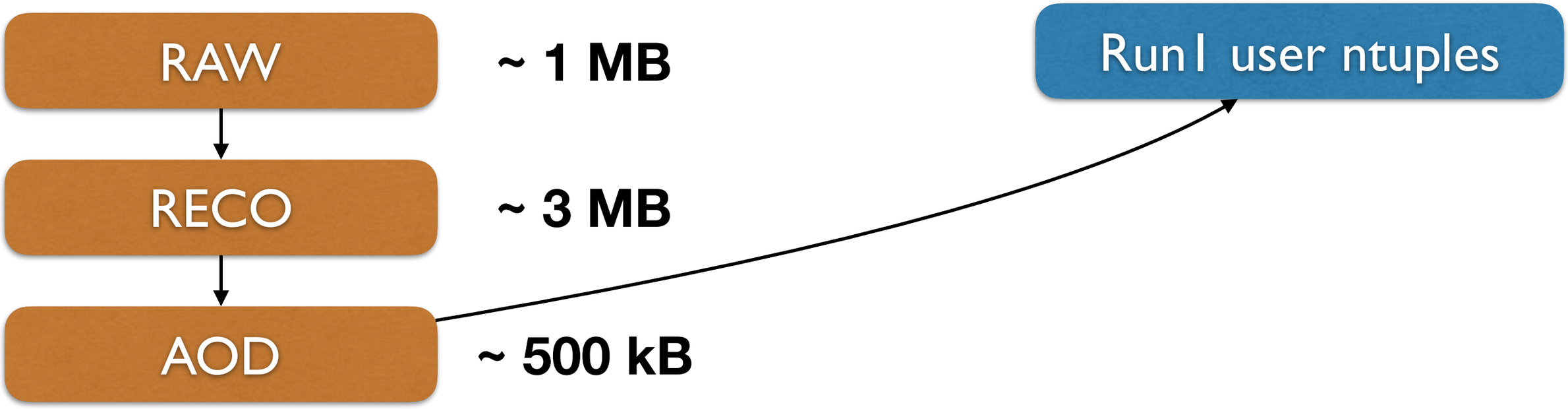
The NanoAOD event data format in CMS

Marco Peruzzi (CERN)
on behalf of the CMS Collaboration

19th International Workshop on Advanced Computing
and Analysis Techniques in Physics Research
Saas-Fee, Switzerland, 11th - 15th March 2019

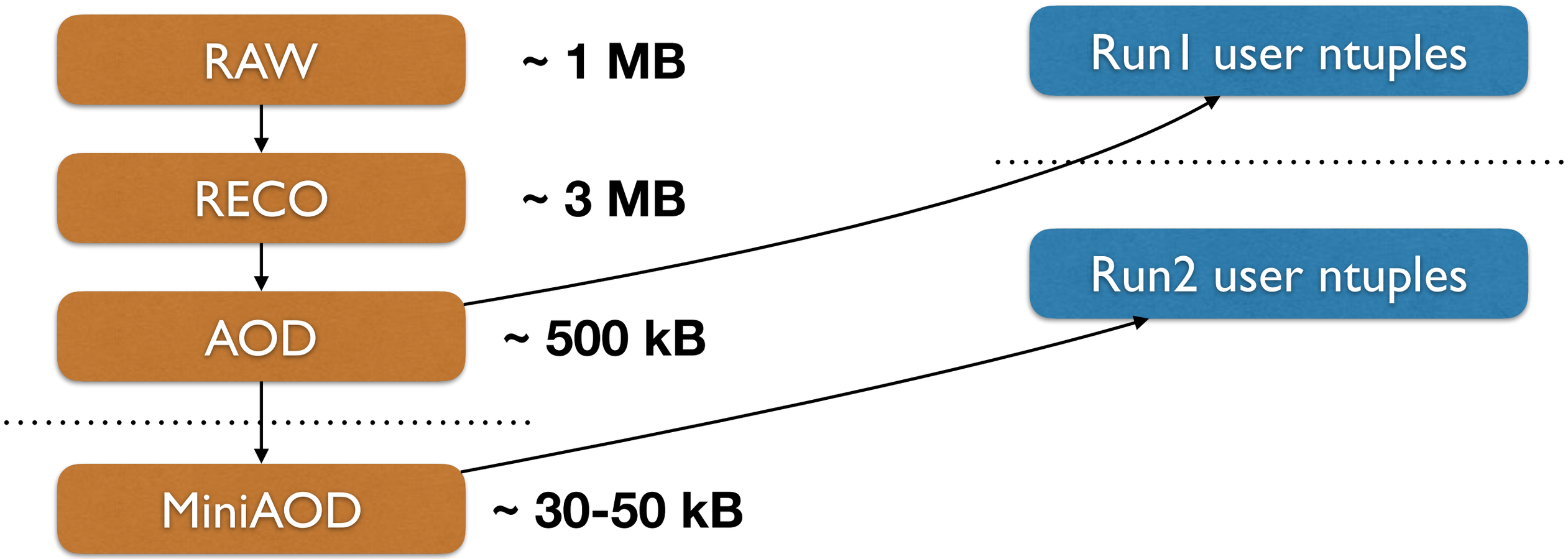
CMS event data formats

- Main event data formats is use for standard pp collision runs:



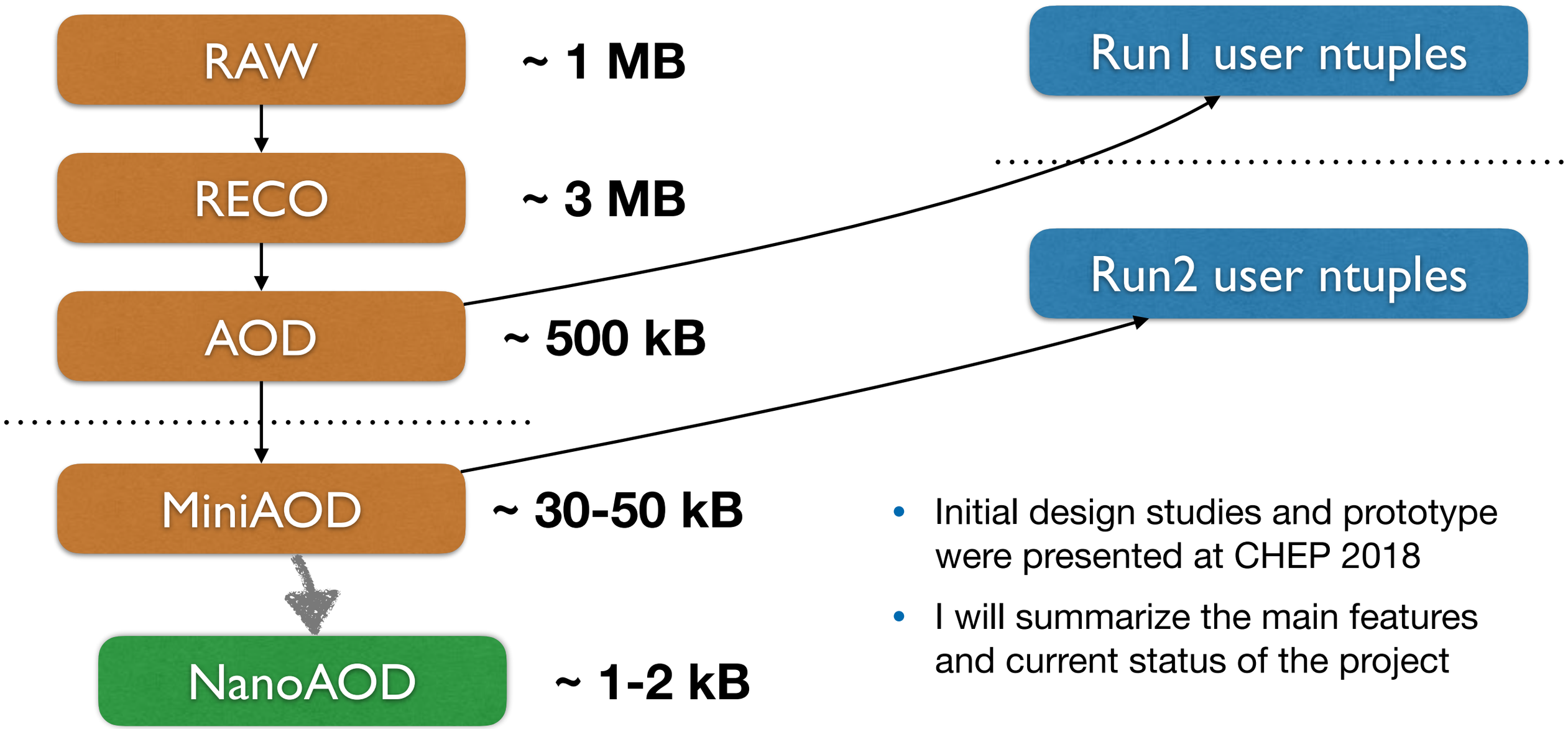
CMS event data formats

- Main event data formats is use for standard pp collision runs:



CMS event data formats

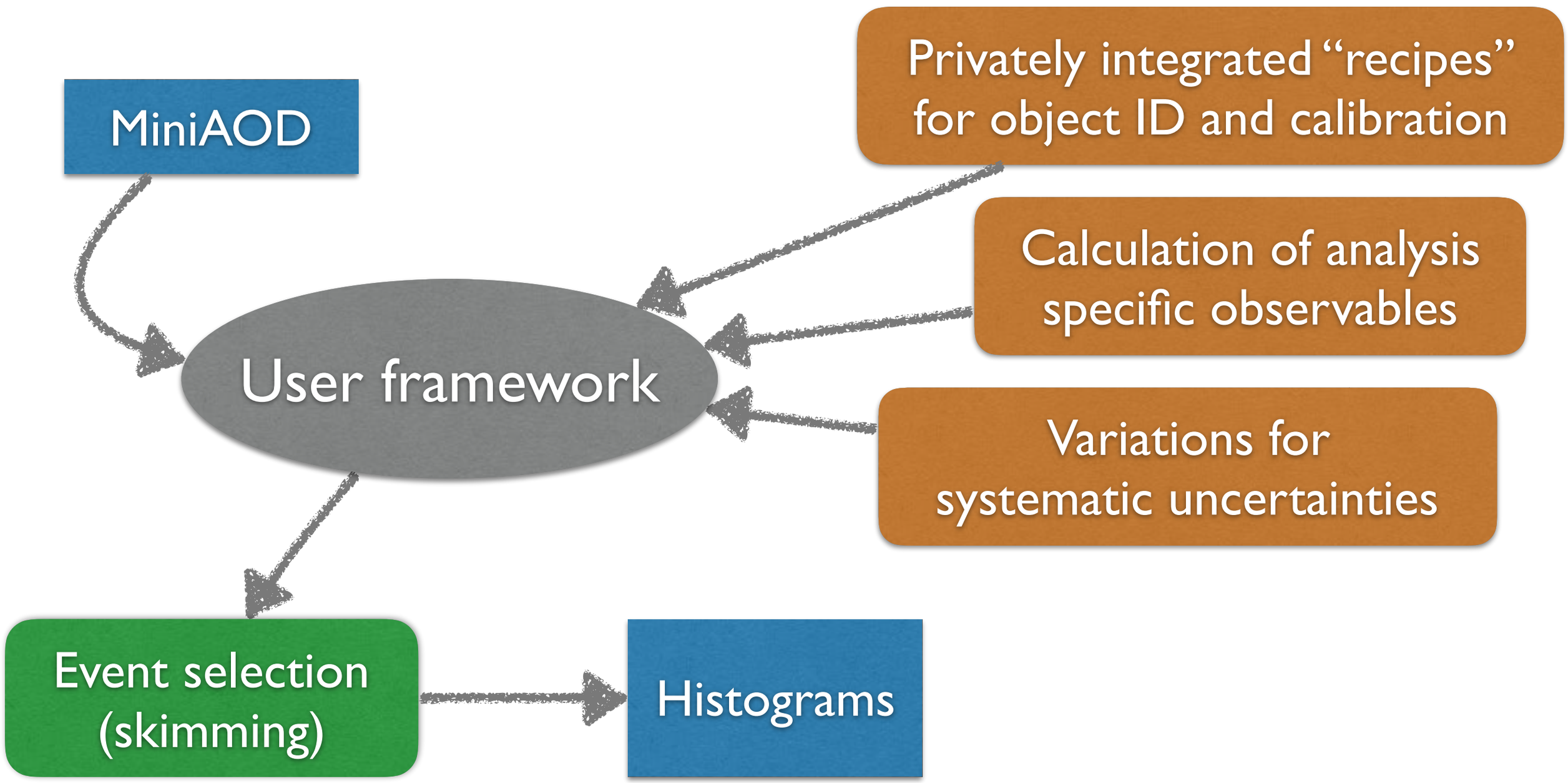
- Main event data formats is use for standard pp collision runs:



- Initial design studies and prototype were presented at CHEP 2018
- I will summarize the main features and current status of the project

Private analysis frameworks

- During Run2, most physics analyses have been based on the MiniAOD data format
- Typical pattern:



Towards NanoAOD

- Developing a framework to support these activities is a **significant workload investment**
  - We have seen "**large frameworks**" emerging, each supporting many different analyses and maintained collaboratively by several institutes
  - **Private ntuples are actually shared and re-used multiple times**, even if they were originally designed for a specific analysis
  - With different detector configurations and object definitions for each data taking year, the complexity of recipes increases dramatically

Towards NanoAOD

- Developing a framework to support these activities is a **significant workload investment**
  - We have seen "**large frameworks**" emerging, each supporting many different analyses and maintained collaboratively by several institutes
  - **Private ntuples are actually shared and re-used multiple times**, even if they were originally designed for a specific analysis
  - With different detector configurations and object definitions for each data taking year, the complexity of recipes increases dramatically

- We have defined a **new compact data format**, with an event size in the ballpark of **1 kB**
  - content choice based on our Run2 analysis experience and interaction with collaborators in several groups: **first of all it's a physics challenge, even before dealing with technical aspects!**
  - resembles the typical structure and size of private ntuples, with new features to make it more universal and interface it with central processing tools
  - **aim at supporting at least 50% of physics analyses** in CMS, as an initial goal

Technical design

- **Technical requirements**, inferred by what has proven popular among analyzers:
  - **flat ROOT::TTree** [1] **event structure**
    - can be read without a dictionary, interfaced to external tools, …
  - maximum **flexibility in the definition of the event content**
  - freedom to efficiently run **private productions** for tests and customised versions

[1] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also "ROOT" [software], Release v6.16/00, 05/02/2019, https://doi.org/10.5281/zenodo.2557526.

Technical design

- **Technical requirements**, inferred by what has proven popular among analyzers:
  - **flat ROOT::TTree** [1] **event structure**
    - can be read without a dictionary, interfaced to external tools, …
  - maximum **flexibility in the definition of the event content**
  - freedom to efficiently run **private productions** for tests and customised versions
  - configuration and code easily **accessible even by collaborators with less technical skills** (e.g. new students joining an ongoing project)

[1] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also "ROOT" [software], Release v6.16/00, 05/02/2019, https://doi.org/10.5281/zenodo.2557526.

Technical design

- **Technical requirements**, inferred by what has proven popular among analyzers:
  - **flat ROOT::TTree** [1] **event structure**
    - can be read without a dictionary, interfaced to external tools, …
  - maximum **flexibility in the definition of the event content**
  - freedom to efficiently run **private productions** for tests and customised versions

  - configuration and code easily **accessible even by collaborators with less technical skills** (e.g. new students joining an ongoing project)

  - modular structure for the **efficiently sharing** additional variables developed by a specific analysis group, and useful for many other
  - cover different object definitions and identification variables in a coherent structure, **abstracting from low level details** (e.g. “tight photon ID” across Run2)

[1] Rene Brun and Fons Rademakers, ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also "ROOT" [software], Release v6.16/00, 05/02/2019, https://doi.org/10.5281/zenodo.2557526.

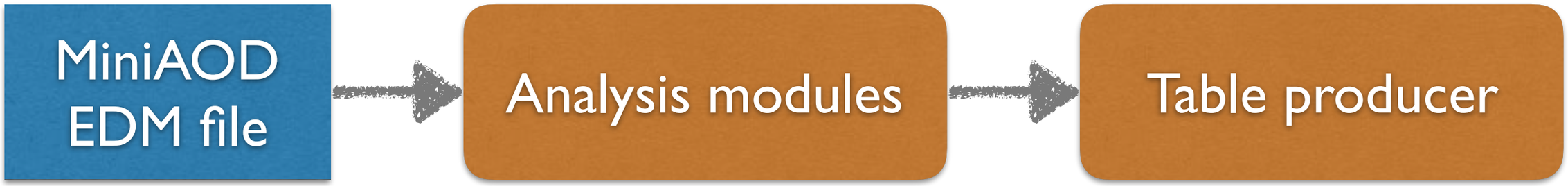
Design choices

- NanoAOD is a **bare ntuple, and all branches follow a naming convention**
  - e.g. MET_pt, MET_eta, Muon_pt [nMuons], Muon_eta [nMuons]
- Each variable is stored with **configurable precision, to save space**
  - useless to store a variable measured with 10% precision as a 32-bit float
  - aggressive ROOT file compression is then applied

- **No cross-cleaning** is performed among physics objects in production (e.g. we don't decide whether a candidate is a jet or an electron)
  - this is an analysis-dependent choice, depends on object selection
  - we store both, establishing a **link** between them for later usage

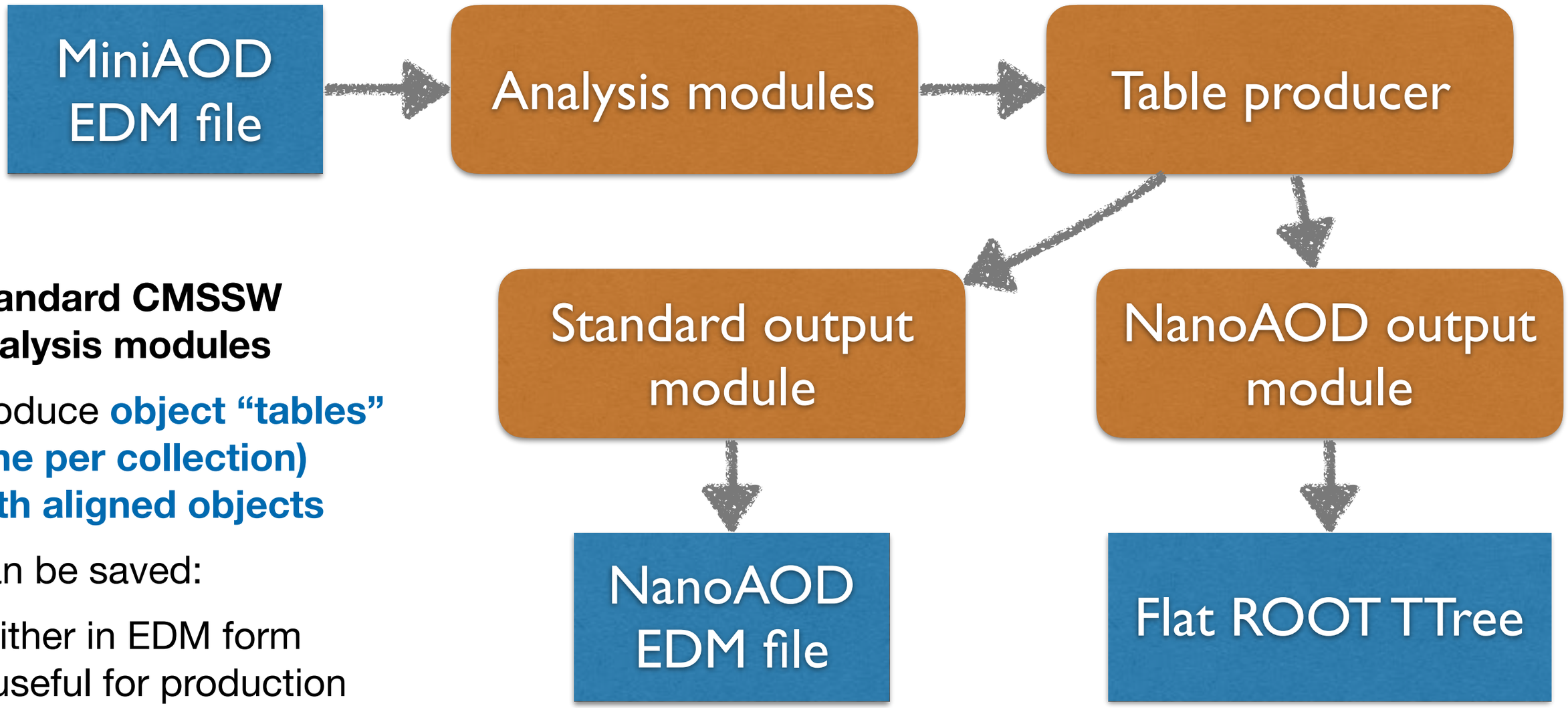
- Auxiliary TTrees in the same file carry **additional metadata**
- **Fully integrates with the CMSSW production infrastructure, for both job configuration and workflow management**

Structure of NanoAOD code



- **Standard CMSSW analysis modules**
- Produce **object "tables" (one per collection) with aligned objects**

Structure of NanoAOD code



- **Standard CMSSW analysis modules**
- Produce **object "tables" (one per collection) with aligned objects**
- Can be saved:
  - either in EDM form (useful for production e.g. input to merge jobs),
  - or to flat ROOT TTree for analysis usage

Configuration

- **Standard CMSSW analysis modules**
  - allows for seamless integration of existing code,
    either developed for reconstruction or by analysis groups
- Modules for **typical analysis operations standardised to a common interface**
  - e.g. evaluation of multivariate discriminant on each object in a collection

Configuration

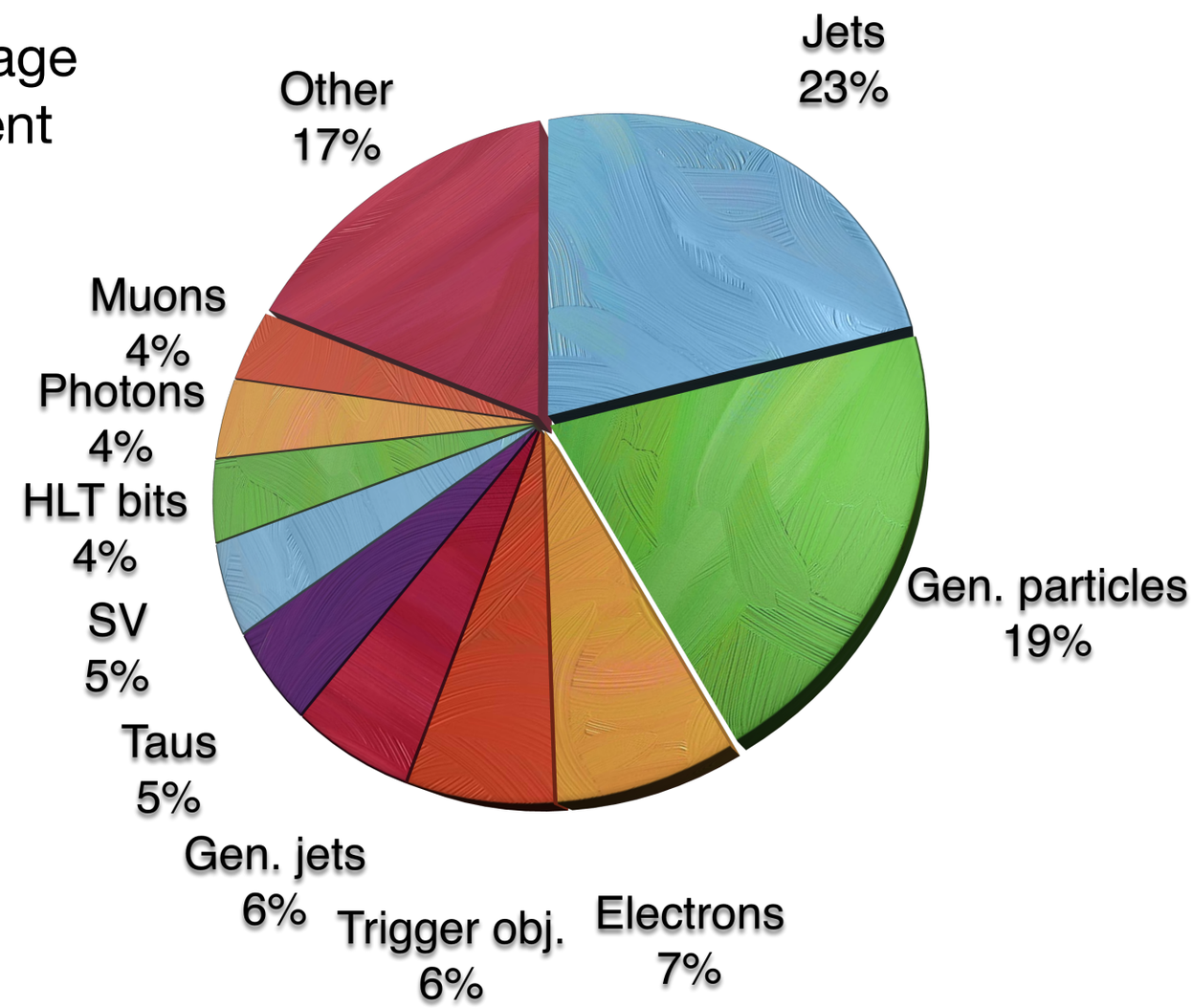
- **Standard CMSSW analysis modules**
  - allows for seamless integration of existing code, either developed for reconstruction or by analysis groups
- Modules for **typical analysis operations standardised to a common interface**
  - e.g. evaluation of multivariate discriminant on each object in a collection
- **Table producer** also configured via **expression strings**
  - **easy to customise!**
  - allow for calling object functions from compiled formulas
  - each variable comes with **integrated documentation**

Muon_jetIdx	Int_t(index to Jet)	index of the associated jet (-1 if none)
Muon_jetRelIso	Float_t	Relative isolation in matched jet (1/ptRatio-1, pfRelIso04_all if no matched jet)

  - align choice of analysis modules and table producer configuration for each data taking period, via global modifiers

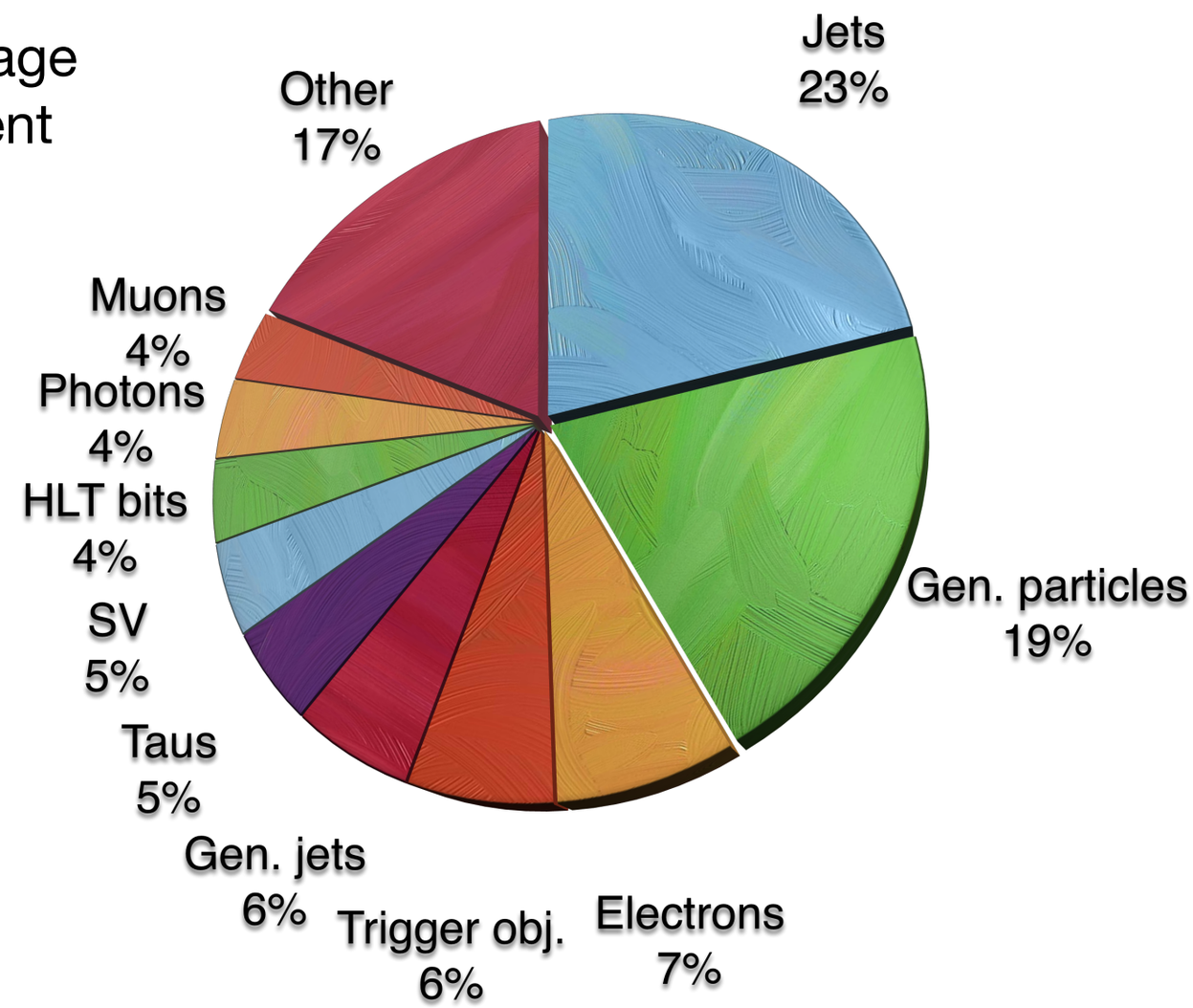
Event size and production speed

- **High-level physics objects** of common usage are **covered** by the current NanoAOD content
- Per-event compressed size is well **below 1 kB in typical data events** with Run2 pileup conditions
- About **1.6 kB/event in ttbar sample** (busy events with many jets), including generator level information



Event size and production speed

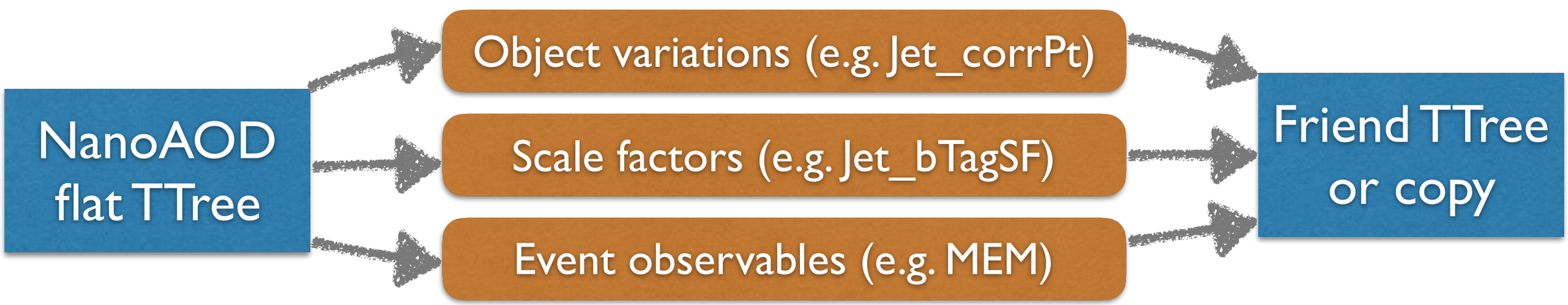
- **High-level physics objects** of common usage are **covered** by the current NanoAOD content
- Per-event compressed size is well **below 1 kB in typical data events** with Run2 pileup conditions
- About **1.6 kB/event in ttbar sample** (busy events with many jets), including generator level information



- **Production speed** from MiniAOD is about **10 Hz on one CPU core**
  - again, this figure is quite sample-dependent
  - we observe **no particular infrastructure bottlenecks:** most of the time goes in the evaluation of complex multi-variate jet taggers based on deep neural networks (could be moved to MiniAOD processing)

Post-processing

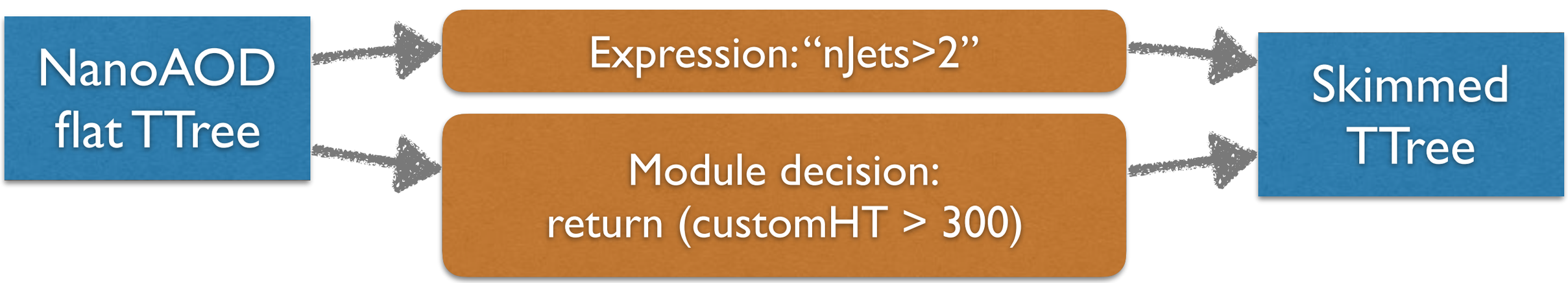
- A **framework for post-processing of NanoAOD trees** is also centrally maintained
- **Plugin modules** for:
  - calculating variations (e.g. from systematic uncertainties) of object quantities
  - obtaining data / MC corrections in form of event weights (“scale factors”)
  - even higher level event observables such as matrix element amplitudes, deep neural network classifiers, multi-object taggers, …



Post-processing

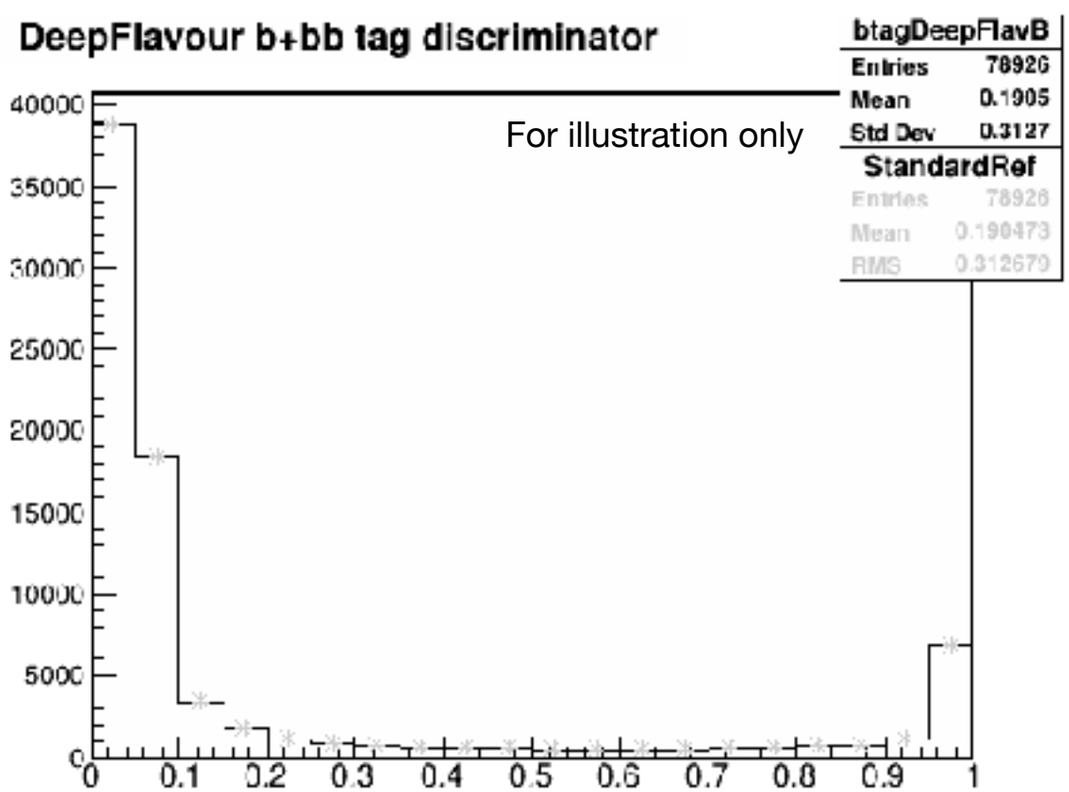
- A framework for post-processing of NanoAOD trees is also centrally maintained
- Plugin modules for:
  - calculating variations (e.g. from systematic uncertainties) of object quantities
  - obtaining data / MC corrections in form of event weights ("scale factors")
  - even higher level event observables such as matrix element amplitudes, deep neural network classifiers, multi-object taggers, …

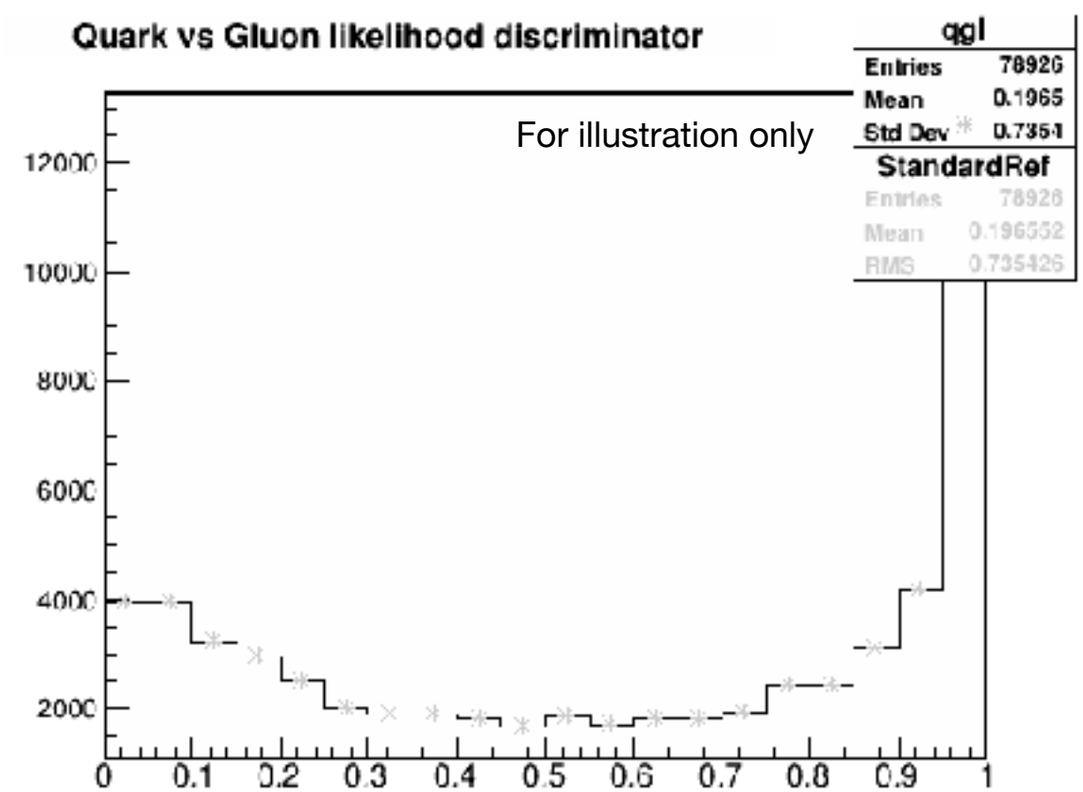
- **Event skimming** based on both string expressions (à - la - TTree::Draw) and modules
- Strong advantage in terms of workload and robustness to have these implemented in **common packages, reviewed and maintained by object experts**



Automated workflows

- In the modular NanoAOD framework, users can easily develop **automated analysis workflows** for a variety of repetitive tasks
- For **code development**:
  - **continuous integration workflow** for testing new NanoAOD features and monitor the effect of upstream changes in other packages
  - integrated with the CMSSW **Data Quality Monitoring infrastructure**





Automated workflows

- For derivation of **object calibrations**, each time the detector conditions change:
  - exploring how to automate the derivation of jet energy corrections
  - potential to support central, automated efficiency scale factors derivation

- **The reduction in workload for physics object groups could be substantial**
  - faster object calibrations mean less time between recording the data and publishing a physics analysis
  - also very valuable, allows for **promptly monitoring the data quality with the same high level features that will be used in the analysis**

Additional interfaces

- **Interface to new data processing tools:**
  - **natural environment to profit from the new ROOT RDataFrame features** (*)

    e.g. check how to find the Higgs boson in CERN OpenData in this RDataFrame tutorial using NanoAOD-like ntuples!

  - also **easily interfaced to non-HEP-specific ecosystems**

- NanoAOD promising also from the **data preservation and open access** perspective
  - accessible high level object features
  - no need of experiment-specific software
  - easy interface to a variety of modern data analysis tools (e.g. machine learning)

(*) You will see this in action in L. Moneta's talk, later in this session

Adoption status

- NanoAOD production is now **run by default in all MC and data processing campaigns**
  - including re-processing to be used for most results in the next months: full Run2 CMS results using NanoAOD are a very short time ahead!
  - readiness already shown by early adoption in our recent observation of Higgs boson decay to bottom quarks (*)

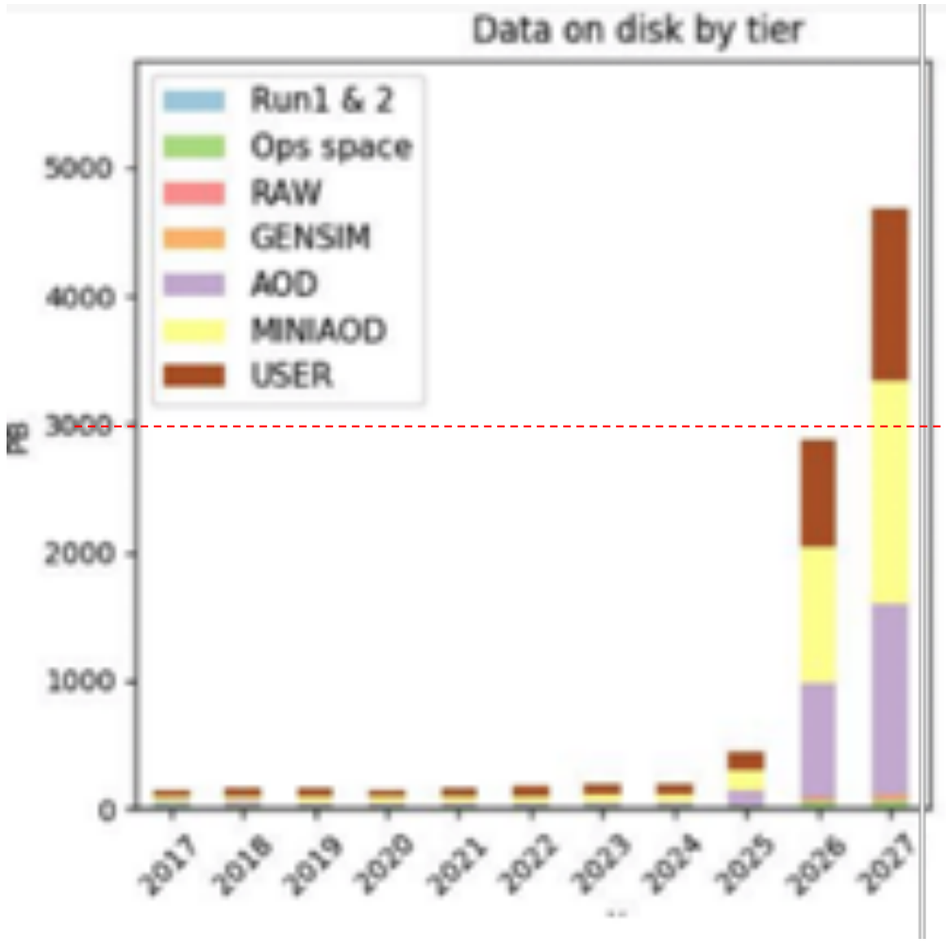
- The impact on the data analysis pattern is still to be observed in its entirety
- The complete set of **full Run2 data and MC, about 60B events, fits in few tens of TB**
  - *" after 2020, the whole Run2 could fit in one big machine! "*
  - even a moderate skimming brings the dataset size to a level where many subsequent analysis steps can be performed "locally"

- As we move **into Run3, we are optimistic that the adoption of the NanoAOD will grow**
  - we continue tuning its content to help this process

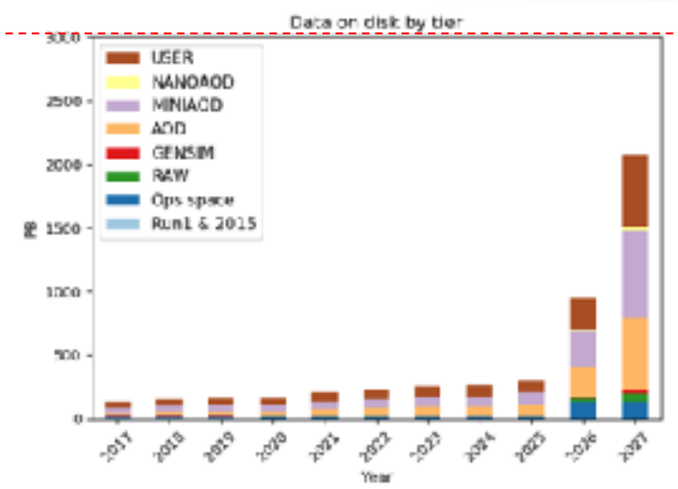
(*) CMS Collaboration, Observation of Higgs boson decay to bottom quarks, Phys. Rev. Lett. 121, 121801, doi:10.1103/PhysRevLett.121.121801

Towards HL-LHC

- We **assume** that, during HL-LHC, **NanoAOD will support 50% of the analyses**
  - we aim to exceed this goal by more content tuning and experience
  - the 70% goal initially set for MiniAOD coverage has now been exceeded to about 95%
- This makes it possible to reduce the number of MiniAOD replicas on disk



- **Current estimates** (*) show that this represents:
  - a **factor 2x reduction in disk** storage needs!
  - **15% less CPU** for analysis



Extracted from T. Boccali, M. Klute, LHCC WLCG CMS report, 11 Sep. 2018

(*) More details in T. Boccali's talk on Wednesday afternoon