Software framework for analysis at the LHC

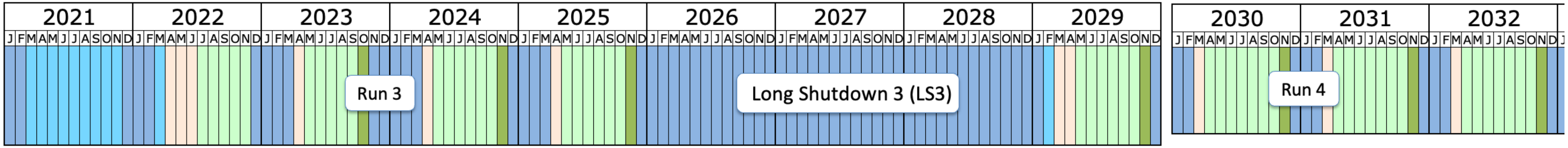
Allison Reinsvold Hall
United States Naval Academy

On behalf of the CMS, ATLAS, ALICE, and LHCb Collaborations

LHCP2024

Computing needs

- ALICE and LHCb: major upgrades between Run 2 and Run 3
  - Expect to process **100x** as much data in Run 3 compared to Run 1 and Run 2
  - ALICE: 1 month of Pb-Pb collisions will lead to ~5 PB of analysis-level data to analyze
- CMS and ATLAS: major upgrades before Run 4 (start of the HL-LHC)
  - ATLAS: data processing needs will increase by ~**10x** due to higher instantaneous luminosity and higher pileup (3x event size)



Goal: Minimize "time-to-insight"

- Broad community acknowledgement that scalable analysis software is key to maximizing physics output of CERN experiments
  - Analysis software was one of the 7 topical groups of the Computing Frontier of the 2021 Snowmass process (report)
  - CMS established the Common Analysis Tools (CAT) group in late 2022
    - Develop and maintain common software; provide a forum for discussion

- **Disclaimer:** can't cover everything in this talk

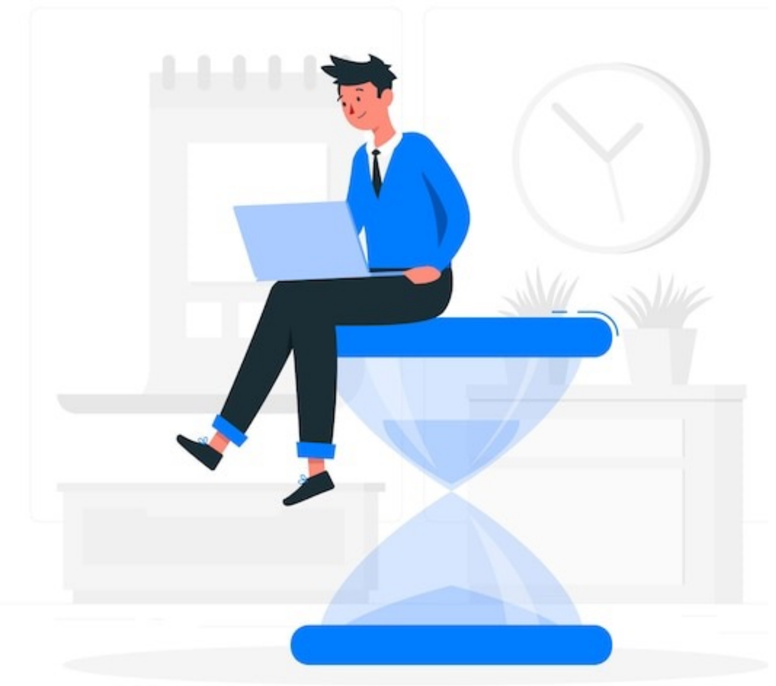
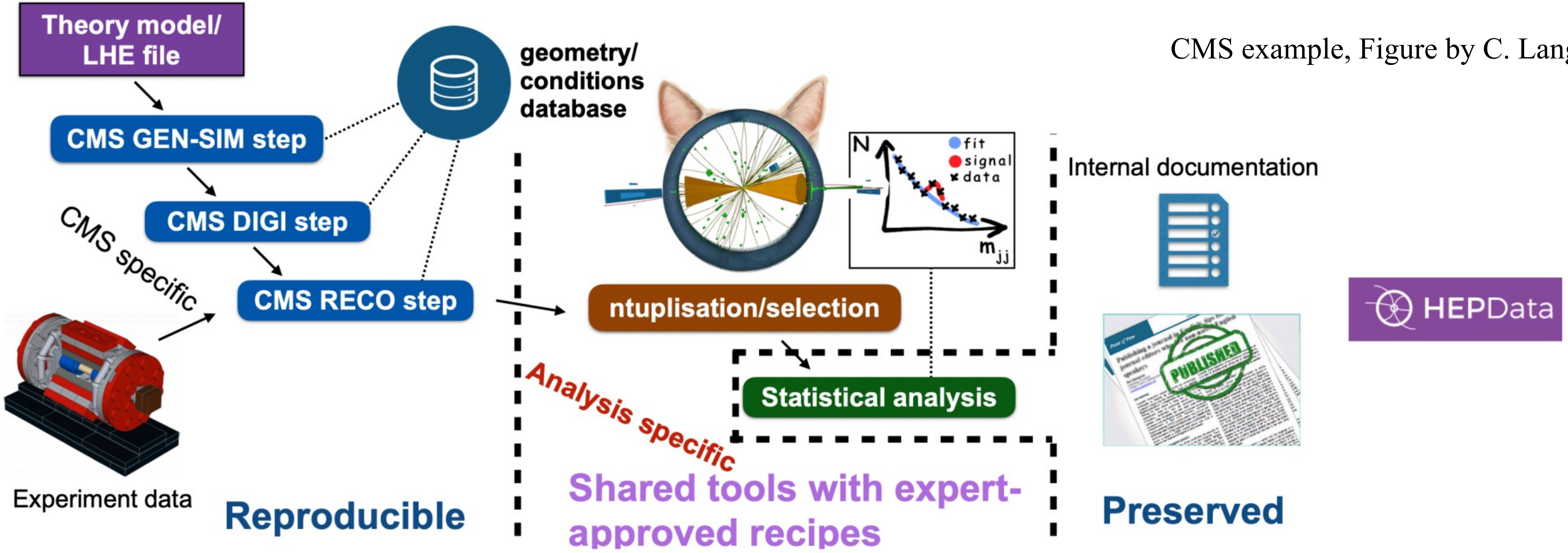


Image by storyset on Freepik

Analysis software: challenging middle ground

- Can turn into the Wild West, with O(100) different analyses leveraging many different analysis 'frameworks' and code
  - Not centrally managed by the experiment
  - Not (in most cases) archived or publicly released

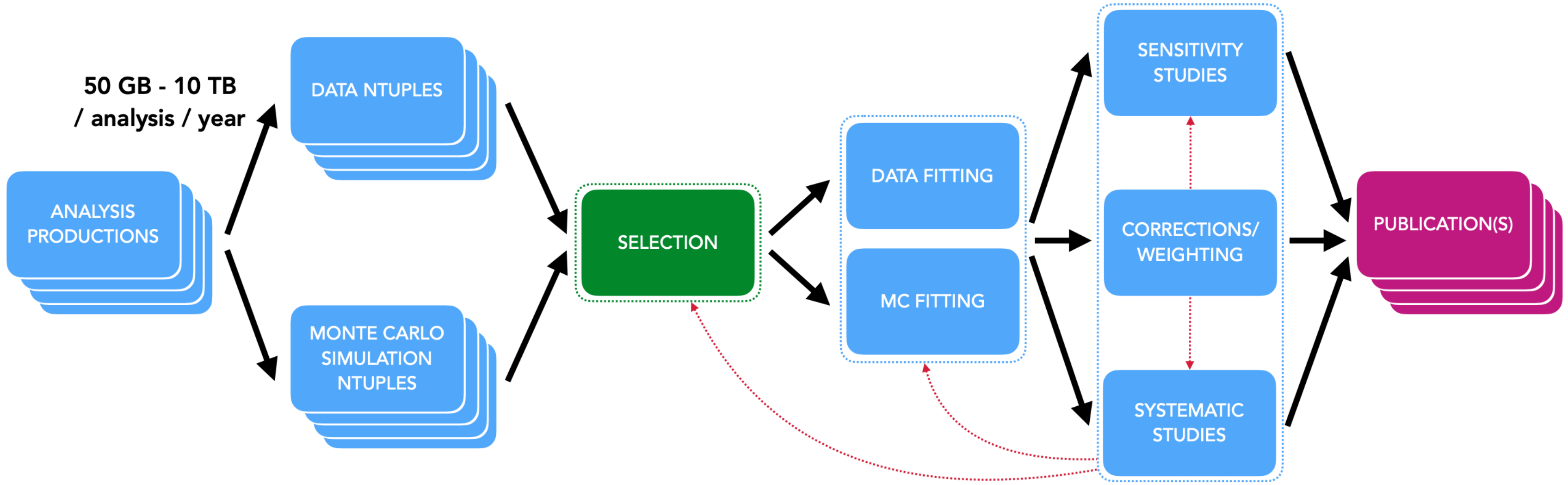
CMS example, Figure by C. Lange



Overview

LHC experiments have different approaches (largely driven by needs of the physics programs), but some common themes, solutions, and challenges:

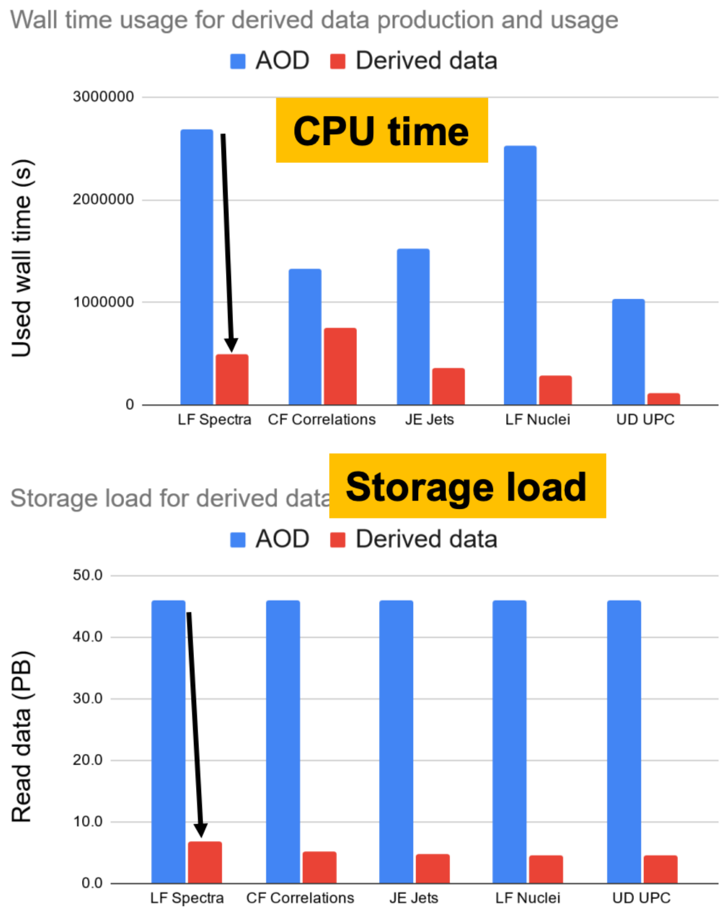
- Use of derived/reduced formats
- Taking advantage of vectorization/columnar processing
- Managing bookkeeping of analysis metadata
- Improving statistical analysis tools
- Writing analysis code that is reproducible and preservable (see Si Hyun's talk yesterday!)



LHCb user analysis dataflow

Starting point: derived formats

- CMS: introduced NanoAOD in 2018
  - 100x smaller than AOD format
  - Flat ROOT trees with variables for high-level physical objects only
- ATLAS: two new unskimmed derived formats
  - DAOD-Phys to be used for most Run 3 analyses (50 kb/evt)
    - Contains all physics objects, allowing for flexibility in object definitions
    - Long-lived particle searches typically have their own formats which are heavily skimmed
  - DAOD-PhysLite, to be used for most HL-LHC analyses (10 kb/evt)
    - Contains calibrated physics objects, after applying the common CP Tools
    - Centrally produced with frequent updates (every few weeks or months)
- ALICE: specific data formats for different analysis groups
  - Large reduction in CPU requirements and storage load

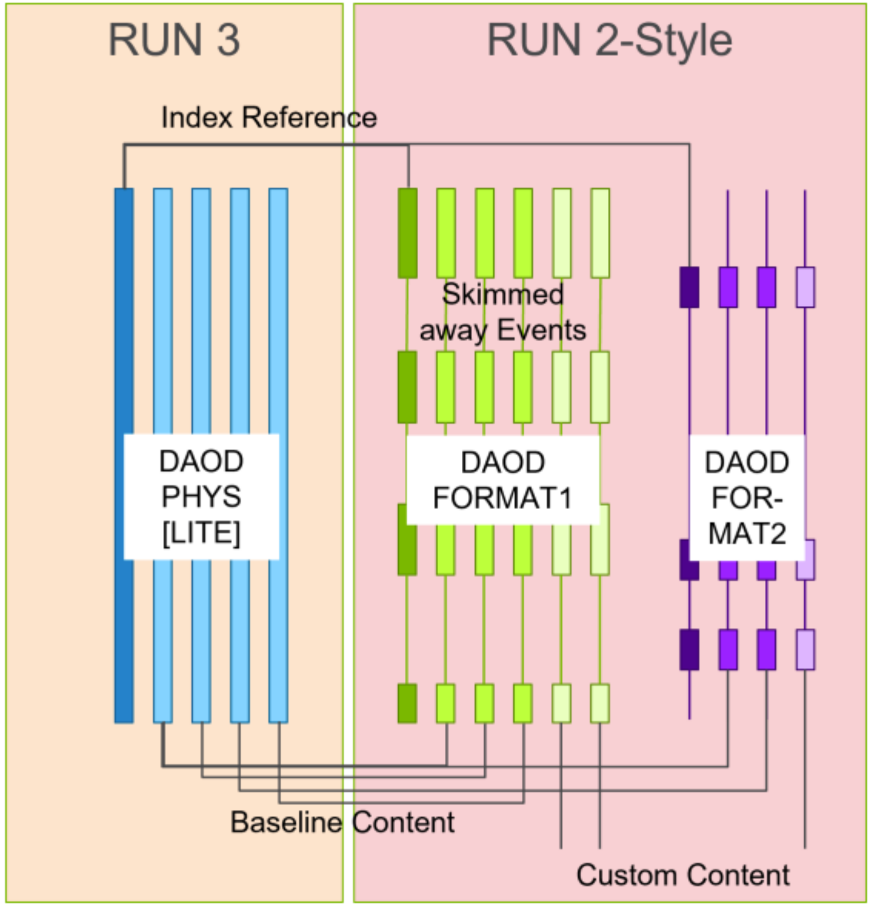


Plots from ALICE (source)

Enabling non-standard analyses – ATLAS

- Downside to derived formats: loss of information
  - Makes analyses like long-lived particle searches especially challenging
  - Adding additional variables for every event to a derived format is a big cost

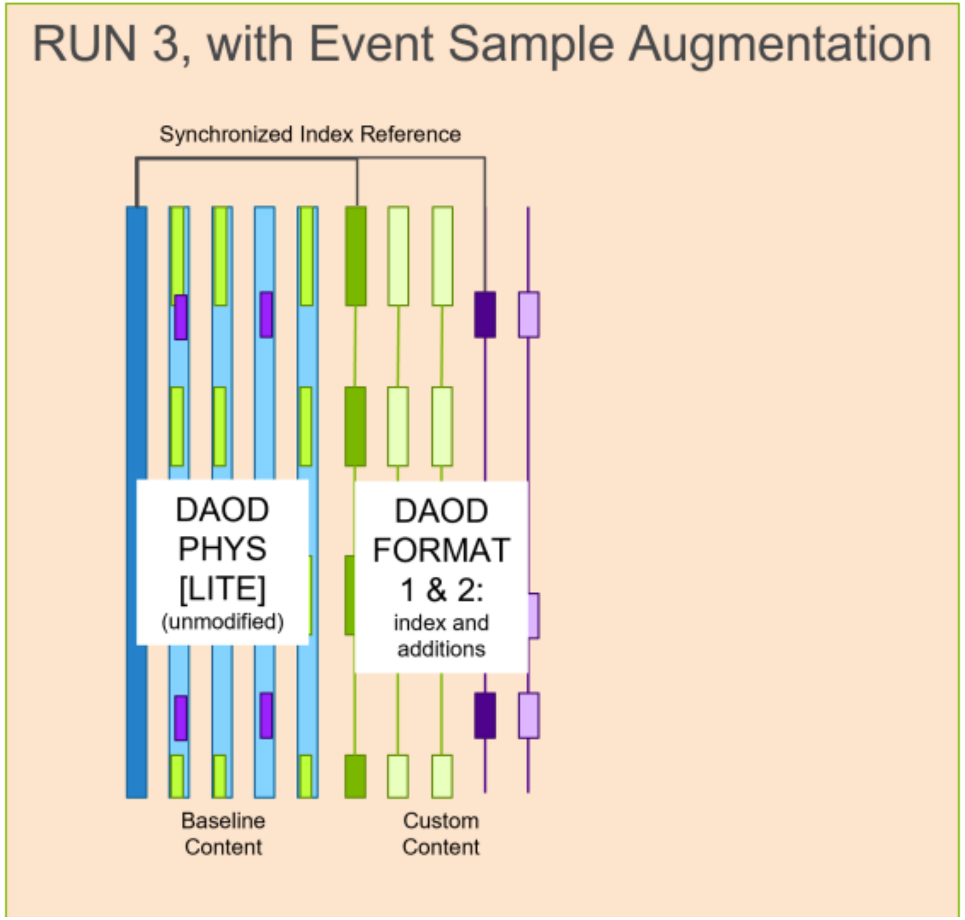
source



Enabling non-standard analyses – ATLAS

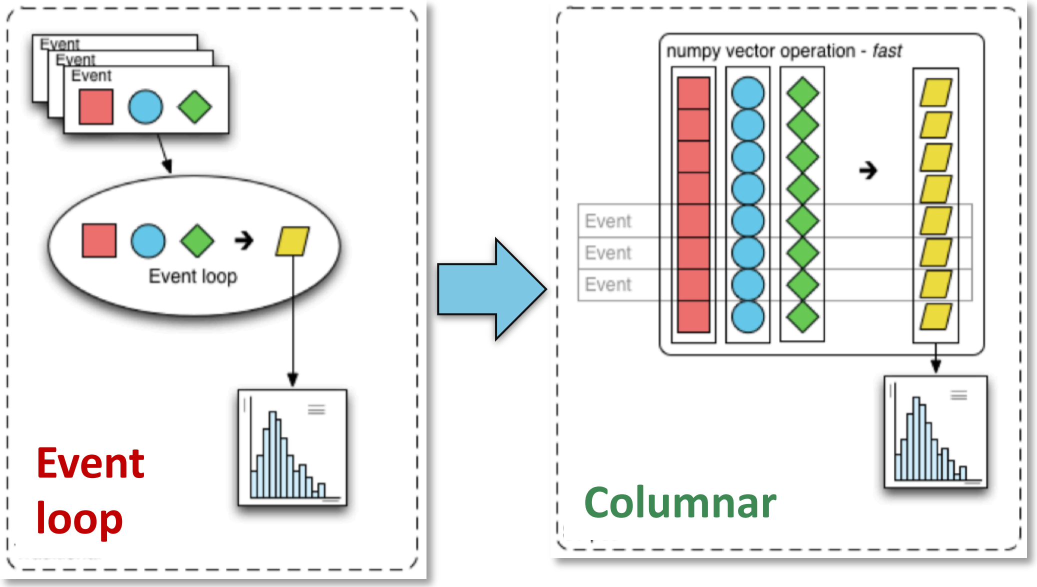
- Downside to derived formats: loss of information
  - Makes analyses like long-lived particle searches especially challenging
  - Adding additional variables for every event to a derived format is a big cost
- ATLAS solution: "event augmentation" to add non-standard variables to a new TTree in DAOD-Phys
  - Additional Ttree contains only the events of interest for that stream
  - New index reference branch is used to link multiple trees with different entry numbers
  - Work in progress: allow baseline DAOD-Phys and event augmentations to be in separate files
- CMS is working on an "LLPNanoAOD" format

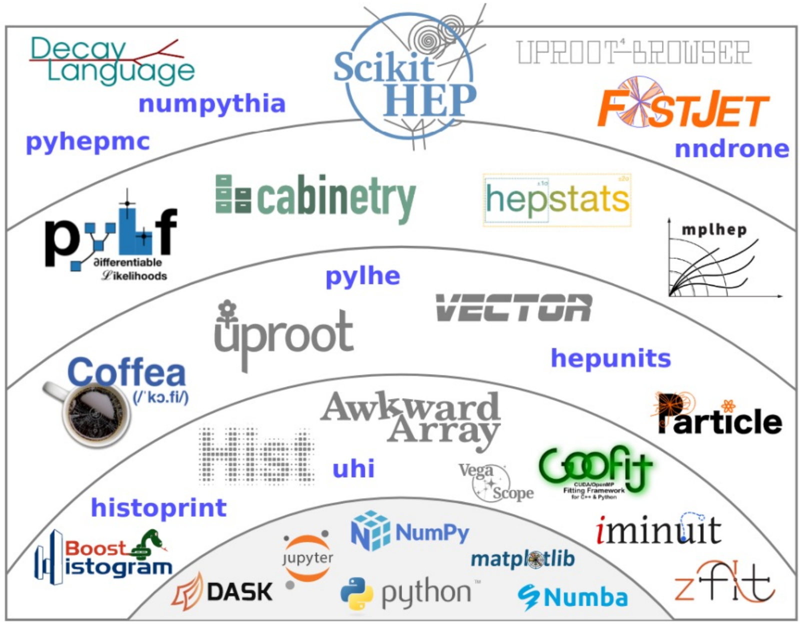
source



Columnar processing – end-user analysis

- Increasingly, experiments using columnar analysis to minimize I/O cost and improve vectorization/parallelism
- Two columnar approaches for user analysis:
  - Python/Scikit-HEP ecosystem using `uproot`, `awkwardarray`, `Coffea`
  - ROOT RDataFrame
- Important that both frameworks are supported and compatible data formats are used



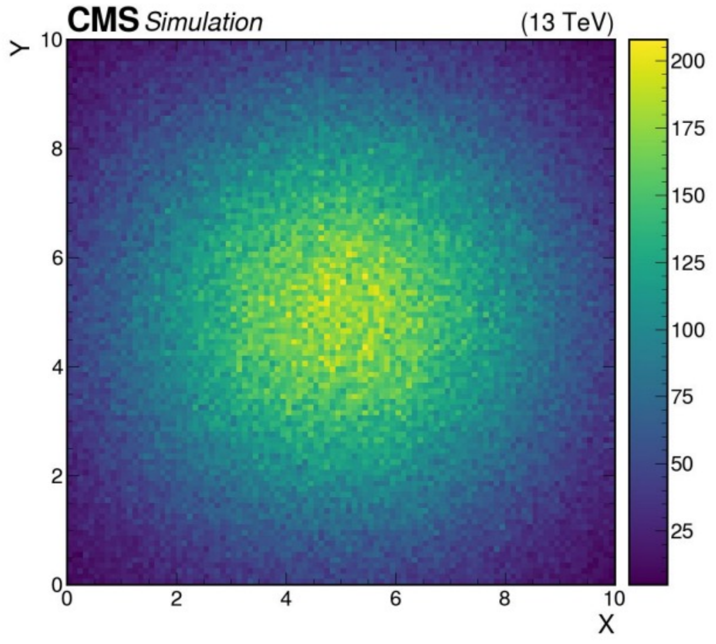


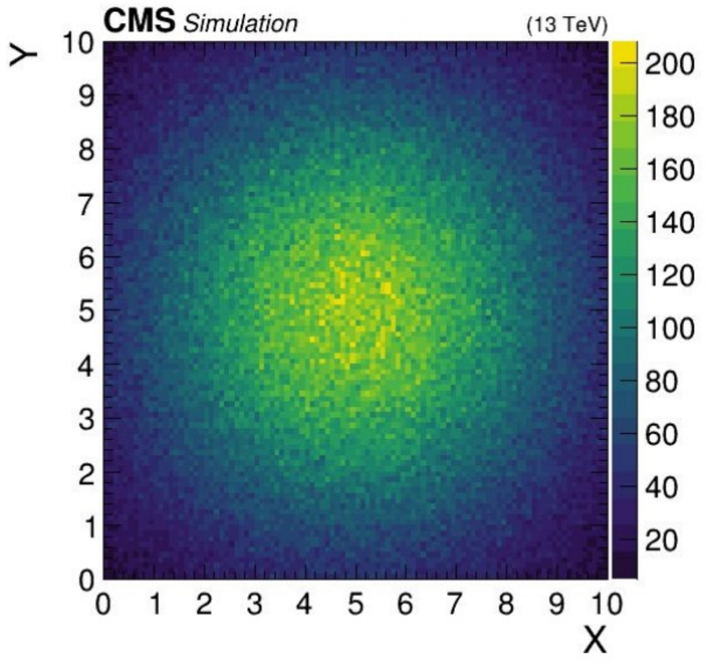
Scikit-HEP: Python ecosystem for HEP analyses

From Analysis Grand Challenge overview, CHEP 2023

Supporting multiple approaches

- CMS contributed to `mplhep` library to produce CMS-style plots using scikit-hep tools (left) and the `cmsstyle` library to make plots using pyROOT (right)





source

- All experiments contribute to different tools in the analysis ecosystem

Columnar processing – ALICE framework

O^2 Framework uses columnar format / flat tables to produce AOD format

- Stored in ROOT files on disk, Apache Arrow format in memory
- Zero-copy operations; underlying data doesn't need to be removed or copied in order to filter, group, or partition data
- No nesting; tracks and collisions are connected through indices in shared memory
- Complexity shielded from the user

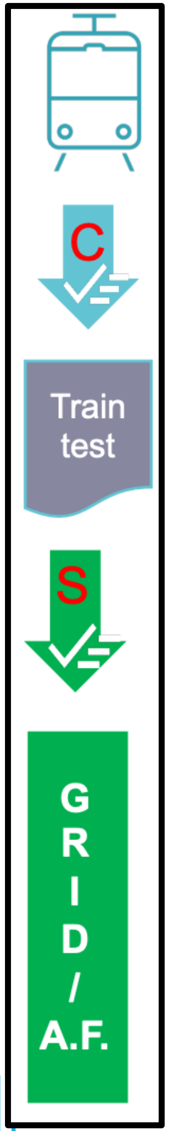
Columnar processing – ALICE framework

O^2 Framework uses columnar format / flat tables to produce AOD format

- Stored in ROOT files on disk, Apache Arrow format in memory
- Zero-copy operations; underlying data doesn't need to be removed or copied in order to filter, group, or partition data
- No nesting; tracks and collisions are connected through indices in shared memory
- Complexity shielded from the user

Hyperloop enables analysis workflows to be run on the Grid and Analysis Facilities

- Fully integrated with O^2
- Individual tasks ("wagons") are defined in JIRA and combined into trains
- "Operators" provide 24/5 support
- Automatic tests before submission and staged submission for large data samples (approval required to run on bigger datasets)
- Full bookkeeping, changelog and several comparison tools
- Mix of imperative and declarative code allowed



Vectorized processing – FunTuple in LHCb

- FunTuple: new tool for Run 3 in LHCb to produce ROOT N-tuples from raw data
  - Built on the `Gaudi` functional framework and the trigger (selection) software
- Utilizes C++ templates and a Structure of Array (SoA) format to take advantage of **vectorization**
- New feature allows users to customize which observables are stored
  - User-friendly python interface, rigorous unit-tests

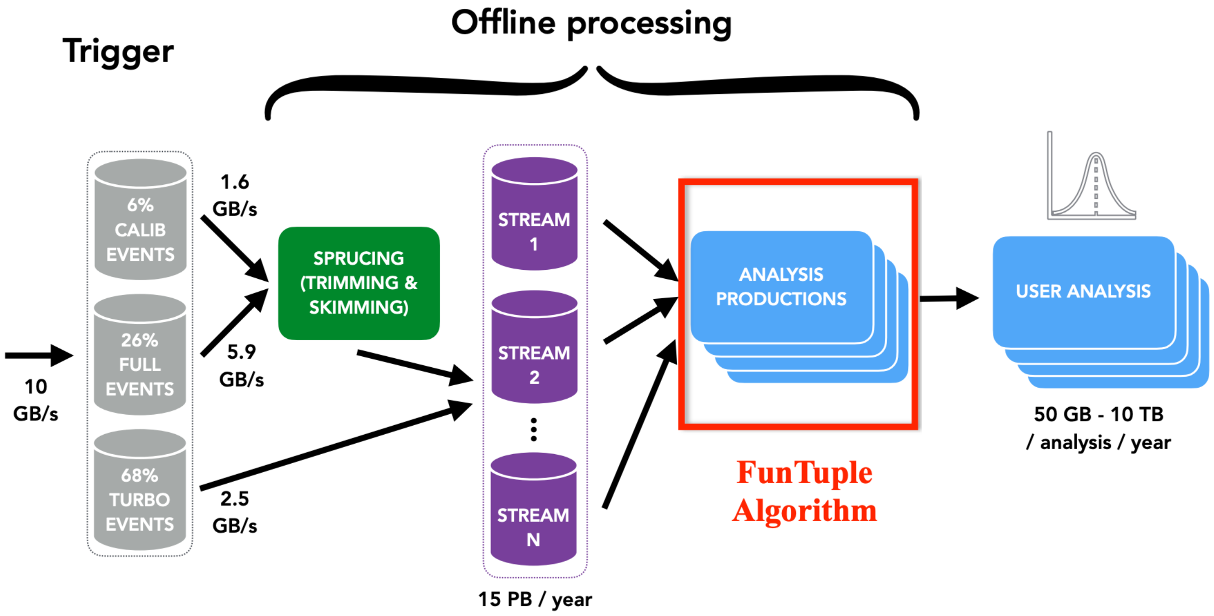
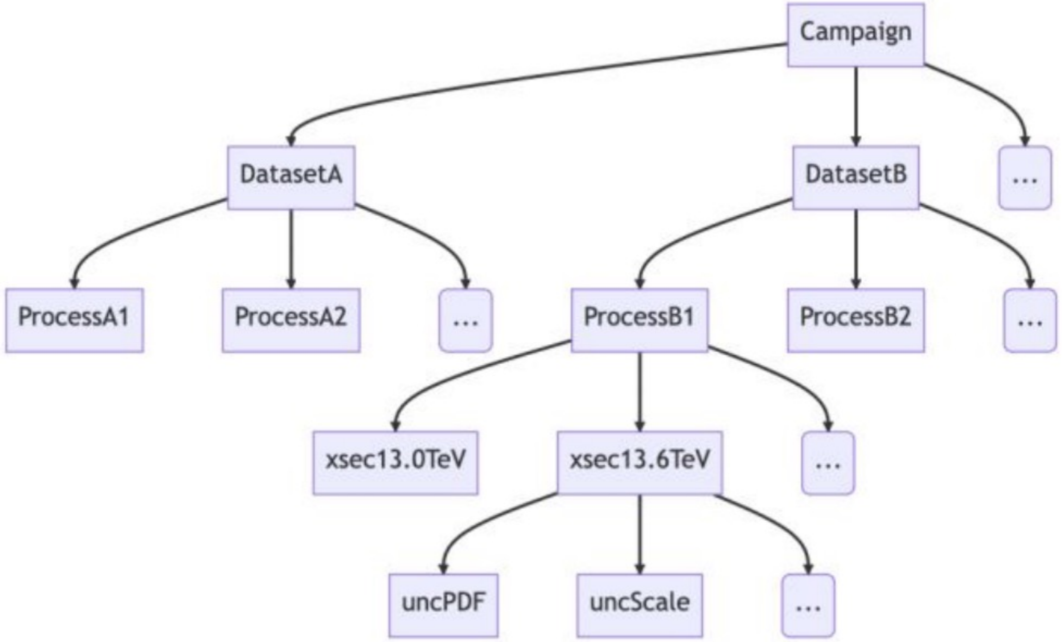


Figure taken from Comput Softw Big Sci 8, 6 (2024)

Analysis Metadata

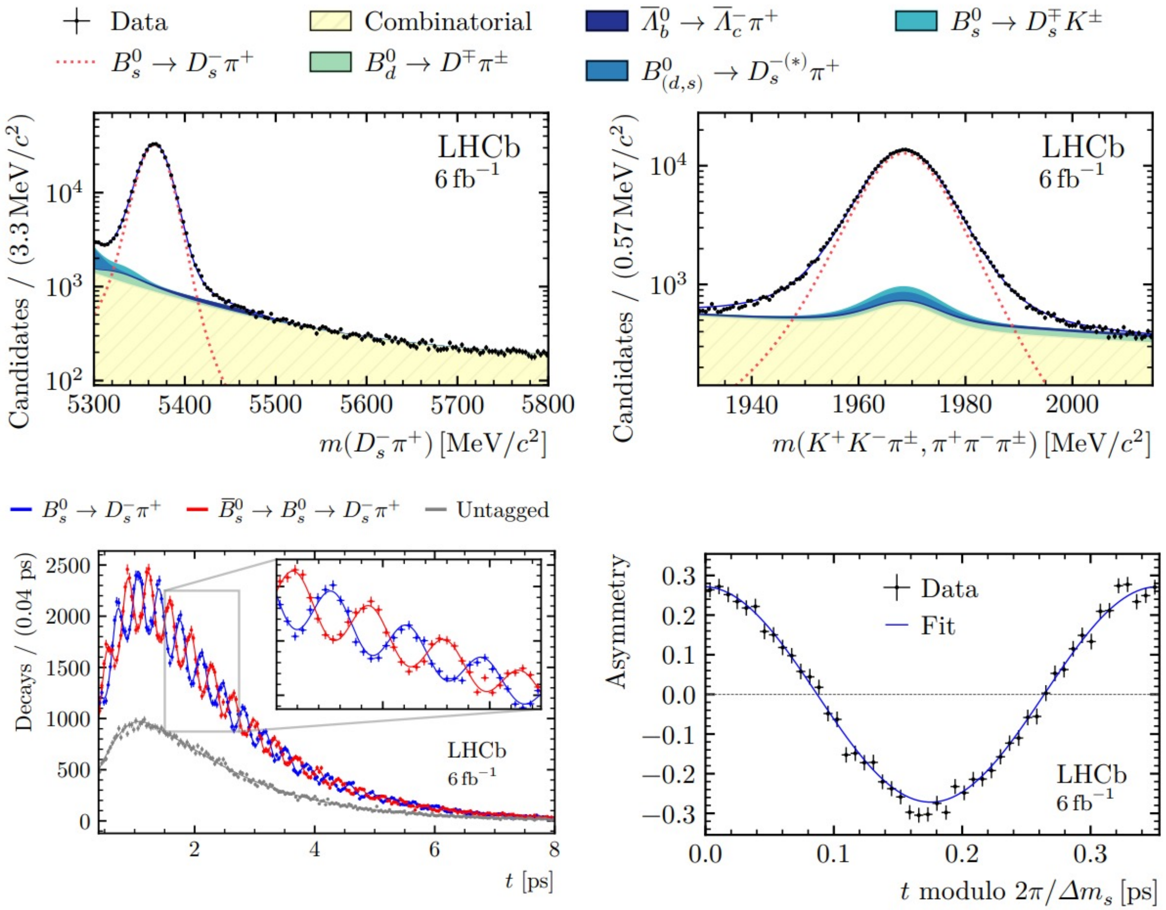
- Includes calibration data, dataset provenance, cross sections, data quality flags, etc.
- HSF Data Analysis Working Group published a paper in 2022 with recommendations for HEP metadata systems
  - Led to an ongoing HSF effort to build a cross-experiment conditions database (API with reference implementation)
- ATLAS uses analysis software release tags within Athena which includes paths to calibration data on `cvmfs`
- CMS: new effort for unified distribution of CMS metadata via `cvmfs`
  - In parallel: designing metadata schema and tools for easy access (like the Order tool in python)



Proposed CMS metadata tree structure (work in progress, source)

Statistical analysis

- **ATLAS:** Primarily uses HistFactory pdf template with RooStats/RooFit
  - Can be used with HistFitter (python wrapper for HistFactory)
  - First LHC experiment to release public likelihoods (see here for example)
- **CMS:** `Combine` package based on RooStats / RooFit (see recent paper)
  - Recently released the first public CMS stat. model!
    - Model = "datacard" (human-readable configuration file) and containerized public release of the code
- **LHCb:**
  - Primarily uses RooFit, but full amplitude analysis (O(100) parameters) requires other frameworks. Mix of CPU- and GPU-based tools are used
  - Functionality from ROOT 6.30 provides big improvements, including GPU backend



LHCb, Nature Phys 18 (2022) 1-5

Statistical analysis – cross-experimental efforts

- pyHF: Scikit-HEP project, supported by IRIS-HEP
  - First non-ROOT implementation of HistFactory pdf template
  - Pure python: uses deep learning frameworks as computational backends
    - Take advantage of auto differentiation and GPU acceleration
  - Large community adoption, especially in ATLAS, Belle-II
- HEP Statistics Serialization Standard (HS3)
  - Ongoing cross-experiment effort to generalize pyHF JSON model spec
  - Goal: define a code-independent standard (that could support eg RooFit, pyhf, BAT) for statistical procedures and results

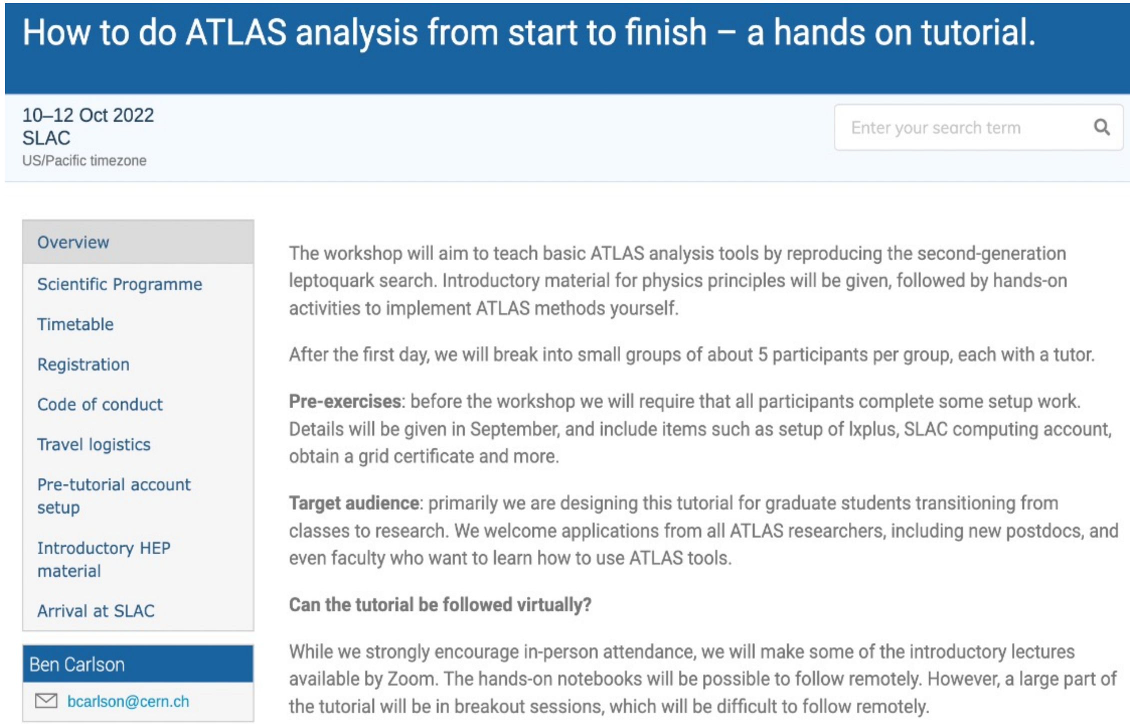
NumPy

PyTorch

TensorFlow

Analysis software training

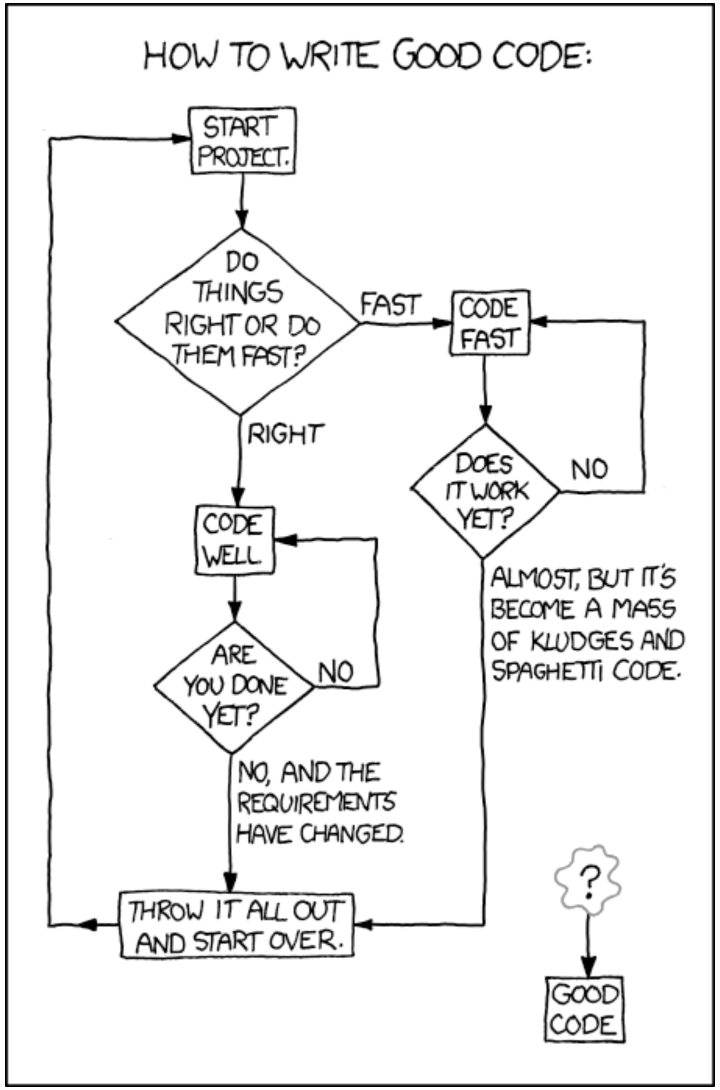
- **Training** is essential to bridge the gap between expert developers and novice users
- CMS provides week-long Data Analysis Schools (DAS) but also focused Hands-on-tutorial sessions (HATS) for specific topics
- ATLAS has a new tutorial format:
  - Interactive, hands-on and project-based structure with the aim of conducting end-to-end physics analysis
  - Working in teams (4-6 people)
- LHCb emphasizes peer-to-peer instruction, with the LHCb Starter Kit workshop introducing new analyzers to the experiment



- See HSF training center and HSF paper on analysis training initiatives
  - Key takeaway: Need to **motivate and reward** training efforts

Conclusions

- "Analysis software" includes many diverse tasks:
  - Data processing/workflows
  - Analysis selections/histograms/plotting
  - Statistical analysis
  - Handling metadata
  - Analysis preservation
- Quality of the **tools** directly impacts the quality and quantity of the **physics**
- Increasing, innovative efforts by all LHC experiments to unify and improve tools for analysis software
  - See references for lots more interesting information!



https://xkcd.com/844

References

- ATLAS:
  - Paper, “Software and computing for Run 3 of the ATLAS experiment at the LHC”, submitted to EPJC
  - Presentation on event sample augmentation, CHEP2023
- CMS:
  - Presentation about the CMS CAT group, ACAT 2024
  - Paper giving overview of Combine tool, submitted to CSBS
- ALICE:
  - Presentation on distributed analysis in ALICE, ICHEP 2022
- LHCb:
  - Presentation on statistical analysis in LHCb, RooFit workshop 2024
  - Presentation on offline data processing at LHCb, ICHEP 2022
  - Presentation on Analysis Productions in LHCb, CHEP2023
  - Presentation on Snakemake Workflows in LHCb, 2024
  - Paper on FunTuple, published in CSBS, 2024

General references

- "Software for analysis" presentation at LHCP2021:
- "Software for analysis" presentation at LHCP2022:
- HSF Data Analysis Working Group (website)
  - Training and onboarding initiatives paper
  - Constraints on analysis metadata systems paper
- HSF training center website
- Report from the Snowmass Computational Frontier, End User Analysis Topical Group (2022)
- pyHF presentation, ICHEP 2022
- LHC Reinterpretation Forum paper, published in SciPost (2020)

Backup

Analysis preservation

- To enable combinations and reinterpretations, need to be able to rerun analysis on new samples…

1.Capture software

Individual analysis stages in an executable way (including all dependencies)

2. Capture commands

How to run the captured software?

3. Capture workflow

How to connect the individual analysis steps?

Figure credit C. Lange

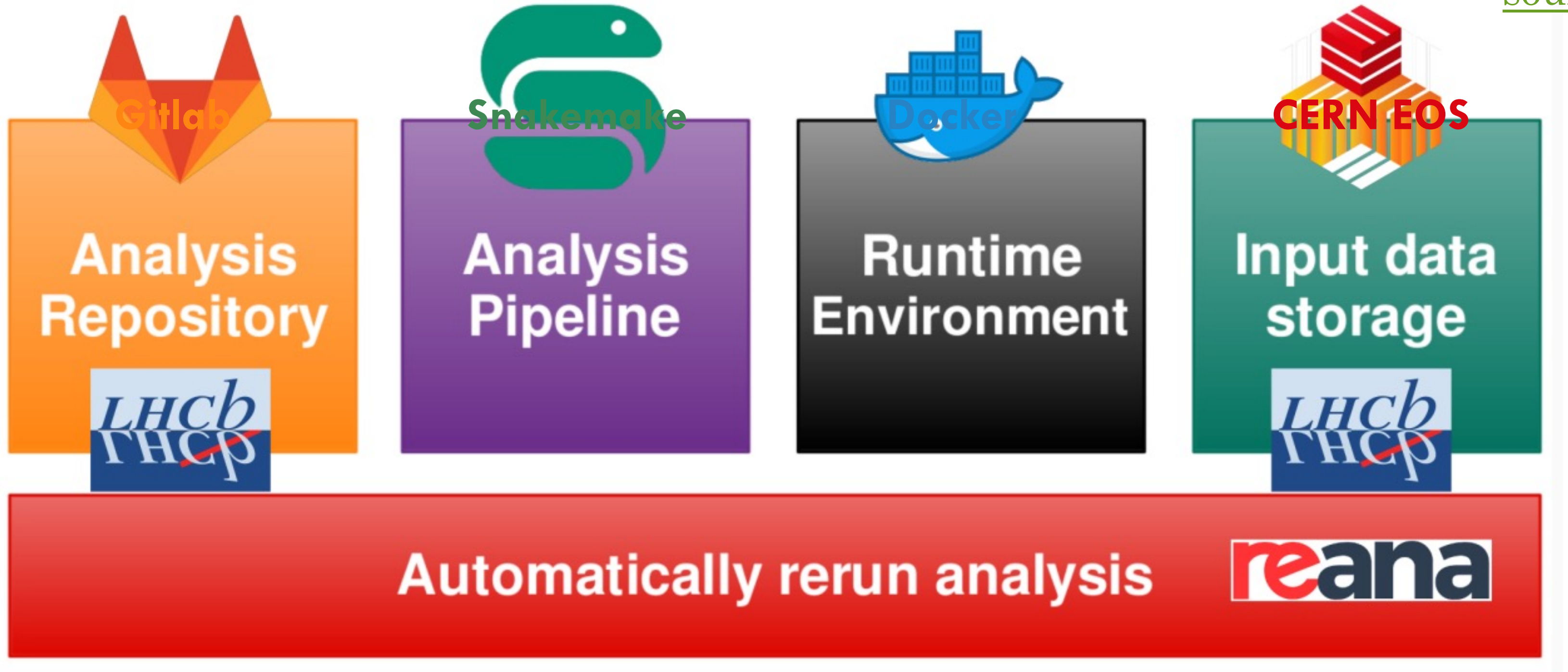
- HEPData is an open-source, publicly available repository for HEP results, used by many experiments
- Variety of tools for analysis preservation: Rivet, Reana, Recast, SnakeMake, LAW/Luigi, and others
- See Si Hyun's talk yesterday on this topic!

Analysis preservation

- To enable combinations and reinterpretations, need to be able to rerun analysis on new samples…

LHCb strategy:

source



- Check out recommendations of LHC Reinterpretation Forum