a lot of additions to entity/activity/agent basics
# - issue for TAP : this is a loop graph



# ProvTAP

- ProvTAP specification is mapping IVOA Provenance data model onto the TAP schema.
- The TAP schema defines
  - table and columns names,
  - Datatypes,
  - Units,
  - Ucds
  - and utypes (model attributed id)
  - → for each model feature !

# TAP SCHEMA = Entity table

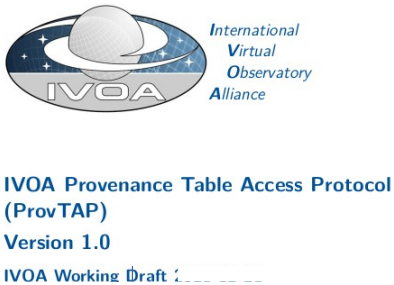



| | colum... | table_... | column_name | datatype | arraysize | size | descri... | utype | unit | ucd | indexed | principal | std |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | entity | e_id | VARCHAR | -1 | -1 | | voprov:Entity.id | | meta.id | 0 | 0 | 0 |
| 2 | 1 | entity | e_name | VARCHAR | -1 | -1 | | voprov:Entity.name | | meta.title | 0 | 0 | 0 |
| 3 | 2 | entity | e_type | VARCHAR | -1 | -1 | | voprov:Entity.type | | meta.code.class | 0 | 0 | 0 |
| 4 | 3 | entity | e_rights | VARCHAR | -1 | -1 | | voprov:Entity.rights | | meta.code.class | 0 | 0 | 0 |
| 5 | 4 | entity | e_location | VARCHAR | -1 | -1 | | voprov:Entity.location | | meta.ref.url | 0 | 0 | 0 |
| 6 | 5 | entity | e_generated | VARCHAR | -1 | -1 | | voprov:Entity.generatedAtTime | | time.start | 0 | 0 | 0 |
| 7 | 6 | entity | e_invalidated | VARCHAR | -1 | -1 | | voprov:Entity.invalidatedAtTime | | time.stop | 0 | 0 | 0 |
| 8 | 7 | entity | e_comment | VARCHAR | -1 | -1 | | voprov:Entity.comment | | meta.description | 0 | 0 | 0 |
| 9 | 8 | entity | e_classtype | VARCHAR | -1 | -1 | | voprov:Entity.classtype | | meta.code.class | 0 | 0 | 0 |
| 10 | 9 | entity | e_value | VARCHAR | -1 | -1 | | voprov:Entity.value | | meta.description | 0 | 0 | 0 |
| 11 | 10 | entity | e_description | VARCHAR | -1 | -1 | | voprov:Entity.description_id | | meta.id | 0 | 0 | 0 |

# ProvTAP specification

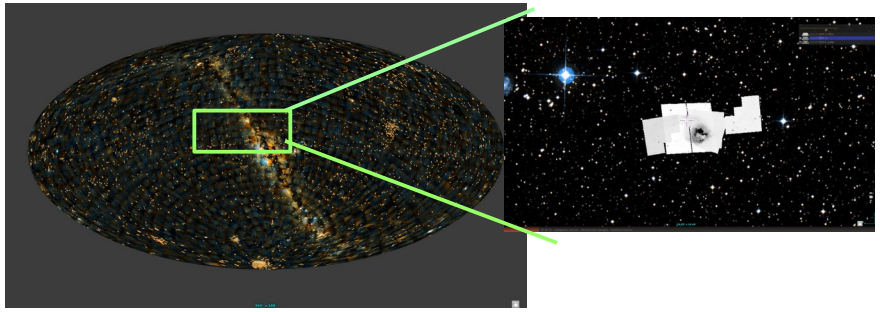
## for IVOA provenance datamodel serialisation and metadata service

- specification will be published as a working draft just after the interop.



# ProvHiPS prototype

- Tracing provenance of HST HiPS and HiPS tiles
  - From HiPS tiles back to raw HST images,
  - through « drizzled » images
  - and « calibrated » images



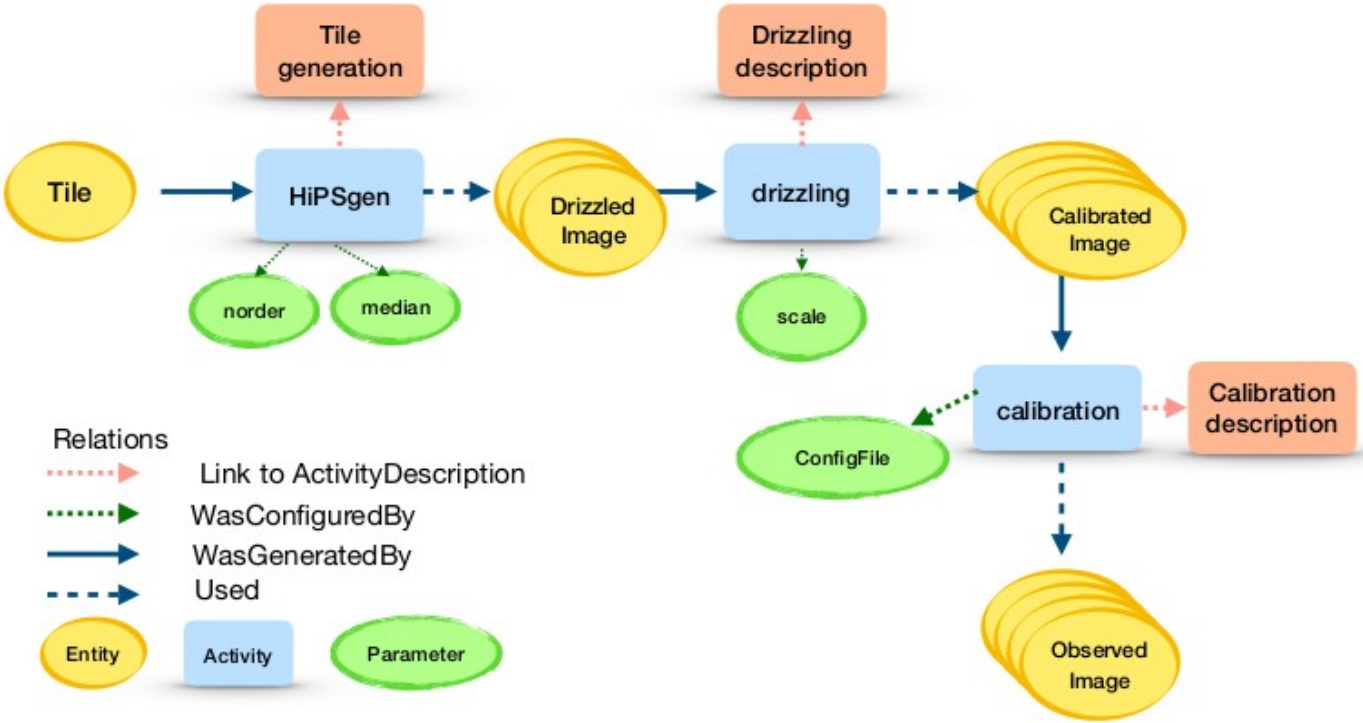
- ProvHiPS is an example for « On Top provenance » for image processing in optical astronomy.
  - See examples next slides
  - *Will not become a real service due to lack of manpower*

# ProvHiPS prototype access : try it to see limitations

Via legacy TapHandle :https://saada.unistra.fr/taphandle/?url=https://saada.unistra.fr/provta

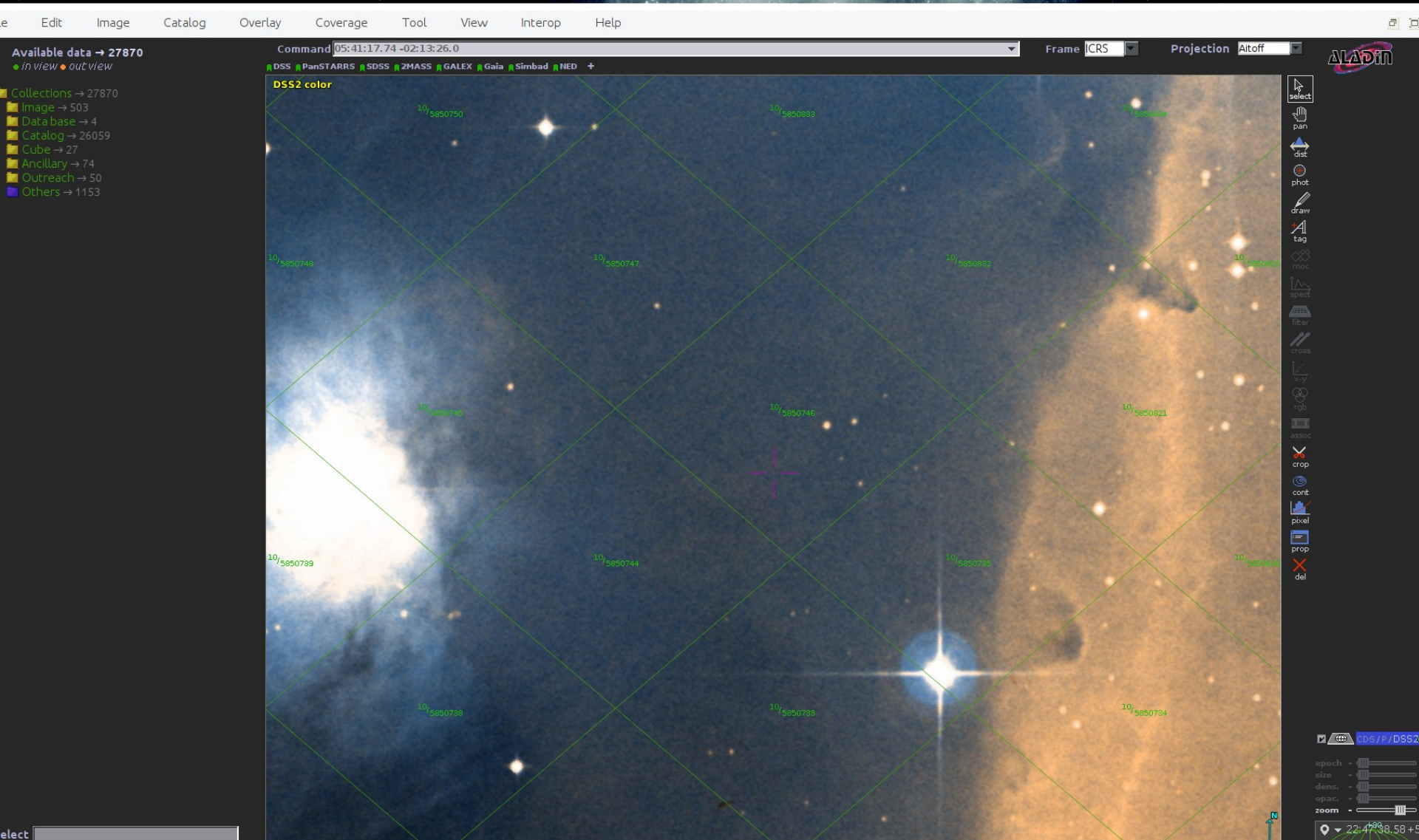
Via  smart TapHandle:
https://taphandle.astro.unistra.fr/tapcomplex/app/Tap_Handle_MK2/taphandev.html?url=//saada.unistra.fr/provtap

# «HiPS» Provenance diagram



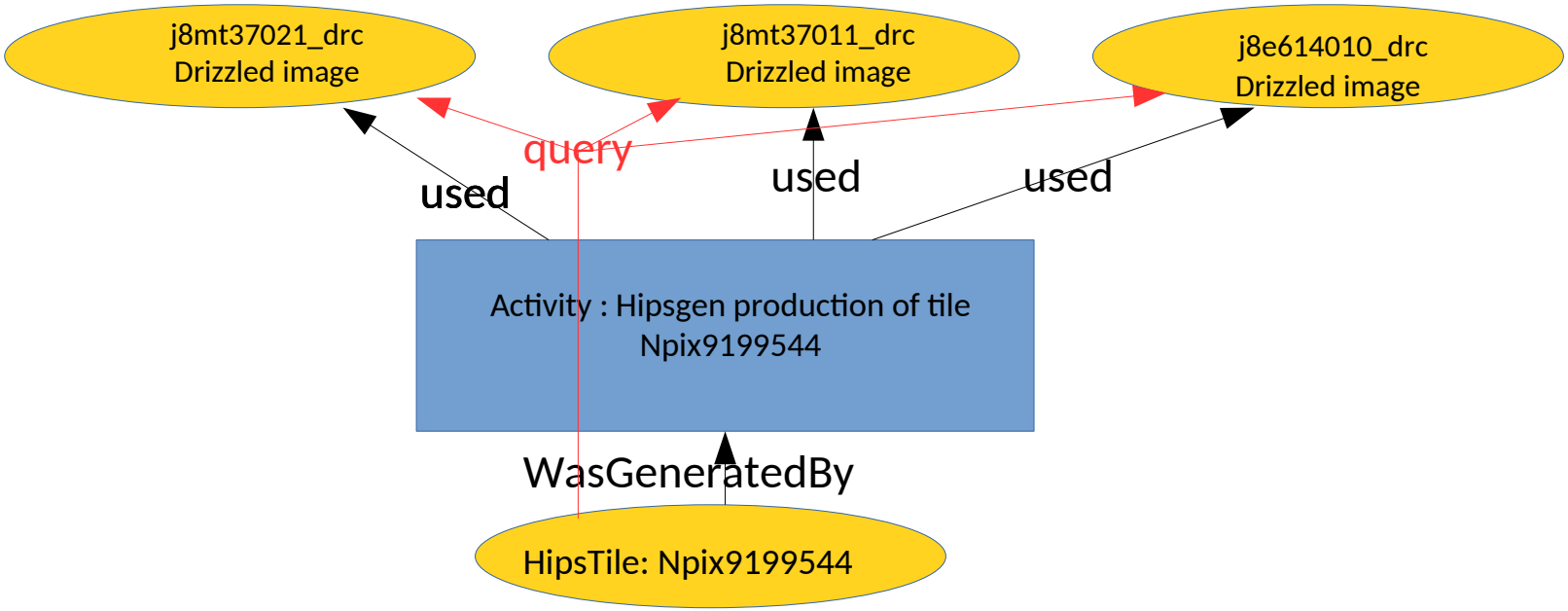

Provenance tracking for Prov-HiPS

# «HiPS» tiles at order 10 history



# ProvHiPS ADQL query examples :

## Finding out drizzled images « progenitors » of a specific HiPS tile.



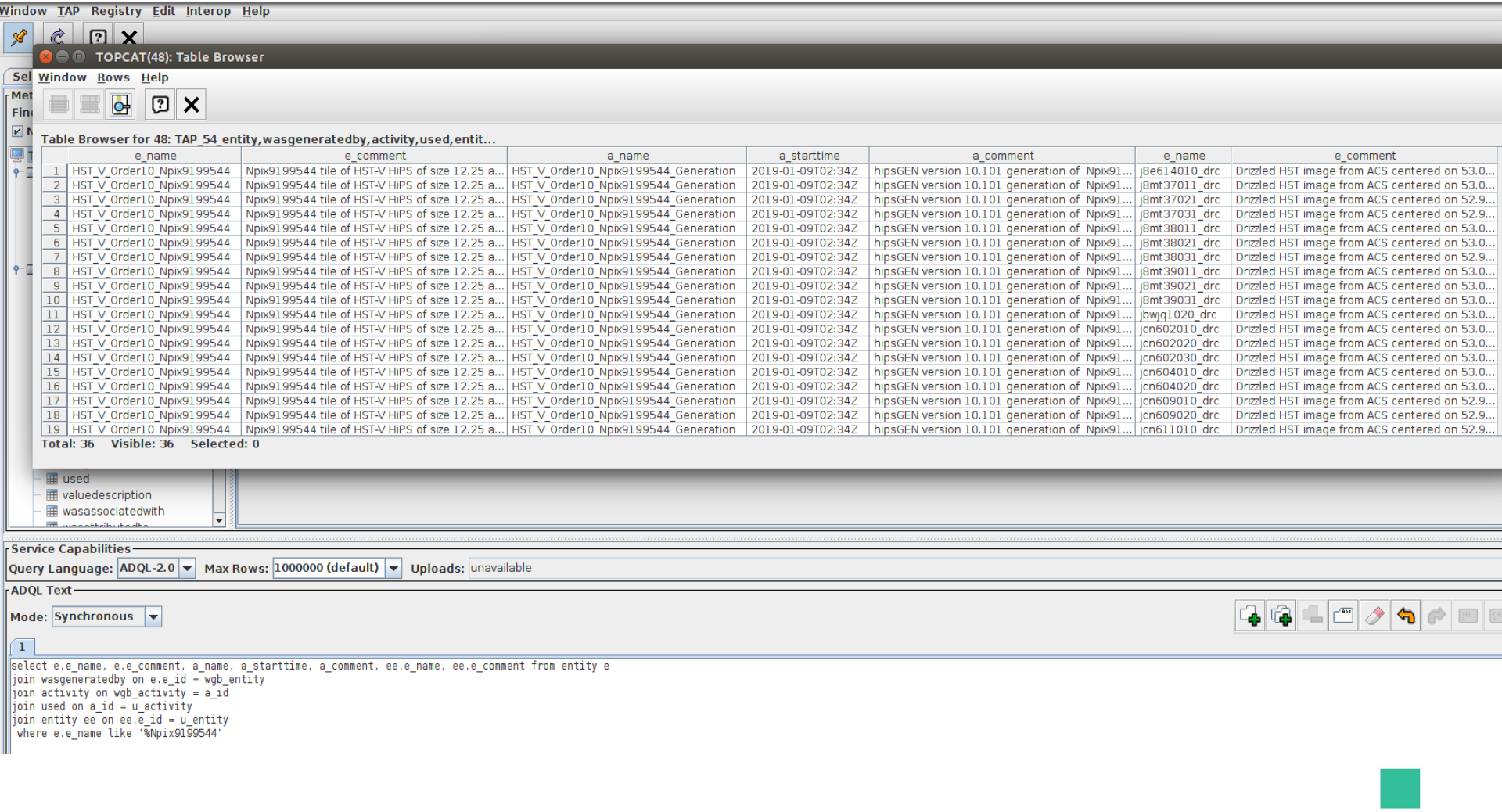
# ProvHiPS ADQL query examples :

## Finding out drizzled images « progenitors » of a specific HiPS tile.

```
select tile.e_name, tile.e_comment, hipsgen.a_name,
 hipsgen.a_starttime, hipsgen.a_comment, drizzle.e_name, drizzle.e_comment from
     entity tile
join wasgeneratedby on tile.e_id = wgb_entity
join activity hipsgen on wgb_activity = hipsgen.a_id
join used on hipsgen.a_id = u_activity
join entity drizzle   on drizzle.e_id = u_entity
where tile.e_name like '%Npix9199544'
```

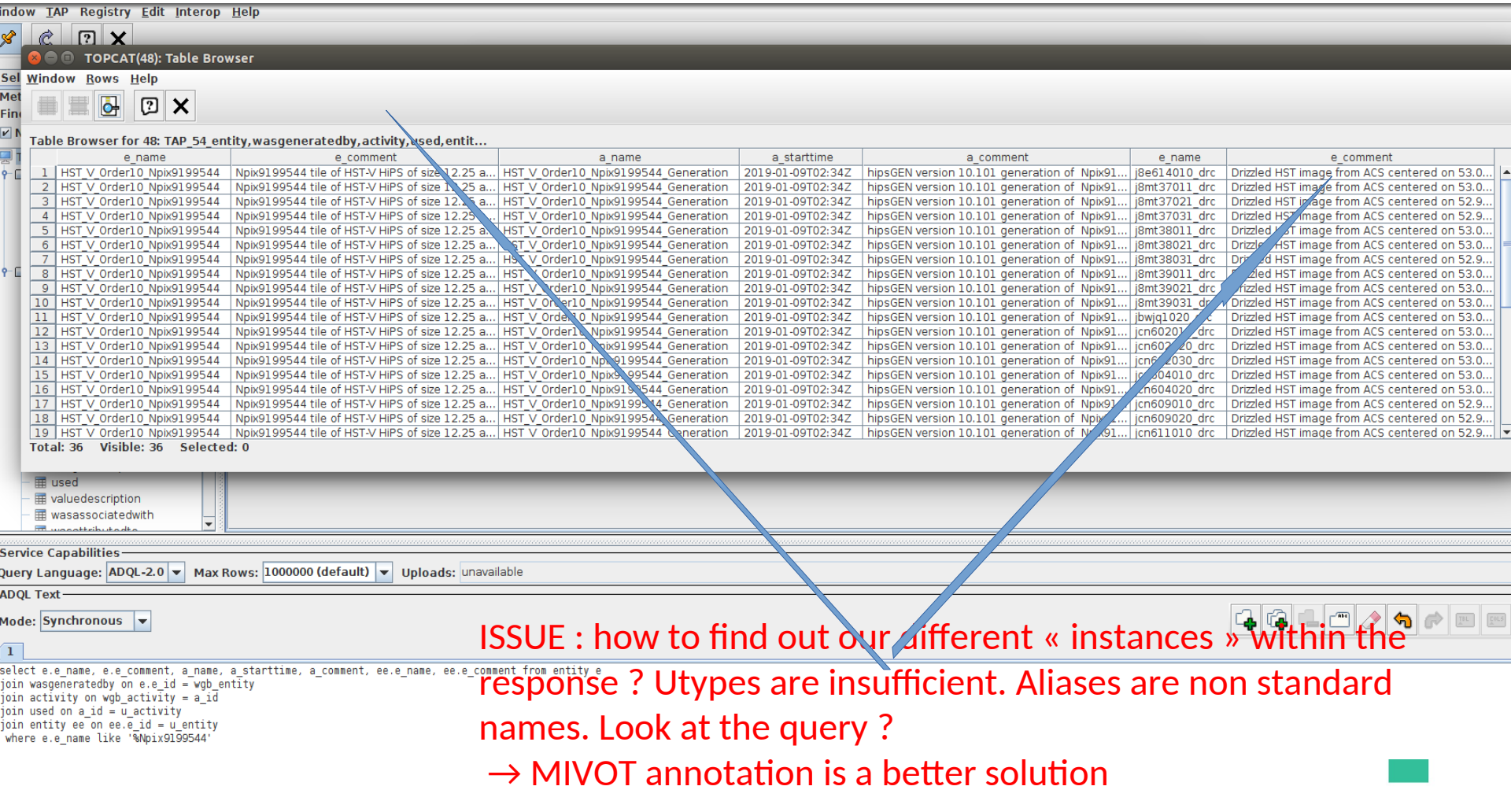
# ProvHiPS ADQL query examples :

## Finding out drizzled images « progenitors » of a specific HiPS tile.



# ProvHiPS ADQL query examples :

## Finding out drizzled images « progenitors » of a specific HiPS tile.




ISSUE : how to find out our different « instances » within the response ? Utypes are insufficient. Aliases are non standard names. Look at the query ?

→ MIVOT annotation is a better solution

ProvHiPS ADQL complex query examples :

# Using MIVOT annotations on the results
# 1 ) snippets

- The snippets are generated by *mivot-snippet-tool*

  belonging to mivot_validator github project :

  http://github.com/ivoa/mivot-validator.git

  from IVOA provenance vo-dml-xml representation :

  **mivot-snippet-model https://www.ivoa.net/xml/VODML/20191125/Provenance-v1.vo-dml.xml**

```
-<INSTANCE dmrole="" dmtype="provenance:Entity">
  -<!--
      A thing which is consumed or produced in a task , namely here in the astronomical domain. It represents some data as single element or file or an instrument.

     "
   -->
   <ATTRIBUTE dmrole="provenance:Entity.id" dmtype="ivoa:string" ref="@@@@@" value=""/>
   <ATTRIBUTE dmrole="provenance:Entity.name" dmtype="ivoa:string" ref="@@@@@" value=""/>
   <ATTRIBUTE dmrole="provenance:Entity.location" dmtype="ivoa:string" ref="@@@@@" value=""/>
   <ATTRIBUTE dmrole="provenance:Entity.generatedAtTime" dmtype="ivoa:datetime" unit="" ref="@@@@@" value=""/>
   <ATTRIBUTE dmrole="provenance:Entity.invalidatedAtTime" dmtype="ivoa:datetime" unit="" ref="@@@@@" value=""/>
   <ATTRIBUTE dmrole="provenance:Entity.comment" dmtype="ivoa:string" ref="@@@@@" value=""/>
   <INSTANCE dmrole="provenance:Entity.wasGeneratedBy" dmtype="voprov:WasGeneratedBy"/>
  -<COLLECTION dmrole="provenance:Entity.wasAttributedTo">
     <INSTANCE dmrole="" dmtype="voprov:WasAttributedTo"/>
   </COLLECTION>
  -<COLLECTION dmrole="provenance:Entity.usedEntity">
     <INSTANCE dmrole="provenance:Entity.usedEntity" dmtype="voprov:Entity"/>
   </COLLECTION>
   <INSTANCE dmrole="provenance:Entity.entityDescription" dmtype="voprov:EntityDescription"/>
 </INSTANCE>
```

*Entity* class snippet

ProvHiPS ADQL complex query examples :

# Using MIVOT annotations on the results

## 1 ) snippets

```
−<INSTANCE dmrole="" dmtype="provenance:Activity">
  −<!--
      A task executed at some time. It consumes products via relations of type Used and produces results via relations of type WasGeneratedBy.
     The method applied for this task is described in the ActivityDescription class and its subtree."
    -->
    <ATTRIBUTE dmrole="provenance:Activity.id" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Activity.name" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Activity.startTime" dmtype="ivoa:datetime" unit="" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Activity.endTime" dmtype="ivoa:datetime" unit="" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Activity.comment" dmtype="ivoa:string" ref="@@@@@" value=""/>
  −<COLLECTION dmrole="provenance:Activity.wasAssociatedWith">
      <INSTANCE dmrole="" dmtype="voprov:WasAssociatedWith"/>
    </COLLECTION>
  −<COLLECTION dmrole="provenance:Activity.used">
      <INSTANCE dmrole="" dmtype="voprov:Used"/>
    </COLLECTION>
  −<COLLECTION dmrole="provenance:Activity.wasConfiguredBy">
      <INSTANCE dmrole="" dmtype="voprov:ActivityConfiguration.WasConfiguredBy"/>
    </COLLECTION>
  −<COLLECTION dmrole="provenance:Activity.informant">
      <INSTANCE dmrole="provenance:Activity.informant" dmtype="voprov:Activity"/>
    </COLLECTION>
    <INSTANCE dmrole="provenance:Activity.activityDescription" dmtype="voprov:ActivityDescription"/>
  </INSTANCE>
```

*Activity* class snippet

```
−<INSTANCE dmrole="" dmtype="provenance:Agent">
  −<!--
      A person or an organisation that was involved in the follow-up of an Activity, or can be credited for the production of an Entity."
    -->
    <ATTRIBUTE dmrole="provenance:Agent.id" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.name" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <INSTANCE dmrole="provenance:Agent.type" dmtype="voprov:AgentType"/>
    <ATTRIBUTE dmrole="provenance:Agent.comment" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.email" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.affiliation" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.phone" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.address" dmtype="ivoa:string" ref="@@@@@" value=""/>
    <ATTRIBUTE dmrole="provenance:Agent.url" dmtype="ivoa:anyURI" unit="" ref="@@@@@" value=""/>
  </INSTANCE>
```

*Agent* class snippet

ProvHiPS ADQL complex query examples : using MIVOT annotations on the results

# 2 ) finding out « drizzle » progenitors response annotation

```
−<RESOURCE type="meta">
 −<VODML>
    <REPORT status="OK"/>
    <MODEL name="voprov" url="https://www.ivoa.net/xml/VODML/20191125/Provenance-v1.vo-dml.xml"/>
    <MODEL name="ivoa" url="https://ivoa.net/xml/VODML/IVOA-v1.vo-dml.xml"/>
  −<TEMPLATES tableref="res1">
   −<INSTANCE dmtype="voprov:Entity">
      <ATTRIBUTE dmrole="voprov:Entity.id" dmtype="ivoa:string" ref="@@@@@"/>
      <ATTRIBUTE dmrole="voprov:Entity.name" dmtype="ivoa:string" ref="tile_name"/>
      <ATTRIBUTE dmrole="voprov:Entity.location" dmtype="ivoa:string" ref="@@@@@"/>
      <ATTRIBUTE dmrole="voprov:Entity.generatedAtTime" dmtype="ivoa:string" ref="@@@@@"/>
      <ATTRIBUTE dmrole="voprov:Entity.invalidatedAtTime" dmtype="ivoa:string" ref="@@@@@"/>
      <ATTRIBUTE dmrole="voprov:Entity.comment" dmtype="ivoa:string" ref="tile_com"/>
      <ATTRIBUTE dmrole="voprov:Entity.classType" dmtype="ivoa:string" ref="@@@@@"/>
      <ATTRIBUTE dmrole="voprov:Entity.value" dmtype="ivoa:string" ref="@@@@@"/>
     −<INSTANCE dmrole="voprov:Entity.wasGeneratedBy" dmtype="voprov:WasGeneratedBy">
       −<INSTANCE dmrole="voprov:WasGeneratedBy.activity" dmtype="voprov:Activity">
          <ATTRIBUTE dmrole="voprov:Actvity.name" dmtype="ivoa:string" ref="gen_name"/>
          <ATTRIBUTE dmrole="voprov:Actvity.startTime" dmtype="ivoa:string" ref="gen_start"/>
          <ATTRIBUTE dmrole="voprov:Actvity.name" dmtype="ivoa:string" ref="gen_com"/>
         −<INSTANCE dmrole="voprov:Activity.Used" dmtype="voprov:Used">
           −<COLLECTION dmrole="Used.entity">
             −<INSTANCE dmtype="voprov:entity">
                <ATTRIBUTE dmrole="voprov:Entity.name" dmtype="ivoa:string" ref="driz_name"/>
                <ATTRIBUTE dmrole="voprov:Entity.comment" dmtype="ivoa:string" ref="driz_com"/>
              </INSTANCE>
            </COLLECTION>
          </INSTANCE>
        </INSTANCE>
      </INSTANCE>
    </INSTANCE>
  </TEMPLATES>
 </VODML>
</RESOURCE>
```

Model structure is mapped
Onto the response TABLE

```
−<RESOURCE type="results">
 −<TABLE ID="res1" name="result_S1716147506717" nrows="3">
    <PARAM arraysize="2" datatype="char" name="QUERY_STATUS" value="OK"/>
    <PARAM arraysize="3" datatype="char" name="PROVIDER" value="CDS"/>
    <PARAM arraysize="482" datatype="char" name="QUERY" value="select tile.e_name as tile_name, tile.e_comment as tile_comment, hipsgen.a_
    as hipsgen_start, hipsgen.a_comment as hipsgen_comment, drizzle.e_name as drizzle_name, drizzle.e_comment as drizzle_comment from entity tile
    join activity hipsgen on wgb_activity = hipsgen.a_id join used on hipsgen.a_id = u_activity join entity drizzle on drizzle.e_id = u_entity where tile.e_n
    <FIELD ID="tile_name" arraysize="*" datatype="char" name="tile_name" ucd="meta.title" utype="voprov:Entity.name"/>
    <FIELD ID="tile_com" arraysize="*" datatype="char" name="tile_comment" ucd="meta.description" utype="voprov:Entity.comment"/>
    <FIELD ID="gen_name" arraysize="*" datatype="char" name="hipsgen_name" ucd="meta.title" utype="voprov:Activity.name"/>
    <FIELD ID="gen_start" arraysize="*" datatype="char" name="hipsgen_start" ucd="time.start" utype="voprov:Activity.startTime"/>
    <FIELD ID="gen_com" arraysize="*" datatype="char" name="hipsgen_comment" ucd="meta.description" utype="voprov:Activity.comment"/>
    <FIELD ID="driz_name" arraysize="*" datatype="char" name="drizzle_name" ucd="meta.title" utype="voprov:Entity.name"/>
    <FIELD ID="driz_com" arraysize="*" datatype="char" name="drizzle_comment" ucd="meta.description" utype="voprov:Entity.comment"/>
   −<DATA>
    −<TABLEDATA>
     −<TR>
        <TD>HST_B_Norder10_Npix9199544</TD>
       −<TD>
          Npix9199544 tile of HST-B HiPS of size 3.5*3.5 and including position 53.069038233321 and -27.822573192919
        </TD>
        <TD>HST_B_Norder10_Npix9199544_Generation</TD>
        <TD>2018-11-25T22:01Z</TD>
       −<TD>
          hipsGEN version 10.101 generation of Npix9199544tile of HST-B HiPS
        </TD>
        <TD>jcn614030_drc.fits</TD>
       −<TD>
          Drizzled HST image from ACS with filterCLEAR1L and F435W centered on 53.00678473642296 , -27.78192665265288 , with a position ang
          arcsec x 4238*0.05 arsec field of view
        </TD>
      </TR>
```

ProvHiPS ADQL complex query examples : using MIVOT annotations on the results

# 2 ) finding out « drizzle » progenitors
# MiVOT « viewer » (pyvo) reading the table

## Producing a dictionary

```
[103]: votable = "../pyvo-ci-sample/ProvTap-result-1.xml"
       # init the MIVOT viewer from a VOTable file, parsed VOTable or DAL response
       mivot_viewer = MivotViewer(votable)
       print(mivot_viewer.dm_instance)
```

```
  },
  "Entity_wasGeneratedBy": {
    "dmrole": "wasGeneratedBy",
    "dmtype": "WasGeneratedBy",
    "WasGeneratedBy_activity": {
      "dmrole": "activity",
      "dmtype": "Activity",
      "name": {
        "value": "hipsGEN version 10.101 generation of  Npix9199544tile of HST-B HiPS"
      },
      "startTime": {
        "value": "2018-11-25T22:01Z"
      },
      "Used": {
        "dmrole": "Used",
        "dmtype": "Used",
        "Used.entity": [
          {
            "dmtype": "entity",
            "name": {
              "value": "jcn614030_drc.fits"
            },
            "comment": {
              "value": "Drizzled HST image from ACS with filterCLEAR1L and F435W centered on
53.00678473642296 , -27.78192665265288 , with a position angle of 85.04776551398835 and 4214
*0.05 arcsec x 4238*0.05 arsec field of view"
            }
          }
```

ProvHiPS ADQL complex query examples : using MIVOT annotations on the results

# 2 ) finding out « drizzle » progenitors
# MiVOT « viewer » (pyvo) reading the table

Class instances generattion

```
mivot_viewer.rewind()
while mivot_viewer.next():
    name = mivot_instance.name.value
    bby = mivot_instance.Entity_wasGeneratedBy.WasGeneratedBy_activity.name.value
    # Object generation flaw with collection management: to be fixed
    using = mivot_instance.Entity_wasGeneratedBy.WasGeneratedBy_activity.Used.__dict__["Used.entity"][0].name.value
    print(f"'{name}' generated by '{bby}' using '{using}'")
```

```
'HST_B_Norder10_Npix9199544' generated by 'hipsGEN version 10.101 generation of  Npix9199544tile of HST-B HiPS' using 'jcn614030_drc.fits'
'HST_B_Norder10_Npix9199544' generated by 'hipsGEN version 10.101 generation of  Npix9199544tile of HST-B HiPS' using 'jcn609020_drc.fits'
'HST_B_Norder10_Npix9199544' generated by 'hipsGEN version 10.101 generation of  Npix9199544tile of HST-B HiPS' using 'j8mt37011_drc.fits'
```

# Another use case with a different solution :
## Finding agent and activity to whom entity is attributed and generating a given entity

- select ag_name,ag_type,ag_comment, e_name, e_generated,e_location, e_comment, a_name, a_comment

  from agent

  join wasattributedto on ag_id = wat_agent

  join entity on wat_entity = e_id

  join wasgeneratedby on e_id = wgb_entity

  join activity on a_id = wgb_activity

  where e_name = 'j8f627010_drz.fits'

# Issues

- Table is denormalized : a lot of redundant information
- Loop issue : several occurances of the same triplet (name,utype,ucd) in the same table for different « objects »
- Let’s try minimum or last step provenance by creating a standardized view

# Solutions
## -1 Single step = single table ( = join)

- The join is a permanent view described in the TAP schema
- Columns :

  entity_name, entity_location, entity_comment, …

  generating_activity_name, generating_activity_starttime, ….

  agent_role, agent_name, ….

  used_entity_list
- → Redundancy may be avoided if we group all used entities ids in a single cell
- → possible Recursivity

# Solutions

## minimum provenance(= join)

## 1 single line per generated entity

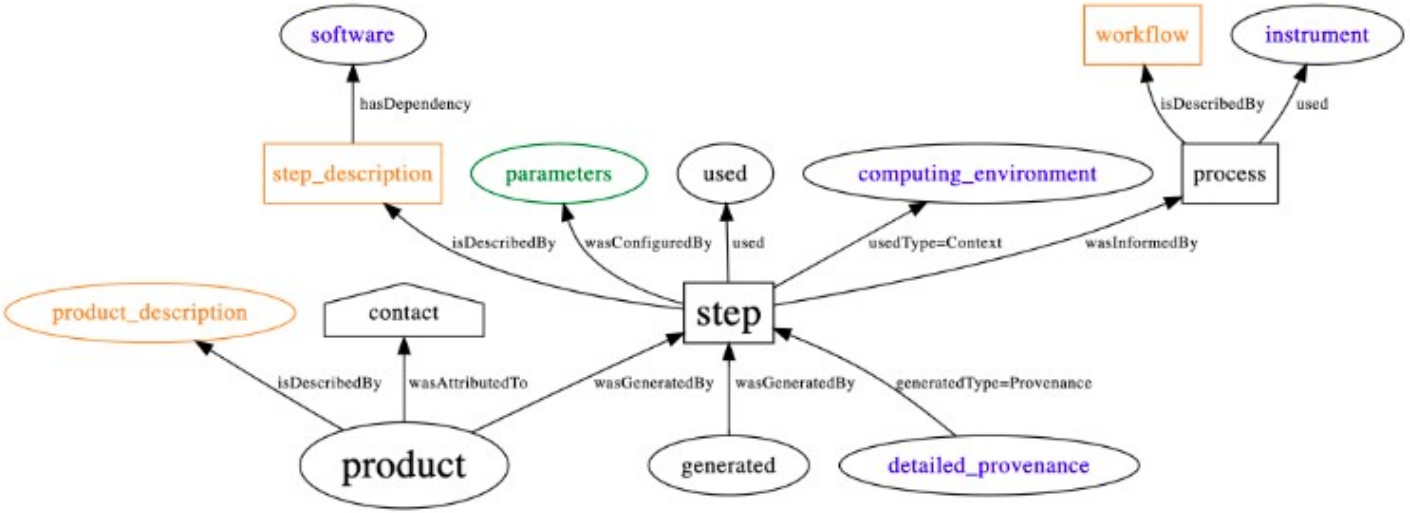
- View (in postgres)

```
create view minimum_provenance as select

e.e_id AS entity_id,  e.e_name AS entity_name, e.e_location AS entity_location, e.e_generated AS entity_generated,
    e.e_invalidated AS entity_invalidated, e.e_comment AS entity_comment,

 activity.a_name AS generating_activity_name, activity.a_starttime AS generating_activity_starttime,
activity.a_endtime AS generating_activity_endtime, activity.a_comment AS generating_activity_comment,

 wasattributedto.wat_role AS agent_role, agent.ag_name AS agent_name, agent.ag_type AS agent_type,
    agent.ag_affiliation AS agent_affiliation, agent.ag_email AS agent_email, agent.ag_address AS agent_address,
    agent.ag_phone AS agent_phone, agent.ag_comment AS agent_comment,

string_agg(used.u_entity::text, ','::text) AS used_entities_list
FROM entity e

 JOIN wasgeneratedby ON e.e_id::text = wasgeneratedby.wgb_entity::text
 JOIN activity ON wasgeneratedby.wgb_activity::text = activity.a_id::text
join used on u_activity = a_id
join entity as ee on ee.e_id = u_entity
join wasattributedto on wat_entity = e.e_id
join agent on ag_id = wat_agent ; yypo
```

# Solutions

## one step provenance
## (see DM2 tomorrow -Mathieu Servillat)

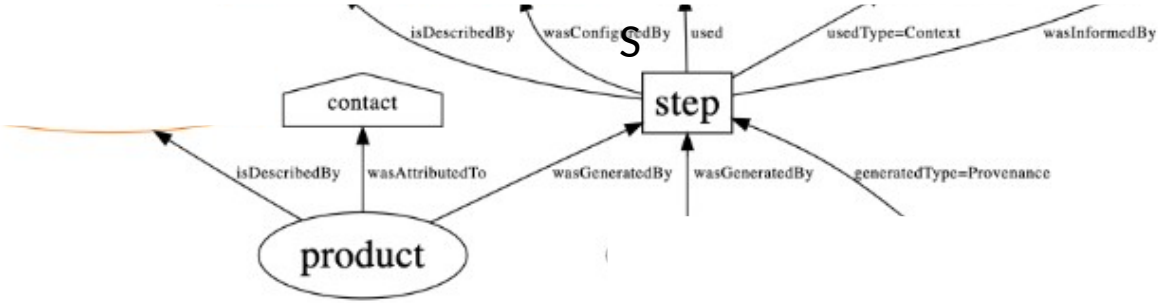
- Changing the names (product, step, process)
- Add « descriptions »



# Solutions

## one step provenance
## (see DM2 tomorrow -Mathieu Servillat)

- To keep it simple
  - Put only references (ids) to additional material
  - Recursivity



# Solutions

## last step provenance in ProvTAP

- View (in postgres)

```
create view one_step_provenance as select
e.e_id AS product_id,  e.e_name AS product_name, e.e_location AS product_location, e.e_generated AS
product_generated, e.e_description as product_description,
   e.e_invalidated AS product_invalidated, e.e_comment AS product_comment,
 activity.a_name AS step_name, activity.a_starttime AS step_starttime,      activity.a_endtime AS step_endtime,
activity.a_comment AS step_comment, a.a_description as step_description,
 wasattributedto.wat_role AS contact_role, agent.ag_name AS contact_name, agent.ag_type AS contact_type,
   agent.ag_affiliation AS contact_affiliation, agent.ag_email AS contact_email, agent.ag_address AS contact_address,
   agent.ag_phone AS contact_phone, agent.ag_comment AS contact_comment,
string_agg(used.u_entity::text, ','::text) AS used_entities_list
FROM entity e
 JOIN wasgeneratedby ON e.e_id::text = wasgeneratedby.wgb_entity::text
 JOIN activity ON wasgeneratedby.wgb_activity::text = activity.a_id::text
join used on u_activity = a_id
join entity as ee on ee.e_id = u_entity
join wasattributedto on wat_entity = e.e_id
join agent on ag_id = wat_agent ;
```

# Future work

- Last step provenance and MIVOT annotation in an appendix of the specification
- Release ProvTAP WD
- Implement this in the CTA data archive project.