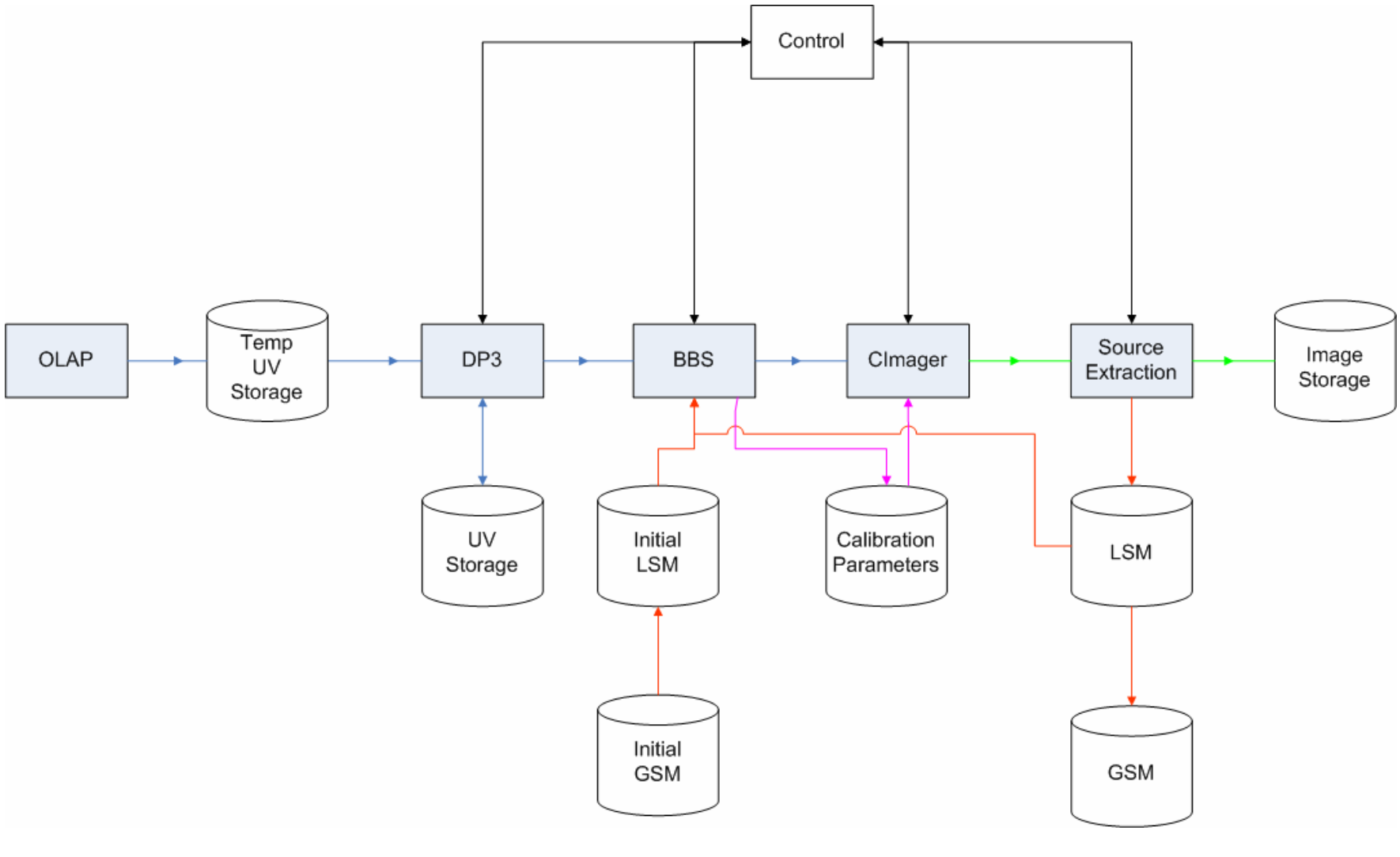
Standard Imaging Pipeline for MSSS



- Real time processing!
  - Distributed storage
    - Uv-data
    - Parameters?
    - GSM?
  - Distributed computing
    - BBS (+ Global Solver)
    - CImager
    - Source Finding (?; Image cubes are small)
  - Bandwidth for Beams
    - Process multiple observations simultaneously
- Control: SAS / MAC vs. scripting
- Calibration Strategy / Approach to be specified
- Performance of components needs to be improved

- Do we need storage in between?
  - 3 separate observations that need to be combined
  - Can we go through the chain a second time (Major Cycle)?
  - Should we first collect all 3 observations and process them together?

- Bandpass correction
- Flagging
- Calibration for clock drifts?
- Calibration for Ionosphere?
- Compression

- Performance
  - Distributed storage needed
  - Improve speed, I/O

- Distributed
- Global Solver needed for S/N
- Sky model
  - Point sources
  - Other models?
  - constant spectral index
  - Polarization?
- Ionosphere
  - What kind of ionospheric modeling and correction?
- Beam model
  - Current models to be validated
- Improve performance

- Distributed
- Application of Image Plane Effects
  - Facet Corrected Imaging
    - How many facets?
  - Convolutional gridding (Bhatnagar)
- Produces
  - Dirty image
    - Spectral cube
    - Continuum image
  - PSF?

Source Finding

- Distributed?
  - Final version must be distributed
  - Version for MSSS may be single core
- Source finding in multiple freq planes
- Performance / speed?
  - PyBDSM
  - Amsterdam real-time Source Finding

- To be initialized with inter- / extrapolated VLSS / WENSS data
- Eventually contains 2.5 million sources
- Performance
  - OR_GSM
  - Amsterdam database GSM

- What do we store?
  - OLAP uv-data: more than 7 Pbyte: infeasible
  - Post DPPP uv-data
  - Images
  - GSM
  - Calibration parameters?