DESDM Refactoring Work at CD-REX

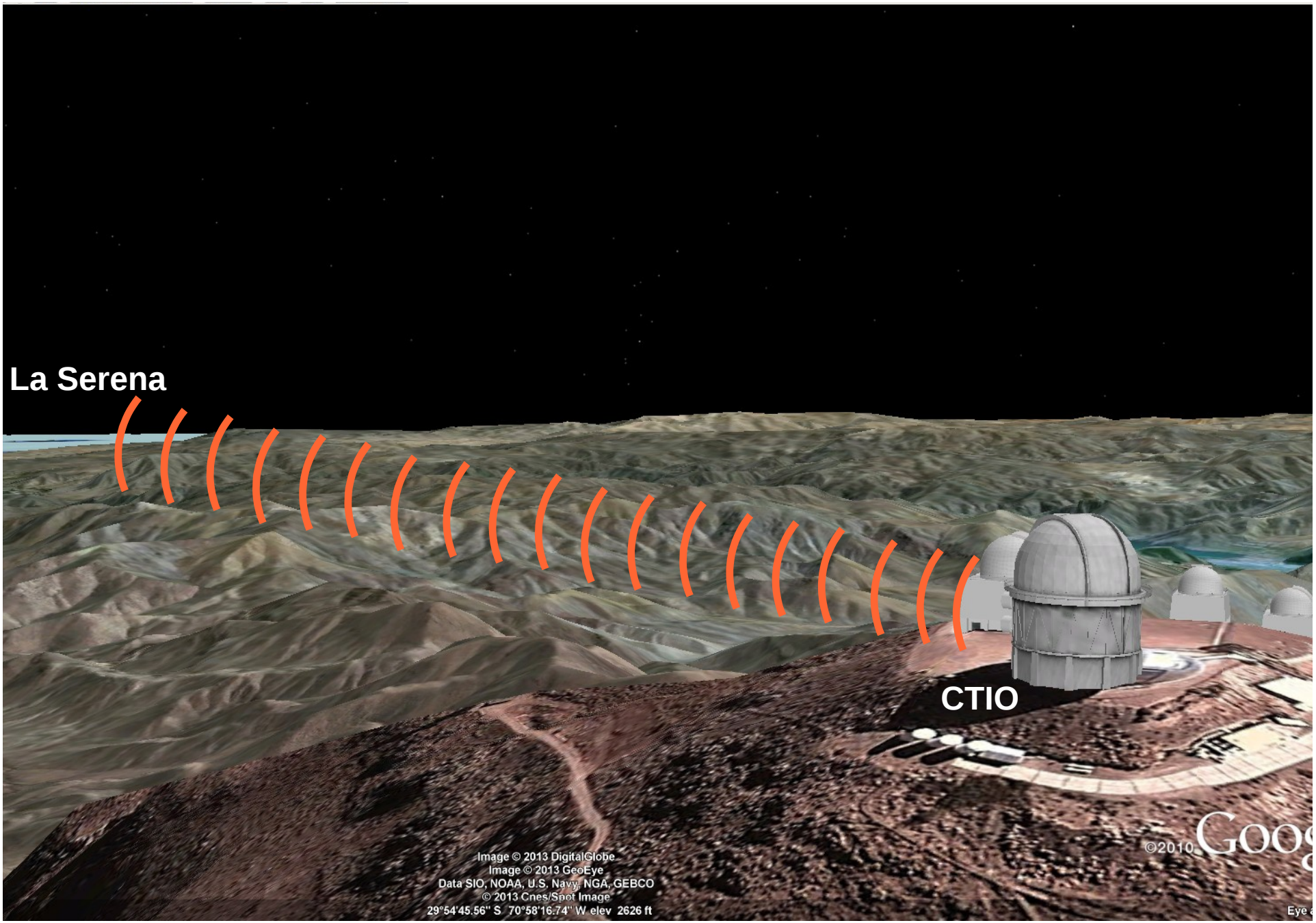
Marc Mengel and Michael Wang for the CD-REX Department

CD Scientific Computing Division Projects Meeting

20 February 2013

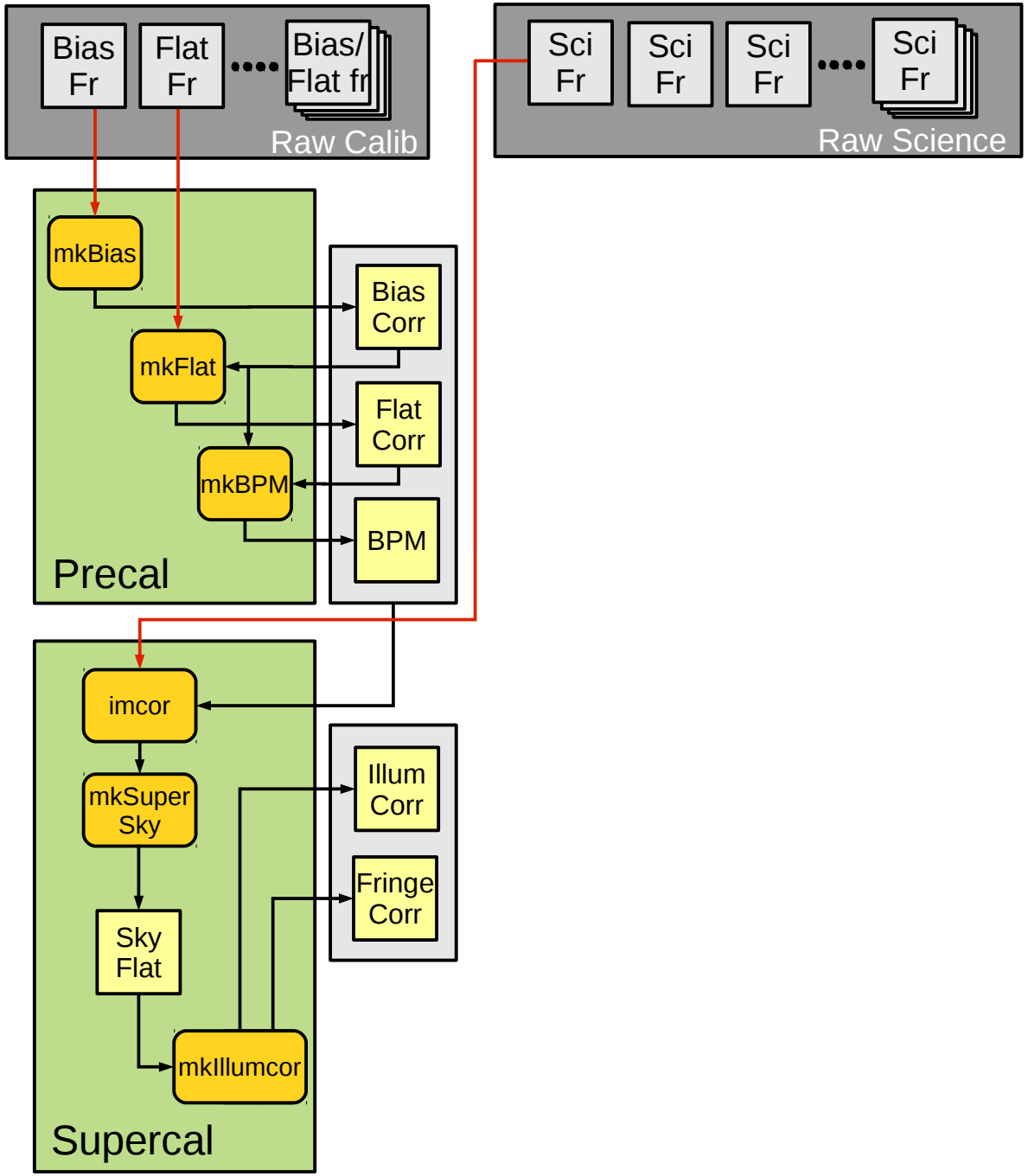
DECAM on the Blanco

Microwave link

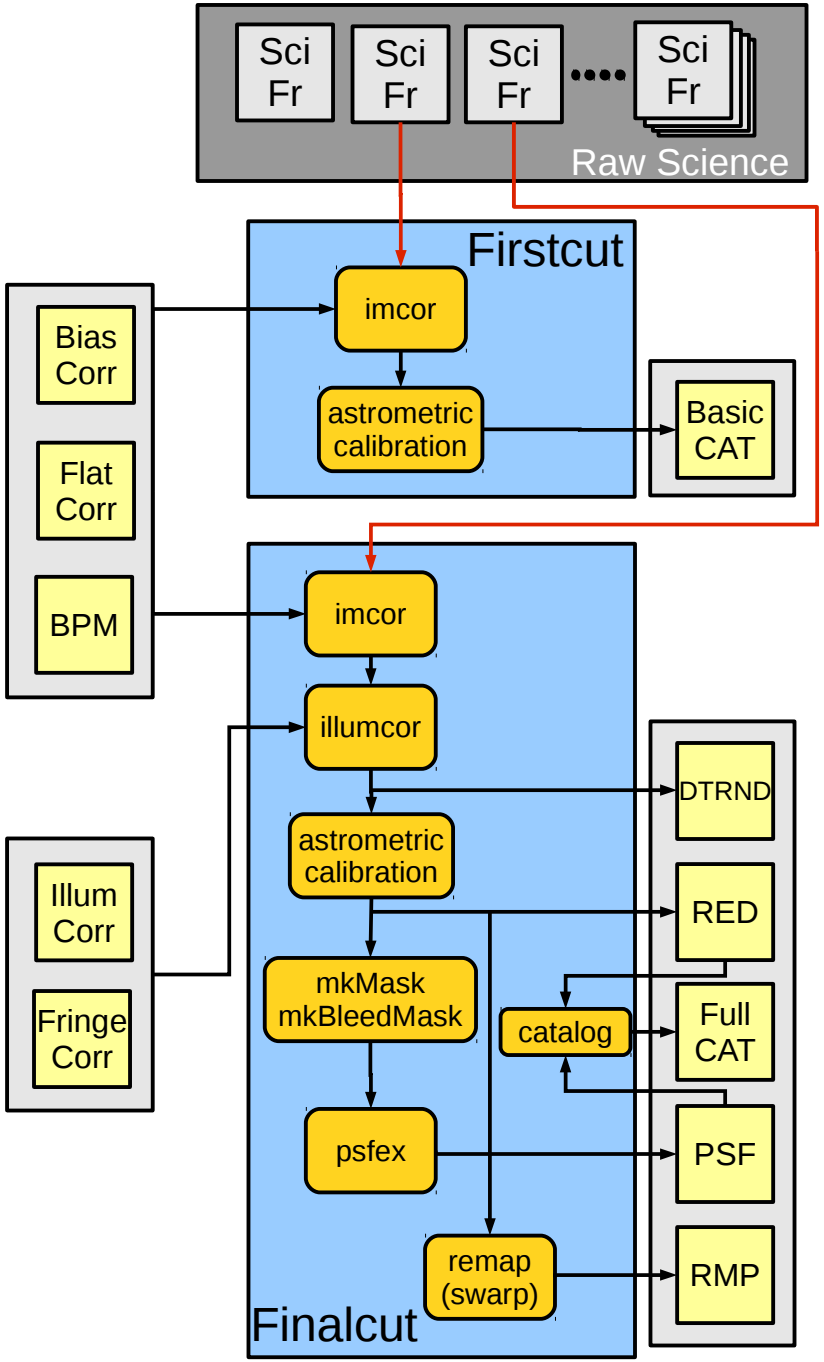


NCSA

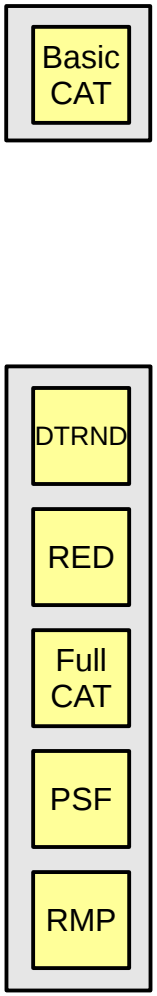
Calibration pipelines



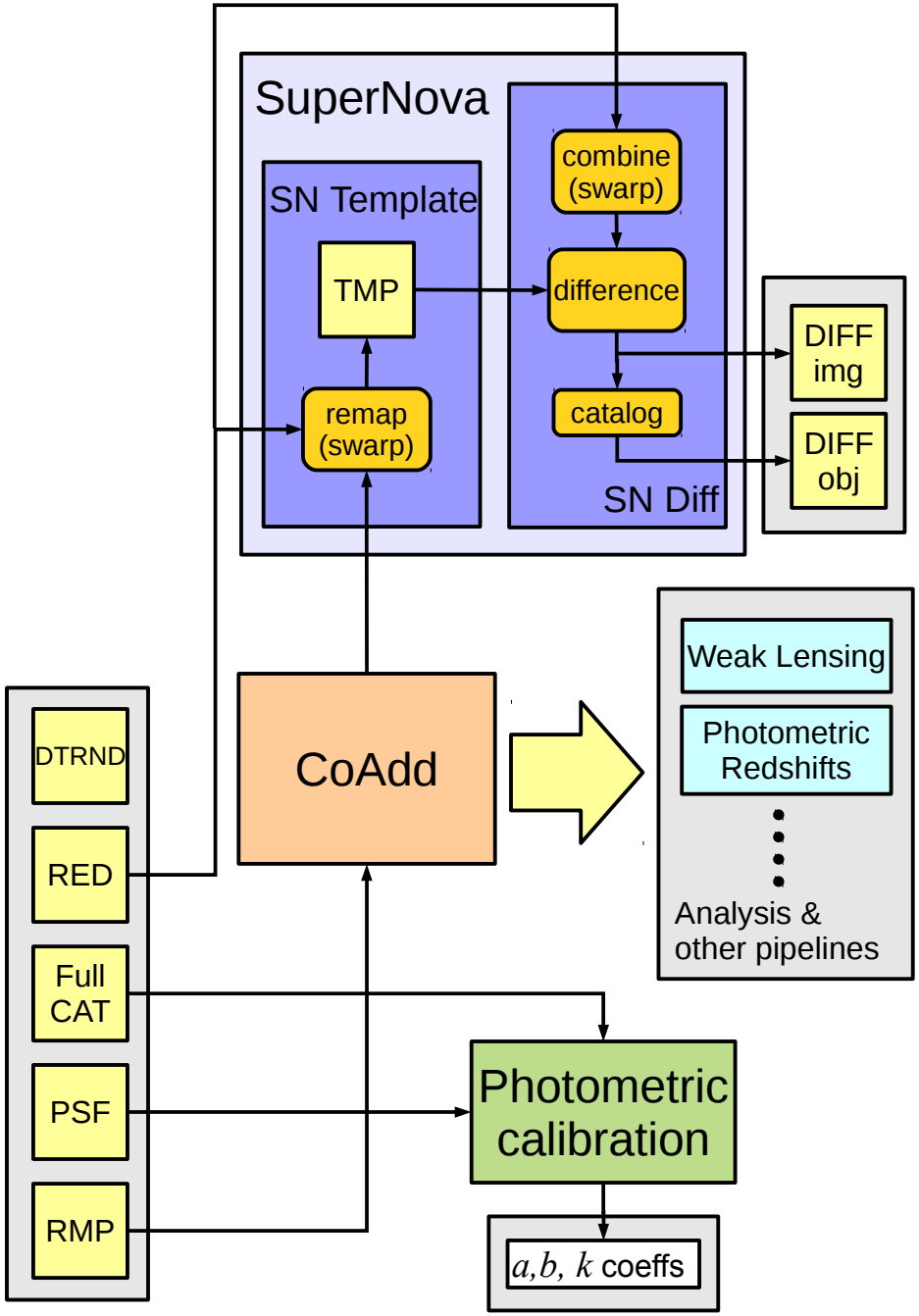
Detrending and Reduction pipelines



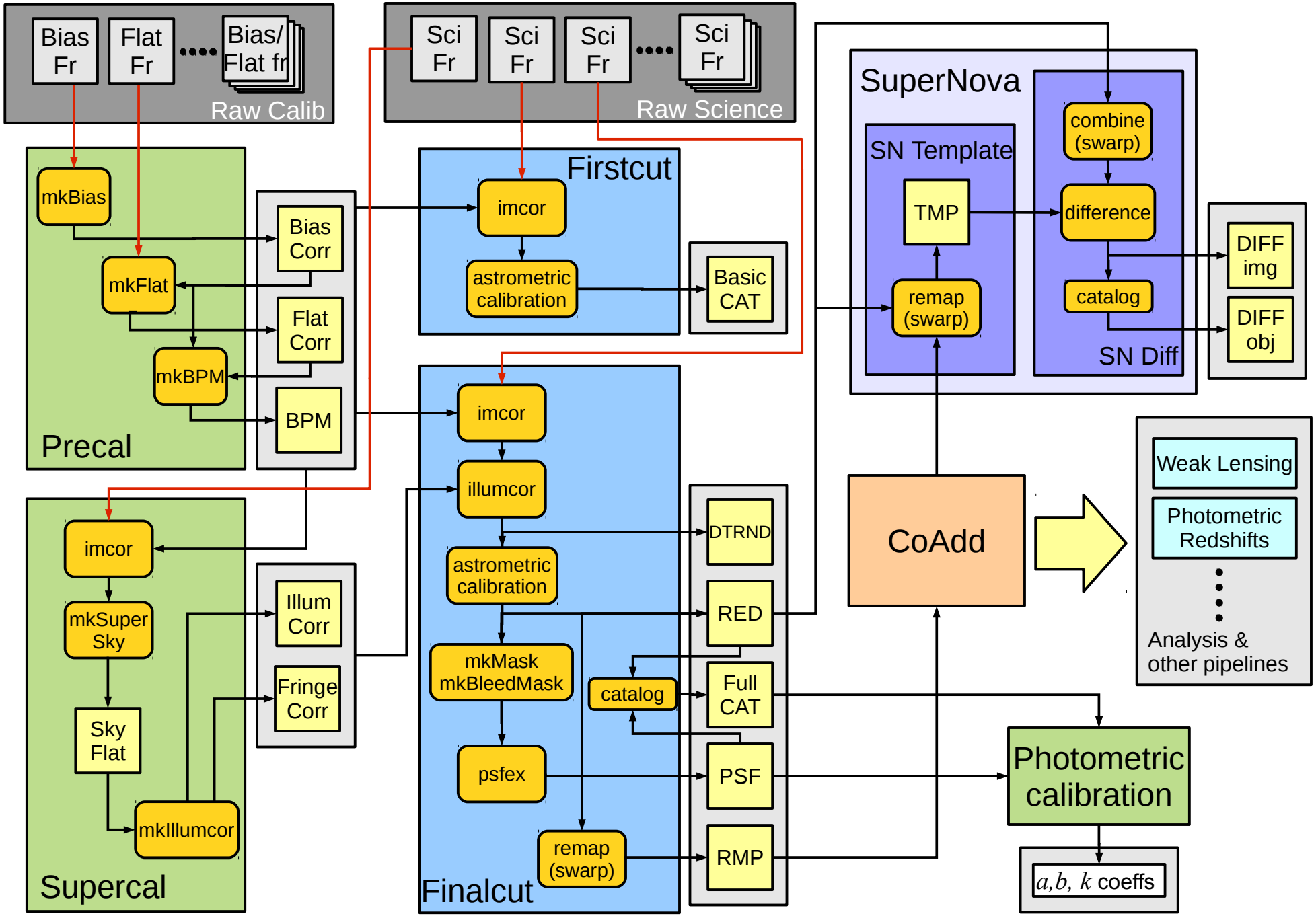
Detrending and Reduction pipelines



Coadd, PSM, SN & other pipelines



DES processing pipelines



Wrapper framework

- The “simplified” diagram of DES data processing on the previous slide shows you that it is quite complex and is made up of several pipelines each of which involve many steps.
- Each orange rectangle represents an executable or more and there are many in DES – DECam_crosstalk, mkbiascorr, mkflatcor, mkbpm, msupersky, mkillumcor, sex, scamp, psfex, swarp, hotpants, etc.
- The next slide gives you an example of how some of these are typically executed on the comand line ➡

Running science code on the command line

▶ *Running **Source Extractor** in the Astrometric Calibration step*

```
sex '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[0]'
  -c '../../DES_HOME/etc/sexforscamp.config'
  -CATALOG_NAME '03_astrf/data/S01000105_z_01_r01p01_scampcat.fits'
  -CATALOG_TYPE 'FITS_LDAC' -WEIGHT_TYPE 'MAP_WEIGHT'
  -WEIGHT_IMAGE '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[2]'
  -FLAG_IMAGE '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[1]'
  -PARAMETERS_NAME '../../DES_HOME/etc/sex.param_scamp'
  -FILTER_NAME '../../DES_HOME/etc/sex.conv'
  -STARNNW_NAME '../../DES_HOME/etc/sex.nnw'
  -WEIGHT_THRESH '0.0000' -SATUR_LEVEL '47452.68' -DETECT_THRESH '2.5000'
```

▶ *Running **SCAMP** in the Astrometric Calibration step*

```
scamp '03_astrf/data/S01000105_z_r01p01_scampcat.fits'
  '../../DES_HOME/etc/decam.cat' -MOSAIC_TYPE 'SAME_CRVAL' -AHEADER_GLOBAL 'decam_nopv.ahead'
  -ASTRINSTRU_KEY 'DUMMY' -AHEADER_SUFFIX '.aheadnoexist' -ASTREFMAG_LIMITS '-99,99'
  -ASTREFCAT_NAME '../../cal/20110906155821_20110810_astrostds.fits,../../DES_HOME/etc/decam.cat'
  -ASTREF_CATALOG 'FILE' -c '../../DES_HOME/etc/default.scamp' -WRITE_XML 'Y'
  -XML_NAME '03_astrf/data/S01000105_z_r01p01_scamp.xml' -MATCH_RESOL '0.0' -POSITION_MAXERR '1.0'
```

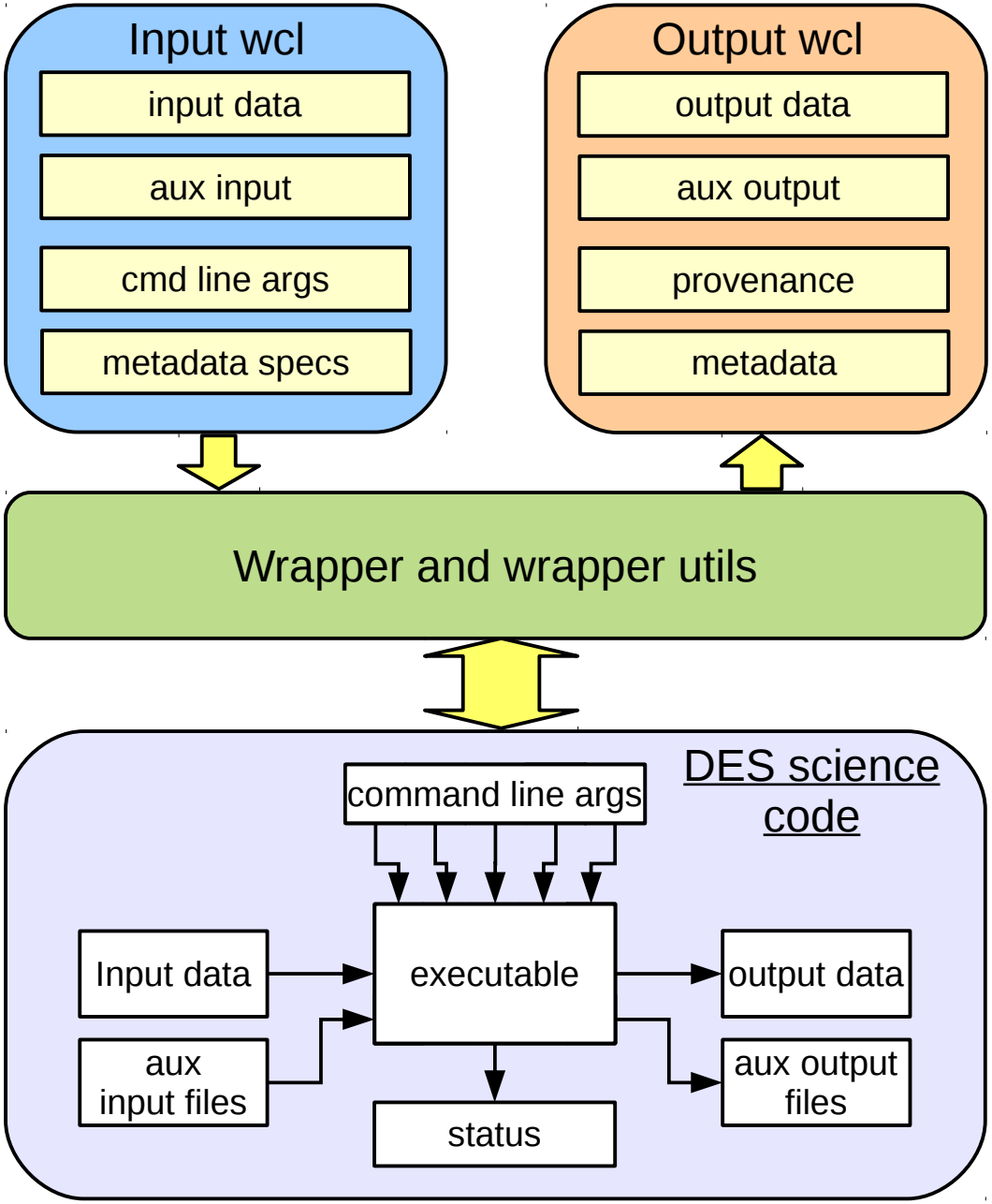
▶ *Running **SWARP** in the remapping step*

```
swarp -NTHREADS '4' -c '../DES_HOME/etc/default.swarp'
  '../data/red_wcs/D00136000_r_06_r01p01_wcs.fits[0]' -PIXELSCALE_TYPE 'MANUAL' -PIXEL_SCALE '0.263'
  -CENTER_TYPE 'MANUAL' -CENTER '17.094410,0.730556' -IMAGE_SIZE '0' -RESAMPLING_TYPE 'LANCZOS3'
  -RESAMPLE 'Y' -RESAMPLE_DIR 'cp_remap/01_remap/data' -SUBTRACT_BACK 'Y' -BLANK_BADPIXELS 'N'
  -DELETE_TMPFILES 'Y' -FSCALASTRO_TYPE 'VARIABLE' -COMBINE 'N' -COMBINE_TYPE 'WEIGHTED'
  -WEIGHTOUT_NAME 'D00136000_r_06_r01p01_wcs_tmp_DES0108+0043_im.weight.fits'
  -RESAMPLE_SUFFIX '_DES0108+0043_im.fits' -WEIGHT_TYPE 'MAP_WEIGHT'
  -WEIGHT_IMAGE '../data/red_wcs/D00136000_r_06_r01p01_wcs.fits[2]'
  -HEADER_ONLY 'N' -VERBOSE_TYPE 'FULL' -WRITE_XML 'N'
  -COPY_KEYWORDS 'OBJECT,OBSERVER,CCDNUM,FILTER,OBSTYPE,FILENAME,TELRA,TELDEC,TELQUIN,AIRMASS,
                  DATEOBS,EXPTIME,SATURATE,DETECTOR,TELESCOP,OBSERVAT,SKYBRITE,SKYSIGMA,PHOTFLAG,
                  FWHM,ELLIPTIC,SCAMPCHI,SCAMPFLG,SCAMPNUM'
```

Wrapper framework

- As you see, these execs take numerous parameters, config files, etc. specified through command line arguments, and there is no uniformity in the interface.
- In the past, these pieces were tied together to form a pipeline through monolithic and inflexible scripts.
- The refactoring effort at REX aims to address this issue by providing a uniform interface to the underlying DES executables through an intermediate “wrapper” layer encapsulating the orange rectangles in the previous diagram:
  - This layer consists of python wrapper code and xml-style input configuration and output wcl (wrapper control language) files.
  - This modular approach should make it much easier to modify existing pipelines or configure new ones

Wrapper interface to underlying executable



Typical wrapper steps

1. Parse input args to wrapper
2. Read input wcl
3. Expand $ variables
4. Check input files
5. Construct cmdline
6. Run executable
7. Check status & output files
8. Generate output wcl with prov & metadata

Running science code on the command line

▶ *Running **Source Extractor** in the Astrometric Calibration step*

```
sex '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[0]'
  -c '../../DES_HOME/etc/sexforscamp.config'
  -CATALOG_NAME '03_astrf/data/S01000105_z_01_r01p01_scampcat.fits'
  -CATALOG_TYPE 'FITS_LDAC' -WEIGHT_TYPE 'MAP_WEIGHT'
  -WEIGHT_IMAGE '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[2]'
  -FLAG_IMAGE '02_imcor/data/S01000105_z_01_r01p01_detrend.fits[1]'
  -PARAMETERS_NAME '../../DES_HOME/etc/sex.param_scamp'
  -FILTER_NAME '../../DES_HOME/etc/sex.conv'
  -STARNNW_NAME '../../DES_HOME/etc/sex.nnw'
  -WEIGHT_THRESH '0.0000' -SATUR_LEVEL '47452.68' -DETECT_THRESH '2.5000'
```

▶ *Running **SCAMP** in the Astrometric Calibration step*

```
scamp '03_astrf/data/S01000105_z_r01p01_scampcat.fits'
  '../../DES_HOME/etc/decam.cat' -MOSAIC_TYPE 'SAME_CRVAL' -AHEADER_GLOBAL 'decam_nopv.ahead'
  -ASTRINSTRU_KEY 'DUMMY' -AHEADER_SUFFIX '.aheadnoexist' -ASTREFMAG_LIMITS '-99,99'
  -ASTREFCAT_NAME '../../cal/20110906155821_20110810_astrostds.fits,../../DES_HOME/etc/decam.cat'
  -ASTREF_CATALOG 'FILE' -c '../../DES_HOME/etc/default.scamp' -WRITE_XML 'Y'
  -XML_NAME '03_astrf/data/S01000105_z_r01p01_scamp.xml' -MATCH_RESOL '0.0' -POSITION_MAXERR '1.0'
```

▶ *Running **SWARP** in the remapping step*

```
swarp -NTHREADS '4' -c '../DES_HOME/etc/default.swarp'
  '../data/red_wcs/D00136000_r_06_r01p01_wcs.fits[0]' -PIXELSCALE_TYPE 'MANUAL' -PIXEL_SCALE '0.263'
  -CENTER_TYPE 'MANUAL' -CENTER '17.094410,0.730556' -IMAGE_SIZE '0' -RESAMPLING_TYPE 'LANCZOS3'
  -RESAMPLE 'Y' -RESAMPLE_DIR 'cp_remap/01_remap/data' -SUBTRACT_BACK 'Y' -BLANK_BADPIXELS 'N'
  -DELETE_TMPFILES 'Y' -FSCALASTRO_TYPE 'VARIABLE' -COMBINE 'N' -COMBINE_TYPE 'WEIGHTED'
  -WEIGHTOUT_NAME 'D00136000_r_06_r01p01_wcs_tmp_DES0108+0043_im.weight.fits'
  -RESAMPLE_SUFFIX '_DES0108+0043_im.fits' -WEIGHT_TYPE 'MAP_WEIGHT'
  -WEIGHT_IMAGE '../data/red_wcs/D00136000_r_06_r01p01_wcs.fits[2]'
  -HEADER_ONLY 'N' -VERBOSE_TYPE 'FULL' -WRITE_XML 'N'
  -COPY_KEYWORDS 'OBJECT,OBSERVER,CCDNUM,FILTER,OBSTYPE,FILENAME,TELRA,TELDEC,TELQUIN,AIRMASS,
                  DATEOBS,EXPTIME,SATURATE,DETECTOR,TELESCOP,OBSERVAT,SKYBRITE,SKYSIGMA,PHOTFLAG,
                  FWHM,ELLIPTIC,SCAMPCHI,SCAMPFLG,SCAMPNUM'
```

Running science code in the refactored wrapper framework

▶ *Running **Source Extractor** in the Astrometric Calibration step*

```
generic.py --input createScampCat_input.wcl >& createscampcats_01.log
```

▶ *Running **SCAMP** in the Astrometric Calibration step*

```
runscamp.py --input runScamp_input.wcl >& runscamp.log
```

▶ *Running **SWARP** in the remapping step*

```
genericSwarp.py --input runSwarpImg_input.wcl >& runswarpimg.log
```

Input/Output wcl

- A typical input wcl has the following sections:
  - Filespecs
    - Specifies input/output files and other auxiliary files
    - Metadata required for each file is also specified here
  - Exec
    - Executable name
    - Command line arguments
    - Relationship between files, needed for provenance
  - Wrapper
    - Pipeline name and version
    - Wrapper specific parameters

- A typical output wcl has the following sections:
  - Exec
    - execname
    - Actual command line used to run exec
    - Execution time
  - Wrapper
    - Pipeline name and version
    - Exit status
    - Total wrapper execution time
  - Provenance information
  - Metadata
    - Metadata associated with each file

Sample filespecs section of input wcl (astrometry)

```
<filespecs>
    <scamp_config>
        fullname = ../../DES_HOME/etc/default.scamp
    </scamp_config>
    <scamp_astrefcat>
        fullname = ../../cal/20110906155821_20110810_astrostds.fits
    </scamp_astrefcat>
    <scamp_decamcat>
        fullname = ../../DES_HOME/etc/decam.cat
    </scamp_decamcat>
    <scamp_cat>
        fullname = $LSTCOL{${list.scamp_cat.listname},-1,scamp_cat.fullname}
    </scamp_cat>
    <scamp_fullcat>
        fullname = 03_astrf/data/S01000105_z_r01p01_scampcat.fits
    </scamp_fullcat>
    <scamp_xml>
        fullname = 03_astrf/data/S01000105_z_r01p01_scamp.xml
    </scamp_xml>
    <fullhead>
        filetype = se_misc
        fullname = 03_astrf/data/S01000105_z_r01p01_scampcat.head
        <req_metadata>
            wcl = filespecs.fullhead.fullname,filespecs.fullhead.filename,filespecs.fullhead.filetype
        </req_metadata>
    </fullhead>
    <head>
        filetype = red_head
        fullname = $LSTCOL{${list.head.listname},-1,head.fullname}
        <req_metadata>
            wcl = filespecs.head.fullname,filespecs.head.filename,filespecs.head.filetype
        </req_metadata>
    </head>
    <scampqa>
        filetype = qa_scamp
        fullname = $RNMLST{${wrapper.qa_file_prefix},${wrapper.qa_file_list}}
        <req_metadata>
            wcl = filespecs.scampqa.fullname,filespecs.scampqa.filename,filespecs.scampqa.filetype
        </req_metadata>
    <scampqa>
</filespecs>
```

Sample exec sections of input wcl (astrometry)

```
<exec_1>
    execname = mergeScampCats()
    used = filespecs.scamp_cat
    was_generated_by = filespecs.scamp_fullcat
    was_derived_from = filespecs.scamp_cat:filespecs.scamp_fullcat
    <cmdline>
        _01_incats = filespecs.scamp_cat.fullname
        _02_oucat  = filespecs.scamp_fullcat.fullname
    </cmdline>
</exec_1>
<exec_2>
    execname = scamp
    used = filespecs.scamp_config,filespecs.scamp_fullcat,filespecs.scamp_astrefcat,filespecs.scamp_decamcat
    was_generated_by = filespecs.fullhead,filespecs.scamp_xml,filespecs.scampqa
    was_derived_from = filespecs.scamp_fullcat:filespecs.fullhead
    <cmdline>
        _01_scampcat = ${filespecs.scamp_fullcat.fullname}
        _02_decamcat = ${filespecs.scamp_decamcat.fullname}
        MOSAIC_TYPE = SAME_CRVAL
        AHEADER_GLOBAL = decam_nopv.ahead
        ASTRINSTRU_KEY = DUMMY
        AHEADER_SUFFIX = .aheadnoexist
        ASTREFMAG_LIMITS = -99,99
        ASTREFCAT_NAME = ${filespecs.scamp_astrefcat.fullname},${filespecs.scamp_decamcat.fullname}
        ASTREF_CATALOG = FILE
        c = ../../DES_HOME/etc/default.scamp
        WRITE_XML = Y
        XML_NAME = ${filespecs.scamp_xml.fullname}
    </cmdline>
</exec_2>
<exec_3>
    execname = splitScampHead()
    used = filespecs.fullhead
    was_generated_by = filespecs.head
    was_derived_from = filespecs.fullhead:filespecs.head
    <cmdline>
        _01_inhead  = filespecs.fullhead.fullname
        _02_ouheads = filespecs.head.fullname
    </cmdline>
</exec_3>
```

Sample provenance & metadata section of output wcl (SExtractor)

```
<provenance>
    <used>
        exec_1 = sexforscamp.config,sex.param_scamp,sex.conv,sex.nnw,
                 S01000105_z_01_r01p01_detrend.fits
    </used>
    <was_generated_by>
        exec_1 = S01000105_z_01_r01p01_scampcat.fits
    </was_generated_by>
    <was_derived_from>
        <derived_1>
            parents = S01000105_z_01_r01p01_detrend.fits
            children = S01000105_z_01_r01p01_scampcat.fits
        </derived_1>
    </was_derived_from>
</provenance>
```

Sample metadata section of output wcl (header update after SCAMP)

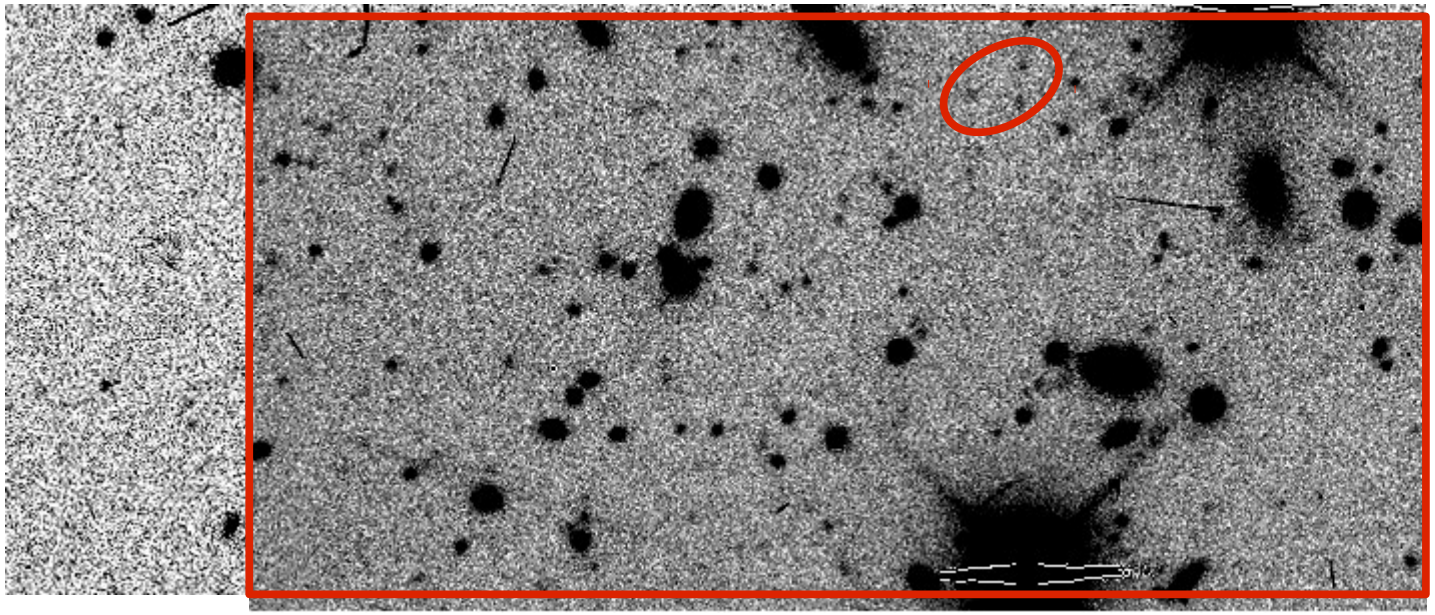
```
<file_metadata>
    <file_1>
        fullname = 03_astrf/data/S01000105_z_01_r01p01_wcs.fits
        band = z
        airmass = 1.26567
        bunit = adu
        ccdnum = 1
        cd1_1 = -2.25791009692e-08
        cd1_2 = 7.50061384973e-05
        cd2_1 = -7.50047201826e-05
        cd2_2 = 1.83571463869e-08
        crpix1 = 13423.200439
        crpix2 = 6307.333293
        crval1 = 339.999969307
        crval2 = 3.79996119941e-05
        ctype1 = RA---TPV
        ctype2 = DEC--TPV
        cunit1 = deg
        cunit2 = deg
        datasec = [1:2048,1:4096]
        detsec = [2049:4096,8193:12288]
        elliptic = 0.0397
        equinox = 2000.0
        exptime = 50.0
        Fwhm = 3.7266
             .
             .
             .
```

“Standalone” framework

- The modular nature of the wrapper+wcl framework makes it straightforward to configure and build elaborate processing pipelines based on “wrapper” building blocks
- We have also developed a “standalone” (no access to DB) framework that provides the “controls” and “interconnects” between the basic wrapper building blocks, serving as a “job control” layer for configuring and sequencing the tasks associated with a pipeline.
- This framework is not meant to replace the orchestration work done by the NCSA production group. With its limited capabilities, the standalone framework is meant for testing and science verfication purposes.
- In this framework, pipelines are described in a wcl-based configuration file which is fed to a “run.py” python script that expands variables and nested loops, and makes substitutions in the configuration file before running each step in the pipeline in the right sequence.
- The framework is used by DES scientists in testing pipelines on DECam data.

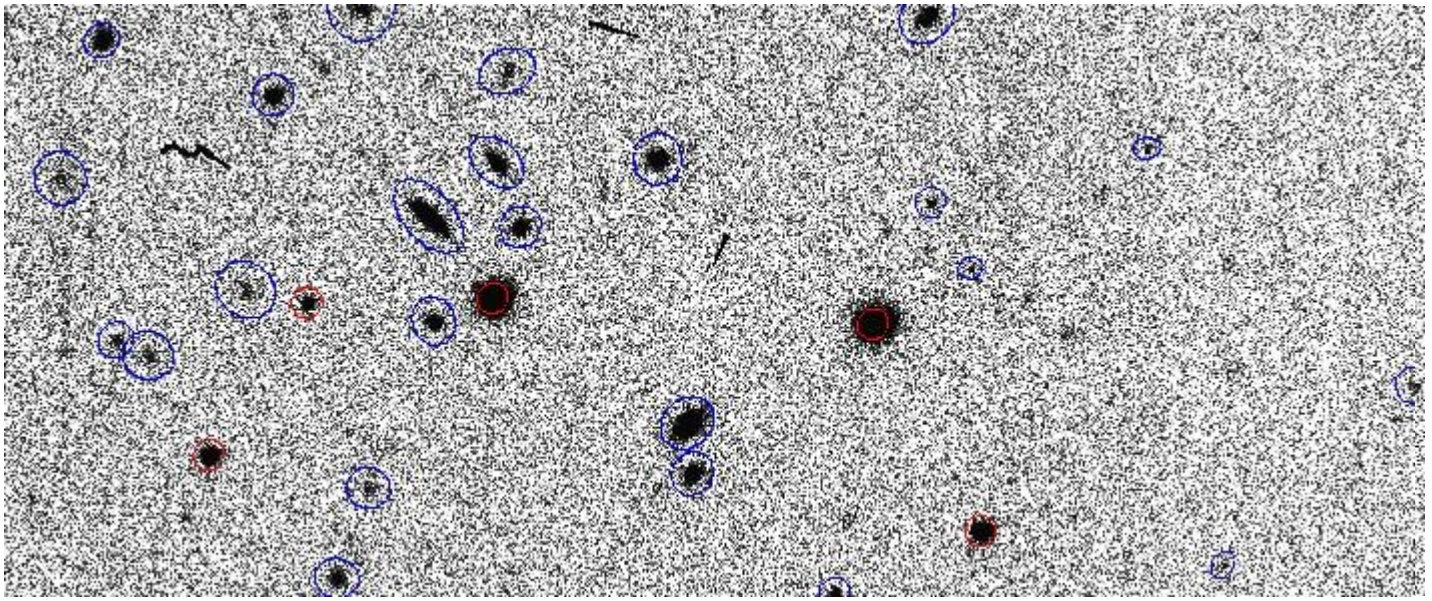
Sample coadd result using standalone pipeline

from Brian Yanny

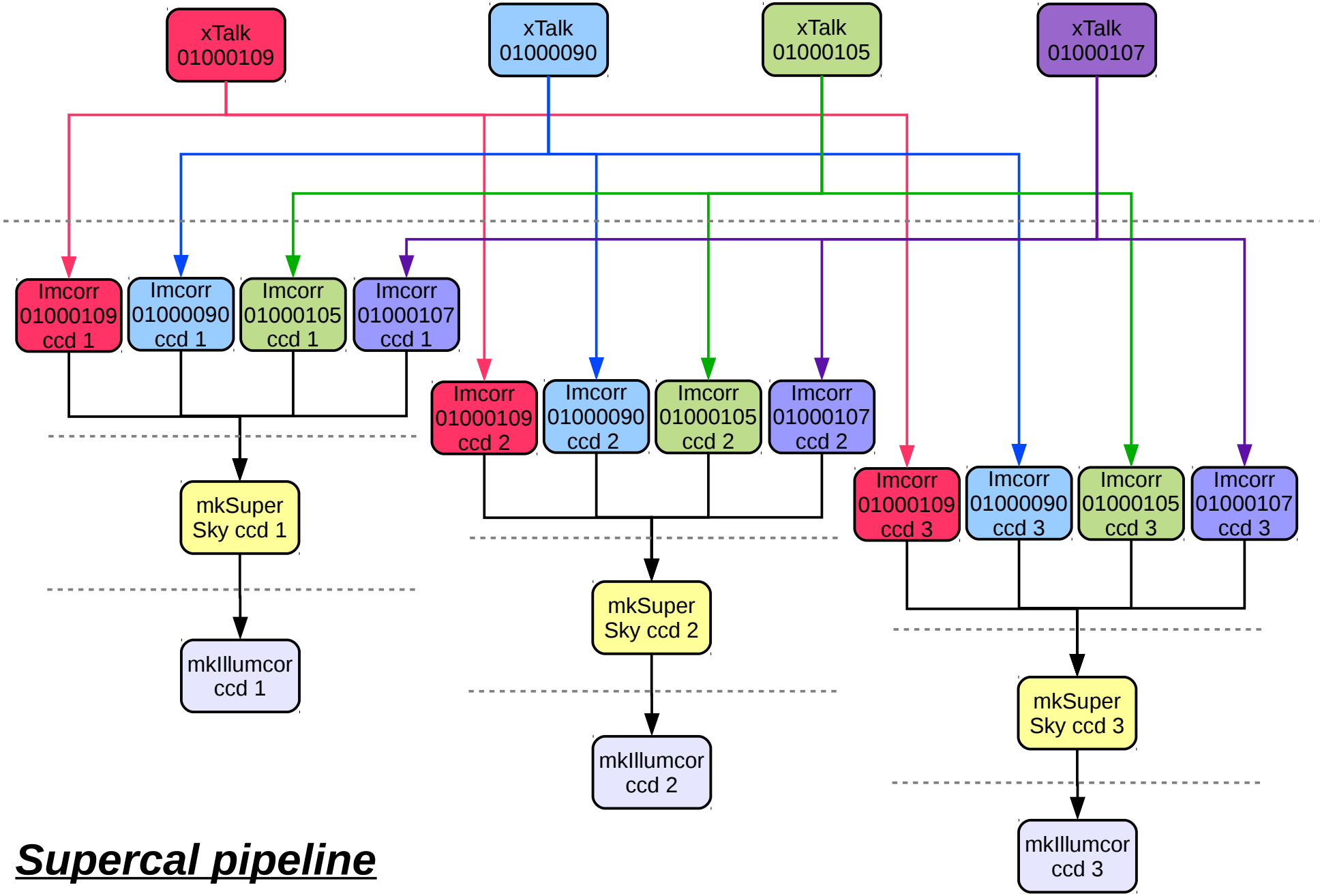


Sample astrometry result using standalone framework

from Brian Yanny



Condor DAG generated by standalone framework



Supercal pipeline

Conclusion

- The DES data processing pipelines are quite complex and it can seem like a daunting task to implement them
- But the wrapper+wcl framework REX has been working on greatly simplifies their implementation and maintenance by providing a modular and uniform interface to the underlying executables that makes orchestration straightforward
- We have also provided a set of tools (standalone framework) that allows the DES scientist to easily configure a pipeline for testing on a laptop or even on an HPC farm or the grid
- The framework and tools developed are quite generic and should easily find application in future astrophysics experiments and perhaps even in running HEP detector-simulation+reconstruction+analysis chains.

End