User documentation and training at Belle II

Spreading and preserving knowledge in sparse collaboration

CHEP 2019, Adelaide

F.Tenchini[1], I. Komarov[1], M. Ritter[2], S. Cunliffe[1], T. Kuhr[2]
ilya.komarov@desy.de

[1] DESY
[2] LMU

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

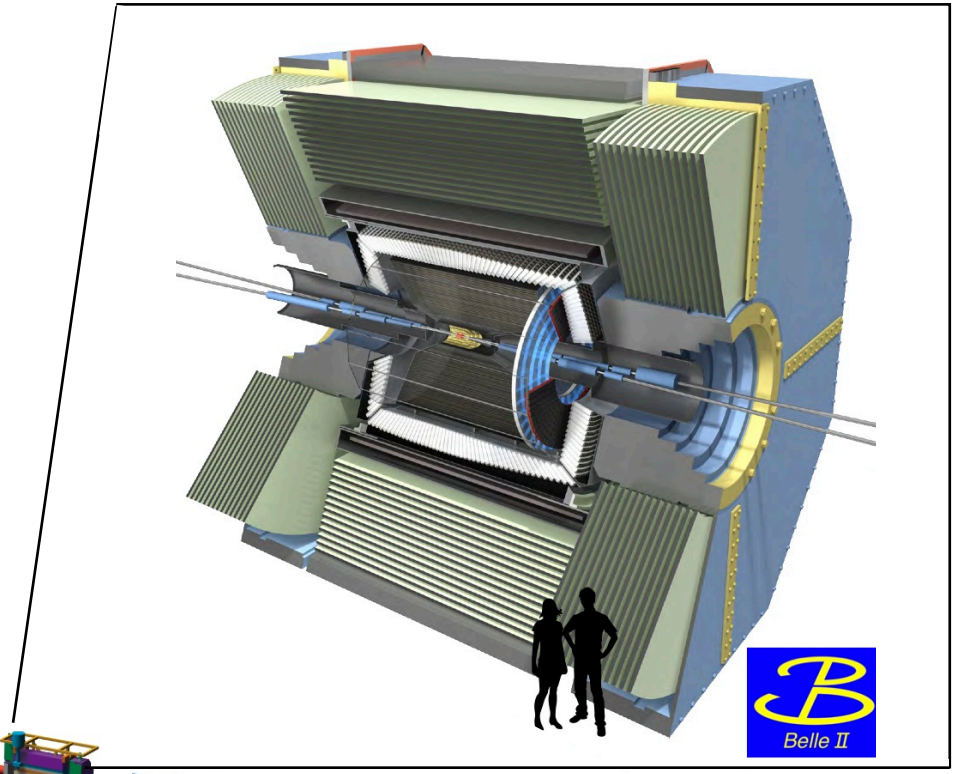
DESY.

Meet the Belle II

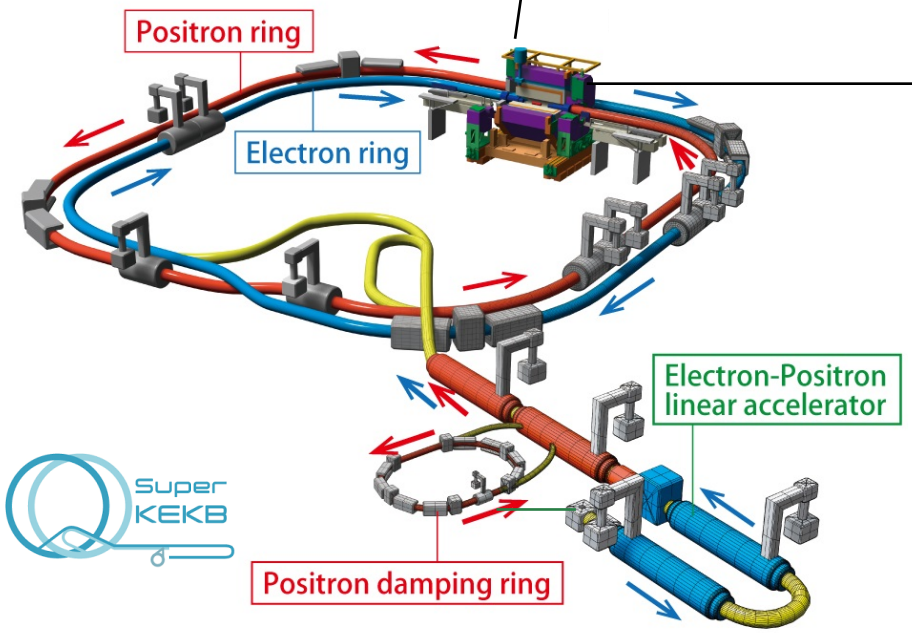
Belle II is multipurpose detector located in Tsukuba, Japan.

It studies processes occurring in e^+e^- collusions at the energy of ~10 GeV.

While Belle II inherits many things from its predecessor, our software is **not** one of them - it's completely new.



Belle II analysis software (basf2) is C++(17) core with python user interface.



Tasks of the documentation and training group

And how they evolved

In Summer 2017 we created “Documentation, training and software outreach” working group.

It was supposed to **maintain documentation** for the analysis software, **organise software tutorials** and **advertise new tools**.

However, quite soon we realised that we need to change priorities to be efficient.

Now, our goal is to **provide smooth start for newcomers and effective support for existing users**.

To accomplish it, we had, in particular, to significantly extend content of our tutorials and redesign the whole documentation system.

Speed of the software development

Challenge #1

- Belle II is young collaboration and our software is in process of active development:
  - We had 814 commits last month
  - We have 2 major releases per year.

- Major releases fix serious bugs and break backward compatibility, which complicates migration but also makes it imminent.

- In terms of our documentation and training, this means:
  - Documentation needs to be permanently updated
  - Users need to be taught how and be ready to migrate to the new release.

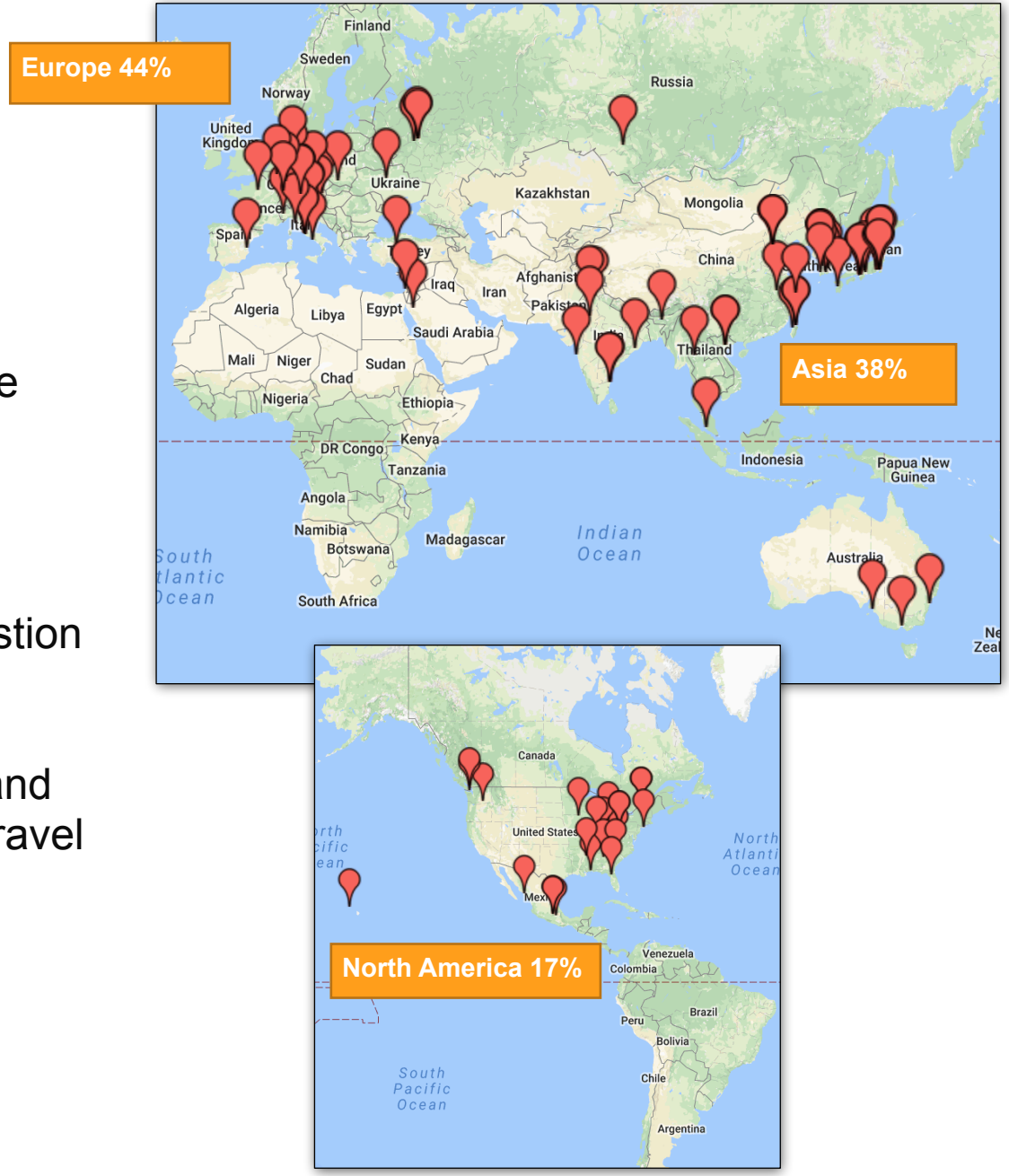
Geography of the collaboration

Challenge #2

Belle II collaboration is scattered around the world: we are 985 members from 118 institutions in 26 countries, most of us reside next to home institute.

This has sequences:

- It's often impossible to "knock the expert's door" to ask a question - **documentation** is essential.
- It's hard to **gather people** for training sessions in one place and people who need it most (master students) often don't have travel funds.
- **Inclusion** is something to be always kept in mind.



Training

a.k.a. B2StarterKits

Schedule and Content

- Three times a year big part of Belle II meets on experimental site for the general meeting. These are unique occasions to get teachers and students around.
- The workshop lasts 3 days, with lecture-like contributions in the morning and hands-ons in the evening.

Content:

- We teach Belle II software
- We teach experimental physics
- We teach data analysis
- We make students to do simple but meaningful analysis.

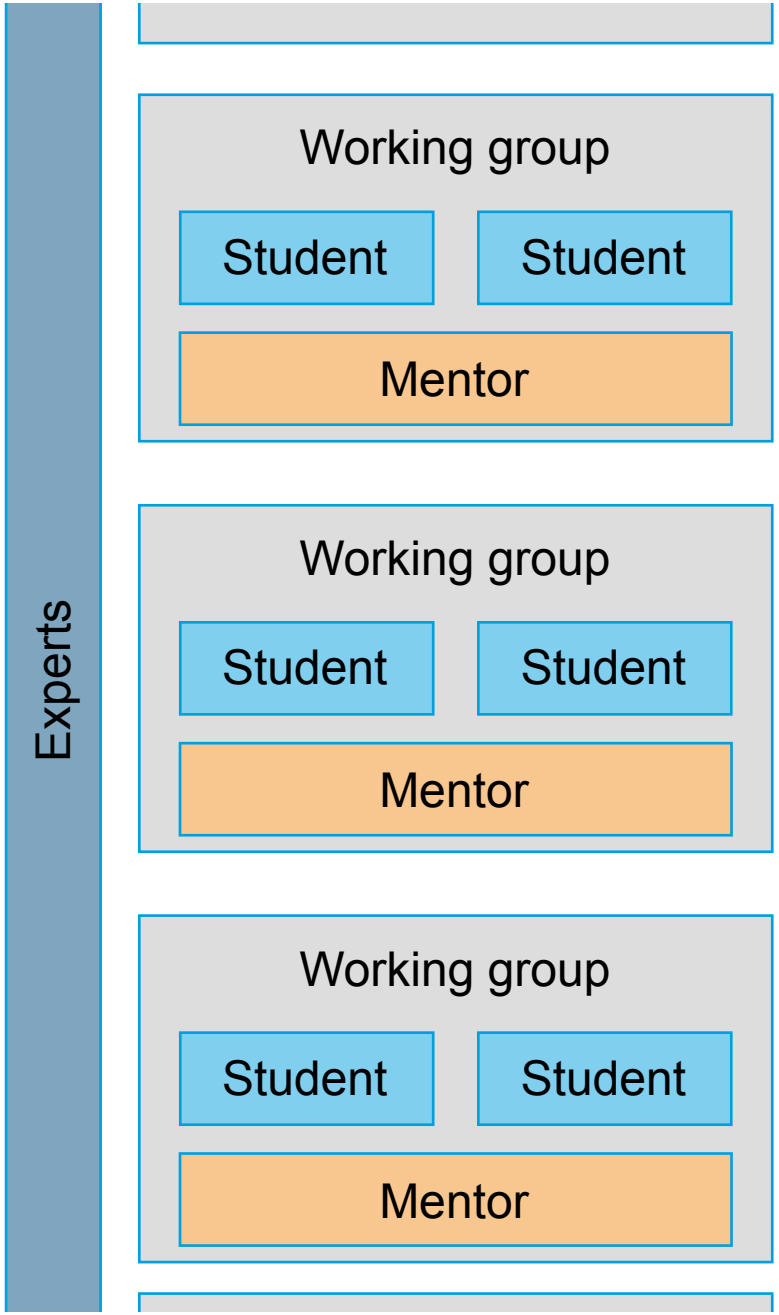
	Day 1	Day 2	Day 3
Morning	Introduction to the software	“Expert” lectures	
Afternoon	Hands-on tutorial	Work in groups	

Addressing the inclusion

Through the organisation

We have ~30 students with different education levels and cultural backgrounds. To make workshop useful for everyone, we

- Set minimal basis: we prepare students by timely communicating the prerequisites.
- Split them in working groups. Typically, its 2 beginners and 1 advanced student. Advanced students work as a mentors.
    - Mentors take care of technical issues.
    - Mentors coordinate analysis in the working group.
    - Mentor is the first point of contact for the student.
    - Mentors follow the progress of their students and organise discussions.
    - If mentor can’t help, group asks for the help of the expert.



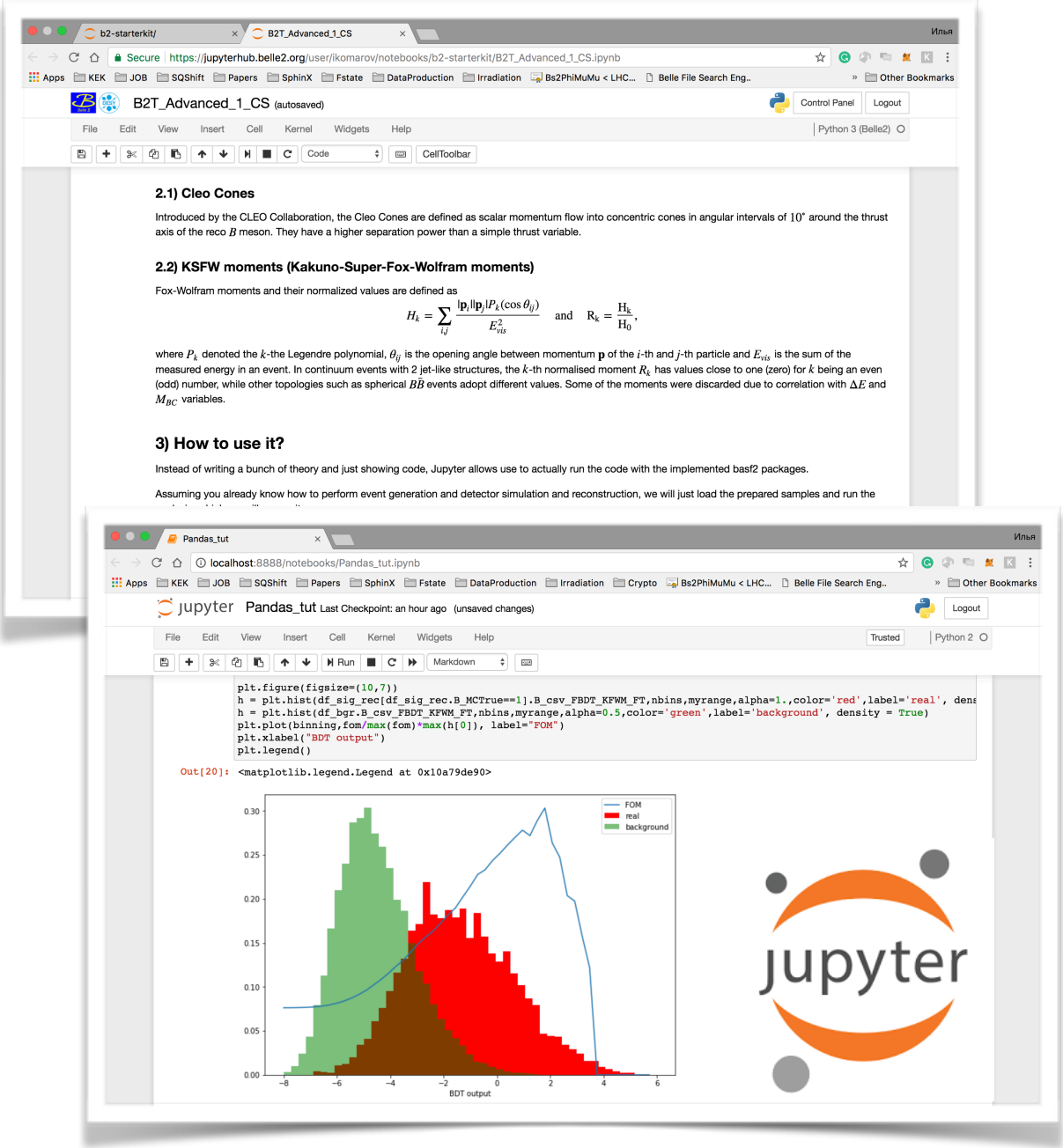
Containerised approach

Reaching those who can't come

Not all students can attend workshops in Japan. For them, we organise StarterKit materials such that they work out-of-the box and the workshop can be easily reproduced locally:

- All StarterKit hadns-on materials are stored in dedicated repository (jupyter notebooks); Lectures are available at indico.
- The virtual machine with repository and Belle II software is accessible through jupyterhub server - this allows for the immediate start with zero background. This is also lifesaver during the workshops for students who have insurmountable troubles with their local machines.

In 2019, we had 2 big (20+ students) local workshops based on StarerKit material.



Documentation

Keep it up-to-date

In-code documentation

Relevance and **coverage** are the two main challenges of the user documentation. At Belle II, we use Sphinx:

- Documentation is part of the code. Changes are reviewed and have version control.

Keep it up-to-date

In-code documentation

Relevance and **coverage** are the two main challenges of the user documentation. At Belle II, we use Sphinx:

- Documentation is part of the code. Changes are reviewed and have version control.
- User documentation is casted from in-code documentation of python modules. Strict software review rules ensures quality and coverage here.
- All documentation is collected in one place with search, links, syntax highlighting etc.

SPHINX
Python Documentation Generator

https://software.belle2.org

https://software.belle2.org/sphinx/release-04-00-01/analysis/doc/FlavorTagger.html

Other Versions

Search docs

Lambda:

In the decay $b \to \Lambda_c^* X \to \Lambda X \to X p \pi^- (b \to c \to s)$ the flavor of the Lambda tags the flavor of the B. For this, a proton and a pion are reconstructed to a Lambda.

In the following the particles providing the flavor tag information, i.e. the flavor signatures, are denoted as target.

Below: Simple draft (no physical magnitudes) to illustrate the different decays providing the signatures belonging to the different categories.

Categories	Targets
Electron	e^-
Intermediate Electron	e^+
Muon	μ^-
Intermediate Muon	μ^+
KinLepton	e^-
Intermediate KinLepton	ℓ^+
Kaon	K^-
KaonPion	K^-, π^+
SlowPion	π^+
FastHadron	π^-, K^-
MaximumP	ℓ^-, π^-
FSC	ℓ^-, π^+
Lambda	Λ
Total= 13	

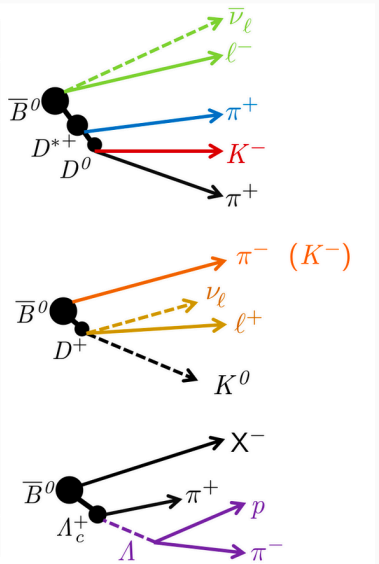


Fig. 6.5 Underlying decay modes of the flavor tagging categories.

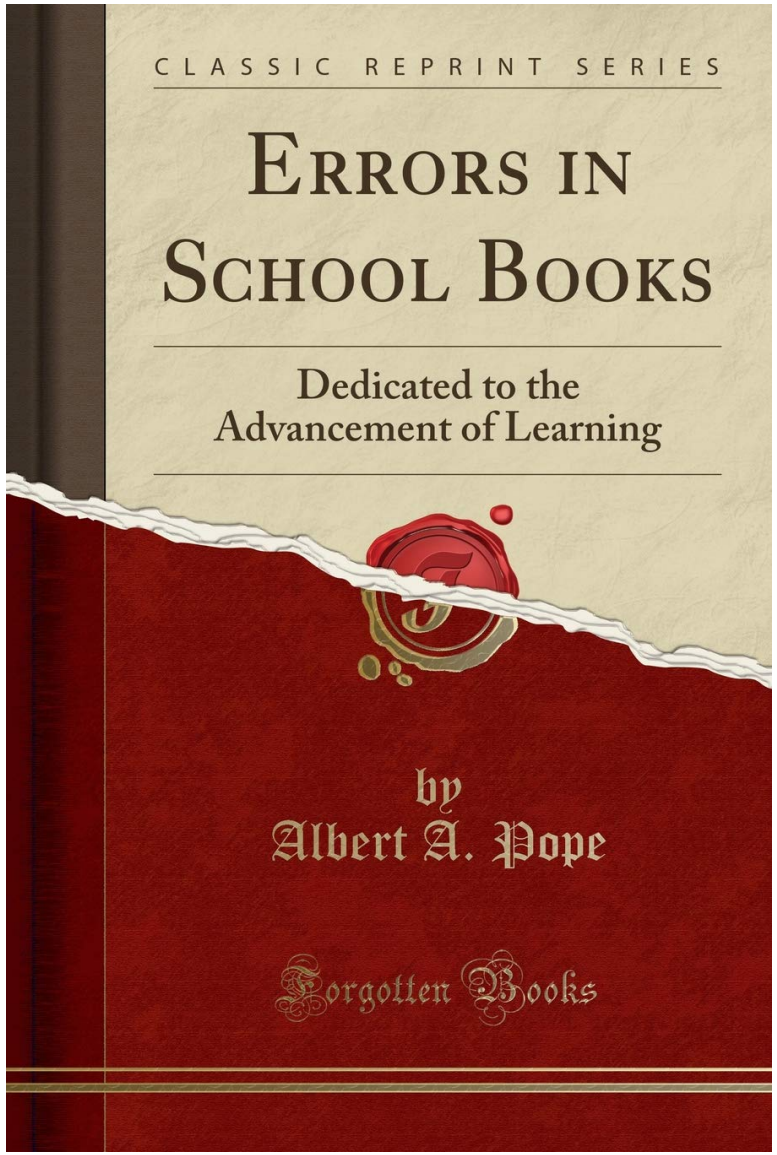
Keep it up-to-date

Unit tests to prevent outdates

- Important part of documentation are tutorials.
- In past, it often happened that changes in python interface were not reflected in tutorials (only in documentation, since it's in-code)
- To cope with than, we covered tutorials with test:

  <u>Build fails if tutorial crashes;</u>
  <u>No build ⇒ no merge.</u>

The mechanism is not perfect yet: we don't check if tutorials still provide sensible results, we require just creation of output file. Good enough for "snippet collection", but some maintenance is still needed: typically, we revisit tutorials before every major release.



Availability of training datasets

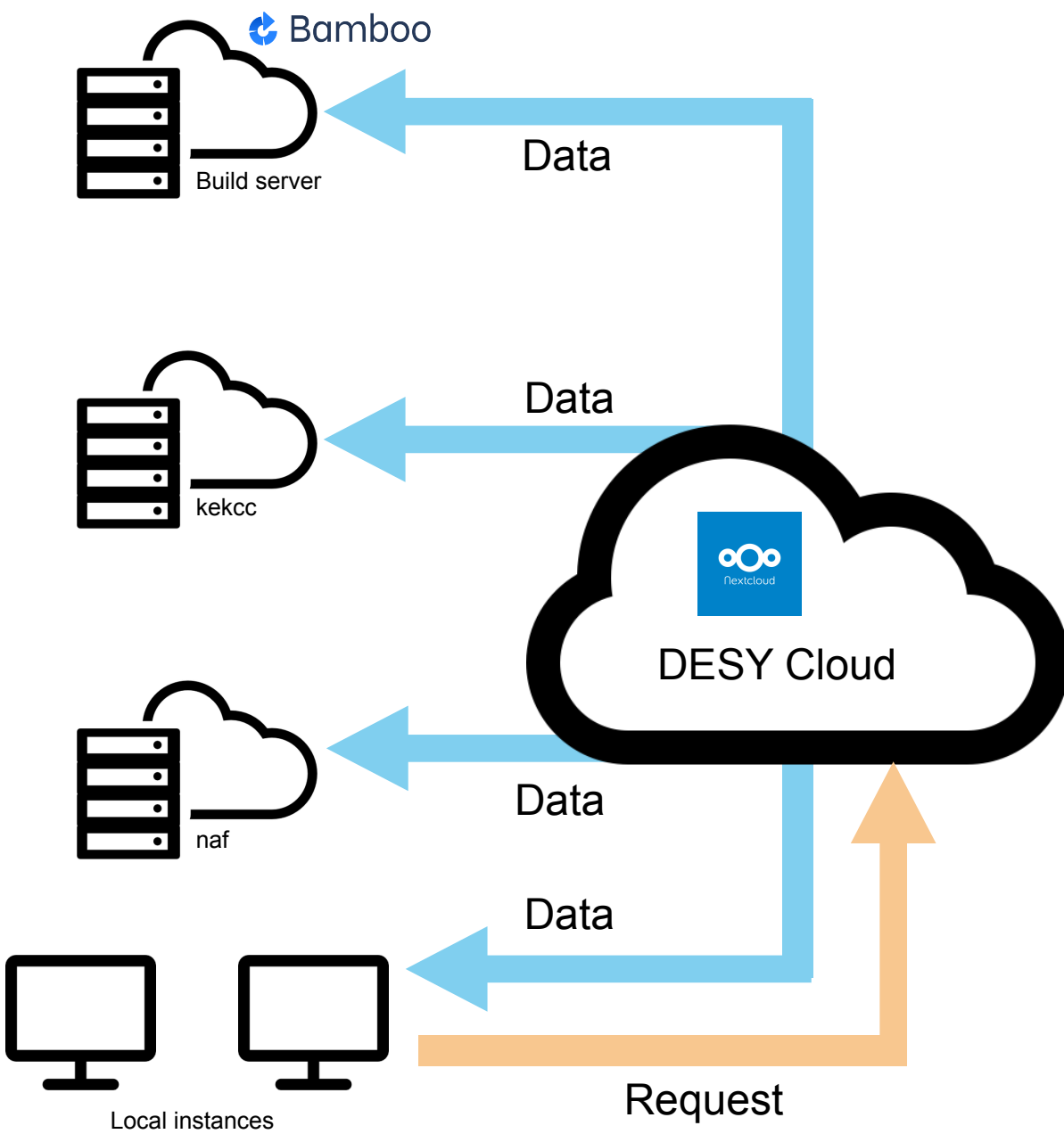
Cloud solution

Datasets used for trainings:

- Should be regularly updated
- Too big to be part of a framework

To establish reproducibility of tutorials on different machines, we implemented cloud solution:

- All training datasets are kept at Nextcloud instance
- Cron job synchronise datasets “known” servers (build server, kekcc (Jaoan), naf (Germany))
- If framework doesn’t find tutorial files inplace (e.g. user runs code at local installation), it requests them from cloud



Questions

Prompt responses

Askbot - create your Q&A forum.

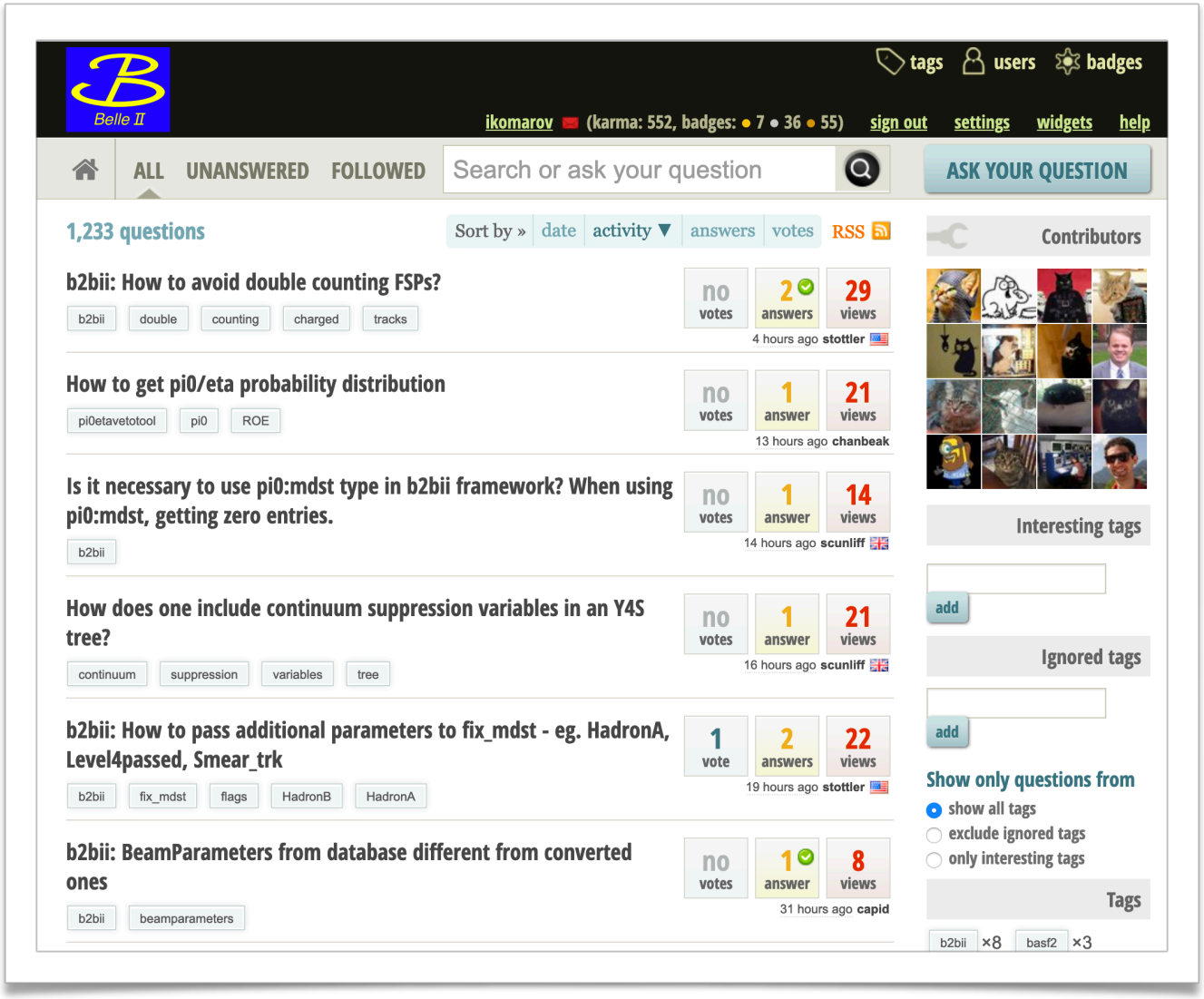
- We have an instance of **Askbot** providing collaborative Q&A service.
- It works quite well:
  - 1233 questions asked in 15 moth
  - 83% of them answered
  - Response time is <1 day.
- Adoption is enforced by developers:

Mail to developer's mailing list:

Dear experts,
I have a problem…

Answer:

```
Dear user,
Please ask at
questions.belle2.org
```



One tool to rule them all: cross-source search engine

Sphinx, questions and wiki heritage

Log out

Amazing Belle2 Search Tool

Collections

1. Enter the query

Select a release

release-04-00-01 (recommended)

2. Select release

Go!

3. Get results

Results from questions.belle2.org

How does one include continuum suppression variables in

Hi, When using Continuum Suppression, how does one write out the B c
We are trying to reduce the number of trees we have in the output to prevent huge fil
possible. Thanks in advance! ...

Answers: 1

tags: continuum suppression variables tree

Results from Sphinx

variables.collections

Python module, in 6.3.1. VariableManager

Results from confluence.desy.de

Conclusion

Documentation and training

At Belle II

Belle II Documentation and Training group works hard to establish smooth start for beginners and effective support for existing users.

- The core of training efforts are **B2StarterKit** workshops.
  We don't only teach software, we teach doing analysis in Belle II, from means of communications within collaboration to industry-standard tools for data analysis.
  - Centrally organised workshops
  - Jupyterhub instance with tutorial materials
  - Materials are available for local events

- We developed and adopted new documentation strategy. Instead of wikis and mailing lists, we use:
  - **Sphinx** instance for core user documentation
  - **Askbot** instance for Q&A service

Books and a bookcase, **Esei Keisai**, XIX c,