ALICE migration to GitHub: debriefing and feedback

Dario Berzano

for the Git migration working group

ALICE Offline Week - March 30, 2017

Motivation

- On March 23 the software repositories for AliRoot and AliPhysics were moved from CERN IT-provided Git service to the popular GitHub platform

- CERN IT has retired git.cern.ch in favor of gitlab.cern.ch
  Originally announced on January 2016 and foreseen for May 2016. We have agreed to postpone the migration for us to mid-March 2017 (after Quark Matter + test period)

- Why not GitLab: 1:1 mapping from old service not offered by CERN IT
  No directory-based permissions (we had them with Gitolite). GitHub more unlikely to change dramatically in the future (and in the middle of a run): too many important companies behind

- Provide ALICE with a more modern software lifecycle already adopted in O^2
  In preparation of Run 3, we have provided current software with a friendly workflow that leverages automatic tests, encourages human reviews and accountability

Git migration working group

- We have many users with a different computing background: form a working group to discuss, get early feedback and accommodate the needs of the community
- Testbed for a (very loose) Scrum approach
  Only two developers. Iterative approach involving prototyping for six months. Testing and feedback from different tiers of users ("golden" users, PWG conveners, analyzers)
- Social and technical aspect of coding also discussed
  Easy to share and test new features before merging them. Many young members see GitHub as the place where to store their code portfolio, which will be useful for their future endeavors
- Mitigate the inevitable quirks of a large migration
  Docs released early for feedback. Wide scale adoption will require an effort: the doc shows how the new features will improve user's life. Learning curve (quite quick) but for a good cause

Lots of work behind the scenes/1

- Lots of development and demonstrators to make the transition one click away during the transition day: if everything works you don't need to know the details!
- Iterative rewriting of the documentation
  A quite diverse community proofread and tested the documented information
- Parametrized procedure to slim & transfer repositories to GitHub
  Large files removed from Git history: repeated tries while we were discussing the actual policy
- Robust infrastructure and monitoring to ensure quality of service
  Pull request checker and builders backed by tools with auto restart in case of node failures
- Retain current egroups and map GitHub accounts with CERN usernames
  Mapping is one click away for users, and egroups are updated instantly (faster than old Git)

Lots of work behind the scenes/2

- Lots of development and demonstrators to make the transition one click away during the transition day: if everything works you don't need to know the details!
- Optimized integration with an external (GitHub) service

  Effort to make the bot respond snappily to pull request events (approvals, tests…)
- Ensure that daily tags still have predictable content in spite of the delay introduced by tests

  We wait for all approved pull requests opened before 4pm to be tested before tagging
- Developed a solution for large files based on EOS and CVMFS

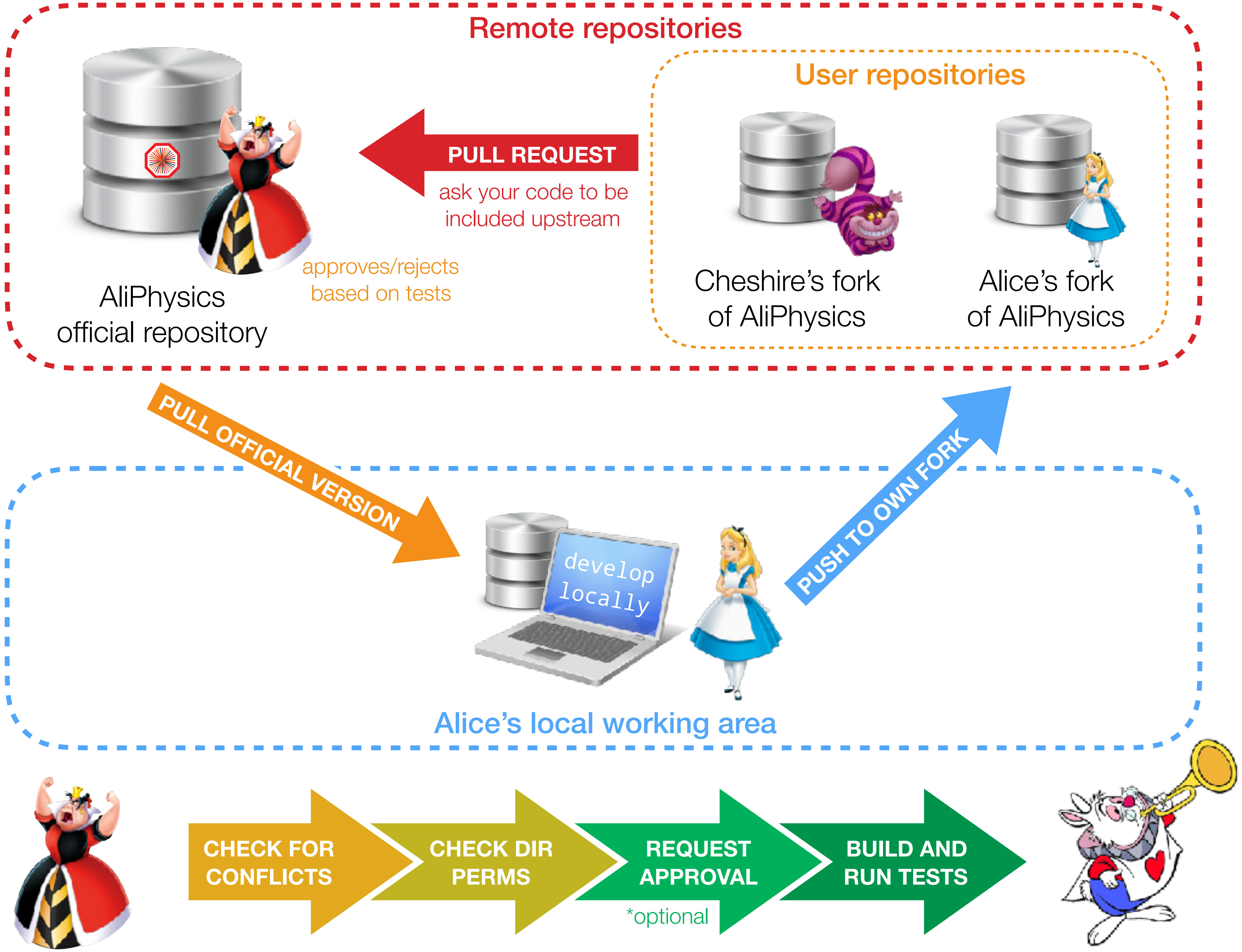
  Easily accessible for all users with a transparent and documented interface
- Everything backed by our tools and infrastructures we have been working on for 1.5 years

  Powered by Mesos, aliBuild and the release validation: standing on the shoulders of the giants

Similarities and differences

- Technical and practical aspects were discussed at the 🔗 analysis tutorial on March 10
- What remained the same
  - Daily AliPhysics tags, analysis trains, Monte Carlo productions…
  - Local builds (whether you use aliBuild or not)
- What has changed
  - "Push" code no longer immediate: may need approval, and must pass build tests. Procedure is clear and gives email notifications on every step
  - OADB: some large files moved away from Git: pushing them to Git is rejected if > 20 MB. Alternative in place based on EOS and CVMFS with transparent interface
  - Privately licensed generators removed: external packages, procedure is transparent with aliBuild, will be seamlessly loaded on the Grid (no JDL change)

Overview of the new GitHub workflow



Remote repositories
User repositories
PULL REQUEST
ask your code to be included upstream
AliPhysics official repository
approves/rejects based on tests
Cheshire's fork of AliPhysics
Alice's fork of AliPhysics
PULL OFFICIAL VERSION
PUSH TO OWN FORK
develop locally
Alice's local working area
CHECK FOR CONFLICTS
CHECK DIR PERMS
REQUEST APPROVAL
*optional
BUILD AND RUN TESTS

- Users have personal forks on GitHub
  - Pull from central repository
  - Do local development on laptop
  - Push to personal fork
  - Issue the Pull Request
- Pull requests handled by a bot
  - Check if code can be merged
  - Run quick tests (permissions, file types, etc.)
  - Run complex tests (e.g. build AliPhysics against tagged AliRoot)

Test phase timeline

- Progress and feedback was reported on 🔗 ALIROOT-7127
- **Jan - Feb:** “alpha” version developed and tested among selected users
- **Feb 20 - Feb 27:** “beta” version opened to the Git working group and PWG conveners
- **Feb 28 - Mar 7:** implemented feedback from the Git working group and PWG conveners
- **Mar 8 - Mar 15:** “release candidate” opened to the whole collaboration for testing
- **Mar 10 (1pm Geneva time):** analysis tutorial dedicated to the new GitHub workflow
- **Mar 15 - Mar 21:** implement feedback from the collaboration
- **Mar 22:** freeze AliRoot/AliPhysics, commence migration operations and testing
- **Mar 23:** migration complete, declaring victory

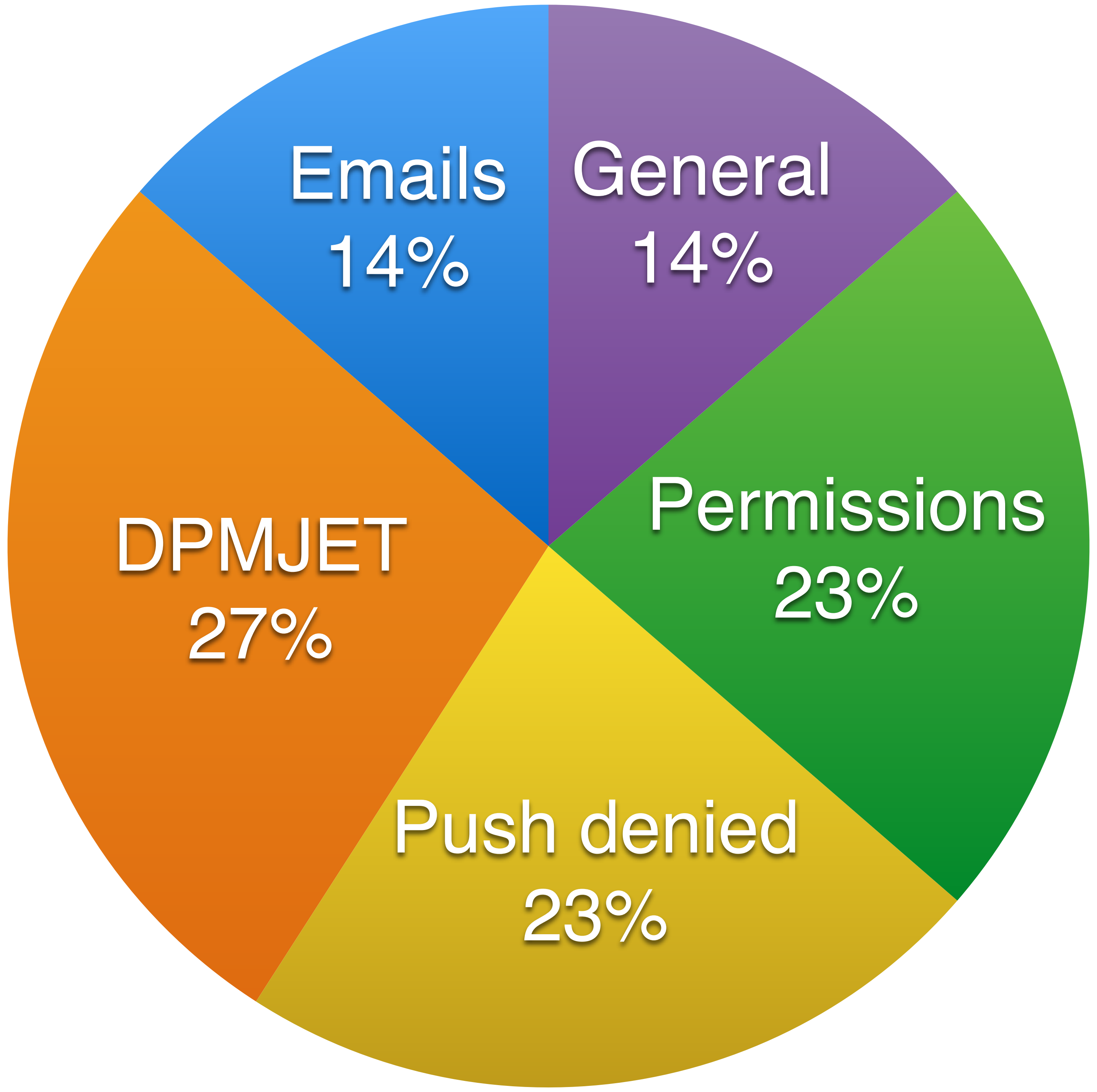
The migration day (and the morning after)

- Based from our previous experiences (*i.e.* AliPhysics split from AliRoot) we prepared a 🔗 pre-flight checklist and discussed it in advance with no upheaval
- All operations were scripted in advance (just clicking buttons). By March 23 afternoon:
  - Build and deploy new AliRoot/AliPhysics (GitHub + aliBuild recipes) 👍
  - Build and deploy old AliRoot/AliPhysics (frozen on CERN GitLab + recipes) 👍
  - Run daily tags automatically 👍
  - Contribute to new AliRoot/AliPhysics via pull requests and automatic tests 👍
- What took time (and finished by March 21 around noon):
  - Running a release validation 👍
  - Running test trains 👍

Problems encountered

Problems encountered (for real!)

- No major problem encountered. Most users were able to work from day 0. A series of minor issues were encountered too
- So far 22 personal/public issues concerning migration received and addressed
  - DPMJET download: 6
  - Push denied: 5
  - Permission issues/doubts: 5
  - Email notifications: 3
  - General questions: 3



- Zero showstoppers preventing users from working

Issues breakdown/1

- **DPMJET download: 6 users**
  DPMJET is private. ALICE users need to authenticate to get it. Impairs unattended installations.
  - **Can't clone:** need to connect by browser to gitlab.cern.ch at least once in order to be able to git clone *any* GitLab repository. Moreover they need to wait at most 1h for sync.
  - **Quick mitigation ("I cannot wait 1h"):** run aliBuild with --disable DPMJET (not ideal)
  - **Avoid inputting password many times:** use GitLab tokens (so you don't need to store your password) and use Git credentials cache (clarified in 🔗 doc)
- **Push denied: 5 users**
  Pushing to the frozen repositories or to the upstream AliRoot/AliPhysics is denied.
  - Pointed to the doc that clarifies that pushes are always to your fork, and you cannot push to the upstream repository any longer: you need to open a pull request instead

Issues breakdown/2

- **Permission issues/doubts: 5 users**
  Permissions were transferred from the old ones and are still based on egroups.

  - **Current model:** PWGXX directories can be written by alice-svn-pwgxx members, who are also approvers for those who cannot push there.

  - **Old perms were broken and were cleaned up:** *e.g.* some users were accidentally admins and could do *anything* - now they cannot commit in their PWG because they're not in the corresponding egroup and they've never noticed (because of old broken perms)

  - **Toplevel PWG permissions are too broad.** Too many people are asked to approve "unknown" analysis code. PWGs should create and maintain smaller egroups and tell us how to reorganize permissions for subdirectories (or even single files)

  - **"I am in the egroup but I am not asked for approval."** In order to be mentioned, we must know the GitHub-to-CERN user mapping. Email was sent to all users who haven't done this yet, and even who does not commit needs to have a GitHub account in order to approve.

Issues breakdown/3

- **Email notifications: 3 users**
  Notifications are handled by GitHub and not by us any longer.
  - Get ~~bothered~~ notified for every commit: tune GitHub settings, explained it in the 🔗 doc
  - Be notified when something that pertains me changed: currently you get notified if you are @mentioned. If this happens too frequently, refactor PWG permissions in order to get notifications only to what you administer (more granularity)
  - Be notified to some specific changes even if I am not an approver: as a proposal we can selectively make users "follow" PRs (needs to be implemented if requested)
- **General questions: 3 users**
  Generic installation problems with aliBuild or genuine doubts about the workflow.

Other useful proposals

- The following requests need some development and we would like to have your feedback.

- Clarify what is the full name of the GitHub user that opened a pull request
  Not everybody out there has got clear usernames (*i.e.* who the hell is ktf?)

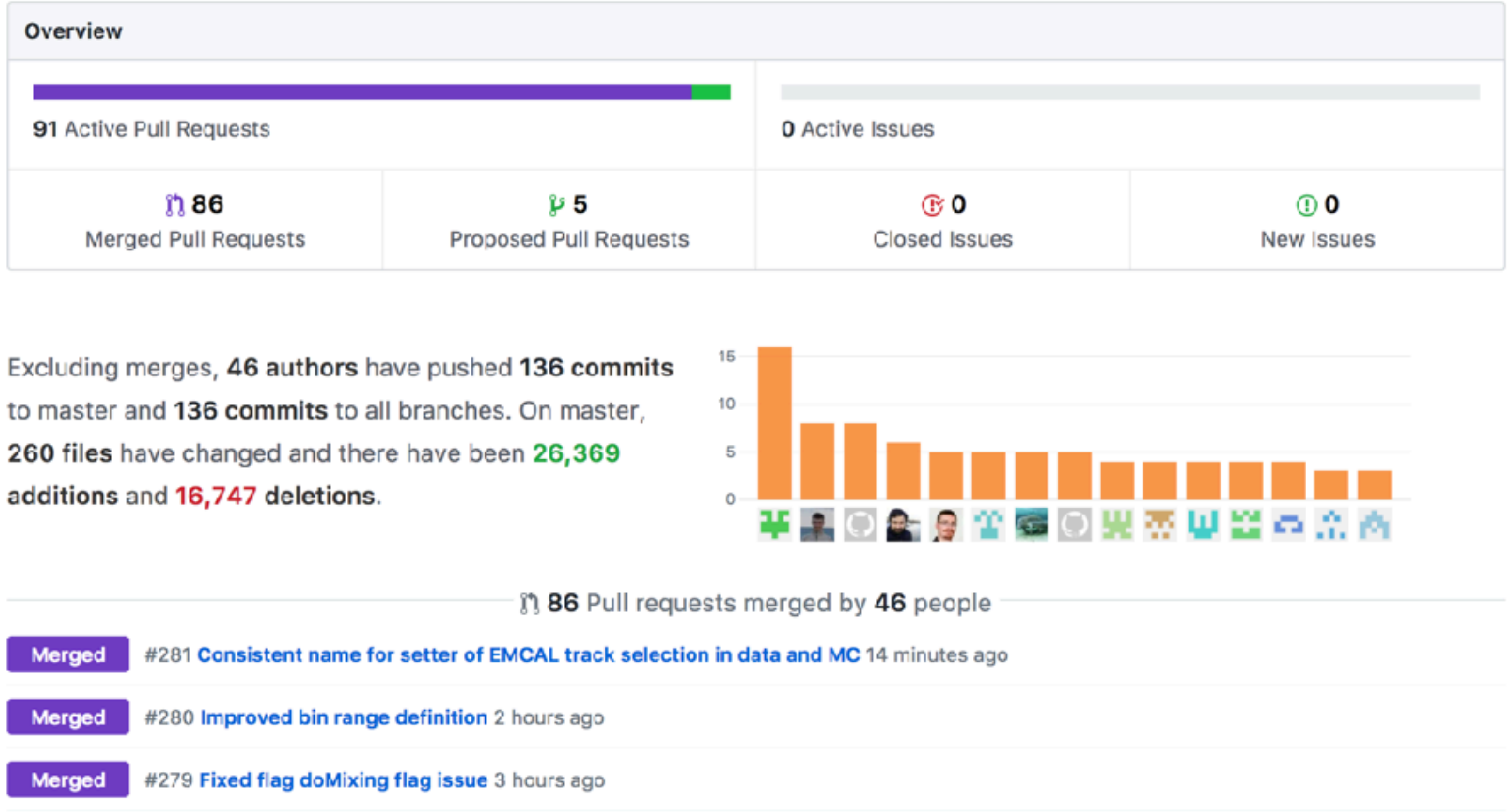
- Automatically start tests for *known* GitHub users
  Reduces turnaround time.

- Automatically close pull requests for *unknown* GitHub users
  Security feature. Approvers will know that if a PR is open it comes from a legit and registered ALICE user. A notification can also be sent that reminds the user how to map the account.

- Enable additional tests
  For instance ROOT 6 tests for AliRoot.

Stats

- GitHub provides us with some stats (the “pulse”). Let’s consider AliPhysics, github.com/alisw/AliPhysics/pulse
- Stats for AliPhysics for the first 7 days:
  - 86 merged pull requests
  - 46 different authors
  - 136 commits (excluding merge commits)
  - 260 files changed



Overview

91 Active Pull Requests

0 Active Issues

86 Merged Pull Requests	5 Proposed Pull Requests	0 Closed Issues	0 New Issues

Excluding merges, **46 authors** have pushed **136 commits** to master and **136 commits** to all branches. On master, **260 files** have changed and there have been **26,369 additions** and **16,747 deletions**.

86 Pull requests merged by **46** people

Merged #281 **Consistent name for setter of EMCAL track selection in data and MC** 14 minutes ago

Merged #280 **Improved bin range definition** 2 hours ago

Merged #279 **Fixed flag doMixing flag issue** 3 hours ago

- The average of ~15-20 commits/day is consistent with what we had before: we might dare to conclude that the workflow was understood by users and the migration was successful

Wrapping up

- The Git working group did a good job

  Only two developers but 43 registered members investing time in testing, writing documentation, providing feedback, and fighting too

- Challenge: make agile development user friendly for the less experienced

  Getting the best from the experiences acquired by others (in particular CMS), and adapting them to our average user. Clarify that all GitHub "complications" (pull requests) are there for a good reason (comments, automatic testing, better code quality)

- Scrum approach…

  …with a hint of anarchy that characterizes our world (and makes us different from the corporate business)

- Our work is *not* over

  What we did is useful for Run 3 too. From now on we will try to listen to users feedback based on field testing our product

Additional information

- Feedback on JIRA
  alice.its.cern.ch/jira/browse/ALIROOT-7127

- Infrastructure monitoring
  alimonitor.cern.ch/display?page=github/combined

- Getting started with GitHub in ALICE
  alisw.github.io/git-tutorial

- Advanced GitHub workflow
  alisw.github.io/git-advanced

- AliRoot on GitHub / old AliRoot on GitLab
  github.com/alisw/AliRoot / gitlab.cern.ch/alice-legacy/AliRoot-legacy

- AliPhysics on GitHub / old AliPhysics on GitLab
  github.com/alisw/AliPhysics / gitlab.cern.ch/alice-legacy/AliPhysics-legacy