

# Data Loading and Ingest at DM AHM 2015-02-11

## Date

06 Feb 2015

## Attendees

- Mario Juric, Kian-Tat Lim, Gregory Dubois-Felsmann, Unknown User (xiuqin), Trey Roby, Tatiana Goldina, Jim Bosch, Simon Krughoff, Jeff Kantor, Unknown User (danielw), Fritz Mueller, Fabrice Jammes, Jacek Becla, plus others, I didn't capture the names in my notes, apologies.

## Discussion items

### pipeline --> database

related reading material: <https://dev.lsstcorp.org/trac/wiki/db/Qserv/DataLoading>

#### SDQA / loading coordination

- will do early data processing on a very small data set (~1%), before DRP starts and will run SDQA on it
- this will catch most problems, but not 100%
- during DRP, dump intermediate results to internal DRP db. Also, perhaps dump a subset of data (say 5%) to separate database and do SDQA there (that is not the final DRP production db). Then after SDQA done, load into final production DR
  - expecting the load will take ~1-2 days, not weeks
  - intermediate products are owned by Middleware, e.g., Qserv should not delete them after loading is done
- It is safe to assume object positions won't be affected by SDQA. In case of problems with astrometric calibration, will issue erratum, and fix in next DR
- Yes, might need to fix some columns - this is the same complexity as adding new columns, flags etc, which we already planned to do

#### Tiling / processing order

- buckets / packets of sky. Few 100s sqdeg (so ~15TB or so)
- nothing global if we make sky tiles large enough

#### DRP db

- stores all bookkeeping (provenance, what run what did not, etc)
- intermediate data products
- a subset of data (what we need by DRP), eg footprints of objects
- might need spatial engine
- will run QA here
- there is desire to keep all files that we will need to ingest to qserv forever
  - this is not in the current storage model, need to add

#### L3 loading

- initial implementation: ok to lock entire database
- if that proves to be too limiting, will switch to more complicated model and will do per-table locking

### pipeline --> images

- just change the location of images
- get all metadata / provenance from DRP internal db