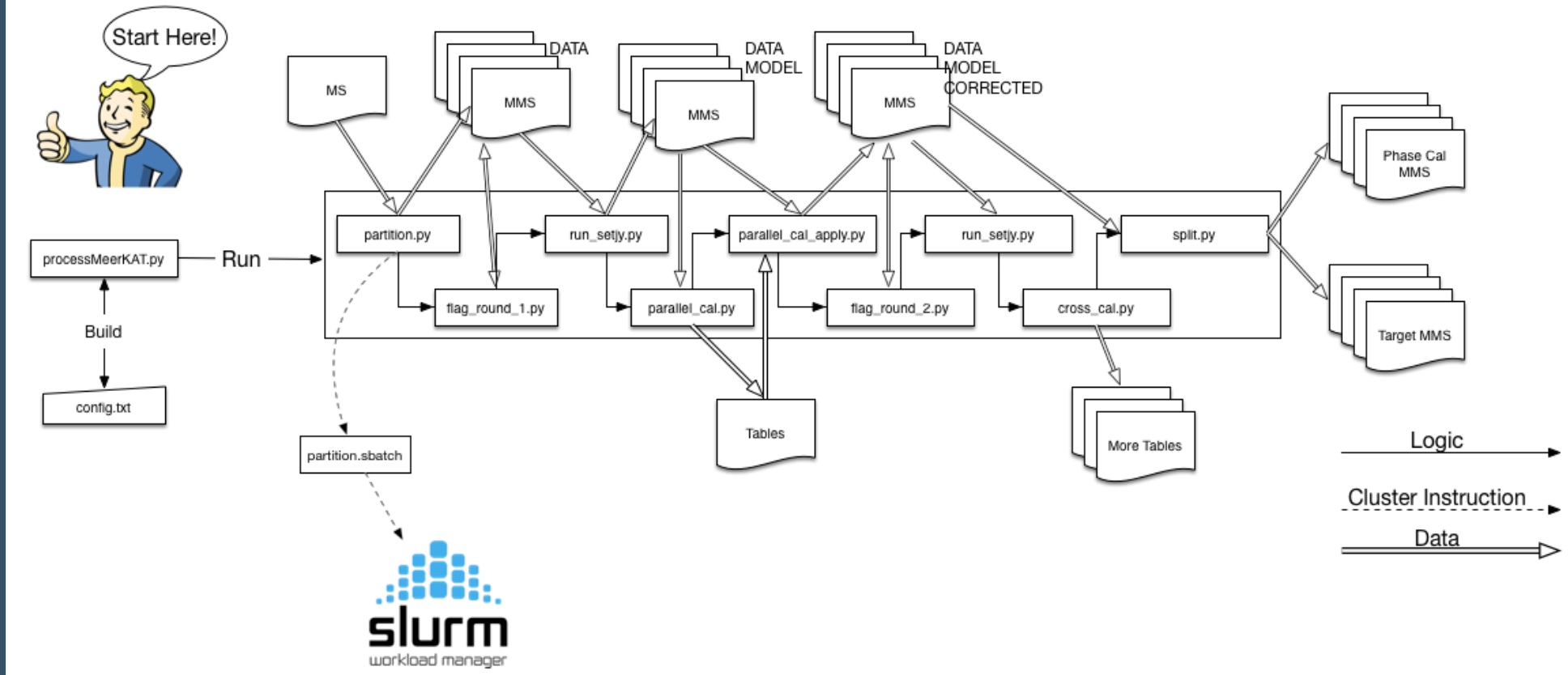
Calibration in the IDIA pipeline

Srikrishna Sekhar

Jordan Collier
Brad Frank
Russ Taylor

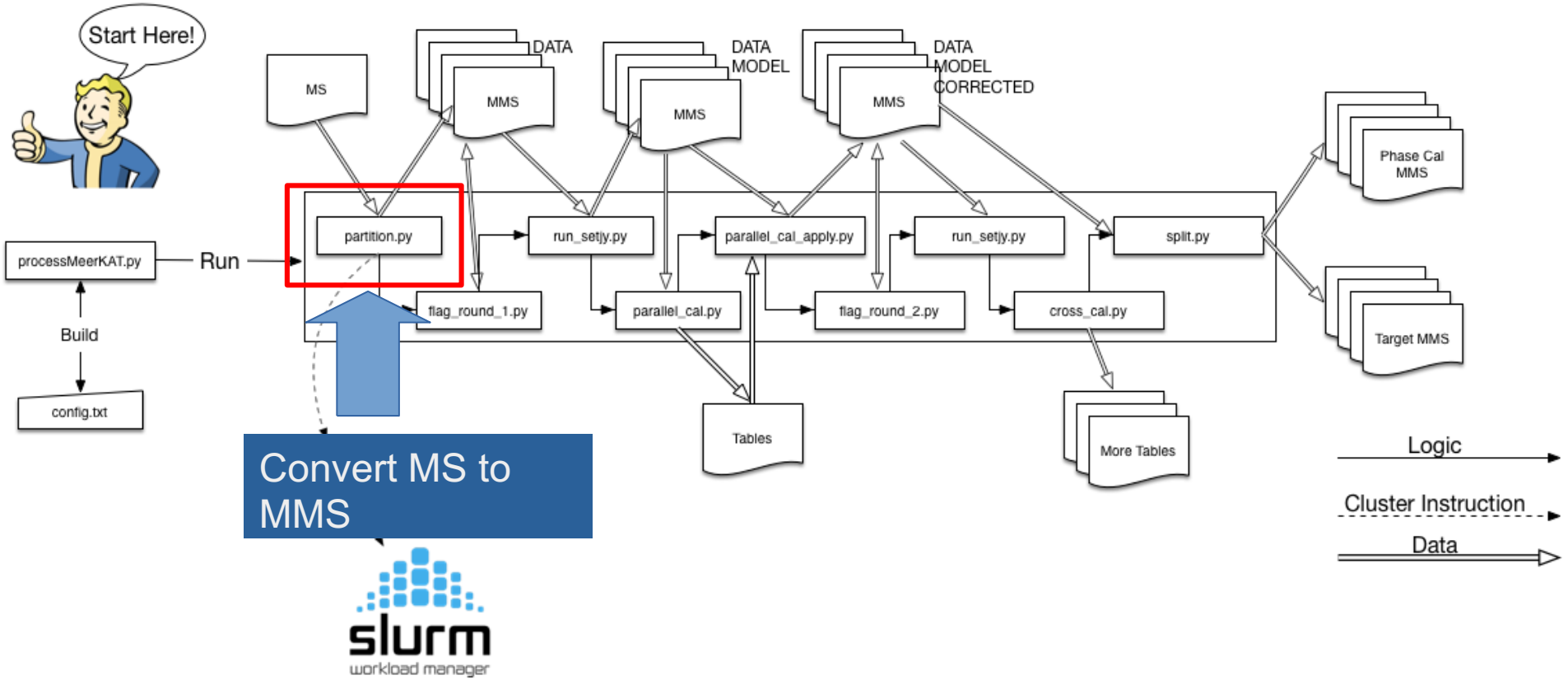

Start Here!
MS
DATA
MMS
DATA
MODEL
MMS
DATA
MODEL
CORRECTED
MMS
Phase Cal
MMS
processMeerKAT.py
Run
Build
config.txt
partition.py
run_setjy.py
parallel_cal_apply.py
run_setjy.py
split.py
flag_round_1.py
parallel_cal.py
flag_round_2.py
cross_cal.py
Target MMS
Tables
More Tables
partition.sbatch
slurm
workload manager
Logic
Cluster Instruction
Data

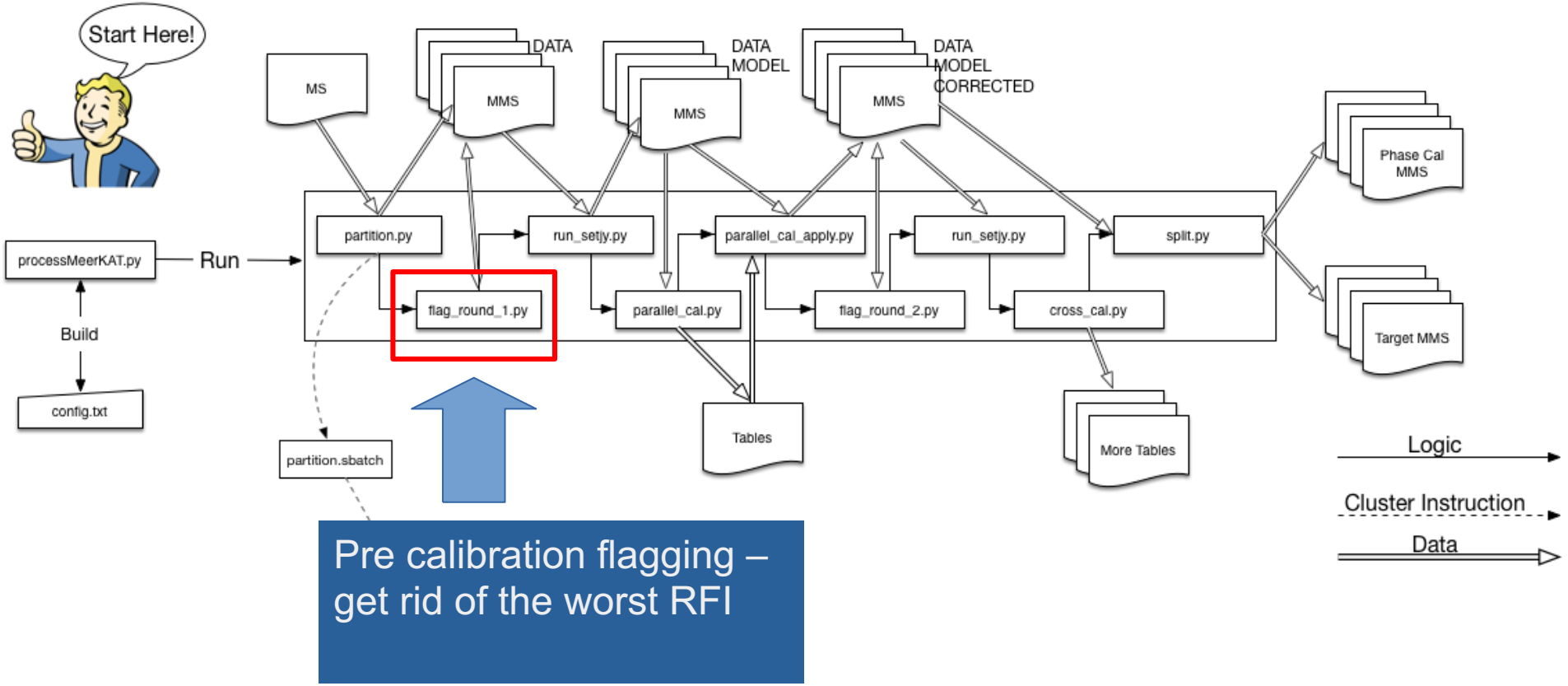
Pipeline goals

- Full Stokes calibration with the aim of producing continuum images/polarization cubes/spectral line cubes
  - (Full Stokes required for maximizing sensitivity in Stokes I)

Pipeline Philosophy

- Use multi-measurement sets (MMS) to parallelize across a cluster, take advantage of MPI aware tasks in CASA
- “Do the right thing” - Sensible defaults, get phases, fluxes, and polarizations right

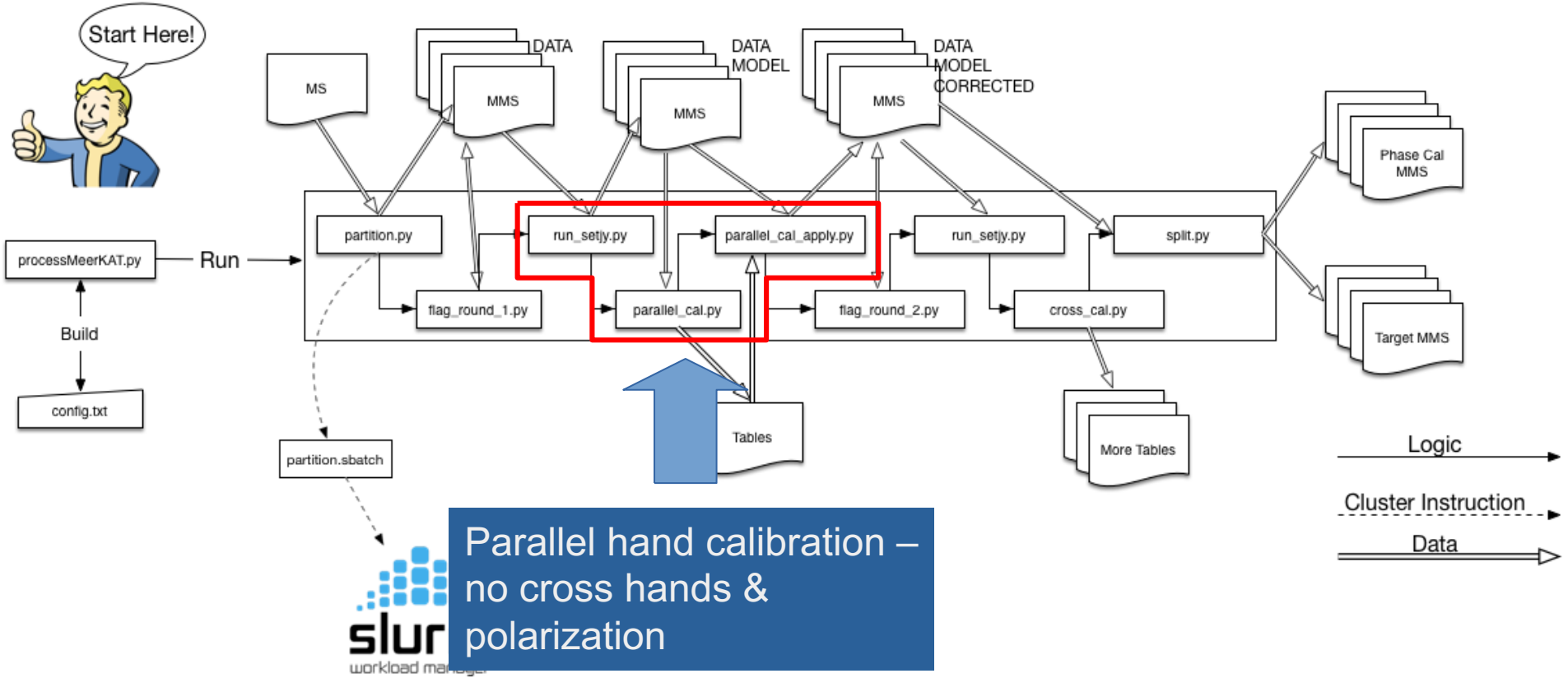

Start Here!
MS
DATA
MMS
DATA
MODEL
MMS
DATA
MODEL
CORRECTED
MMS
Phase Cal
MMS
processMeerKAT.py
Run
partition.py
run_setjy.py
parallel_cal_apply.py
run_setjy.py
split.py
Target MMS
Build
config.txt
flag_round_1.py
parallel_cal.py
flag_round_2.py
cross_cal.py
Tables
More Tables
Convert MS to
MMS
slurm
workload manager
Logic
Cluster Instruction
Data



Pre calibration flagging – get rid of the worst RFI

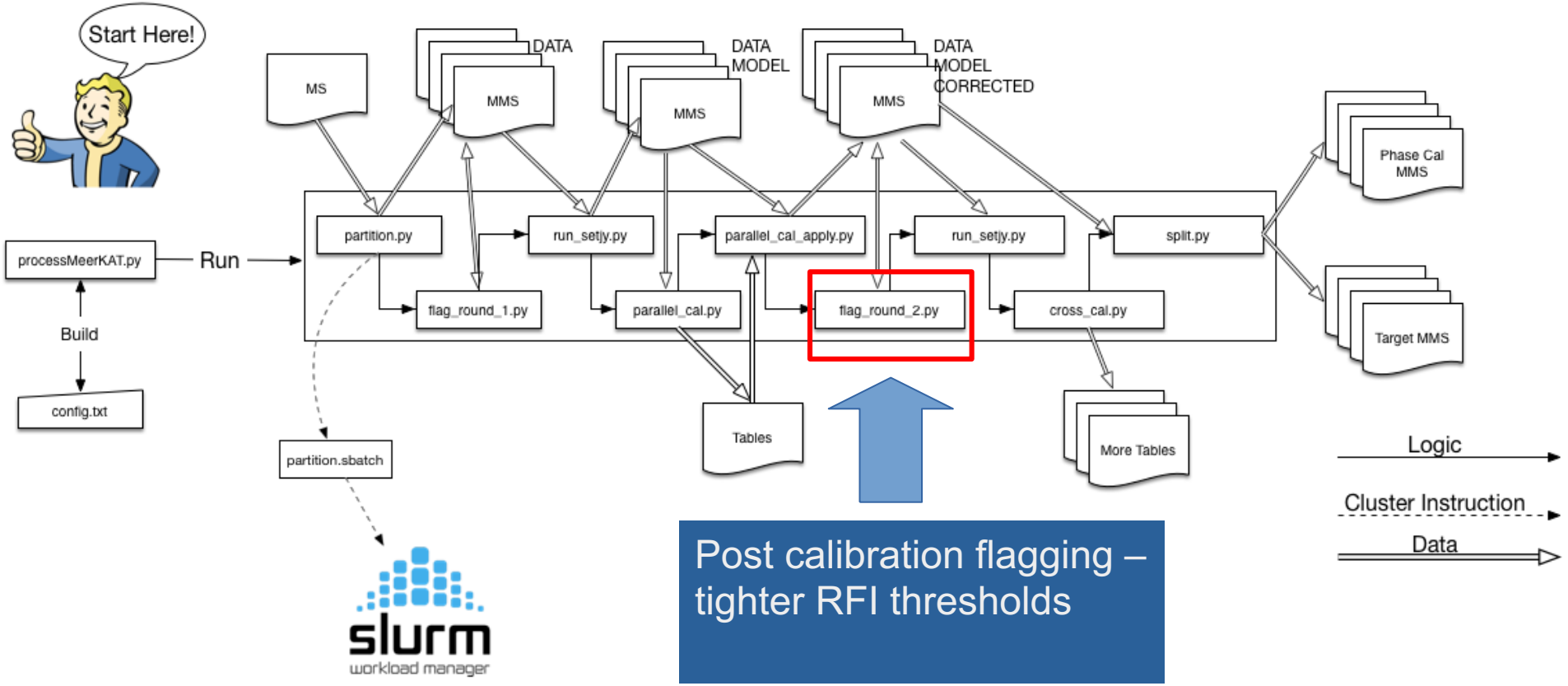
Flagging

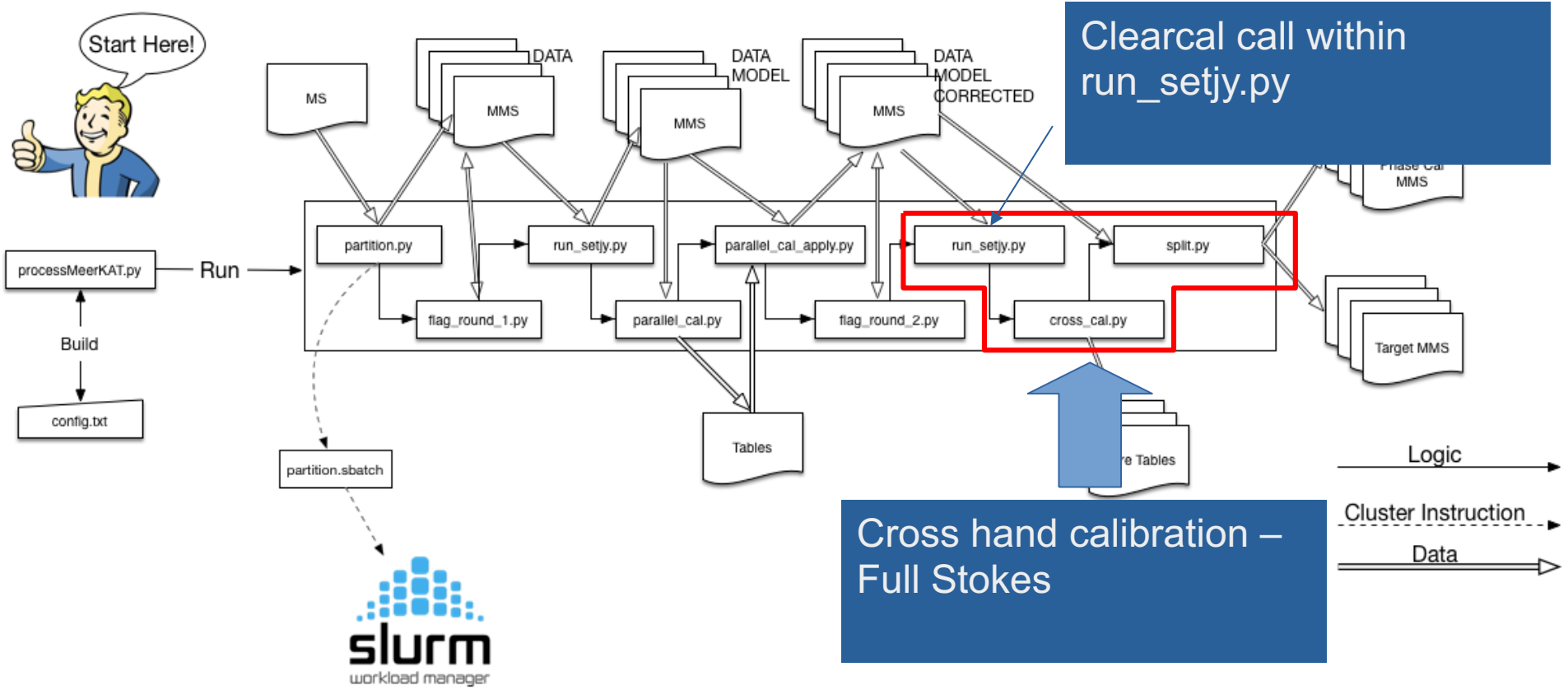
- Combination of clipping, tfcrop and rflag
- Set conservative limits while flagging pre-calibration
  - In principle tfcrop takes out the bandpass shape prior to flagging
- In general rflag more effective, particularly shorter baselines



Parallel hand calibration – no cross hands & polarization

```
logger.info(" starting antenna-based delay (kcorr)\n → %s" % calfiles.kcorrfile)
gaincal(vis=visname, caltable = calfiles.kcorrfile, field
        = fields.kcorrfield, refant = referenceant,
        minblperant = minbaselines, solnorm = False,  gaintype = 'K',
        solint = 'inf', combine = '', parang = False, append = False)

logger.info(" starting bandpass → %s" % calfiles.bpassfile)
bandpass(vis=visname, caltable = calfiles.bpassfile,
        field = fields.bpassfield, refant = referenceant,
        minblperant = minbaselines, solnorm = True,  solint = 'inf',
        combine = 'scan', bandtype = 'B', fillgaps = 8,
        gaintable = calfiles.kcorrfile, gainfield = fields.kcorrfield,
        parang = False, append = False)

logger.info(" starting gain calibration\n → %s" % calfiles.gainfile)
gaincal(vis=visname, caltable = calfiles.gainfile,
        field = fields.gainfields, refant = referenceant,
        minblperant = minbaselines, solnorm = False,  gaintype = 'G',
        solint = 'inf', combine = '', calmode='ap',
        gaintable=[calfiles.kcorrfile, calfiles.bpassfile],
        gainfield=[fields.kcorrfield, fields.bpassfield],
        parang = False, append = False)

# Only run fluxscale if bootstrapping
if len(fields.gainfields) > 1:
    fluxscale(vis=visname, caltable=calfiles.gainfile,
            reference=[fields.fluxfield], transfer='',
            fluxtable=calfiles.fluxfile, append=False)
```


Start Here!
processMeerKAT.py
Run
Build
config.txt
partition.sbatch
slurm
workload manager
MS
DATA
MMS
DATA
MODEL
MMS
DATA
MODEL
CORRECTED
MMS
partition.py
flag_round_1.py
run_setjy.py
parallel_cal.py
parallel_cal_apply.py
flag_round_2.py
run_setjy.py
cross_cal.py
split.py
Tables
More Tables
Phase Cal
MMS
Target MMS
Logic
Cluster Instruction
Data
Post calibration flagging –
tighter RFI thresholds


Start Here!
processMeerKAT.py
Run
Build
config.txt
MS
DATA
MMS
DATA
MODEL
MMS
DATA
MODEL
CORRECTED
MMS
partition.py
run_setjy.py
parallel_cal_apply.py
run_setjy.py
split.py
flag_round_1.py
parallel_cal.py
flag_round_2.py
cross_cal.py
Target MMS
Tables
partition.sbatch
slurm
workload manager
Clearcal call within run_setjy.py
Cross hand calibration – Full Stokes
Logic
Cluster Instruction
Data

Polarization calibration

- Need to estimate :
  - Instrumental leakage
  - Calibrator source polarization
- Do both by observing a single source over several parallactic angles

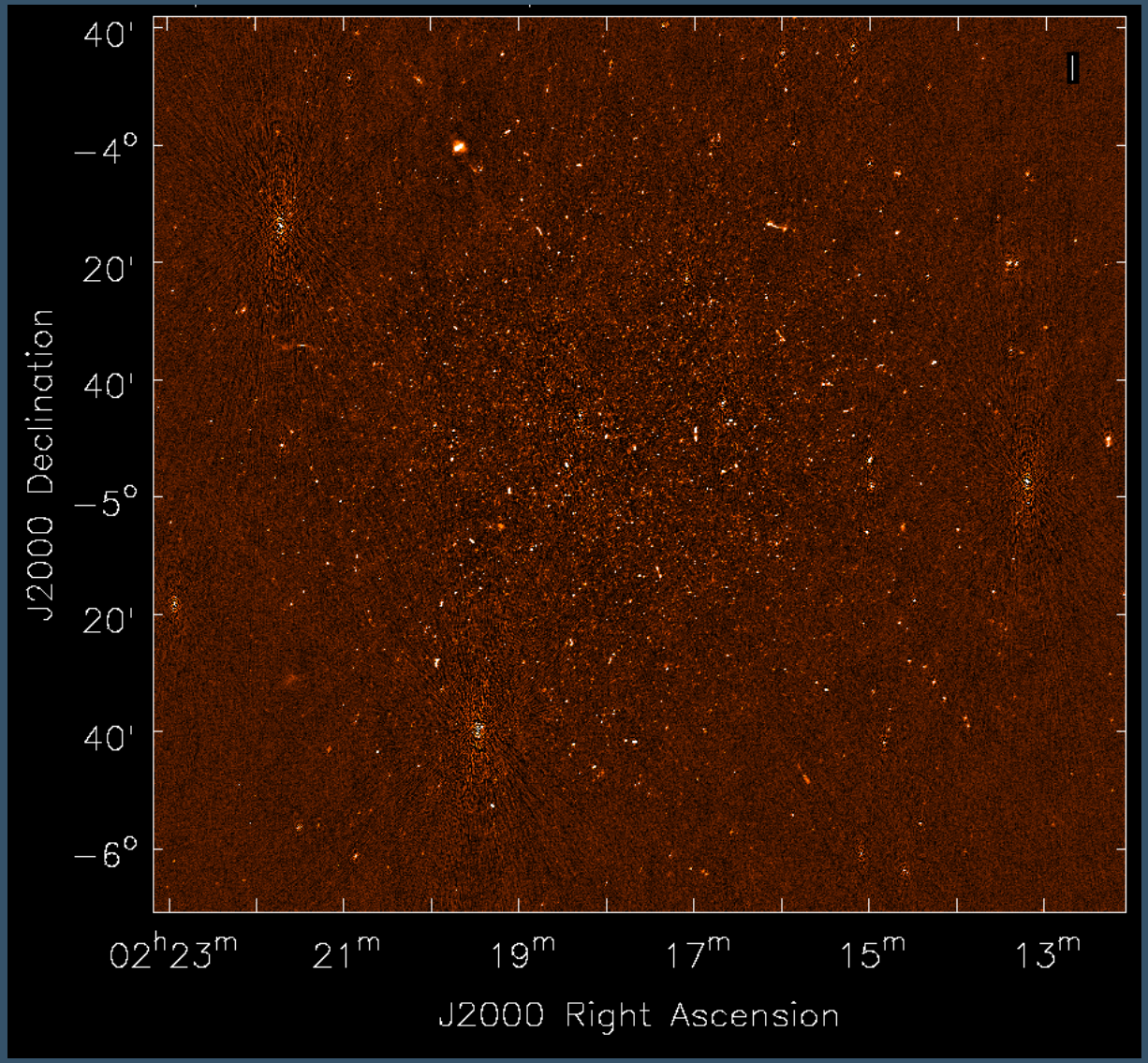
Polarization calibration

- CASA “helper” task GainfromQU estimates Q, U of source from gain variation vs. Parallactic angle
- Use this information to solve for X-Y phase and instrumental leakage

```
gaincal(vis=visname, caltable=gain1file, field=fields.fluxfield,
        refant=referenceant, solint='10min', minblperant=minbaselines,
        solnorm=False, gaintype='G',
        gaintable=[calfiles.kcorrfile, calfiles.bpassfile,
            calfiles.xdelfile],
        gainfield = [fields.kcorrfield, fields.bpassfield,
            fields.xdelfield], append=False, parang=True)

gaincal(vis=visname, caltable=gain1file, field=fields.secondaryfield,
        smodel=[1,0,0,0], refant=referenceant, solint='10min',
        minblperant=minbaselines, solnorm=False, gaintype='G',
        gaintable=[calfiles.kcorrfile, calfiles.bpassfile,
            calfiles.xdelfile],
        gainfield = [fields.kcorrfield, fields.bpassfield,
            fields.xdelfield],
        append=True, parang=True)

# implied polarization from instrumental response
logger.info("\n Solve for Q, U from initial gain solution")
GainQU = qufromgain(gain1file)
logger.info(GainQU[int(fields.dpolfield)])

logger.info("\n Starting x-y phase calibration\n → %s" % xy0ambpfile)
gaincal(vis=visname, caltable = xy0ambpfile, field = fields.dpolfield,
        refant = referenceant, solint = 'inf', combine = 'scan',
        gaintype = 'XYf+QU', minblperant = minbaselines,
        smodel = [1.,0.,1.,0.], preavg = 200.0,
        gaintable = [calfiles.kcorrfile,calfiles.bpassfile,
            gain1file, calfiles.xdelfile],
```

Polarization calibration

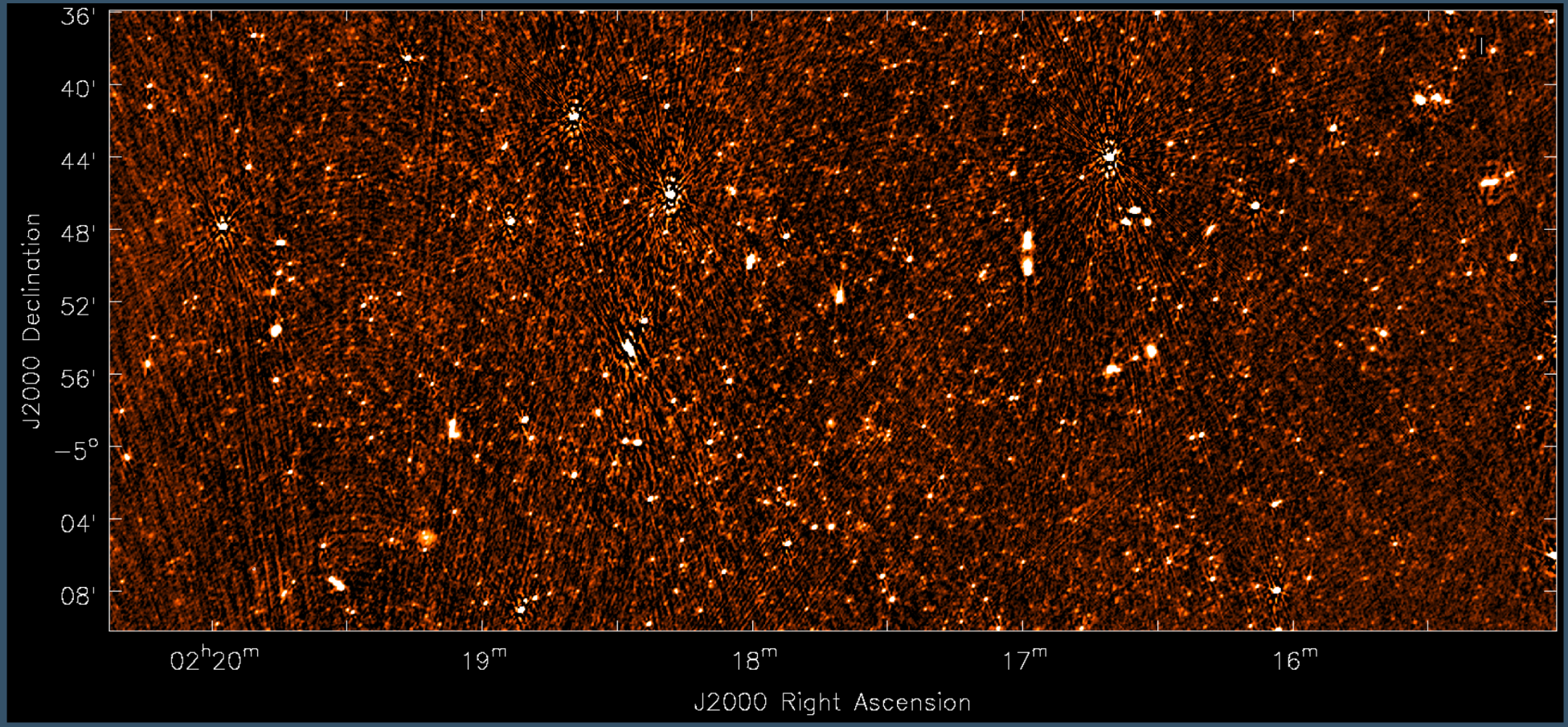
- Limitations: Leakage and QU estimation assume a constant value across an SPW.
  - Reasonable for VLA/ALMA, with several small (~ 64 MHz) SPWs
  - Problematic for MeerKAT with one ~ 800 MHz SPW
- On axis calibration still good, in reasonable agreement with VLA polarization measurements



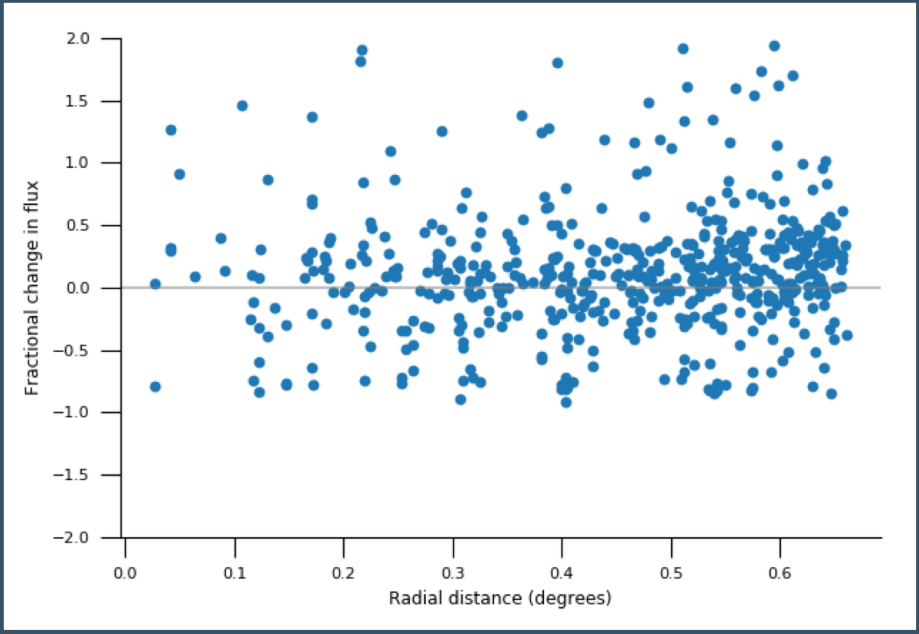
MIGHTEE XMM LSS 12

RMS : ~ 4 uJy/beam

Self-cal improvements

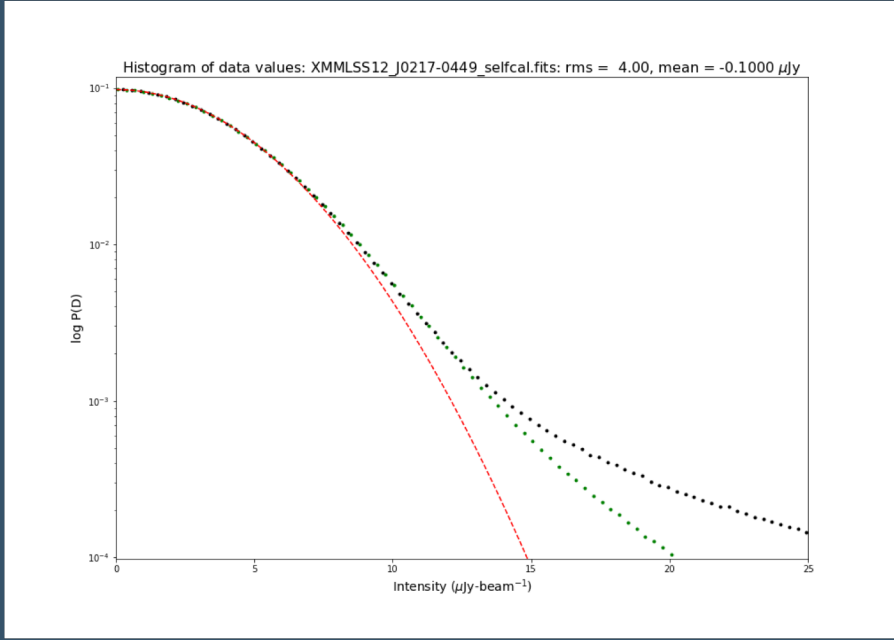


Fluxes



VLA vs MeerKAT XMM LSS

Catalogs courtesy Ian Heywood

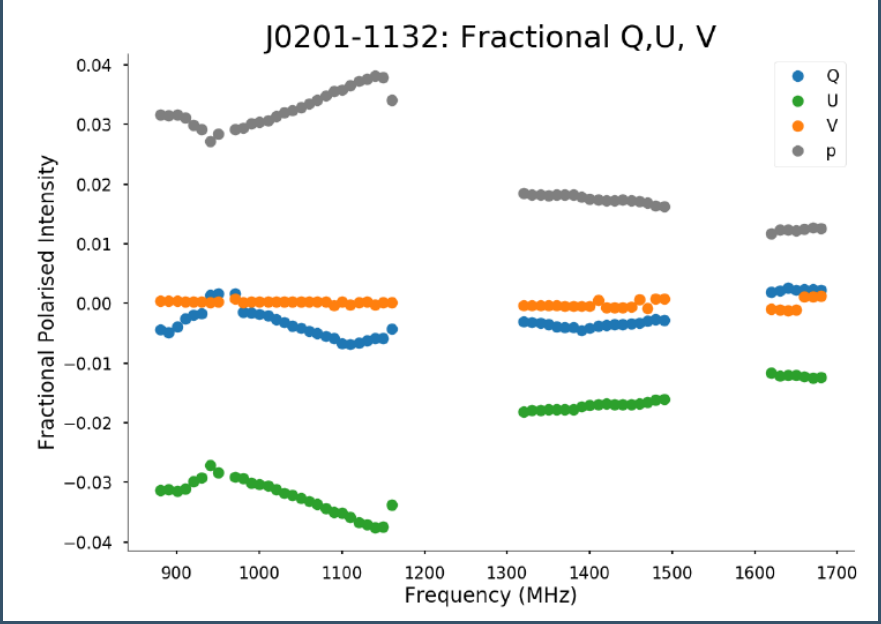


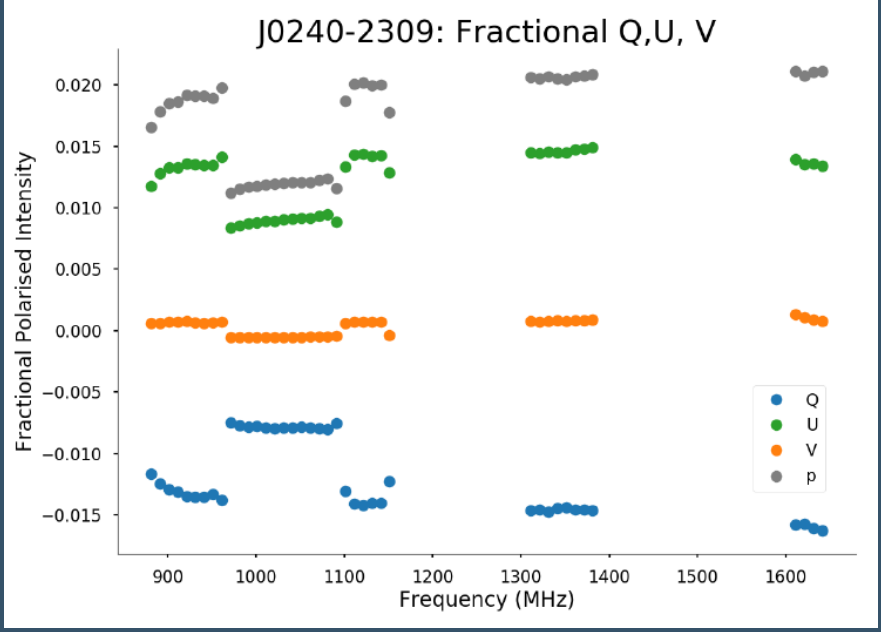
Pixel histogram of the MeerKAT XMM LSS image

Plot courtesy Russ Taylor

Polarization cubes

- Stokes V a good indicator of calibration quality

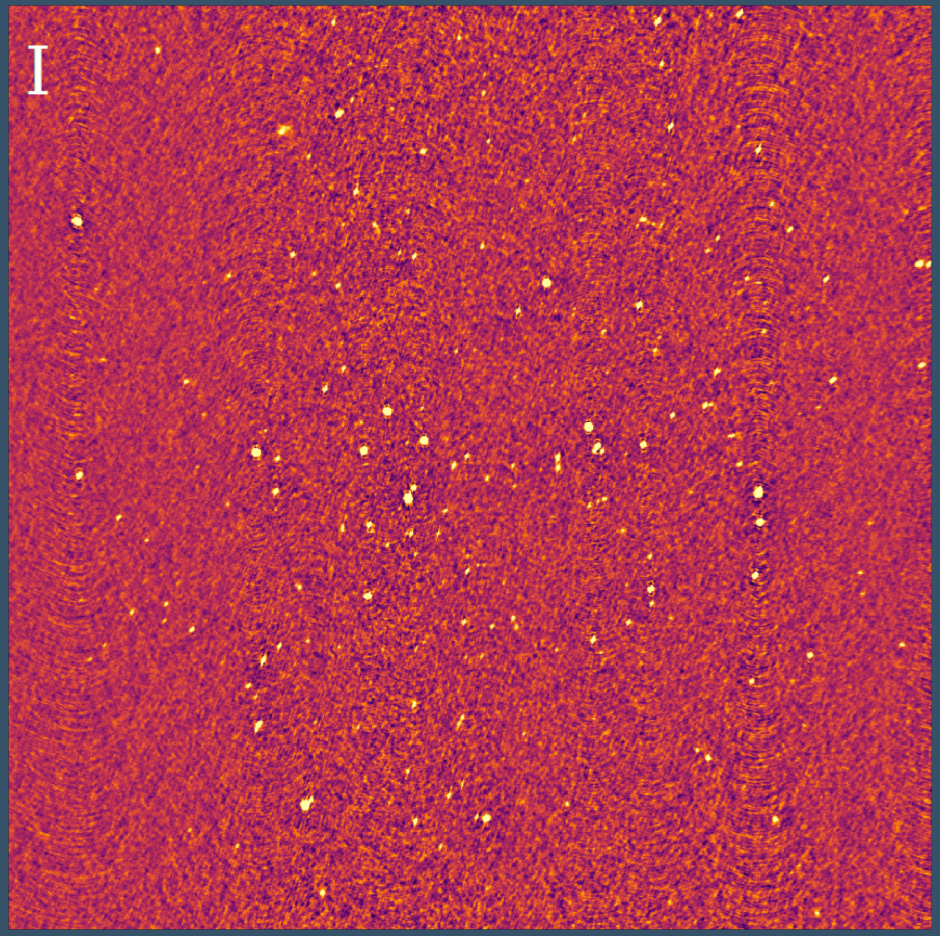




Fractional polarization

Calibrator	Field	NVSS (1.4 Ghz) %	MeerKAT band average (1.2 Ghz) %
J0240-2309	CDFS	0.80	1.72
3C237	COSMOS	< 0.13	0.1
J0201-1132	XMM LSS	3.41	2.31

Polarization cubes



Stokes images of a single channel of the XMM LSS pointing at 1120 MHz

Next steps

- Get off-axis, wideband polarizations working
  - Full Stokes AW projection in collaboration with NRAO, talk this afternoon
- Get more reliable wideband polarization
  - CASA is not RM “aware” over a wideband, single SPW. Workarounds are being tested.

Summary

- The IDIA pipeline produces Stokes I images with ~ 8-10 uJy/beam rms with no self-cal, and (currently) 4 uJy/beam after self-cal
- Fluxes typically accurate to ~ 5%
- On axis polarization is good, but will improve once the wideband is accounted for correctly