

# Tracking provenance information with the OPUS job controller

Mathieu Servillat

Observatoire de Paris - LUTH  
Paris Astronomical Data Centre



Laboratoire Univers et Théories



# 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**

# OPUS source code and docs



**OPUS** (**O**bservatoire de **P**aris **UWS** **S**ystem) is a job control system developed using the Python micro-framework `bottle.py`.

The **Universal Worker System pattern v1.1 (UWS)** as defined by the International Virtual Observatory Alliance (IVOA) is implemented as a REST service to control job execution on a work cluster.

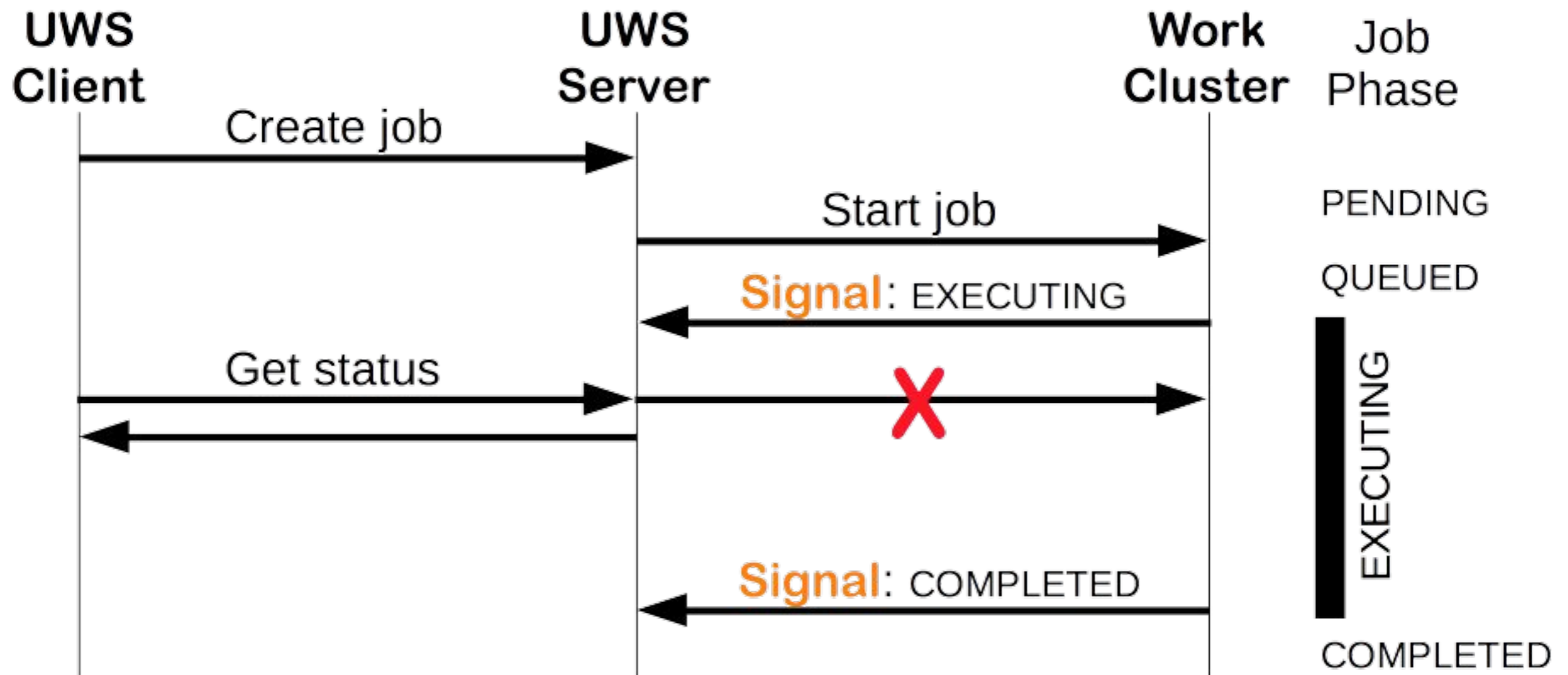
OPUS also follows the proposed **IVOA Provenance Data Model** to capture and expose the provenance information of jobs and results.

<https://github.com/ParisAstronomicalDataCentre/OPUS>

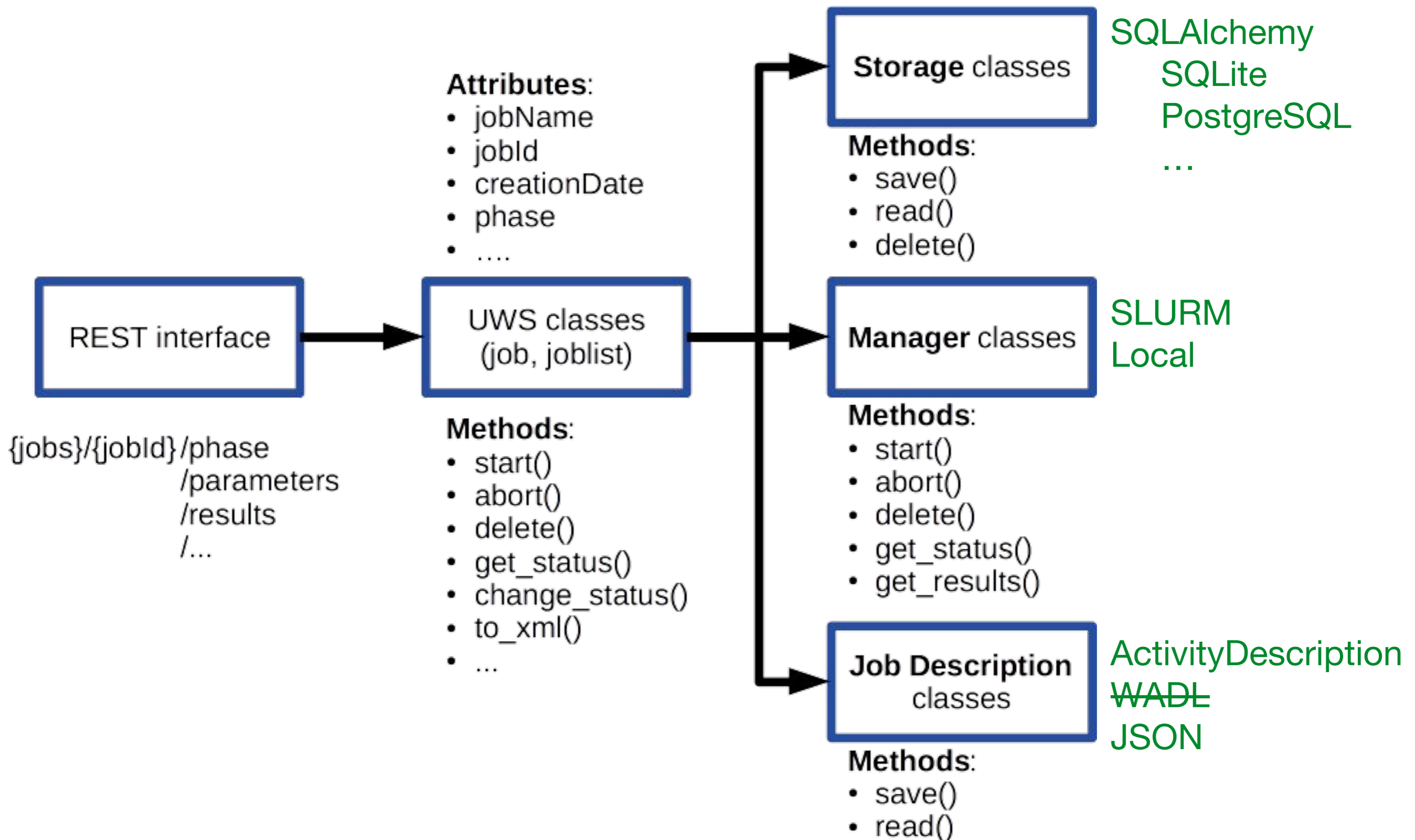
<https://uws-server.readthedocs.io/en/latest/>

# Inner workings

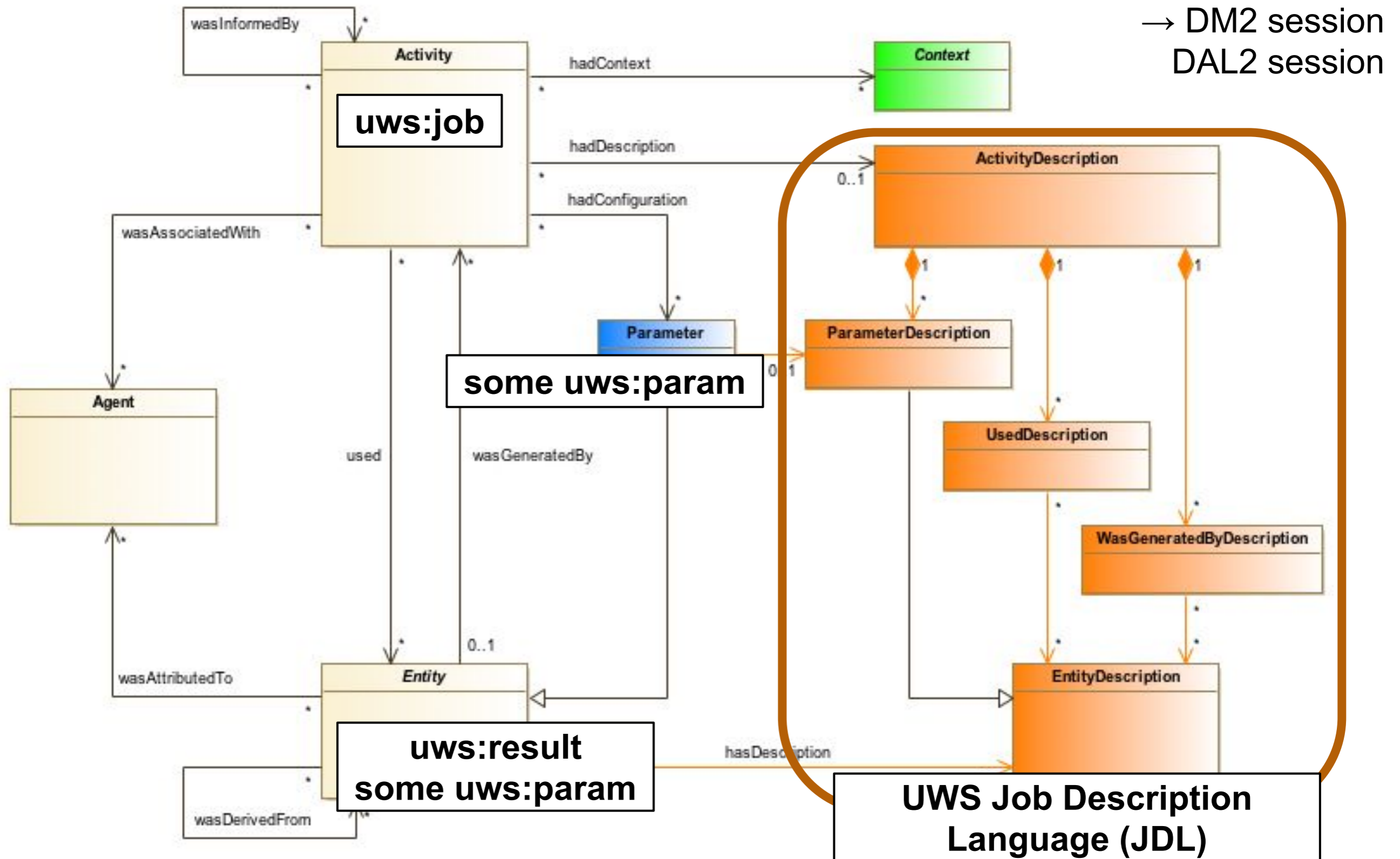
- Separate **job controller** from **work cluster**
  - Wait for work cluster **signals**
  - Avoid (too many) status queries to work cluster



# Server main classes



# IVOA Provenance Data Model diagram



# Serializations - ActivityDescription

```
<RESOURCE ID="gammapy_maps" name="gammapy_maps" type="meta" utype="voprov:ActivityDescription">
```

```
<DESCRIPTION>Use gammapy to generate a count map from a list of observations</DESCRIPTION>
```

```
<!-- Service Descriptor -->
```

```
<PARAM name="accessURL" datatype="char" arraysize="*" value="https://voparis-uws-test/rest/gammapy_maps" />
```

```
<PARAM name="standardID" datatype="char" arraysize="*" value="ivo://ivoa.net/std/SODA#1.0" />
```

```
<!-- Activity Description -->
```

```
<PARAM name="type" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.type"/>
```

```
<PARAM name="subtype" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.subtype"/>
```

```
<PARAM name="annotation" datatype="char" arraysize="*" value="Use gammapy to generate a count map from a list of
```

```
<PARAM name="version" datatype="char" arraysize="*" value="None" utype="voprov:ActivityDescription.version"/>
```

```
<PARAM name="doculink" datatype="char" arraysize="*" value="https://luthgitlab.obspm.fr/jlefaucheur/hess_release
```

```
<PARAM name="contact_name" datatype="char" arraysize="*" value="Julien Lefaucheur" utype="voprov:Agent.name"/>
```

```
<PARAM name="contact_email" datatype="char" arraysize="*" value="" utype="voprov:Agent.email"/>
```

```
<!-- UWS job attributes -->
```

```
<PARAM name="executionDuration" datatype="int" value="600" utype="uws:Job.executionDuration"/>
```

```
<PARAM name="quote" datatype="int" value="120" utype="uws:Job.quote"/>
```

```
<!-- UWS parameters (Provenance Entities or Parameters) -->
```

```
<GROUP name="InputParams">
```

```
<PARAM ID="obs_ids" arraysize="*" datatype="char" name="obs_ids" value="47802 47803 47804
```

```
<DESCRIPTION>List of runs</DESCRIPTION>
```

```
</PARAM>
```

```
<PARAM ID="RA" datatype="double" name="RA" value="329.7169379" unit="deg"...>
```

```
<PARAM ID="Dec" datatype="double" name="Dec" value="10.1234567" unit="deg"...>
```

```
<PARAM ID="nxpix" arraysize="*" datatype="int" name="nxpix" value="1000" />
```

```
<DESCRIPTION>Number of pixels
```

```
<VALUES>
```

```
<MIN value="0"/>
```

```
<MAX value="1000"/>
```

```
</VALUES>
```

```
</PARAM>
```

```
<PARAM ID="nypix" arraysize="*" datatype="int" name="nypix" value="1000" />
```

```
<PARAM ID="binsz" datatype="float" name="binsz" value="0.5" />
```

```
</GROUP>
```

```
<!-- Used Entities -->
```

```
<GROUP name="Used">
```

```
<GROUP name="obs_ids" utype="voprov:UsedDescription" ref="obs_ids">
```

```
<PARAM arraysize="*" datatype="char" name="role" utype="voprov:UsedDescription.role" value="DL3"/>
```

```
<PARAM arraysize="*" datatype="char" name="location" utype="voprov:EntityDescription.location" value="" />
```

```
<PARAM arraysize="*" datatype="char" name="content_type" utype="voprov:EntityDescription.content_type" value="" />
```

```
</GROUP>
```

```
</GROUP>
```

```
<!-- Generated Entities / UWS results -->
```

```
<GROUP name="Generated" utype="voprov:WasGeneratedBy">
```

```
<GROUP name="count_map" utype="voprov:EntityDescription">
```

```
<DESCRIPTION>Count map</DESCRIPTION>
```

```
<PARAM arraysize="*" datatype="char" name="role" utype="voprov:UsedDescription.role" value="DL4 image"/>
```

```
<PARAM arraysize="*" datatype="char" name="default" utype="voprov:Entity.id" value="count_map.fits"/>
```

```
<PARAM arraysize="*" datatype="char" name="content_type" utype="voprov:EntityDescription.content_type" value="" />
```

```
</GROUP>
```

```
<GROUP name="count_preview" utype="voprov:EntityDescription">
```

```
<DESCRIPTION>Count map preview</DESCRIPTION>
```

**VOTable**

**DataLink Service Descriptor**

**UWS Job Description Language**

**Provenance ActivityDescription**

# Client main features

## Javascript based + Python Flask

- `UwsLib.js`: sends AJAX requests to the server
- `uws_client.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

## Client/Server administration

- users, authorizations, server config, jobs available



# Definition of the `gammapy_maps` job

OPUS [Job Definition](#) [Job List](#) Signed in as `testuser`

### Job Definition

**Name**  [Load JDL](#) [Get JDL](#) Job name.

**Description**  Job description.

**URL**

**Contact name**

**Contact email**

**Parameters**

<b>RA</b>	=	329.7169379	Req.? <input checked="" type="checkbox"/>	xs:string	↑	↓	×
Desc.	Target Right Ascension						
Options	List of possible choices (comma-separated values)						
Attr.	unit=... ucd=... utype=... min=... max=...						
<b>Dec</b>	=	-30.2255883	Req.? <input checked="" type="checkbox"/>	xs:string	↑	↓	×
Desc.	Target Declination						
Options	List of possible choices (comma-separated values)						
Attr.	unit=... ucd=... utype=... min=... max=...						
<b>npxix</b>	=	400	Req.? <input checked="" type="checkbox"/>	xs:string	↑	↓	×

**Input data**

<b>obsids</b>	=	47802 4780:	Mult.	*	image/fits	↑	↓	×
Desc.	List of runs							
File <input type="radio"/> or value <input type="radio"/> or ID <input checked="" type="radio"/> + access URL	<input type="text" value="http://url_to_the_input_file?id=\$ID"/>							

**Results**

<b>count_map</b>	=	count_map.fits	image/fits	↑	↓	×
Desc.	Description					
<b>count_preview</b>	=	count_map.png	image/png	↑	↓	×
Desc.	Description					
<b>significance_map</b>	=	significance_map.fits	image/fits	↑	↓	×

List of parameters, with name, default value, type and description. Specify if the parameter is required by checking the box. A restricted list of options can be specified (comma-separated values). Additional attributes can be defined (unit, ucd, utype, min, max).

List of input entities (e.g. files) 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

List of possible results with their name and content type.

**OPUS**  
Observatoire de Paris  
UWS Server

# Submission of a `gammapy_maps` job

- OPUS reads the **ActivityDescription** file to generate a form
- This form also carries the **Obscore** metadata

OPUS [Job Definition](#) [Job List](#) Signed in as **testuser** ▾

Create new `gammapy_maps` job ← Back to job list

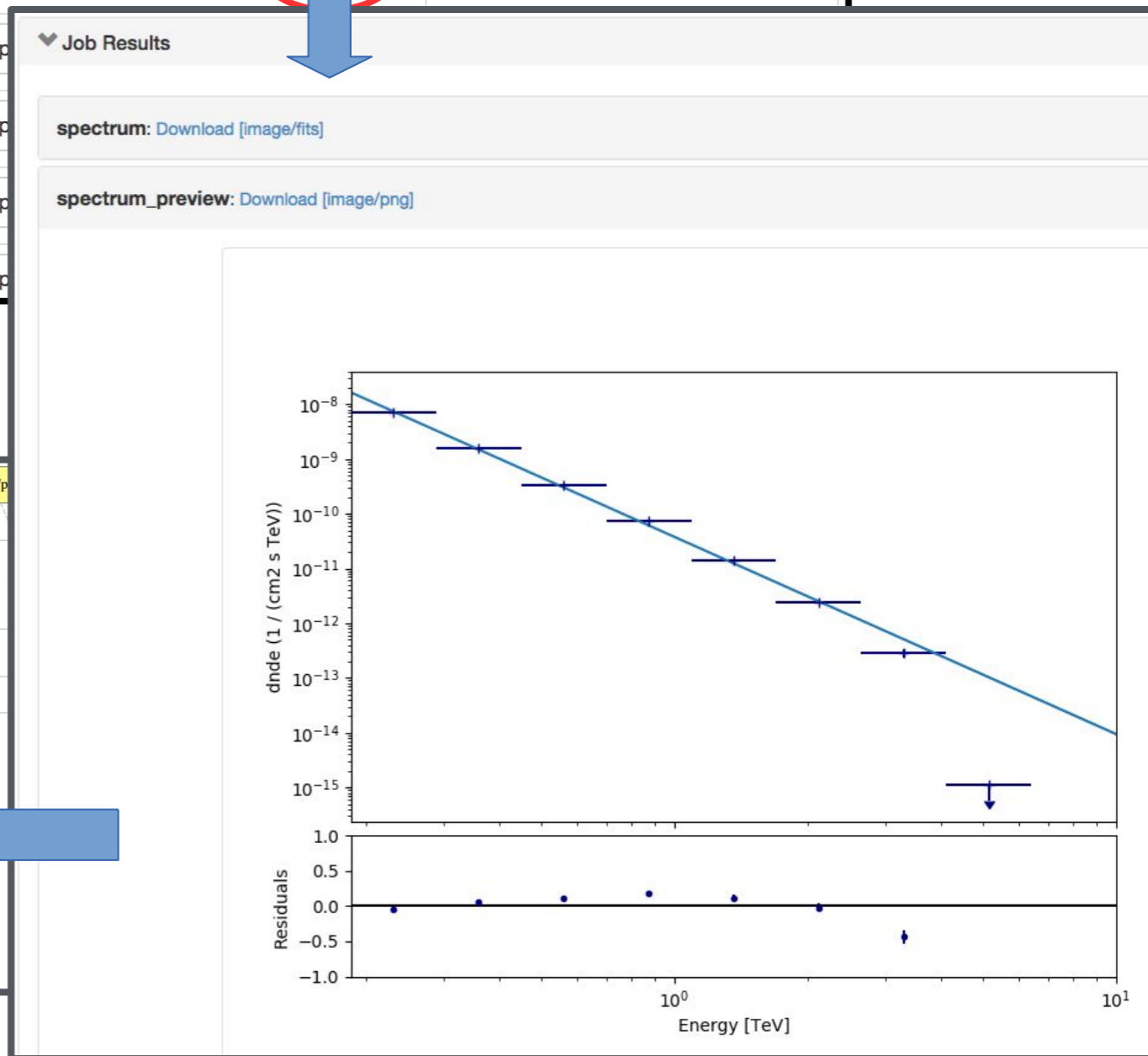
<b>obs_ids</b>	<input type="text" value="47802 47803 47804 47827 47828 47829 33787 33788 33789 33790"/>	List of runs
<b>RA</b>	<input type="text" value="329,7169379"/>	Target Right Ascension
<b>Dec</b>	<input type="text" value="-30,2255883"/>	Target Declination
<b>nypix</b>	<input type="text" value="400"/>	Number of pixels on the X axis
<b>nypix</b>	<input type="text" value="400"/>	Number of pixels on the Y axis
<b>binsz</b>	<input type="text" value="0,02"/>	Pixel size
<b>Add control parameters</b>	<input type="text" value="Chose parameter"/>	

# Results and Provenance

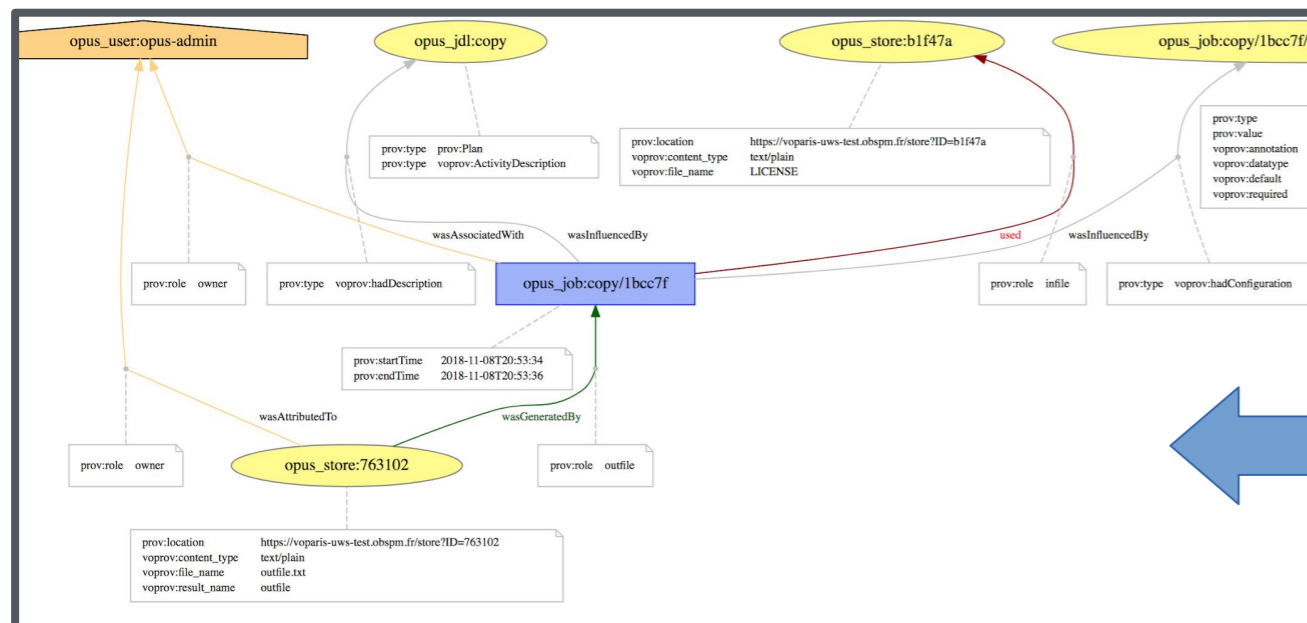
OPUS [Job Definition](#) [Job List](#) Signed in as user ▾

Job List for **gammapy\_spectra** Refresh Job List Create Test Job Create New Job

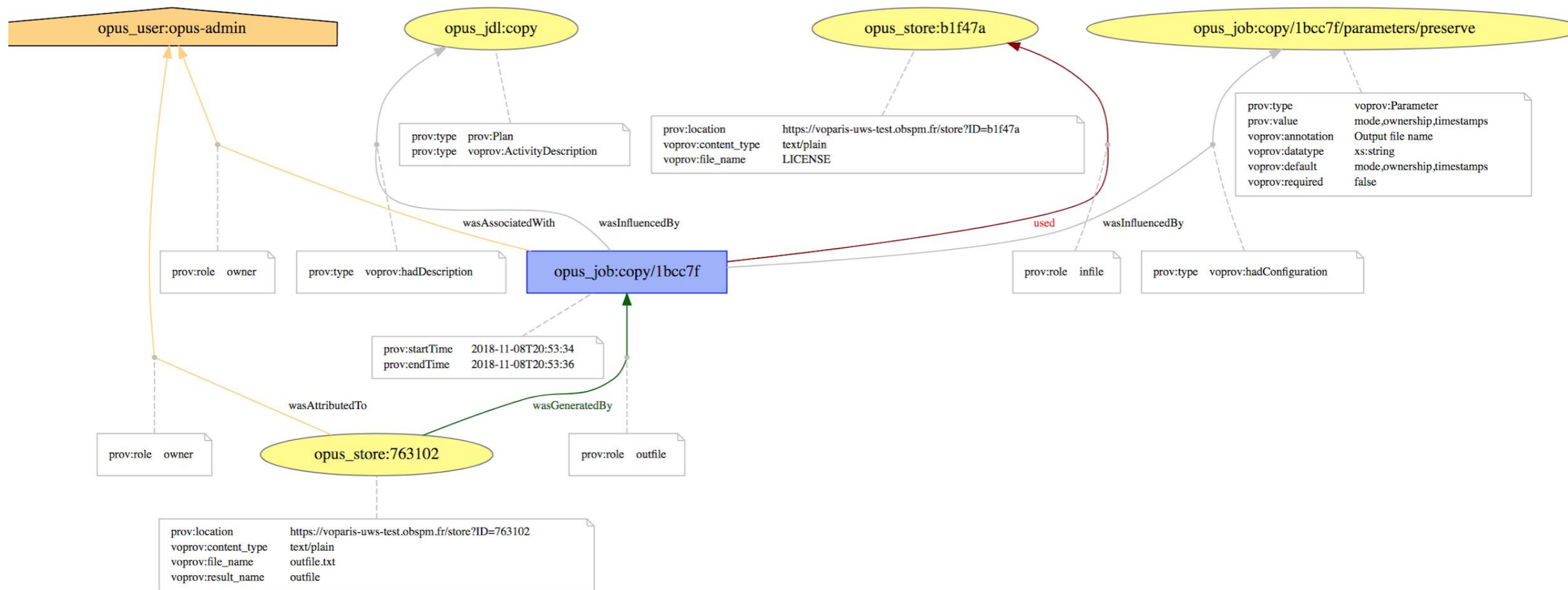
Type	Start Time	Destruction Time	Phase	Details	Control
gammapy_spectra	2017-10-02 10:47:07	2017-11-01 10:47:05	COMPLETED	<a href="#">Properties</a> <a href="#">Parameters</a> <a href="#">Results</a>	<a href="#">Start</a> <a href="#">Abort</a> <a href="#">Delete</a>
gammapy_spectra		2017-11-01 10:47:03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



## Tracking of Provenance informations

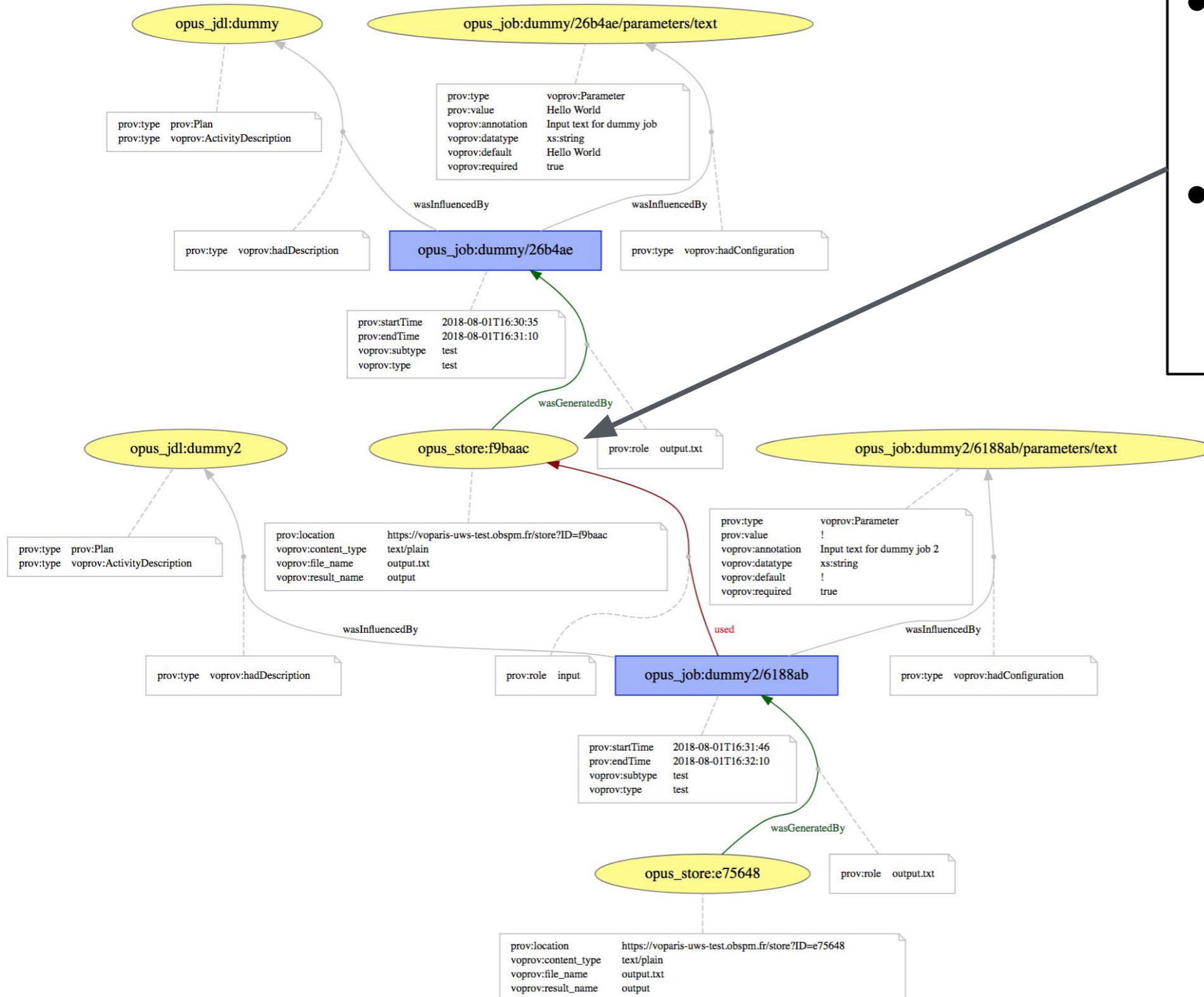


# Provides Provenance files



# Multiple-step provenance - ProvSAP

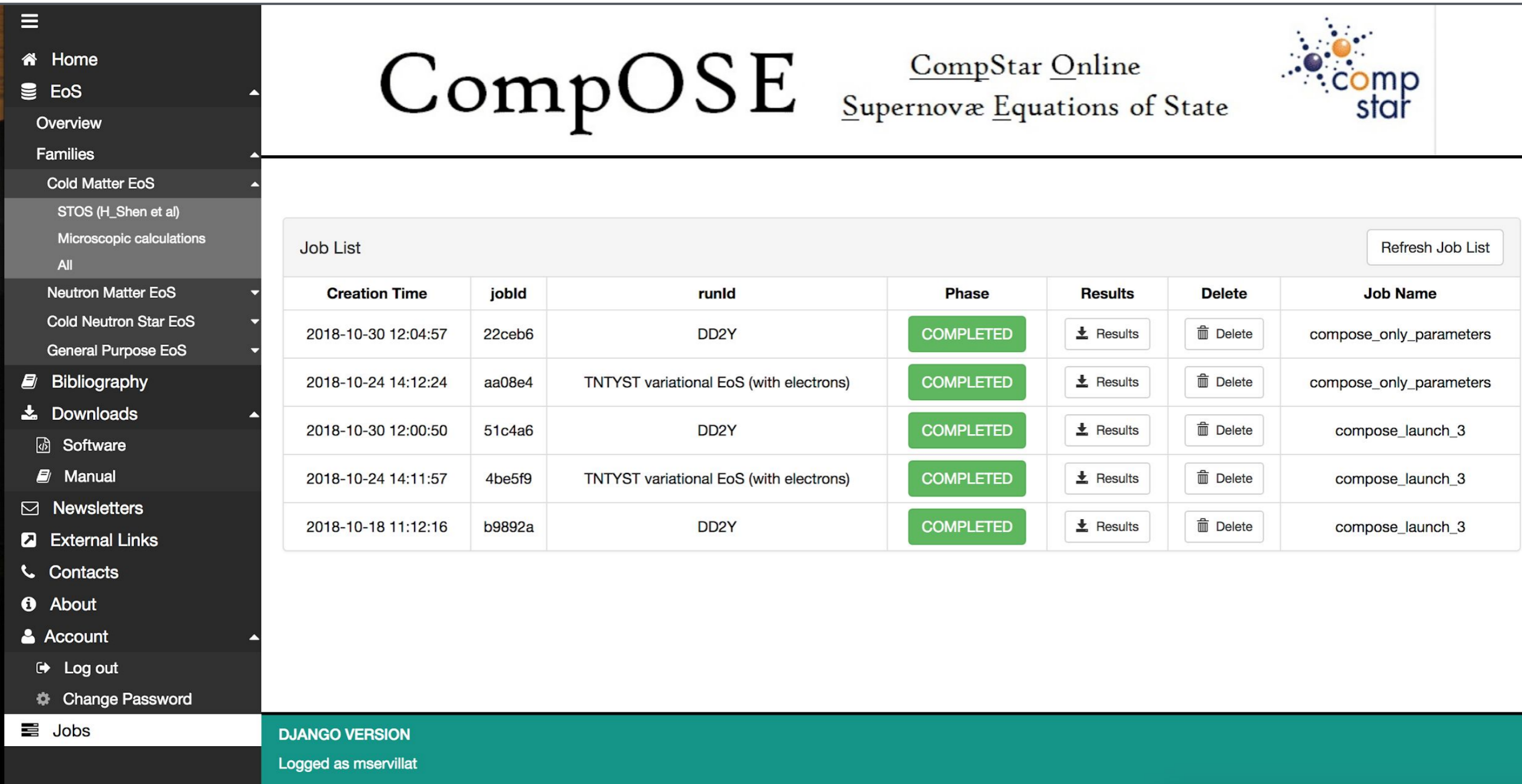
<https://voparis-uws-test.obspm.fr/provsap?ID=6188ab&DEPTH=ALL>



- Register all entities
  - input data
  - results→ id
- Check if already exist in the system
  - hash
  - + id or jobid

# OPUS inside CompOSE

- javascript client + Django pages on <https://compose.obspm.fr>
- OPUS server on a dedicated server (composecalc)



The screenshot displays the CompOSE web interface. At the top, the title "CompOSE" is prominently shown, followed by the subtitle "CompStar Online Supernovæ Equations of State" and the CompStar logo. A navigation sidebar on the left lists various categories like Home, EoS, Families, and Bibliography. The main content area features a "Job List" table with columns for Creation Time, jobId, runId, Phase, Results, Delete, and Job Name. All jobs listed are in the "COMPLETED" phase. A "Refresh Job List" button is located in the top right of the table area. At the bottom, a teal footer bar indicates the Django version and the user is logged in as mservillat.

Creation Time	jobId	runId	Phase	Results	Delete	Job Name
2018-10-30 12:04:57	22ceb6	DD2Y	COMPLETED	↓ Results	🗑️ Delete	compose_only_parameters
2018-10-24 14:12:24	aa08e4	TNTYST variational EoS (with electrons)	COMPLETED	↓ Results	🗑️ Delete	compose_only_parameters
2018-10-30 12:00:50	51c4a6	DD2Y	COMPLETED	↓ Results	🗑️ Delete	compose_launch_3
2018-10-24 14:11:57	4be5f9	TNTYST variational EoS (with electrons)	COMPLETED	↓ Results	🗑️ Delete	compose_launch_3
2018-10-18 11:12:16	b9892a	DD2Y	COMPLETED	↓ Results	🗑️ Delete	compose_launch_3