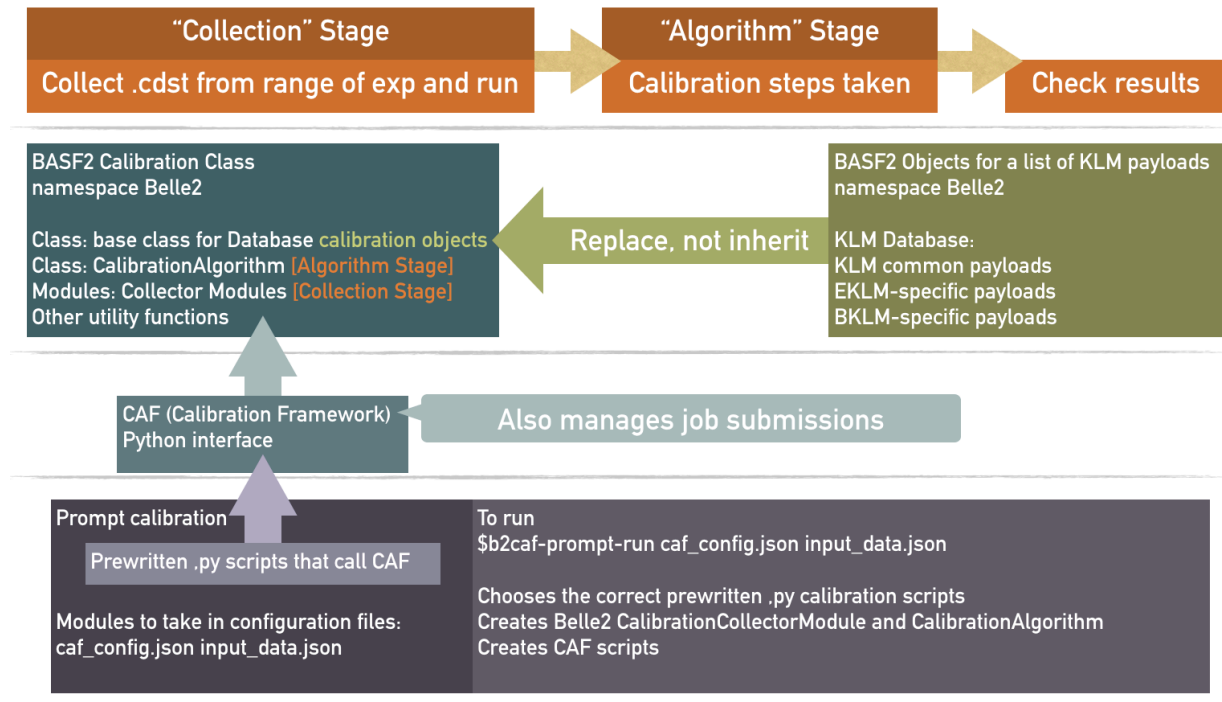


# KLM calibration meeting March24 notes

## Overview

1. [Note: these .cdst already produced by others, we only work on calibration]



2. Also to develop basf2 myself:
  - /cvmfs/belle.cern.ch/tools/b2setup; which creates /development dir locally
  - Documentation (separated from precompiled releases) [https://b2-master.belle2.org/software/development/sphinx/build/tools\\_doc/index-01-tools.html](https://b2-master.belle2.org/software/development/sphinx/build/tools_doc/index-01-tools.html) on the master branch
  - ^^ The above documents the project development in progress, will only update to the regular documentation page when it's frozen release

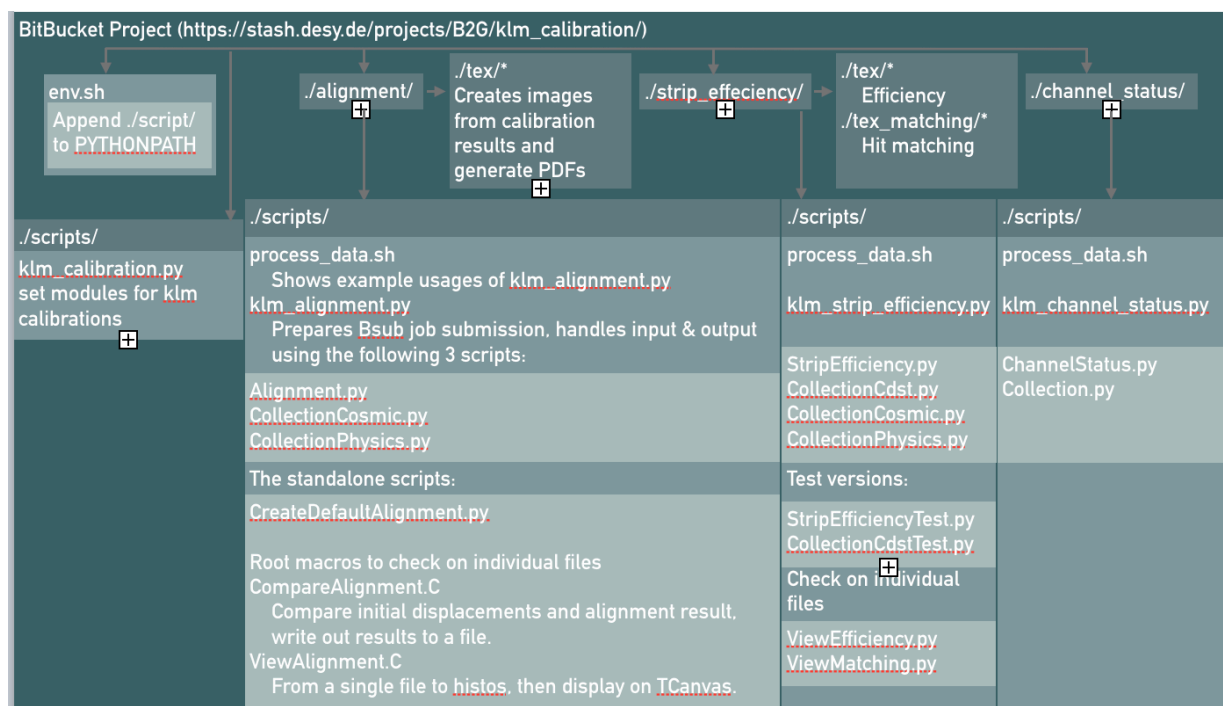
## Analysis note klm\_calibration:

1. <https://stash.desy.de/projects/B2D/repos/belle2-note-te-2020-010/browse>
2. Explaining the offline calibration in the /B2G/klm\_calibration project and also prompt calibration
3. Serves to explain both for users of basf2 klm calibration as well as people who will develop, as the populations mostly overlap
4. Will make PR to connect the edits that I have written

## Seperate klm\_calibration project:

1. [https://stash.desy.de/projects/B2G/repos/klm\\_calibration/browse](https://stash.desy.de/projects/B2G/repos/klm_calibration/browse)

2. This is the offline calibration that already worked and also had already been used for calibration
3. Mostly on offline calibration, eventual goal is to integrate this into automated calibration of basf2
4. Some example usage in the project:
  - `./klm_channel_status.py --experiment=8 --first_run=0 --last_run=3480 --global_tag=data_reprocessing_prompt --use_physics --use_beam --use_cosmic`
  - `./klm_channel_status.py --run_algorithm --collection_results=e8r0r3480_physics_beam_cosmic`
5. Below shows the overall structure, this also includes the macro to generate plots to check the calibration results



## Ongoing development on feature branches to basf2

1. To catch up with KLM related issues, check related JIRA tickets on [agira.desy.de](https://agira.desy.de), which are generally linked to feature branches.
2. This include many aspects of feature patches, basically anything related to KLM calibration, and not limited to just prompt.
3. Developing scripts to include in calibration/scripts/prompt/calibrations so that people can run b2caf-prompt-run, so far strip efficiency and alignment implemented.
  - <https://stash.desy.de/projects/B2/repos/software/pull-requests/5682/overview>
  - <https://stash.desy.de/projects/B2/repos/software/pull-requests/5799/overview>
4. Giacomo's answer to my questions can be found on: <https://stash.desy.de/>

[projects/B2/repos/software/pull-requests/5818/overview](https://stash.desy.de/projects/B2/repos/software/pull-requests/5818/overview), which lists a few projects

- This includes explaining standard cDST(a lot of data objects) and raw+tracking\_dataobjects versions
- The standard cDST will always be there for the previous processings
- Every new processing including all the data we have so far, and will have raw cDST standing proc11
- Phase II data (Experiment 3)discarded due to detectors not in good shape etc
- Phase I cosmic, Phase 2 beam line, Phase 3 physics runs

## Other information

1. No need for Confluence page, instead use sphinx documentation <https://stash.desy.de/projects/B2/repos/software/browse/klm/doc/index.rst> [Doxygen: C++, sphinx: Python]
2. AirFlow: Used by prompt to for automated calibration.
3. Bamboo is a build system, branches on PRs (plan, job, task related branches) linked on PR pages, no direct interactions needed.
4. Development directories to check on
  - klm/data/BKLM.xml
  - klm/bklm/geometry
  - Klm/bklm/dbobjects/