

# Tracking provenance information with the OPUS job controller

Mathieu Servillat

Observatoire de Paris - LUTH  
Paris Astronomical Data Centre

IVOA Santiago - Oct. 2017

# OPUS: Observatoire de Paris UWS Server

## Main features

- IVOA standards
  - Universal Worker System (**UWS**)
  - **Provenance** data model
- REST architecture
  - Python micro-framework: bottle.py
- Collaborative development
  - Git server at PADC (gitolite)
  - GitHub: <https://github.com/mservillat/OPUS>
- Tests and quality
  - Unit tests with unittest and webtest
  - Activity history with logging

## Prototype available

- <https://voparis-uws-test.obspm.fr>



# Job management at PADC

## Available structure

- Work cluster (Tycho)
- Job scheduler (SLURM)



## PADC projects

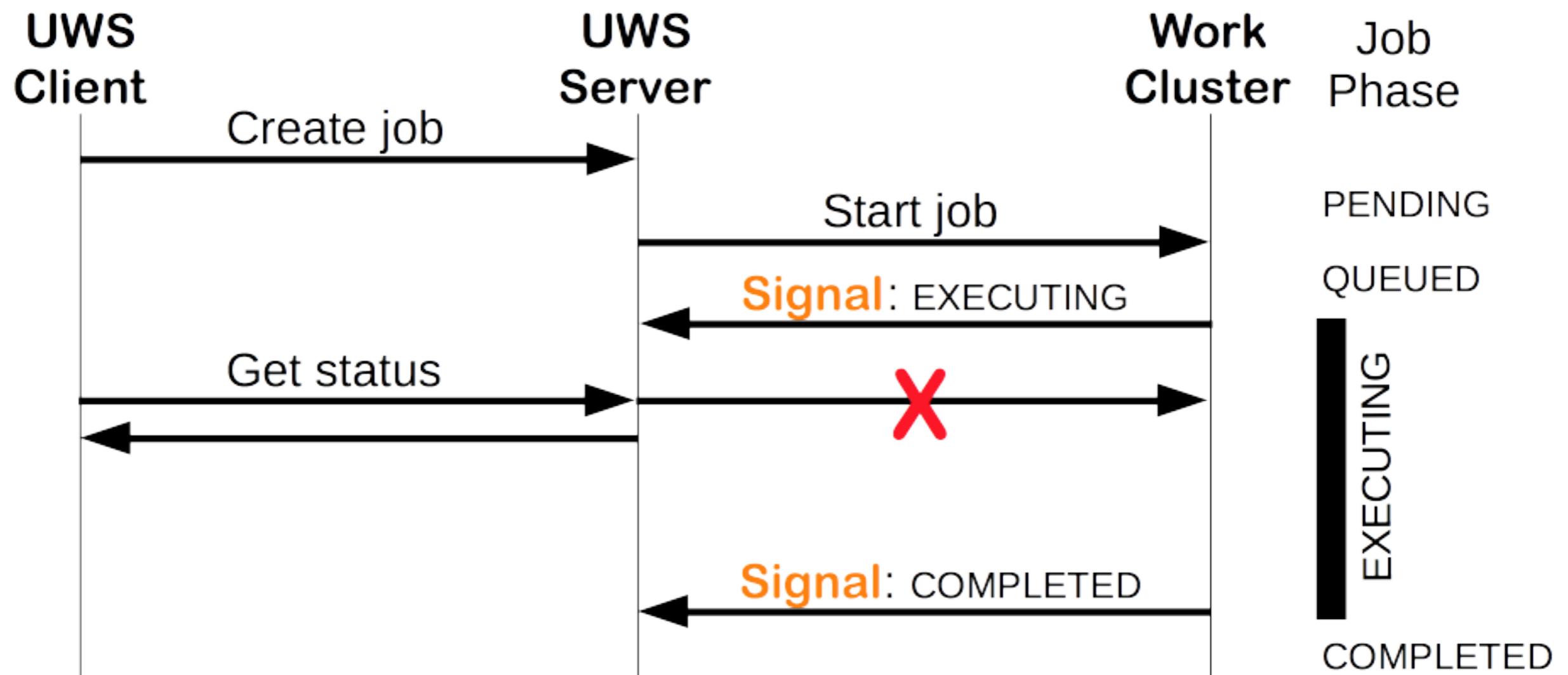
- Web based clients
  - Data access
  - Online data processing
  - Wrap simulation codes



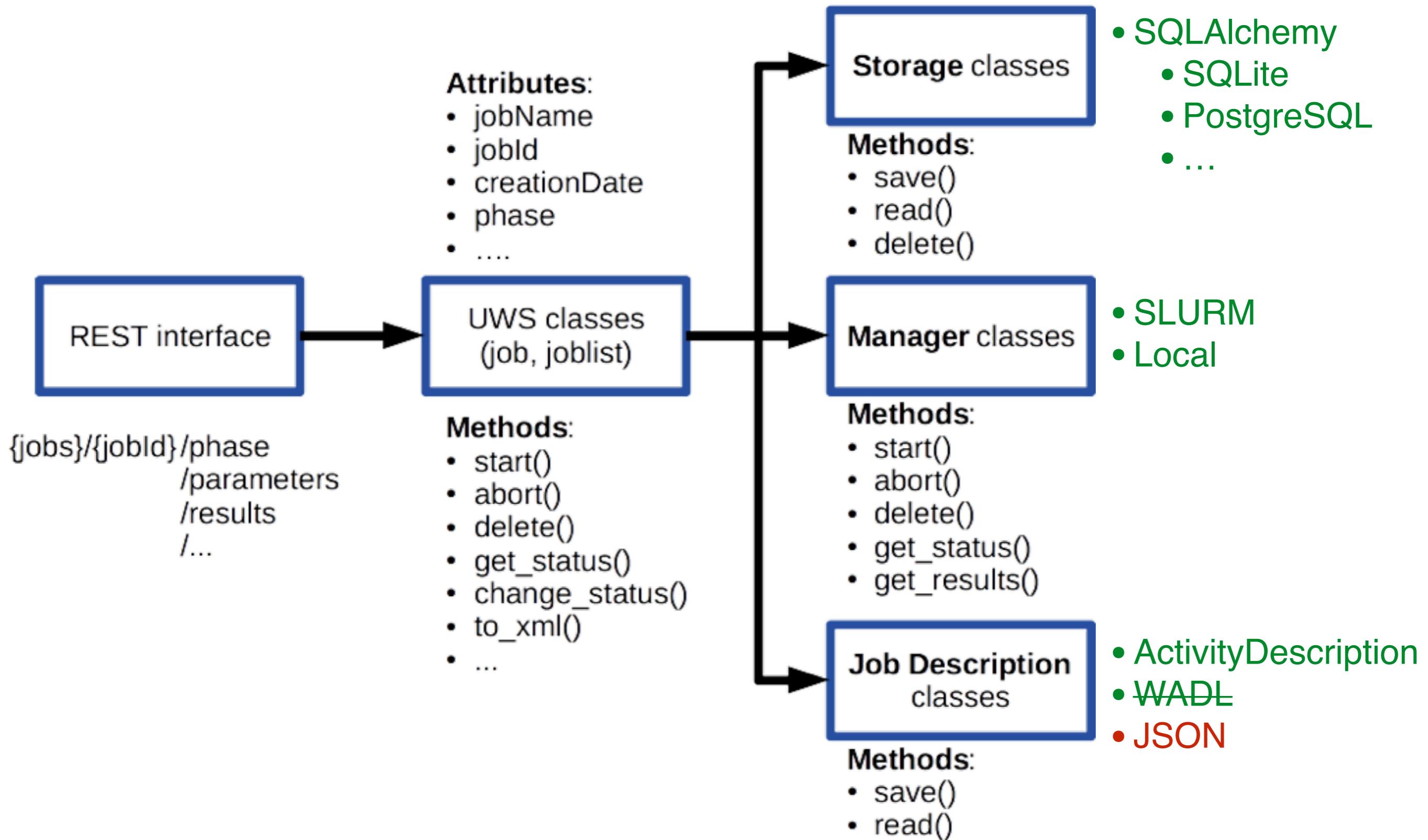
**Need a simple interface to computational resources**

# Inner workings

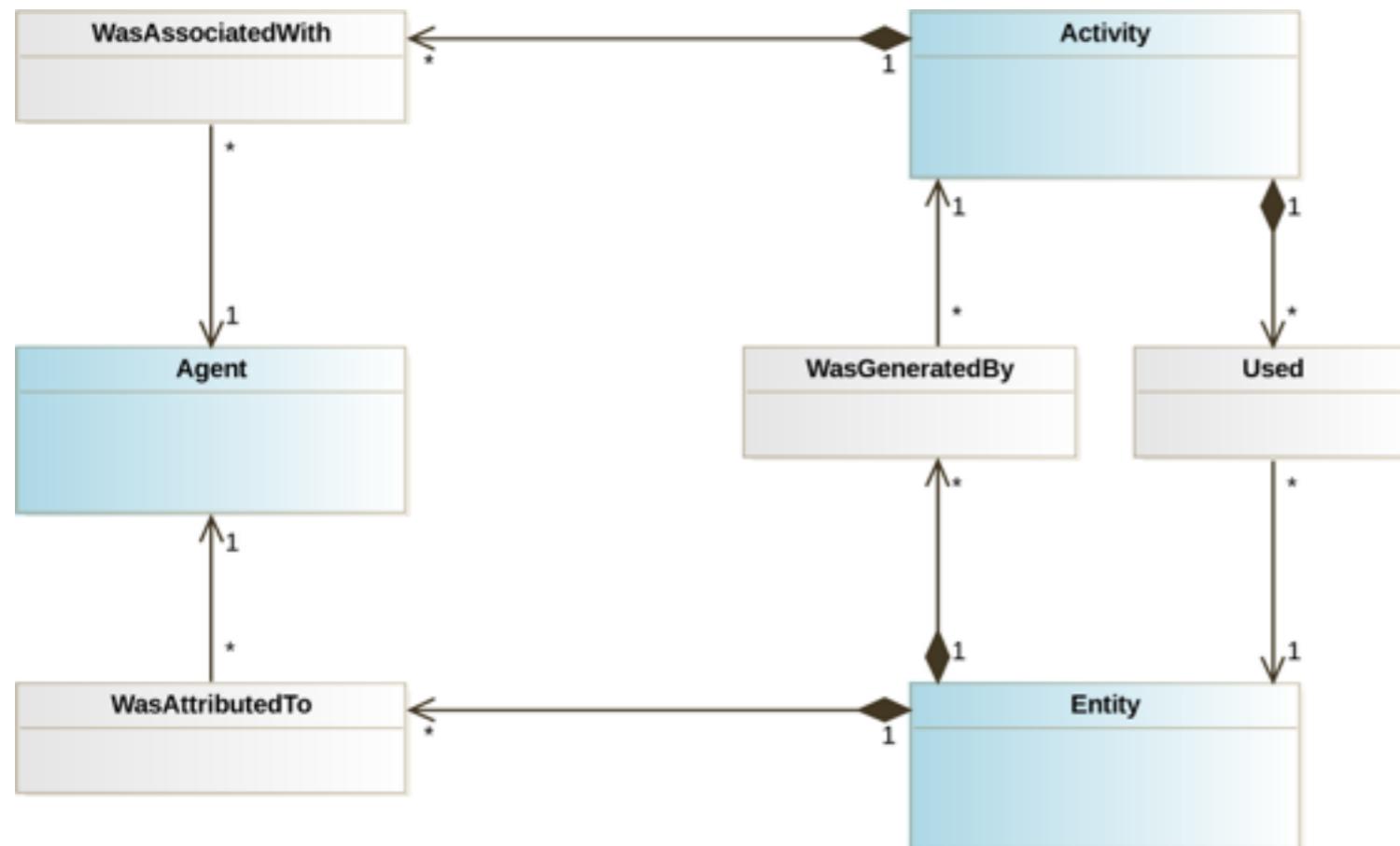
- Separate **job controller** from **work cluster**
  - Wait for work cluster **signals**
  - Avoid (too many) status queries to work cluster



# Server main classes



# Core Provenance Data Model



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

30 April 2013



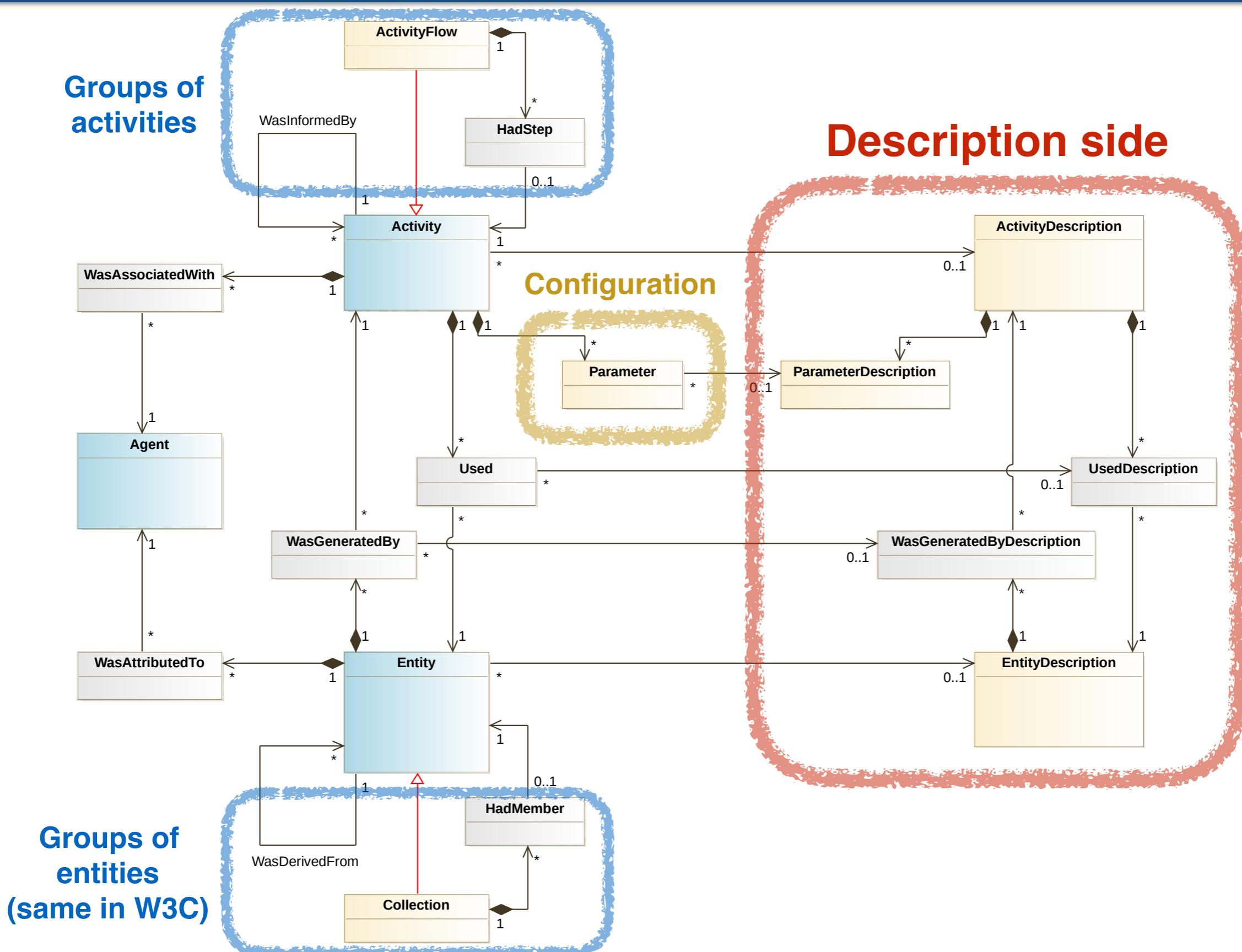
Word Wide Web Consortium

<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**

# IVOA Provenance Data Model diagram



# Serializations - ActivityDescription

```
<?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:WasGeneratedByDescription.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

# Client main features

- Javascript based + Python Flask
  - `UwsLib.js`: sends AJAX requests to the server
  - `uws_manager.js`: handles and displays responses
    - Integration with Bootstrap3
    - HTML pages with specified `<div>` elements  
(`id=joblist, parameters, results...`)
- **Job definition editor**
  - Interface to create the `ActivityDescription` file
    - parameters, used and generated entities
    - bash execution script
- **Job manager**
  - list jobs, create jobs, control jobs, view results

# Client - Job definition

OPUS   Job Definition   Job List   Signed in as admin

Job Definition   Validate Job   Copy script

Name: **gammapy\_spectral**   Load JDL   Get JDL   Job name.

Description: Use gammapy to generate spectra from a list of observations   Job description.

URL: <https://luthgitlab.obspm.fr/jlefa>

Contact name: Julien Lefaucheur

Contact email:

**Input**   obsids = 47802 47803 47804 47827 47   image/fits     List of input entities (e.g. files) used with their name and content type. The input is a File or an ID, possibly with a URL to resolve the ID and download the file (use \$ID in the URL template). If no URL is specified, the script itself should be able to resolve the ID and get the file. Note that an input can refer to a parameter (if it has the same name), e.g. the name of an input file used in the script.

Desc. List of runs  
File  or ID  and URL [http://url\\_to\\_the\\_input\\_file?id=\\$ID](http://url_to_the_input_file?id=$ID)

Add input   Remove all input

**Results**   spectrum = spectrum.fits   image/fits     List of possible results with their name and content type. A default name can be provided. Note that a result can refer to a parameter (if it has the same name), e.g. the name of an output file generated by the script.

Desc. Description

spectrum\_preview = spectrum.png   image/png  

Desc. Description

Add result   Remove all results

**Parameters**   configfile = make\_spectra.cfg   Req.?    xs:string     List of parameters, with name, default value, type and description. Specify if the parameter is required by checking the box (if not, the parameters won't be shown by the client and the default value will always be used).

Desc. Configuration file (generated by the script)

Options List of possible choices (comma-separated values)

Attr. unit=... ucd=... utype=... min=... max=...

RA = 329.7169379   Req.?    xs:double     A list of options can be specified (comma-separated values).

Desc. Target Right Ascension

# Client - Job list

OPUS Job Definition Job List Signed in as user

Job List for gammipy\_spectra Refresh Job List Create Test Job Create New Job

Type	Start Time	Destruction Time	Phase	Details	Control
gammipy_spectra	2017-10-02 10:47:07	2017-11-01 10:47:05	COMPLETED	<a href="#">Properties</a> <a href="#">Parameter</a> <a href="#">Results</a>	<a href="#">Start</a> <a href="#">Abort</a> <a href="#">Delete</a>
gammipy_spectra		2017-11-01 10:47:03	PENDING	<a href="#">Properties</a>	
gammipy_spectra	2017-09-29 15:07:52	2017-10-29 15:07:51	COMPLETED	<a href="#">Properties</a>	
gammipy_spectra	2017-09-29 14:55:10	2017-10-29 14:55:09	ABORTED	<a href="#">Properties</a>	
gammipy_spectra	2017-09-29 14:21:20	2017-10-29 14:21:19	COMPLETED	<a href="#">Properties</a>	

**Results** (highlighted with a red circle and arrow)

**Job Results**

- spectrum: [Download \[image/fits\]](#)
- spectrum\_preview: [Download \[image/png\]](#)

Tracking of Provenance informations

The provenance graph shows the following relationships:

- A blue rectangle labeled "gammipy\_spectra:a0664356-e938-4404-b995-307a72ab6f23" is associated with a yellow oval "gammipy\_spectra:spectrum\_preview" via "wasGeneratedBy".
- The same blue rectangle is associated with a yellow oval "gammipy\_spectra:spectrum" via "wasGeneratedBy".
- The blue rectangle is associated with a yellow box "ctao:user" via "wasAssociatedWith".
- A small box at the bottom left contains provenance details:
  - prov:startTime: 2017-10-02T10:47:07
  - contact\_email: Julien Lefuscheur
  - contact\_name: Julien Lefuscheur
  - gammipy\_spectra:Dec: -30.2255883
  - gammipy\_spectra:RA: 329.7169379
  - gammipy\_spectra:configfile: make\_spectra.cfg
- Below the graph, two small boxes show the types of the generated files:
  - "gammipy\_spectra:spectrum\_preview": voprov:type image/png
  - "gammipy\_spectra:spectrum": voprov:type image/fits

Red arrows point from the "Results" section in the top panel to the "gammipy\_spectra:spectrum" oval in the provenance graph, and from the "gammipy\_spectra:spectrum" oval to the corresponding plot.

The figure consists of two vertically stacked plots sharing a common x-axis: Energy [TeV].

The top plot shows the differential flux  $dnde$  (erg / (cm<sup>2</sup> s)) on a logarithmic y-axis (from  $10^{-11}$  to  $10^{-10}$ ) versus Energy [TeV] on a logarithmic x-axis (from  $10^0$  to  $10^1$ ). Data points are shown with error bars, and a solid blue line represents the model fit.

The bottom plot shows the Residuals on a linear y-axis (from -1.0 to 1.0) versus Energy [TeV] on a logarithmic x-axis (from  $10^0$  to  $10^1$ ). The residuals are mostly centered around zero, indicating a good fit.

Mathieu Servillat (Observatoire de Paris)

IVOA Santiago, Chile - Oct 2017

11

# Serializations - W3C PROV formats

```
<prov:document xmlns:ctadata="ivo://vopdc.obspm/cta#" xmlns:ctajob="http://www.cta-observatory.org/ns/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/reast",
    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"
    },
  }
},
```