

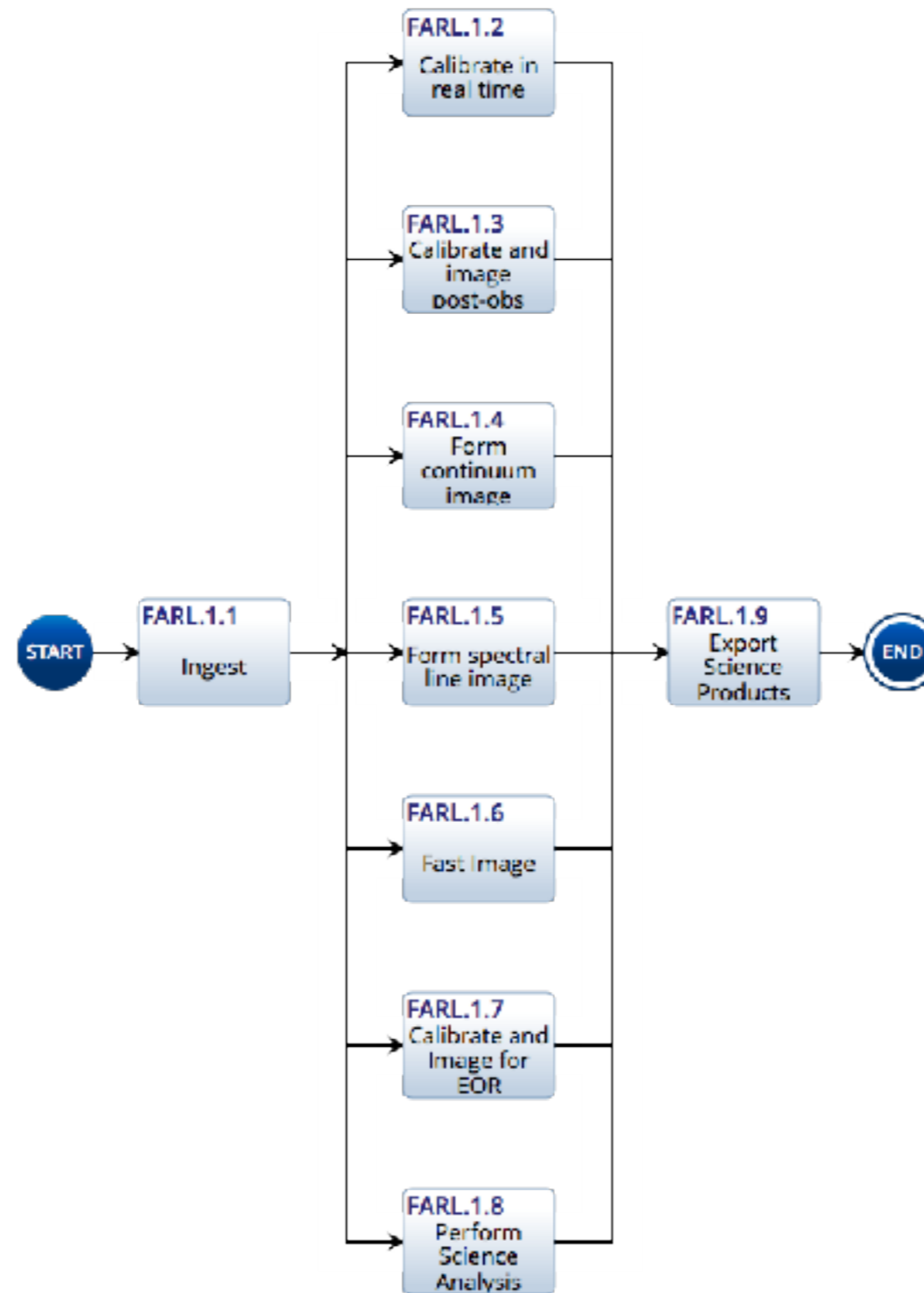
# Progress on ARL

Tim Cornwell  
with help from Peter Wortmann

# Current status

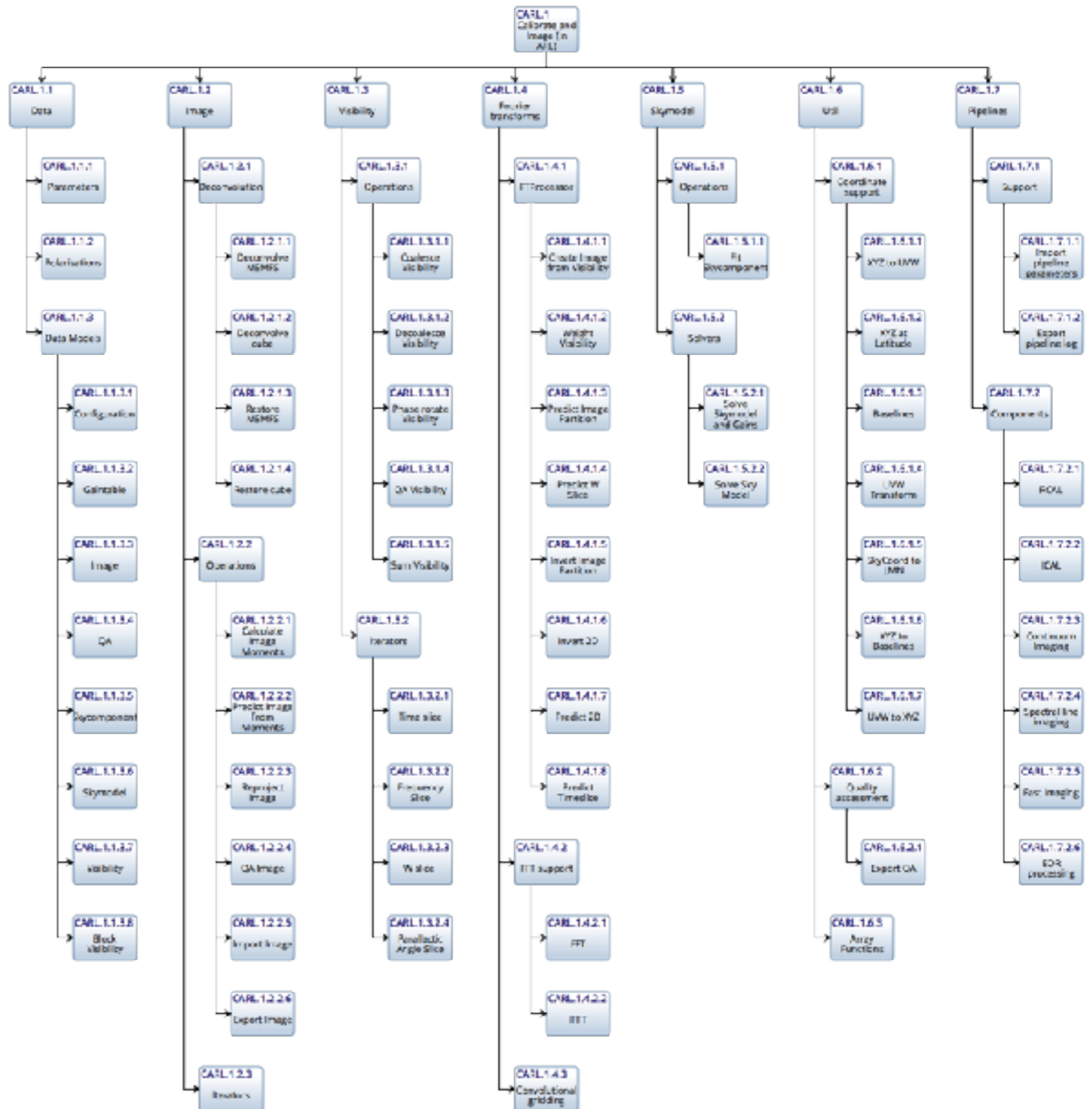
- Functional and product views complete
- Data models in place
- Synthesis framework complete
- Lots of Fourier transform approaches incorporated
- Calibration started
- <http://www.mrao.cam.ac.uk/projects/jenkins/algorithm-reference-library/docs/build/html/index.html>

# Functional view

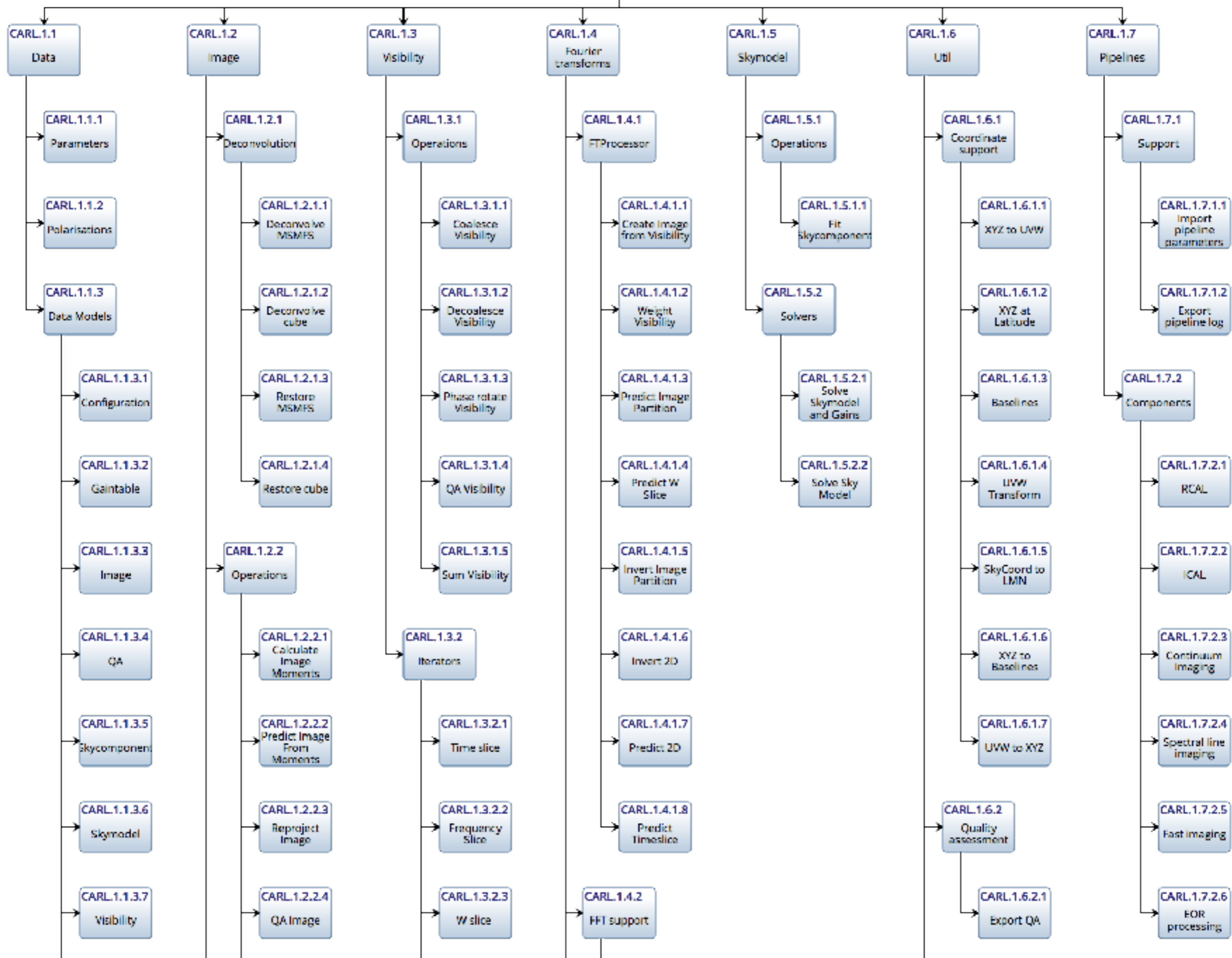


# Product tree

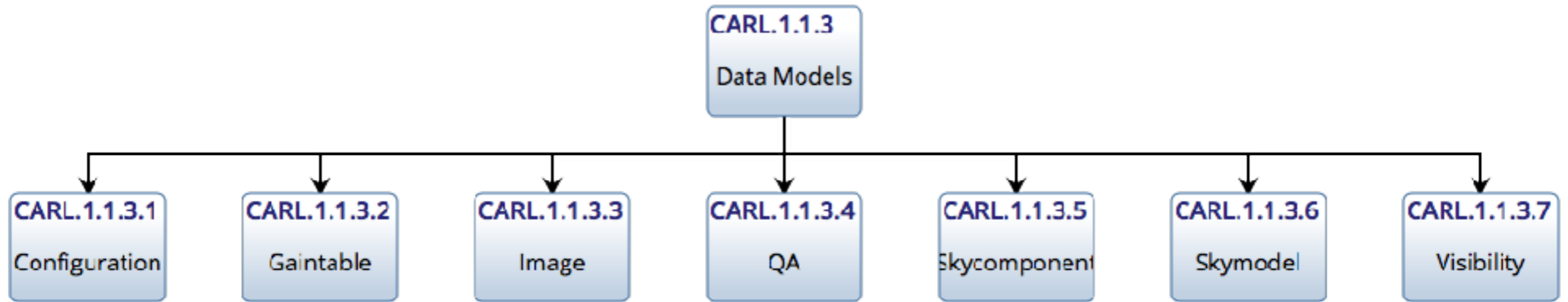
- Data models = 8 classes
- Components =  $O(70)$  state-free functions
- ~ 6000K LOC



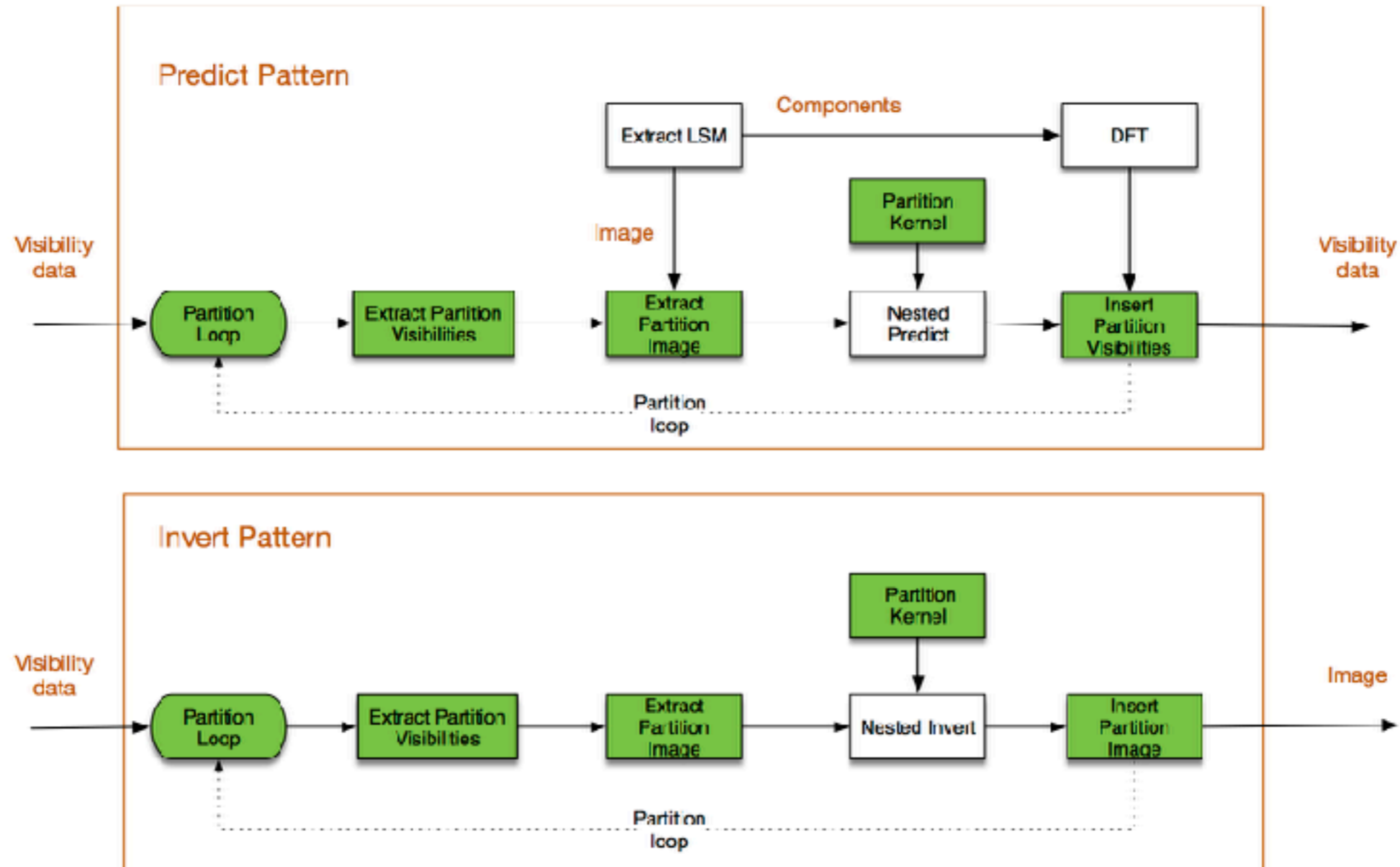
CARL 1  
Calibrate and Image (in ARL)



# Data models



# Synthesis framework



# Synthesis capabilities

- Fourier transforms
- 2d, facets, w-projection, w-slicing, time-slicing
- Everything is layered on top of 2d predict, invert
- BDA inside
- Uses python iterators to coordinate
- Will facilitate move to an execution framework



# Pipelines

- Should be able to implement full but scaled down pipelines in python or execution framework

# Product tree

- Provides an alternate view of how the Calibration and Imaging product tree should be structured
- Compare to SDP Product Tree

# Done so far this quarter

- BDA
- Move to atomic format: only one freq, pol per row
- w-slicing
- Refactoring of synthesis using python generators
- Model for LOW sky (S3)
- Model for LOW station beam (OSKAR)

# Before May 2017

- Aim for discussion on data formats
- Calibration
  - Application of gaintable
  - Solution of gaintable using stefcal or other
  - Peeling
- Conclusion on data formats

# Visibility formats

- Calibration solution works naturally with block-oriented rows:
  - $\text{Vis}(\text{rows}, \text{antenna1}, \text{antenna2}, \text{frequency}, \text{polarisation})$
- Imaging works naturally with simpler rows: frequency, polarisation
  - $\text{Vis}(\text{rows})$
- BDA causes fragmentation of Imaging format
- One format is possible but lots of data re-arrangement needed

# May / June...

- MSMFS
- Execution framework
- Improve tests and coverage
- Release
- Investigate some pressing questions about requirements?

# ARL performance

- Has to be fast enough to test some interesting cases
- e.g. BDA for LOW
- Numba - Peter testing for gridding
- Pympl - currently in place
- Execution frameworks e.g. Dask

# Support

- Peter W helping with coordinates, Python niceties, debugging some code, Github/Jenkins wrangling
- Larger platform for documentation builds
- Access to cluster for tests