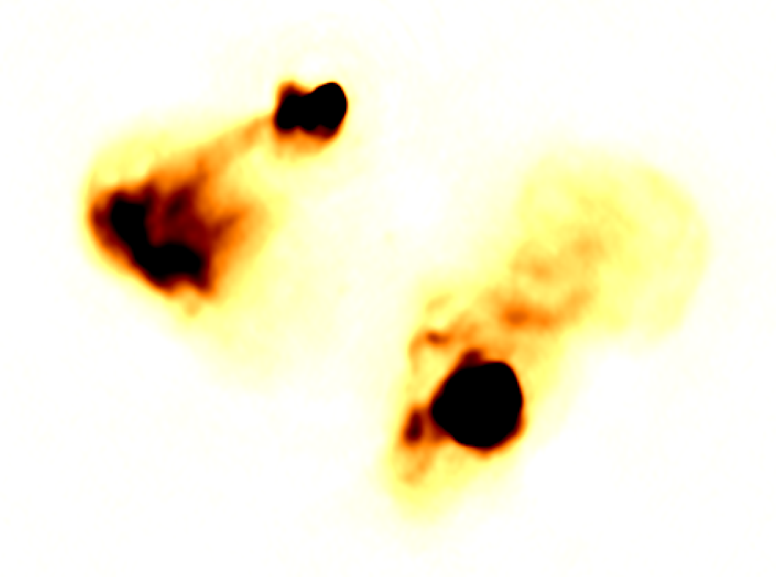
Designing radio-astronomical software for delivering science-ready products



André Offringa

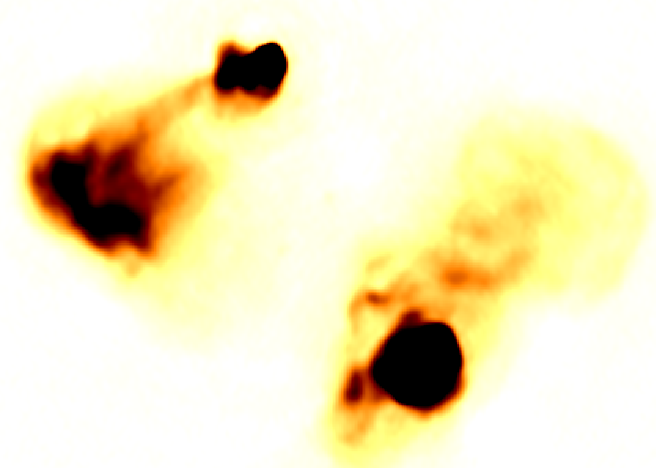
Astronomer at *ASTRON* & *Kapteyn Institute Groningen*

Co-PI of LOFAR EoR project

ASTRON
Netherlands Institute for Radio Astronomy

Designing radio-astronomical software for delivering science-ready products

André Offringa
Astronomer at *ASTRON* & *Kapteyn Institute Groningen*
Co-PI of LOFAR EoR project



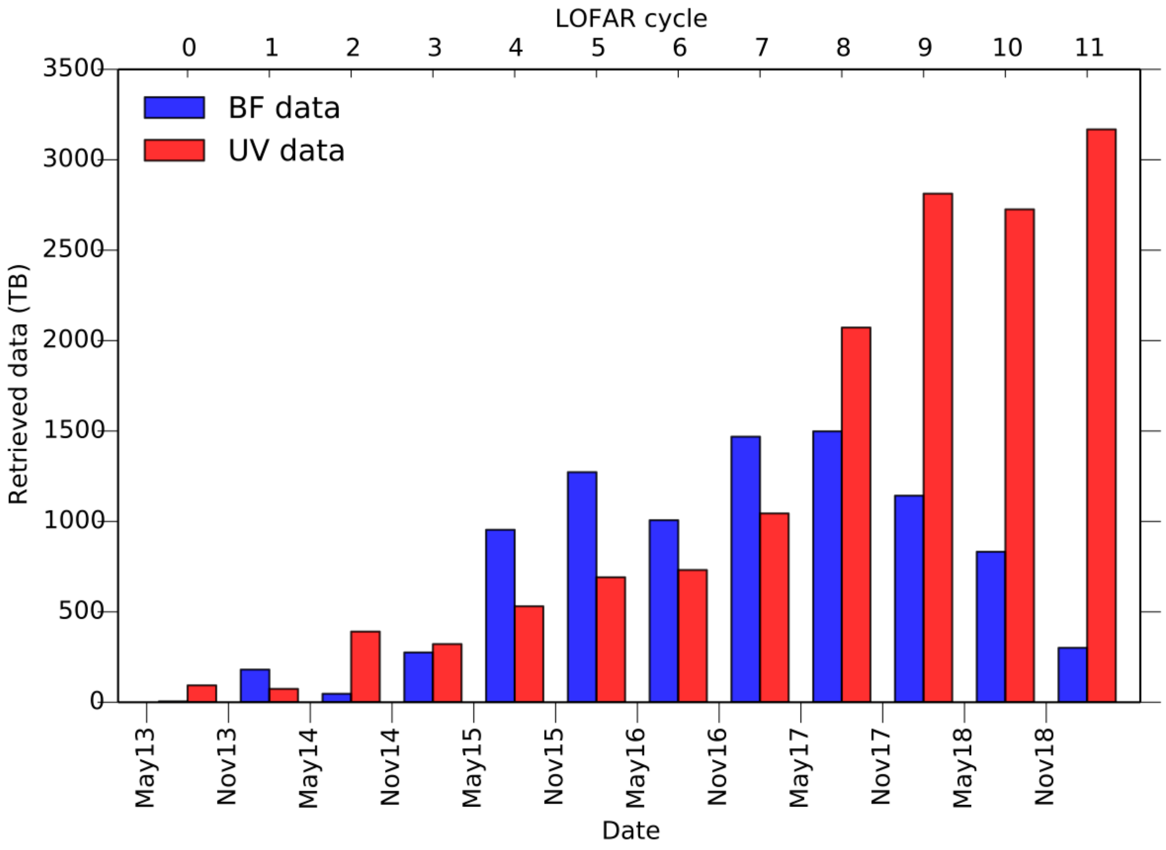
ASTRON
Netherlands Institute for Radio Astronomy

Radio data

- Increase in computing power makes it attractive to develop (physically) “simpler” telescopes with better electronics
  - E.g. LOFAR: simple antennas, but large N
- Large field of view, high spatial, time and frequency resolutions
- Increases processing challenge

Radio data

- Large data volumes
  - 1-10 GB/s for LOFAR
- Requires lots of processing & computing
- Novel algorithms required to reach scientific quality

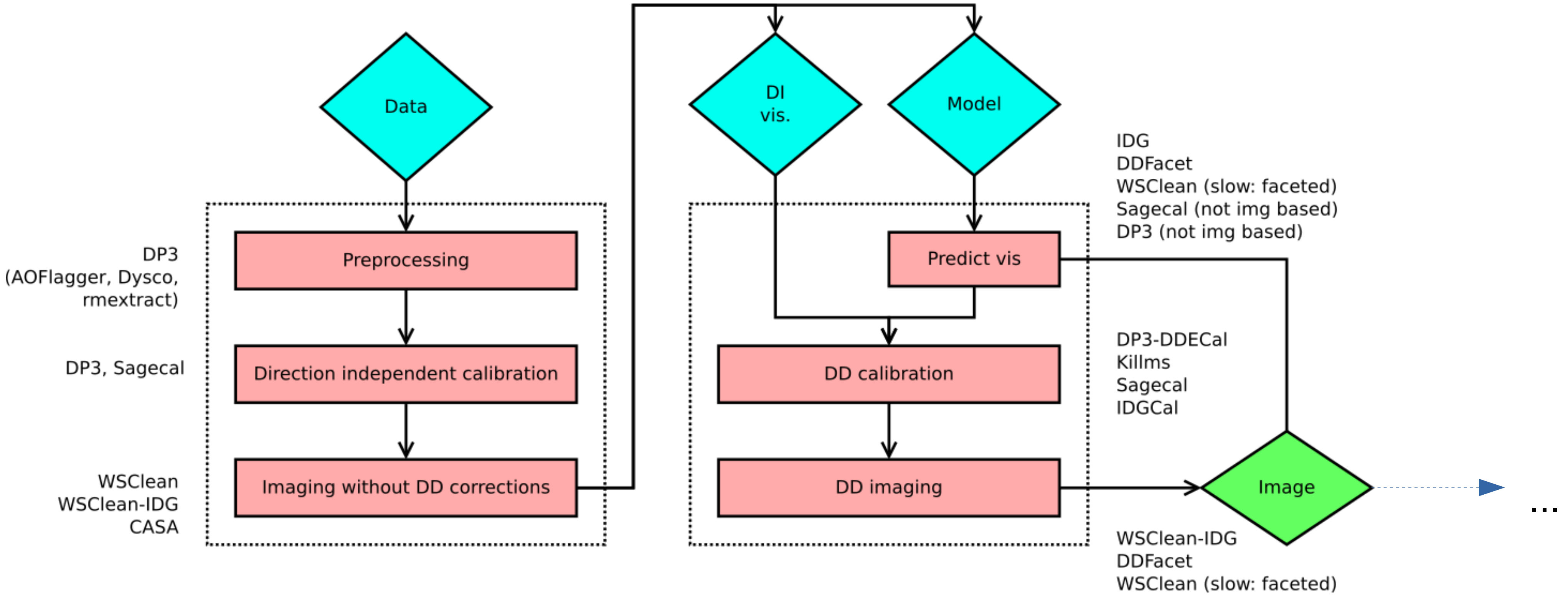


Downloaded LOFAR data per cycle (half year)
BF=beam formed, **UV**=imaging

Square-Kilometre Array

- More antennas, more data (~TB/s)
- Higher accuracy requirements
- Design finished, construction soon to start!

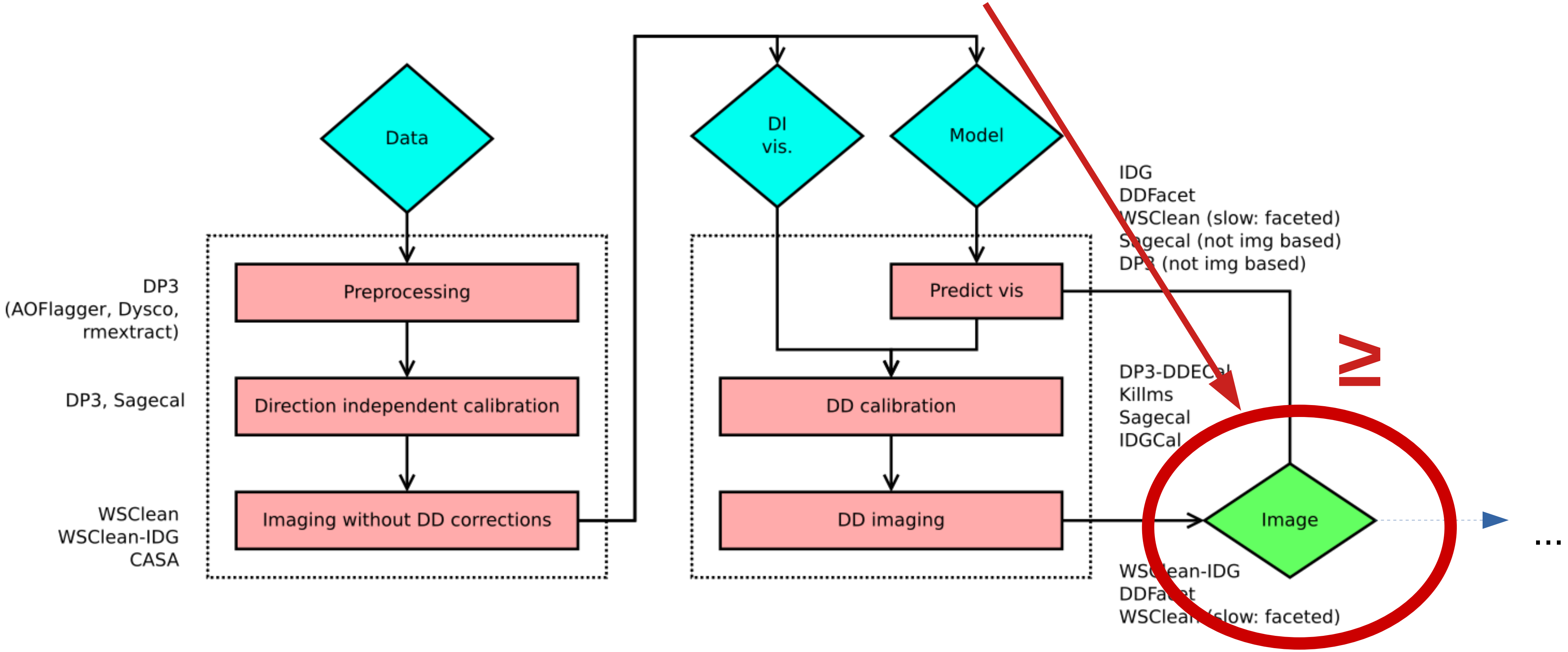
Example processing overview



Pipeline overview for generic LOFAR imaging

ASTRON
Netherlands Institute for Radio Astronomy

What is science-ready data?

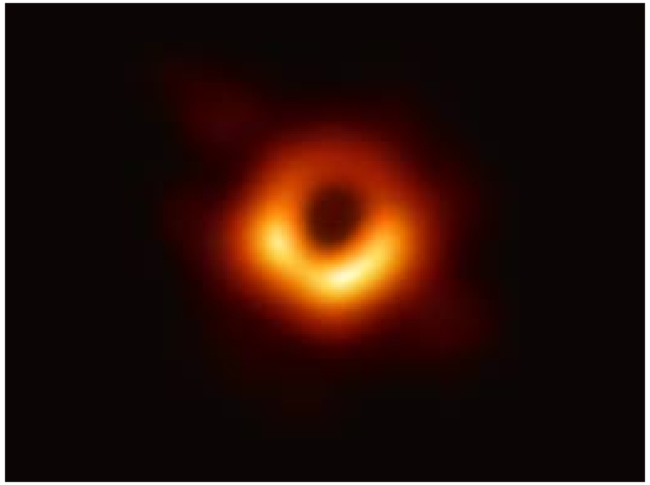


Pipeline overview for generic LOFAR imaging

ASTRON
Netherlands Institute for Radio Astronomy

What is science-ready data?

- At least a high-quality image
  - E.g. for Dutch LOFAR: 10k x 10k, 5'' (and similar for SKA)
  - 0.1'' for international ('long baseline') LOFAR
  - Enough for some science goals (e.g. event horizon telescope)
- Often, more is required to extract science:
  - Source positions, size
  - Spectral indices (or spectral information)
  - Recover diffuse emission
  - Include international baselines with full FOV (100k x 100k images!)
  - Power spectra (e.g. Epoch of Reionization)
  - Polarization
  - Long observing runs (e.g. Epoch of Reionization: 100 nights)
  - Need to model off sources away from the pointing centre



The EHT Collaboration et al. 2019

ASTRON
Netherlands Institute for Radio Astronomy

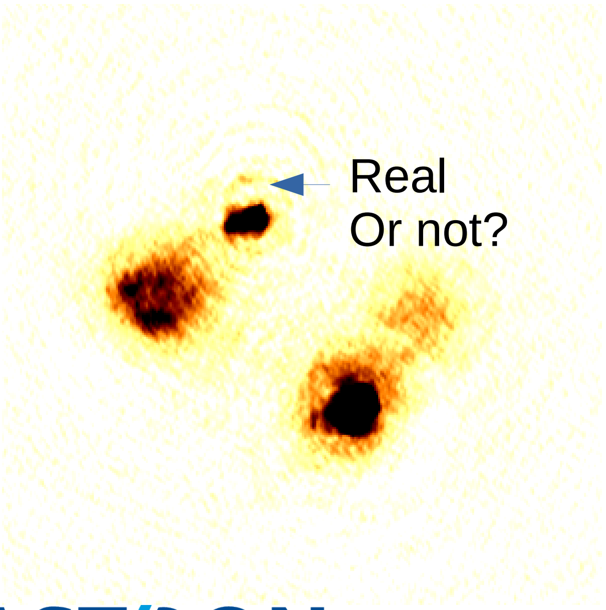
What is science-ready data?

- In the ideal situation, an astronomer:
  - has an idea, with a certain hypothesis
  - requests (and is awarded) observing time
  - receives the "science-ready" data products
  - is able to immediately answer the hypothesis
  - Nobel price.
- Advantages:
  - (Almosts) no *redundant* processing knowledge required by astronomer
  - Less time in learning instrument → more time for science!
  - Accessible to any astronomer → hence more science!
  - Nobel price.

What is science-ready data?

- In the ideal situation, an astronomer:
  - has an idea, with a certain hypothesis
  - requests (and is awarded) observing time
  - receives the “science-ready” data products
  - is able to immediately answer the hypothesis
  - Nobel price.
- Advantages:
  - (Almosts) no *redundant* processing knowledge required by astronomer
  - Less time in learning instrument → more time for science!
  - Accessible to any astronomer → hence more science!
  - Nobel price.

Even when all processing is done by an observatory, astronomer’s still need to *understand their data*



ASTRON
Netherlands Institute for Radio Astronomy

Not how radio astronomy traditionally works

- Observatories just provide the data
- Many PhDs are spend on data processing
- Many tools are written to solve the same problem
- Telescope is only fully accessible to expert teams

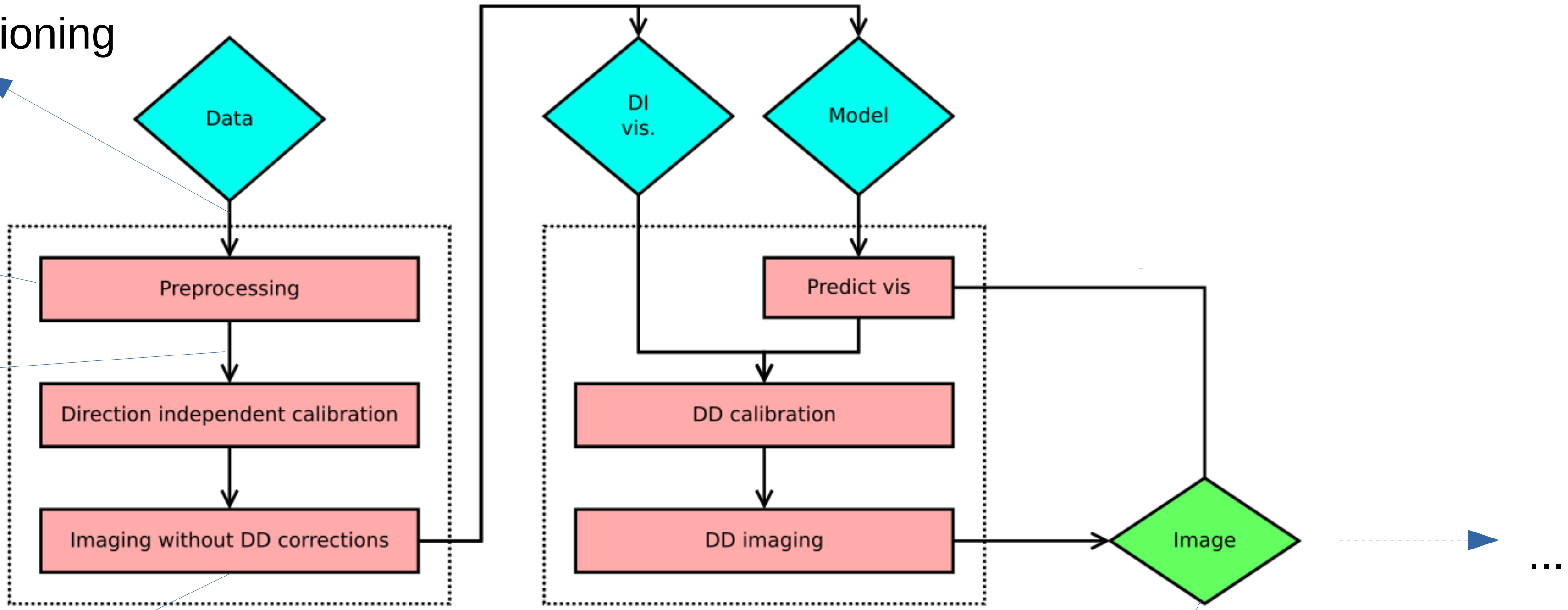
- Recently, this is changing:
  - E.g. LOFAR, ALMA, APERTIF, SKA (want to) provide higher level products
  - (Also many posters about great pipelines here at ADASS!)

What can the LOFAR observatory do for you?

Raw data
During commissioning
(2010)

Some preprocessing
In first cycles
(2013)

Include lossy
compression
(Dysco; 2018)



Full direction-independent calibration
done by LOFAR observatory
(2019)

Planned for later this year!

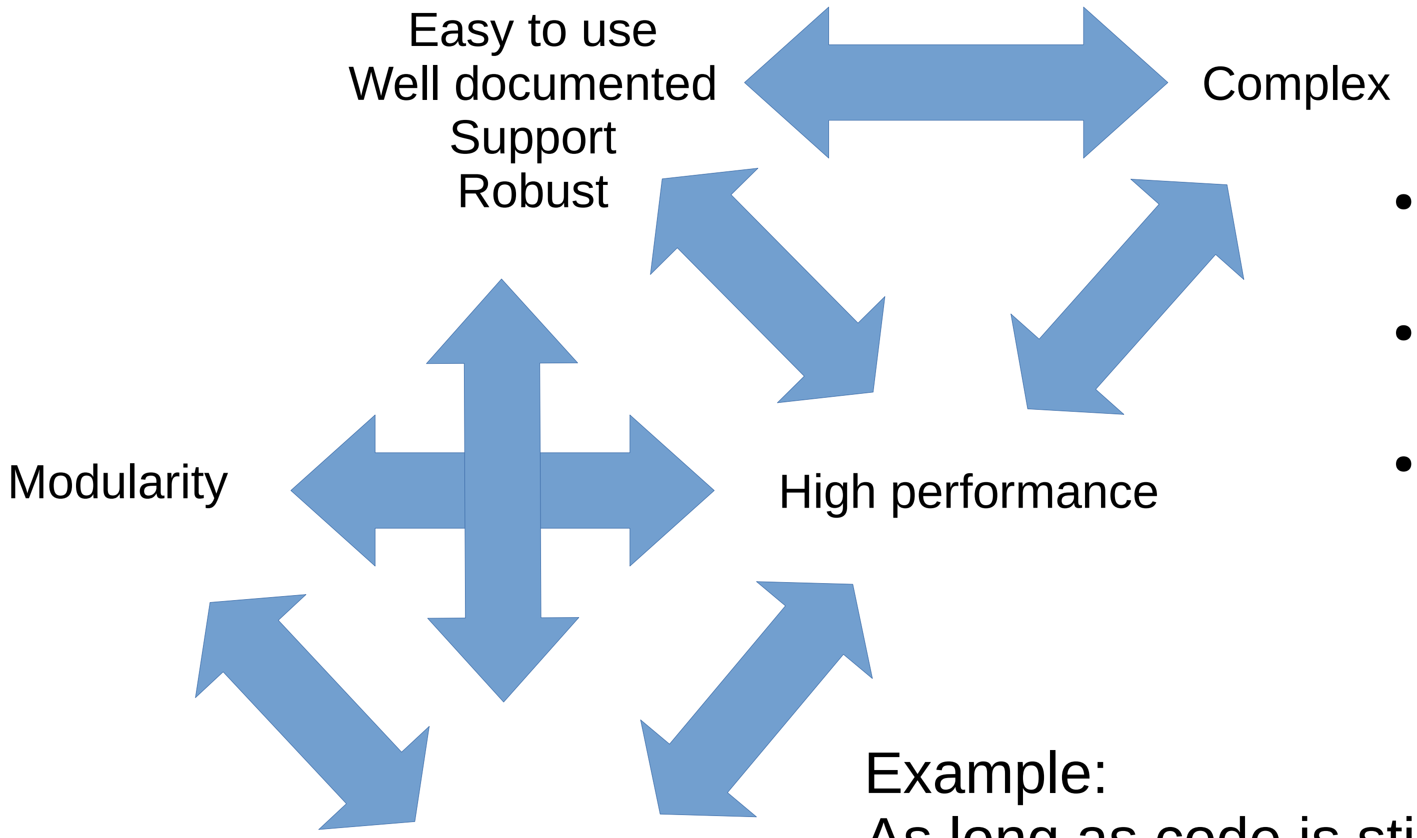
Pipeline overview for generic LOFAR imaging

ASTRON
Netherlands Institute for Radio Astronomy

Making radio-data processing pipelines is challenging!

- Complex
- High performance
- State of the art, experimental
  - Involves trial and error with algorithms
- Needs astronomical domain knowledge
  - Translates into a large number of 'heuristics' (sometimes even machine learning)
- Hard to get a grant to "write a generic pipeline"
  - Common answer: "that's not science!"
- No money / resources / credits / plan for support
- No formal software engineering processes used
- Difficulty often underestimated / not understood

Processing requirements



- Many of these software attributes ‘clash’
- Requiring one of these can be hard.
- Requiring *all of them simultaneously* is really, really difficult

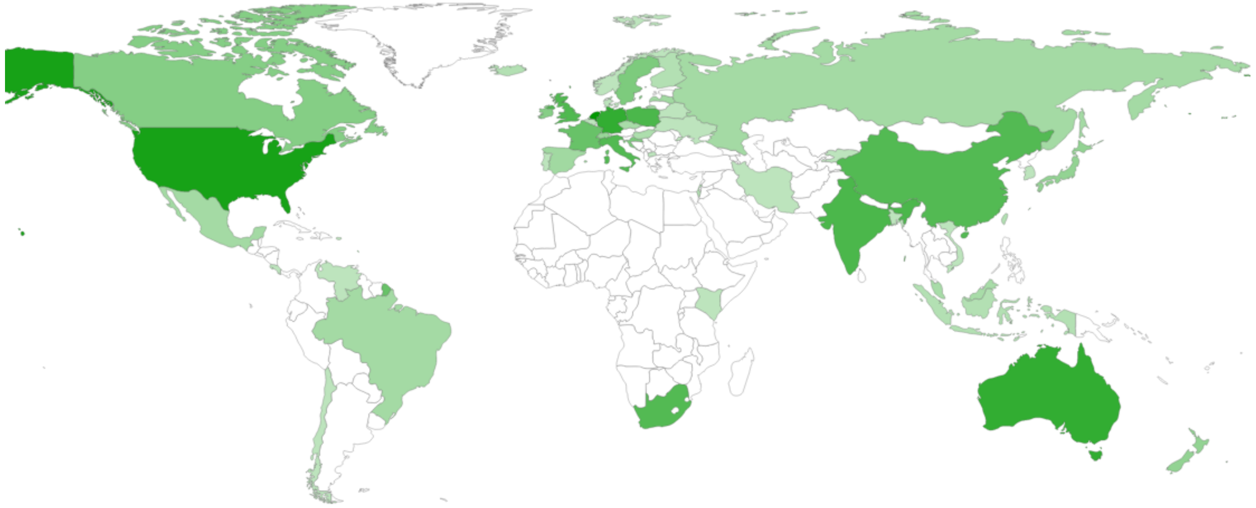
Example:
As long as code is still experimental (changing), it is difficult to support / document it.

Challenge of radio-data processing pipelines

- Many *unused* radio-astronomy tools have been published
  - Might be slightly diferent from what an astronomer want
  - No money for extension, support or maintenance available
- Next team needs to re-invent the wheel :(
  - Constructing a new algorithm is much more rewarding
- A tool that is not used might still provide new insights
- → Why is it not used?
- → publish your insights *including the negatives*

An example: AOFlagger

- AOFlagger is a tool for detecting interference in radio data
- Relatively large user base (for radio astronomy)
- Written in C++

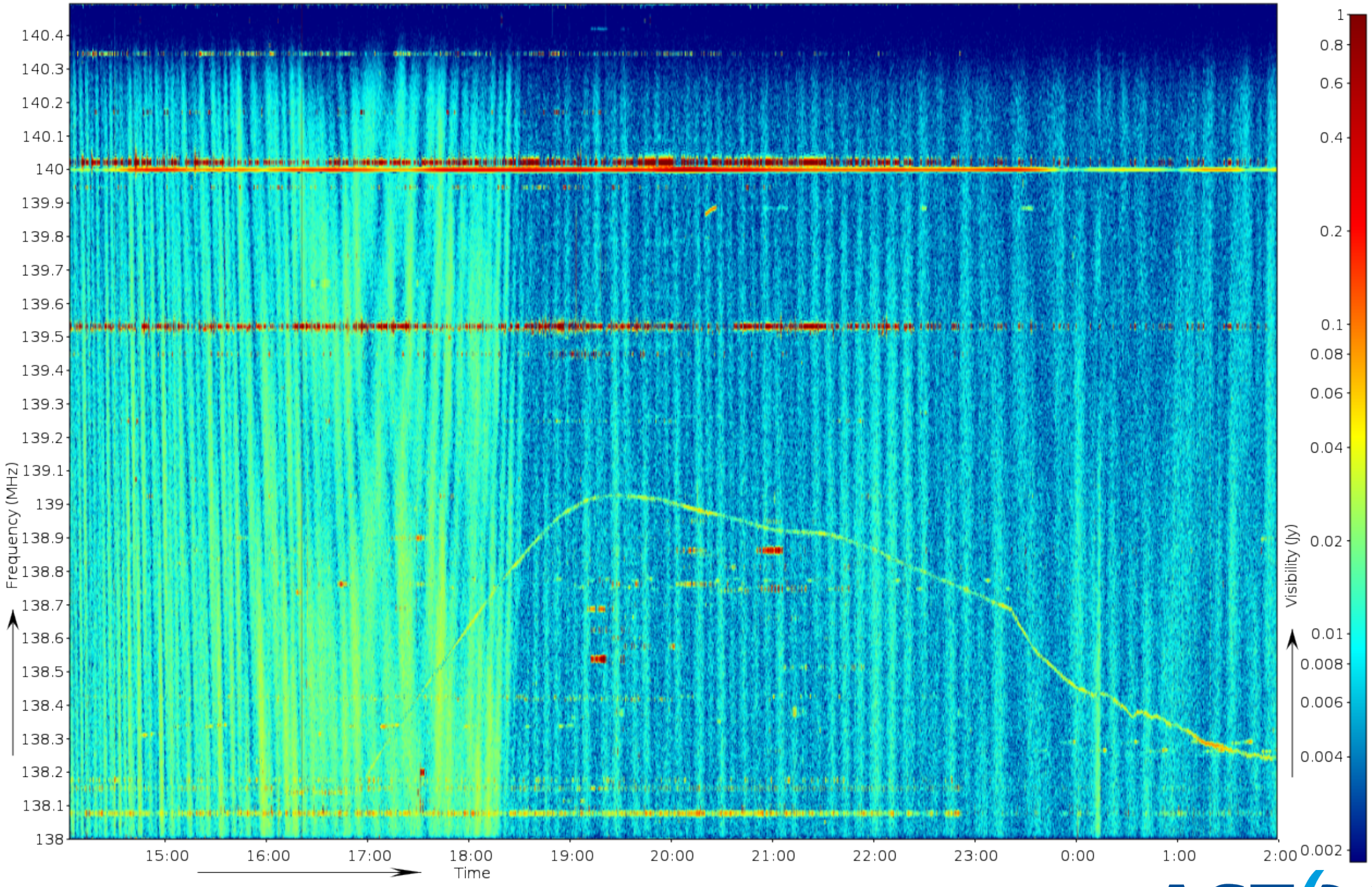


Source downloads per country
of latest version
(3000 total – excludes binary downloads)

- http://aoflagger.sourceforget.net

ASTRON
Netherlands Institute for Radio Astronomy

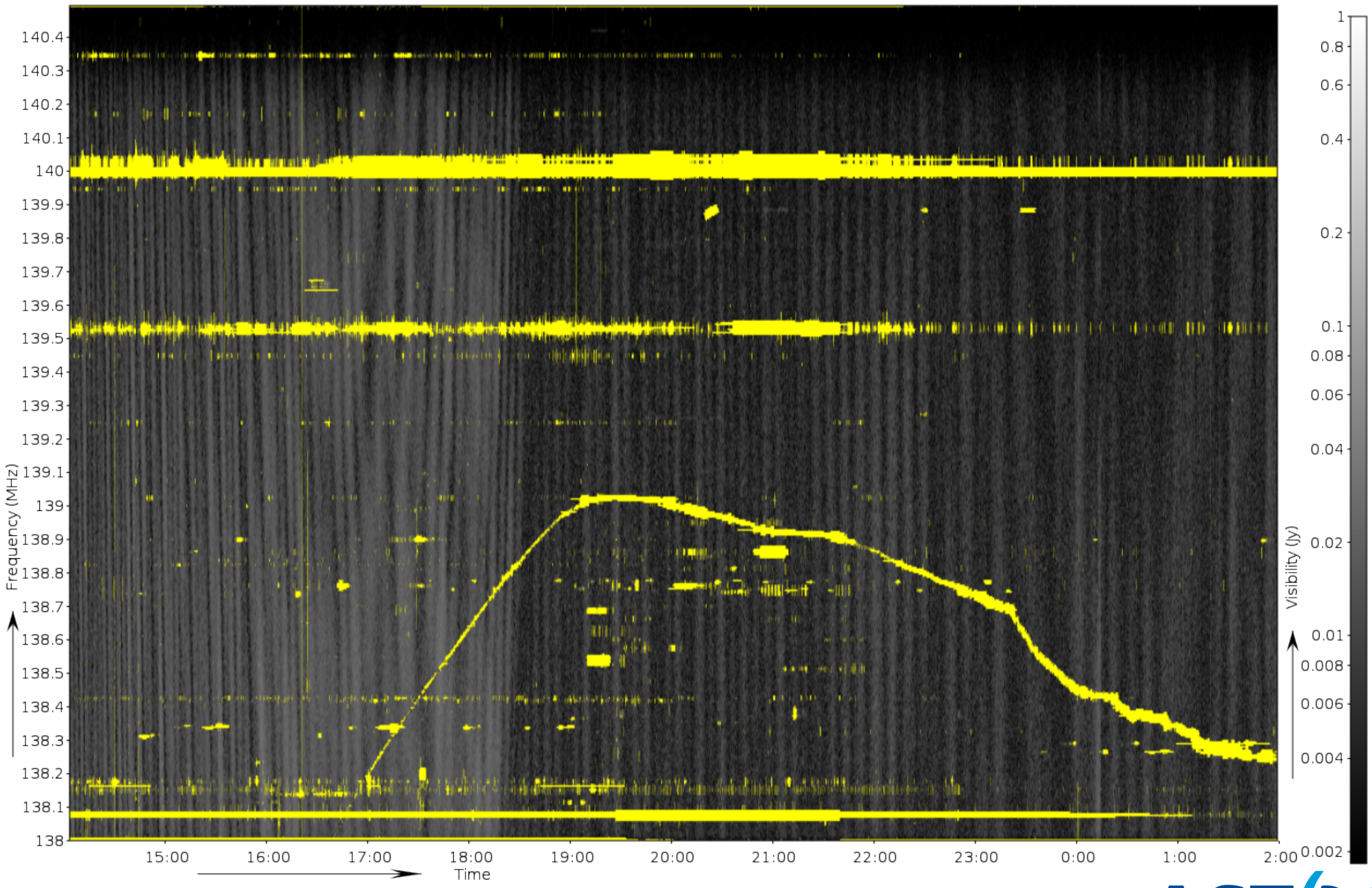
An example: AOFlagger



Part of a WSRT data set

ASTRON
Netherlands Institute for Radio Astronomy

An example: AOFlagger

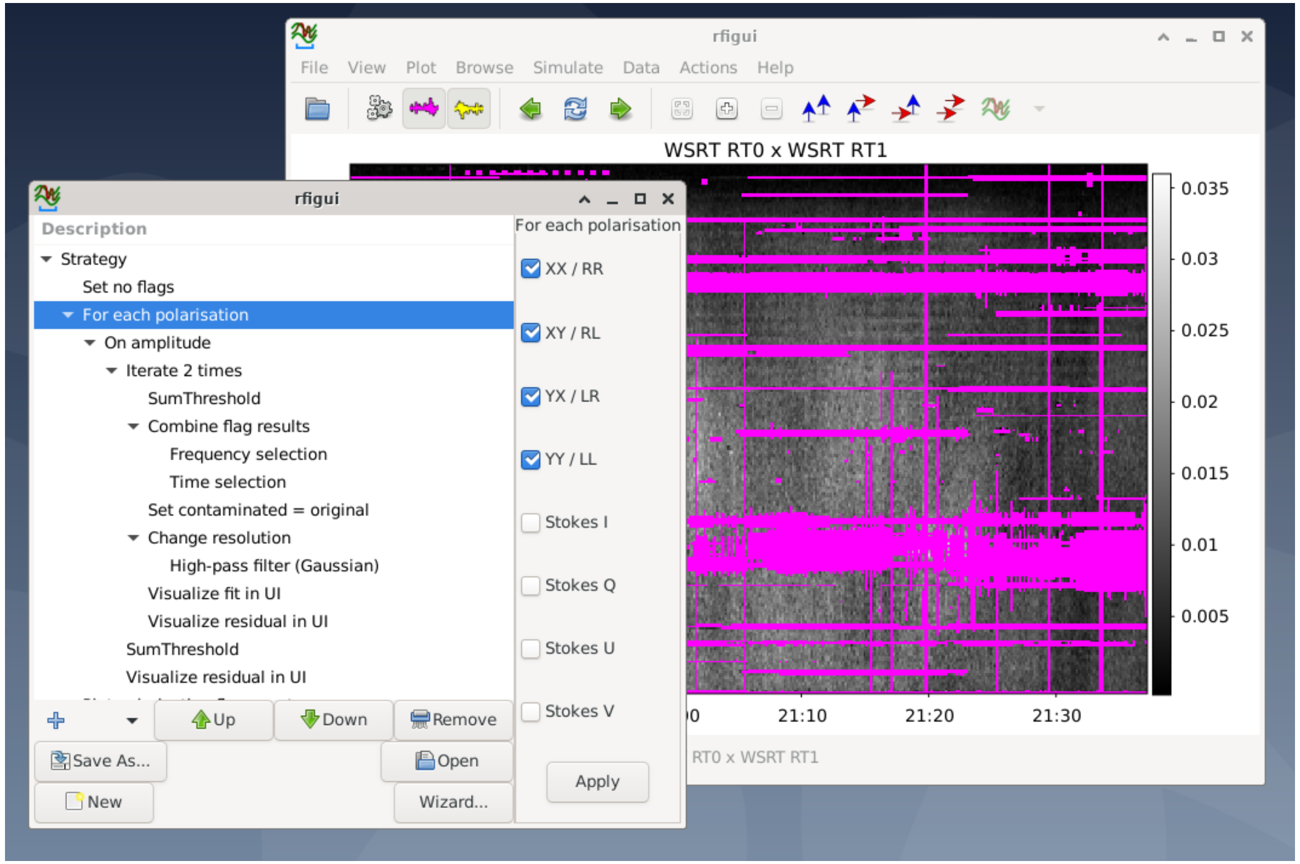


Part of a WSRT data set, flagged by AOFlagger

ASTRON
Netherlands Institute for Radio Astronomy

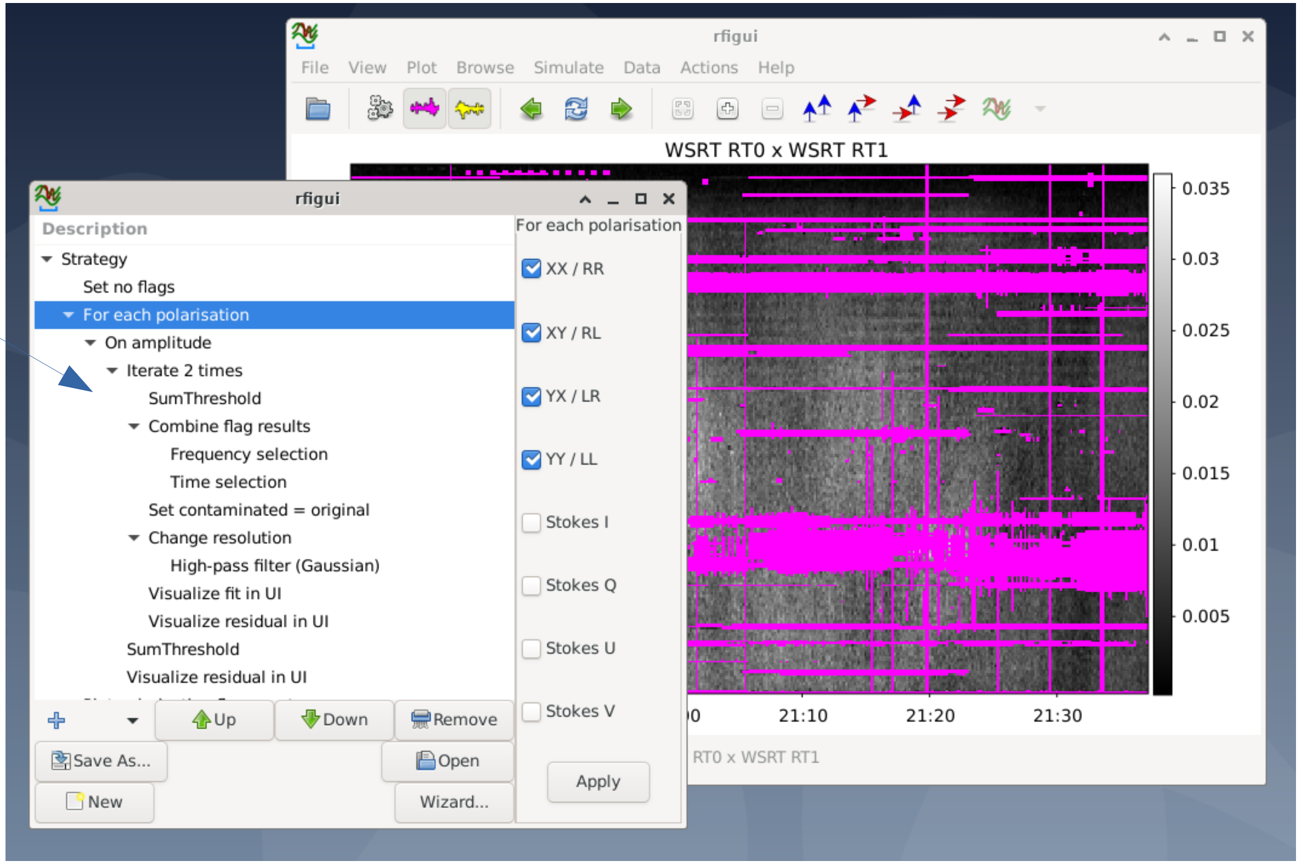
An example: AOFlagger

- Works with lots of specialized algorithms & heuristics
(66K lines of code.)
- Default strategy works reasonably well for many telescopes...
- But is not always optimal.



An example: AOFlagger

- So I wrote a gui to experiment with the settings
- Full list of settings is a “script” of actions
- Hard to understand for other astronomers!



ASTRON
Netherlands Institute for Radio Astronomy

An example: AOFlagger

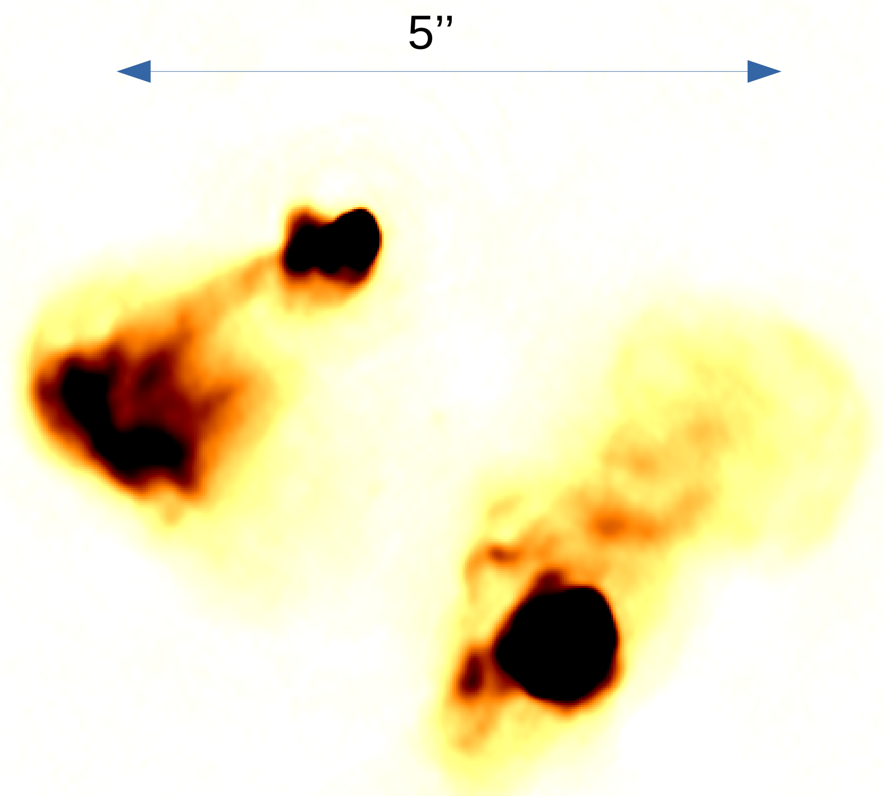
- Solution (or so I thought): A Python interface!
- Algorithms in C++, “glue” code in Python
- Far too slow :(
  - Need for very low-level managing and synchronization of memory
  - Synchronization of threads major issue
- Old interface is still used.
  - Example of difficulty of experimental, high-performance, yet user-friendly software

```
import aoflagger
import copy
import numpy

def flag(input):

  # Values below can be tweaked
  flag_polarizations = input.polarizations()
  flag_representations = [ aoflagger.ComplexRepresentation.AmplitudePart ]

  iteration_count = 3
  threshold_factor_step = 2.0
  base_threshold = 1.4

  # Use above values to calculate thresholds in each iteration
  r = range((iteration_count-1), 0, -1)
  threshold_factors = numpy.power(threshold_factor_step, r)

  inpPolarizations = input.polarizations()
  input.clear_mask()

  for polarization in flag_polarizations:

    data = input.convert_to_polarization(polarization)

    for representation in flag_representations:

      data = data.convert_to_complex(representation)
      original_image = copy.copy(data)

      for threshold_factor in threshold_factors:
```

An example: IDG with WSClean

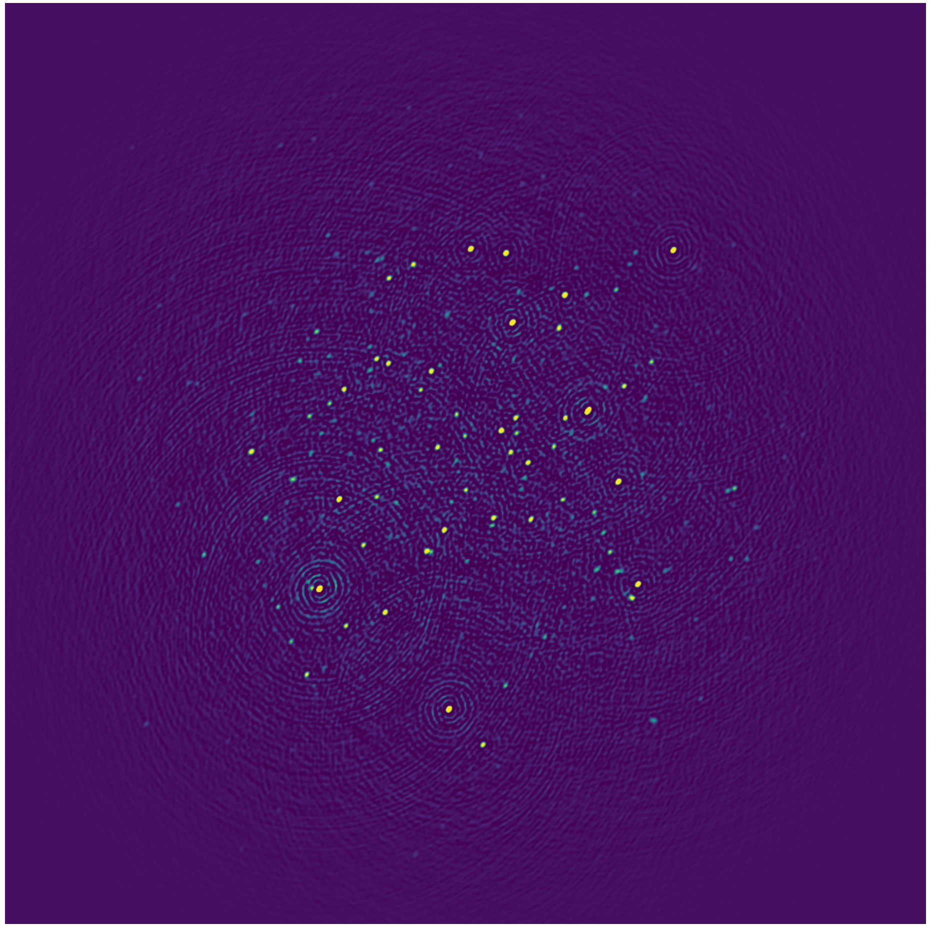
- WSClean is an imaging algorithm
- Transforms interferometric data into images
  - Inverse transform of instrument
  - Deconvolution
- Used for many telescopes
- About 40K lines of C++ code

- http://wsclean.sourceforge.net/



Best image available of 3C 196
(Made with WSClean from LOFAR data)

An example: IDG with WSClean

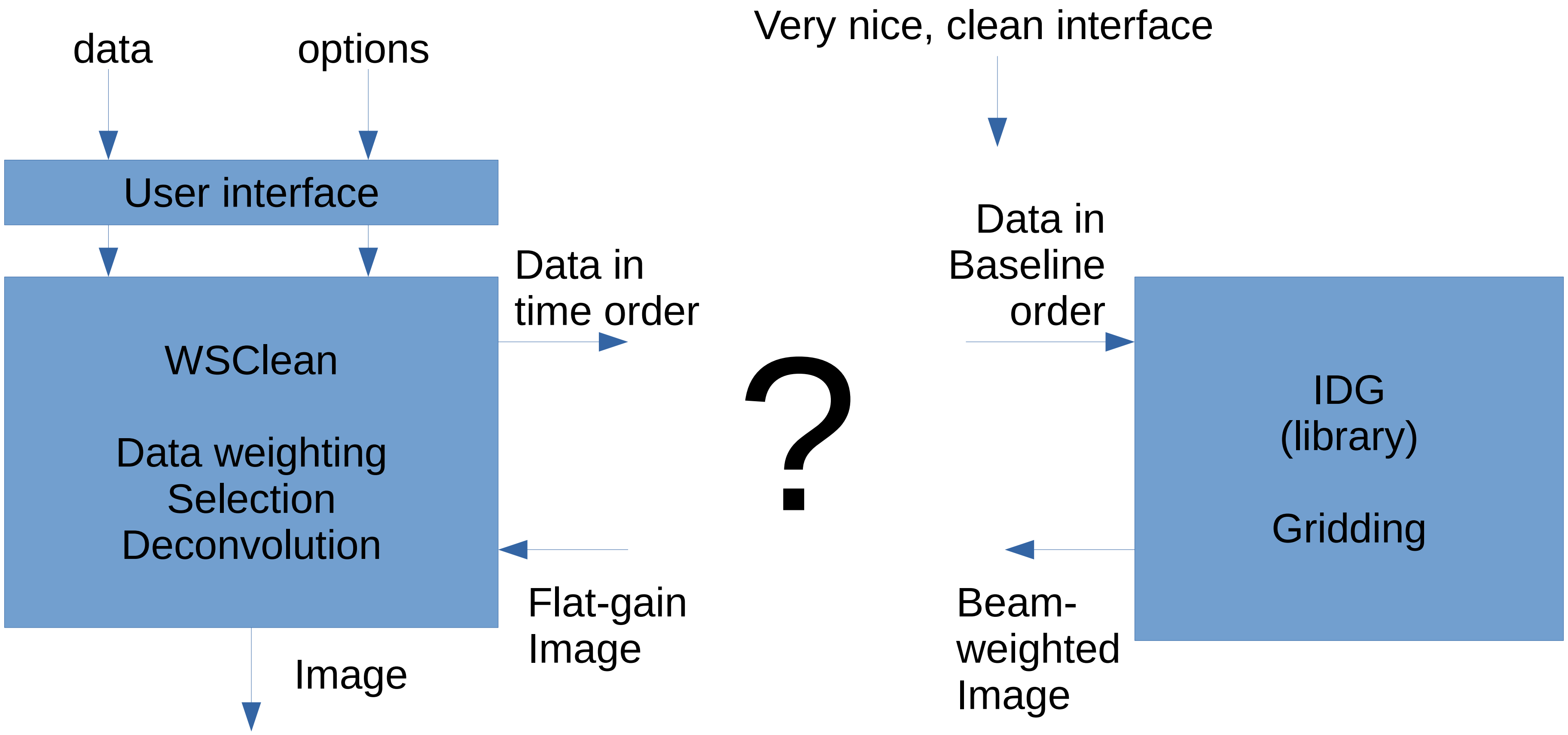


LOFAR beam applied during imaging stage
Producing “optimally weighted” image

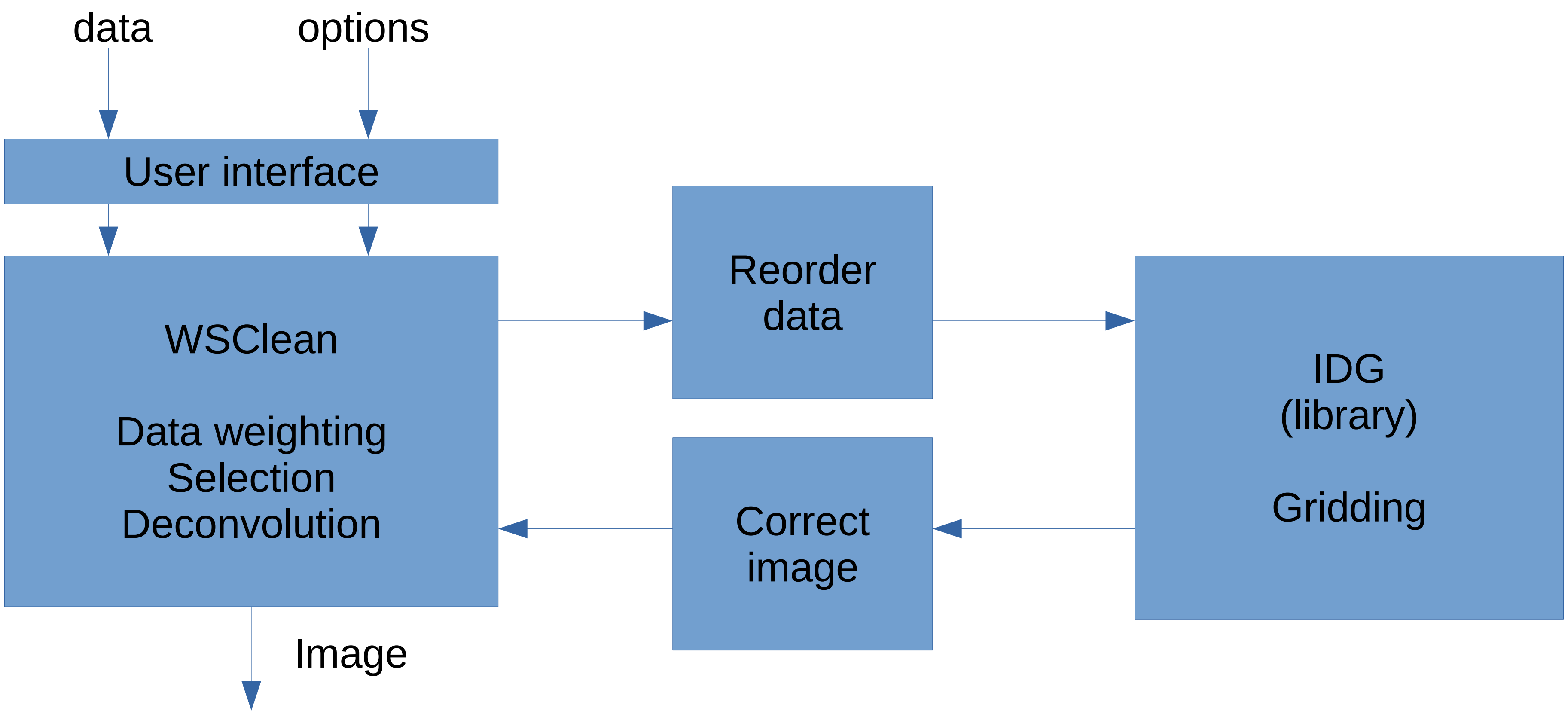
- Image domain gridding (IDG) is a new algorithm
  - Van der Tol, Veenboer, Offringa (2018)
  - See poster by Bas van der Tol
- Performs one step of the imaging process (gridding)
- Implemented as a library
- Allows better&faster imaging:
  - Can use GPUs
  - Allows simultaneous corrections for ionosphere and instrument response
  - Allows images of ~30k x 30k
- https://gitlab.com/astron-idg/

ASTRON
Netherlands Institute for Radio Astronomy

An example: IDG with WSClean

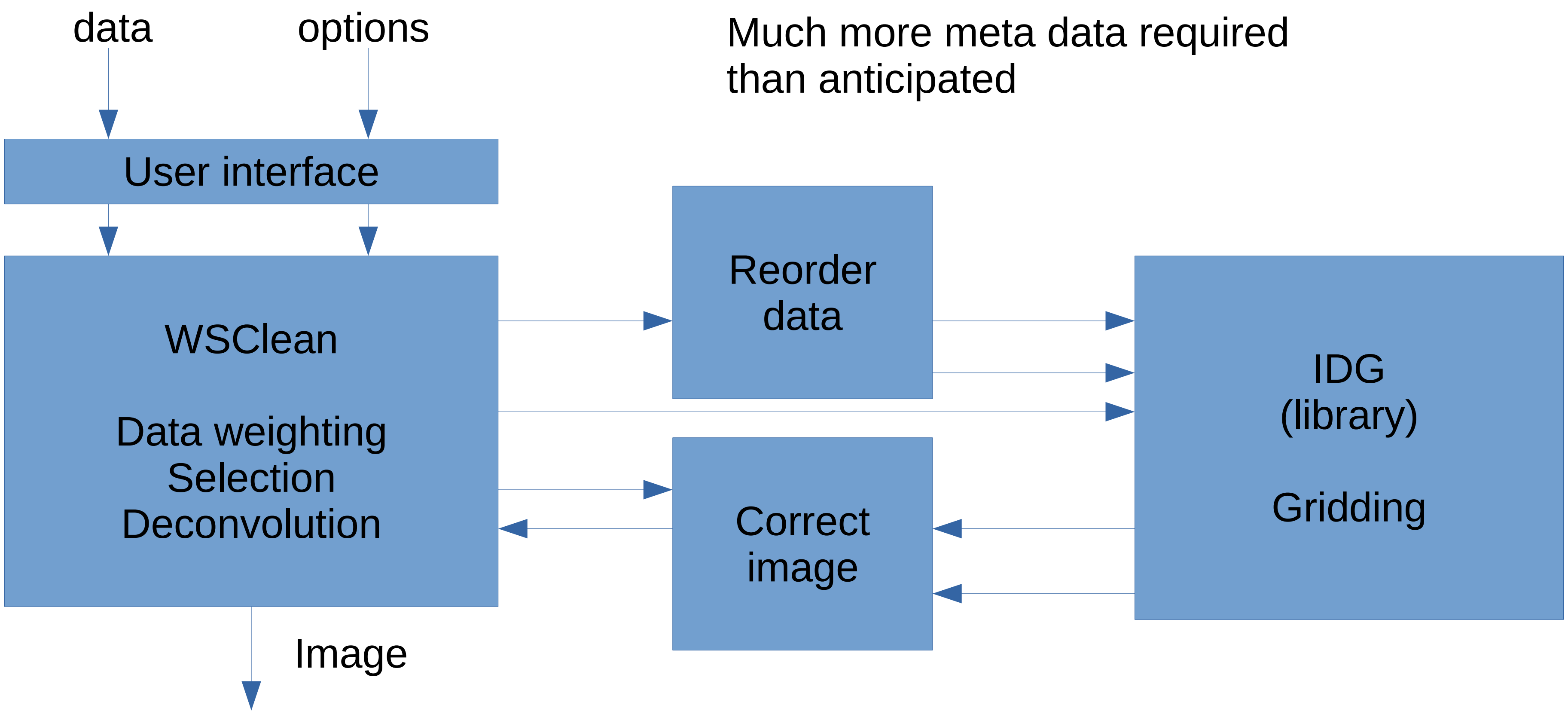


An example: IDG with WSClean



State of the software in 2017

An example: IDG with WSClean

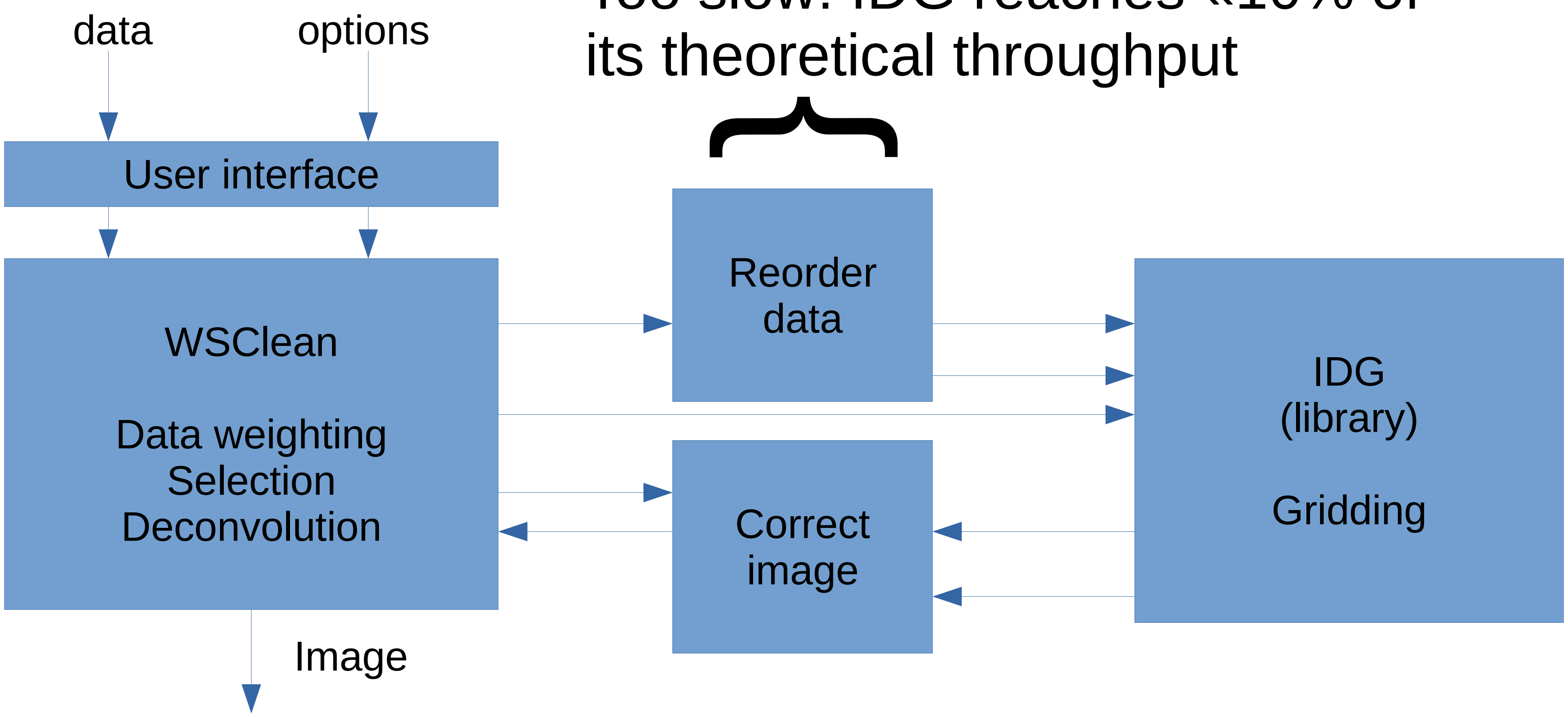


Much more meta data required than anticipated

State of the software in 2017

An example: IDG with WSClean

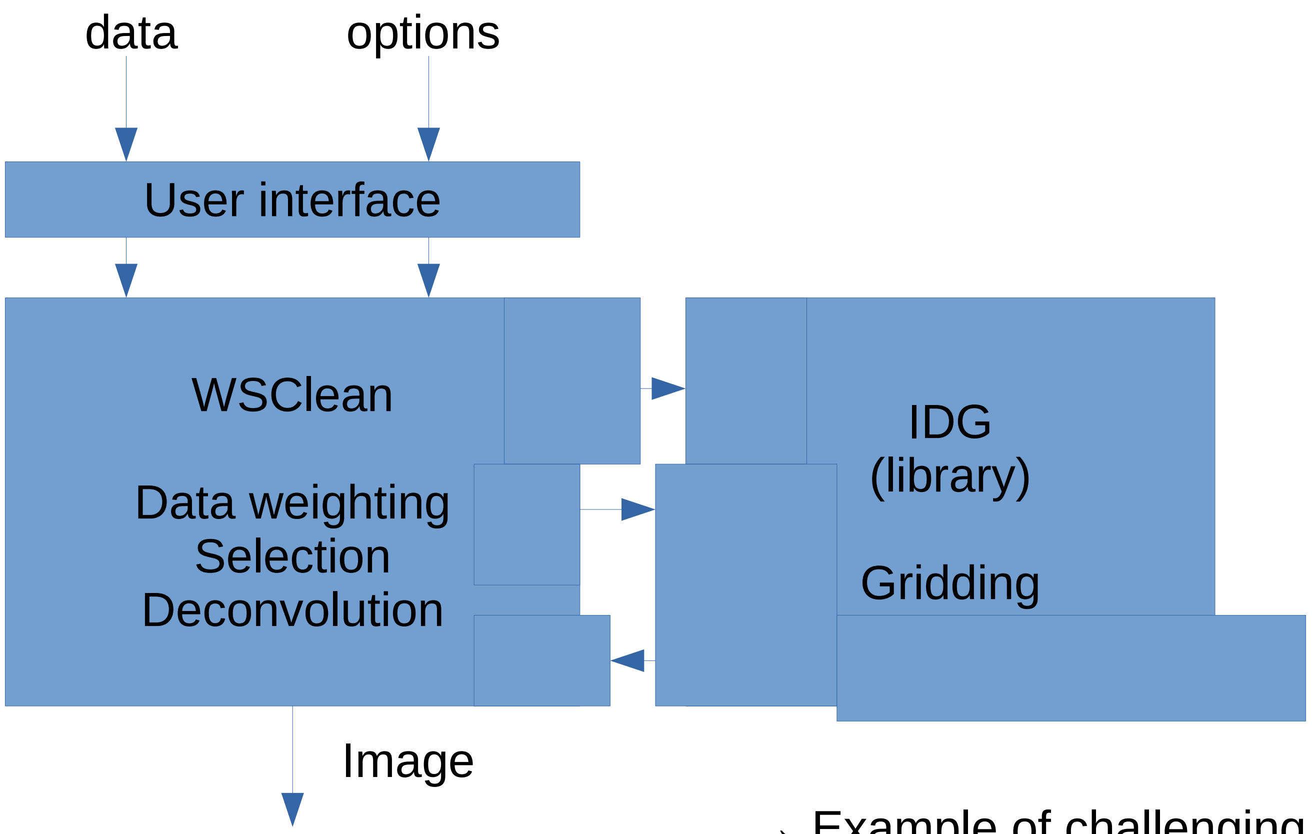
Too slow: IDG reaches «10% of its theoretical throughput



State of the software in 2017

ASTRON
Netherlands Institute for Radio Astronomy

An example: IDG with WSClean



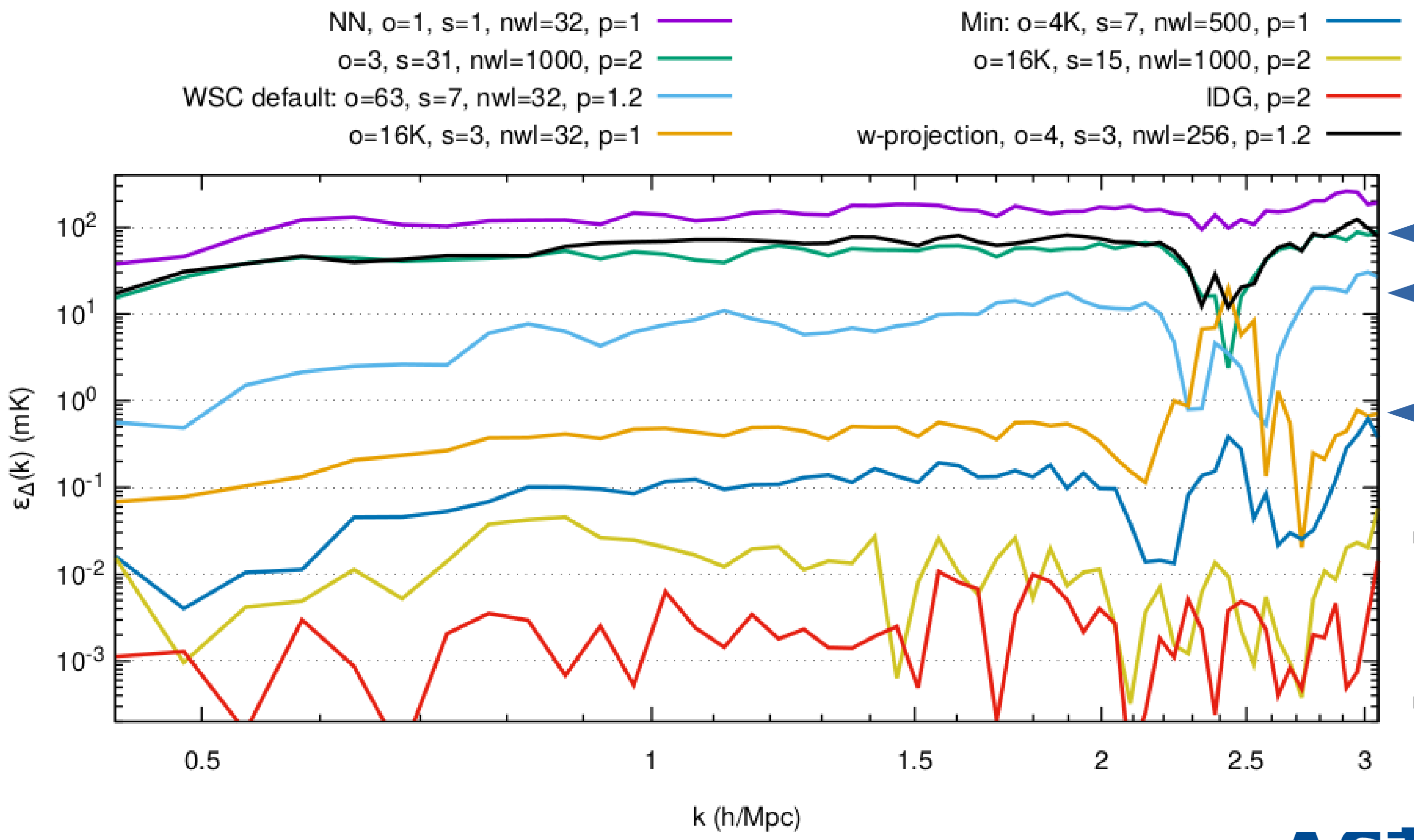
State of the software in 2019

→ Example of challenging modularity + high performance
→ Also hard to explain to management why it takes 2 years to combine two existing tools

An example: IDG with WSClean

Despite being a lot of work, IDG was shown to be the only gridder that is accurate enough for (LOFAR / SKA) Epoch of Reionization science:

Gridding error →
(in power spectrum)



Casa

WSClean default

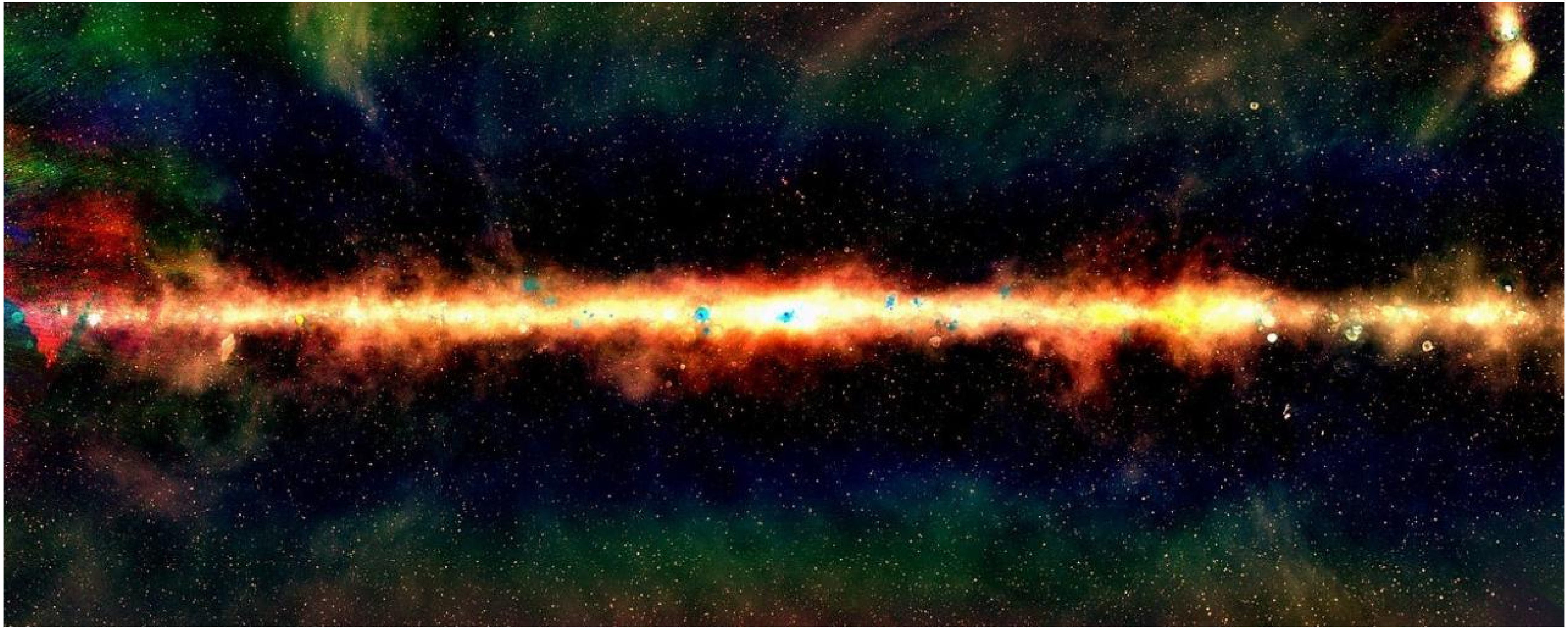
WSClean, more accurate + slower

IDG

(Offringa et al. 2019, A&A)

ASTRON
Netherlands Institute for Radio Astronomy

An example: MWA’s GLEAM survey



Hurley-Walker et al. 2016

Netherlands Institute for Radio Astronomy

An example: MWA's GLEAM survey

- Murchison Widefield Array (MWA)
- MWA Phase 1 has ~2' resolution
  - No "direction-dependent corrections" necessary
  - Easier (but not easy) to process compared to LOFAR data
- Pipeline steps:
  - RFI detection (using AOFlagger)
  - Averaging (Cotter)
  - Format conversion (to casacore Measurement Set format)
  - Calibration (+ transfer)
  - Imaging (WSClean)
  - Mosaicking (SWarp)
  - Source detection (Aegean)
  - Source matching + correction

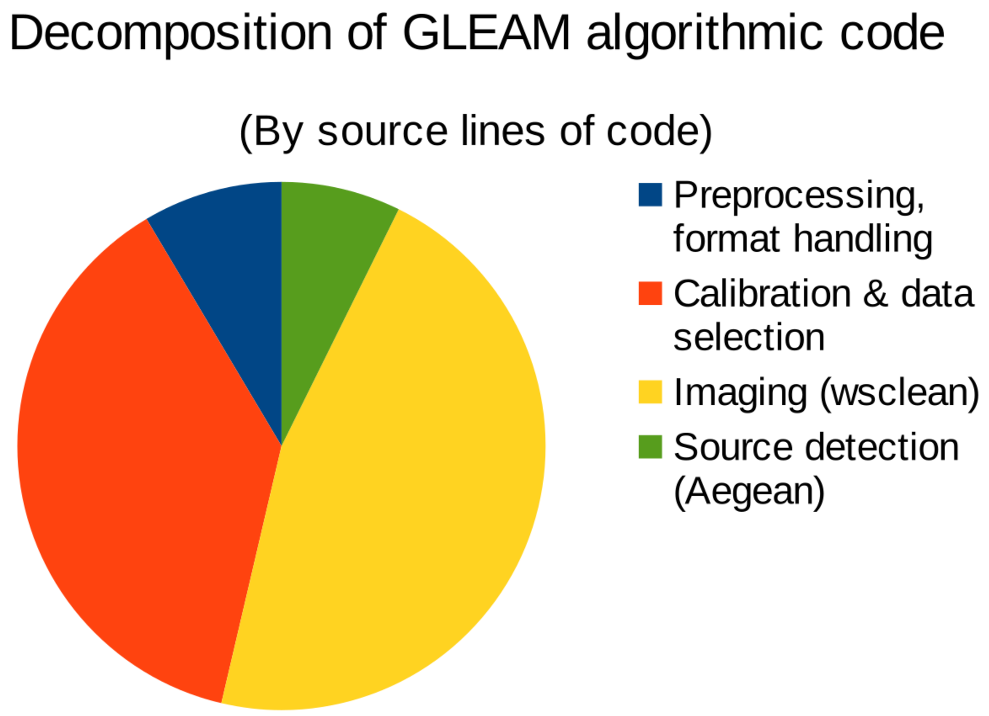
} Multiple times (selfcal) — applies to Calibration (+ transfer) and Imaging (WSClean)

An example: MWA's GLEAM survey

- Murchison Widefield Array (MWA)
- MWA Phase 1 has ~2' resolution
  - No "direction-dependent corrections" necessary
  - Easier (but not easy) to process compared to LOFAR data
- Pipeline steps:
  - RFI detection (using AOFlagger)
  - **Averaging (Cotter)**
  - **Format conversion (to casacore Measurement Set format)**
  - **Calibration (+ transfer)**
  - **Imaging (WSClean)**
  - Mosaicking (SWarp)
  - **Source detection (Aegean)**
  - Source matching + correction

} Multiple times (selfcal)

An example: MWA’s GLEAM survey



- Approximately 100k lines of code were written for the GLEAM survey
  (excludes monitoring, scheduling & control software)

- Constructive cost model says:
- – 100k lines of code
  – of “average complexity”
  – costs $2.5M USD

- That’s for a single science case
- ...and just *the final software*

Challenges of radio data processing

- Need to reuse software
  - We can't write & maintain 100 K lines of code for every science case / survey / …
  - But reuse requires modularity
- Challenge of high performance:
  - Harder to modularize: reusable interfaces often too slow
  - Harder to reuse code: needs to be written for (streaming) data in a specific order
  - Can't reorder or write intermediate products to disk
- Challenge of experimental code:
  - End up writing several different algorithms until the "correct" one is found
    - Maybe as much as 200-300 K lines of code were *actually written* to process the survey
  - Can't really "quickly prototype" algorithms, because they need to perform well to even test them

Summary

- Radio processing is challenging
- Making observatories produce Science-ready data is of high importance:
  - MUCH lower learning curve for astronomers
  - Processing experts at observatories, reuse of code
  - Science accessible to wider community
  - Increased science output!
- Bottomline:

  An increase in resources for the central development of processing algorithms (including maintenance + support!) will result in larger science output