VERA C. RUBIN OBSERVATORY

Prompt Processing Status

Kian-Tat Lim

DMTN-219

Playbook for Prototype

Quick Live Demo

Future Task List

Future Task List

- Actually build out the middleware interface with a real calibration repo. Ensure that performance is adequate. Work with Middleware to optimize anything that isn't.
- Choose a messaging infrastructure. Kafka seems reasonable for `next_visit`: it already exists, performance is less critical, it is reliable, it is well-understood. Apache Camel might be a candidate for connecting Kafka with a webhook for invoking the Prompt Processing framework.
- Implement DMTN-143 copy to object store. KTL will work on this with TonyJ.
- Implement object store notifications to Prompt Processing. The alternatives here common to both MinIO and Ceph are Kafka and AMQP. While I have some concerns about Kafka in this low-latency use case, it seems the simplest to start with.

Productionize and Optimize

- Move from GCP to SLAC K8s. Set up an ingress; investigate ways to get affinity.
- If near-zero edit-to-execute latency is required, at the cost of reliability, use a shared filesystem for the stack code.
- If full-focal-plane visit-wide processing is required, at the cost of reliability and latency:
  - Use a shared Butler repo for intermediate results
  - Manually split pipelines into multiple steps
  - Poll Butler repo or possibly use object store notifications again
  - Using BPS has virtually no advantages over using OCPS for the same thing; preloading is basically impossible