Provenance information management

ESCAPE

European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

IVOA

Mathieu Servillat, Catherine Boisson, François Bonnarel, Mireille Louys, Michèle Sanguillon

FAIR principles for data sharing

https://www.go-fair.org/fair-principles

FINDABLE

Unique identifiers and metadata are used to allow data to be located quickly and efficiently

ACCESSIBLE

Data is open, free and universally available for research discovery efforts

INTER-OPERABLE

A common programming language is used to allow use in a broad range of applications

REUSABLE

All data is clearly described and outlines associated data-use standards

FAIR principles for data sharing

https://www.go-fair.org/fair-principles

Findable

F1. (Meta)data are assigned a globally unique and persistent **identifier**
F2. Data are described with rich metadata
F3. Metadata clearly and explicitly include the **identifier** of the data they describe
F4. (Meta)data are **registered** or **indexed** in a searchable resource

Accessible

A1. (Meta)data are retrievable by their **identifier** using a **standardised** communications **protocol**
  A1.1. The **protocol** is open, free, and universally implementable
  A1.2. The **protocol** allows for an authentication and authorisation procedure, where necessary
A2. Metadata are accessible, even when the data are no longer available

Interoperable

I1. (Meta)data use a formal, accessible, shared, and broadly applicable **language** for knowledge representation.
I2. (Meta)data use **vocabularies** that follow FAIR principles
I3. (Meta)data include **qualified** references to other (meta)data

Reusable **(+ Reproducible?)**

R1. (Meta)data are richly described with a plurality of **accurate** and **relevant** attributes
  R1.1. (Meta)data are released with a **clear** and accessible data usage **license**
  R1.2. (Meta)data are associated with detailed **provenance**
  R1.3. (Meta)data meet domain-relevant community **standards**

International Virtual Observatory Alliance

IVOA Documents

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

IVOA Provenance Data Model
Version 1.0

IVOA Recommendation 11 April 2020

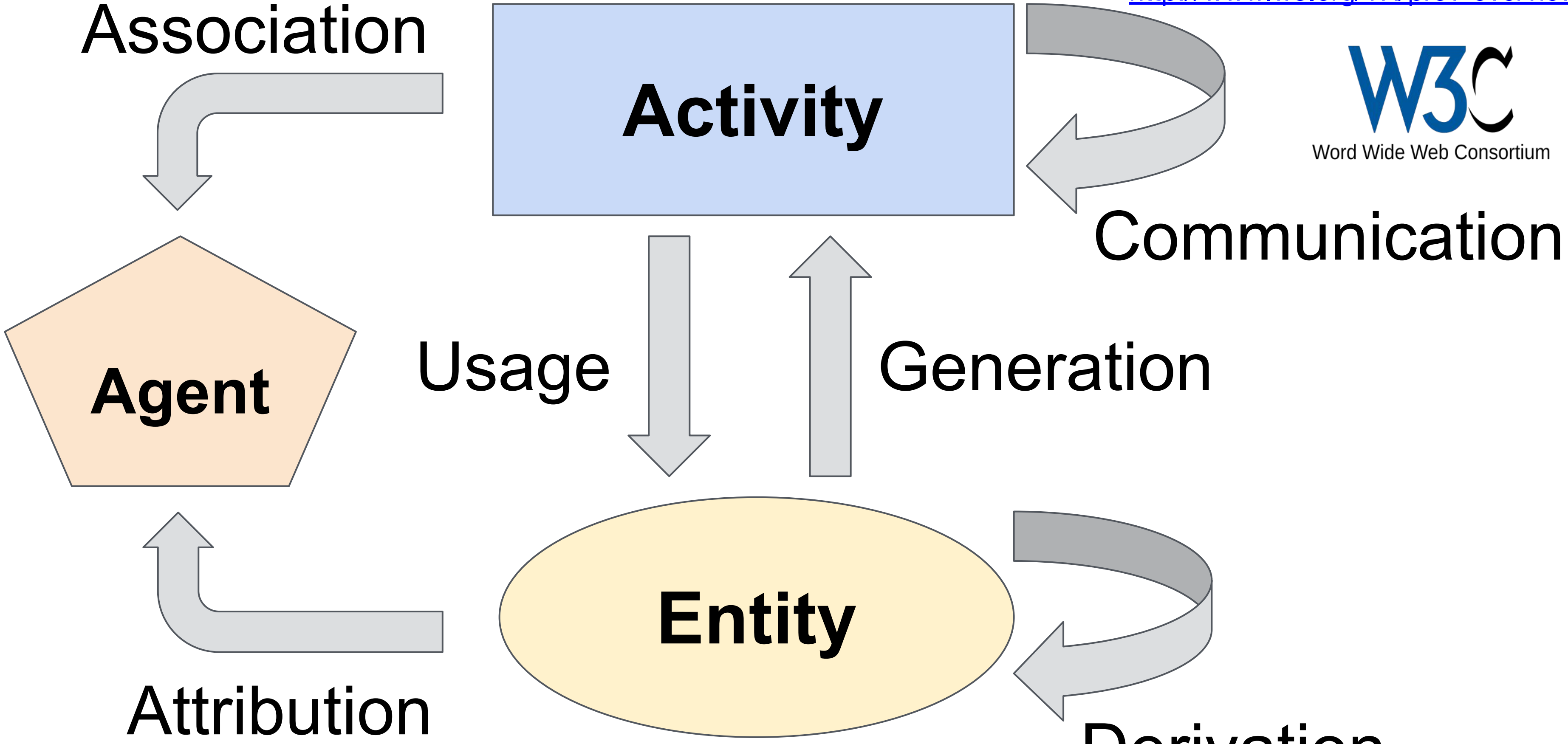
Interest/Working Group:
    http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel
Author(s):
    Mathieu Servillat, Kristin Riebe, Catherine Boisson, François Bonnarel, Anastasia Galkin, Mireille Louys, Markus Nullmeier, Nicolas Renault-Tinacci, Michèle Sanguillon, Ole Streicher
Editor(s):
    Mathieu Servillat

Why recording structured provenance?

- **Make data FAIR** (Findable, Accessible, Interoperable, Reusable)
  - https://www.go-fair.org/fair-principles/
  - "**rich**" metadata, following standard data model, protocols and formats
  - "**detailed provenance**"
- **Quality** / **Reliability** / **Trustworthiness** of the products
- **Reproducibility requirement** in projects
  - Be able to rerun each activity (maybe testing and improving each step)
  - Not necessary to keep every intermediate file that is easily reproducible (possible gain on disk space and costs)
- **Debugging**
  - Not necessary to restart from scratch: locate in the provenance tree the faulty parts or the products to be discarded

→ We often realize too late that there are missing elements or links in the provenance. The capture of the provenance should be as detailed as possible and as naive as possible (simply record what happens).

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



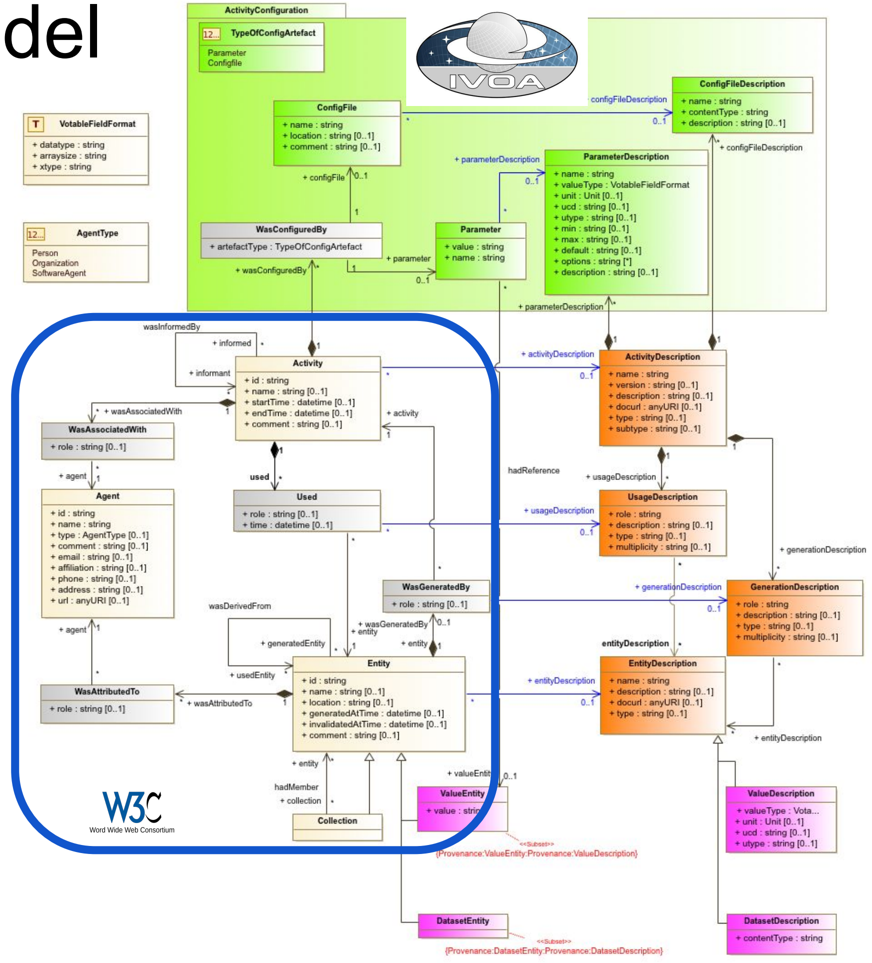
Word Wide Web Consortium

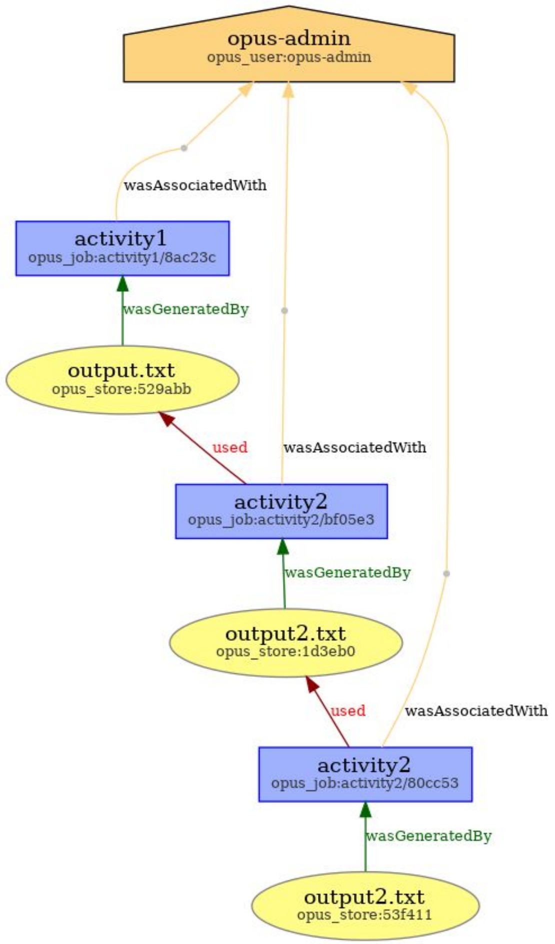
IVOA Provenance Data Model

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

Recommendation in April 2020

- Adds "**Description**" classes
- Adds "**Configuration**" classes
- Plugged in with
  - **VO** data models and concepts (UCD, VOUnit, VOTable…)
  - **VO** access protocols (ProvTAP, ProvSAP)
- Serializations
  - W3C PROV (XML, JSON, SVG…)
  - **VO** specific (VOTable)





Provenance graph

Provenance is :

- a **chain** of activities and entities (used and generated)
- that occured in the **past**

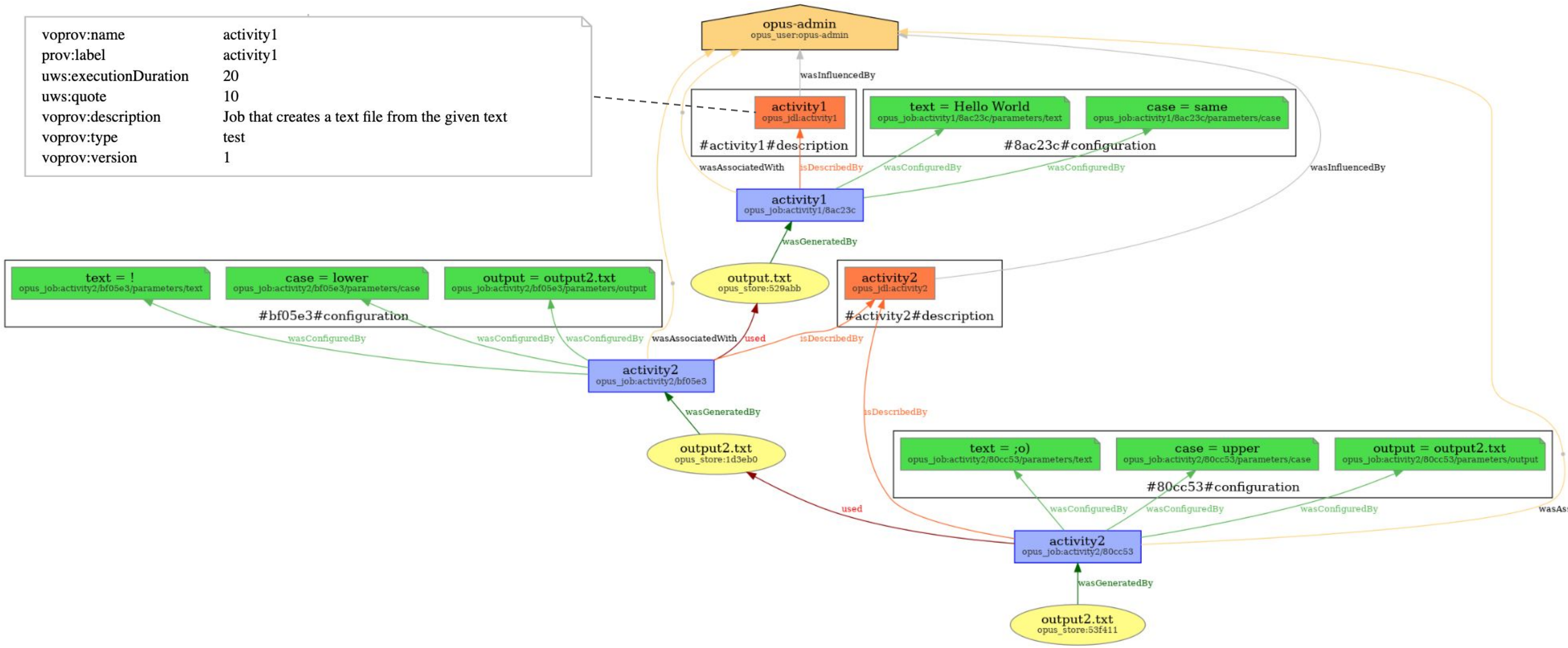
Using the **core data model**, some goals are achieved:

- **Traceability and Unique identifiers**
- **Contact and Acknowledgement**

By using the **full IVOA data model**, more questions are answered:

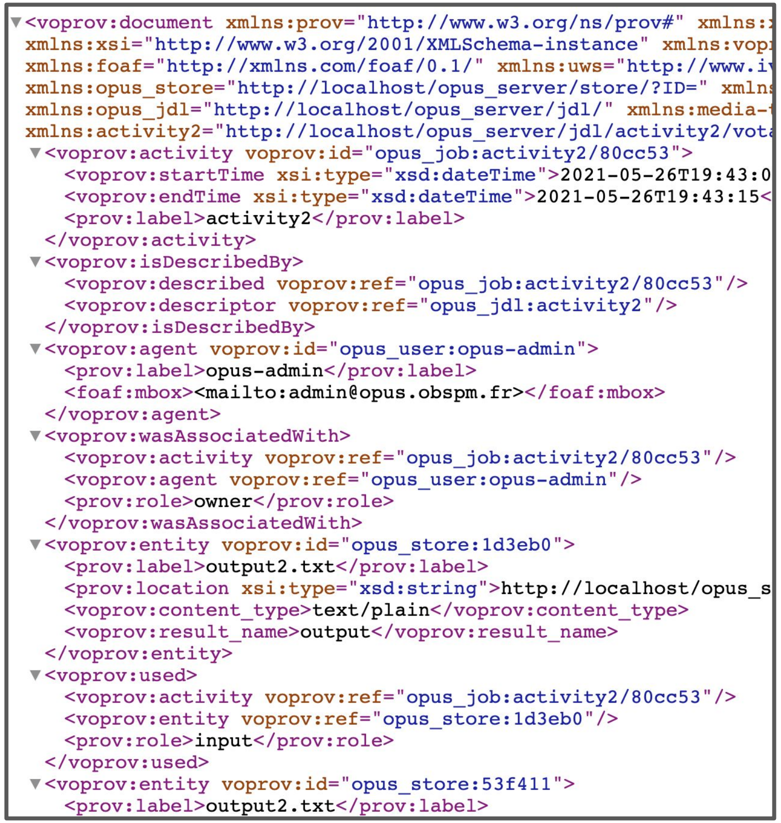
- What **happened** during each **activity**?
- How was the **activity tuned** to be executed properly?
- What **kind of content** is in the **entities**?

Full IVOA Provenance graph

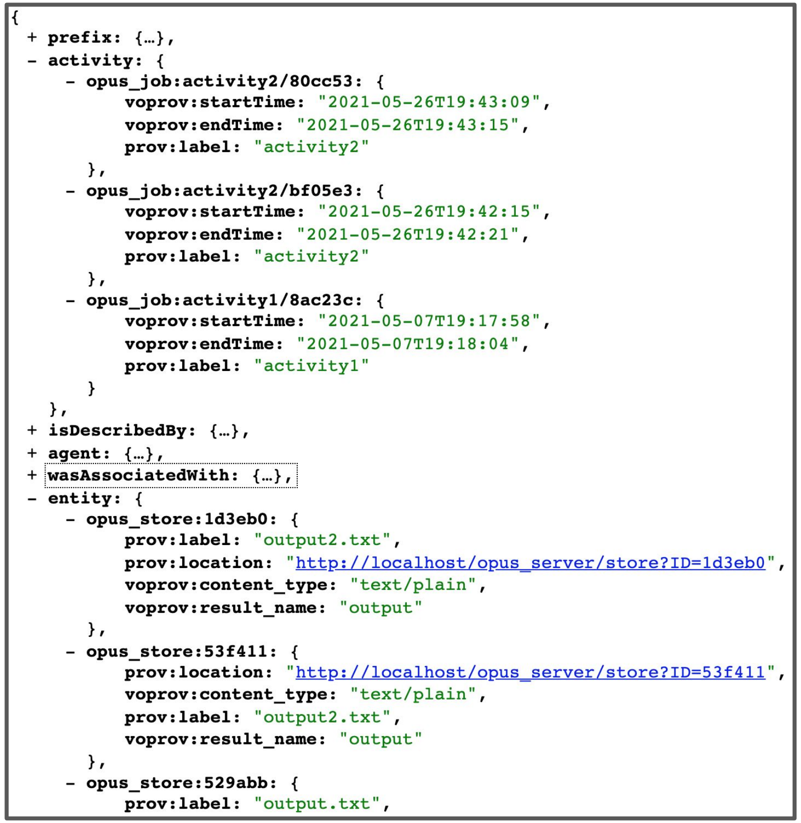


voprov:name	activity1
prov:label	activity1
uws:executionDuration	20
uws:quote	10
voprov:description	Job that creates a text file from the given text
voprov:type	test
voprov:version	1

W3C serializations: XML and JSON



```
▼<voprov:document xmlns:prov="http://www.w3.org/ns/prov#" xmlns:
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:vop
 xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:uws="http://www.i
 xmlns:opus_store="http://localhost/opus_server/store/?ID=" xmlns
 xmlns:opus_jdl="http://localhost/opus_server/jdl/" xmlns:media-
 xmlns:activity2="http://localhost/opus_server/jdl/activity2/vota
  ▼<voprov:activity voprov:id="opus_job:activity2/80cc53">
     <voprov:startTime xsi:type="xsd:dateTime">2021-05-26T19:43:0
     <voprov:endTime xsi:type="xsd:dateTime">2021-05-26T19:43:15<
     <prov:label>activity2</prov:label>
   </voprov:activity>
  ▼<voprov:isDescribedBy>
     <voprov:described voprov:ref="opus_job:activity2/80cc53"/>
     <voprov:descriptor voprov:ref="opus_jdl:activity2"/>
   </voprov:isDescribedBy>
  ▼<voprov:agent voprov:id="opus_user:opus-admin">
     <prov:label>opus-admin</prov:label>
     <foaf:mbox><mailto:admin@opus.obspm.fr></foaf:mbox>
   </voprov:agent>
  ▼<voprov:wasAssociatedWith>
     <voprov:activity voprov:ref="opus_job:activity2/80cc53"/>
     <voprov:agent voprov:ref="opus_user:opus-admin"/>
     <prov:role>owner</prov:role>
   </voprov:wasAssociatedWith>
  ▼<voprov:entity voprov:id="opus_store:1d3eb0">
     <prov:label>output2.txt</prov:label>
     <prov:location xsi:type="xsd:string">http://localhost/opus_s
     <voprov:content_type>text/plain</voprov:content_type>
     <voprov:result_name>output</voprov:result_name>
   </voprov:entity>
  ▼<voprov:used>
     <voprov:activity voprov:ref="opus_job:activity2/80cc53"/>
     <voprov:entity voprov:ref="opus_store:1d3eb0"/>
     <prov:role>input</prov:role>
   </voprov:used>
  ▼<voprov:entity voprov:id="opus_store:53f411">
     <prov:label>output2.txt</prov:label>
```



```
{
  + prefix: {…},
  - activity: {
      - opus_job:activity2/80cc53: {
            voprov:startTime: "2021-05-26T19:43:09",
            voprov:endTime: "2021-05-26T19:43:15",
            prov:label: "activity2"
        },
      - opus_job:activity2/bf05e3: {
            voprov:startTime: "2021-05-26T19:42:15",
            voprov:endTime: "2021-05-26T19:42:21",
            prov:label: "activity2"
        },
      - opus_job:activity1/8ac23c: {
            voprov:startTime: "2021-05-07T19:17:58",
            voprov:endTime: "2021-05-07T19:18:04",
            prov:label: "activity1"
        }
    },
  + isDescribedBy: {…},
  + agent: {…},
  + wasAssociatedWith: {…},
  - entity: {
      - opus_store:1d3eb0: {
            prov:label: "output2.txt",
            prov:location: "http://localhost/opus_server/store?ID=1d3eb0",
            voprov:content_type: "text/plain",
            voprov:result_name: "output"
        },
      - opus_store:53f411: {
            prov:location: "http://localhost/opus_server/store?ID=53f411",
            voprov:content_type: "text/plain",
            prov:label: "output2.txt",
            voprov:result_name: "output"
        },
      - opus_store:529abb: {
            prov:label: "output.txt",
```

Serializations https://etherpad.in2p3.fr/p/provyaml

- **W3C PROV**
  - XML & JSON, not easy to read by humans : lists of classes (entity, activity, agent, used, ...)
  - But visualizations possible : SVG, PNG, PDF
- **YAML** proposed solution
  - Machine readable **and** human readable
  - Structure with only 3 main groups :
    - activities : contains the links to reconstruct the chain of activities-entities
    - entities : contains information on how to find the entities and what they are
    - agents : contains contact information only (no roles)
  - **Relations** are attached to the above objects
  - Can be extended to descriptions
- Use case with Vizier catalogues
  - Model Mapping in VOTable with similar structure

```
agents:
    mservillat:
        name: MS
        email: m..@obspm.fr
entities:
    1234:
        name: input.txt
    5678:
        name: output.txt
        generatedAtTime: 2021-...
activities:
    9876:
        name: transform
        startTime: 2021-...
        endTime: 2021-...
        parameters:
            <name>: <value>
        used:
          - entity_id: 1234
            role: input
        generated:
          - entity_id: 5678
            role: output
        associated:
          - agent_id: mservillat
            role: operator
```

Some terminology

See ADASS XXX proceedings : https://arxiv.org/abs/2101.08691

- **full provenance**: graph/tree/chain of activities and entities up to the raw data. This information is not embedded in the entities themselves (stored on an external server? as separate files?)

- **last-step provenance**: attached to an entity, list of keywords that gives some context and info on **last activity** (general process/workflow, software versions, contacts...).
  Note: it would be interesting to include used entities, so that a full provenance may be reconstructed from each last-step minimum provenance.

- **end-user/specific “provenance”**: attached to an entity, list of keywords or data that provides **key information to use/analyse** the entity (e.g. for CTA: event class, event type, telescope configuration, sky conditions, reco method...)
  Note: may be extracted from full provenance (some parameters or parts of entities generated at a given step), but it is considered as data here. Reversely, this specific "provenance" information may be a source of information to be mapped in the standard in order to fill the full provenance graph with more details.

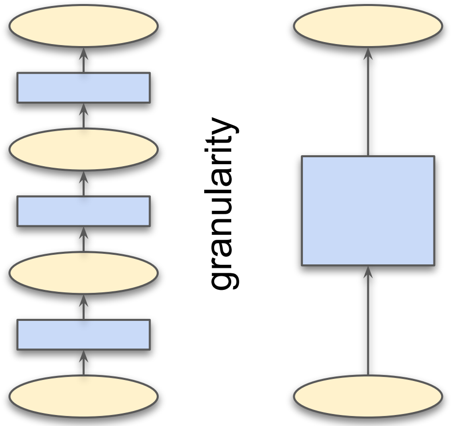
Applying the model

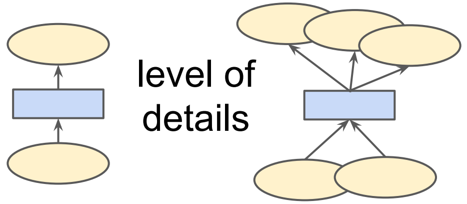
Different contexts in use cases

- Two flavours:
  - **on-top** (data products/collection already exist)
  - **inside** (save provenance information during the processing)
- **Identifiers:** unique and without meaning (if possible)
- **Granularity** (what steps? what objects?)
- **Level of details** (descriptions? configuration?)

Different steps in provenance management

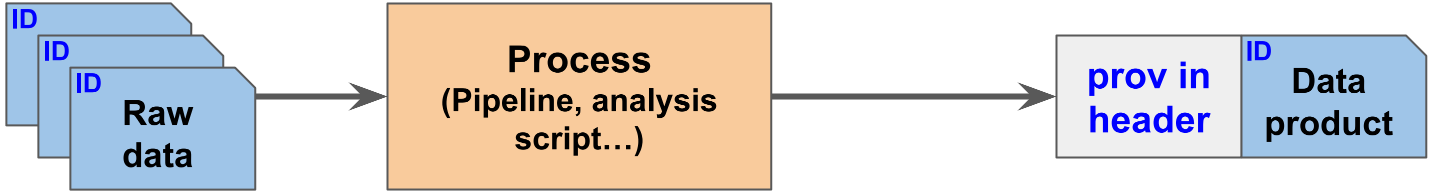
- How to **capture** the provenance information
- How to **store** this information
- How to **access** it
- How to **visualize** a provenance graph





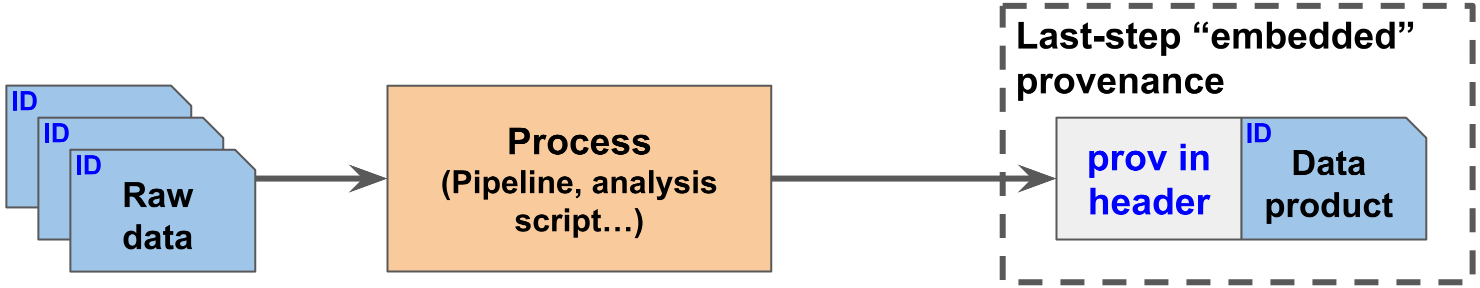
A provenance information management system

- What scientists generally have in mind:



A provenance information management system

- What scientists generally have in mind:



Last-step minimum provenance

https://etherpad.in2p3.fr/p/miniprov

- List of keywords
- Embedded in the entity
- Gives information on last activity and context
- Should enable full provenance reconstruction

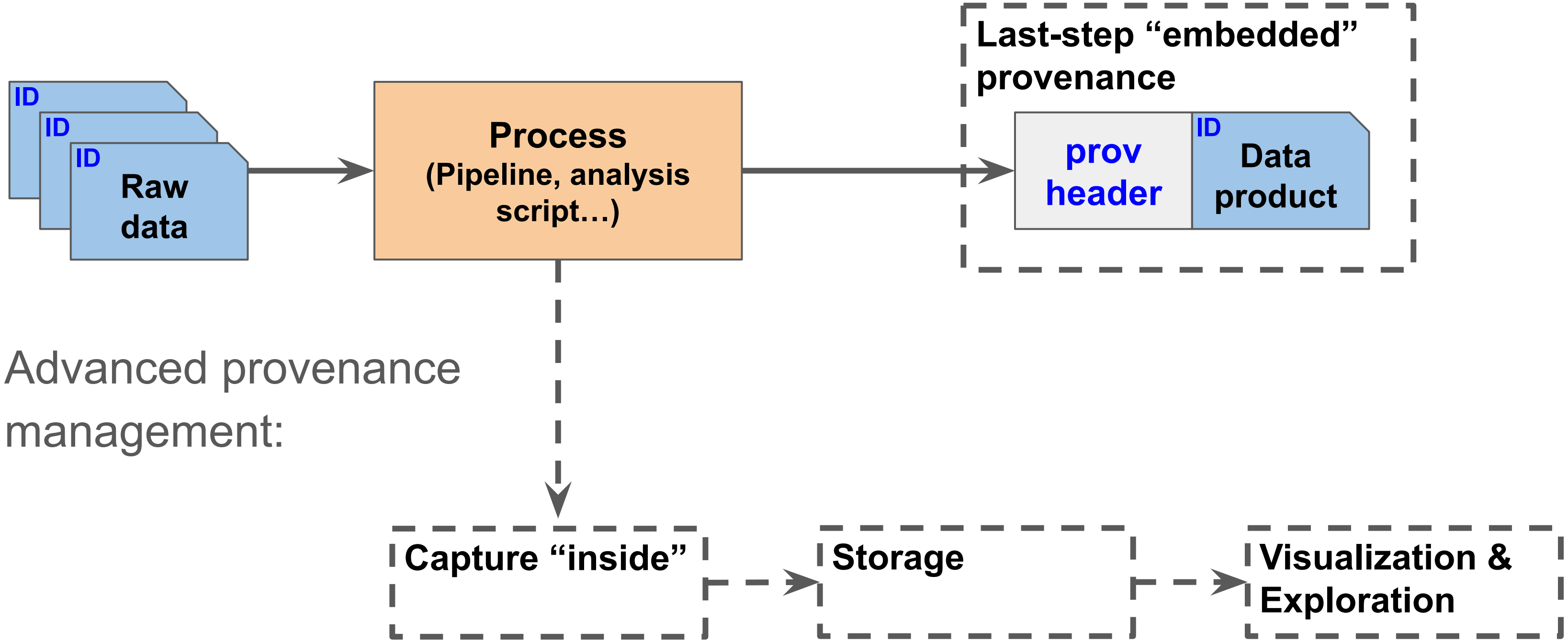
⇒ Such a list is a restriction of the full provenance information, that is more easily stored in a header (FITS) or a flat table (**same as ObsCore**)

⇒ Identifiers must be “resolvable” to provide further provenance information (**through an access protocol**)

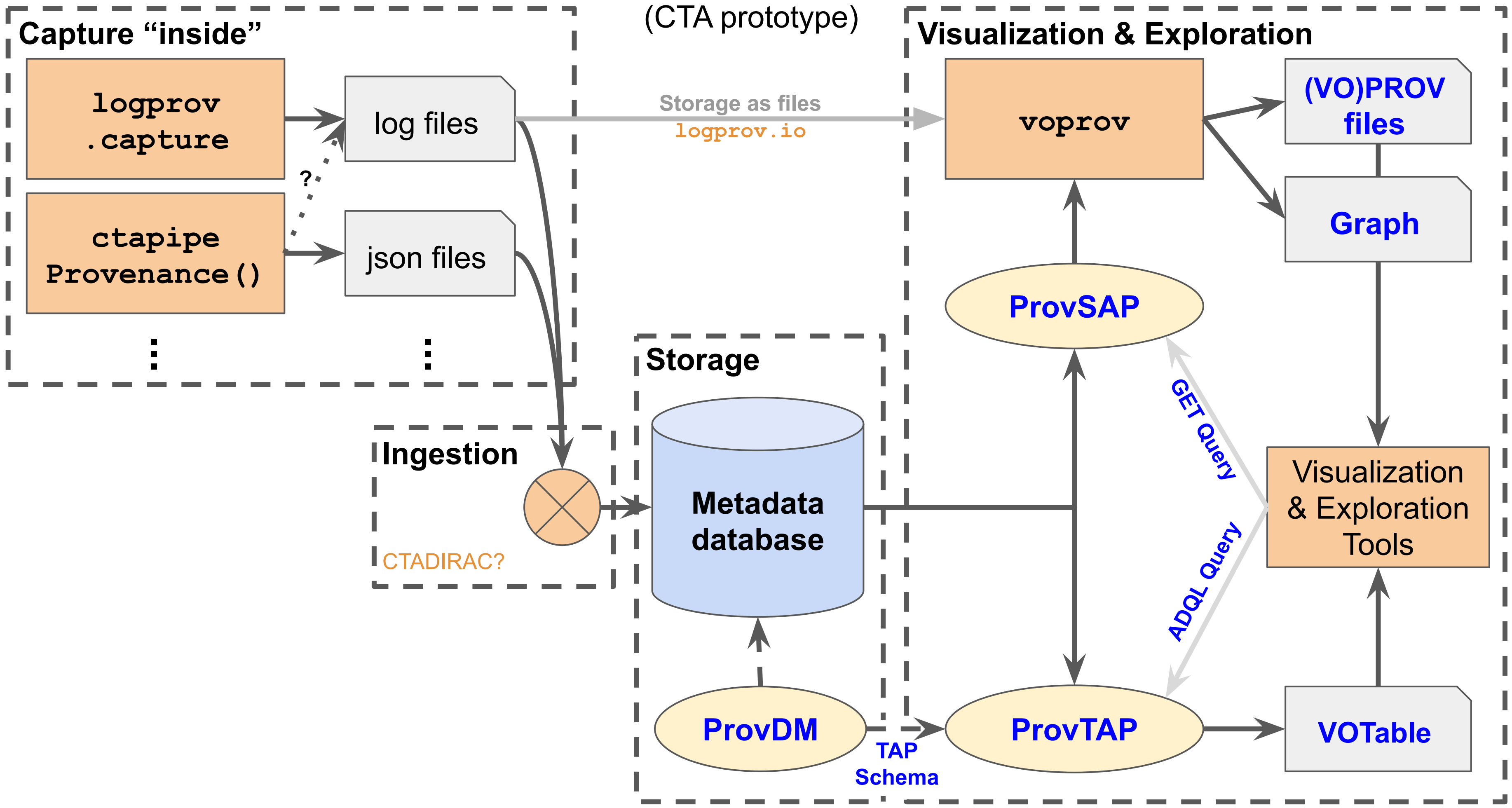
- entity_
  - id
  - creation_time
  - name
  - location
  - comment
  - content_type
  - description
- agent_
  - name
  - email
  - affiliation
- activity_description_
  - name
  - version
  - docurl
- activity_
  - id
  - name
  - type
  - start_time
  - stop_time
- used
  - [entity_id]
- generated
  - [entity_id]

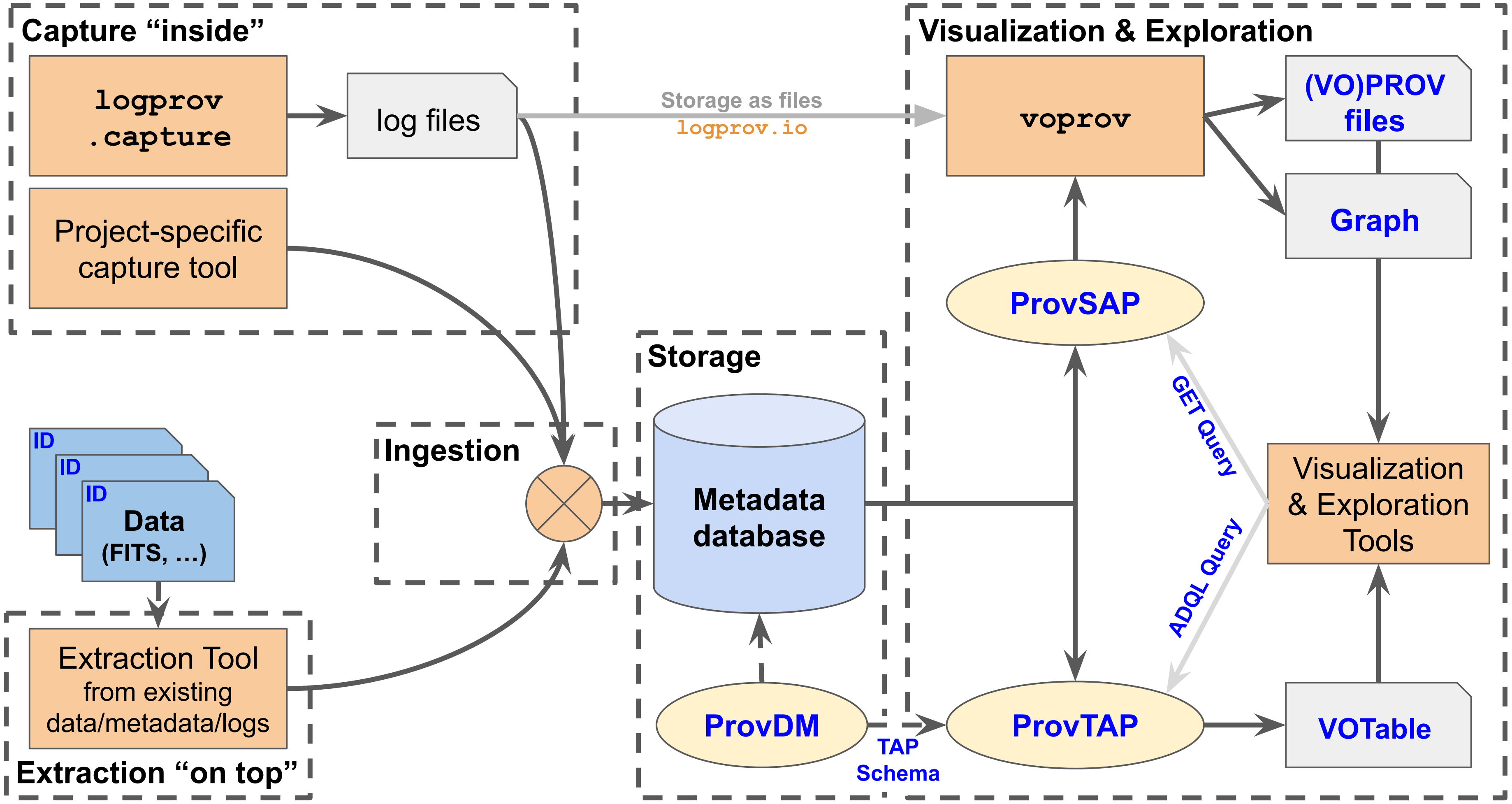
A provenance information management system

- What scientists generally have in mind:


ID
ID
ID
Raw data
Process
(Pipeline, analysis script…)
Last-step “embedded” provenance
prov header
ID
Data product
Capture “inside”
Storage
Visualization & Exploration

- Advanced provenance management:


(CTA prototype)
Capture “inside”
logprov
.capture
ctapipe
Provenance()
?
log files
json files
Storage as files
logprov.io
Ingestion
CTADIRAC?
Storage
Metadata
database
ProvDM
TAP
Schema
Visualization & Exploration
voprov
(VO)PROV
files
Graph
ProvSAP
GET Query
ADQL Query
Visualization
& Exploration
Tools
ProvTAP
VOTable


Capture "inside"
logprov .capture
log files
Project-specific capture tool
Storage as files
logprov.io
Visualization & Exploration
voprov
(VO)PROV files
Graph
ProvSAP
GET Query
ADQL Query
Visualization & Exploration Tools
ID
ID
ID
Data (FITS, ...)
Ingestion
Storage
Metadata database
Extraction Tool from existing data/metadata/logs
Extraction "on top"
ProvDM
TAP Schema
ProvTAP
VOTable

`voprov` Python package

- `prov` is a Python package that implements W3C PROV
    - https://github.com/trungdong/prov
- `voprov` extends `prov` to implement the IVOA Provenance DM
    - https://github.com/sanguillon/voprov
    - *Development Lead*: Jean-François Sornay, Michèle Sanguillon
    - *Contributors*: Mathieu Servillat, Mireille Louys, François Bonnarel, Catherine Boisson
- Main features
    - Description and Configuration classes and relations
    - Coloured graph to visualize IVOA objects
    - Conversion to a W3C graph

`logprov` Python package

- `logprov`: https://github.com/mservillat/logprov
- Capture provenance transparently
  - For Python tools (in particular data analysis on a laptop)
  - Make provenance info compatible between different software (ctapipe, gammapy, …)
  - Minimum effort from the developers, reuse existing tools
- Development drivers
  - Use **logging** system
    - known by developers, already integrated (base Python package)
    - log **dictionaries** to structure the information
  - **Non-intrusive** addition of code
    - use **decorators** for classes and methods
  - Fill with useful information
    - **descriptions** in a `definition.yaml` file
    - log parameters, arguments...

Provenance in **`gammapy`**

https://openprovenance.org/store/documents/1191.svg

1/ definition.yaml file for description/template
(already **integrated** to the code by the developers)

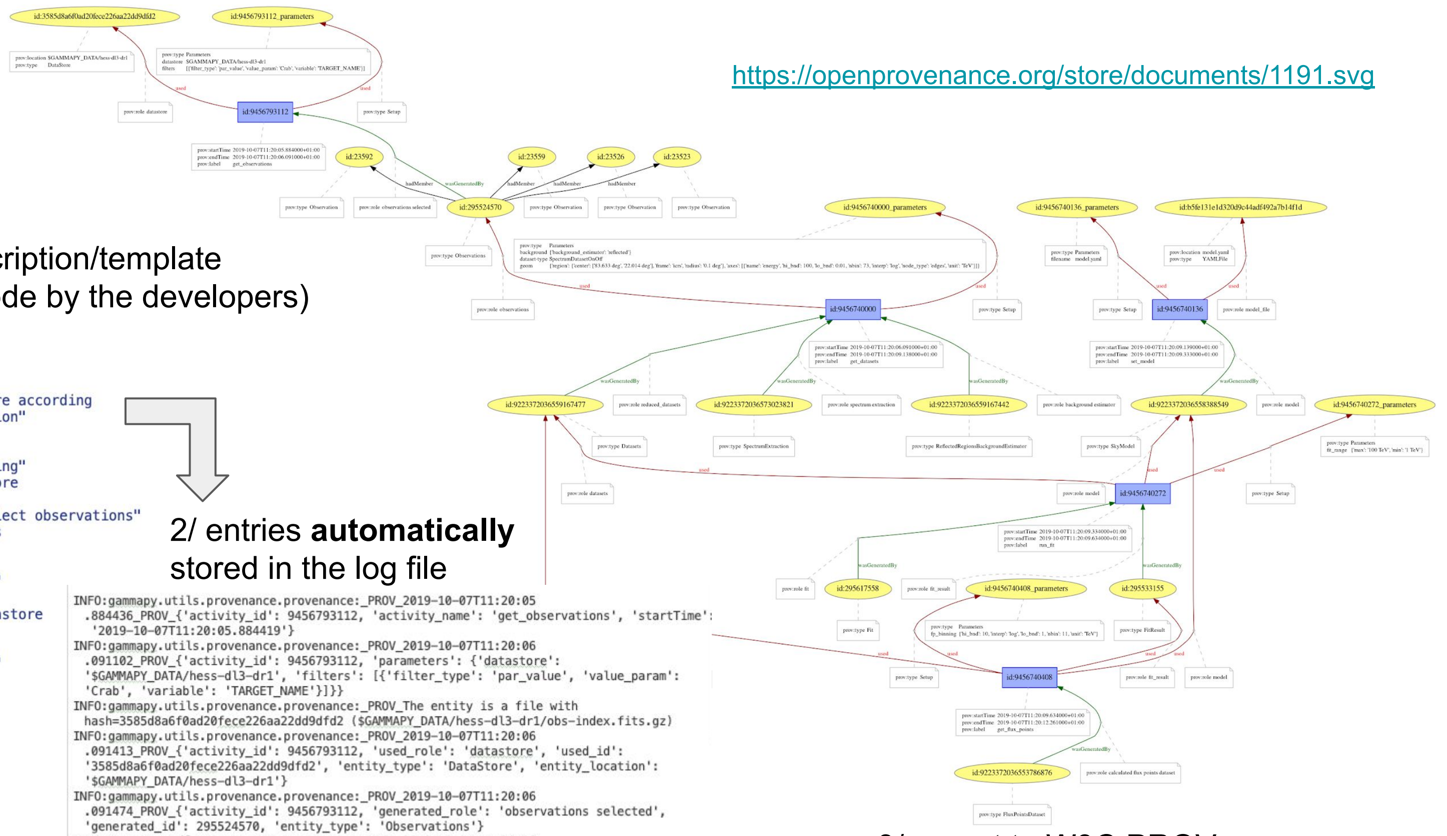
```yaml
activities:

    get_observations:
        description:
            "Fetch observations from the data store according
            to criteria defined in the configuration"
        parameters:
            - name: datastore
              description: "DataStore path as string"
              value: settings.observations.datastore
            - name: filters
              description: "Filter criteria to select observations"
              value: settings.observations.filters
        usage:
            - role: datastore
              description: "DataStore object file"
              entityType: DataStore
              location: settings.observations.datastore
        generation:
            - role: observations selected
              description: "Observations selected"
              entityType: Observations
              value: observations
              has_members:
                  entityType: Observation
                  list: observations.list
                  id: obs_id
                  namespace: ""

    get_datasets:
        description: "Produce reduced datasets"
        parameters:
```

2/ entries **automatically** stored in the log file

```
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:05
 .884436_PROV_{'activity_id': 9456793112, 'activity_name': 'get_observations', 'startTime':
 '2019-10-07T11:20:05.884419'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091102_PROV_{'activity_id': 9456793112, 'parameters': {'datastore':
 '$GAMMAPY_DATA/hess-dl3-dr1', 'filters': [{'filter_type': 'par_value', 'value_param':
 'Crab', 'variable': 'TARGET_NAME'}]}}
INFO:gammapy.utils.provenance.provenance:_PROV_The entity is a file with
 hash=3585d8a6f0ad20fece226aa22dd9dfd2 ($GAMMAPY_DATA/hess-dl3-dr1/obs-index.fits.gz)
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091413_PROV_{'activity_id': 9456793112, 'used_role': 'datastore', 'used_id':
 '3585d8a6f0ad20fece226aa22dd9dfd2', 'entity_type': 'DataStore', 'entity_location':
 '$GAMMAPY_DATA/hess-dl3-dr1'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091474_PROV_{'activity_id': 9456793112, 'generated_role': 'observations selected',
 'generated_id': 295524570, 'entity_type': 'Observations'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091527_PROV_{'entity_id': 295524570, 'member_id': 23592, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091571_PROV_{'entity_id': 295524570, 'member_id': 23523, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091613_PROV_{'entity_id': 295524570, 'member_id': 23526, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091653_PROV_{'entity_id': 295524570, 'member_id': 23559, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
 .091691_PROV_{'activity_id': 9456793112, 'endTime': '2019-10-07T11:20:06.091068'}
```



3/ export to W3C PROV
or search in provenance records

@ M. Servillat & J. E. Ruiz

ProvSAP: Simple Access Protocol

- Simple HTTP endpoint
- Query string with arguments:
  - **ID** = a9b7e2 (activity or entity)
  - **DEPTH** = ALL / 1..
  - **DIRECTION** = FORWARD / BACKWARD
  - **RESPONSEFORMAT** = PROV-SVG / PROV-JSON / PROV-XML
  - **MODEL** = IVOA / W3C
  - **AGENTS** = 0 / 1
  - **CONFIGURATION** = 0 / 1
  - **DESCRIPTIONS**= 0 / 1 / 2
  - **ATTRIBUTES** = 0 / 1
- https://voparis-uws-test.obspm.fr/provsap?**ID**=a9b7e2&**DESCRIPTIONS**=1&**CONFIGURATION**=1&**ATTRIBUTES**=0&**DEPTH**=ALL&**MODEL**=IVOA

See ADASS XXX proceedings on OPUS : https://arxiv.org/abs/2101.08683

ProvTAP and last-step provenance

François Bonnarel @ IVOA Nov 2020:
https://wiki.ivoa.net/internal/IVOA/InterOpNov2020DM/ProvTAPevolution.pdf

- ProvTAP allows to discover « data » by constraining provenance features
    - **Table Access Protocol** using ADQL for queries and a TAP Schema
    - It's a « reverse » mechanism
- TAP Schema
    - From Provenance Data Model
    - **Full**: maps all classes to a relational database
    - denormalize some classes? (descriptions, configuration)
    - add simplified views?

Last-step provenance of an entity in ProvTAP

Create a view on top of **full ProvTAP** service joining :

- The entity details
- The generating activity details
- The associated agents details
- eventually the list of used entity ids

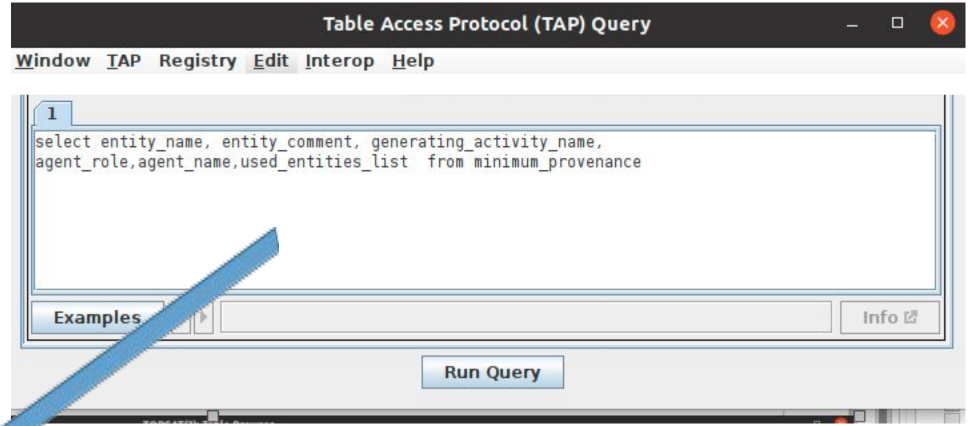


Table Access Protocol (TAP) Query
Window TAP Registry Edit Interop Help
1
Examples
Info
Run Query

```
select entity_name, entity_comment, generating_activity_name,
agent_role,agent_name,used_entities_list  from minimum_provenance
```

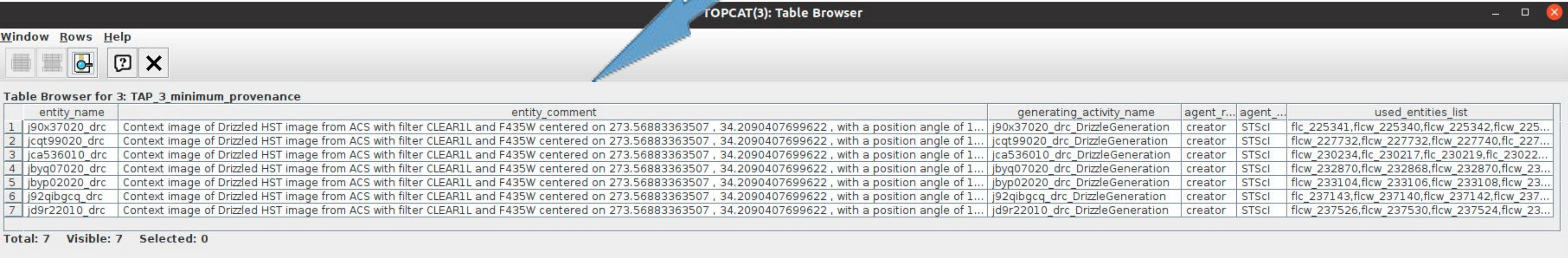


Table Browser for 3: TAP_3_minimum_provenance

	entity_name	entity_comment	generating_activity_name	agent_r...	agent_...	used_entities_list
1	j90x37020_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	j90x37020_drc_DrizzleGeneration	creator	STScI	flc_225341,flcw_225340,flcw_225342,flcw_225...
2	jcqt99020_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	jcqt99020_drc_DrizzleGeneration	creator	STScI	flcw_227732,flcw_227732,flcw_227740,flc_227...
3	jca536010_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	jca536010_drc_DrizzleGeneration	creator	STScI	flcw_230234,flc_230217,flc_230219,flc_23022...
4	jbyq07020_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	jbyq07020_drc_DrizzleGeneration	creator	STScI	flcw_232870,flcw_232868,flcw_232870,flcw_23...
5	jbyp02020_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	jbyp02020_drc_DrizzleGeneration	creator	STScI	flcw_233104,flcw_233106,flcw_233108,flcw_23...
6	j92qibgcq_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	j92qibgcq_drc_DrizzleGeneration	creator	STScI	flc_237143,flcw_237140,flcw_237142,flcw_237...
7	jd9r22010_drc	Context image of Drizzled HST image from ACS with filter CLEAR1L and F435W centered on 273.56883363507 , 34.2090407699622 , with a position angle of 1...	jd9r22010_drc_DrizzleGeneration	creator	STScI	flcw_237526,flcw_237530,flcw_237524,flcw_23...

Total: 7   Visible: 7   Selected: 0

@ F. Bonnarel