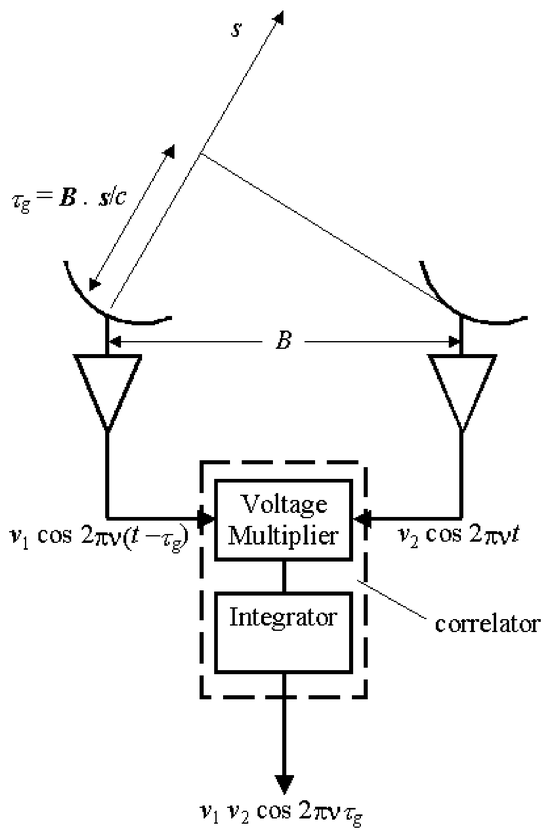
Implementation of external delay calculator to MeerKAT

Buntu Ngcebetsha, Software Engineer, South African Radio Astronomy Observatory
Engineering Division, Software Team (CAM)

Motivation & Goals

In this poster we briefly describe how we dealt with a migration of `katpoint` from using `ephem` to using `astropy`. The `katpoint` is a Python library used to calculate celestial coordinates relative to the MeerKAT telescope, and use the results to point at on-sky objects. The telescope is a distributed system of antennas, the time of arrival of the signal needs to be corrected as part of standard operation procedure during observations. It uses a `DelayUpdateManager`, which handles the correction of arrival times. The `katpoint` is a major dependency for the `DelayUpdateManager`, we describe how we handled the migration from `ephem` to `astropy` (which also involved Py2toPy3 migration). In this poster we explore the lessons learned when implementing part of a package to use `Python3` whilst the rest of our code-base was still in `Python2`. The technical benefit of this update was an improvement in the astrometry for delay calculations which will enhance the MeerKAT science and images.

Methods




```
{'delay_result':
    {'m008h': [1.968411243784553e-06, 0.0],
    'm008v': [1.968411243784553e-06, 0.0],
    'm001h': [1.0084078291945032e-07, 0.0],
    'm001v': [1.0084078291945032e-07, 0.0]},
'fringe_result':
    {'m008h': [-3025.2007294890336, 0.0],
     'm008v': [-3025.2007294890336, 0.0],
     'm001h': [12041.62927381297, 0.0],
     'm001v': [12041.62927381297, 0.0]}
}
```

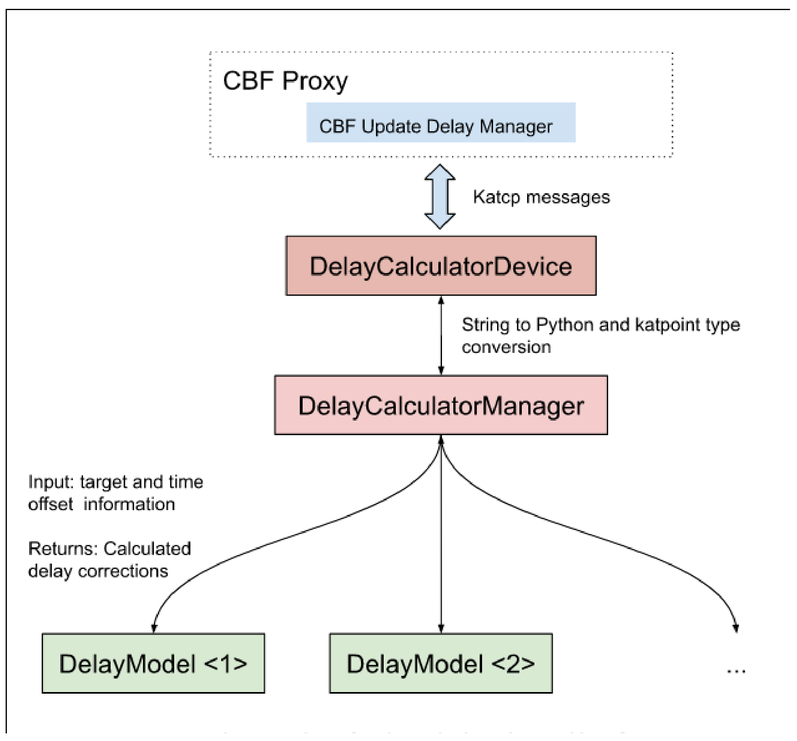
An example JSON string sent by the Delay Calculator back to CBF proxy. The keys are time delays and phase fringes for each antennas polarization h and v and the values are the delay adjustments in seconds and the fringes are in radians. the second element on the lists are the rates of each correction.

astropy

ephem

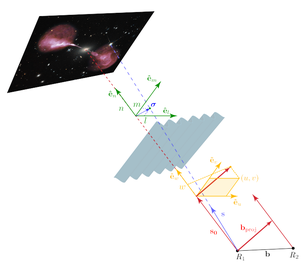
Selected Results

[2]The old `katpoint` imports and uses from `ephem` (see logo below) classes such as `StationaryBody`, `NullBody`. It also uses constants and conversions such as `lightspeedrad2deg` or `deg2rad`. This is the library being replaced by this improvement



The `DelayModel` is a `katpoint` object which does the actual calculation of the delay coefficients. In this modification, we replace `ephem` implementation with `astropy`. The goal is achieved by building a separate package from scratch, due to constrains of migration from version 2 to version 3 of Python in the desired time.

Explain here how astropy calculates the delays



The antennas sample the signal in an arbitrary plane known as the `uvw` plane, and this is the coordinate system in which the baselines have been positioned. There exists, theoretically - a Fourier relationship between the `uvw` plane and the actual image plane, see figure 2. The output of the correlator is a table of complex numbers known as visibility data. The data represents snapshots at every feasible `uvw` coordinate at the location of every baseline for the duration of the observation. In order to make the data scientifically useful, astronomers make an image of the sky through the use of deconvolution algorithms built into imaging pipelines. This process produces better results with improved and accurate measurement of sky positions which also relies on accurate timing of the arrival of the signal from vast distances from outer space. [3].