Analysis at Belle II

Michel Hernandez Villanueva
DESY

Analysis in the wider HEP/nuclear community
Jun 16, 2021

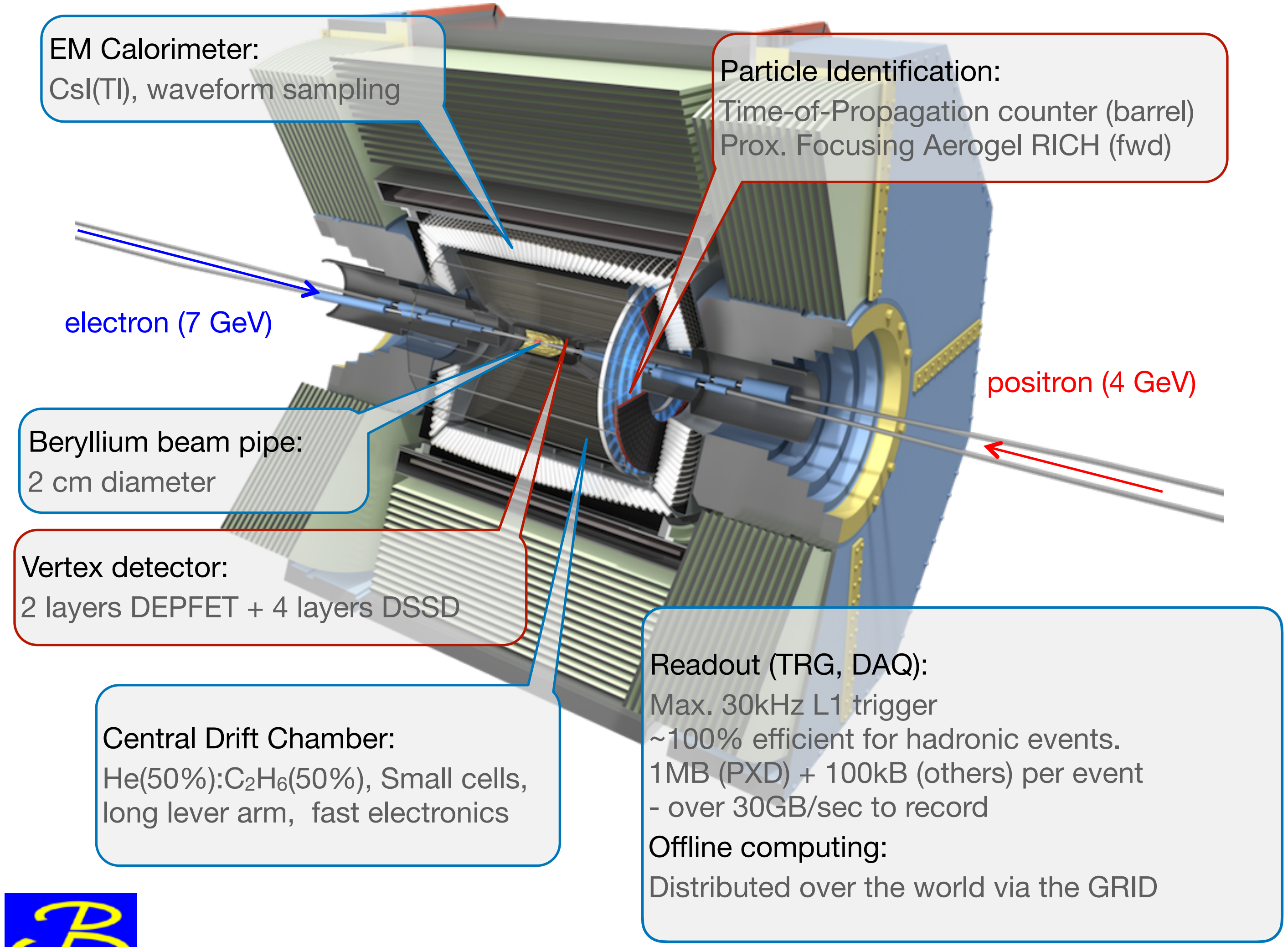
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

Belle II

DESY.

The Belle II Experiment

1100 members, 123 institutions, 26 countries



EM Calorimeter:
CsI(Tl), waveform sampling

Particle Identification:
Time-of-Propagation counter (barrel)
Prox. Focusing Aerogel RICH (fwd)

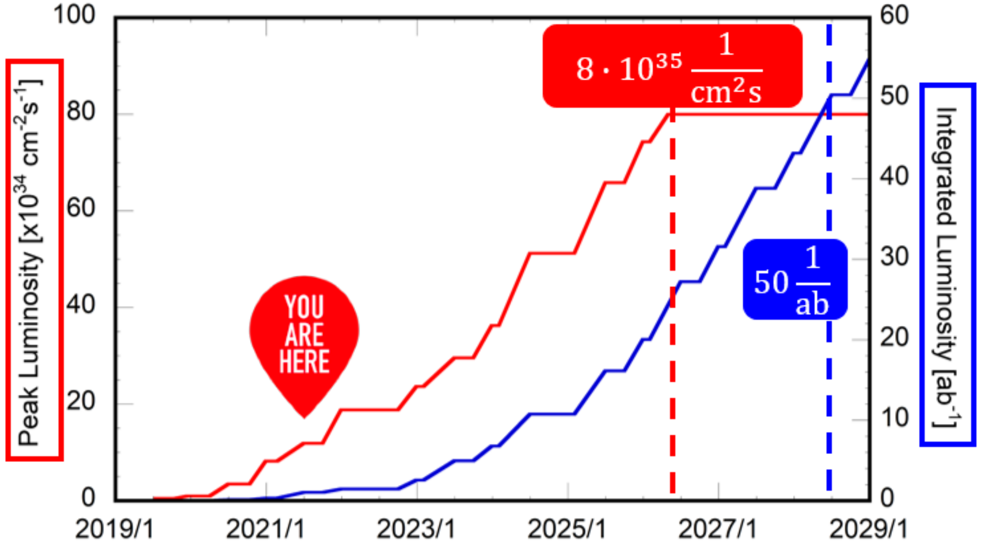
Beryllium beam pipe:
2 cm diameter

Vertex detector:
2 layers DEPFET + 4 layers DSSD

Central Drift Chamber:
He(50%):C_2H_6(50%), Small cells, long lever arm, fast electronics

Readout (TRG, DAQ):
Max. 30kHz L1 trigger
~100% efficient for hadronic events.
1MB (PXD) + 100kB (others) per event
- over 30GB/sec to record

Offline computing:
Distributed over the world via the GRID



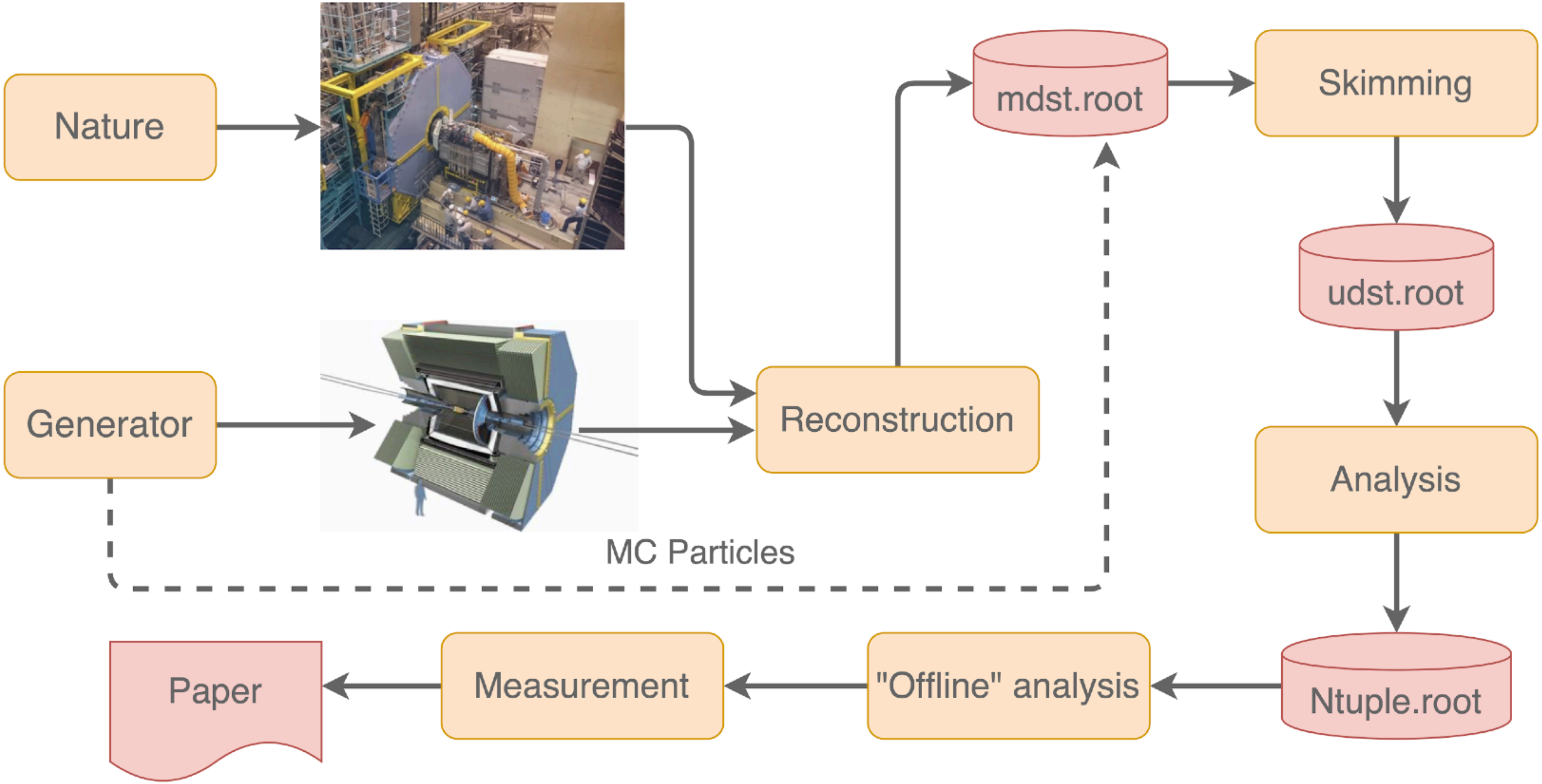
- Integrated luminosity expected: **50 ab^{-1}**

  (x50 than the previous B factories)

The estimated size of the dataset collected by the experiment is ~ 10 PB/year.

Analysis Workflow

From data taking to physics results



Data Formats

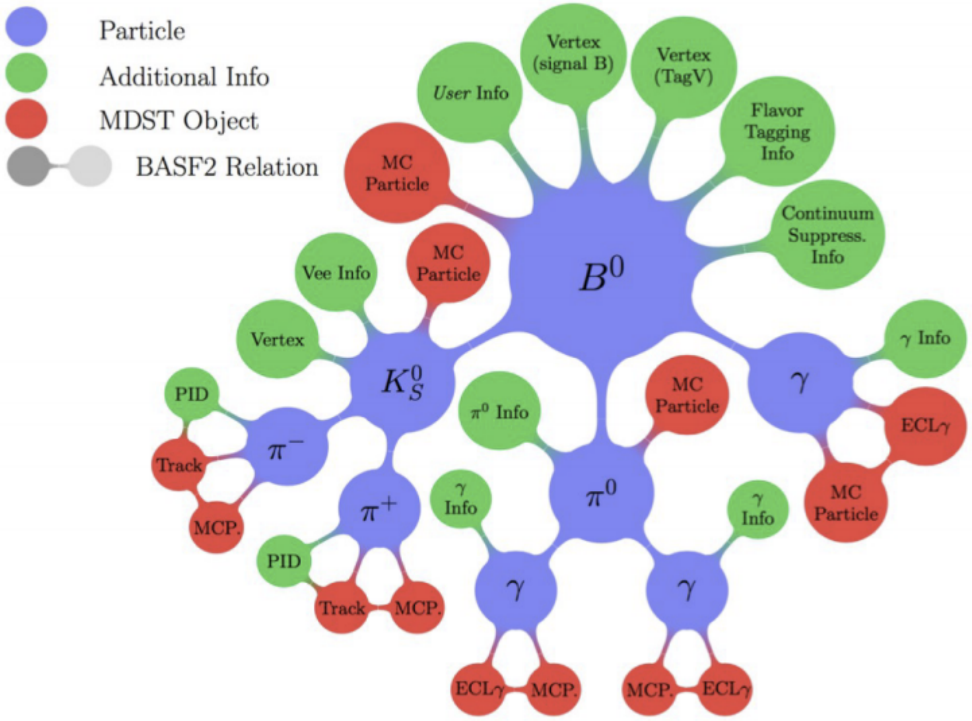
In general, Belle II output is stored in ROOT files containing subsets of dataobjects:

- **RAW**: raw data containing detector information.
  - ~70 kB/event
  - Raw data set during 2019-2021 operation: 5 PB

- **cDST**: calibration Data Summary Table
  - ~120 kB/event
  - Contains objects needed for calibration. Locally produced.

- **mDST**: mini Data Summary Table
  - ~15 kB/event
  - Strictly controlled version intended for physics results.

- **uDST**: user data summary table
  - ~20 kB/event
  - uDST has 10% of the events contained in mDST files.
  - mDST objects + analysis objects (ParticleLists). Produced from skims.



Belle II analysis software framework

A high-level analysis software

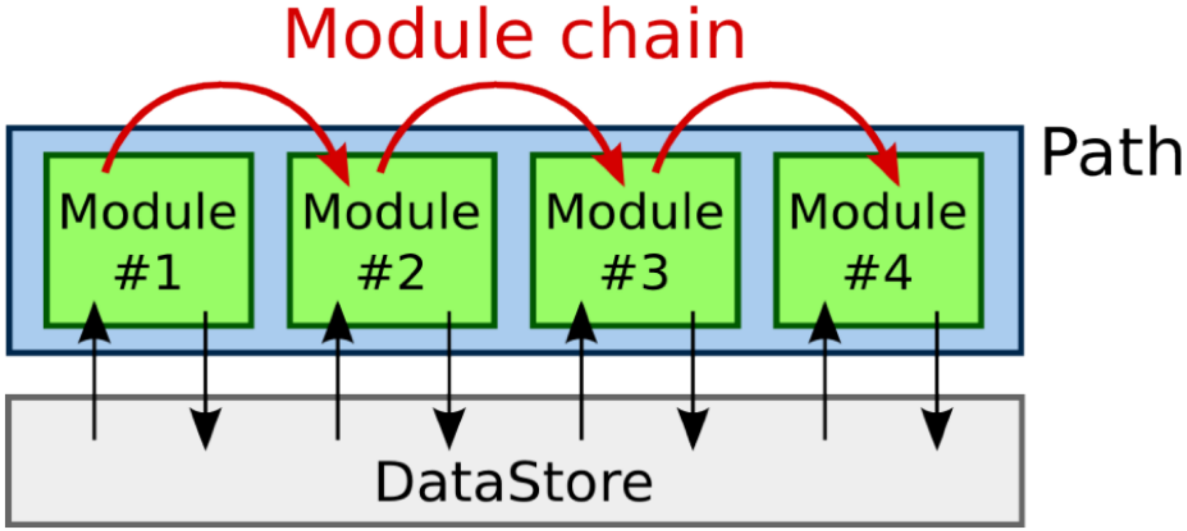
- **Basf2**: Belle II Analysis Software Framework.
  - arXiv:1809.04299

- More of a software framework than an “analysis framework” (name is historic)**.**
  - It performs the unpacking of raw data, detector simulation, tracking, calorimeter clustering, …

- The executable is a wrapper for IPython 3, which controls the setup and configuration of a path.

- Modules are blocks of code that does a specific unit of data processing.
  - They are added to the path calling them inside the steering file.

- User analysis is performed using the **analysis package,** with udst as input.



Distributed Computing

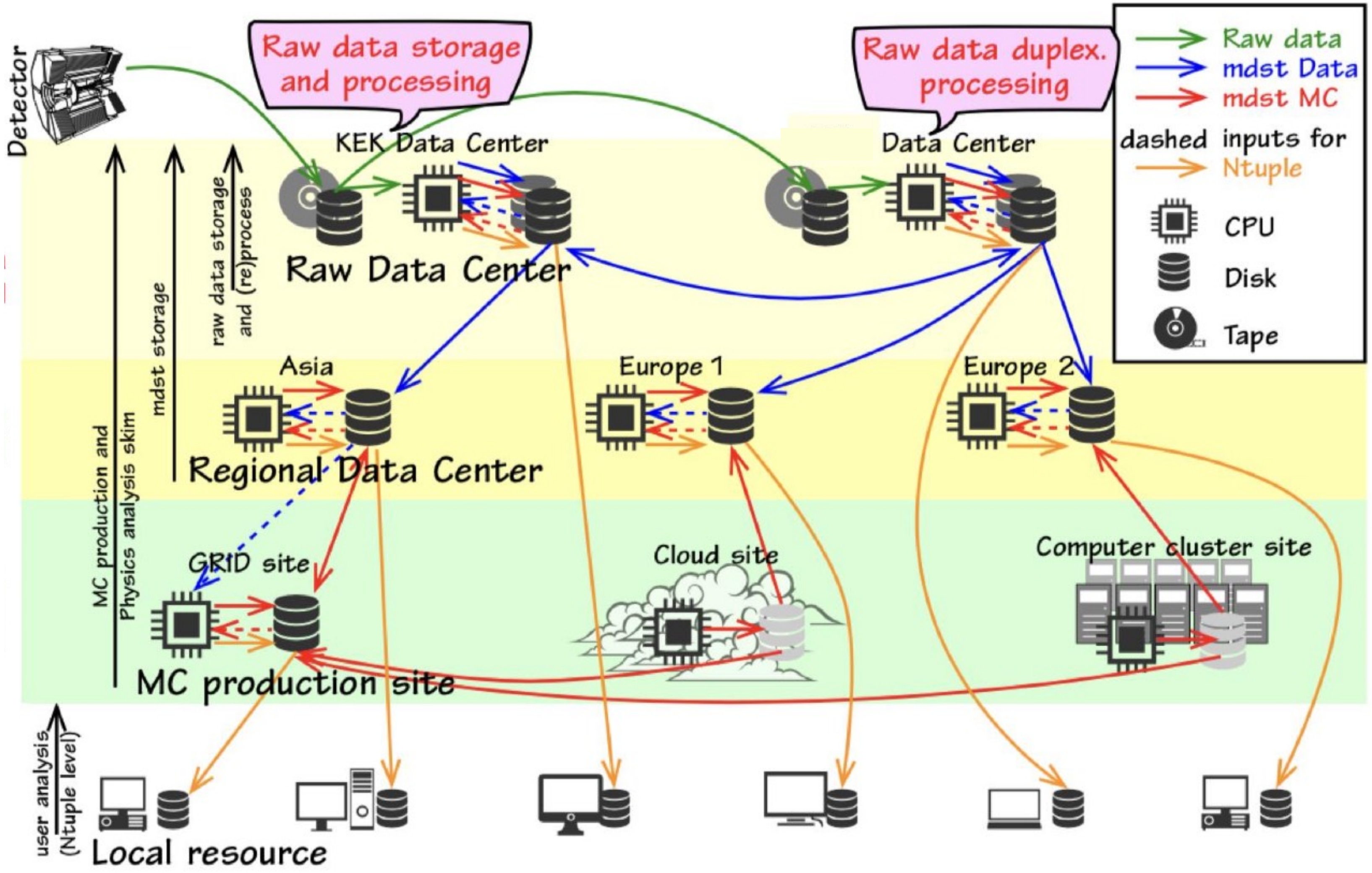
The computing model

- The grid system is conformed by 60 computing sites around the world.
  - The Belle II analysis framework is distributed through **CMVFS**.

- Dedicated data centers keep two copies of the full raw data set.

- Raw data is staged, reprocessed, skimmed and distributed over storage sites .

- Analyzers access data and MC sending jobs to the grid and downloading the output to local resources.



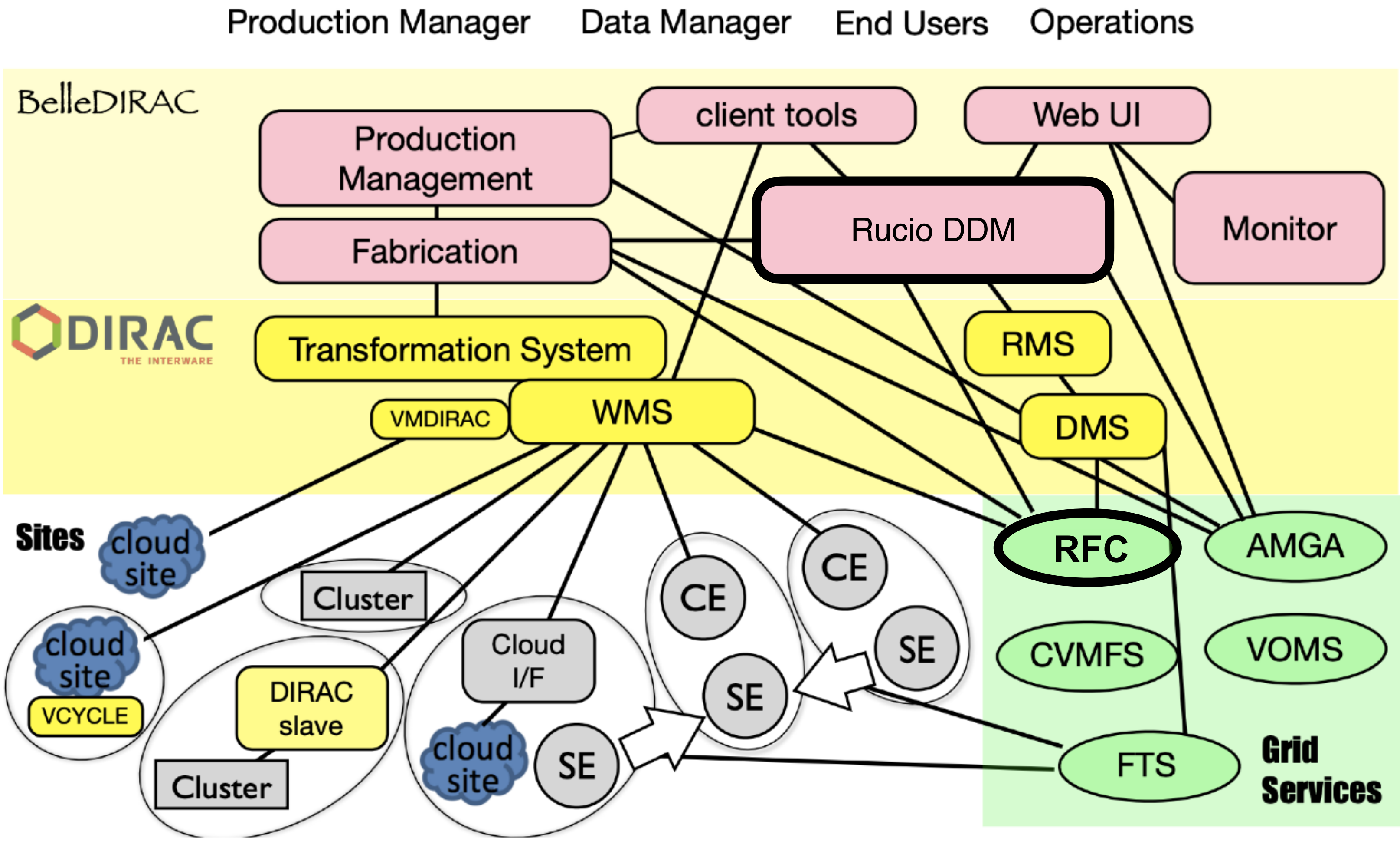
Distributed Computing

Architecture Overview

- We adopted DIRAC as the main framework with an extension (BelleDIRAC).

- This year, the Distributed Data Management system was successfully integrated with **Rucio.**

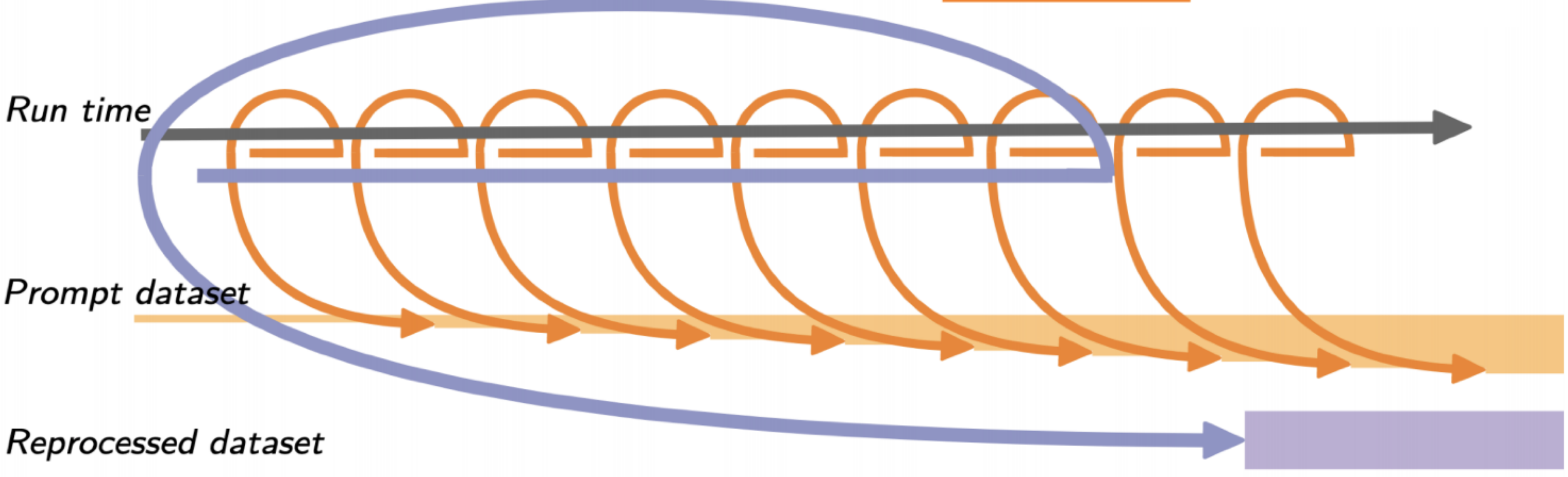
- Rucio provides new features that will exploited for improving the analysis on grid:
  - User replica management.
  - Async deletion.
  - Data popularity.



Processing Scheme

- Ensure smooth, timely production of data for performance studies and physics analysis.

- Data is calibrated weekly in “prompt buckets”, containing ~ 2 TB in mDST format.
- A full reprocessing is performed ~yearly, aiming for physics publications.



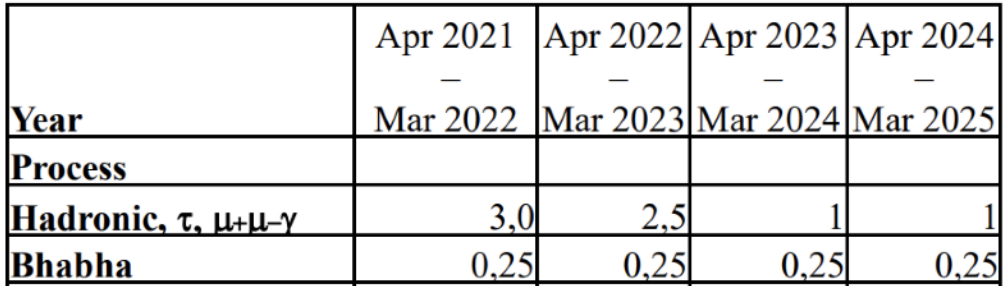
MC production campaigns

- Centralized MC production with unique campaign names.
  - Generic MC (BB, qqbar, tau pair, etc).
  - Signal requests by each physics WG.

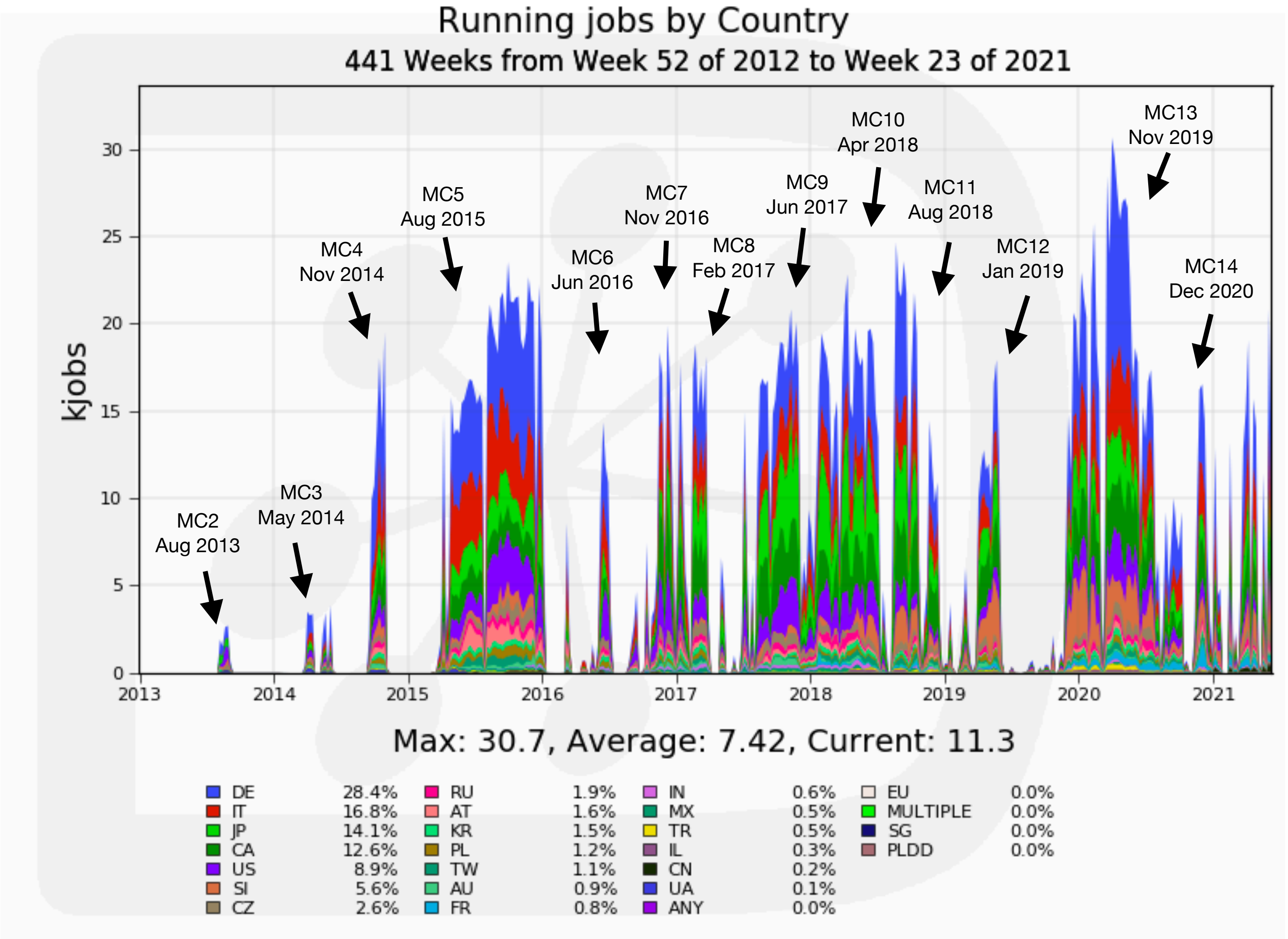
- Belle II policy:
  - **Two replicas** of the latest two campaigns.

- Data set available for analysis
  MC13: **1.5 PB**; MC14 (ongoing)**: 700 TB**

- Ratio to data for generic MC event samples:



Year	Apr 2021 – Mar 2022	Apr 2022 – Mar 2023	Apr 2023 – Mar 2024	Apr 2024 – Mar 2025
Process				
Hadronic, τ, μ+μ−γ	3,0	2,5	1	1
Bhabha	0,25	0,25	0,25	0,25



Generated on 2021-06-16 07:53:00 UTC

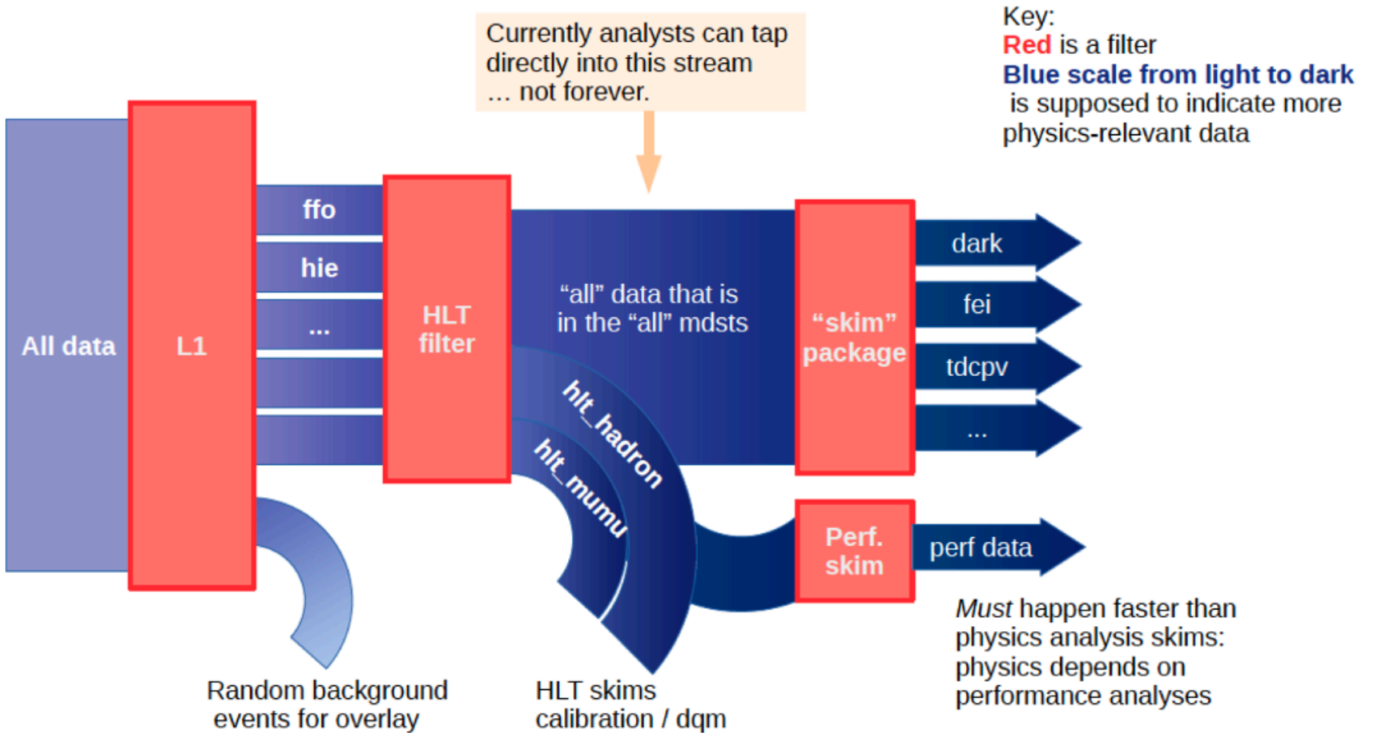
Skimming Scheme

- Each physics working group defines skims, which are also centrally managed producing uDST files.

- The skimming package contains python-based classes developed by liaisons of each WG.

- **Skim usage is highly correlated with grid performance.**
  - Analysts should be working primarily with skimmed datasets (for now, access to mDST is allowed).
  - Some analyses will be challenging to skim, since they can have a high retention.

- Requirements:
  - Retention should be less than 10% of the mDST sample.
  - Processing time for should be less than 500 ms per event.
  - Maximum memory usage is 2GB.
  - Maximum log file size is 30 MB.



Currently analysts can tap directly into this stream … not forever.

Key:
Red is a filter
Blue scale from light to dark is supposed to indicate more physics-relevant data

Random background events for overlay

HLT skims calibration / dqm

Must happen faster than physics analysis skims: physics depends on performance analyses

Gbasf2

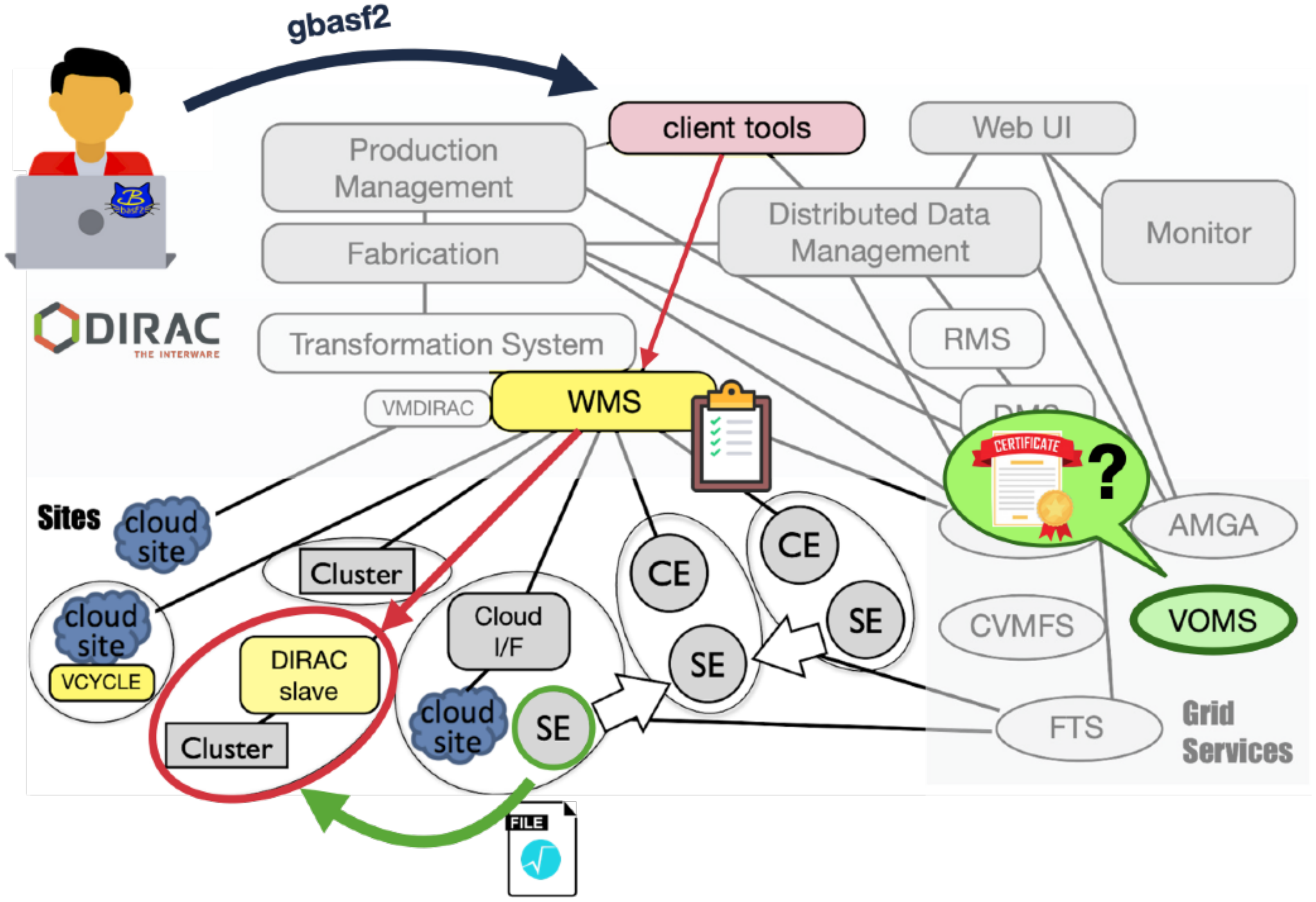
The distributed analysis client for Belle II

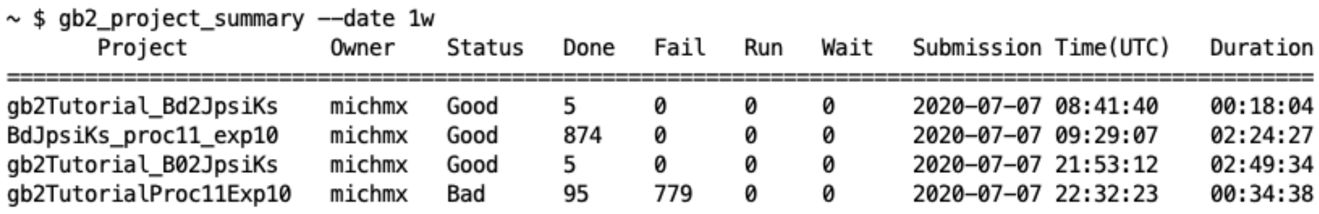
- Gbasf2 is a command-line tool for users intended to submit grid-based jobs.

- The same Python steering files used with Basf2, work with gbasf2 on the grid.
    - User develop his/her job on local resources at first, then submit the job with same steering file.

- Authentication is performed presenting x509 certificates to a VOMS server.

- Users monitor jobs and download the output through a set of command-line tools provided within the gbasf2 environment:





```
~ $ gb2_project_summary --date 1w
       Project                  Owner    Status   Done   Fail   Run   Wait   Submission Time(UTC)   Duration
=============================================================================================================
gb2Tutorial_Bd2JpsiKs         michmx   Good     5      0      0     0      2020-07-07 08:41:40    00:18:04
BdJpsiKs_proc11_exp10         michmx   Good     874    0      0     0      2020-07-07 09:29:07    02:24:27
gb2Tutorial_B02JpsiKs         michmx   Good     5      0      0     0      2020-07-07 21:53:12    02:49:34
gb2TutorialProc11Exp10        michmx   Bad      95     779    0     0      2020-07-07 22:32:23    00:34:38
```

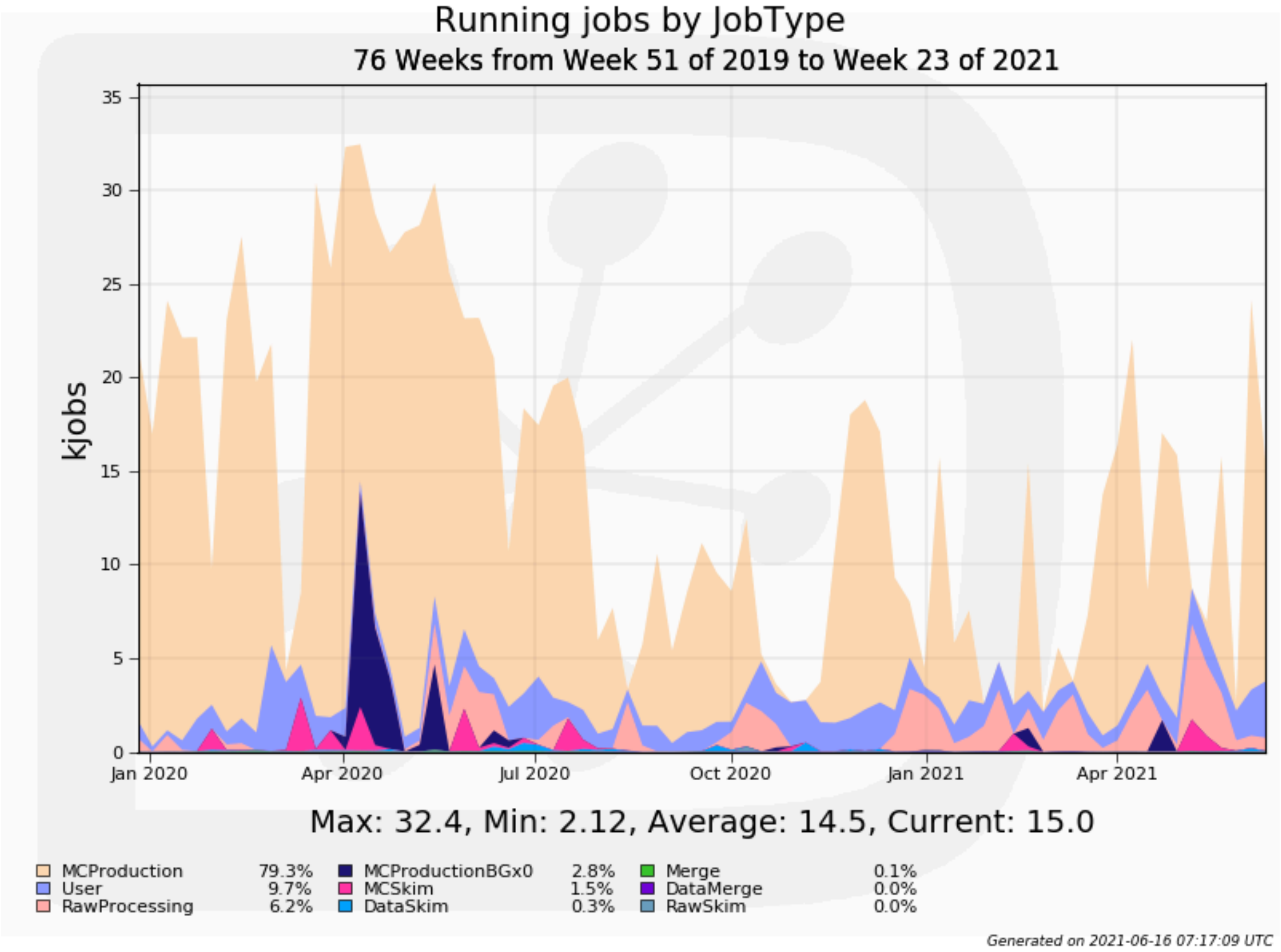
Analysis on the grid

Performing grid-based analysis on data since Jan 2020

- Production activities dominate the grid CPU usage
  - MC production: 81%
  - Data processing: 7%
  - Skimming: 2%

- User analysis represents the 10%.

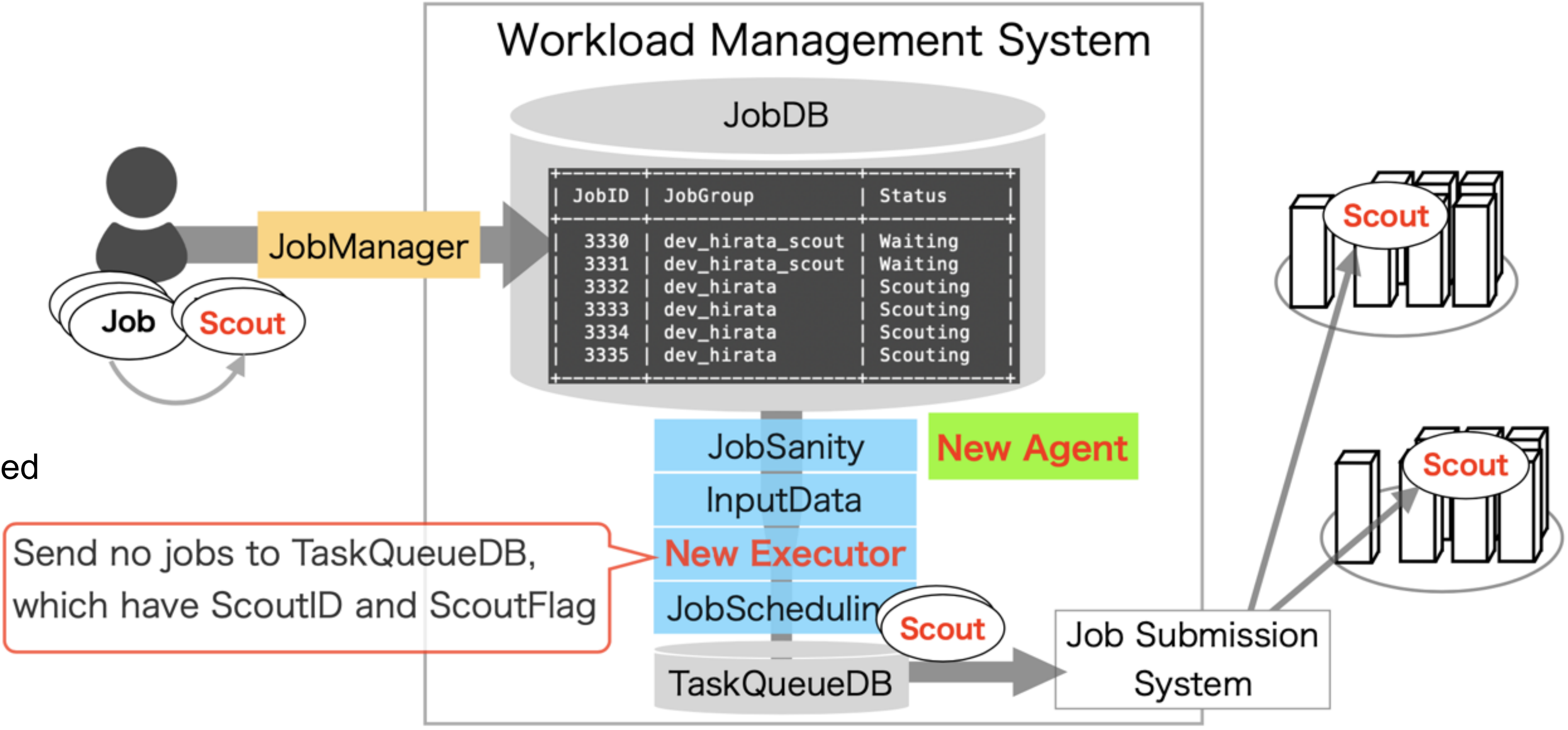
- **Issues identified:**
  - Analysis with non-skimmed data put a heavy load on the grid services.
  - Sometimes, large projects submitted with errors keep the resources busy.
  - Current limit in the size of the sandbox for user analysis is 5 GB. Advanced usage, like training of BDTs, reach that limit.



Scout Jobs

Preventing failed jobs from users

- If the main project has a large number of jobs, a part of them are copied as a group of scout jobs.
  - ○ Main submission proceed only if scout jobs finish without errors.
  - ○ Otherwise, user is notified.
- Successfully implemented in production.



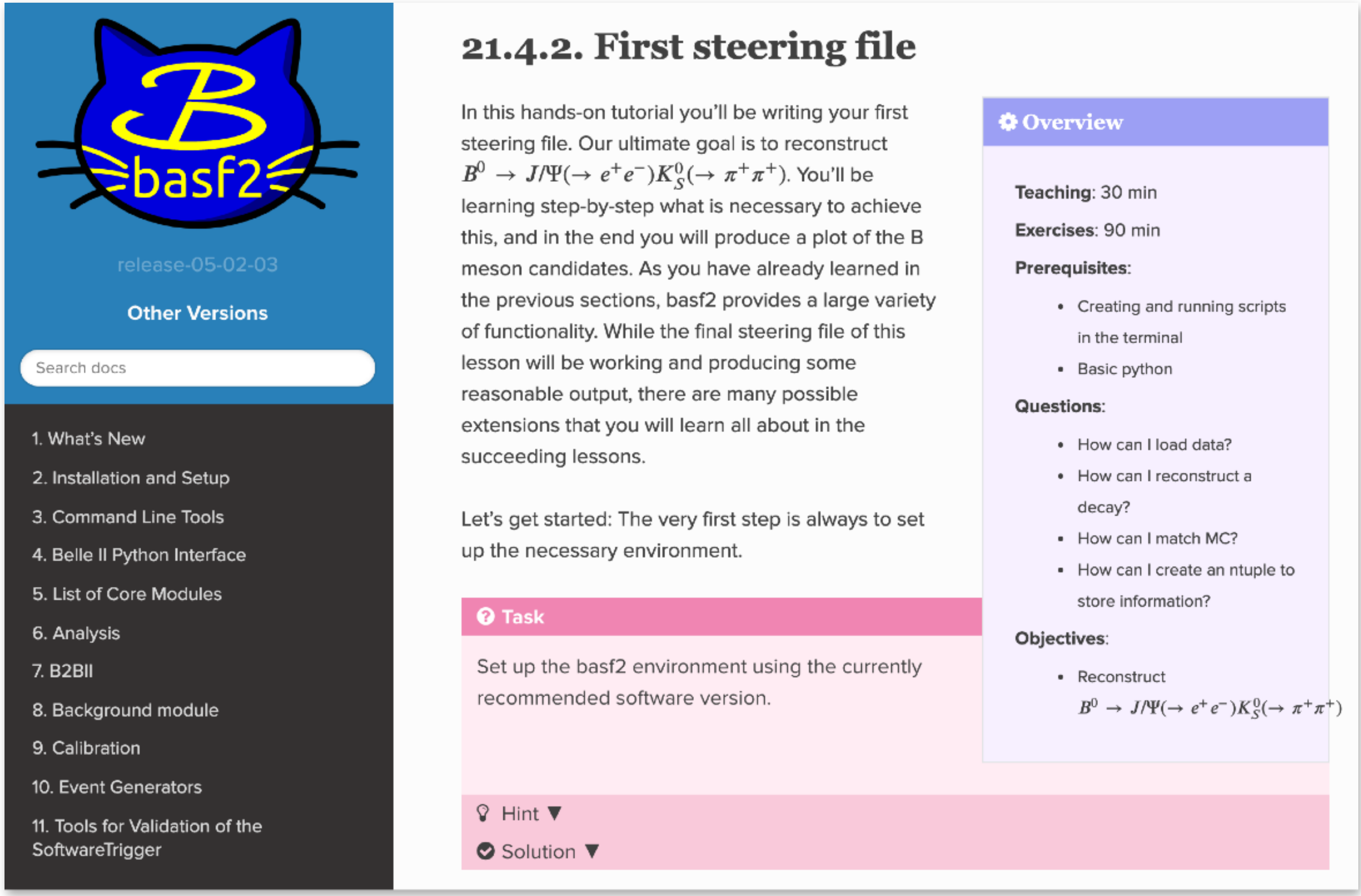
```
+-------+-------------------+-------------+
| JobID | JobGroup          | Status      |
+-------+-------------------+-------------+
|  3330 | dev_hirata_scout  | Waiting     |
|  3331 | dev_hirata_scout  | Waiting     |
|  3332 | dev_hirata        | Scouting    |
|  3333 | dev_hirata        | Scouting    |
|  3334 | dev_hirata        | Scouting    |
|  3335 | dev_hirata        | Scouting    |
+-------+-------------------+-------------+
```

Send no jobs to TaskQueueDB, which have ScoutID and ScoutFlag

Documentation and Training

Belle II for newcomers and experts

- Efforts to maintain a clear documentation for beginners and advanced users.
- Tutorials developed in order to introduce the framework and the analysis on the grid.
- Belle II performs Starterkit workshops three times per year.
- Additionally, we encourage users to participate as data production shifters, where they learn concepts about the software and the computing system.



21.4.2. First steering file

In this hands-on tutorial you'll be writing your first steering file. Our ultimate goal is to reconstruct $B^0 \to J/\Psi(\to e^+e^-)K_S^0(\to \pi^+\pi^+)$. You'll be learning step-by-step what is necessary to achieve this, and in the end you will produce a plot of the B meson candidates. As you have already learned in the previous sections, basf2 provides a large variety of functionality. While the final steering file of this lesson will be working and producing some reasonable output, there are many possible extensions that you will learn all about in the succeeding lessons.

Let's get started: The very first step is always to set up the necessary environment.

Task

Set up the basf2 environment using the currently recommended software version.

Overview

Teaching: 30 min

Exercises: 90 min

Prerequisites:

- Creating and running scripts in the terminal
- Basic python

Questions:

- How can I load data?
- How can I reconstruct a decay?
- How can I match MC?
- How can I create an ntuple to store information?

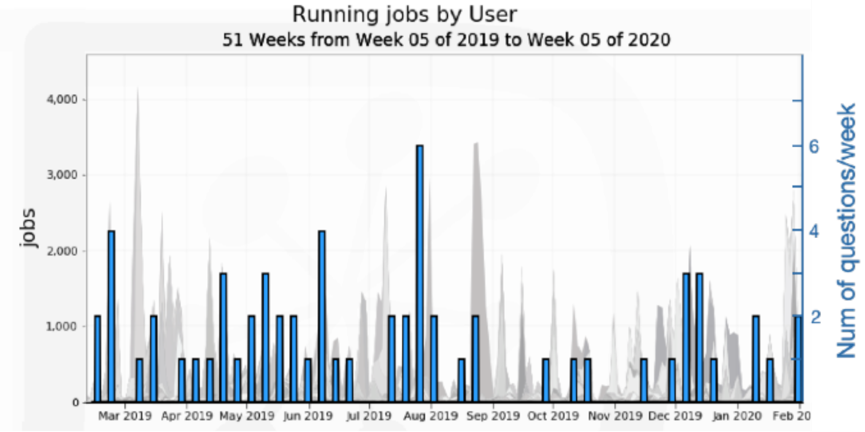
Objectives:

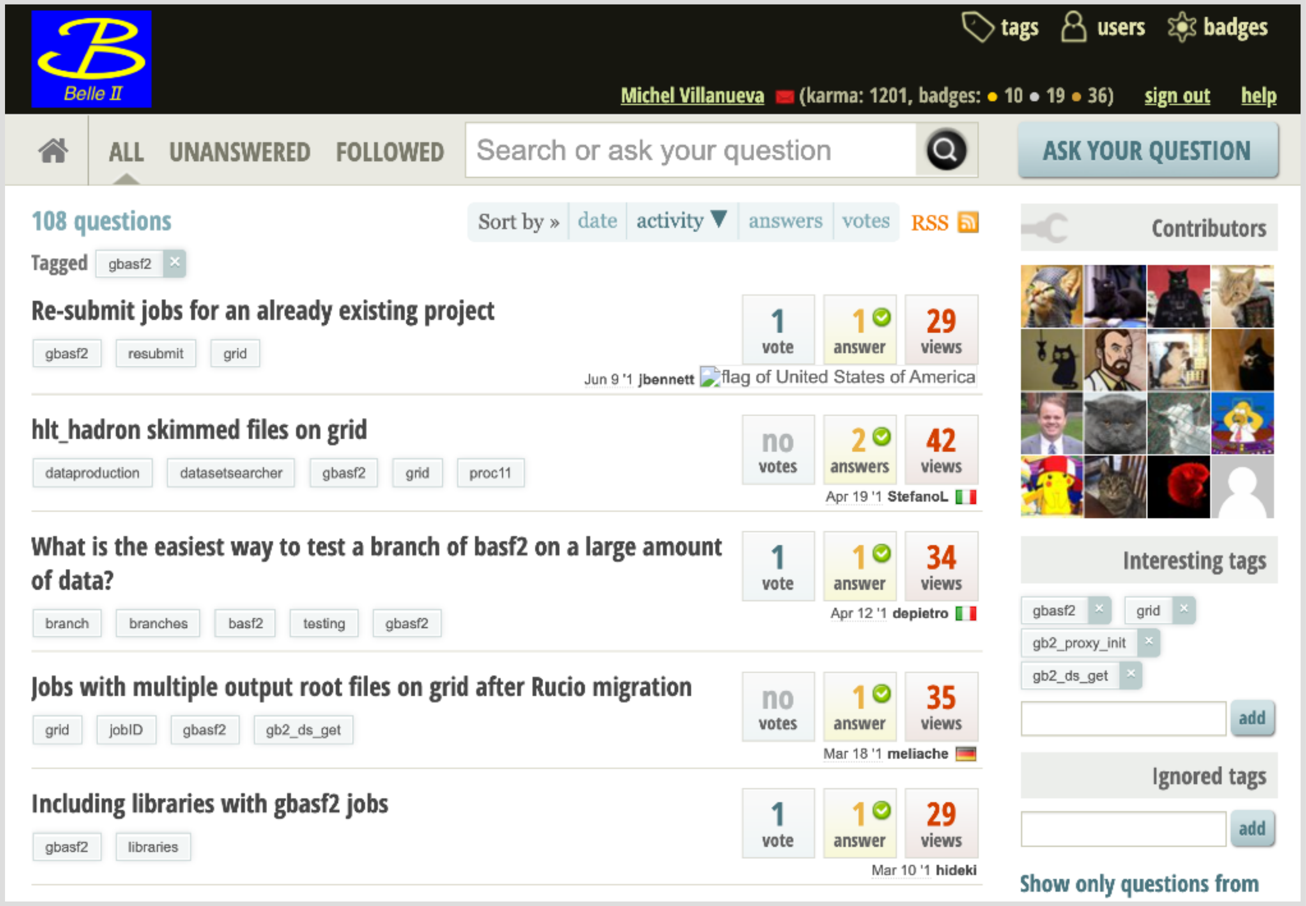
- Reconstruct $B^0 \to J/\Psi(\to e^+e^-)K_S^0(\to \pi^+\pi^+)$

Collaborative Services

Support for Analyzers

- Collaborative services as a mail forum and an Askbot server have been deployed to provide support.
  - ~100% messages answered. In some cases, multiple solutions.
  - Not only experts, but users provide help too.
  - Users also suggest new features. Great feedback for developers.





Michel Villanueva (karma: 1201, badges: ● 10 ● 19 ● 36) sign out help

108 questions

Tagged gbasf2 ×

Re-submit jobs for an already existing project — 1 vote, 1 answer, 29 views
gbasf2 · resubmit · grid
Jun 9 '1 jbennett flag of United States of America

hlt_hadron skimmed files on grid — no votes, 2 answers, 42 views
dataproduction · datasetsearcher · gbasf2 · grid · proc11
Apr 19 '1 StefanoL

What is the easiest way to test a branch of basf2 on a large amount of data? — 1 vote, 1 answer, 34 views
branch · branches · basf2 · testing · gbasf2
Apr 12 '1 depietro

Jobs with multiple output root files on grid after Rucio migration — no votes, 1 answer, 35 views
grid · jobID · gbasf2 · gb2_ds_get
Mar 18 '1 meliache

Including libraries with gbasf2 jobs — 1 vote, 1 answer, 29 views
gbasf2 · libraries
Mar 10 '1 hideki

Interesting tags: gbasf2 × grid × gb2_proxy_init × gb2_ds_get ×

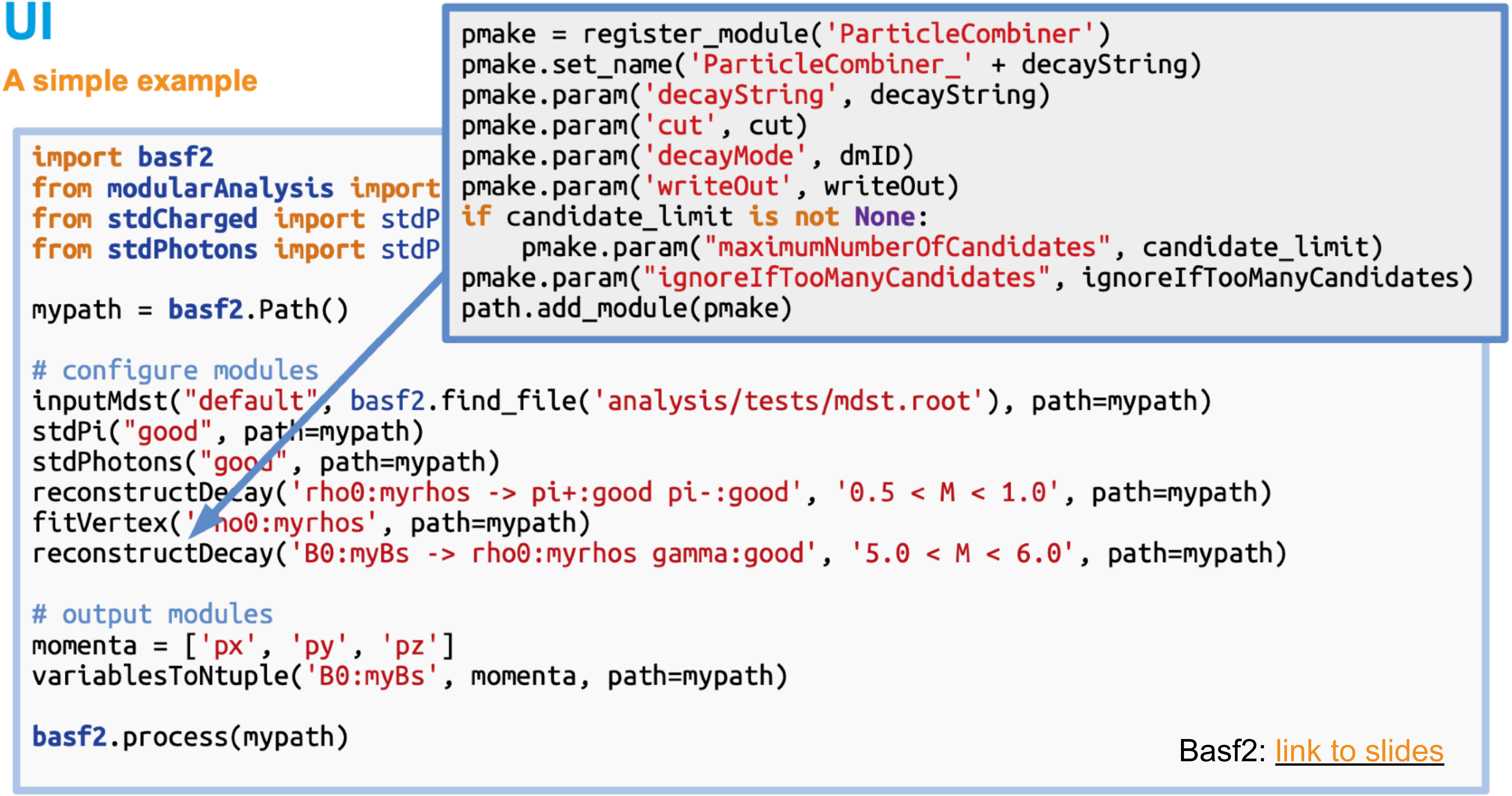
Summary

- Belle II is expected to produce tens of petabytes of real and simulated data per year.
- Datasets intended for analysis are produced by data production experts on the grid. Skims are defined for each physics working group.
- Some analysis are not compatible with our current skimming scheme (the retention rate is too high). Several solutions are on discussion.
- The Integration of the DDM system with Rucio was successfully performed this year.
- Gbasf2 is the command line client for submitting grid-based Basf2 jobs.
  Allow submission with the same high-level steering files used in offline analysis.
- We are working with scout jobs and improving of documentation to prevent a large number of failed jobs.

Backup

UI

A simple example



```
import basf2
from modularAnalysis import
from stdCharged import stdP
from stdPhotons import stdP

mypath = basf2.Path()

# configure modules
inputMdst("default", basf2.find_file('analysis/tests/mdst.root'), path=mypath)
stdPi("good", path=mypath)
stdPhotons("good", path=mypath)
reconstructDecay('rho0:myrhos -> pi+:good pi-:good', '0.5 < M < 1.0', path=mypath)
fitVertex('rho0:myrhos', path=mypath)
reconstructDecay('B0:myBs -> rho0:myrhos gamma:good', '5.0 < M < 6.0', path=mypath)

# output modules
momenta = ['px', 'py', 'pz']
variablesToNtuple('B0:myBs', momenta, path=mypath)

basf2.process(mypath)
```

```
pmake = register_module('ParticleCombiner')
pmake.set_name('ParticleCombiner_' + decayString)
pmake.param('decayString', decayString)
pmake.param('cut', cut)
pmake.param('decayMode', dmID)
pmake.param('writeOut', writeOut)
if candidate_limit is not None:
    pmake.param("maximumNumberOfCandidates", candidate_limit)
pmake.param("ignoreIfTooManyCandidates", ignoreIfTooManyCandidates)
path.add_module(pmake)
```

Basf2: link to slides

Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Michel Hernandez Villanueva
michel.hernandez.villanueva@desy.de
Orcid: 0000-0002-6322-5587