

LHC Experiments Feedback to SFT PoW 2024

James Catmore¹, John Chapman¹, Marco Clemencic², **Giulio Eulisse**³,
Peter Hristov³, James Letts⁴, Elizabeth Sexton-Kennedy⁴, Nicholas Styles¹

¹ Atlas, ² LHCb, ³ ALICE, ⁴ CMS

General remarks

All the experiments would like to say **thank you to EP-SFT** for last year support and the detailed presentations about their plans.

In particular, we all thank **Pere** for his leadership over the years and we congratulate **Gerri** for his new role!

Single day meeting works much better than multiple ones over several weeks.

Being able to provide **feedback early** is much better than before Easter.

Monday is complicated for at least 2 of the 4 experiments. If expectation is that people from experiments should join full day, then some more discussion is needed.

While it's good to have an **optimistic spin on future plans**, it would be good to have some **critical look at the past** to identify places where previous plans did not work out as expected and **identify lessons learned**.

ROOT

All the experiments would like to **thank Axel** for his work over the years and **welcome Danilo** as new project leader.

As a general remark **ROOT remains a central** component of our software stacks.

All experiments welcome **RNTuple** prioritization in the plan. A few critical features:

- Copy-less bulk reading (ALICE). **Performance is the feature** sometimes.
- Evolution path for **polymorphic types** (CMS, see also November plan at ROOT workshop)
- **Unaligned friends** (LHCb)

ROOT 6.32.0 release should be early (i.e. CMS: August, LHCb: September, ALICE: November) to be adopted for 2025 datataking.

Security of the Web GUI: it would be good to see explicitly mentioned a path to a more secure solution (e.g. the one Jupyter uses).

Everyone feels that **Windows support should be descoped** to “Community Effort”.

Everyone welcomes the rebasing on clang-repl. In general, **reducing customisation on top of CLANG / LLVM** is very welcome.

ALICE: **memory overhead** of cling should still be on the radar and it would appreciate progress on the **TBufferFile 1GB limit**.

Simulation

All experiments **happy with their feedback loop with Geant4**. Detailed feedback will be given through the usual channels. Everyone is using / looking forward to use Geant4 11.2.

Threading model changes in Geant4 should be presented early and discussed together with the experiments.

Everyone is interested in **GPU related developments**. Particular interest in G4HepEM, Adept & Celeritas integration.

Everyone is looking forward improvements in VecGeom.

CMS: **parallel initialization** before the start of a run would be very welcome.

ATLAS: We are glad to see that the improvement of the **ParticleHP** physics is on the plan of work for this year.

LHCB: happy to see will to integrate experiments validations in **geant-val**, but it has to go beyond “test beam” data.

ALICE: we welcome Geant4 / **FLUKA** integration. It will allow us to streamline / remove our abstraction layer.

CERNVM(FS)

All the experiments are in general **happy with the performance and developments of CVMFS**.

CERNVM (i.e. the VM part) effort should be kept to a minimum.

ALICE / LHCb are very interested in the **ephemeral publishers**. This might also indirectly solve CMS need for non-x86_64 publisher (see below).

ALICE: Particularly glad the **issue with file descriptors** was fixed. [3066](#) still open though.

CMS:

- Deployment support for **multi-architecture** containers under /cvmfs/unpacked.cern.ch, so that we can build multi-arch containers with /cmssw/el8:latest resolving automatically to the arch-specific container.
- Support for native non-x86_64 publishers, e.g. CVMFS publishers for aarch64 and ppc64le, which would allow us to install our packages without going through the qemu, which is slow.

Stacks

LCG Releases:

- **ATLAS** very happy with the prompt and efficient collaboration with SFT in the preparation of LCG releases.
- **LHCb** hopes for a lcgcmake replacement.
- CMS and ALICE do not use them, nevertheless they are happy to participate in LIM.

Spack:

- **LHCb: looking forward to** testing Spack to replace lcgcmake. If proven to be desirable in SFT, they will start using it ASAP.
- **ATLAS: happy to experiment** with it, but not baseline for now.
- **ALICE:** not interested until the missing “**developer story**” and “production HEP stack story” is there.
- **CMS: still valid feedback** given in detail in [March 2023](#) (“not the right tool for packaging large software stack”).

Key4HEP:

- **Not used as “turnkey solution”**. Some **components are** / will be used though (e.g. **DD4Hep, Gaudi**).

Vision for new ML/AI effort

Too early to provide concrete feedback.

Effort looks like going in the **correct direction**, do not reinvent the (Open Source) wheel, but provide **easy interfacing in ROOT for the already existing tools (and viceversa)**.

Ability to **avoid format conversion from ROOT files to do the training** would be a massive advantage (RNTuple reader library usable with major ML packages?)

Inference should be as transparent as possible. In particular:

- analysis steered from Python **and** (particularly) for production-level reconstruction/simulation
- as **few dependencies** as possible
- emphasis on performance and memory footprint

Summary

SFT plays a **central role** in developing and maintaining strategic software components for all the LHC experiments (e.g. ROOT, Geant4, CVMFS, etc.).

Experiments praise the **openness of the planning** effort and they consider the **communication channels with SFT well established**.

Experiments welcome any development allowing us to better exploit **GPUs**.

Experiments look forward to the new **AI/ML project**, and how it will play out.