



cadc-tap

Brian Major, Jade Dunn, Adrian Damian  
IVOA, November 2018, College Park



# CADC Python Tools

- vos (VOSpace client)
- cadc-data (data download service)
- get-cert (get a proxy certificate)
- caom2 tools
- and more...
- what, no python TAP client?
  
- CADC python commonalities:
  - Similar usage (debug/verbose mode, auth support, etc...)
  - All do VOSI capabilities lookup, good for:
    - finding right auth URL
    - finding new version of service

# Python TAP client

Goal: To give CADC users a python library and command line tool

Requirements:

- TAP 1.x compliant
- Support authentication (more formalized in TAP 1.1)
- Be able to point to *any* TAP service -- use the VO Registry
- Collaborative development: be an open-source project

Close: `astroquery/gaia` + `astroquery.utils.tap`

# cadc-tap: integration with astroquery

existing/active: `astroquery.utils.tap` - Tap+

in progress: `astroquery.vo` - registry lookup support

new: `astroquery.cadc` - library API

- forked from `astroquery.gaia`
- registry lookup: `astroquery.vo` + `cadsc` auth endpoint support

`opencadc/cadctools.cadc-tap` - command line wrapper to `astroquery.cadc`

How will the future unfold?

# goal: throw out astroquery.cadc!

Should be able to interface with a TAP library directly

If VO is implemented correctly in astropy / astroquery, then:

remove astroquery.cadc and use astroquery.utils.tap and/or astroquery.vo

- This should be possible for all compliant TAP 1.x based services in astroquery

Secondary goal: cadc-tap to tap

- Move command line wrapper to astroquery.vo?
- Need a common auth support mechanism

# Future for a general python TAP client?

- Many factors and interested parties
- PyVO?
- astroquery.vo has VO registry with authenticated auth points support
  - this could be used to find/connect with other VO services... should it be outside of astroquery?
- different types of authentication mechanisms (as defined in SSO 2.0) supported in astroquery.util.tap / astroquery.vo

# viva open-source development!

- Open-source development should be an important focus for IVOA
- **ready-to-use software** - don't make new projects write code for VO protocols
- How can IVOA define a strategy for open-source development?
  - A single code repository? Probably Not: already different repositories have strengths in different areas of IVOA standards.
  - Where to put the common code? Registry (service discovery), authentication, data models and representations...