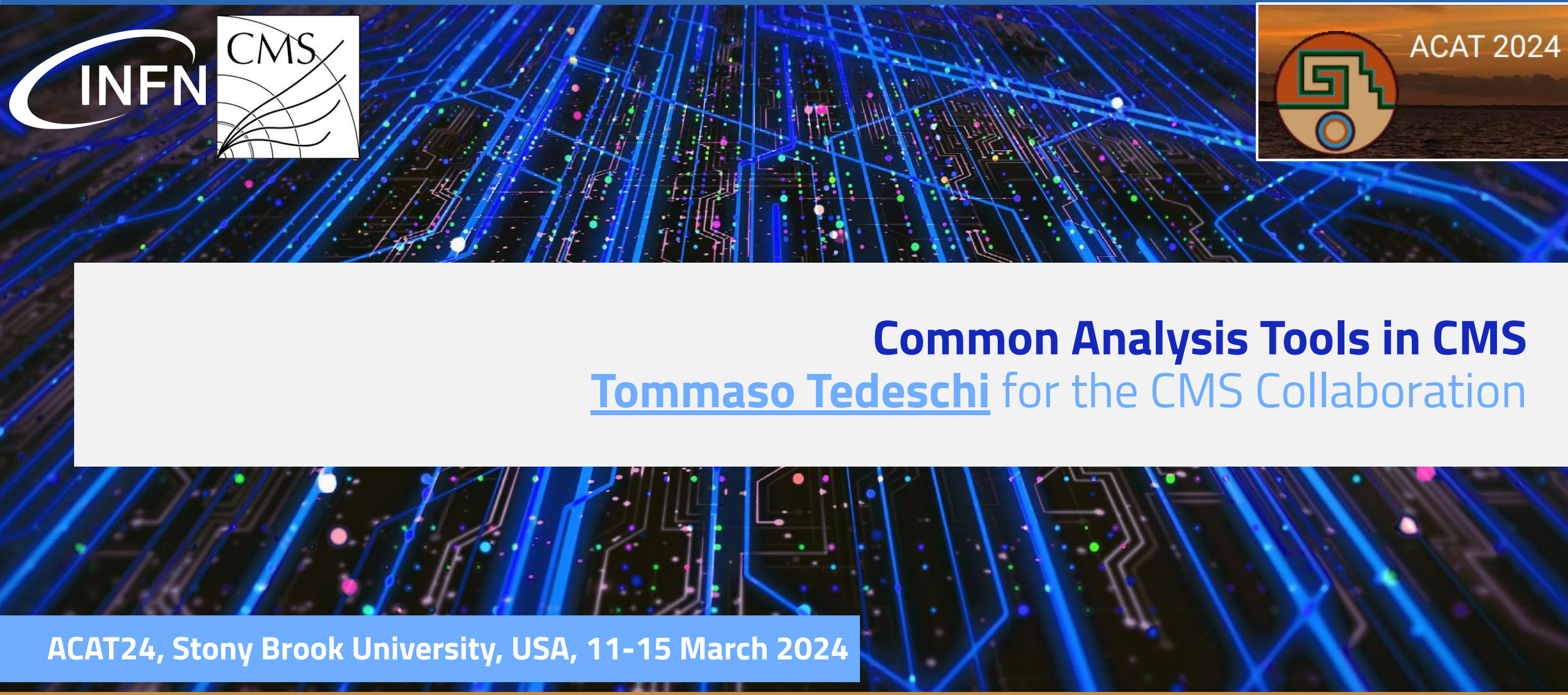
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing



Common Analysis Tools in CMS

Tommaso Tedeschi for the CMS Collaboration

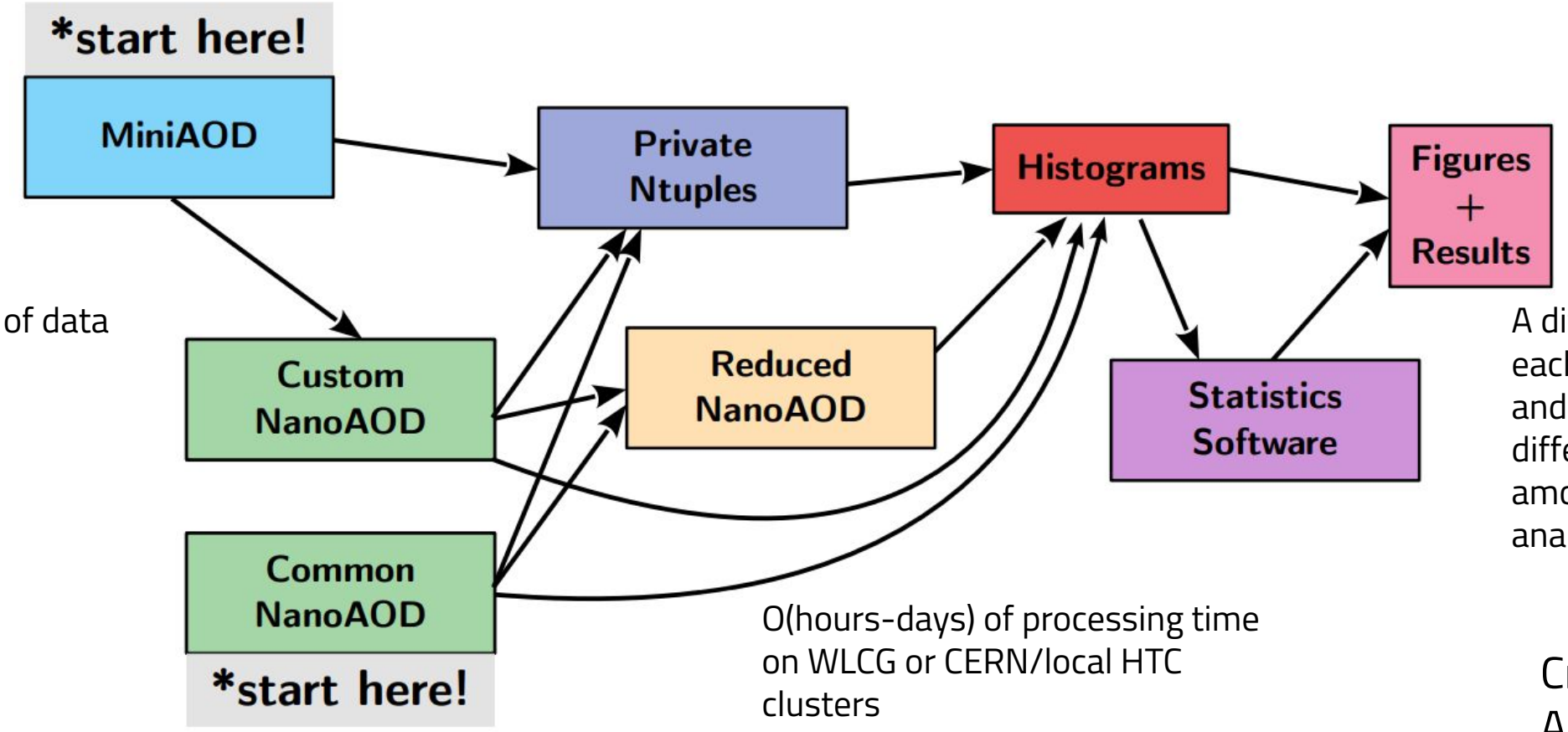
ACAT24, Stony Brook University, USA, 11-15 March 2024

A bit of context

- The **Compact Muon Solenoid (CMS)** experiment collaboration consists of over 4000 particle physicists, engineers, computer scientists, technicians and students from around 240 institutes and universities from more than 50 countries.
- Data collected (or simulated) by CMS are stored into **ROOT files**
  - Different data formats (**datatiers**) introduced to substitute the one containing the full set of objects created by the event reconstruction program (O(1MB) per event)
    - **AOD** (acronym of Analysis Object Data, introduced in 2011, ~2x smaller than the RAW),
    - **MiniAOD** (introduced in 2013, ~10x smaller than AOD),
    - **NanoAOD** (introduced in 2018, about an order of magnitude smaller than MiniAOD):
      - Use of basic data types (e.g. float, int, arrays),
      - Structure based on simple ROOT TTrees
      - Only variables related to high-level physical objects, including pre-calculated quantities related to their identification:
      - filtered using appropriate thresholds

A bit of context

Most common possible different workflows of O(100) ongoing CMS Run2/Run3 analyses



O(10-100) TB of data + Monte Carlo samples to be processed

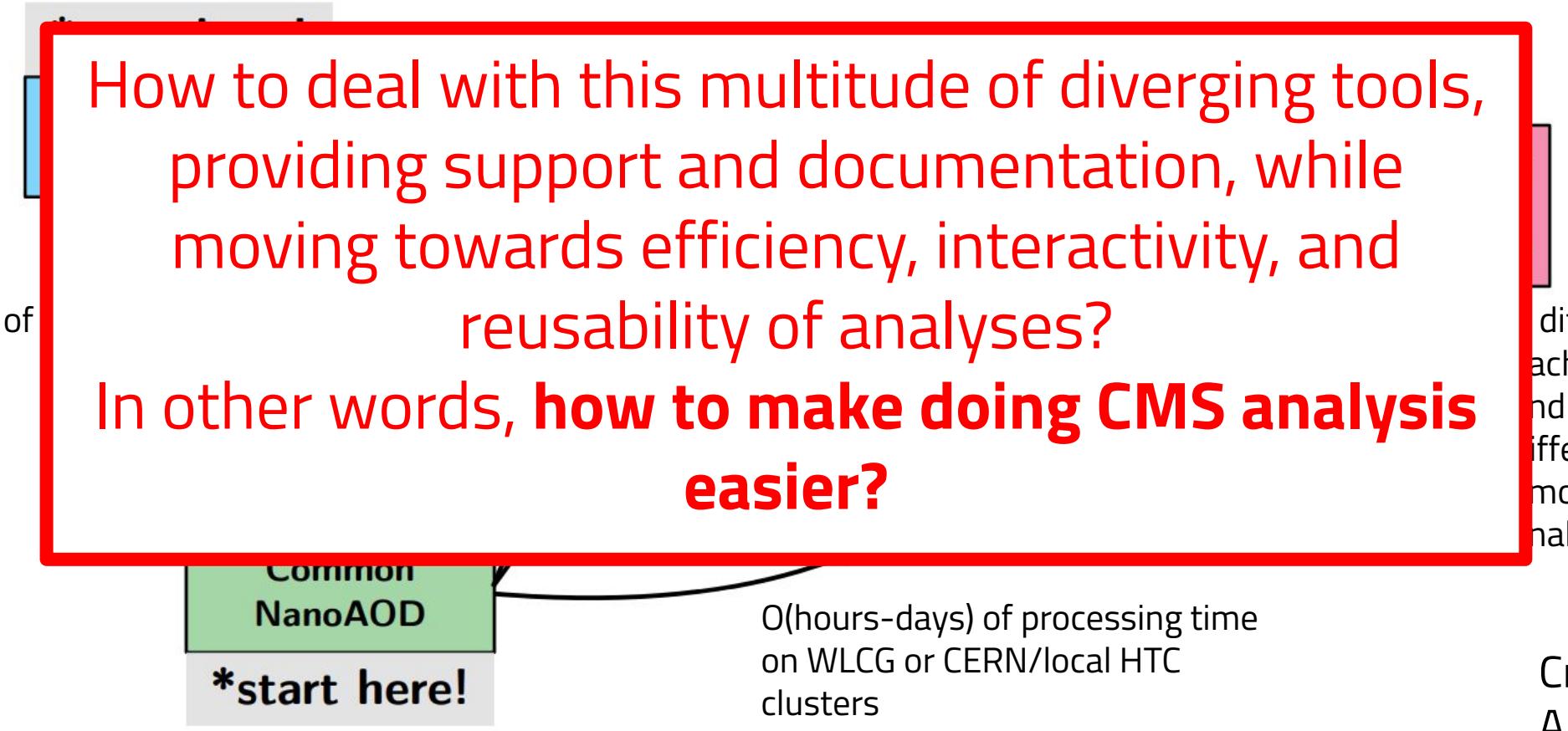
A different tool for each of these steps, and possibly different tools among different analysis groups

O(hours-days) of processing time on WLCG or CERN/local HTC clusters

Credits to Evan Altair Ranken

A bit of context

Most common possible different workflows of O(100) ongoing CMS Run2/Run3 analyses



How to deal with this multitude of diverging tools, providing support and documentation, while moving towards efficiency, interactivity, and reusability of analyses?
In other words, **how to make doing CMS analysis easier?**

O(10-100) TB of
+ Monte Carlo
samples to be
processed

different tool for
ach of these steps,
nd possibly
ifferent tools
mong different
nalysis groups

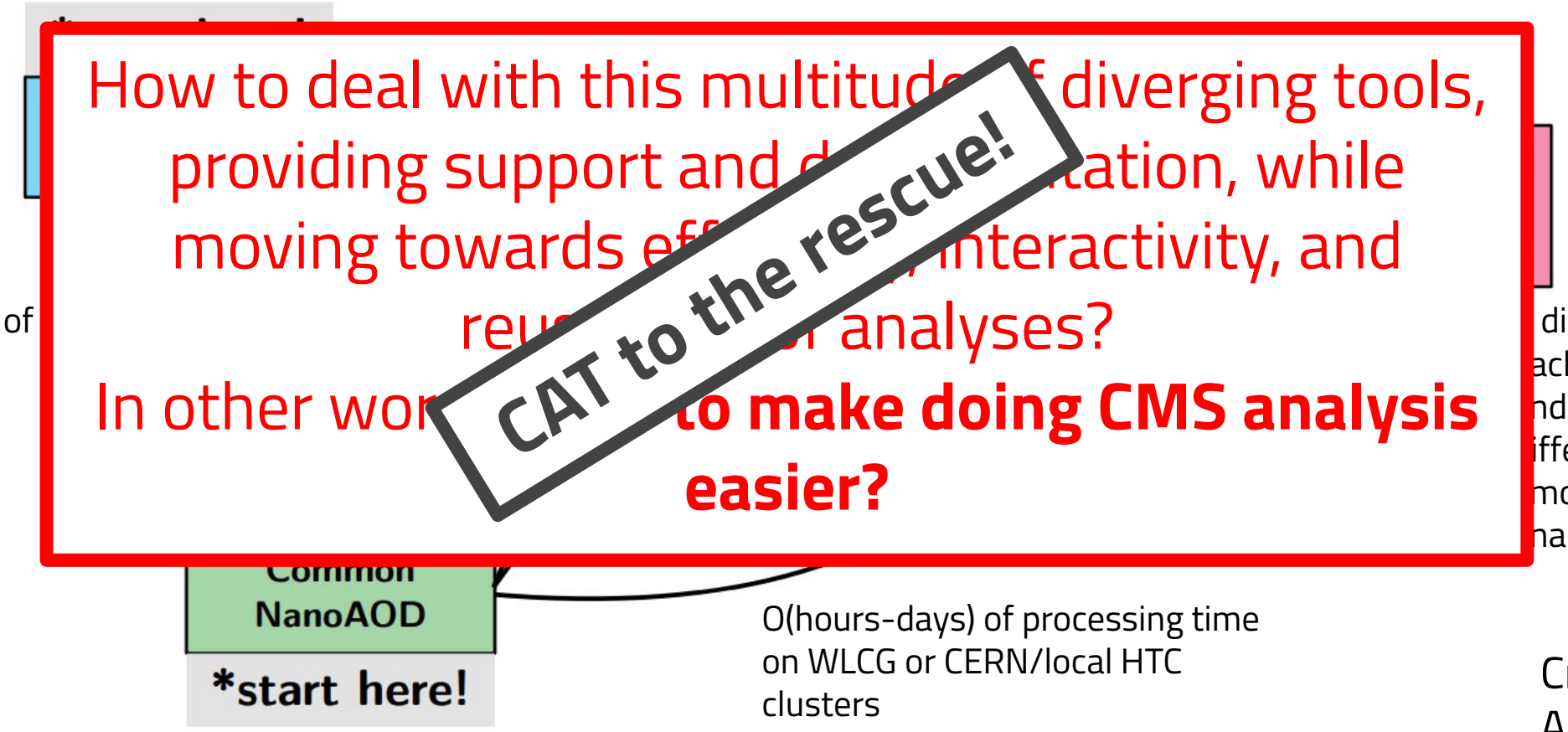
O(hours-days) of processing time on WLCG or CERN/local HTC clusters

Credits to Evan Altair Ranken

A bit of context

INFN CMS

Most common possible different workflows of O(100) ongoing CMS Run2/Run3 analyses



How to deal with this multitude of diverging tools, providing support and d… …tation, while moving towards eff… …, interactivity, and reu… …r analyses?

In other wor… **…to make doing CMS analysis easier?**

CAT to the rescue!

O(10-100) TB of + Monte Carlo samples to be processed

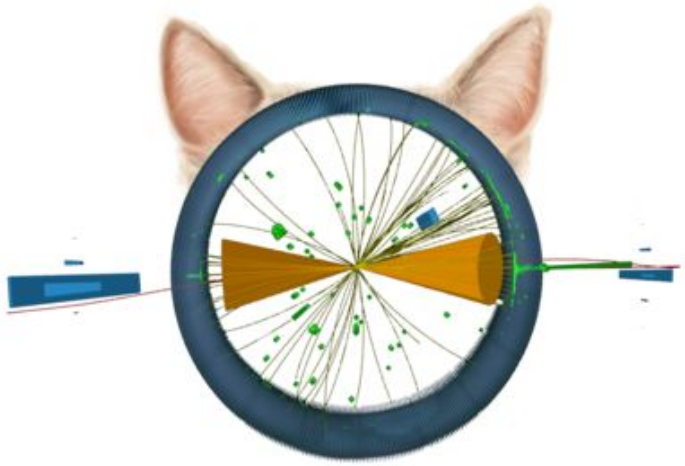
…different tool for …ach of these steps, …nd possibly …ifferent tools …mong different …nalysis groups

O(hours-days) of processing time on WLCG or CERN/local HTC clusters

Credits to Evan Altair Ranken

What is CAT?

- The **CMS Common Analysis Tools (CAT)** group was established in Sep 2022 and is charged with two main tasks:
  - Take ownership of the development, maintenance and documentation of analysis tools of common interest
  - provide a forum to discuss developments of new analysis tools, offering guidance
- Its organization includes three subgroups:
  - **Data Processing Tools (DPROC)**
    - support, management, and development of tools running directly on the CMS centrally-produced datasets
  - **Workflow Orchestration and Analysis Preservation (WFLOWS)**
    - support, management, and development of tools for the orchestration of physics analysis workflows, promoting tools that ease the long-term reproducibility of analyses
  - **Statistical Interpretation Tools (STATS)**
    - support, management and development of statistical interpretation tools (most importantly of Combine, the RooStats / RooFit - based software tool used for statistical analysis within CMS)



CAT operations

CAT-related discussions, developments and disseminations happen on several venues:

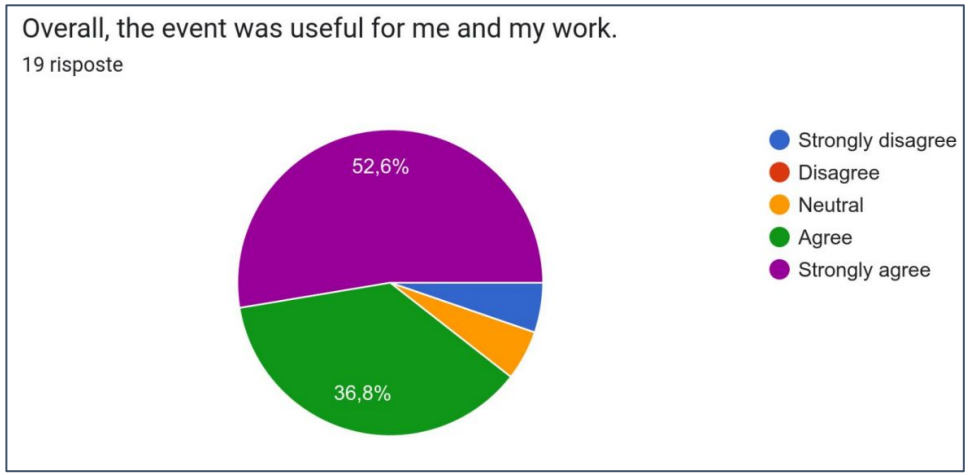
- **General meetings** every two weeks:
  - news and contributions on recent developments, with dedicated slots for introducing new work
- Main communication channel is **CMS-talk** (a customized version of Discourse)
- CAT **documentation** website
- Regular organization of **HaCAThons** (mixed hacking and training events):
  - 3 such events as far (~30/40 participants each):
    - **1st HaCAThon** - Apr 3-6 2023 (CERN)
      - CI for Combine input files, analysis examples, metadata management, systematics propagation
    - **2nd HaCAThon** - Sep 25-29 2023 (CERN)
      - plotting styles, docs, metadata management, analysis areas on GitLab, Combine unfolding tutorial
    - **3rd HaCAThon** - Feb 19-23 2024 (GGI - Florence)
      - Workflow management (with tutorials), metadata management, frameworks and tools developments, corrections application, open data, preparations for likelihood release

CAT operations

CAT-related discussions, developments and disseminations happen on several venues:

- **General meetings** every two weeks:
  - news and contributions on recent developments, with dedicated slots for introducing new work
- Main communication channel is **CMS-talk** (a customized version of Discourse)
- CAT **documentation** website
- Regular organization of **HaCAThons** (mixed hacking and training events):

Very useful events according to participants feedbacks



CAT Docs

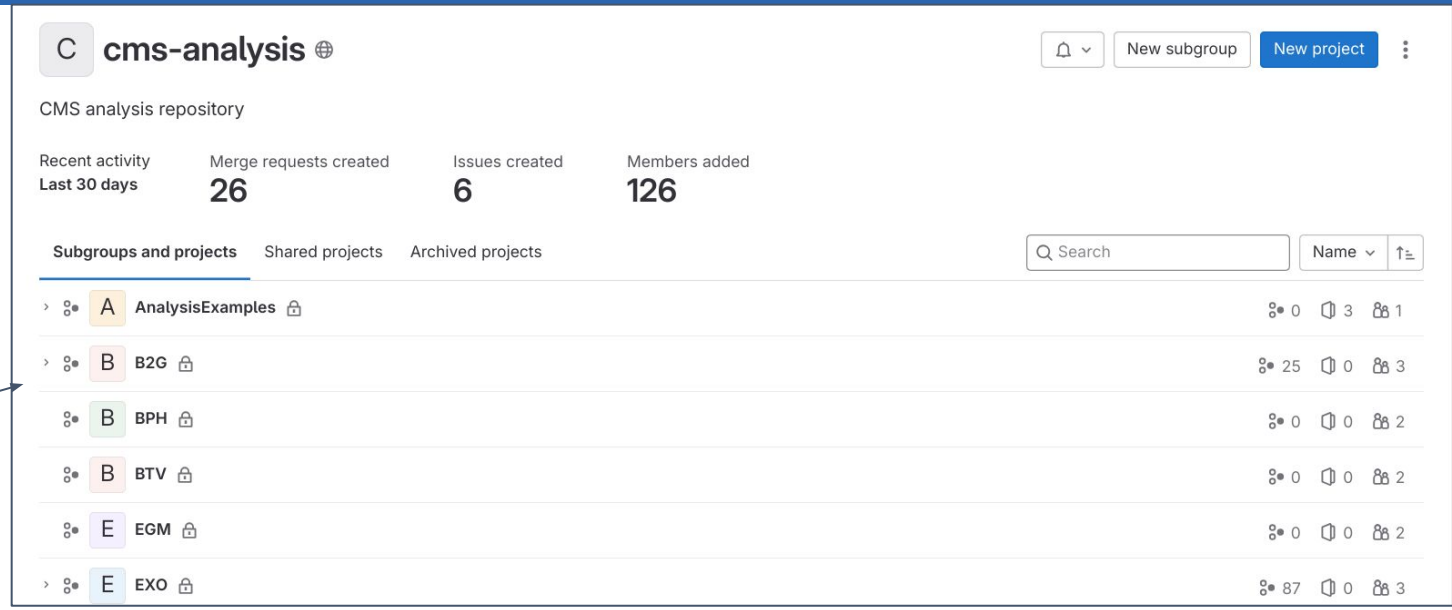
CAT continuously collects info expanding the current **CAT documentation website**, which now includes several items:

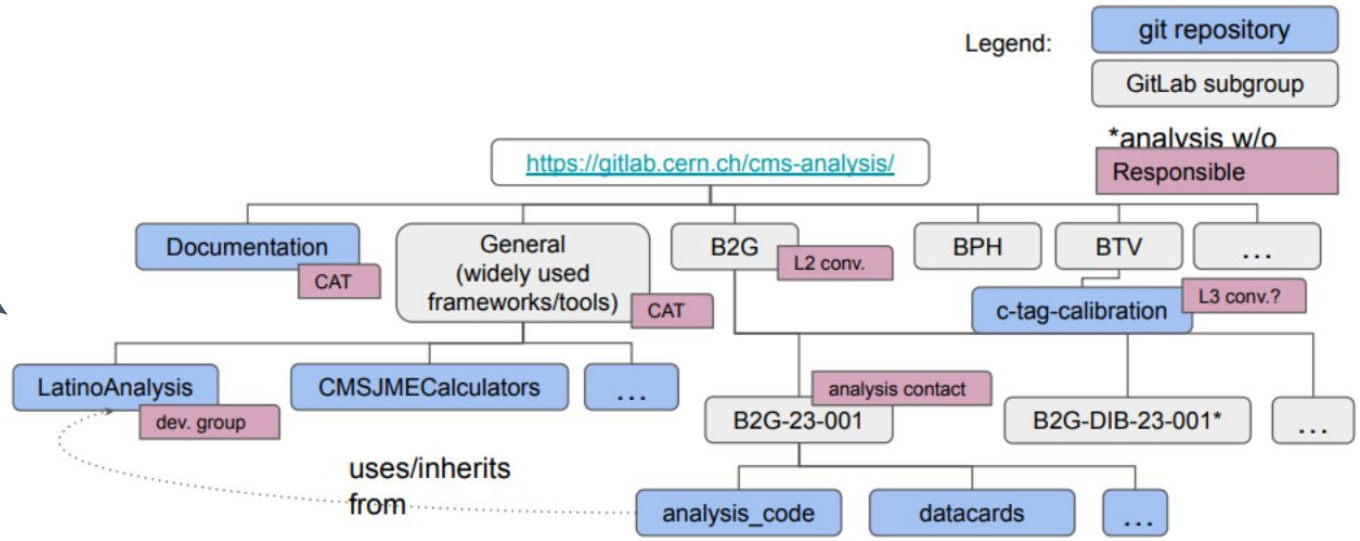
- **Recommendations** for CMS NanoAOD analysis and instructions on how to setup analysers' analysis code areas
- Overview of **supported tools** for data processing, workflow management and statistical analysis and any useful snippet
- Collection of links to Collaboration-wide accessible **Analysis Facilities** (in addition to LXPLUS and SWAN services)
- **Links** to useful tutorials and communication channels
- **Plotting** guidelines

Analysis code areas

With the goal of reusability, reproducibility and preservation of analysis, CAT now hosts **unified code areas for analyses**

- CAT asks that analysis code is at least mirrored there if not directly developed
- This procedure is already well established for Combine input files
- With newer frameworks, analysis code is represented by just a **configuration layer** (implemented with one or more files) on top of a common framework
- CAT encourages the implementation of **CI** via **templates**:
  - with the aim to make it easy for users to use CI to check their code



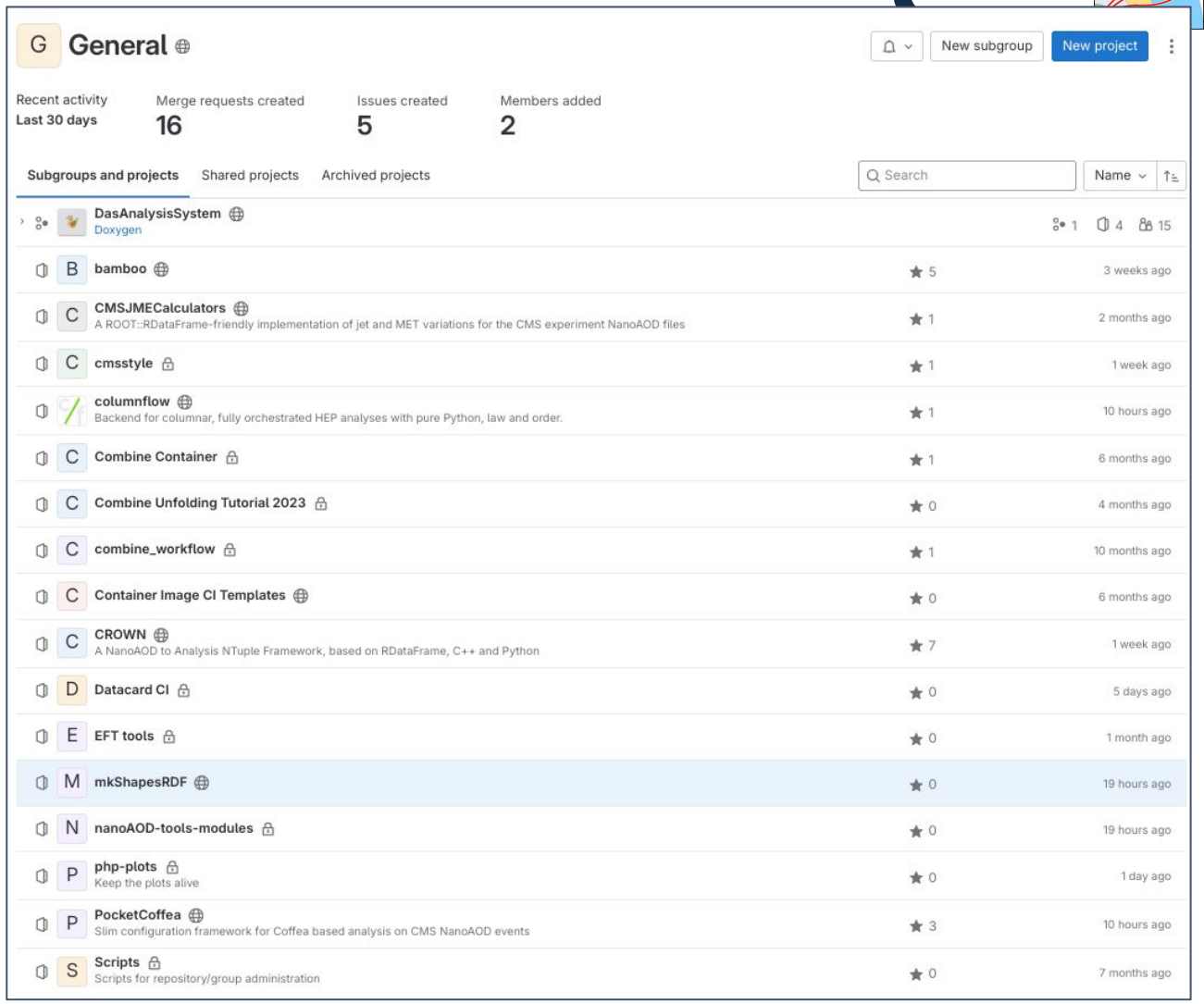


Supported tools

CAT **supported tools** are CMS specific tools with community support that:

- are residing or mirrored on cms-analysis/General or in CMSSW
- are actively developed, documented and maintained by identified support teams
- are supported via CMS-talk

A dedicated page in CAT docs describes their functionalities and point to relevant documentation



Supported tools

Analysis frameworks for both physical object studies and end-user analysis are supported aiming at **declarativeness**, **efficiency** and (quasi-)**interactivity** of analysis, reducing time-to-insight

- mostly based on emerging next-gen data processing tools, **ROOT's RDataFrame** (RDF) and **HSF's Awkward Arrays/Coffea**
- targeting NanoAOD format

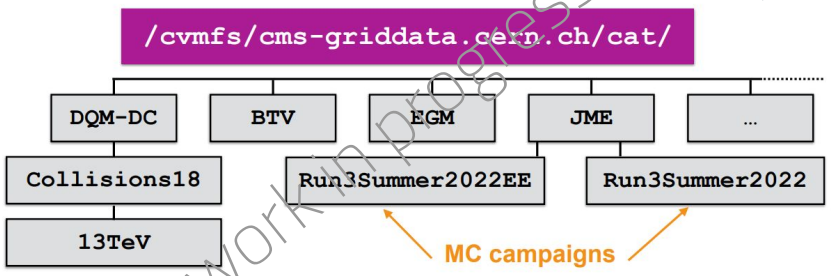
Such frameworks are, at the moment:

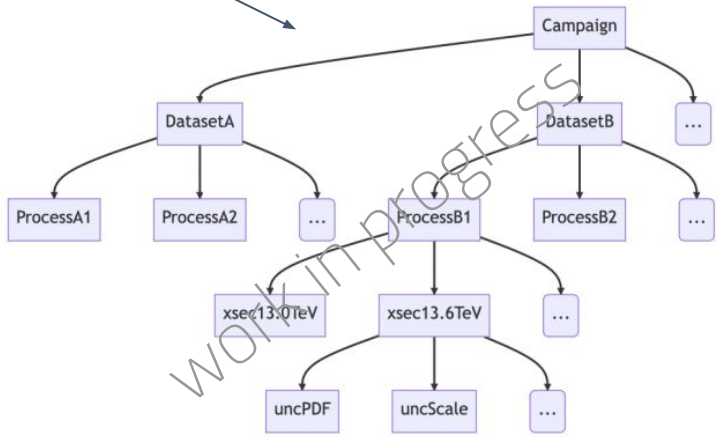
- **nanoAOD-tools:** legacy pyROOT-based sequential framework to skim/extend nanoAODs, and produce plots (modules here)
- **bamboo:** RDF-based python framework that allows to express analysis in a functional style
- **CMSJMECalculators:** RDF-friendly implementation of the recipes for jet and MET variations for CMS
- **CROWN:** RDF-based (C++ and python) framework to generate analysis ntuples (and friends)
- **columnflow:** python (Awkward Arrays)-based backend for columnar, fully-orchestrated HEP analyses
- **DasAnalysisSystem:** ROOT-based tools for analysis with high-level objects
- **PocketCoffea**: configuration framework for Coffea-based analyses on NanoAODs
- **mkShapesRDF**: RDF-based framework for analyses on NanoAODs, which are implemented through config files

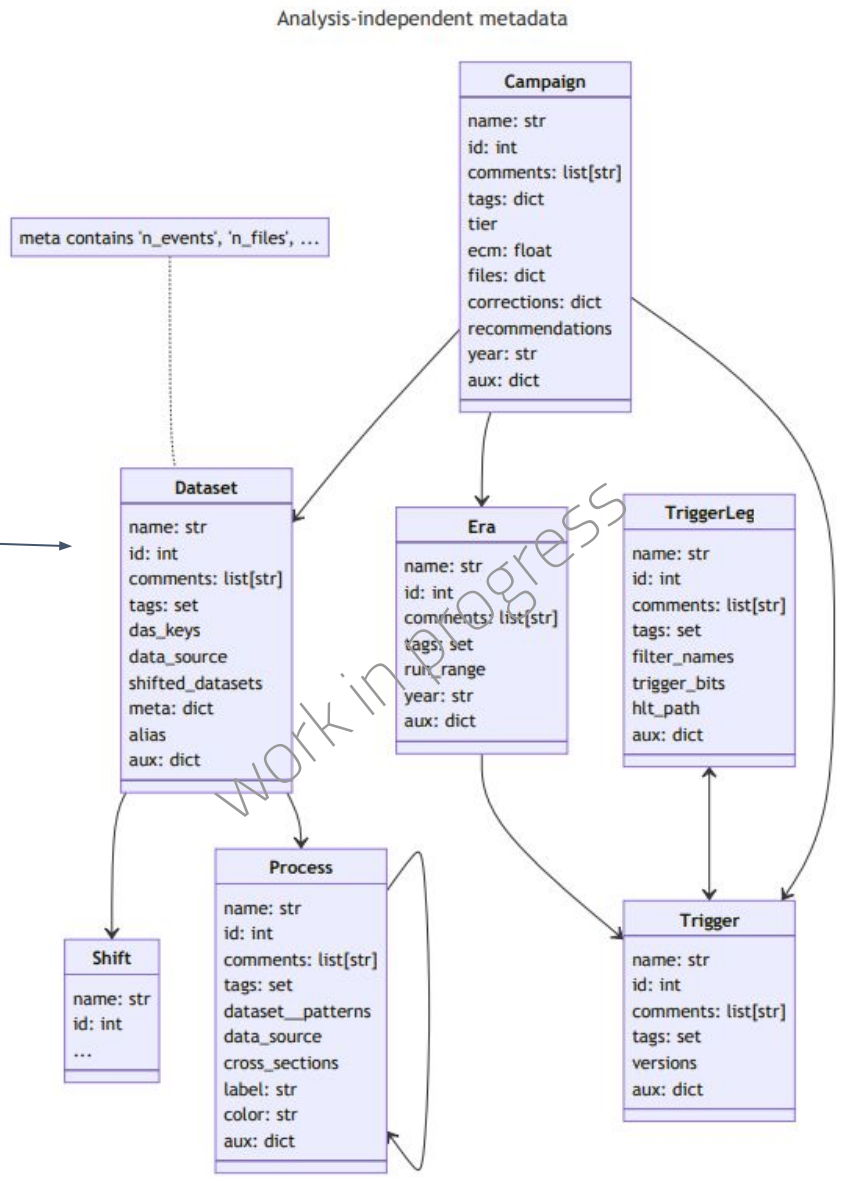
Metadata management

CAT also focuses on the metadata management:

- distributing analysis metadata via /cvmfs for easy (programmatic) access
  - easy-to-understand versioning
- ongoing work
  - to design a **schema** for metadata
  - on **tools** for accessing them
    - development of [order]







Analysis wflow management and preservation

CAT promotes the usage of **workflow management tools** in order to ensure reusability and reproducibility of analyses:

- this can be achieved with several tools, which are regularly presented (possibly along with tutorial sessions) at HaCAThons and general meetings. Examples are:

- Orchestration & workflow tools
  - luigi: Package for building complex pipelines with dependency resolution, workflow management, and visualization.
  - law: Extension of luigi with full decoupling of resources on HEP infrastructure
  - airflow: Platform to programmatically author, schedule and monitor workflows
  - snakemake: Workflow management system to create reproducible and scalable data analyses

- Preservation
  - HEPData portal: Repository for publication-related High-Energy Physics data
  - Reana: Reproducible research data analysis platform
  - Rivet: Toolkit for robust independent validation of experiment and theory
  - MadAnalysis: Framework for phenomenological investigations at particle colliders
  - CheckMate: Toolkit for checking models at terascale energies
  - SModelS: A tool for interpreting simplified-model results from the LHC

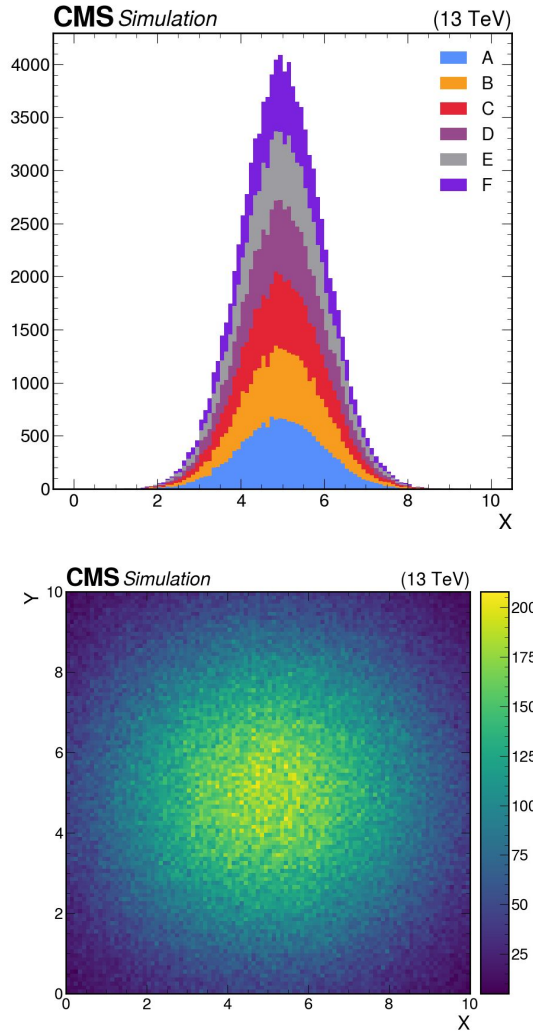
- **CI/CD** best practices are promoted:
  - work on enabling CI jobs **offloading on Analysis Facilities** has been also carried out

Plotting tools

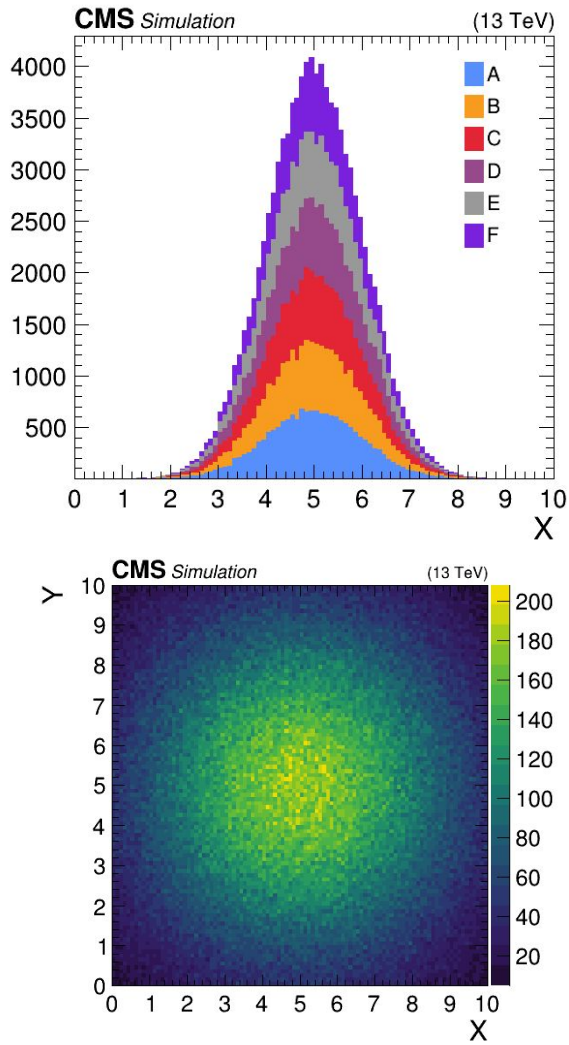
CAT also contributed to the recent update of **mplhep** and to the introduction of the new **cmsstyle** package, which allow CMS users to easily **produce production-ready plots**:

- this can be done either in the python/scikit-hep ecosystem (`mplhep`) or in the pyROOT ecosystem (`cmsstyle`)
- CMS default color-vision-deficiency friendly color schemes (recently voted by the Collaboration) are used as defaults

`mplhep` result



`cmsstyle` result



Plotting tools

Here is some example code to easily reproduce last slide's result with both tools

pip install mplhep

mplhep

```python
import numpy as np
import matplotlib.pyplot as plt
import mplhep as hep
import hist, uproot


h1d = hist.new.Reg(100, 0, 10, label="X").StrCat([], label="Sample", growth=True).Weight() \
        .fill(np.random.normal(5, 1, int(1e5)), np.random.choice(list("ABCDEF"), int(1e5)))

rf = uproot.recreate("test_file.root")

rf['h1d'] = h1d

for sample in sorted(list(h1d.axes[1])):
    rf[f'h1d_{sample}'] = h1d[:, sample]

rf.close()

# Load CMS style including color-scheme
hep.style.use("CMS")

# Setup matplotlib figure
fig, ax = plt.subplots()

# Plot histograms
h1d.plot1d(ax=ax, stack=True, histtype='fill', sort='label');

# Style
plt.legend()
hep.cms.label();
```

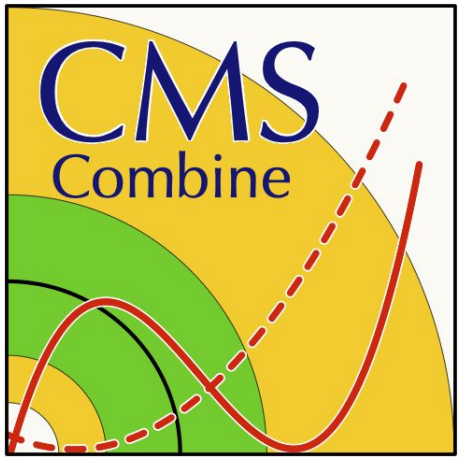
pip install cmsstyle

cmsstyle

```python
import ROOT as r
import cmsstyle as CMS

# File reading
f = r.TFile.Open('test_file.root')
th1_names = [k.GetName() for k in f.GetListOfKeys() if k.GetName().startswith("h1d_")]
th1s = [f.Get(sample) for sample in th1_names]

# Styling
CMS.SetExtraText("Simulation")
iPos = 0
canv_name = 'hist1d_root'
CMS.SetLumi("")
CMS.SetEnergy("13")
CMS.ResetAdditionalInfo()

# Plotting
stack = r.THStack("stack", "Stacked")

canv = CMS.cmsCanvas(canv_name,0,10,1e-3,4300,"X","",square=CMS.kSquare,extraSpace=0.01,iPos=iPos)

leg = CMS.cmsLeg(0.81, 0.89 - 0.05 * 7, 0.99, 0.89, textSize=0.04)

# Put samples in a dict {sample: th1} and draw
hist_dict = dict(zip([name.split("_")[-1] for name in th1_names], th1s))
CMS.cmsDrawStack(stack, leg, hist_dict)
```

Combine

CAT is also charged with contributing to the support of **Combine**, the RooStats / RooFit - based software tool used for statistical analysis within the CMS experiment



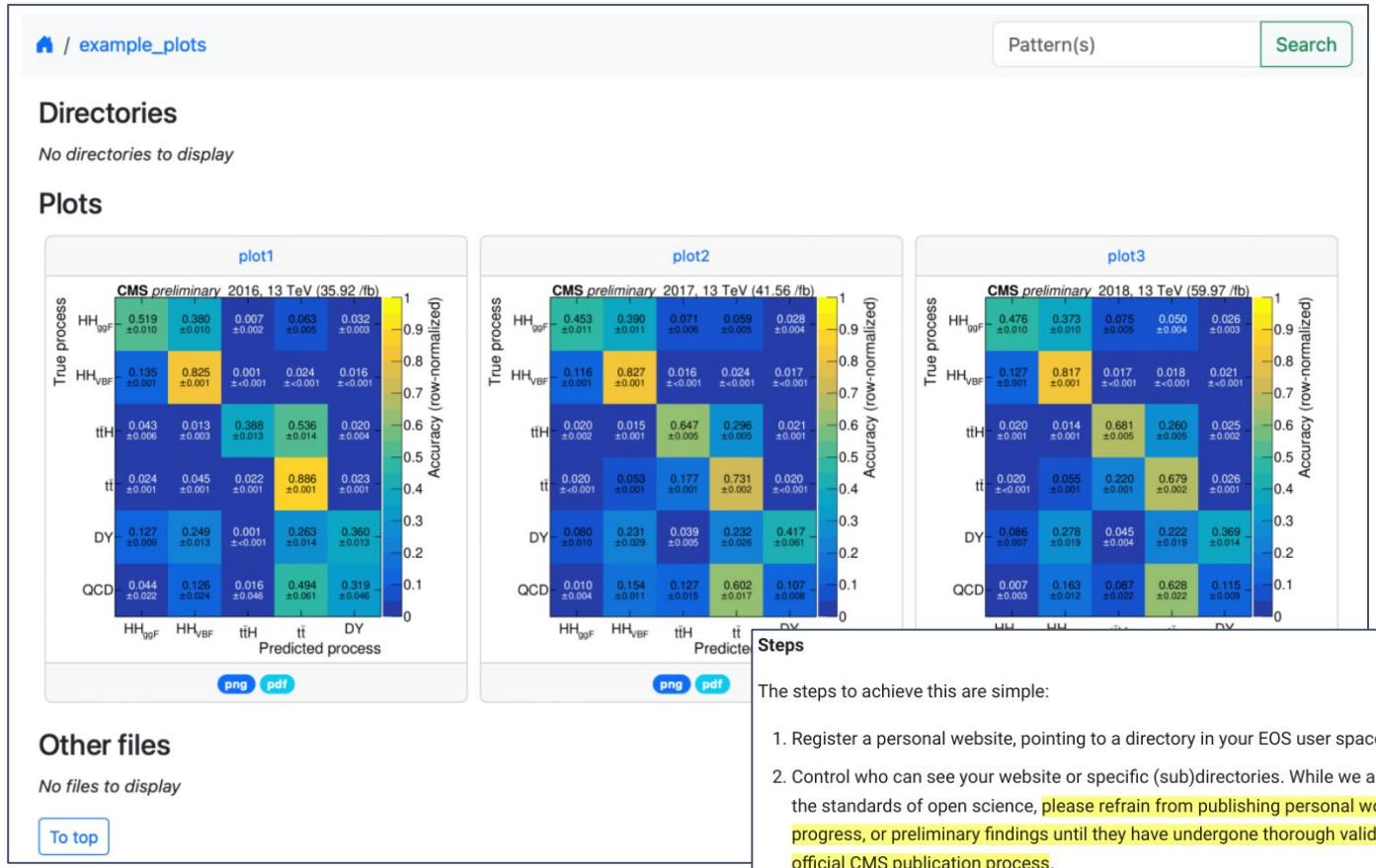
- it provides a command-line interface to many **different statistical techniques**, available inside RooFit/RooStats, that are used widely inside CMS
- statistical models are encapsulated using a **human-readable configuration file** (commonly referred to as "datacard")
- the package exists in **separate repository** wrt to CAT, on GitHub under https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit
- **documentation** is hosted here

Combine developments

- CAT contributed to the writing of a **paper on Combine**, describing its main features, that will be published soon
  - along with the paper, likelihoods will be released in the form of combine datacards + inputs
    - under a Creative Commons (CC) BY 4.0 licence
    - will be linked to HEPData record
  - CAT contributes to the discussion on a **HEP-wide standard** for the description of likelihoods in collaboration with other experiments:
    - HEP Statistics Serialization Standard (HS3) initiative is a promising candidate
  - Work on formalizing systematics conventions is also being carried out
  - CombineHarvester (framework for the production and analysis of datacards for use with the CMS combine tool) to be decoupled from CMSSW (official CMS software stack) and merged into Combine

Easy sharing plots via web pages

- CAT also produced an updated extensive step-by-step docs on how to enable the **interactive browsing of plots** and other files located in a EOS user directory
  - through a personal `YOUR_NAME.web.cern.ch` website
  - it is often helpful to be able to see multiple plots at once



Summary and outlook

- We have presented some of the achievements of CAT group in 1.5 years of operations
- Some progress has been done in all steps of data analysis, moving towards efficient, reproducible, and easy ways of doing analysis
- Still much work to do in various directions:
  - Moving further towards automation
  - Metadata unification
  - Widen (the already high) NanoAOD adoption
  - And much more!
- Next haCAThon in June!