LHCC Software Report
Geant4 and ROOT Foundations Update, with Experiment Views

Many thanks to all the experiments and the projects, particularly Marc Verderi and Alberto Ribon; Jakob Blomer, Axel Naumann and Lorenzo Moneta; Stefano Piano and Andreas Morsch; Zach Marshall and Alessandro Di Girolamo; Danilo Piparo and James Letts; Concezio Bozzi, Gloria Corti, Ben Couturier and Marco Clemencic

Graeme Stewart and Liz Sexton-Kennedy

2023-03-07

Geant4 contribution to LHCC software review

Marc Verderi & Alberto Ribon, for the Geant4 Collaboration

Geant4 & Experiments, Geant4 Evolution

- **Close collaboration with experiments remains a line of action:**
  - Support to experiments, responses to expressed needs (functionalities, physics, performances, etc.), tight collaboration on specific subjects (eg: ATLAS speed-up), etc.
  - **Joint events:** specific ones – like LPCC simulation workshop (Nov. 2021) dedicated to fast simulation and AI-based one– or general ones –technical forum, HSF simulation working group–

- **Geant4 evolution:**
  - **Software Evolution**:
    - Major release 11.0 (dec. 2021)
      - **New parallelism scheme :** multi-threading (core-bound) replaced by **tasking** (logical layer for tasks, with choice of backend to execute the task, investigated now for seamless bridge for heterogeneous platforms)
      - C++17 as minimum standard
      - *To be noted* : new example demonstrating ML-based Fast Simulation
    - Minor release 11.1 (dec. 2022)
      - New or improved physics capabilities (see after)
  - **Collaboration Evolution**:
    - Actions to address the **"generation transfer knowledge"** challenge
      - Strong warning to funding agencies on vital need for **new, stable, young generation of developers**
      - Creation of an **"Early Career Researcher Committee"**, with representative at Steering Board
      - And also : internal seminars, detailed minutes, WG meetings open to all, etc.
    - Creation of **"contributor"** status:
      - **O(10) contributors** this past and first year; status seems to lower the barrier to join the Collaboration
    - Geant4 website renovated, with design initiated by "new generation"

Highlights of Physics Developments

- Transport of **light hypernuclei & anti-hypernuclei**
  - Request of ALICE
  - Decays and EM interactions provided in G4 11.0; **hadronic interactions** in G4 11.1
- Improved string fragmentation in both FTF and QGS hadronic models
  - To better describe NA61/SHINE experimental data
- Other physics developments
  - Added **new rare electromagnetic processes**
    - Positron annihilation into tau pairs included in G4 11.0
    - Muon pair production by muon in G4 11.1
  - New, **more accurate modeling of ion ionization** : Linhard-Sorensen model
  - **Improved annihilation of anti-baryons** in FTF hadronic string model
  - **Improved gamma-nuclear cross sections** for energies below ~150 MeV
  - Possibility to simulate **quantum entanglement** in $e^+ e^- \rightarrow \gamma \gamma$
  - **Major improvement** in the treatment of thermal neutrons (< 4 eV)

Speed-up of Full Simulation

- Approaches to speed-up the detailed simulation @ no cost on the physics:
  - Special treatment of electron, gamma and positron in HEP applications
  - Reduce the number of steps and/or the work (computation and memory access) done in each step
  - Potential speed-up of O(~10%) for each of these solutions, without compromising the physics
- **Already in production:**
  - GammaGeneralProcess − reduced number of cross sections evaluations for gamma particles
  - **Woodcock Tracking of gamma particles** − avoid to stop at volume boundaries
    - Particularly useful in granular geometries, e.g. ATLAS EMEC and CMS HGCAL (in Phase 2)
    - Unprecedented speed-up of **~20%** observed in ATLAS grid production !
    - Close collaboration between EP-SFT Geant4 (Mihaly Novak) and ATLAS teams is essential !
- Yet to be integrated, tested and validated in the experiments' frameworks:
  - **G4HepEm** − compact, specialized physics library for e- / γ / e+ interactions for HEP applications
  - Custom stepping − specialized tracking (skip general stepping mechanisms for some particle types)
  - Combining Transportation and Multiple Scattering for charged particles
    - Reduce the number of "full" steps, in particular for granular geometries

Simulation R&D

- **Fast Simulation**
  - Implementation of necessary tools within Geant4 for ML fast sim (used in LHCb Gaussino)
    - Integration of inference libraries to demonstrate full ML fast sim cycle within Geant4
    - **As Geant4 example Par04, available in 11.0 release**
  - **Publication of Par04 data on Zenodo**, as Open Data Detector for benchmarking detector algorithms
  - Implementation of a **VAE** (Variational Auto-Encoder) model for detector-readout independent sim
  - On-going work on **MetaHEP** : generic ML fast shower model, able to retrain quickly to new detectors
    - Train once, then adapt quickly to a new detector geometry
- **GPU prototypes**
  - **AdePT** (Accelerated demonstrator of electromagnetic Particle Transport)
    - First prototype for EM showers in calorimeters – using G4HepEm for physics, VecGeom for geometry
    - Main performance bottleneck : current geometry model.
    - On-going work on an alternative, surface-based geometry approach
  - **Celeritas**
    - GPU-focused implementation of HEP detector sim, motivated by recent success in GPU MC (ECP ExaSMR)
    - Version 0.2.0 released January 2023
  - **HSF Detector Simulation on GPU Community Meeting**, 3-6 May 2022

Geant4 human resources & expertise

- Most of the core developers will **not be active during HL-LHC**
  - This is particularly true for the physics part and the kernel ones
- Therefore, **expertise disruption is a serious risk for HL-LHC**
  - Very high risk for physics expertise; moderate for technical expertise
- Risk mitigation and medium/long-term sustainability of Geant4
  - **Need a new generation of developers, competent, motivated and eager to learn**
    - Some expertise can be hard to find
  - They need a **long (> 2-3 years) overlapping period** with the senior experts
    - For a **proper transfer of knowledge** and for **acquiring the needed expertise**
    - **Simply hiring a new person when a developer retires will not be enough**
  - They need secure positions to be committed and present for the long-term
    - **Replacing short-term positions** (fellows/postdocs) with new ones is **not sustainable**
  - In practice, for the HL-LHC, the **main priorities for Geant4** seem to be the following:
    - 1. Electromagnetic physics
    - 2. Physics validation (together with testing and quality assurance) – geant-val
    - 3. Direct contribution of developers in the experiments' simulations

Experiment Views on Geant4

Geant4 status and perspective in ALICE

- **ALICE is a happy user of Geant4**
- Geant4 is now the default transport engine for Run3 and beyond; Currently using version 11.0.p3
  - Use still through the Virtual Monte Carlo (VMC) layer
  - New features in Geant4 often need accompanying changes in VMC (**continued support and contributions from Geant4 experts vital)**
- ALICE full simulation has already undergone major change within the recent upgrade programme and we are prepared for the next years. Gained factor 4-8x with respect to Run2, through
  - Geant4 developments
  - Embedding / signal biasing techniques and improvements in digitization chain
- Ready to run on ARM architectures
- **Important features for ALICE in the next years:**
  - FLUKA hadronic physics inside Geant4 to have alternative physics models for tuning / comparison (might eventually be incentive to drop VMC)
  - (GPU) Acceleration support to speedup transport … preferably inside Geant4 for better and faster integration into existing framework
  - Faster startup:
    - possibility to init complete Geant4-user state from binary checkpoint representation (in certain cases Geant4 needs 10min to initialize physics tables and startup)
    - Faster routines during initialization (energy cut conversion calculations; mapping to regions etc)

Simulation

- Full simulation now ~2 times faster than in 2021.
  - Close collaboration with Geant4 members (embedded in the experiment) was crucial to achieve this.
- Full simulation is validated to run on ARM
- Intending to use Geant4 10.6 for full simulation throughout Run 3
  - Attempts to move to newer version stalled due to unexpected physics changes (10.7 vs 10.6) and poor hadronic physics data/MC agreement (since 10.6 vs 10.1)
  - Will test newer versions (e.g. 11.X), but unlikely to move production before 2025
- Fast Simulation: several years of tuning and validation by multiple FTE are needed to have good accuracy.
  - R&D with generic detector and ML challenges on public data: useful as incubators
  - But the production software will probably be experiment specific (and need significant support inside the experiment)
- Some areas where Geant4 physics modeling can be improved:
  - Jet energy resolution and neutron fluences (for radiation damage and activation studies). Need support from G4 collaboration.

Status and Prospects of Geant 4 Integration

- **The excellent collaboration and direct communication with the Geant 4 project continues.**
- We strive to integrate the latest G4 version in production: cost benefit ratio largely beneficial for us
  - e.g. 2022: Geant 4 10.7, 2024: aim for Geant 4 11.1
  - Recent example of harvested benefit: expected sw performance benefit at no cost for fidelity
- **Successful integration of new G4 versions in CMSSW can happen only with a G4 developer fully engaged in CMS.**
  - e.g. member of CMS, able to build CMSSW, perform basic physics validation using CMSSW software (for example while developing new features).
  - This figure is essential today and in the future (succession plan, new acquisition of expertise are key)
- ML inference in G4: great new feature
  - Not enough to allow for this. Must work within CMSSW (e.g. same inference engine). Transparent usage by the experiment is key, where clear benefits for CMS are demonstrated.
- GPU accelerated simulation R&D
  - Good communication with AdePT and Celeritas projects
  - We would like to see some test integration in CMSSW asap as well as decision points in the planning of the prototypes to clarify the way forward.
  - Augmenting G4 with new GPU features is highly preferable than switching to an entirely new tool.

LHCb Simulation

- Using Geant4 10.6 in production, fully validated for physics in 2022
  - Saw 40% improvement vs previous simulation campaign with Geant4 9.6
  - Boundary processes for reflective surfaces backported from 10.7 (relevant for RICH).
    - Further speedup of a factor of 2 for Run3 geometry
    - Verified no effects on physics results
- Latest version of Simulation stack also built for ARM
  - Pending physics validation and commissioning on production systems
- Exploiting new fast simulation features available in 10.7 in development stack
  - May deploy in production before the end of the year, depending on physics validation
  - Planning collaboration with Geant4 R&D to explore AdePT in new Gaussino framework
- Will start investigating integration of Geant4 11 this year

ROOT Foundation

ROOT Foundation Upgrade for HL-LHC

- **Major I/O upgrade** of the event data file format and access API: TTree ➜ RNTuple
  - Less disk and CPU usage for same data content
    - **10-20% smaller** files, **at least x3-5 better single-core performance**
  - Give access to **novel and future storage technologies**
    - Native support for HPC and cloud object stores
    - Direct disk-to-GPU data transfers
  - Systematic use of checksumming and exceptions to prevent silent I/O errors

- **Generation hand-over** of I/O experts to ensure availability of I/O expertise compatible with the HL-LHC lifetime

- Est. 50 MCHF/year on storage in WLCG
  ➜ strong incentive for **common, highly efficient I/O layer**

Supported by

EP R&D

iris hep

Coordination with Experiments

Approach to adding support for new Event Data Models (EDMs)

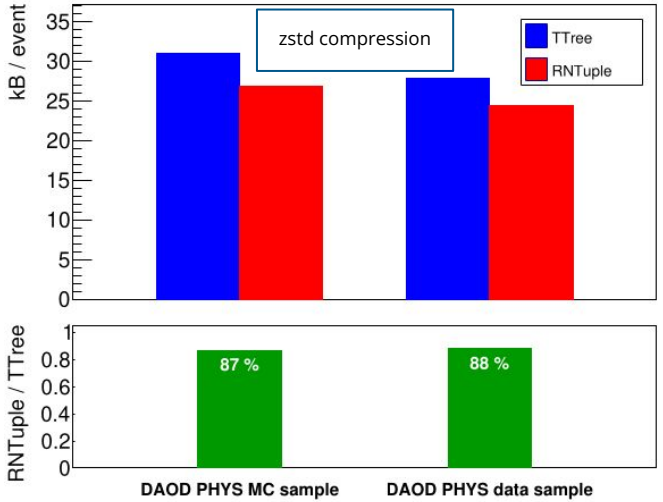
1. EDM representation in RNTuple (import / conversion support)
2. Integration in the experiment framework (writer / output module)
3. EDM R&D: optimizing EDMs for best performance with RNTuple

- The first few EDMs require substantial R&D and are onboarded one-by-one
- Support since 2021: CMS nanoAOD, LHCb final-stage ntuples (import only)

Milestone reached in 2022: Support for ATLAS xAOD (PHYS & PHYSLITE)

- Support for import as well as generation from ATLAS software framework
- Required support for *read rules* and *custom collections* in RNTuple
- Fruitful and very efficient collaboration of ROOT and ATLAS Core I/O teams to support RNTuple

Encouraging first results with ATAS DAOD_PYHS:



Goals for 2023:

- RNTuple support for all ATLAS & CMS AOD variants with initial framework integration
- Planning for onboarding of EDMs of LHCb and ALICE in 2024
- Validation of format forward-compatibility

Goal for 2024: Release of RNTuple 1.0, i.e. promise to keep backwards-compatibility

→ Requires firm understanding of requirements for RECO and AOD formats of all LHC experiments

(RAW formats have little impact because their format is typically stored as experiment-specific binary object in ROOT)

ROOT RNTuple Technical Progress

Milestones reached

- TTree to RNTuple importer
- XRootD support for remote reading
- Lossless compression R&D: compression-friendly data encodings
- Full exploitation of HPC object store performance
- First validation of distributed analysis throughput

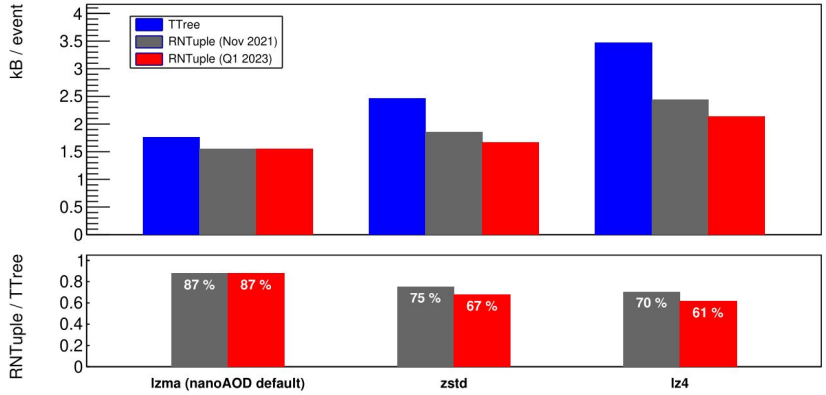
Progress on track

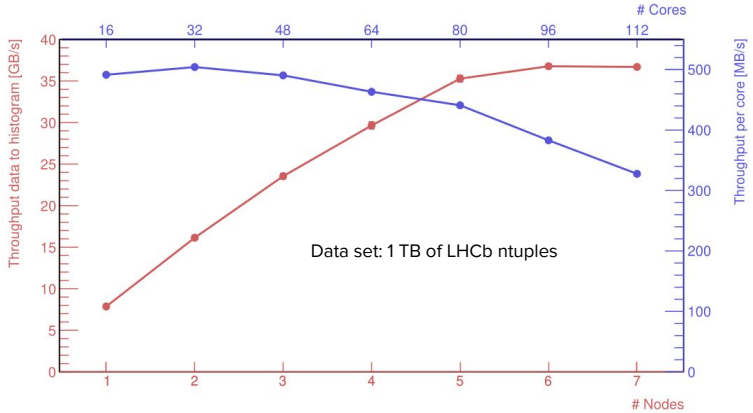
- Late schema extension (adding a column to the dataset being written after the first event)
  - Essential for production-grade AOD writer modules in ATLAS & CMS software frameworks
- (Distributed) RDataFrame integration
- Lossy compression (`double` in memory, `float` on disk)

Moved to 2023/2024

- Schema evolution (initial design ready)
- Meta-data API & data set combinations (pending EP R&D funding)
- RNTupleLite: low-level RNTuple C API
- Validation on large storage pools (pending collaboration with infrastructure providers, e.g. ATLAS cloud R&D, CERN IT)

CMS NanoAOD storage efficiency





First scaling tests:

Distributed RDataFrame with RNTuple, data stored in HPC object storage (Intel DAOS)

RNTuple Funding

Person Power	Funding Source
Staff	EP-SFT
Fellow / Quest	EP R&D
PhD Student	EP R&D
Technical Student	Intel (40%) and EP-SFT (60%)
Technical Student	ATLAS, co-supervised by ROOT
IRIS-HEP Fellow	IRIS-HEP, co-supervision CMS & ROOT

- Pending EP R&D phase II approval
- Past funding

- **Current funding shared by EP-SFT, experiments, and R&D efforts**
  - **Optimization for large compute & storage pools would greatly benefit from joint effort with a large infrastructure provider**
- **Overall development throughput limited by expert staff availability**
- **Expecting stable, if not increasing, I/O workload well into Run 4**

Language Interoperability / C++ Type Description

Basis for I/O *and* Python binding to HEP's C++ libraries

- C++ interpreter uses llvm.org's clang/LLVM as library; Python binding uses C++ interpreter and LBNL's cppyy

ROOT 6.28/00 from February 2023 has upgraded LLVM to version 13, after one year of work

- Required help from compiler engineers, I/O experts, experiments' validation
- Will soon enable C++20 support, much demand from experiments

To reduce cost, increase agility and sustainability: ongoing redesign and "upstreaming" of components

- Clang adopted (rather: re-implemented) ROOT's C++ interpreter cling as clang-repl!
- Cppyy will soon depend only on cling

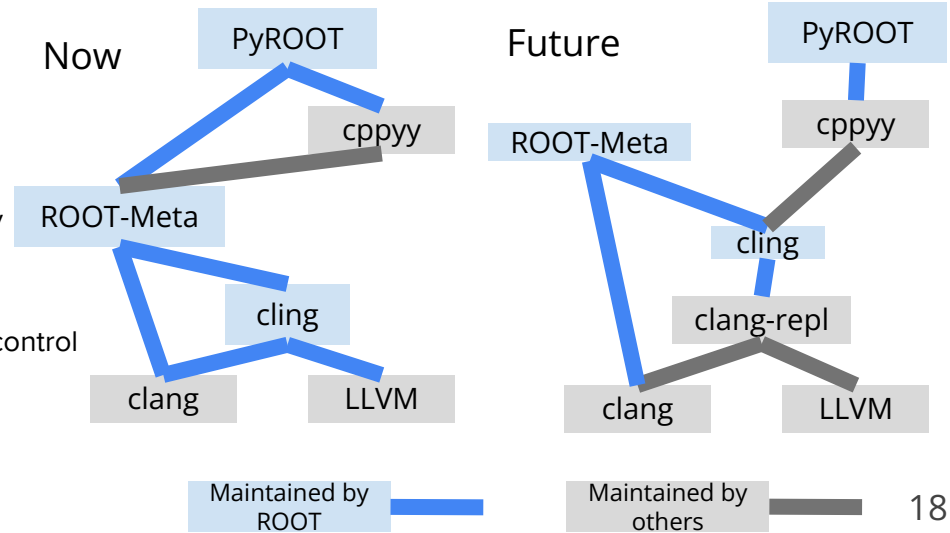
Ongoing, significant re-design of these non-ROOT components

- Requires matching redesign of ROOT's usage (*)
- Sizable expert investment needed (*)

Benefits (*)

- **Much reduced code size maintained by HEP**
- Code previously owned by ROOT now maintained externally
- World experts find solutions to HEP's problems
- Faster adoption of new language standards
- Removing historical baggage: simpler, faster, better feature control

(*) Depends on funding by EP R&D Phase II



Experiment Views on ROOT Foundation

ALICE & ROOT Foundations

- The ALICE experiment uses ROOT for its data and MC storage, in the calibration, reconstruction, simulation and analysis.
- Important developments
  - RNTuple as faster and optimised replacement of TTree, providing also additional compression. Several features are of particular importance in our view:
    - Bulk IO
    - Native support for HPC and cloud object stores
    - Direct disk-to-GPU data transfers
    - Integration with RDataFrame
    - Support for ML pipelines
  - Cling
    - Cling re-implementation using clang-repl
    - Support for new platforms (RISC-V)
- RDataFrame
  - Bulk processing
  - Possibility to use GPUs
- Math & RooFit
  - Development of proper Python interfaces
  - Possibility to use on GPUs
  - Additional minimizers (possibly integration of some Python packages)
  - Automatic differentiation (CLAD) in the optimisation
- ML
  - Further development of SOFIE as replacement of ONNX
  - Integration of ML algorithms in RDataFrame
- M & O
  - Build infrastructure, CI, testing, debugging, optimisation, documentation, user support

ROOT

- Currently using ROOT 6.26/08 for our production software branches
- Several analysis groups interested in using new improvements for statistical analysis in 6.28
  - Currently discussing when this will be used in our releases
- Looking forward to being able to use C++20 features in near future
- Studies ongoing into optimal use of ROOT compression schemes for our use cases
  - Comparing different algorithms, compression levels, and Autoflush/Split Level settings to assess impact on size and speed
- Studies ongoing for ~2 years on RNTuple formats as an alternative to TTree-based I/O
  - Advanced features relied on in TTree (I/O rules in XML dictionary description files, xAOD containers) have been added and others
  - Decorations/schema additions (core functionality for ATLAS => critical feature) bringing RNTuple closer to the required feature set are expected soon
  - Once DAODs (ATLAS primary analysis format) can be written to RNTuple, studies storage needs, I/O speed, etc will be performed

Status and Prospects of "ROOT Foundation" Integration

- **The excellent collaboration and direct communication with the ROOT project continues.**
- We strive to integrate the latest ROOT version in production: cost benefit ratio largely beneficial for us
  - e.g. 2023: ROOT 6.26, 2024: ROOT 6.28
  - Recent example of harvested benefit: reduce RAW event size by 10/15% moving to LZMA compression
- Web-based ROOT (e.g. *REve*) visualisation at the core of the "FireworksWeb" event visualisation
  - Even more important these days to debug and improve Phase-2 detector geometry
- New I/O layer, RNTuple: fruitful collaboration ongoing with the ROOT team
  - Can write NanoAOD in RNTuple format with CMSSW, same layout of current IO (e.g. no change required in existing RDataFrame analysis code)
- **Need to expand the current set of supported event contents by RNTuple**: RAW, AOD, MiniAOD
  - At first with ROOT tooling, e.g. RNTupleImporter converters, next embedded in CMSSW
  - **Must be ready by 2025**
  - Students and other temporary labour dedicated to the task: need more (expert) effort to secure timeline
- More needs, driven by heterogeneity paradigm shift
  - Simple support Structures of Arrays IO (e.g. enhance dictionary generation) already for *today's TTree*
  - Low level tools for numerical operations on GPUs (e.g. small matrices): like TMath, but for GPUs
  - **Mass scale production-tests of RNTuple I/O will have to be performed** in collaboration with providers of large storage pools to complement the small scale testing

LHCb ROOT Integration

We really appreciate the commitment from ROOT team towards experiments

- Planning to use ROOT 6.28 for production in 2023 (as part of LCG 103)
  - Using C++17 for data taking
- We are going to test with C++20 when possible
- Looking forward to use RNTuple, not expecting to complete commissioning before end of 2024
  - with the hope to deploy for 2025 data taking.
- Now depend heavily on ROOT geometry through DD4hep
  - Jsroot and WebGL visualization are of great help in development and debugging of geometry

Observations and Conclusions

- Projects make excellent progress
  - These are bringing significant improvements, minimizing resource needs for HL-LHC
    - Some R&D, such as simulation on GPUs, remains high-risk
  - Physics improvements in Geant4 also very welcome
- Experiments are taking advantage of these improvements and building integration work into their planning
- However, this is not ‘free’ - significant efforts are often required to reap the benefit of new software
  - Physics validation of newer Geant4 versions can be long and not all changes are reflected in better simulation ‘out of the box’ for experiments
    - Note the variety of different G4 versions used in production currently (10.6 x 2, 10.7, 11.0)
  - RNTuple is a major evolution in a key component of our stack
    - Close work between the ROOT team and the experiments is required, with non-trivial efforts to understand and support all use cases
    - Also need to *test at scale now*, with facilities, e.g., CERN-IT, needed to help
- Continued close collaboration between the experiments and the projects is needed
  - Sustaining this effort is very important, building up a new generation of experts in projects and experiments with credible career opportunities
  - Flexible funding to be able to rapidly test new R&D ideas is generally felt to be lacking

Backup - Simulation

Responses to recommandations (1/2)

- **General recommendations:**
  - The experiments have strong relationships with Geant4 that should continue. More coordination and collaboration on fast simulation would be beneficial.
    1) LPCC simulation workshop (Nov. 2021) on fast simulation, including AI-based ones, is along this line
    2) But experience shows that even AI-based fast-simulation remains quite experiment specific
       - Collaboration is essentially through building & sharing a common background on these techniques
  - We urge the experiments to consider whether their existing project management processes are sufficient to address the following:
    1) The (evolving) degree of certainty that the overall project will deliver the required performance in time for HL-LHC.
    2) Recognizing the level of risk to the project from internal and external dependencies, assumptions, and unexpected changes.
    3) Establishing the decision point when choices need to be made, priorities established or changed.
       - The three above points are considered and are part of “routine” management

    Person-power requirements and projections.
       - This remains by far the highest risk for Geant4 –and for the users’ community–, with a high risk of expertise disruption. Funding agencies have been strongly warned about this risk, for Geant4 in general, and for each of the Geant4 working groups.
  - [About additional collaboration mechanisms with experiments] We encourage Geant4 and the experiments to reinforce such collaborative efforts that not only streamline communication but also facilitate sharing effort towards common goals such as, for example, AI-based fast Monte Carlo projects.
    1) Any opportunity to strengthen collaborative efforts are taken, we believe.
    2) For example, but not limited to:
       - LPCC simulation workshop, dedicated to fast-simulation, including AI-based ones
       - Large ATLAS performance increase, obtained thanks to tight collaboration

Responses to recommandations (2/2)

- **Specific recommendations to Geant4:**
  - We note that improvements in the speed of Geant4 do not automatically translate into equal gains within the experiments' workflows and we encourage Geant4 experts to reinforce their involvement in the tuning of Geant4 in the experiments' software stacks to maximize the benefits.
    1) Example of collaboration with ATLAS
  - [About fast simulation] We encourage Geant4 and the experiments to increase coordination on these developments.
    1) LPCC workshop mentioned before
  - In addition, a new role of Geant4 Contributor has been created to facilitate contributions from people who are not full members of the collaboration. We are keen to see if this new initiative adds flexibility and enables new engagement.
    1) About 10 contributors joined this past year
    2) Role looks to indeed lower the barrier for future membership (but need several years feedback for conclusion)
  - It is also crucial that the physics description of the detailed simulation continues to improve, and care should be taken in maintaining a sufficient core developer team with appropriate expertise.
    1) We fully support this conclusion.
    2) This is tightly linked to high risk of expertise disruption mentioned already
    3) And is tightly linked with the appropriate balance between adiabatic evolution and risky R&D.
  - The inclusion of FLUKA models in Geant4 may improve the overall precision but is a time-consuming task. We suggest that the experiments are involved in the prioritization of this activity in the overall project schedule.
    1) This is under discussion, but is also a way to respond to above point of improving the physics description for what hadronic physics is concerned.

Expertise Disruption Risk @ O(10) years

- **In working groups:**
  - Run, Event, and Detector Response — Very High
  - Particles and Tracking — High
  - Geometry and Transport — Moderate (recent hiring at CERN)
  - Generic Processes and Materials — Very High
  - Electromagnetic Physics — Very High for HEP, Low for Low EM
  - Hadronic Physics — High
  - Persistency — High
  - User and Category Interfaces — High
  - Visualization — Moderate
  - Physics Lists and Validation Tools — Very High
  - Testing and Quality Assurance — Moderate
  - Software Management — Moderate
  - Documentation Management — Moderate
  - Novice & Extended Examples — Moderate
  - Advanced Examples — Moderate
- **In general tasks:**
  - Release coordination — Very High / **Certain**
  - General support (registration, web services, etc.) — Very High / **On-going**

- **Funding agencies are key to improve on these perspectives !**

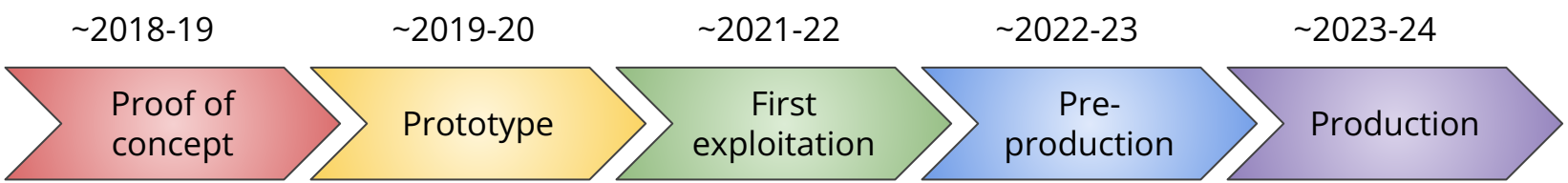
Risk table, as presented during the review

Table 1: Table of high impact risks. The probability of each risk gives the risk itself (as "risk = probability×impact").

Risk	Probability	Risk mitigation
Human resources and expertises risks		
Expertise disruption	• Moderate for technical expertise • High for physics expertise	• Anticipation for long term involvements (> O(10 years)) • Anticipation for large overlap (> 2–3 years) between successive experts
Insufficient hiring rate	• Moderate	• Advertise work interest to externals • Advertise jobs importance to funding agencies
Too high leaving rate	• High (observed)	• Anticipate for retirement case • Secure positions, secure working climate for non-retirement case
Insufficient human resources	• Moderate to high	• Keep human resources at least at the current level • Ensure proper hiring/leaving rate balance under the constraint of sufficiently long overlap between experts
Accuracy, Computing Performance, Technical and Technological Risks		
Insufficient throughput at requested physics quality under HL-LHC computing budget	• Certain if using only detailed tracking on CPU • Certain if using detailed tracking on CPU + EM calorimetry on GPU accelerators • Subject to evaluation when combined with fast simulation (see below)	• Needed but not sufficient adiabatic improvement of detailed physics (both CPU and physics quality) • Needed but not sufficient investment in detailed EM physics for calorimeters on accelerators • Investments in detailed physics on accelerators for specific cases (optics, maybe low E neutrons) • Investment in detailed hadronic physics on accelerators but high human investment cost and performance gain difficult to anticipate • Investment in fast simulation techniques (both Geant4 and experiments) • Evaluation of new architectures (A64FX, M1, etc.)
Insufficient quality of fast simulation at requested coverage	• Subject to evaluation but anticipated moderate if coverage limited to EM calorimetry • Subject to evaluation but anticipated moderate to high if coverage extended to many detector parts (hadronic calorimeter, tracking system,)	• Investment in fast simulation techniques (both Geant4 and experiments) for EM calorimeters • Investment in fast simulation techniques (both Geant4 and experiments) for hadronic calorimeters • Investment in particular in ML technique
Insufficient compute capacity for Continuous / Nightly / Monthly / Yearly testing and validation of new developments to requested physics quality under HL-LHC or other computing budgets	• Moderate	• Investment in CI/Testing infrastructure, corresponding investment in physics expertise to analyse results
Low sustainability and high maintenance cost of material specialized codes	• Moderate to high	• Focus on standards / prevent usage of proprietary solutions • Preserve portability and generality

Backup - ROOT Foundation

RNTuple Development Plan (State Nov 2021)



- ✅ Architecture
- ✅ Review on state-of-the-art
- ✅ First prototypes

- ✅ Adoption in ROOT::Experimental
- ✅ I/O scheduler for local and remote access
- ✅ Performance validation

- ☁️ Object store support
  - ✅ DAOS (HPC)
  - ☁️ S3 (Cloud)
- ☁️ RNTuple version 1 spec
- ☁️ RNTupleLite
- ☁️ Schema evolution
- ☁️ Disk-to-disk conversion
- ✎ Virtual data sets for skims and selections
- ✅ First exposure to frameworks:
  - ✅ CMSSW nanoAOD output module
  - ✅ Prototyping by ATLAS, CMS, LHCb I/O experts

- ✎ RDataFrame bulk processing
- ✎ Debugging and inspection tools
- ✎ Metadata API
- ✎ Special use case support: e.g. backfill, in-memory adapters
- ☁️ XRootD support
- ✎ Validation of feature coverage
- ✎ Training experiments' core developers
- ✎ Large-scale experiment benchmarks

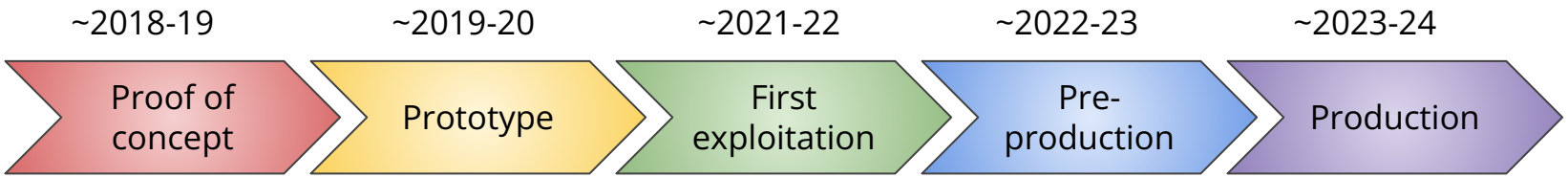
- ✎ PB scale tests
- ✎ Automatic optimization features
- ✎ Low-precision floats
- ✎ ML Training: direct GPU transfer
- ✎ End-user training
- ✎ Training and support for code and data migration

Expecting stable, if not increasing, I/O workload well into Run 4

✅ = available
☁️ = under development
✎ = programme of work
— = in collaboration with users/experiments

Growing importance of coordination & collaboration with experiment I/O experts

RNTuple Development Plan (Q1/2023)



- ✅ Architecture
- ✅ Review on state-of-the-art
- ✅ First prototypes

- ✅ Adoption in ROOT::Experimental
- ✅ I/O scheduler for local and remote access
- ✅ Performance validation

- 🌤️ Object store support
  - ✅ DAOS (HPC)
  - 🌤️ S3 (Cloud)
- ✔️ RNTuple version 1 spec
- 🌤️ RNTupleLite
- 🌤️ Schema evolution
- ✅ Disk-to-disk conversion
- ✎ Virtual data sets for skims and selections
- ✅ First exposure to frameworks:
  - ✅ CMSSW nanoAOD output module
  - ✅ Prototyping by ATLAS, CMS, LHCb I/O experts

- 🌤️ RDataFrame bulk processing
- ✎ Debugging and inspection tools
- ✎ Metadata API
- 🌤️ Special use case support: e.g. backfill, in-memory adapters
- ✅ XRootD support
- 🌤️ Validation of feature coverage
- ✎ Training experiments' core developers
- ✎ Large-scale experiment benchmarks

- ✎ PB scale tests
- ✎ Automatic optimization features
- ✎ Low-precision floats
- ✎ ML Training: direct GPU transfer
- ✎ End-user training
- ✎ Training and support for code and data migration

Expecting stable, if not increasing, I/O workload well into Run 4

✅ = available
✔️ = release candidate
🌤️ = under development
✎ = programme of work
— = in collaboration with users/experiments

Growing importance of coordination & collaboration with experiment I/O experts

RNTuple Class Design

Seamless transition from TTree to RNTuple

Event iteration
Reading and writing in event loops and through `RDataFrame`
`RNTupleDataSource`, `RNTupleView`, `RNTupleReader/Writer`

Logical layer / C++ objects
Mapping of C++ types onto columns
e.g. `std::vector<float>` ↦ index column and a value column
`RField`, `RNTupleModel`, `REntry`

Primitives layer / simple types
"Columns" containing elements of fundamental types (`float`, `int`, ...)
grouped into (compressed) pages and clusters
`RColumn`, `RColumnElement`, `RPage`

Storage layer / byte ranges
`RPageStorage`, `RCluster`, `RNTupleDescriptor`

Modular storage layer that supports files as data containers but also file-less systems (object stores)

Approximate translation between TTree and RNTuple classes:

TTree	≈	RNTupleReader RNTupleWriter
TTreeReader	≈	RNTupleView
TBranch	≈	RField
TBasket	≈	RPage
TTreeCache	≈	RClusterPool

→ RNTuple v1 RC1 Format Specification