Achievements in 2024 and First Version of the 2025 PoW

D. Piparo (CERN, EP-SFT) for the ROOT Project

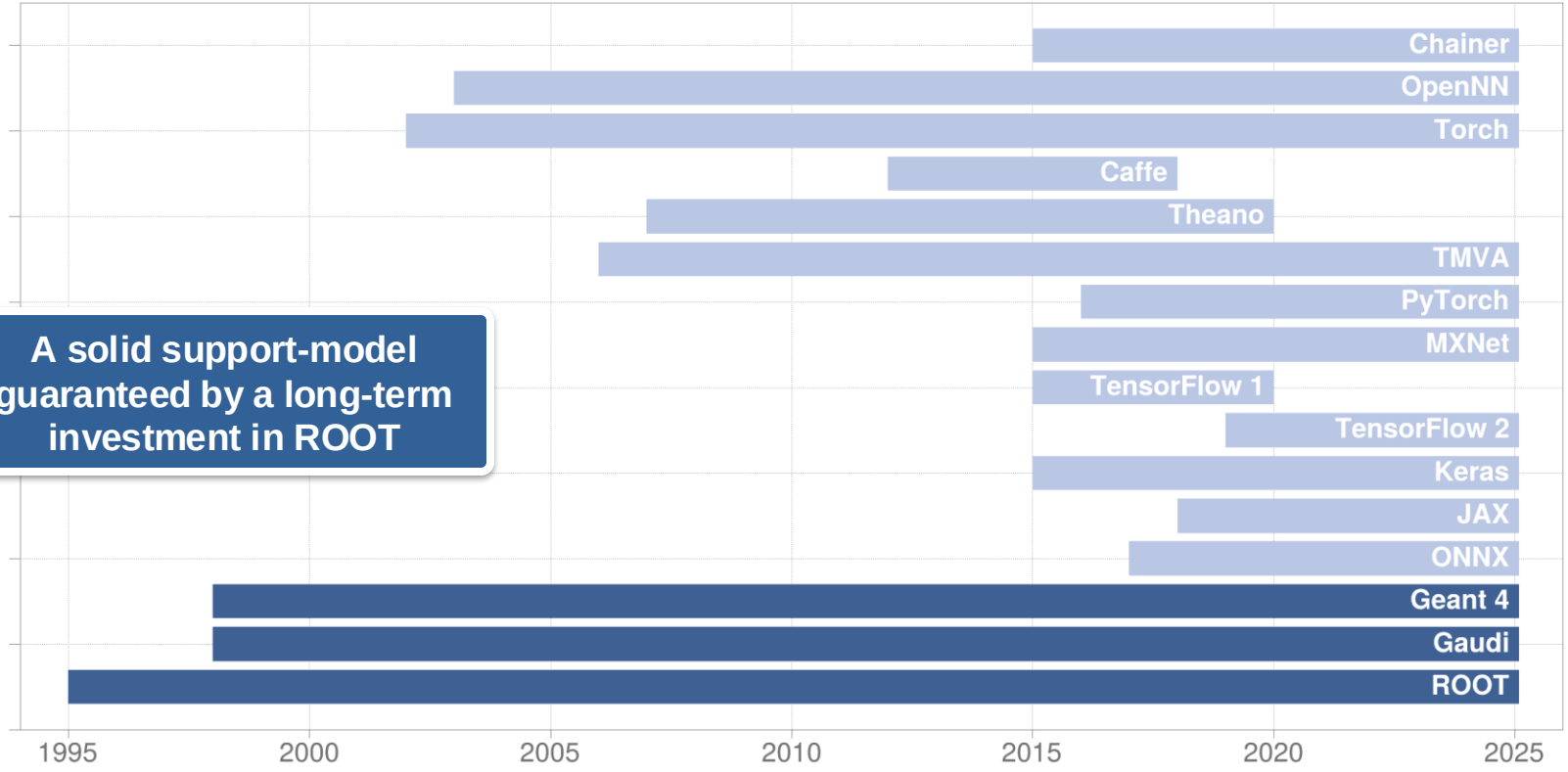
28-1-2025

The Project, Today

Provide a unified software package for the storage, processing, visualisation and analysis of scientific data that is reliable, performant, supported and sustainable, that is easy to use and obtain, and that minimises computing resources and scientists' time needed to achieve results.

The success of experiments and all ROOT users at large is our priority

HEP Common Software Support Timeline



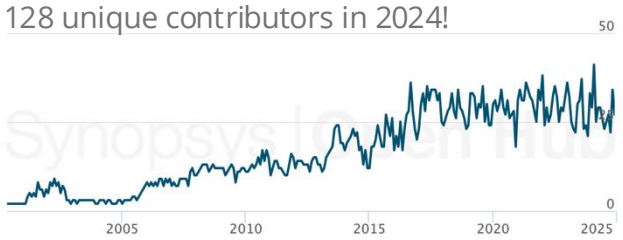
A solid support-model guaranteed by a long-term investment in ROOT

Plot inspired by M. Mazurek

ROOT: An Open International Collaboration

Open-source and Open-development

- On GitHub, LGPL 2.1
- PR based model with public review process
- **Very visible authorship of contributions**



Per unit time: $N_{contributors} \simeq 1.5 \times N_{Core\ Dev\ FTEs}$

For ROOT, the open approach pays off!

Open-planning: https://cern.ch/root-pow

- **The PoW can be influenced with input, active engagement and contributions!**
- Quarterly progress reporting process engaging experiments and funders is in place

Short term internships also supported by:

CERN openlab · Google Summer of Code · iris hep

ROOT: a Platform to make R&D Blossom

Examples:

- Cling (C++ interpreter) and Clang-REPL ('lightweight Cling')
- Automatic differentiation with Clad, now in RooFit
- RNTuple – 6y of R&D, now transitioning to production

ROOT: a project where result-oriented R&D activities can blossom and reach thousands of scientists.

Current prominent R&D Projects:

- HighLO: Project **High** Energy Physics Tools in **L**imit **O**rder Book Analysis (HighLO) applies particle physics methods and tools to financial market data.
- SYCLOPS (EU): Advancing AI/data mining for extremely large and diverse data for Europe and beyond, by democratizing its acceleration through open standards.
- CERN Experimental Physics Department R&D (EP R&D). Topics: innovative I/O formats (RNTuple), Python-C++ interoperability, analysis at scale.

2024 Statistics and some Highlights

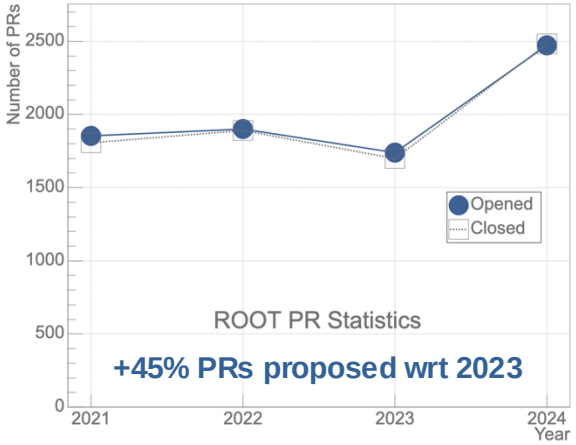
ROOT in the Community: Code, Issues and Support

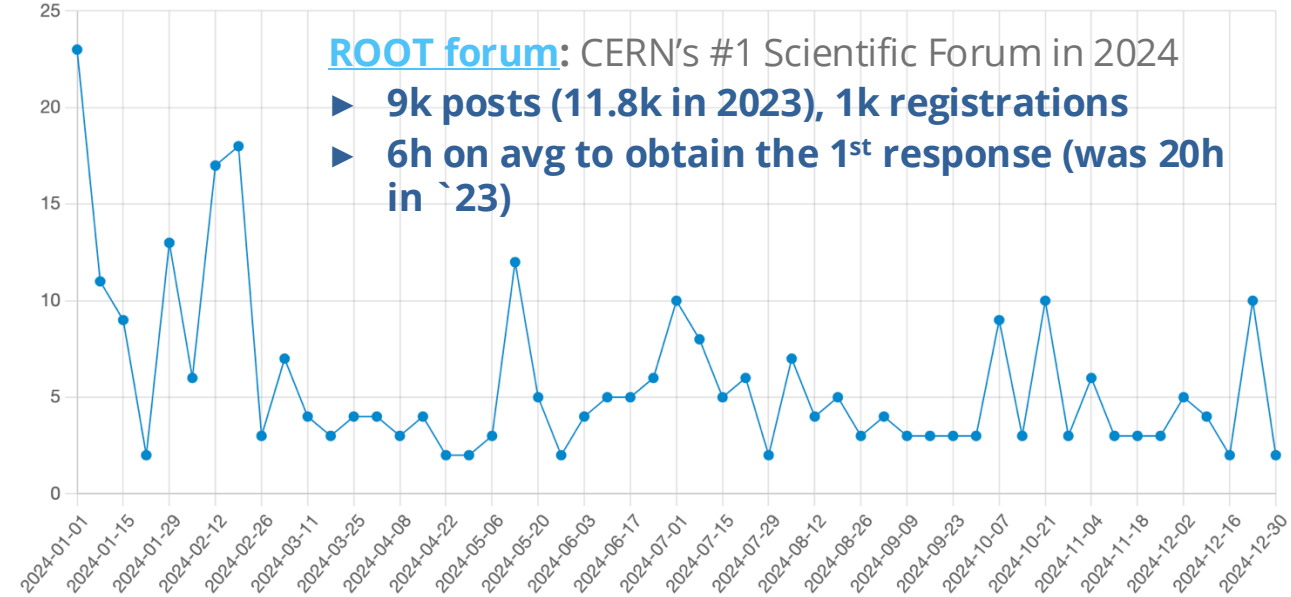
The backlog still there, even if slightly reduced during 2023

Challenge for 2024: while keeping high quality support level, increase development capacity and reduce backlog (see 2024 team composition and size later in the slides)

ROOT 2024 Plan of Work - D. Piparo, CERN EP-SFT - 15-1-2024 7

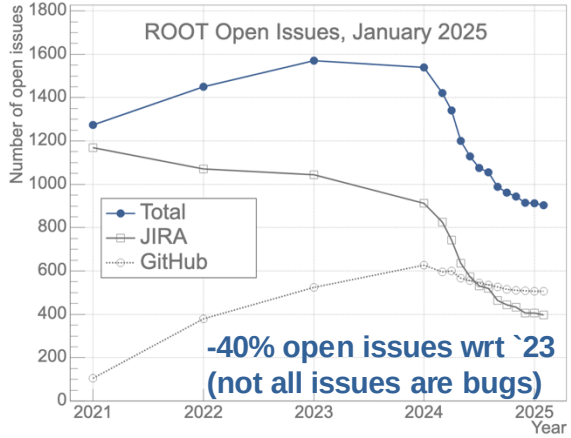
Success!





ROOT forum: CERN's #1 Scientific Forum in 2024

- **9k posts (11.8k in 2023), 1k registrations**
- **6h on avg to obtain the 1st response (was 20h in `23)**



We'll strive for a monotonic reduction of the number of open issues also in 2025, even if a further 40% reduction seems unlikely to achieve given the available effort.

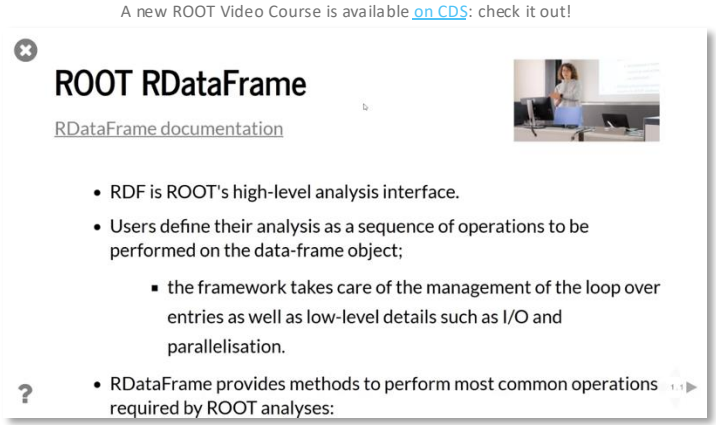
ROOT in the Community: Conferences and Training

ROOT was very present on the international stage, thanks to the support of our FAs

- 3 at ACAT (1, 2, 3), 1 at ICHEP, 9 (2 plenaries) at CHEP (all linked here)
- Workshops, e.g.WLCG/HSF (plenary, parallel), PyHEP-dev, and hackathons…
- … Plus several contributions about ROOT, not authored by core developers

We scaled up training: ~300 participants

- *CERN Summer Student Course* ~200 participants, 5 instances
- With IRIS-HEP & HSF: *Python for analysis course* ~90 participants, 3 instances
- **We took the opportunity to renovate documentation and training material**



New in 2024: ROOT Hackathons

- **Focus on specific topics, for 2-3 days, at Idea2**
- ~30 participants: students, senior scientists, CERN, universities, companies, Europe, Asia, America
- 1st Hackathon, February, fix as many issues as possible: **>90 issues closed**
- 2nd Hackathon, November, documentation, code examples modernisation, Python interface: **>80 PRs integrated (!)**
- **Extremely successful, beyond our initial expectations!**

Significantly advance the PoW and connect with our community

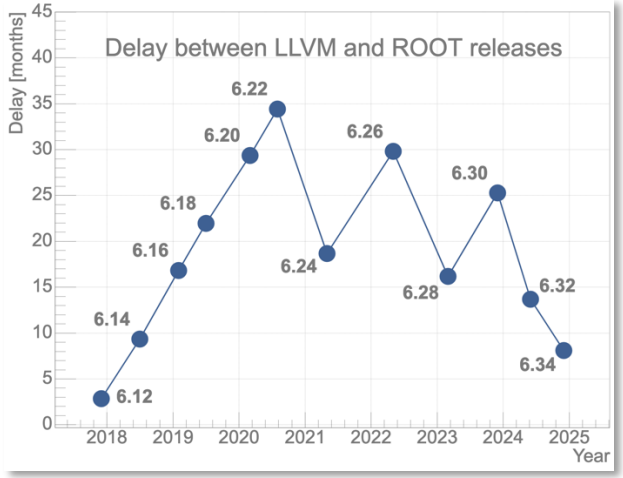
More information about the Hackathons can be found on the 1st and 4th quarterly reports

ROOT continued to be integral part of the Python Ecosystem

- Updated to the latest version of Cppyy, ROOT's C++-Py 'interoperability engine'
- Provided a demo infrastructure to `pip install ROOT`
- Improved the usage of several classes from Python through "pythonisations"

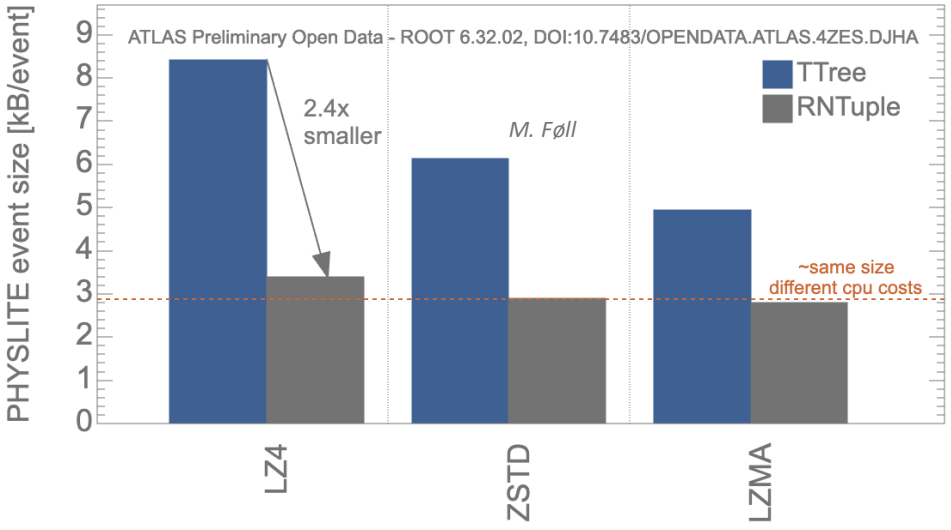
Python and C++ Interpreters:

- **LLVM-based Cling C++ interpreter and the Python compatibility layer simplified**
  - E.g. through CppInterOp, that exposes APIs from Clang and LLVM in a backward compatible way
- **LLVM version from 16.0 to 18.0 updated**



Also with the support of NSF grant OAC-1931408

RNTuple: Some Results



Faster, smaller files, more scalable and sustainable

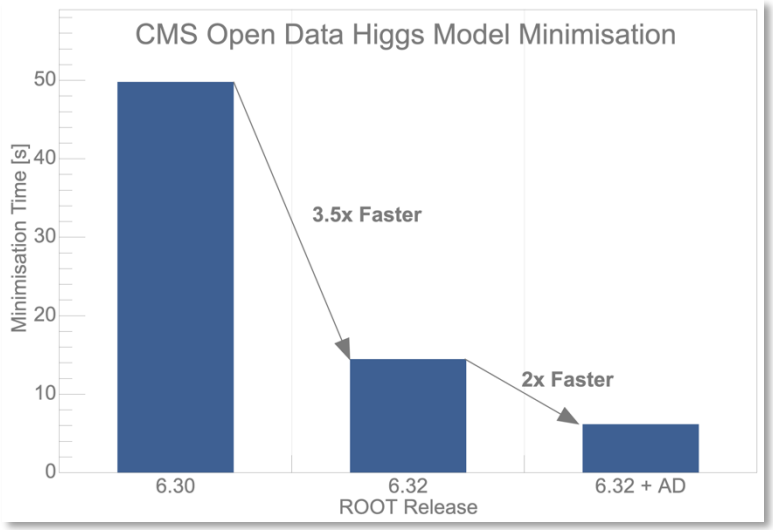
- RNTuple 1.0 delivered, integrated with experiment stacks
- We solved enough problems that we have **confidence we can fix the remaining ones, too**

See the RNTuple Workshop in December

RNTuple underwent an API formal Review by HEP-CCE (Center for Computational Excellence). We are implementing the very useful recommendations which were formulated.

- **Automatic Differentiation (AD) in RooFit with Clad**, in release 6.32.06
  - Improved numerical stability + reduced runtime
- RooFit runtime greatly reduced during the last few years, not only by AD

AD: R&D that became production grade, now available to the entire HEP Community



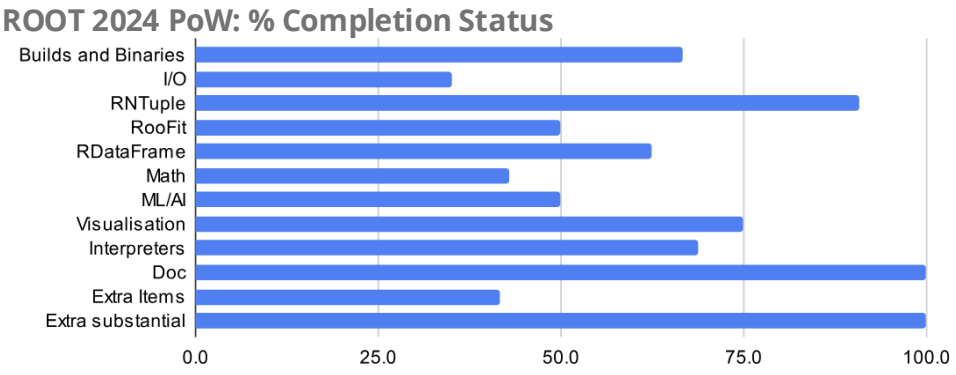
Also with the support of NSF grant OAC-2311471

2024 PoW Completion

Builds and Binaries	1	pip install ROOT for some selected platforms	1	1	
	2	Complete transition to GH Actions for builds, adding GPU runners	1	1	
	3	Reduce number of services hosted by root.cern with a combination of CERN IT central services	1	0	
	4	Win: Replace Debug builds with ReleaseWithDebInfo in the CI	1	1	
	5	Optimise dictionary dependencies to minimise build real time	2	0	
	6	Win: Add support for Ninja	2	1	67 %
I/O and TTree	1	Support std::variant, both in TTree and RNTuple (CMS)	2	0.5	
	2	Support writing objects larger than 1GB (TBufferFile > 1 GB, ALICE)	1	0	
	3	Complete schema evolution improvements	2	0.5	
	4	Ensure consistency of std::int types across ROOT I/O	2	0	
	5	Address residual scaling issues with MT writing	2	0.75	35 %
RNTuple	1	Complete implementation of merging	1	0.5	
	2	Complete implementation of datasets chains	1	1	
	3	Limit testing in collaboration with CERN IT	1	1	
	4	Follow-up on API review by HEP-CCE	1	1	
	5	Implement unsplit ("blobbified") encoding	1	1	
	6	Support for unaligned friends and joins	1	1	
	7	RNTuple: schema evolution	1	0.5	
	8	Further develop support for lossy compression with low-precision floats	2	1	
	9	Design compression dictionaries and understand implications for the specification	2	1	
	10	First implementation of highly-scalable parallel writing	2	1	
	11	Organise a Design Workshop to discuss intra-link events, metadata, native SoA layout for events	2	1	91 %
RooFit	1	Workshop with Experiments: promote features, gather input, speedup integration of RooFit in the existing sw setups	1	1	
	2	Numeric integrals in n-dim with CUDA	1	0	
	3	Evaluation of custom user functions in CUDA	1	0	
	4	Group similar PDFs to speed up evaluation	1	0	
	5	Make the new vectorized CPU likelihood evaluation interface the default	1	1	
	6	Reduce JITting time for AD in RooFit	1	1	
	7	PyROOT: express RooStats configuration with C++-oriented Set" as kwargs	2	1	
	8	Integration of Fumili in RooFit	2	0	50 %
RDataFrame	1	Put existing bulk processing in prod	1	0.5	
	2	DistRDF: reduce memory usage on HTCondor Workers	1	1	
	3	DistRDF: improve user experience when integrated with notebooks and nb services like SWAN	1	1	
	4	Make the TTree → RNTuple transition transparent for analysers	1	1	
	5	Further Pythonise the interface	2	0	
	6	Deliver varied snapshots	2	0.25	63 %
Math	1	Python interface: better histos and graph interoperability with NumPy and UHI protocol	1	0.25	
	2	Histos: advance current RHist implementation to one testable by experiments	1	0.25	
	3	Add interface to pass initial error values/cov matrix to Minuit2	1	1	
	4	Release a library for Lorentz vector computations on accelerators in SYCL	1	0.5	
	5	Deliver plan and prototype of algorithmic improvements when dealing with param constraints in ROOT's minimisers	2	0	
	6	PyROOT: Pythonise TF{1,2,3} and numerical algorithms interfaces (e.g. minimisers)	2	0.5	
	8	Histograms: Model and prototype of pipelining GPU histogram filling	2	0.5	43 %
ML/AI	1	Put RBatchGenerator in production	1	0.5	
	2	Consolidate RBDT	1	1	
	3	Support of integration of SOFIE in experiments Fast Simulation pipelines	1	0.5	
	4	Add support in SOFIE for NVidia GPUs in CUDA	1	0	
	5	Continue to add support for the ONNX operators requested by experiments	1	1	
	6	Make HLS4ML interoperable with SOFIE	2	0	
	7	Streamline ROOT's inference interface, making it able to use models for Python ML fwks (e.g. Keras/TF) directly	2	0.5	50 %
Visualisation and UI	1	Automated placement/tune of plot elements, "Auto Style"	1	0	
	2	Add missing features of classic graphics to the web-based one	1	1	
	3	Automate web-based graphics test suite	1	1	
	4	Add residual missing TEve features to REve, e.g. digit visualisation and text elements overlay	1	1	
	5	Visualization of flat ntuples using predefined visual summary data structures	1	0.75	
	6	Improve REve window manager and browser, polish render engine	2	0.75	75 %
Interpreters	1	Cling: identify potential Cling codebase reductions through the reuse of parts of clang-repl	1	1	
	2	Cling: cppyy rebase on top of cling/clang-repl	1	0.25	
	3	Migrate PyROOT to the latest Cppyy	1	1	
	4	Cling: Prototype SYCL support	2	0.5	69 %
Doc and education	1	Re-evaluate, update, and improve course material, making it more visible and better organised on the website	1	1	
	2	(Re-)evaluate tuts, eliminating what's outdated, newer features would benefit from a (better) tutorial, improve visibility	1	1	100 %
Extra items from experiments	1	Copyless reading in RNTuple - ALICE	1	1	
	2	Physics objects representations out of NanoAOD in RDataFrame - CMS CAT	1	0.5	
	3	Bulk Processing + GPU offloading for distRDF - CMS CAT	1	0	
	4	include the open source Tex Gyre Heros clone of Helvetica in root fonts - CMS CAT	2	1	
	5	Multithreading-friendly interfaces to the histogram types - CMS CAT/TSG	1	0	
	6	A library of matrix operations that can run on GPUs - CMS TSG	1	0	42 %
Substantial extra items not initially foreseen	1	RDataFrame: drastically reduce memory usage to enable very deep computation graphs (O(10K) observables)	1	1	
	2	PyROOT: Improve thread-safety of CPython extension (Fixes Ubuntu 24 and distributed RDataFrame)	1	1	
	3	Maintenance & update of clangdev-feedstock and root-feedstock (the conda-forge repositories for ROOT packaging)	1	1	
	4	RDataFrame: Enable working with missing values (#7713, #9137) (Enabling ATLAS event matching use cases)	1	1	
	5	Upgrade to LLVM 18	1	1	
	6	Add Alma9 ARM runners to CI and address all issues which may result from running on that platform	1	1	100 %

- Keep PM formalism as lightweight as possible
- **Two priority levels: 1 (high) and 2 (medium)**
  - **Assign to each item a "completion score": 0 not done, 0.5 partially done, 1 done.**
  - Work needed to complete each item varies. We assume, for simplicity, that overall it averages out.

2023: total completion 55%, i.e. 31/56 items
2024: total completion 65%, i.e. 48/74 items



Partial Bottomline about 2024 PoW Completion

- **Approximately +50% item capacity wrt 2023 (48 vs 31 items)**
  - Useful measure to calibrate 2025 PoW size
- Completion percentage also increased but not as the capacity: **can we be more efficient? Was the 2024 PoW too ambitious?**
  - Hit by delays not under control in the hiring plan
  - 6 months Staff and 10 months Graduate less than initially planned
- **Some focus areas are less complete than others**. Reasons:
  - I/O: prioritisation of RNTuple delivery (flagship development project)
  - Math: effort lower than expected due hiring difficulties
  - Extra items from experiments: higher priorities elsewhere (also dictated by experiments), effort level lower than expected due hiring difficulties

2025 PoW and Longer Term Vision

1. Support LHC smooth data processing (including the MC chain) and analysis, also addressing experiments' emerging blockers
2. Reinforce ROOT's presence in the Python ecosystem, prioritising Python users' experience
3. Make ROOT installation even easier
4. Design, creation and initial testing of more ROOT 7 building blocks

The 2025 ROOT PoW in a Table, Today

DONE
PARTIALLY DONE
NOT DONE

			Priority	Compl.	
Builds and Binaries	1	Make all 6.3X releases available on Conda (continuous)	**1**	0	
	2	Upgrade the Windows builders to Windows 11 & Add MSVC preview builds in the CI	**1**	0	
	3	Move the ROOT doxy doc generation to the GitHub CI, including its upload for visibility on the web	**1**	0.25	
	4	Decommission the existing root.cern server in favour of a combination of CERN IT provided services	**1**	0	
	5	Make at least one release available for PIP, bringing this distribution channel to a beta for ROOT	**2**	0	5 %
I/O and TTree	1	Enable schema evolution for std::auto_ptr<T> into std::unique_ptr<T> (2024)	**1**	0	
	2	Remove the 1GB size limitation for objects written via TTree and row-wise IO (2024)	**1**	0	
	3	Consistency of std::int types across ROOT I/O (needs changes in TTree I/O) (2024)	**2**	0	
	4	Create a new prototype Experimental::RFile (replacement for TFile) that works smoothly with old and new APIs	**2**	0	0 %
RNTuple	1	Take RNTuple classes out of experimental	**1**	0	
	2	Complete the first coherent set of schema evolution features	**1**	0	
	3	*EP R&D:* Implement a demonstrator of arbitrary combinations of chains and friends in the RNTupleProcessor	**2**	0	
	4	*EP R&D:* Design a first version of RNTuple metadata	**2**	0	0 %
RooFit	1	Numeric integration in n-dim with CUDA (2024)	**1**	0	
	2	*EP R&D*: Evaluation of custom user functions in CUDA (2024)	**1**	0	
	3	Perform analytical minimization of nuisance parameters related to MC statistical uncertainties (upstream from CMS Combine)	**1**	0	
	4	Enable discrete profile likelihood (upstream from CMS Combine)	**1**	0	
	5	Speedup the computation of the Hessian for big Higgs combinations at least by factor of 2 (currently takes several hours)	**2**	0	0 %
Analysis	1	Reach feature parity of TTree and RNTuple processing with RDF	**1**	0	
	2	RDataFrame: enable processing through internal bulk APIs (2024)	**1**	0	
	3	*EP R&D*: Deliver RDataFrame varied snapshots (2024)	**1**	0	
	4	Provide a mechanism to expose objectified NanoAOD preserving lazy reads (2024)	**2**	0	0 %
Math	1	Improve histos and graphs interoperability with NumPy and UHI protocol and write code examples (2024)	**1**	0	
	2	BMBF: Advance current new histo implementation to one testable by experiments, e.g. integrated at a protoype level with rdf (2024)	**1**	0	
	3	Make numerical algorithms interfaces better accessible from Python e.g. minimisers like Minuit (2024)	**1**	0	
	4	Update the documentation for the Minuit algorithm	**2**	0	
	5	*SYCLOPS*: Release a library for Lorentz vector computations on accelerators in SYCL (2024)	**2**	0	0 %
Graphics and Visualisation	1	TScatter2D: Extension of TScatter (4-dim visualisation) to 5-dim	**1**	0	
	2	Implement auto generated GUI for selected REveElement members	**2**	0	
	3	Reduce the time needed and improve user experience of batch image production with web graphics	**2**	0	
	4	Re-write the documentation of the TWebCanvas and RWebWindow classes	**2**	0	
	6	REve client's window manger: undock action to spawn a new window, and increase performance of Geo Browser	**2**	0	0 %
Interpreters	1	*EP R&D* Use CppInterOp to replace internals of TClingCallFunc, most notably the need to JIT strings	**1**	0	
	2	*SYCLOPS* Expose SYCL (prototype) support of ROOT's interpreter	**2**	0	
	4	Review and cleanup whenever possible downstream Clang patches and C++ language extensions	**2**	0	0 %
			overall:	**1**	**% i.e. 0 / 35 items**

https://cern.ch/root-pow

JB, GP, PC

Priority 1:

- Enable schema evolution for std::auto_ptr<T> into std::unique_ptr<T> (from 2024)
- Remove the 1GB size limitation for objects written via TTree and row-wise IO (from 2024)

Priority 2:

- Consistency of std::int types across ROOT I/O (needs changes in TTree I/O) (from 2024)
- Create a new prototype Experimental::RFile (replacement for TFile) that works smoothly with old and new APIs **(ROOT 7)**

Consume debt and prepare for the future

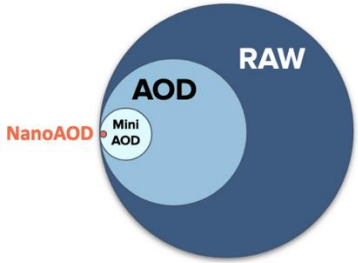
VP, SH, MC

Priority 1:

- Reach feature parity of TTree and RNTuple processing with RDF
- RDF: Enable processing through internal bulk APIs (from 2024)
- EP R&D: Deliver RDataFrame varied snapshots (from 2024)

Priority 2:

- Provide a mechanism to expose objectified NanoAOD preserving lazy reads (from 2024)



Performance and ease of use for analysts

RDF: RDataFrame, ROOT's recommended interface for data analysis

FdG , JB, JH, GP

Priority 1:

- Take RNTuple classes out of the `ROOT::Experimental` namespace
- Complete the first coherent set of schema evolution features

Priority 2*:

- EP R&D: Implement a demonstrator of arbitrary combinations of chains and friends in the RNTupleProcessor
- EP R&D: Design a first version of RNTuple metadata

Carefully navigate R&D into production

* R&D and prototyping has priority 2, unless other factors, e.g. delays, come into play. However, effort provided by "special" funding (EP R&D, EU) is dedicated to specific tasks, irrespective of the priority

*BB, DP, OC, VP**

Priority 1:

- Make all 6.3X releases available on Conda (continuous)
- Upgrade the Win builders to Win11 and add MSVC preview builds in the CI
- Move the ROOT doxy doc generation to the GitHub CI, including its upload for visibility on the web
- Decommission the existing, homegrown `root.cern` server in favour of a combination of CERN IT provided services

Maintenance cost reduction
and ease of installation

Priority 2:

- Make at least one release available for PIP, bringing this distribution channel to a beta for ROOT

** Initials of core developers involved. For the legend, see the backup*

JR, LB, SH, VV

Priority 1:

- Numeric integration in n-dim with CUDA (2024)
- EP R&D: Evaluation of custom user functions in CUDA (2024)
- Perform analytical minimization of nuisance parameters related to MC statistical uncertainties (upstream from CMS Combine)
- Enable discrete profile likelihood (upstream from CMS Combine)

Performance and ease of use for analysts

Priority 2:

- Speedup the computation of the Hessian for big Higgs combinations at least by factor of 2 (currently takes several hours)

JH, JR, MD, SH, ST, VP

Priority 1:

- Improve histos and graphs interoperability with NumPy and UHI protocol and write code examples (from 2024)
- Advance current new histogram implementation to one testable by experiments, e.g. integrated at a prototype level with rdf (**2024, ROOT 7**)
- Make numerical algorithms interfaces better accessible from Python e.g. minimisers like Minuit (from 2024)

Usability and innovation

Priority 2:

- Update the documentation for the Minuit algorithm
- SYCLOPS: Release a library for Lorentz vector computations on accelerators in SYCL (from 2024)

AMT, OC, MT, SL

Priority 1:

- TScatter2D: Extension of TScatter (4-dim visualisation) to 5-dim

Data and Event visualisations
that make the difference

Priority 2:

- Implement auto generated GUI for selected REveElement members
- Reduce the time needed and improve user experience of batch image production with web graphics
- Improve documentation of TWebCanvas and RWebWindow classes
- Improve REve client's window manger: undock action to spawn a new window, and increase performance of Geo Browser

Interpreters

AJ, DS, VV

Priority 1

- EP R&D Use CppInterOp to replace internals of TClingCallFunc, most notably the need to JIT strings

Maintenance cost reduction and innovation

Priority 2

- SYCLOPS Expose SYCL (prototype) support of ROOT's interpreter
- Review and cleanup whenever possible downstream Clang patches and C++ language extensions

An Evolutionary Approach Leading to ROOT 7

ROOT has to evolve to meet the challenges of future scientific computation

- Possible only by **dropping some of its older components or changing some behaviour**
  - E.g. is automatic memory management for histograms and other classes still needed?
- **Not a revolution, but a piecewise renovation**, leading to a completely new system
  - Not new: code using ROOT today has little to do with the one written in early Run 2
- New components are being introduced and adopted *today* or in the near future
  - `RNTuple`, `RDataFrame`, `RFile`. new *Python Interface*, completely new `RooFit`, …
- With the early adoption of new components, **the jump to ROOT 7 will not be large**
  - **Some changes will be backwards incompatible**, but a much smaller jump than ROOT5 →ROOT6
  - Migration or transition paths will be documented and clearly communicated
- **ROOT 7.00.00 will be released during LS3**, well on time for Run 4 MC productions
- A process to regularly discuss ROOT 7 internally is in place: Path to ROOT 7 (PTR7)
  - As time passes, a concrete list of deliverables will be formed

Conclusions

- PoW creation happens through a **consolidated and inclusive process**
- **2025 initial PoW formed**, to be refined with input from users and experiments
  - Bleeding edge R&D items embedded in it
  - Incorporates quantitative estimates about completion, regularly reported to stakeholders
- **One year planning complemented with multi-year vision**
- **Open-source, open-development, open-planning: a strategy that pays off**
  - Will be preserved in 2025: you can influence ROOT's PoW (feedback, advice, suggestions and contributions)

ROOT Users Workshop: 17-21 Nov 2025

- **A welcoming, positive and inclusive atmosphere**
- **An opportunity to shape together the future of ROOT!**
- A venue for ROOT users, world-class experts of scientific computing and the ROOT core team to exchange ideas and learn from each other
- A rich program of presentations, tutorials, and most importantly, discussions
- Strong emphasis on the perspectives of early-career scientists and students

The registration will be possible via the Indico page of the event, in the next couple of months

In Europe
17-21 November 2025
Save the dates!

Backup

Core Team

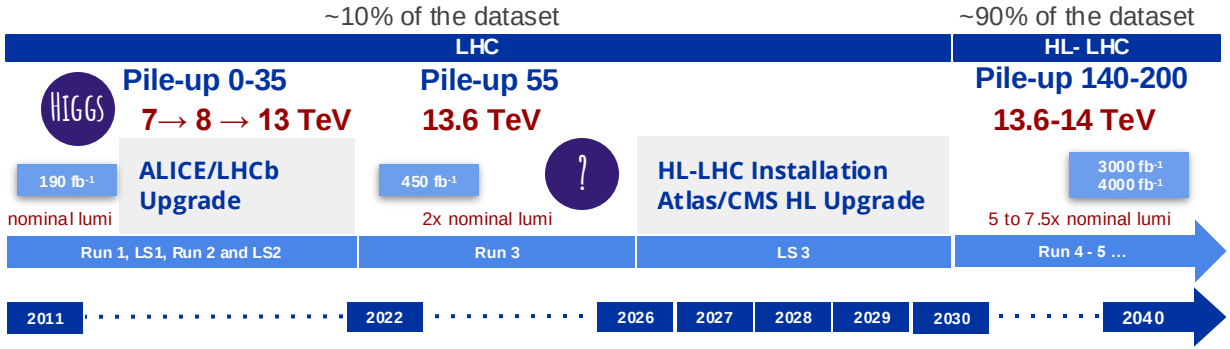
- AJ Aaron Jomy (CERN, EP R&D)
- JB Jakob Blomer (CERN)
- JH Jonas Hanfeld (CERN and Goethe University of Frankfurt, Gentner Programme)
- JR Jonas Rembser (CERN)
- GP Giacomo Parolini (CERN, EP R&D)
- LM Lorenzo Moneta (CERN)
- MC Marta Czurylo (CERN)
- MD Monica Dessole (CERN, SYCLOPS)
- OC Olivier Couet (CERN)
- VP Vincenzo Padulano (CERN)
- ST Silia Taider (CERN – Starting March 1st)
- MV Martin Føll (CERN and University of Oslo)
- PC Philippe Canal (FNAL)
- LB Lukas Breitwiser (CERN, EP R&D)
- VV Vassil Vassilev (Princeton)
- SL Serguei Linev (GSI)
- MT Matevz Tadel (UCSD)
- AMT Alia Mrak Tadel (UCSD)
- BB Bertrand Bellenot (CERN)
- DP Danilo Piparo (CERN)
- DS Devajith Valaparambil Sreeramaswamy (CERN, SYCLOPS)
- FdG Florine de Geus (CERN and University of Twente, EP R&D)

Not all members of the core team dedicate 100% of their time to ROOT. Not all collaborators are listed, e.g. short term internships or secondments from universities. This list also does not reflect the many precious contributions to the ROOT project by our user community.

Stretch Goals

- Further explore S3 backend for RNTuple
- Support for STL Collection of std::array
- Allow users to run hybrid multithreaded distributed applications
- Decide whether CDash represents an advantage and, if yes, integrate ROOT in my.cdash.org or equivalent
- Enable debug info for jitted code
- Add interactive feature to overlay elements in REve(move and resize with mouse drag&drop events)
- Allow users to develop certain (isolated) ROOT components against a pre-built release
- Add weekly CI job for “Win VS-beta”
- Evaluate "Read the Docs" for the Python Interface, e.g. by creating a prototype
- Optimise dictionary dependencies to minimise build real time, #6432 (2024)
- Optimize jitting of virtual calls
- Add support for std::variant in TTree (2024)
- Enable RDatasetSpec usage in distributed RDataFrame
- Introduce Numba pythonizations of RDataFrame

The Need of Strategic Thinking: Storage



HL-LHC timescale: the EIC will also start producing data!

QoS	ALICE	ATLAS	CMS	LHCb	*Total*
Disk [PB]	199	406	304	93	1002
Tape [PB]	283	666	673	250	1875

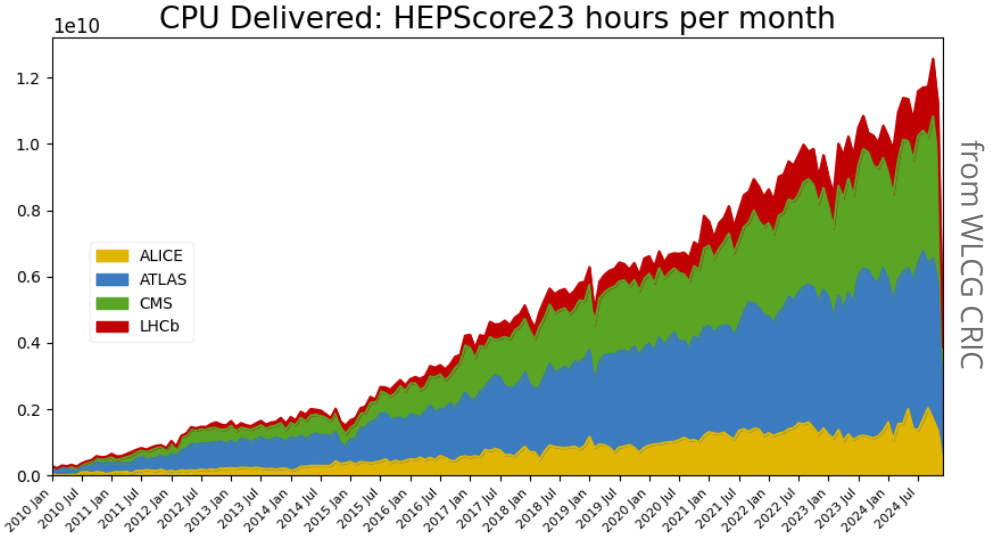
`24 Pledges: source WLCG CRIC

The full exploitation of the physics potential of present and future accelerators also passes through ROOT

LHC Runs 1,2,3 → **Today, >2 EB in ROOT format**

HL-LHC: ~30 EB in ROOT format?

The Need of Strategic Thinking: Compute

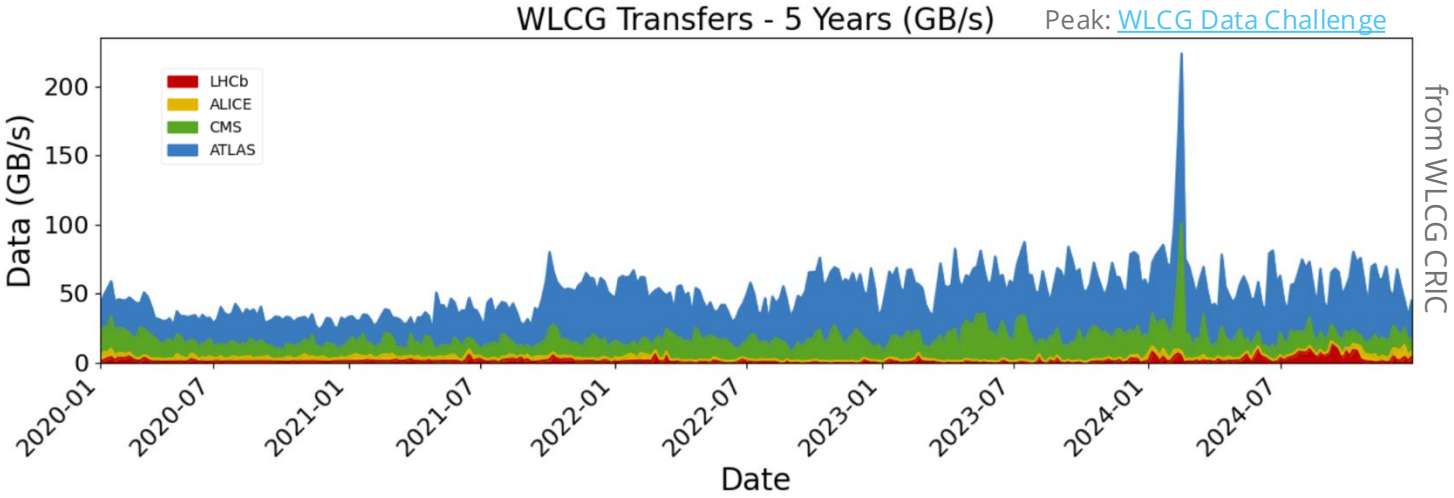


- Approximate compute capacity provided to LHC experiments: O(1M) cores*, running 24/7!
- **Virtually all Grid jobs use ROOT, as part of Experiments' stacks or otherwise!**

Incidentally, most jobs use Geant4, and ~all software is distributed through CVMFS…

Sources: CRIC, HEPiX, RRB Oct 2024 * pledged and opportunistic, approximate calculation performed assuming average HS23 per core from HEPiX tables

The Need of Strategic Thinking: Network

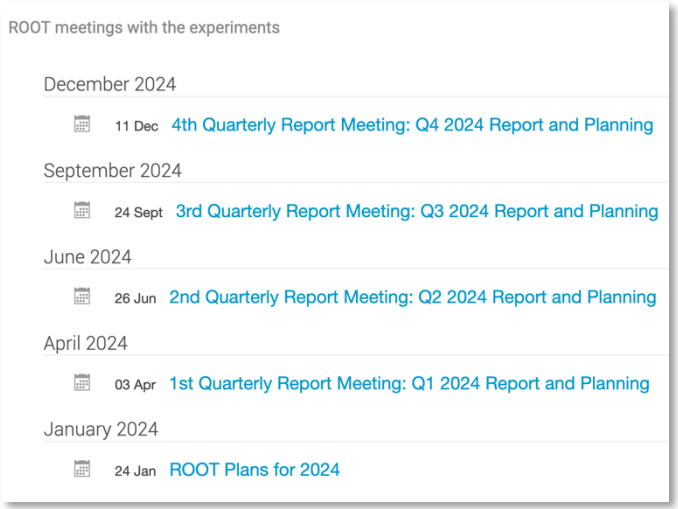


- **Virtually all the traffic is made of ROOT files**
  - With notable exceptions, such as ATLAS/LHCb raw files

Sources: CRIC

Quarterly Meetings with Experiments

- **ROOT, experiments, stakeholders and funders**
- **Schedule and content of releases**, PoW completion and metrics (e.g. issues in the trackers)
- **Priorities and needs**, e.g. to adapt to data taking, analysis or Monte Carlo campaigns of experiments
- Achievement: **agreement with experiments to provide consolidated (e.g. core sw, analysis, trigger, quality monitoring…) prioritised list of needs at each meeting**



Category on Indico

We thank the experiments and stakeholders for their active engagement in this process!

Recap of 2024

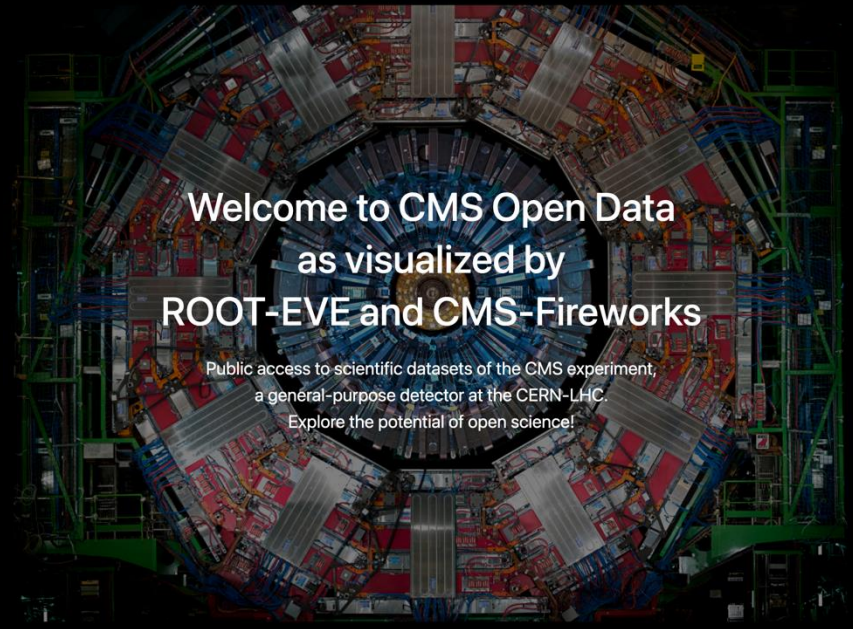
- ROOT 6.32.00 Production Release in May 2024
- ROOT 6.34.00 Development STS Release in November 2024

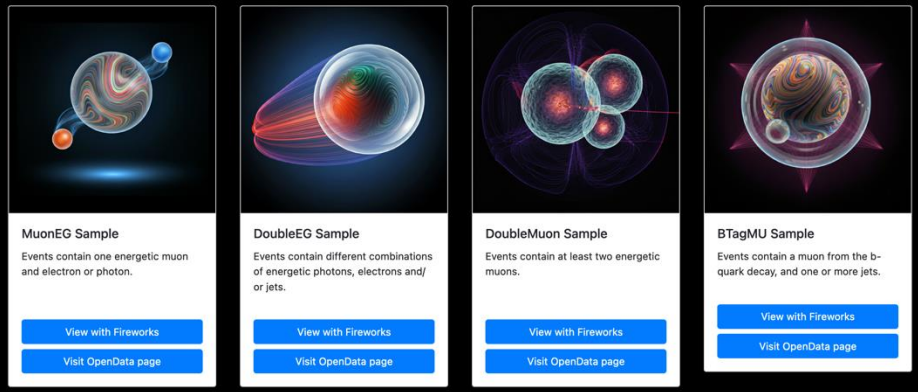
Proposal for 2025: timeline analogous to 2024

- ROOT 6.36.00 Production Release in May 2025
  - Data taking release, long support cycle
- ROOT 6.38.00 Development Release in November 2025
  - Mainly targeting analysis and exposing new features for all users
  - The following production Release will depend on LS3 plans of experiments (Legacy Processing campaigns, large MC productions, RNTuple commissioning…)

ROOT new web-based event display: REve
Currently working closely with CMS OpenData, towards a potential adoption for Run 4

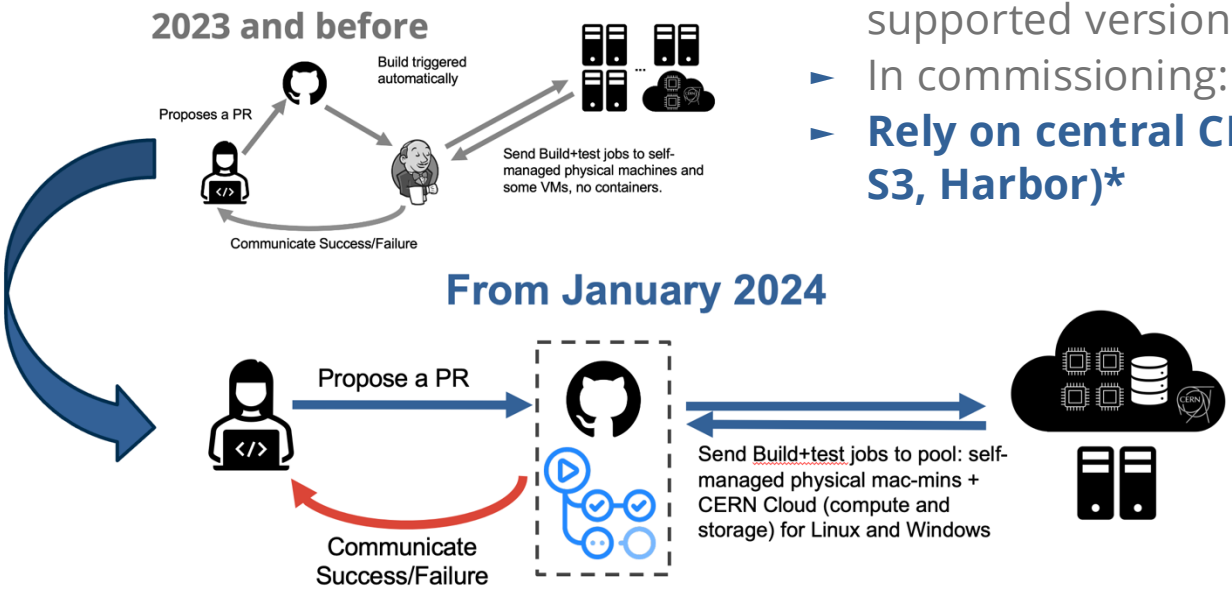
https://fireworks-open.cern.ch





The New GitHub-based CI

- Moved from Jenkins to GitHub Actions
- **Behave like many other open source products**
- **Reduces overall maintenance costs**
- 30-50 build+~3000 tests jobs per day
- Linux ~10 flavours (containers), macOS (all supported versions + beta), Windows
- In commissioning: Linux ARM and GPU nodes
- **Rely on central CERN services (e.g. Openstack, S3, Harbor)***



** Thanks to the CERN IT-CD group for this scalable, powerful and reliable services!*

The Process Leading to the Final PoW

- Sept 2024: PoW 2025 Jamboree
  - Presentations about plans of different ROOT focus areas, prepared and delivered by team members *not* directly responsible for them: incentive to motivate and pre-discuss future directions
- 22 Jan 2025: SFT PoW Day
- 28 Jan 2025: ROOT Funders and Experiments meeting 
- February 2025: Consolidated Experiments' feedback at the AF
  - Last year discussions happened with users and experiments' groups *before* the AF meeting: we are open to this option also this year, if that's helpful.

ALICE already provided some prioritised input for the PoW, and more is coming (thanks!). We discussed internally and decided to wait if other experiments wish to do the same to study a coherent update of the PoW