

# Observation Locator TAP - ObsLocTAP

*Jesús Salgado<sup>1</sup>, Aitor Ibarra<sup>1</sup>, Karl Foster<sup>5</sup>, Richard Saxton<sup>2</sup>,  
Jan-Uwe Ness<sup>4</sup>, Erik Kuulkers<sup>4</sup>, Carlos Gabriel<sup>4</sup>, Bruno Merin<sup>4</sup>,  
Peter Kretschmar<sup>4</sup>, Matthias Ehle<sup>4</sup>, Emilio Salazar<sup>3</sup>, Celia Sánchez<sup>3</sup>*

*1 Quasar for ESA*

*2 TPZ-VEGA for ESA*

*3 ATG for ESA*

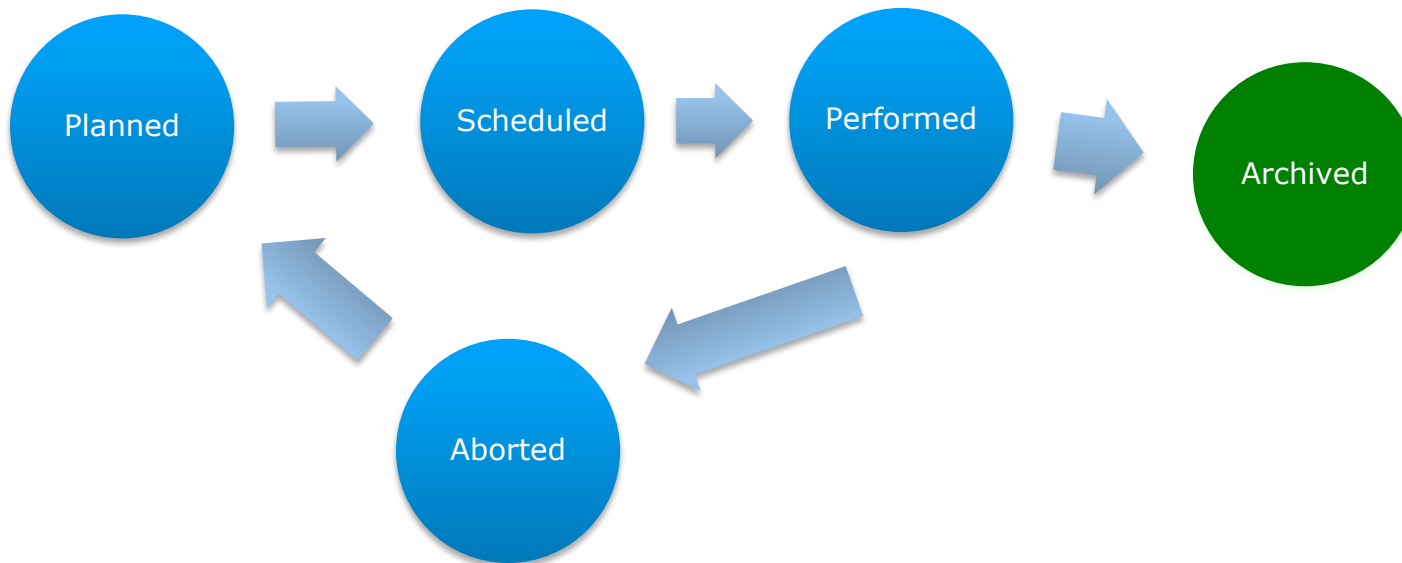
*4 ESA*

*5 NuSTAR*

A large graphic consisting of a light blue arrow pointing to the right, with a darker blue rounded rectangle centered inside it. The text 'Use Case' is written in white inside the rectangle.

# Use Case

# Observations Life cycle



# Planned Observations Services



## Integral Target and Scheduling Information

Schedule: All executed Current revolution (1872) Future schedule    Revolution 1872 to 1872    Show... show plot

### Schedule for revolution 1872

(this list is also available in csv-format, click [here](#) to download)

Rev	Start time (UTC)	E	Target	Science	Apertures	Spectral Elements	Exposure Time(sec)	AL	EX
1872	2017-10-10 13:29:15	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-10 17:13:34	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-11 08:16:46	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-11 13:27:21	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-11 15:00:12	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-11 18:41:00	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-12 09:06:18	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01
1872	2017-10-12 13:16:06	2	1872	Lockwood	35-002 WASP-69	MIRVIS	1300.00	12	01 01

ion	Notes
/0022	Public
/0011	
/0039	
/0038	
/0040	
/0040	
/0008	
/0041	
/0042	



### Short Term Schedule

#### XMM-NEWTON SHORT-TERM SCHEDULE

The Short-term Schedule gives an overview of scheduled observations covering the time range from the past week until the upcoming ~2-4 weeks.



SOC HOME OPERATIONS TEAM LOGIN SCIENCE TEAM LOGIN TOO TEAM LOGIN LINKS

### Observing schedules

Short Range Observability Schedule [Download](#)

This is the confirmed schedule of NuSTAR observations. This sequence of observations has been uploaded to the spacecraft and will execute autonomously unless interrupted by a new schedule, Target of Opportunity, or instrument and spacecraft anomalies. This schedule will cover various time ranges depending on the exposure time goal of the observations, but will usually be for a period of at least one week. The times reported here are the start and end of the on-target period (day of year UTC). The estimated exposure time takes into account Earth occultation and the SAA passage time where detector background is increased. The end time of the observation is the start of the slew to the next target. Please examine the NuSTAR As-Flown Timeline (AFT) for the log of past observations.

Table Header Explanations

obs_start	obs_end	sequenceID	Name	J2000_RA	J2000_Dec	Exp	Notes
2017:283:01:11:23	2017:283:02:40:00	90311211001	Sol_17282_AR2683_POS11	195.15715	-6.38520	3.4	ToO
2017:283:02:40:32	2017:283:04:20:00	90311212001	Sol_17282_AR2683_POS12	195.21879	-6.41062	3.4	ToO
2017:283:04:20:32	2017:283:05:50:00	90311213001	Sol_17282_AR2683_POS13	195.28046	-6.43604	3.4	ToO
2017:283:06:55:11	2017:284:09:20:00	60376001002	2MASXJ19301380p3410495	292.557500	34.180500	55.3	Extragalactic Legacy Survey
2017:284:09:45:09	2017:284:20:35:00	60360008002	SDSSJ152132d21p391206d9	230.3874232	39.2007671	22.0	Extragalactic Legacy Survey
2017:284:21:10:03	2017:285:21:00:00	90301320002	NGC_6440	267.218083	-20.538944	49.5	ToO
2017:285:21:20:06	2017:286:08:20:00	30302020004	GRS_1915p105	288.79813	10.94578	21.9	(2/4) coordinated with XMM and VLT
2017:286:08:35:06	2017:286:19:30:00	60160701002	2MASXJ18560128p1538059	284.00210000	15.63200000	23.3	BAT AGN
2017:286:20:05:11	2017:287:15:05:00	60376007002	UGC06728	176.316800	79.681500	61.4	Extragalactic Legacy Survey
2017:287:15:50:11	2017:288:03:20:00	60368001002	NGC_1144	43.80083	-0.18361	22.0	
2017:288:04:05:09	2017:288:23:00:00	60301004002	ESO_103m35	279.58458	-65.4275	50.3	
2017:288:23:30:08	2017:290:05:45:00	30301026002	AX_11841d0m0536	280.25179	-5.59625	59.7	phase constrained
2017:290:06:00:04	2017:290:17:00:00	60160670002	2E1739d1m1210	265.47600000	-12.19700000	23.5	BAT AGN
2017:290:17:15:01	2017:291:04:20:00	30363001002	GX_3p1	266.98333	-26.56361	21.8	

Long Range Observability Schedule [Download](#)

This is the latest NuSTAR long-term schedule. Observations have been sorted into one-week intervals, taking into account Sun, Moon, required exposure time, and other constraints. So the date is the Monday of the week in which the observation is scheduled to begin.

E.g. An observation with a date **2017-12-18** in this table is scheduled to have the observation **starting** sometime between **2017-12-18 0000Z and 2017-12-25 0000Z**.

Currently the schedule is driven by the large number of observations coordinated with other observatories and the need to complete the NuSTAR Guest Observer programs. The exposure goal for targets allotted within one week may appear to fill more than the available NuSTAR exposure time in that week (average is 330 ks per week) but many observations start in one week and complete in the following week.

Targets of opportunity and any instrument or spacecraft anomalies may also cause the observing times of targets to shift. This long-term schedule is our present estimate of the future order of observations. Please be aware of the uncertainties.

TOO = Target of Opportunity    DDT = Directors Discretionary Time    N03 = NuSTAR GO cycle-3    I15 = INTEGRAL GO cycle-15  
 X16 = XMM-Newton GO cycle-16    C18 = Chandra GO cycle-18    ELS/GLS = Extragalactic/Galactic legacy surveys

- Understand the role of all elements needed to implement an ObsLocTAP server
- Explain how to follow the implementation guide:  
[https://www.cosmos.esa.int/web/vovisobs\\_protocols/implementation-guides](https://www.cosmos.esa.int/web/vovisobs_protocols/implementation-guides)
- Identify difficult steps into the implementation
- Get feedback from the different partners to understand the status and possible support

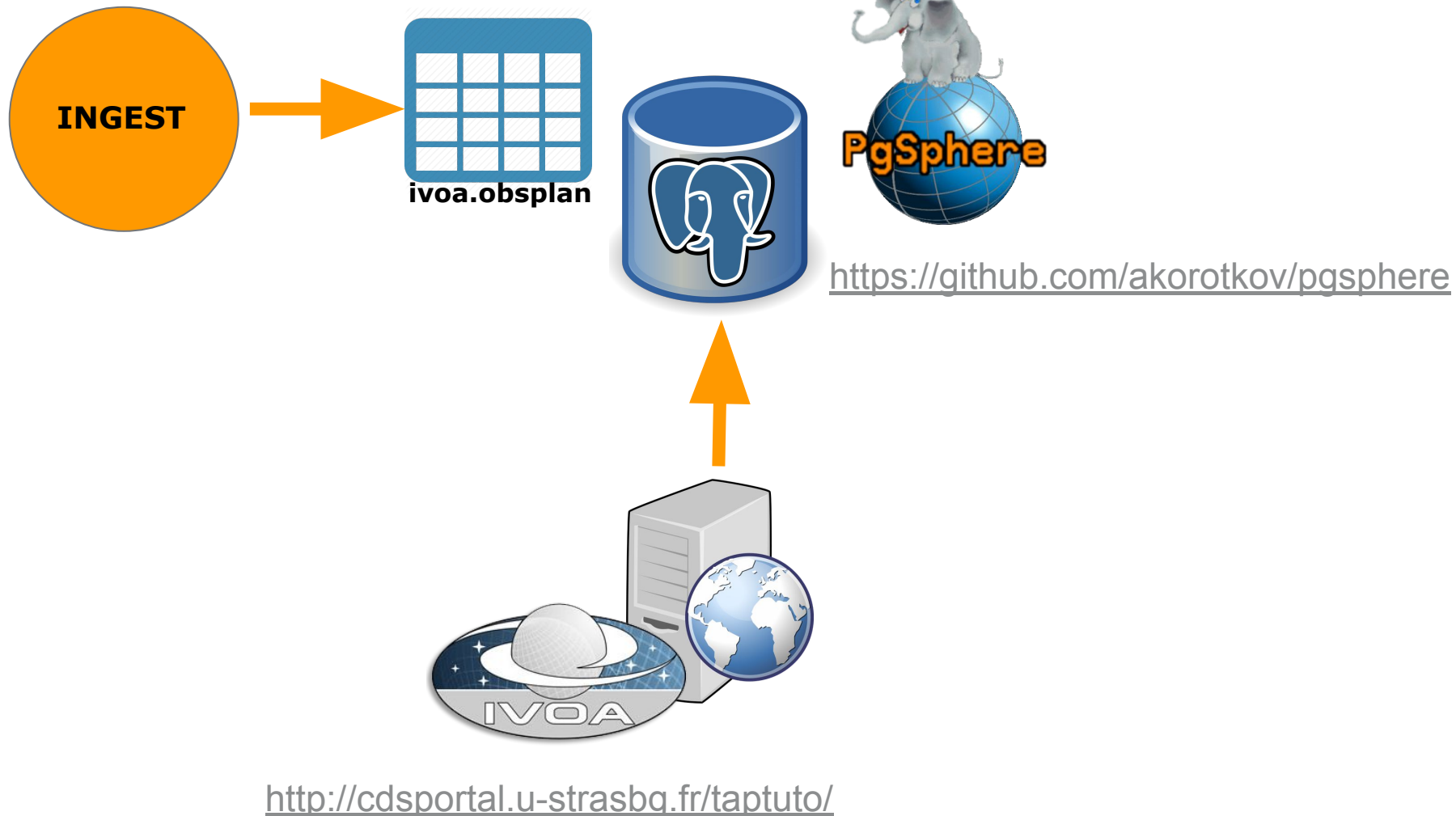


A diagram consisting of a large, light blue arrow pointing to the right. Inside the arrow are two rounded rectangular boxes, one blue and one white, representing 'Use Case' and 'Architecture' respectively.

Use Case

Architecture

# ObsLocTAP architecture



<http://cdsportal.u-strasbg.fr/taptuto/>

# Why TAP?

- **TAP (Tabular Access Protocol)** is a standard defined by the IVOA (International Virtual Observatory Alliance) to query astronomical databases
- It allows synchronous and asynchronous queries
- Used for many astronomical data providers (ESA, CDS, Heasarc, LSST...)
- A lot of client applications, including User Interfaces (e.g. TopCat) and command line interfaces (e.g. python astropy/pyvo)
- It is powerful and allows users (and applications) to create many complex exploration queries
- It allows the use of **ADQL**



- **ADQL** (Astronomical Data Query Language) is a SQL based query language designed for Astronomical Data Bases
  - No limitations on WHERE conditions
  - No restrictions on columns for SELECT
  - Spatial support based on coordinates
    - CONTAINS/INTERSECTS
    - CIRCLE, POLYGON, POINT
  - Enables temporary tables uploaded from users (persistent tables for TAP+)
  - Due to security restrictions, it does not support procedures and functions to modify alter database contents

## Gaia Archive

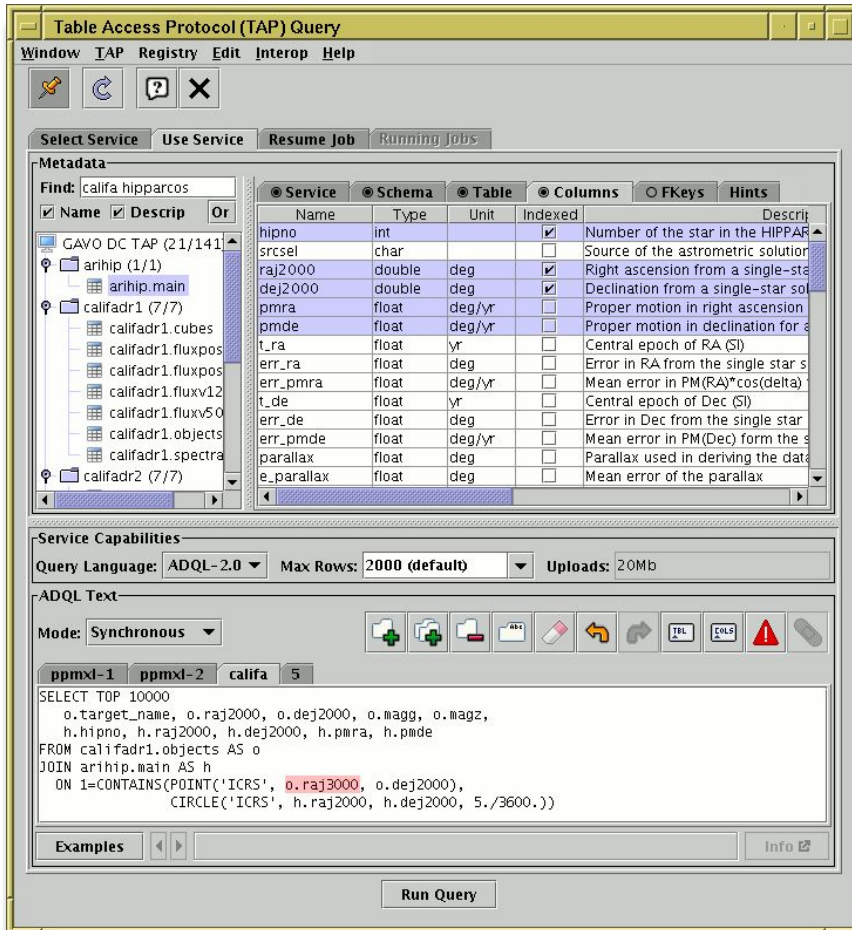
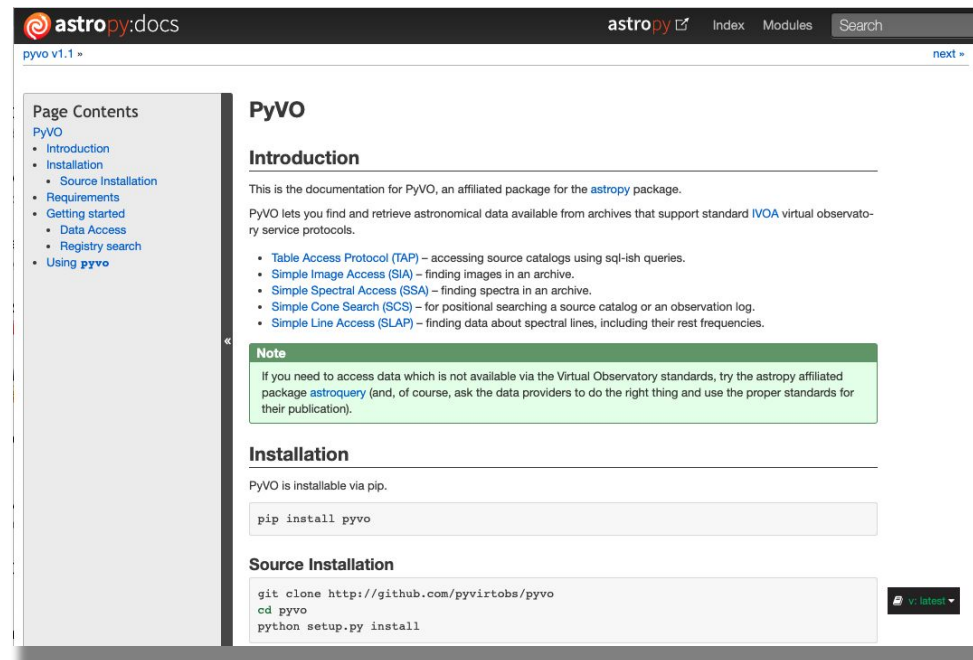
```
SELECT DISTANCE(  
    POINT('ICRS', ra, dec),  
    POINT('ICRS', 266.41683, -29.00781)) AS dist, *  
FROM gaiadr2.gaia_source  
WHERE 1=CONTAINS(  
    POINT('ICRS', ra, dec),  
    CIRCLE('ICRS', 266.41683, -29.00781, 0.08333333))  
ORDER BY dist ASC
```

## ObsLocTAP Query

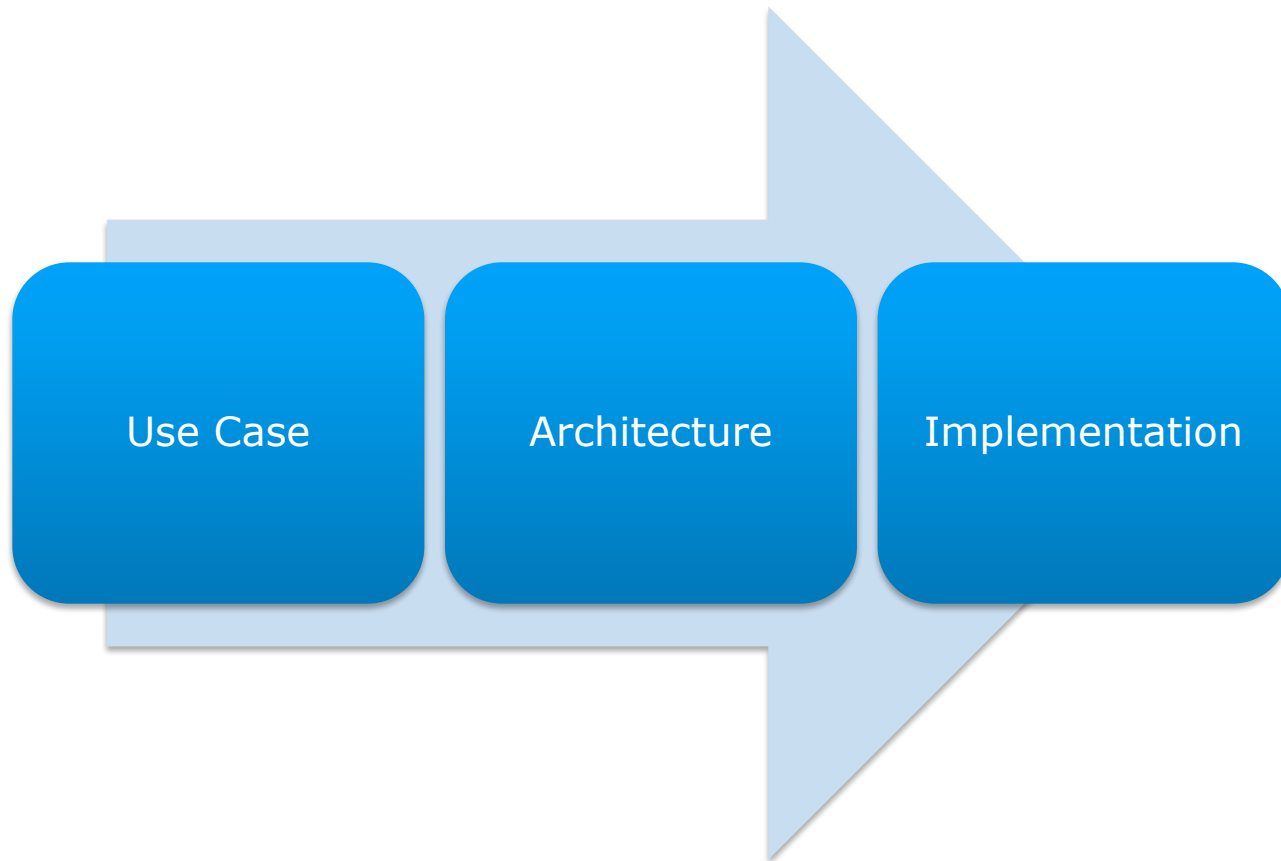
```
SELECT * FROM
ivoa.obsplan
WHERE
t_planning > 58500 AND t_max < 58502
AND 1=INTERSECTS(s_region, CIRCLE('ICRS', 114.8251,
1.6179, 0.016666 ))
```

where we are checking whether there are any newly planned observations from 58500 (17/01/2019) on the next two days (58502 or 19/01/2019) around the target PKS 0736+017, with a radius of 1 arcmin

<http://www.star.bris.ac.uk/~mbt/topcat/sun253/TabTableLoadDialog.html>

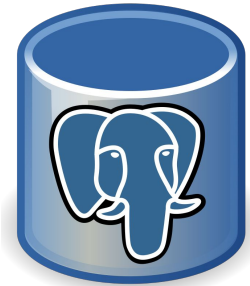
<https://pyvo.readthedocs.io/en/latest/>



- **Install a PostgreSQL DB:** Recommended 9.6

- Commonly used in Astronomy
- Geometrical indexing support
- IVOA Toolkits compatible with PostgreSQL

<https://www.postgresql.org/download/>



- **Install pgSphere:** PostgreSQL modules which adds spherical data types.

- containing, overlapping, and other operators
- point, circle, polygon
- indexing of spherical data types

<https://pgsphere.github.io/>



- There are some shortcuts for some operating systems like:

<https://zoomadmin.com/HowToInstall/UbuntuPackage/postgresql-pgsphere>

but it is usually better to install both packages sequentially



- IVOA offers different toolkits
  - DACHS from GAVO:
    - <http://soft.g-vo.org/dachs> <https://dachs-doc.readthedocs.io/tutorial.html>
  - SAADA:
    - <http://saada.unistra.fr/saada/>
  - TAPTuto:
    - <http://cdsportal.u-strasbg.fr/taptuto/>

<http://cdsportal.u-strasbg.fr/taptuto/>

- Very well documented
- Clear instructions
- Always updated to latest version (e.g. ADQL)
- Note: It makes use of a Tomcat web server (if not available, it needs to be installed too)

✔ Version **2.3** of the TAP library released since the 10<sup>th</sup> April 2019!  
 See [the modifications](#) for more details

# TAP Library v2.3

This Java library is a framework aiming to build quickly and easily a TAP service.

Getting started !

Download ↓

## What is TAP ?

TAP is a protocol defined by the IVOA in the [Recommendation of 27 March 2010 \(Version 1.0\)](#) which lets access table data.

The standards [ADQL](#) and [UWS](#) are used in this protocol.

Remind me! ⚡

## Why this library?

In order to help Java developers to quickly build a TAP service with as less code writing as possible. A service built using this framework will be conform to the IVOA definition of the protocol TAP.

### Functionalities:

- **Configuration file:** a TAP service can be created with just a simple key-value configuration file...nothing else.
- **Metadata management:** a TAP service let querying a database whose some tables and columns are accessible. All information (name, url, type, description, ...)

## How to use it ?

- **Getting started:** to start with this library.
- **Migration help:** to migrate from v1.0 to v2.0.
- **Documentation:** to have more details about all provided fonctionnalities.

⌚ The documentation is not yet complete. Content will be added piece by piece.

- **Javadoc:** Java documentation of all available classes.
- **What's new ?:** Last modifications of the library.

# ObsLocTAP implementation: TAP trick I



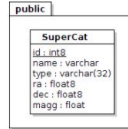
TAP Lib. ▾ Home TAP Reminder Getting Started Documentation ▾ Javadoc Download What's new?  
▪ (optional) Connection pool library

## Use case

Let's suppose we have a PostgreSQL database named `MyStarCatalogues` containing several astronomical data. Among all these data we want to publish in a TAP service only one table called `SuperCat` (in the database schema `public`).

Using this TAP library, we should first fulfil the above prerequisites:

- Web Application Server? [Apache Tomcat](#)
- Database? a PostgreSQL database called `MyStarCatalogues`.
- JDBC driver? [org.postgresql.Driver](#) (can be found [here](#))
- File directory? `file:///somewhere/TAPFiles`
- (optional) Connection pool library? the [Apache Tomcat](#) embedded connection pool.



## Goal

In both of the above methods, the goal will be to set a TAP service having the following specification:

- ✓ 4 output formats: VOTable (binary), VOTable (TableData), FITS and CSV.
- ✓ UPLOAD ability
- ✓ Simple coordinate system verification (only ICRS coordinates must be allowed)
- ✓ All ADQL geometries (depending of your DBMS and its extension(s))
- ✓ 4 common simple user defined functions (random, rtrim, rpad, initcap)

Configuration file

Create and configure an entire TAP service with only one single text file.

OR

HTTP Servlet

Extend 2 simple interfaces of the library and write an HTTP servlet to use them.

- To configure the TAP service the easiest way is to use the configuration file option
  - No code to be written
  - Change MyTAP servlet-name to your project name

Author: Grégory Mantelet (CDS)  
Last modification: 11 April 2019



```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee"
3. xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" version="3.0">
4.   <servlet>
5.     <display-name>tap</display-name>
6.     <servlet-name>MyTAP</servlet-name>
7.     <servlet-class>tap.config.ConfigurableTAPServlet</servlet-class>
8.   </servlet>
9.   <servlet-mapping>
10.    <servlet-name>MyTAP</servlet-name>
11.    <url-pattern>/tap/*</url-pattern>
12.  </servlet-mapping>
13. </web-app>
```

- **Connection to DB can be done by properties or JNDI**
  - Through properties is very simple. JNDI for better performance
  - Change jdbc\_url schema to your scheme (probably ivoa)

## Method 1: JDBC parameters

Just 1 parameter to select the JDBC method and 4 additional parameters to configure it are required here, nothing else:

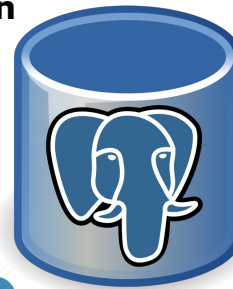
```
1. database_access = jdbc
2. jdbc_driver = org.postgresql.Driver
3. jdbc_url = jdbc:postgresql:MyStarCatalogues
4. db_username = tapuser
5. db_password = *****
```

- **DB schema** `TAP_SCHEMA`
  - TAP\_SCHEMA created at DB. Scripts provided by the instructions
- Use `sql_translator = pgsphere`
  - The translator (pgsphere instead of postgresql) is one of the few differences of the instructions

## INGEST



- obsplan table needs to be created into DB
- Table should be populated



- Defines with tables and columns are published into the TAP service
- Can be used for aliases
- Security layer. Not all the tables and columns are published



- Create a basic/plain TAP\_SCHEMA using TAPTUTO
  - **TAP\_SCHEMA.schemas:** One row per schema where TAP tables are located
  - **TAP\_SCHEMA.tables:** One row per table published on TAP
  - **TAP\_SCHEMA.columns:** One row per column published on TAP
- A new schema (ivoa) a new table (obsplan) and some columns need to be added to the ObsLocTAP service TAP\_SCHEMA
- These are described and an example of the SQL inserts could be found at the implementation guide:
  - [https://www.cosmos.esa.int/web/vovisobs\\_protocols/implementation-guides](https://www.cosmos.esa.int/web/vovisobs_protocols/implementation-guides)



- Next step is the obsplan DB creation
- A script can be found at:  
[https://www.cosmos.esa.int/web/vovisobs\\_protocols/implementation-guides](https://www.cosmos.esa.int/web/vovisobs_protocols/implementation-guides)
- CREATE EXTENSION pg\_sphere;
- Pgsphere will be needed to create the s\_region field. Two possible types:
  - Circle
  - Polygon
- Unfortunately, there is no way to express a mixture of circles and polygons in the same row
- Ingestion can be done talking directly with the database (e.g. using psql)

# INSERT example



```
INSERT INTO ivoa.obsplan
```

```
(t_planning,target_name,obs_id,obs_collection,s_ra,s_dec,s_fov,s_region,s_resolution,t_min,t_max,t_exptime,t_resolution,em_min,em_max,em_res_power,o_ucd,pol_states,pol_xel,facility_name,instrument_name,obs_release_date,t_plan_exptime,category,priority,execution_status)
```

```
VALUES (
```

```
58809.42866, 'Cas A / Tycho SNR', '21710053', '16200160002', 4.4687083, 59.9733583, 12.0,
```

```
<(4.4687083d , 59.9733583d), 12d> ,
```

```
720.0, 58833.67603, 58833.71689, 3530.0, 6.1E-5, 1.241528257871965E-13, 8.27685505247976
```

```
6E-11, 0.00386, 'photon.flux', NULL, 0, 'INTEGRAL', 'IBIS,
```

```
SPI', NULL, 1500000.0, 'Normal', 0, 'Scheduled');
```

- Please check pgsphere notation:  
<**(4.4687083d** , **59.9733583d)**, **12d**>  
equivalent to CIRCLE('ICRS', ra, dec, radius)
- Note: If “d” is not added, values in radians.
- In case of doubt understanding a field, please also check ObsCore  
<http://www.ivoa.net/documents/ObsCore/>

# Testing the system



- By using, e.g. TopCat, you can test your system by adding your service URL into the application. Generally, something like

[http://localhost:8080/<tap\\_name>/tap](http://localhost:8080/<tap_name>/tap)

### TAP HOME PAGE

---

**Available resources**

- [tables](#)
- [sync](#)
- [capabilities](#)
- [async](#)
- [availability](#)

---

**ADQL query**

Query:

```
SELECT *  
FROM TAP_SCHEMA.tables;
```

Execution mode:  Asynchronous/Batch  Synchronous

Format:

Result limit:  rows (0 to get only metadata ; a value < 0 means 'default value')

Duration limit:  seconds (a value ≤ 0 means 'default value')

---

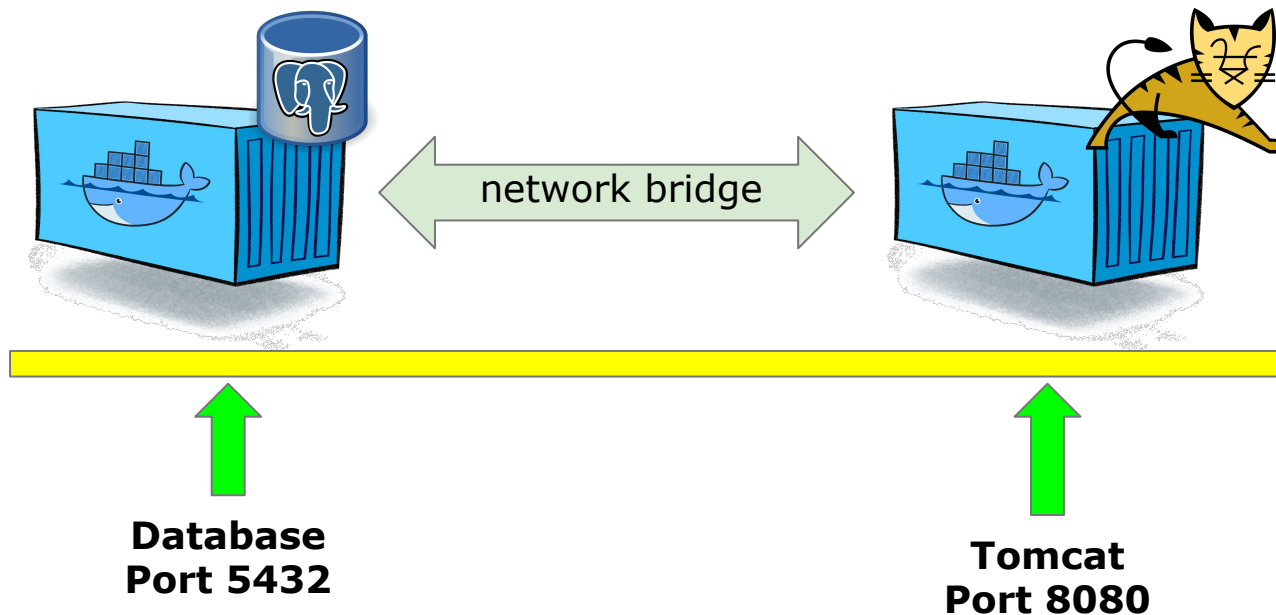
Page generated by [TAPLibrary](#) v2.0

Try some queries, like:

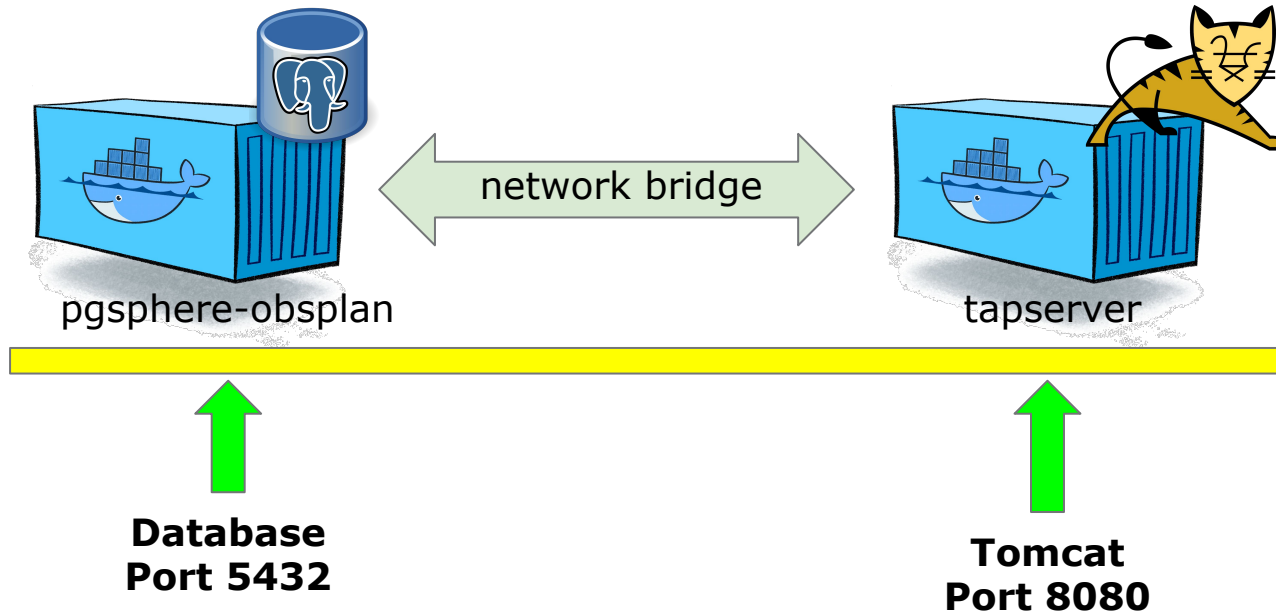
```
SELECT * FROM ivoa.obsplan  
WHERE  
t_max < 58502 AND  
1=INTERSECTS(s_region,  
CIRCLE('ICRS', 114.8251, 1.6179,  
0.016666))
```

# ObsLocTAP Docker short-cut

- Either to have a service to play with or to have a service without major effort, a couple of Docker instances are also provided:
  - PostgreSQL database with pgsphere installed, TAP\_SCHEMA and ivoa.obsplan table with some rows
  - Tomcat server with taptuto installed
- As Docker containers are executed on a sandbox (isolated from the host machine) a network bridge is created to communicate them



# ObsLocTAP by Docker start-up



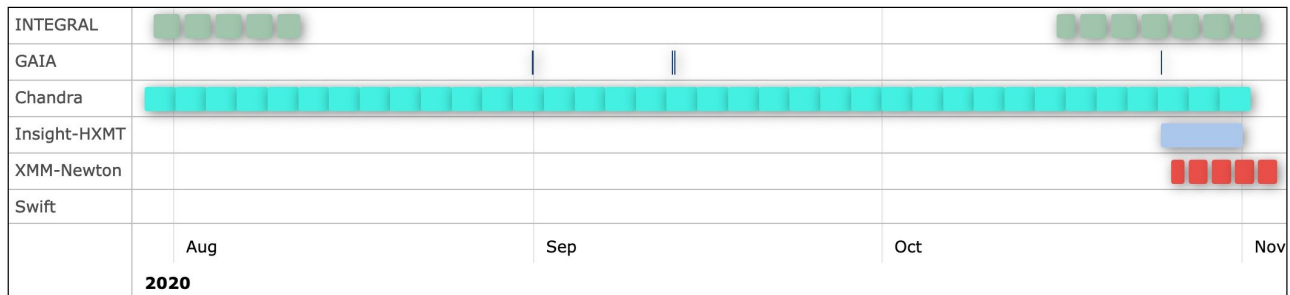
```
docker pull jsalgadodocker/pgsphere-obsplan:latest
docker pull jsalgadodocker/tapserver:latest
docker network create --driver=bridge db-network
docker image ls
docker run -p 8080:8080 --net=db-network --name tap jsalgadodocker/tapserver:latest
docker run -p 5432:5432 --net=db-network --name db jsalgadodocker/pgsphere-obsplan:latest
```

# Reference implementation client

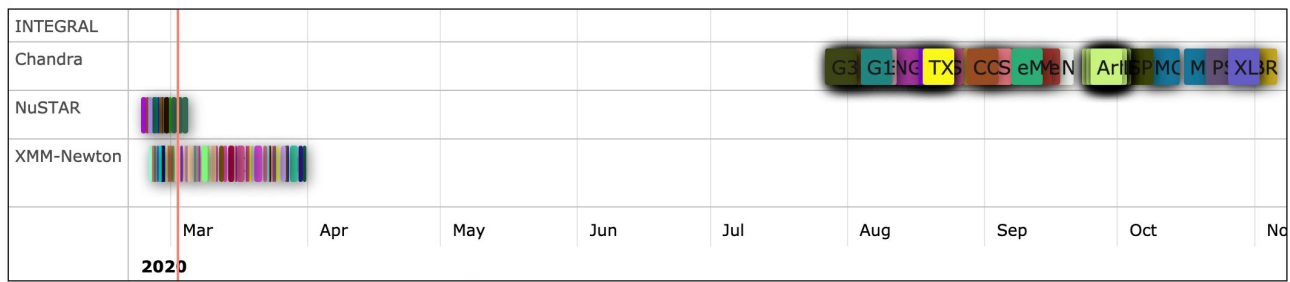


After successful implementation, contact E. Salazar for integration into:  
<http://integral.esa.int/toby/>

## Visibility



## Schedule



Other clients expected: MySpaceCal (calendar for many missions), ESASky, etc