



Automated Data Calibration and Processing Procedures at the Belle II Experiment

(Presenter)

David Dossett

University of Melbourne
David.dossett@unimelb.edu.au

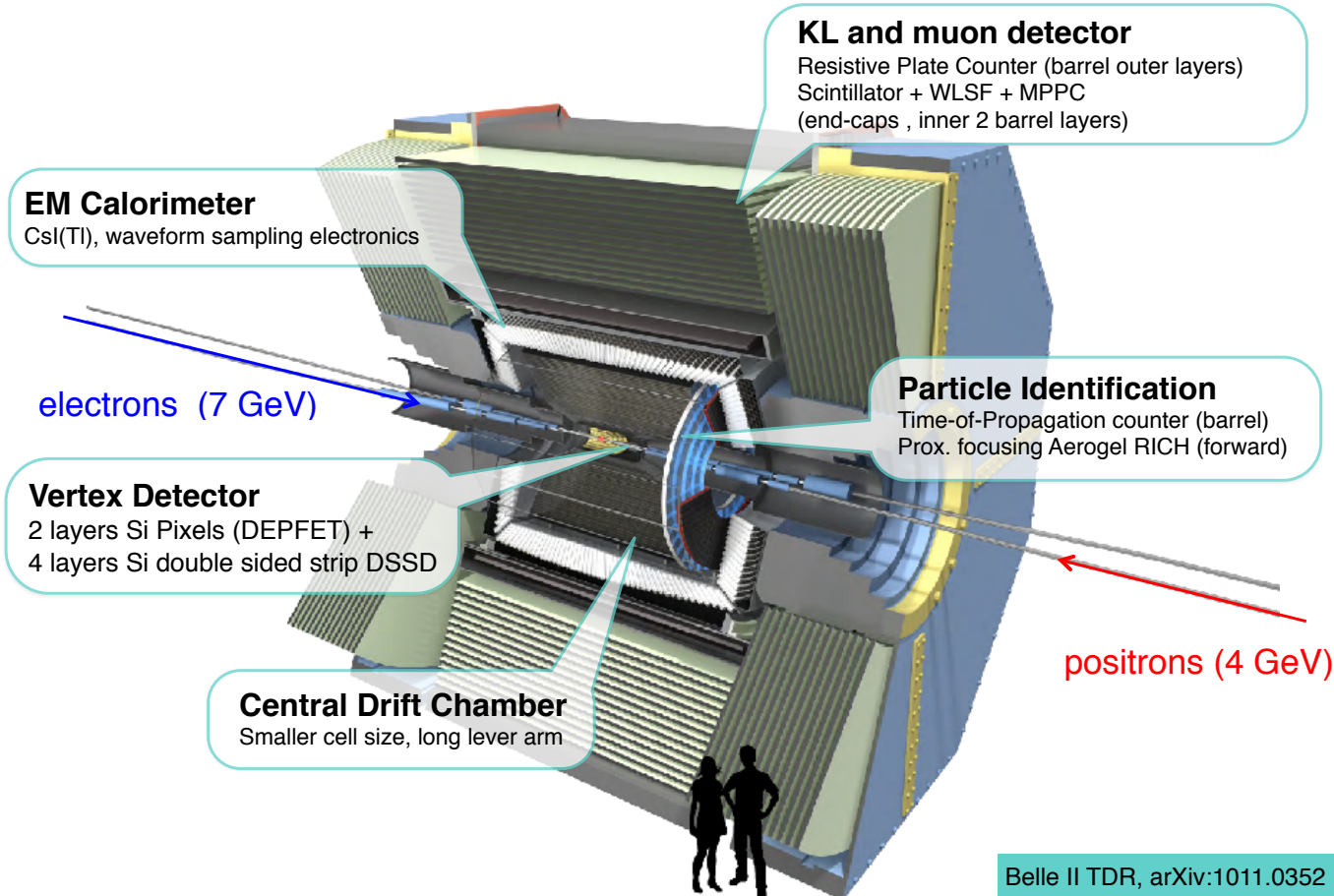
Martin Sevier

University of Melbourne

Francis Pham

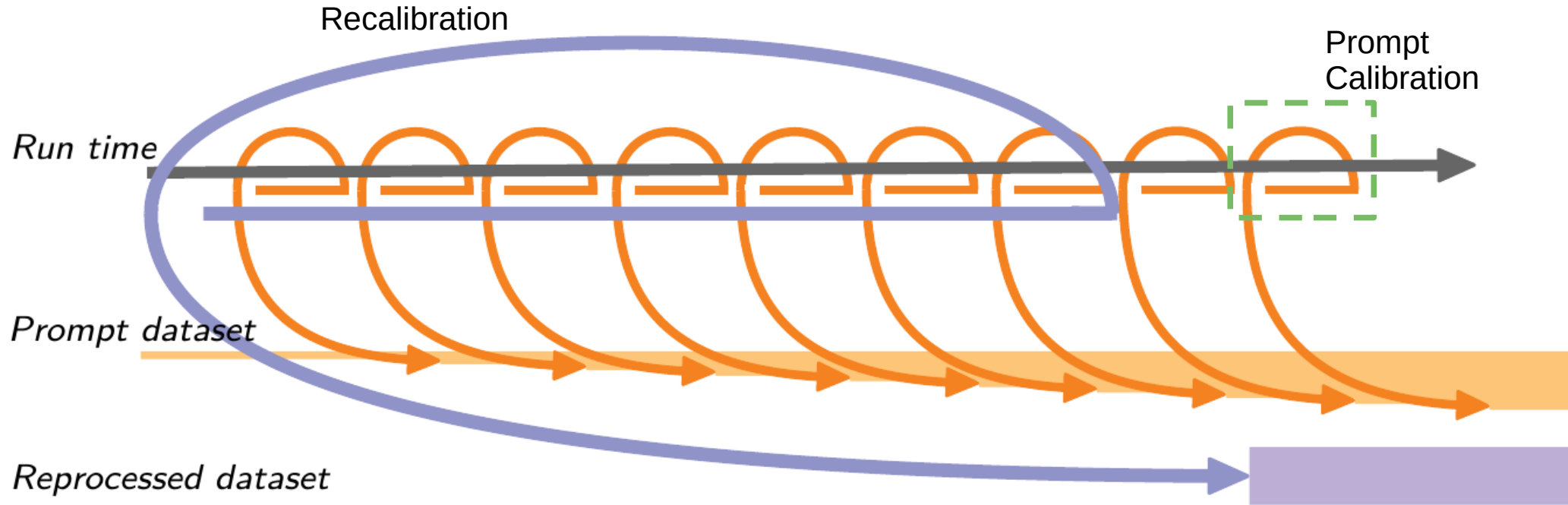
University of Melbourne

The Belle II Experiment



Belle II TDR, arXiv:1011.0352

Calibration Strategy



- Calibration is done on HLT skims of the raw data
- See Markus Prim's previous talk: <https://indico.jlab.org/event/459/contributions/11452>



Automating Calibration

Apache Airflow is software used to create Python workflows using Directed Acyclic Graphs (DAGs)

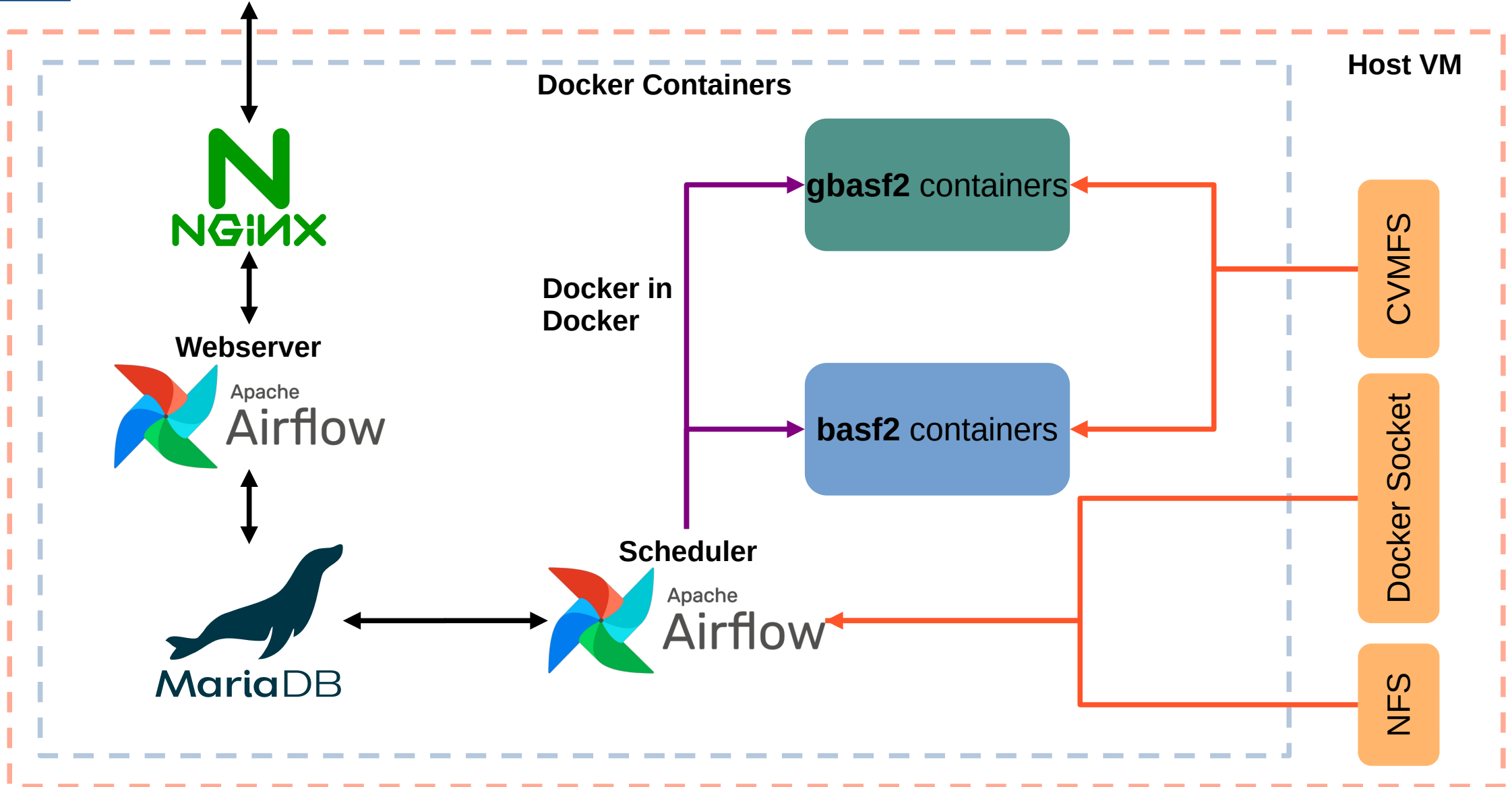


We created the **b2cal** Python package that defines DAGs to create Jira issues, automate prompt calibration processes at BNL, check payloads, and add webpages for calibration experts to change default job settings + sign-off

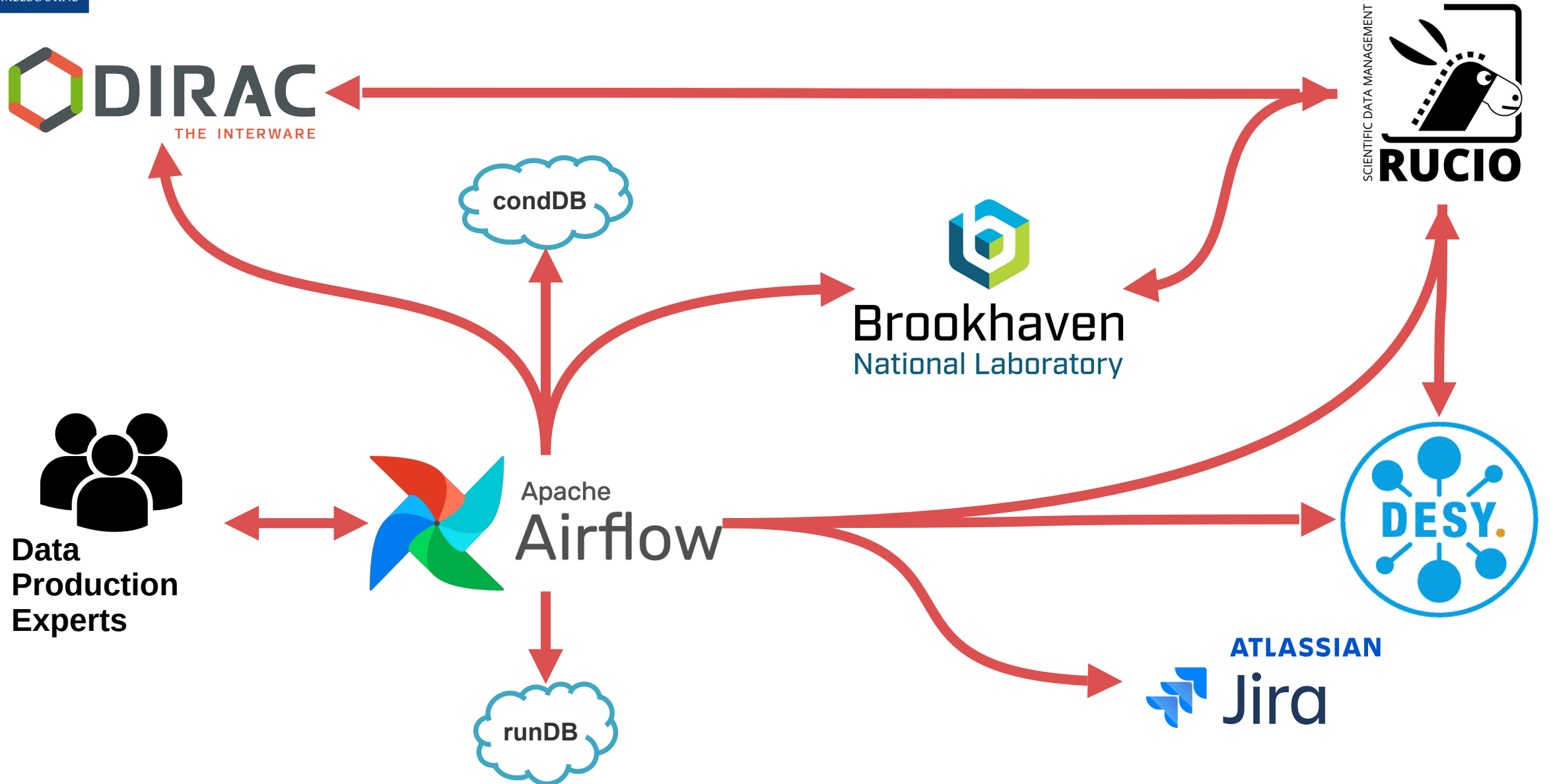
Now expanded to include:

- Larger recalibration process at DESY
- Organising the run dependent MC calibration payloads

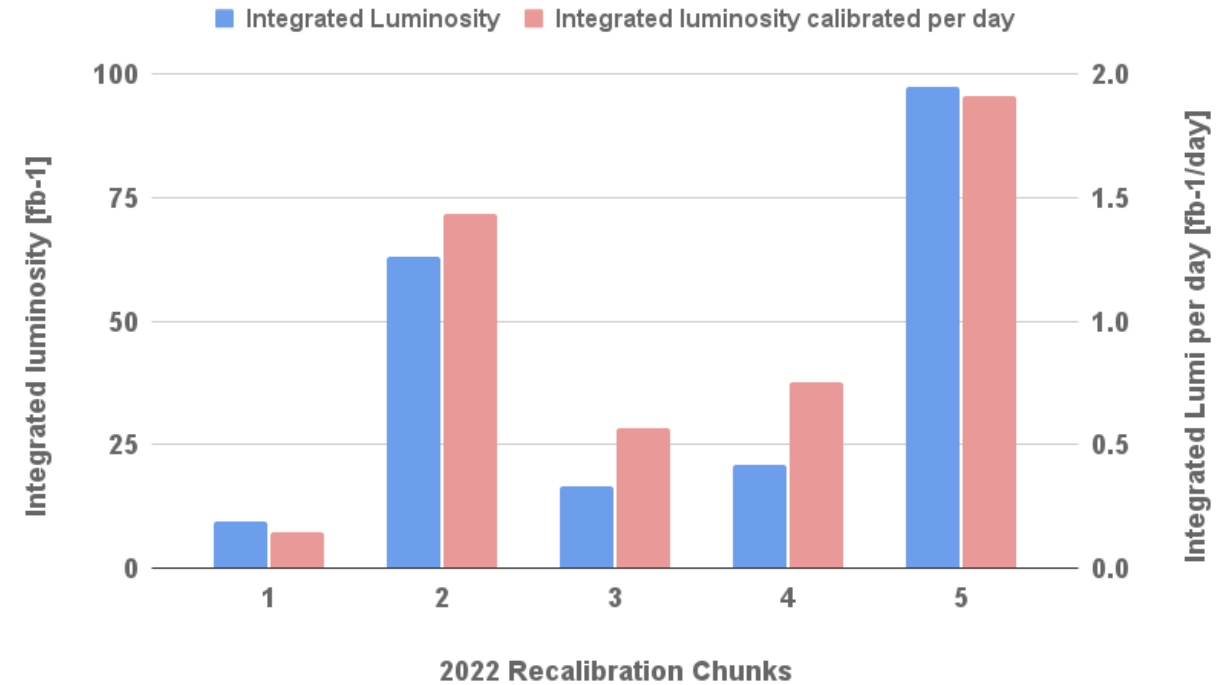
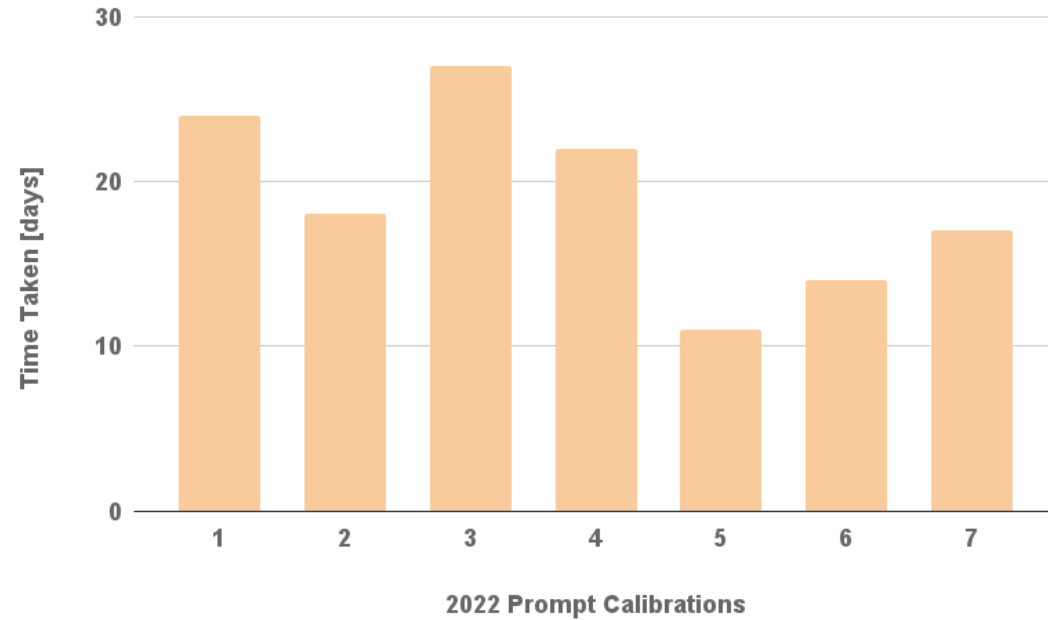
Server Setup (DESY)



Some Connections



2022 Automated Calibration

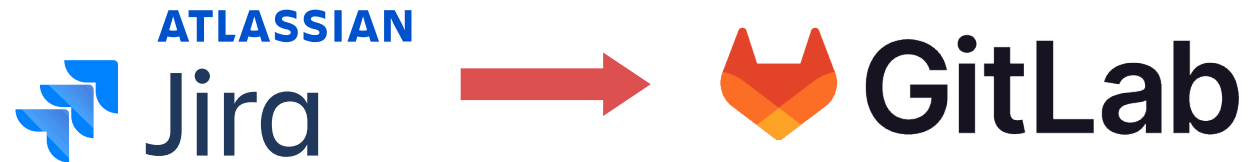


- 2-3 weeks prompt calibration, 50,000 HTCondor jobs, 15TB of calibration DST files produced, using 30 kHEPSpec06 days
- Recalibration chunk size is limited by tape → disk staging space at DESY

Service Changes

At Belle II we use Jira + Bitbucket to track code changes but also many types of Data Production tasks

Belle II is migrating to GitLab, still hosted at DESY, for both the repository hosting and issue tracking

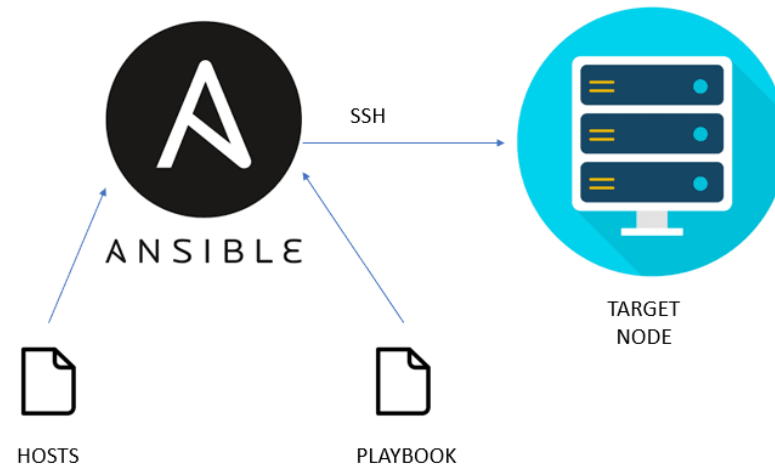


We are in the process of switching the calibration over to the GitLab REST api for calibration issue tracking

Taking the opportunity to merge our Docker Compose code and Airflow code into one repository

Refactoring Setup

Ansible is an automation tool that lets you write infrastructure as code



Usually you use it to set up multiple hosts via SSH, but you can also set up a local machine like your laptop/VM

Created Ansible playbooks to set up our Docker Compose environment and build/run/restore the b2cal services



Current Work

With the infrastructure updated we're moving to Airflow v2

- Newer package versions
- Better UI in admin webpages
- Faster scheduler → tasks start earlier

Giving calibration experts more ability to investigate calibration job failure from the web interface

Future Plans

Automating the submission of the reconstruction jobs to the grid after prompt calibration

Recalibration is a large procedure that may be difficult to keep local at a single centre in the future. Investigating feasibility of submitting calibration processes to a few dedicated sites via our DIRAC system

Dedicated manpower for calibration is a persistent issue. Training of new people (in different timezones) to respond quickly to any problems during prompt calibration is a top priority before data taking restarts



Backup



Refactoring the Infrastructure

Originally we used two Bitbucket git repositories to set up the automated calibration system:

- **b2cal**: Airflow plugin + database models + Flask routes
- **docker-compose**: Builds and runs docker containers for NginX, basf2, gbasf2, Airflow, and MariaDB

The idea was to separate the infrastructure setup from the Python package that is actually running calibration logic

In practice they are tightly coupled and never used separately, so we're merging them together as we move to GitLab



Adding Run Dependent MC

Run dependent MC requires gathering a large number of correct calibration payloads from many experts → a human organisation problem

We have extended b2cal to automatically create/close the tickets, assign experts, and to check + merge the payloads according to the calibration rules

Helps to minimise human error and confusion

| | | |
|----------------------------------|------------|--------|
| GT preparation MC15rd_a bucket29 | BIIDP-5670 | Closed |
| GT preparation MC15rd_a bucket30 | BIIDP-5661 | Closed |
| GT preparation MC15rd_a bucket31 | BIIDP-5671 | Closed |
| GT preparation MC14rd bucket32 | BIIDP-5861 | Closed |
| GT preparation MC15rd_a bucket32 | BIIDP-5998 | Closed |
| GT preparation MC15rd_a bucket33 | BIIDP-6008 | Closed |
| GT preparation MC15rd_a bucket34 | BIIDP-6018 | Closed |
| GT preparation MC15rd_a bucket35 | BIIDP-6028 | Closed |
| GT preparation MC15rd_a bucket36 | BIIDP-6038 | Closed |

The next step is to automate the creation of the run dependent MC grid submission configuration files