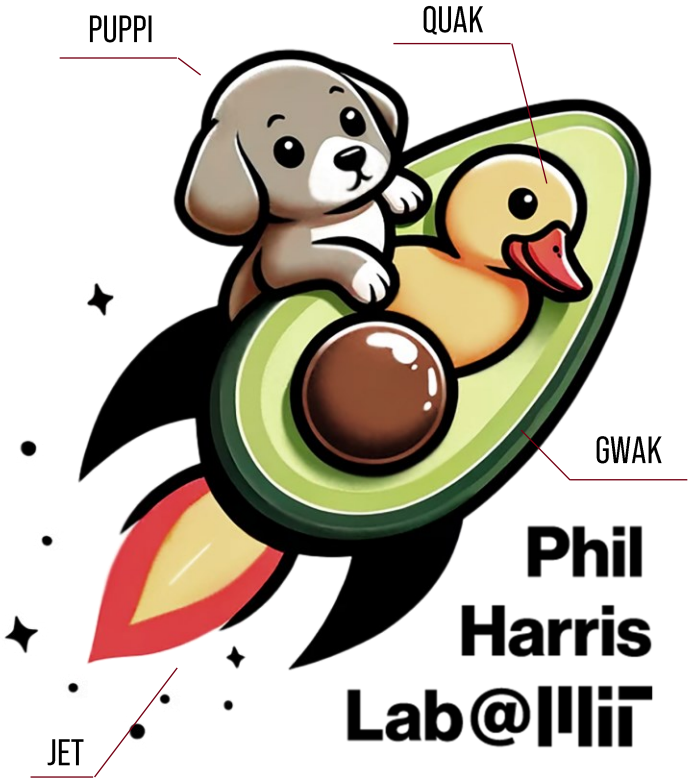
Massachusetts Institute of Technology

Common Analysis Tools in CMS

... bridging the gap from datasets to publications



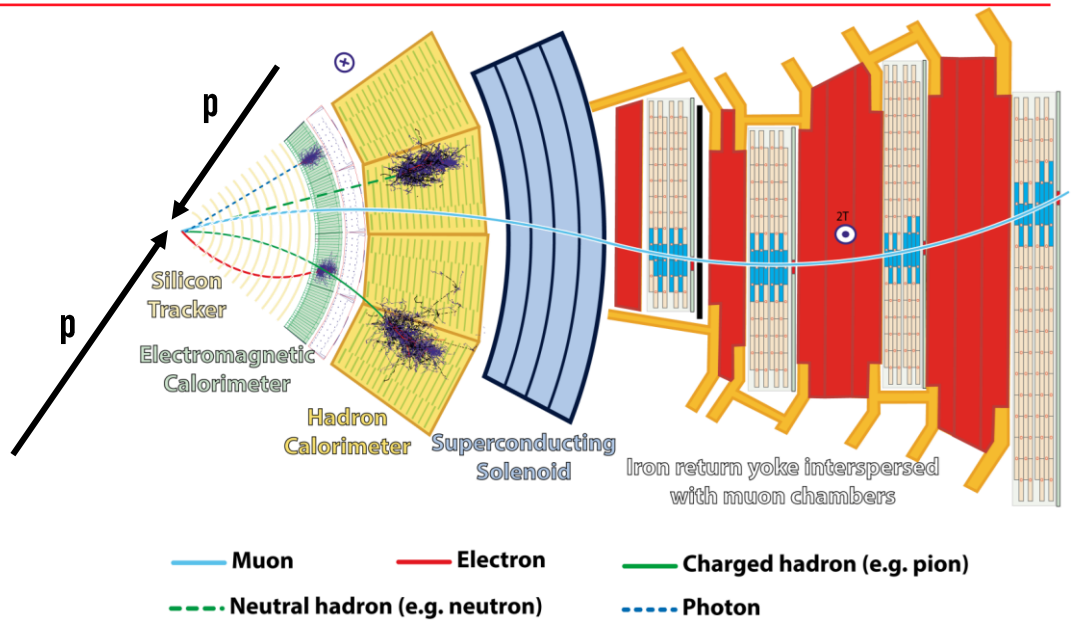
Andrzej Novak for the CMS Collaboration

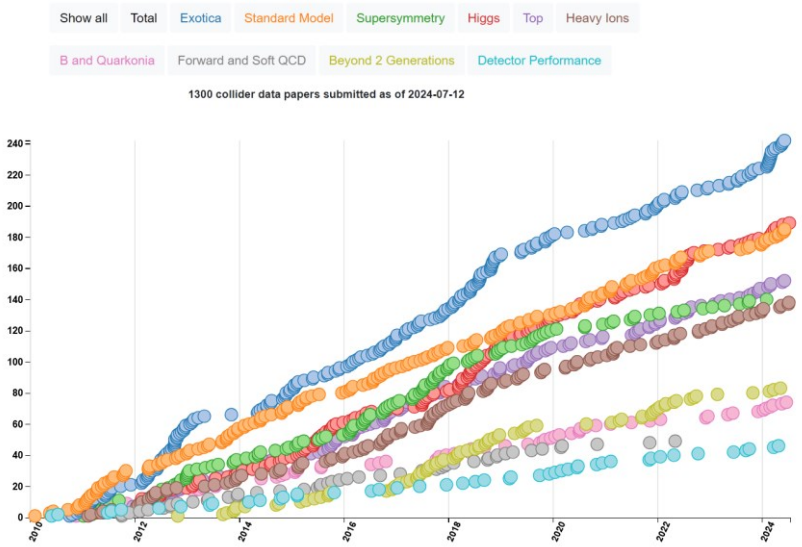
07/19/2024

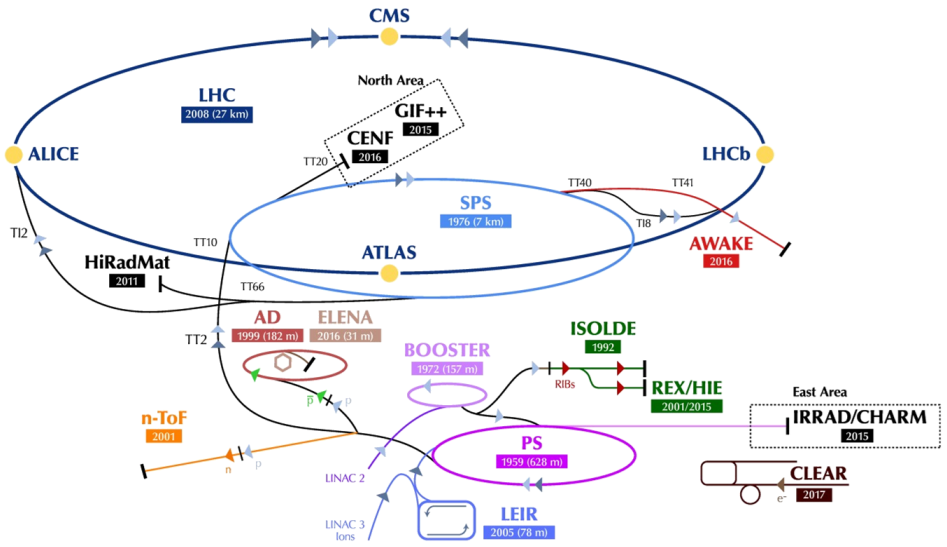
ICHEP 2024 (Prague)

The CMS Experiment

- Compact Muons Solenoid (CMS) Collaboration
  - 4000+ physicists, engineers, computer scientists, technicians, students from ~240 institutes and 50+ countries
  - LHC collision rate at 40 MHz, full detector readout ~ 1MB/evt
    - 100 MB/s of interesting collision events passing trigger (~5 years of data)
  - At any given point nearly 200 analyses are worked on…





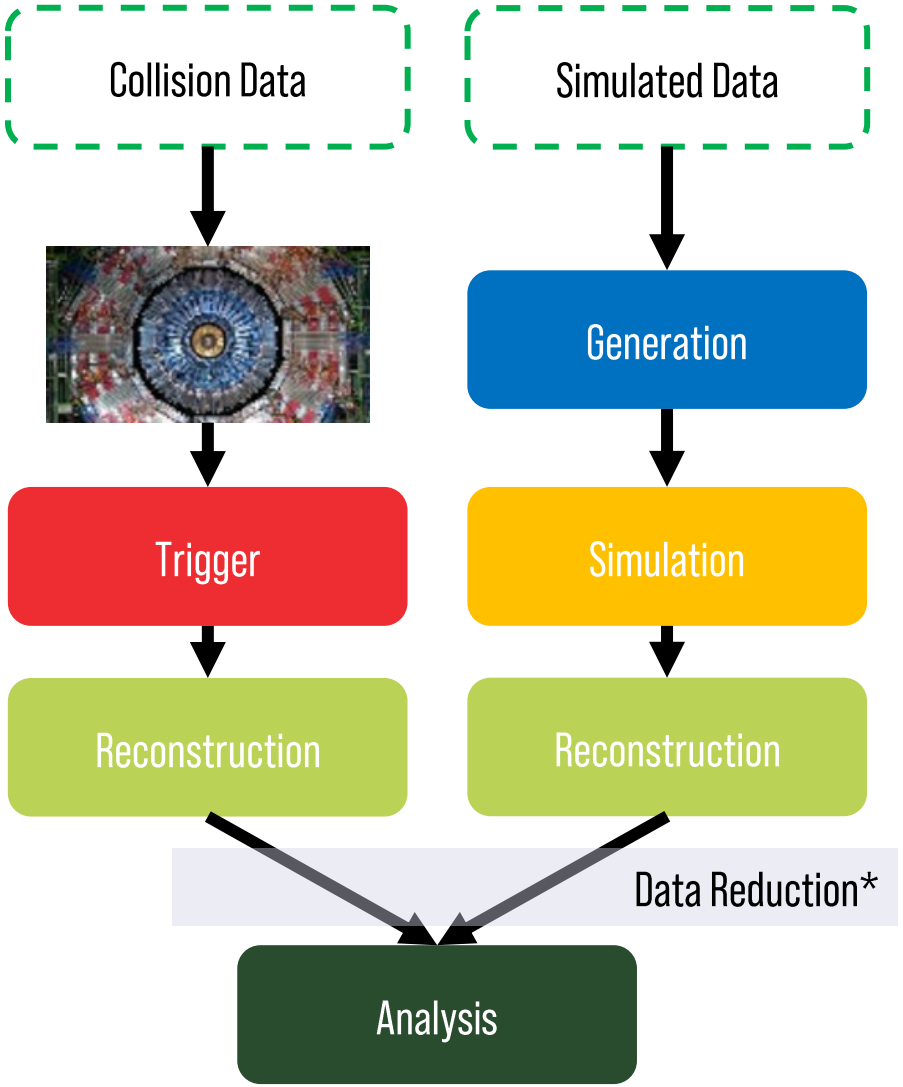


- How do we manage?

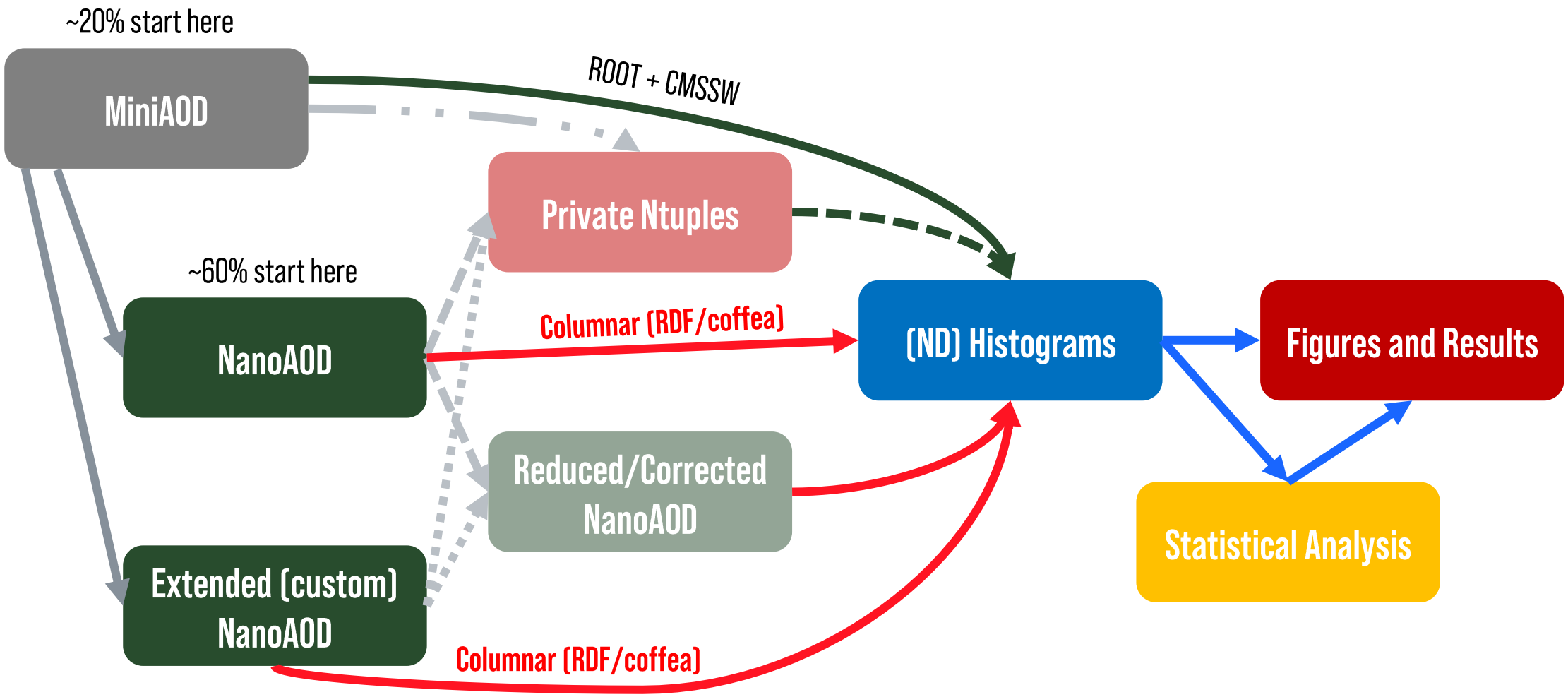
Data Processing & Tiers (Central Processing)

- RAW – detector readout
- AOD - Analysis Object Data (Introduced in 2011, ~2x smaller than the RAW)
- MiniAOD (Introduced in 2013, ~10x smaller than AOD)
- **NanoAOD** (~2018, ~5x smaller than MiniAOD)
  - Primary analysis format, sufficient in ~60% cases
  - "Flat" structure based on simple ROOT TTrees
  - Only basic data types (e.g. float, int, arrays)
  - Only variables related to high-level physical objects like m/p4/id of jet/e/mu

Tier	Event size [MB]	
	200 PU	140 PU
RAW	5.9	4.3
AOD	2	1.4
MiniAOD	0.25	0.18
NanoAOD	0.004	0.004



Typical Analysis Schematic

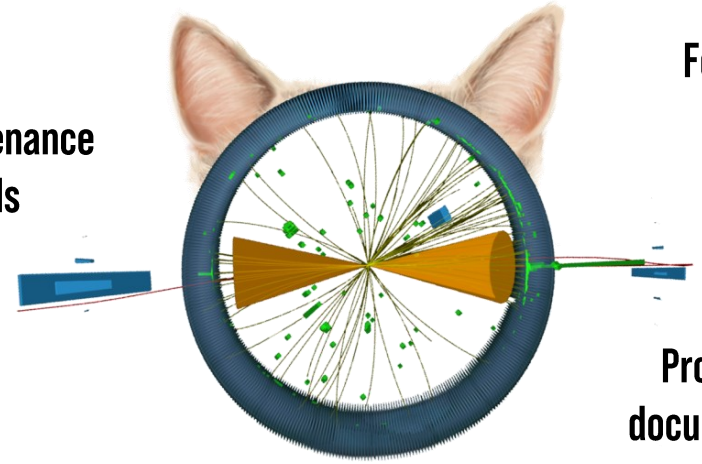


Common Analysis Tools (CAT) Group

Established in 2022 with the Physics Coordination area

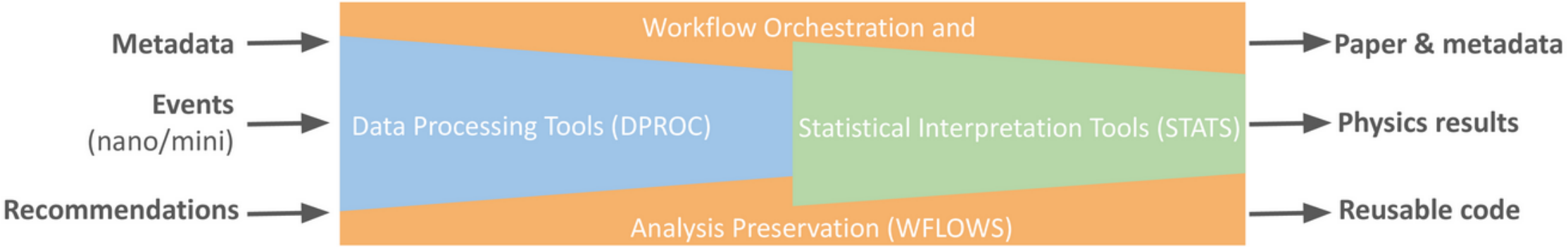
- On recommendation from “Analysis Tools Task Force”
- 2 conveners, 3 subgroups

Development and maintenance of CMS analysis tools



Forum to discuss developments, offer guidance

Provide centralized documentation to users



CAT Activities

General meetings every two weeks

- News about recent developments and contributions

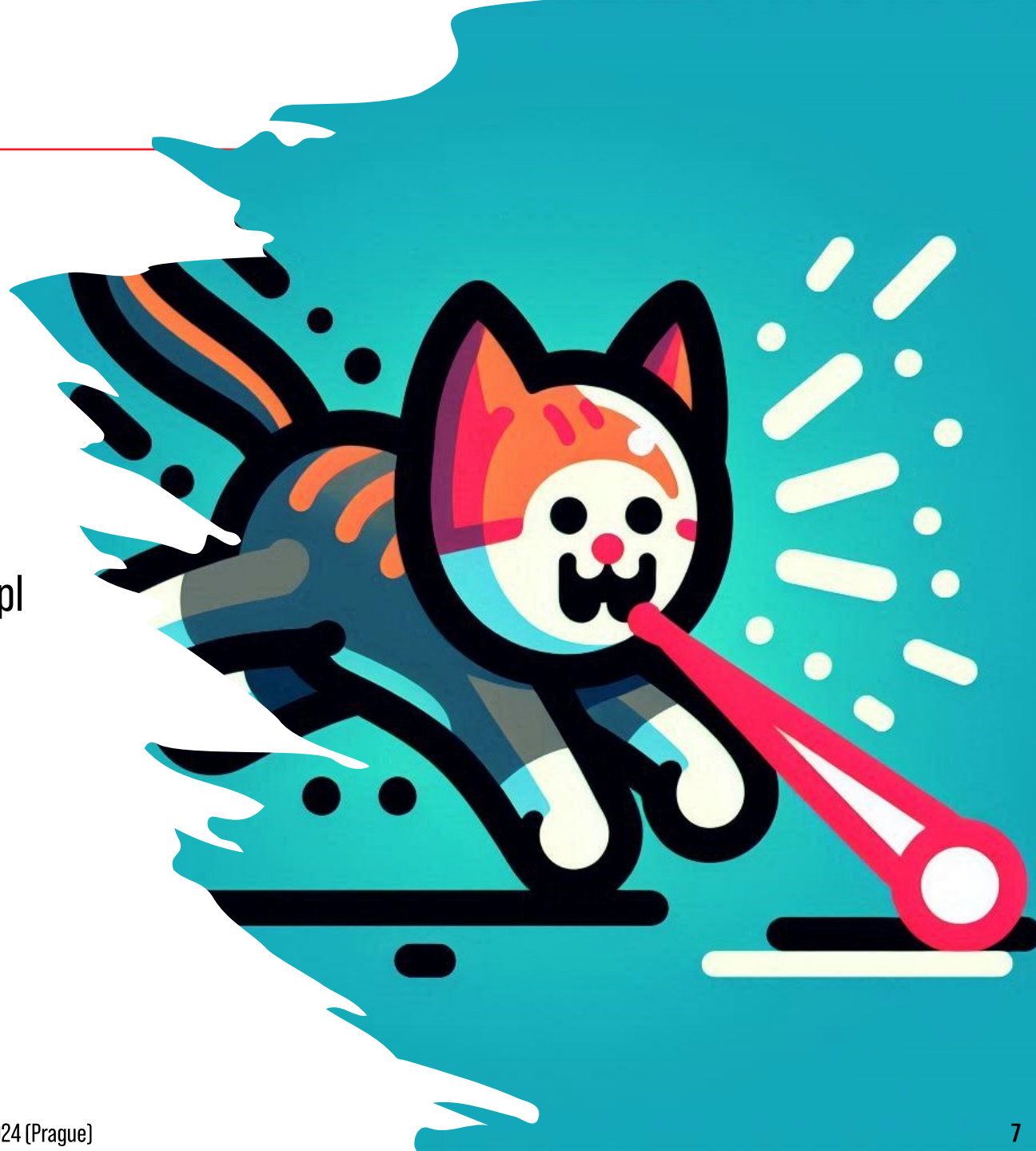
CMS-talk (a customized version of Discourse)

- Forum for users to ask questions and raise issues

CAT documentation website (for now internal only)

CAT **HaCAThons** (both hacking and training events) ~30/40 ppl

- 1st HaCAThon - Apr 3-6 2023 (CERN)
- 2nd HaCAThon - Sep 25-29 2023 (CERN)
- 3rd HaCAThon - Feb 19-23 2024 (GGI - Florence)
- 4th HaCAThon - Jun 17-28 2023 (CERN/Remote/Async)



CAT Activities

General meetings every two weeks

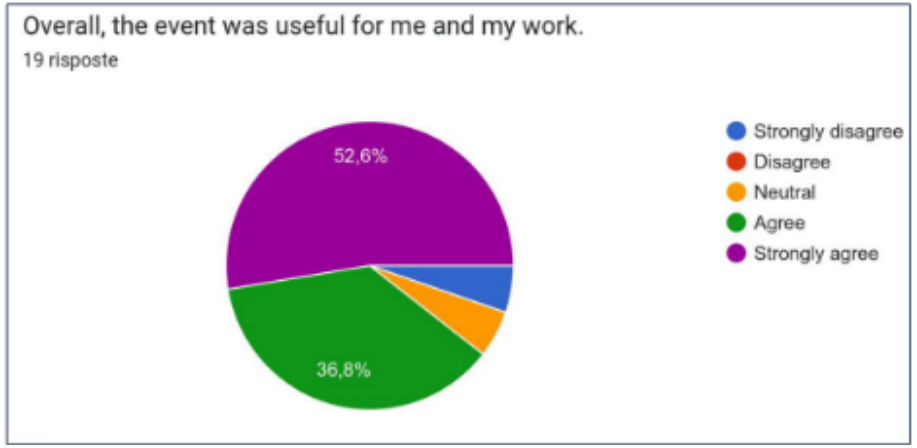
- News about recent developments and contributions

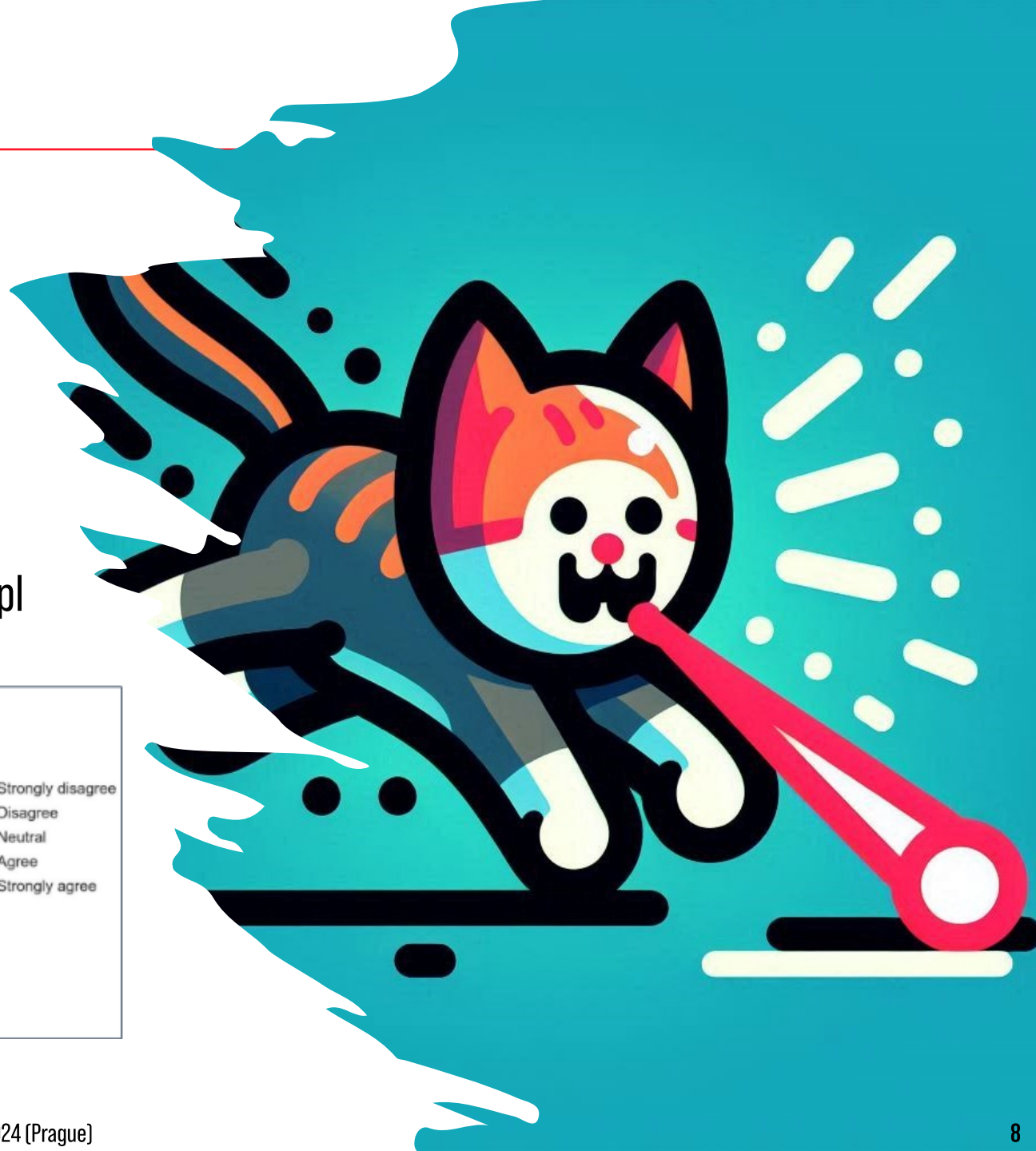
CMS-talk (a customized version of Discourse)

- Forum for users to ask questions and raise issues

CAT documentation website (for now internal only)

CAT **HaCAThons** (both hacking and training events) ~30/40 ppl



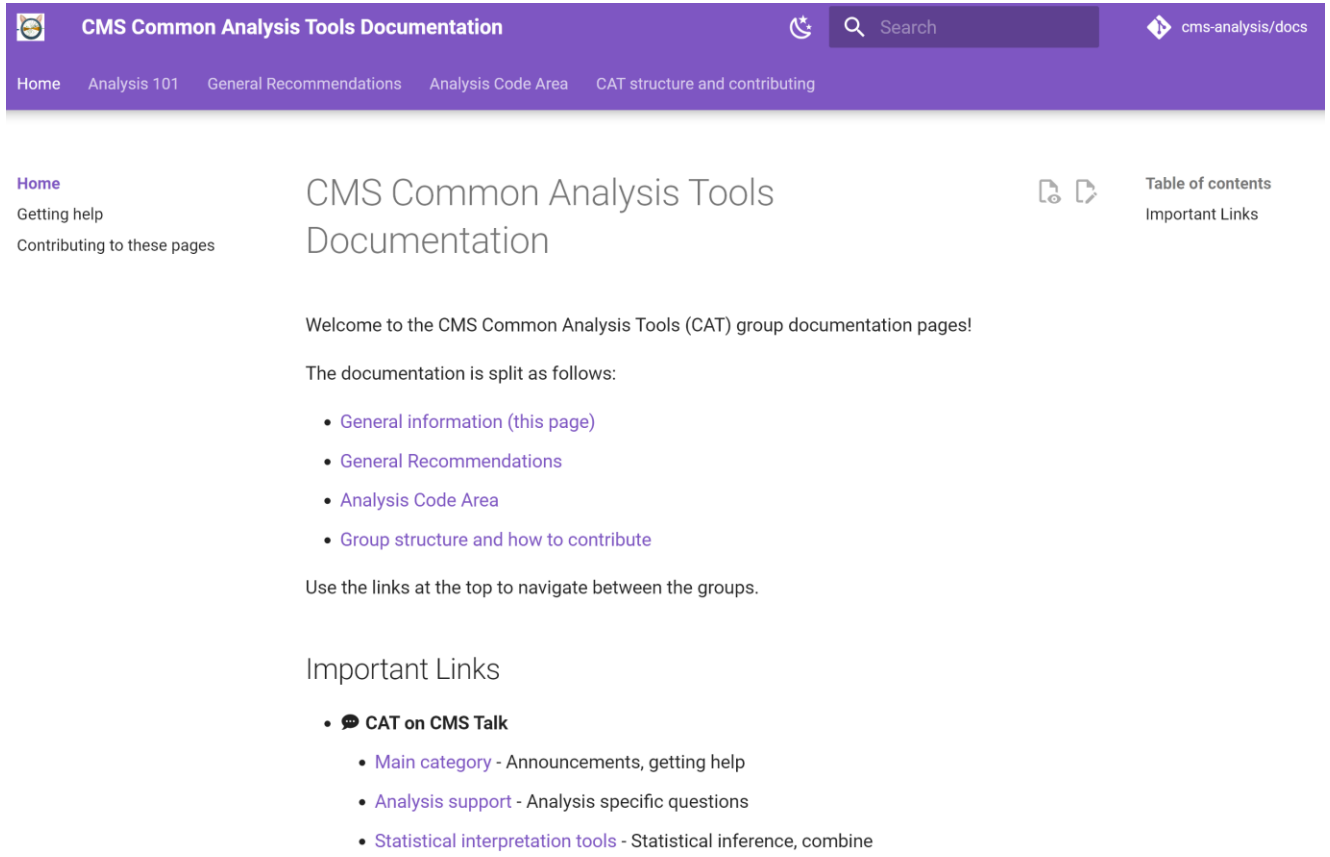


CAT Documentation

Maintaining a documentation page that should take a junior analyst from NanoAOD to limits...

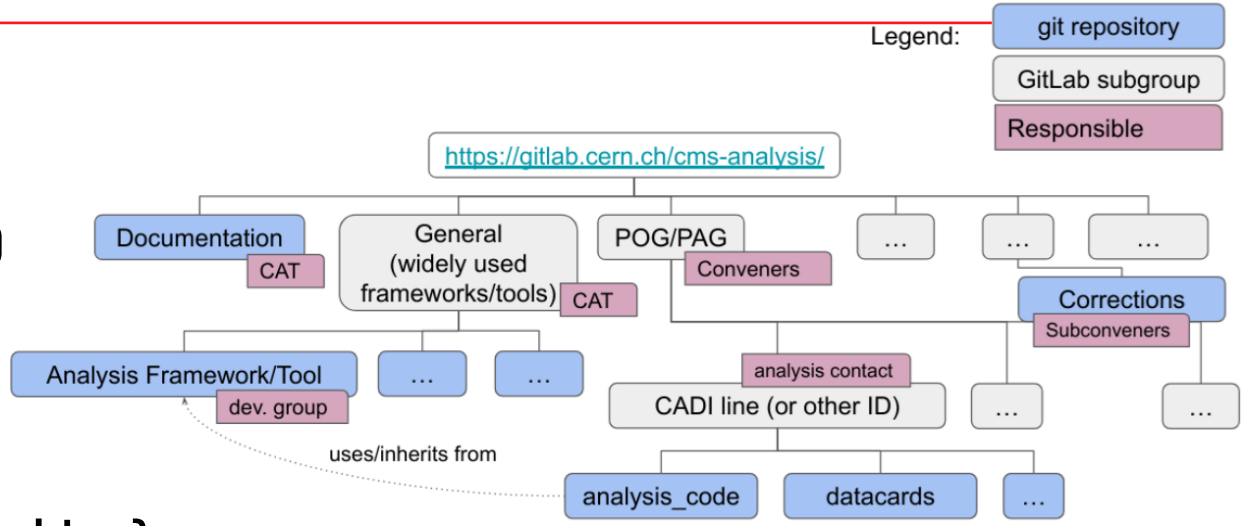
- Or at least link to all other relevant resources...

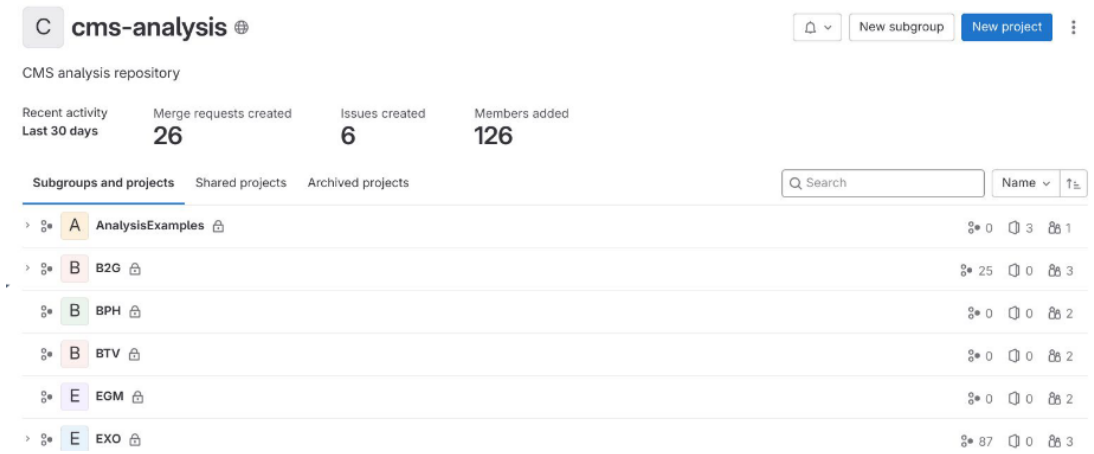
- Recommendations for **CMS NanoAOD analysis**
  - Instructions on how to setup code areas
- Overview of supported tools
  - **Data processing**
  - **Workflow management**
  - **Statistical analysis**
  - Miscellaneous snippets
- **Plotting guidelines**
- Collection of links available **Analysis Facilities**
- Links to useful **tutorials** and communication channels



Analysis Code Areas

- Version control is crucial to **reproducibility**
- CAT hosts unified code areas for analyses on GitLab
  - Full analysis code to be accessible in the repo
    - Either developed directly there or **mirrored** from private GitHubs
- Already well established for statistical analysis (combine)
- Ideally, analysis code just a configuration layer on top of existing framework code
- CI use is encouraged (tutorials/demos)
  - Some analyses fully ran in CI already





CAT Supported Tools

CAT arranges support/maintenance for CMS-specific tools

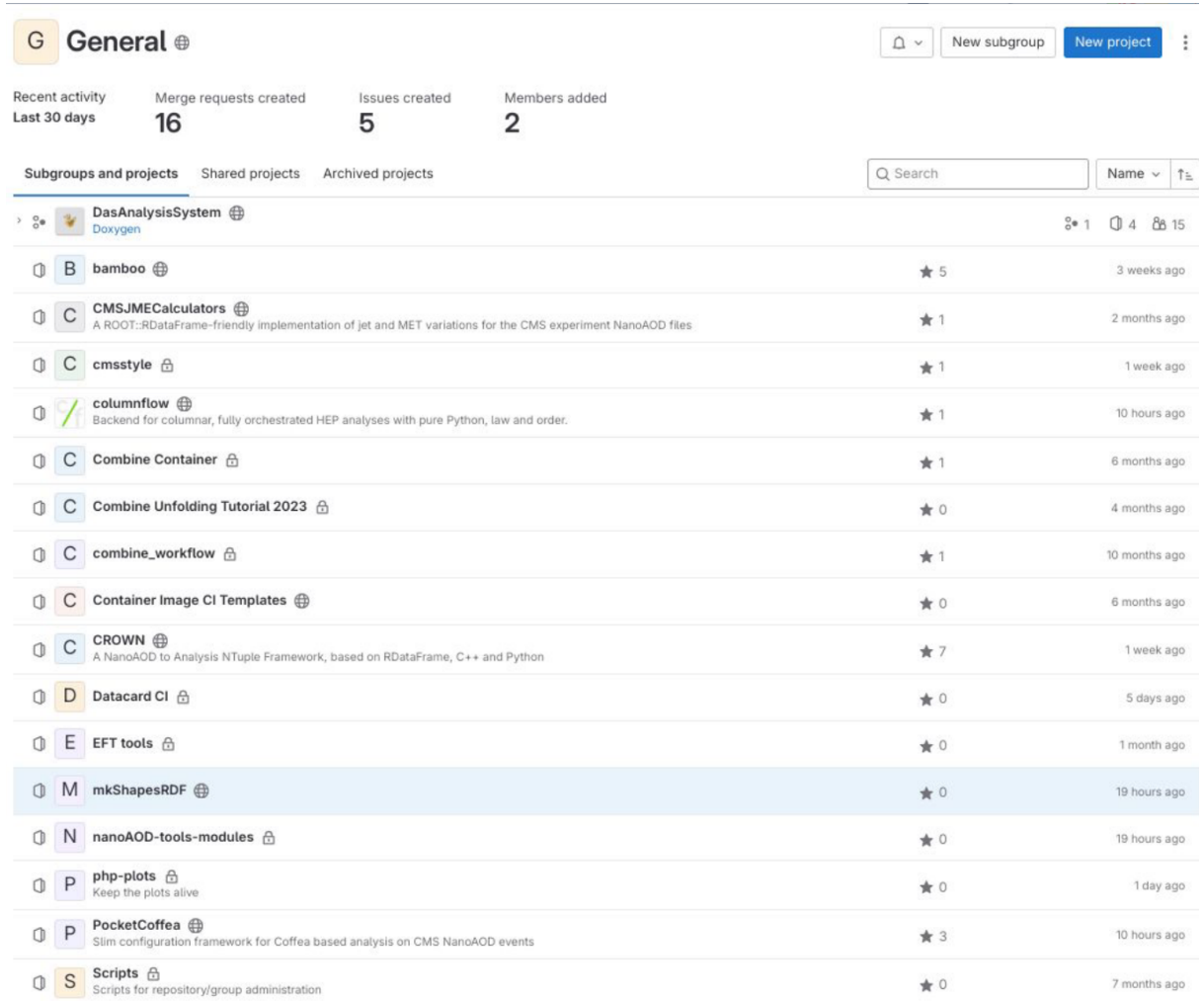
- Mostly coffea or RDF based
- Targeting NanoAOD format

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)



CAT Supported Tools

CAT arranges support/maintenance for CMS-specific tools

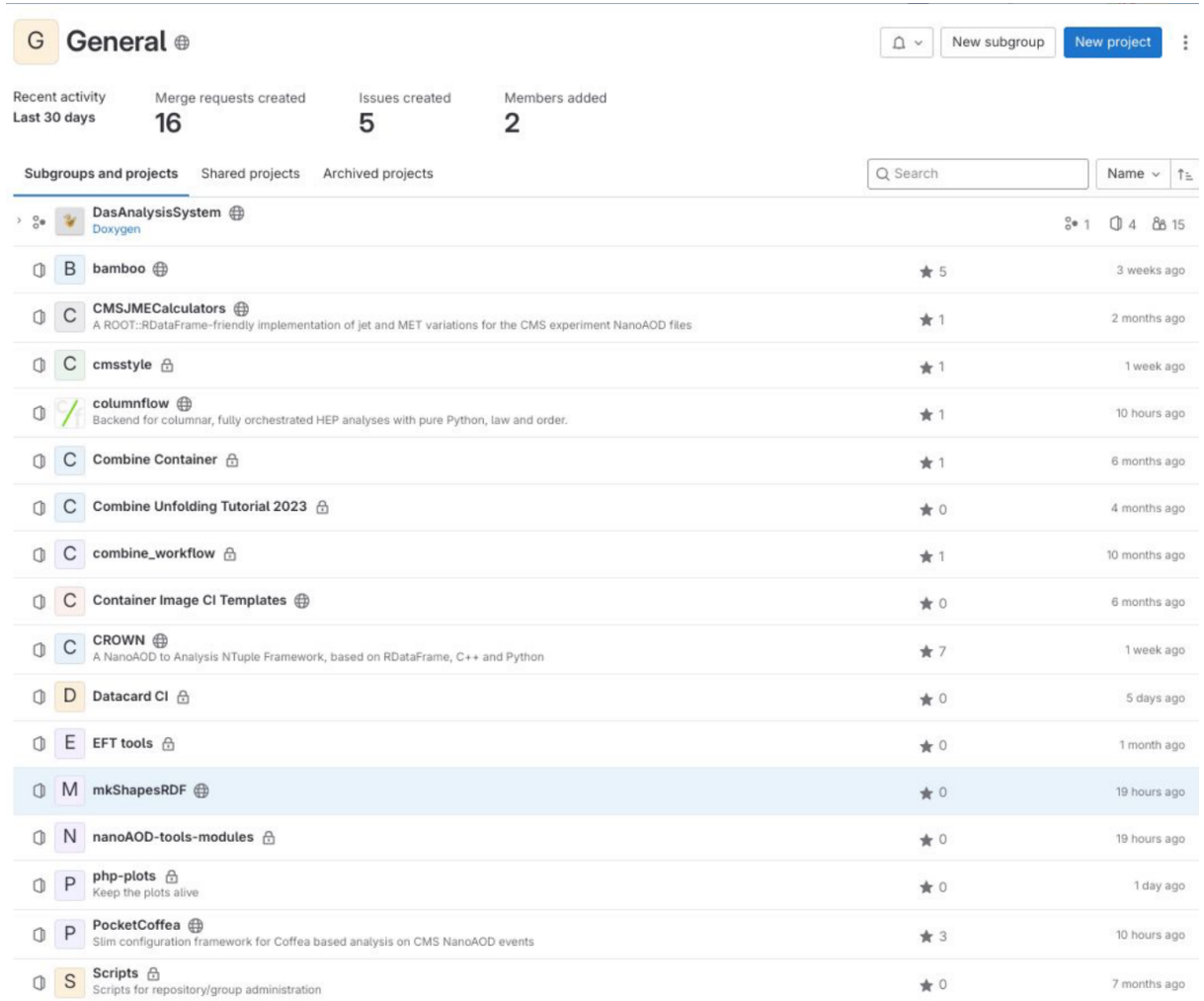
- Mostly coffea or RDF based
- Targeting NanoAOD format

columnflow: python (Awkward Arrays)-based backend for columnar, fully-orchestrated HEP analyses

DasAnalysisSystem: ROOT-based tools for analysis with high-level objects

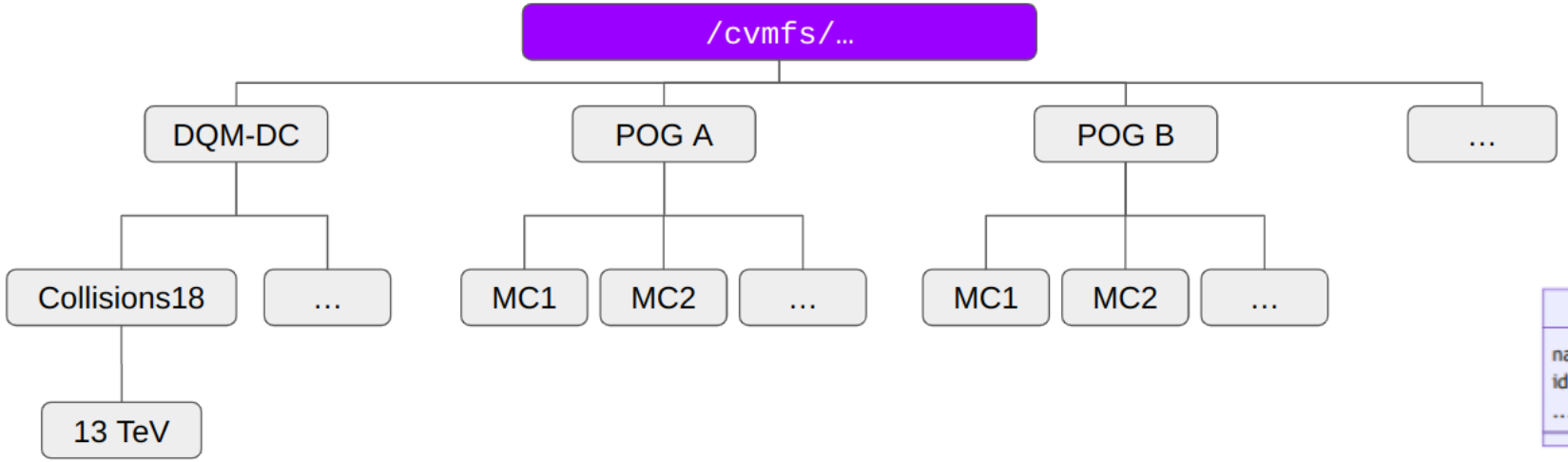
PocketCoffea: configuration framework for Coffea-based analyses on NanoAODs

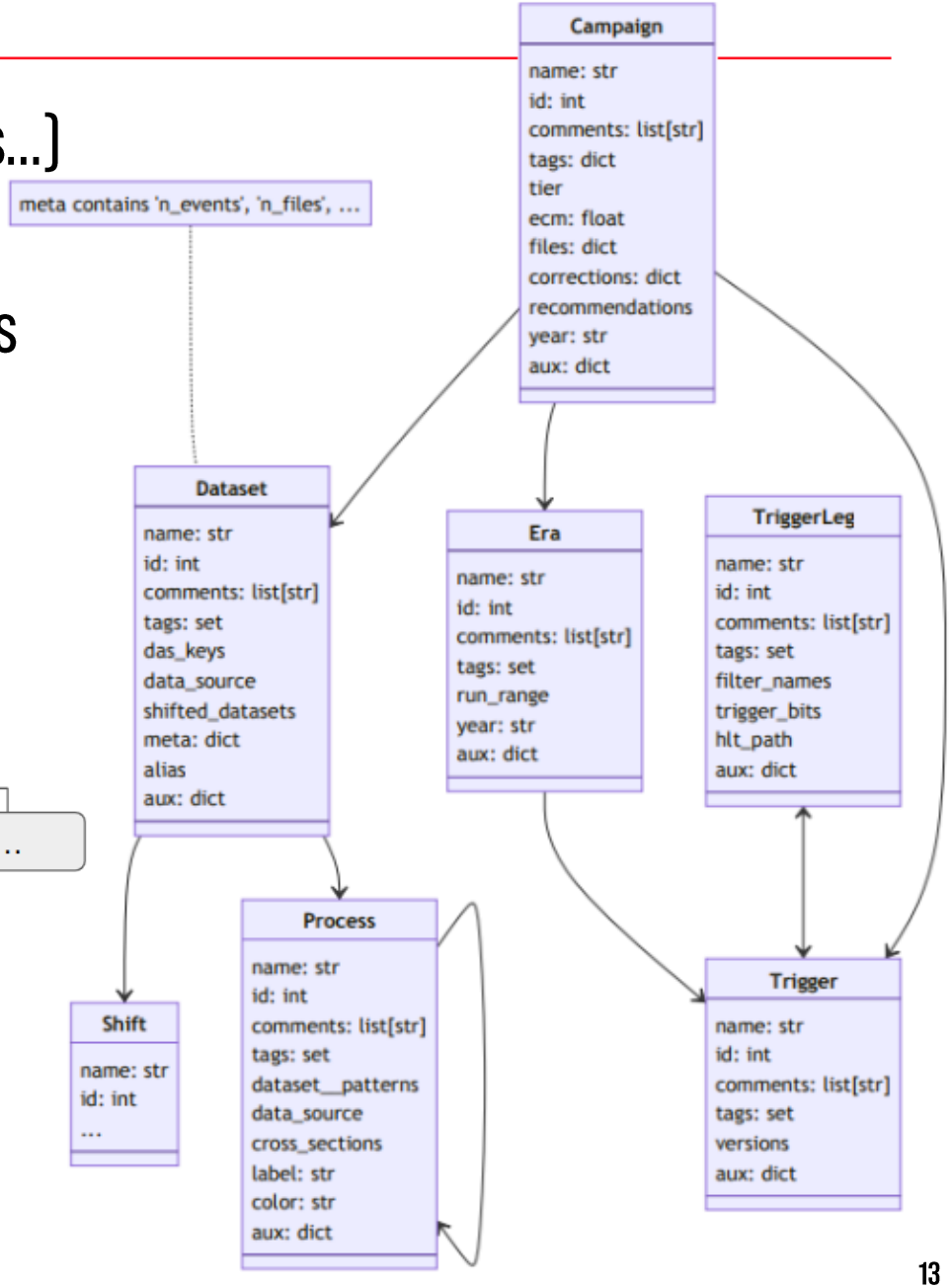
mkShapesRDF: RDF-based framework for analyses on NanoAODs, which are implemented through config files



Metadata Management

- Analyses require non-trivial amount of metadata (random TWikis...)
  - Cross-sections, DQM-validations, calibrations, corrections, systematics...
- Ongoing work to design a metadata schema and a tool for access
  - Majority of corrections already aggregated in a **single repo** in **json** format
  - Corrections applied via correctionlib (C++ with pybind11 bindings)
- Target – easy to understand versions, distribution via /cvmfs





Workflow Management and Analysis Preservation

- **Workflow management tools** are a way of helping analysis reusability and reproducibility
  - Organizing, managing, and scaling job submission
  - Publishing results into reusable formats e.g. HEPData, Rivet

- 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

- Complimentary to efforts for establishing CI pipelines for analysis
  - Best practices recommended in CAT docs
  - Ongoing work to offload CI jobs to analysis facilities

Plotting Guidelines and Tools

Established uniform style guide for the experiment

- Maintenance of packages and canned recipes for application both in python (mplhep) and ROOT (cmsstyle)
- **Plots in both python and ROOT should look the same** (also regardless of OS)

In-frame

Python ROOT

```python
hep.cms.label("Preliminary", loc=2, ax=ax)
```

Out-of-frame

Python ROOT

```python
hep.cms.label("Preliminary", loc=0, ax=ax)
```

In-frame

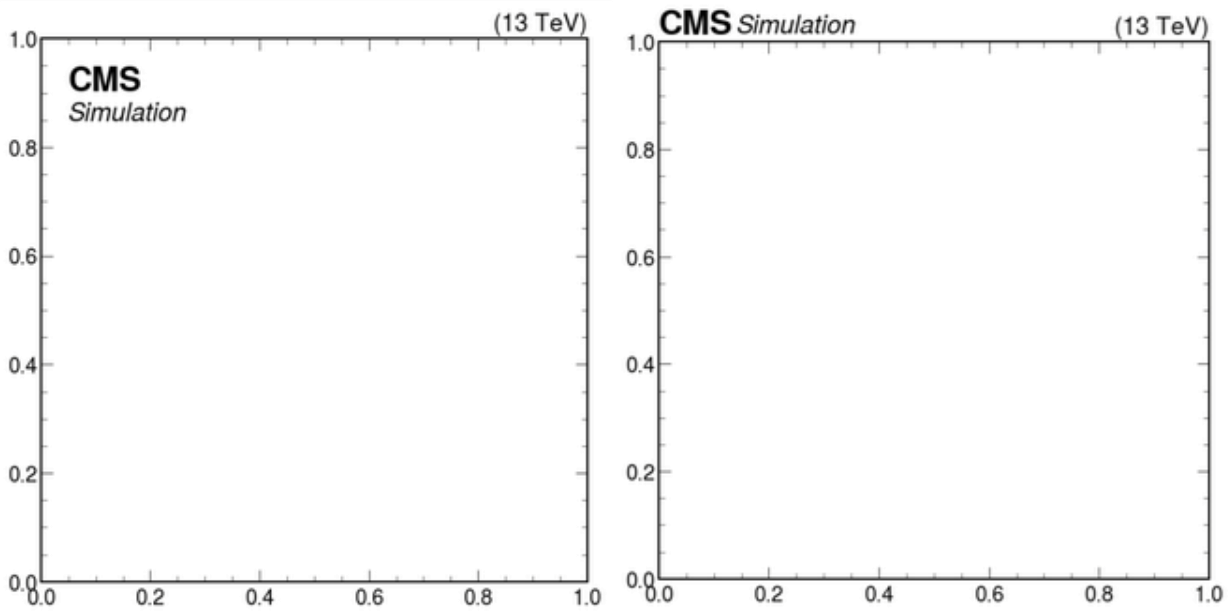
Python ROOT

```root
CMS.SetExtraText("Simulation Preliminary")
CMS.SetLumi("")
canv = CMS.cmsCanvas('', 0, 1, 0, 1, '', '', square = CMS.kSquare, extraSpace=0.01,
```

Out-of-frame

Python ROOT

```root
CMS.SetExtraText("Simulation Preliminary")
CMS.SetLumi("")
canv = CMS.cmsCanvas('', 0, 1, 0, 1, '', '', square = CMS.kSquare, extraSpace=0.01,
```



- Tex Gyre Heros - open license Helvetica clone
  - Now also available in ROOT

DP-2024-040

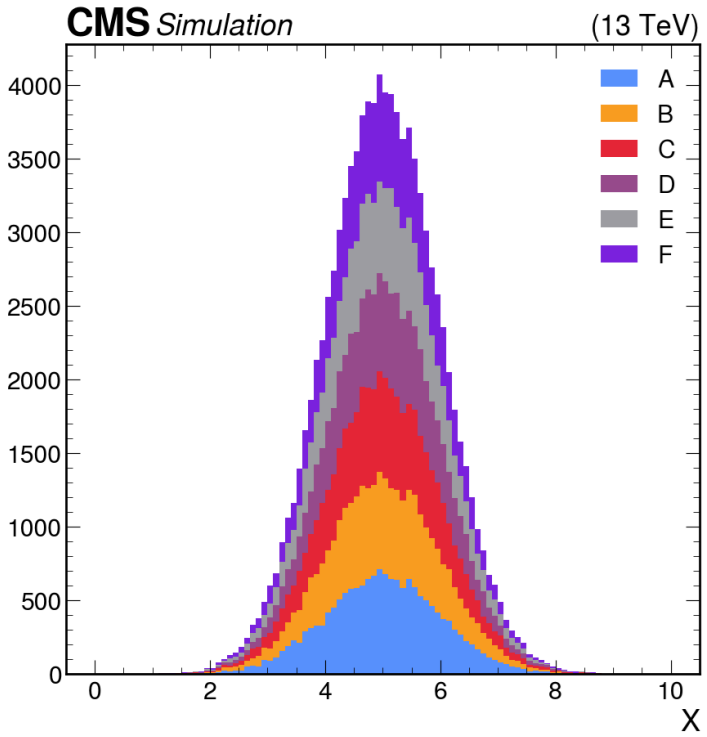
Accessible Color Schemes

Recently standardized CMS *recommended* color scheme

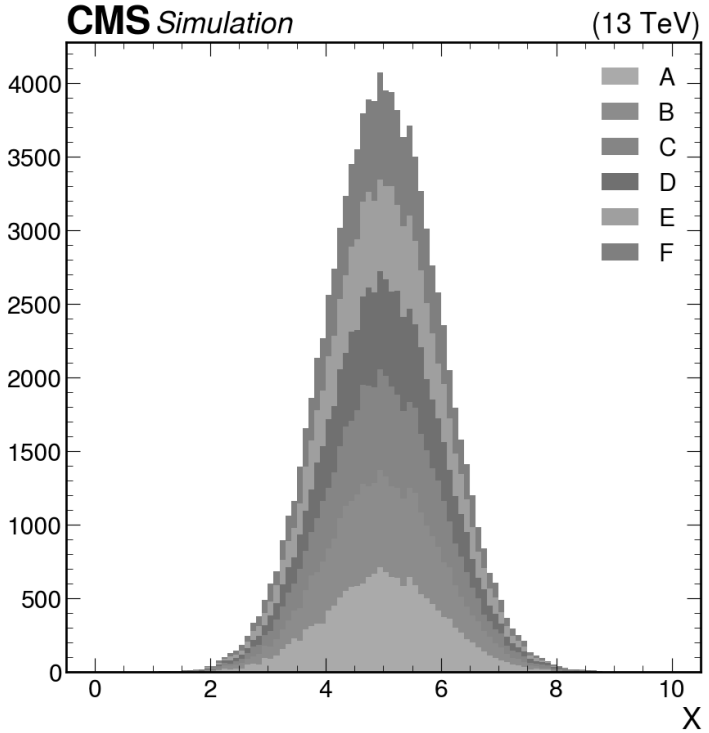
Chosen for **colorblind accessibility** (joint effort with **CMS Diversity Office**)

- From M. Petroff arxiv:2107.02270 – petroff6 or petroff10

Now also used by ATLAS



Greyscale conversion is also acceptable



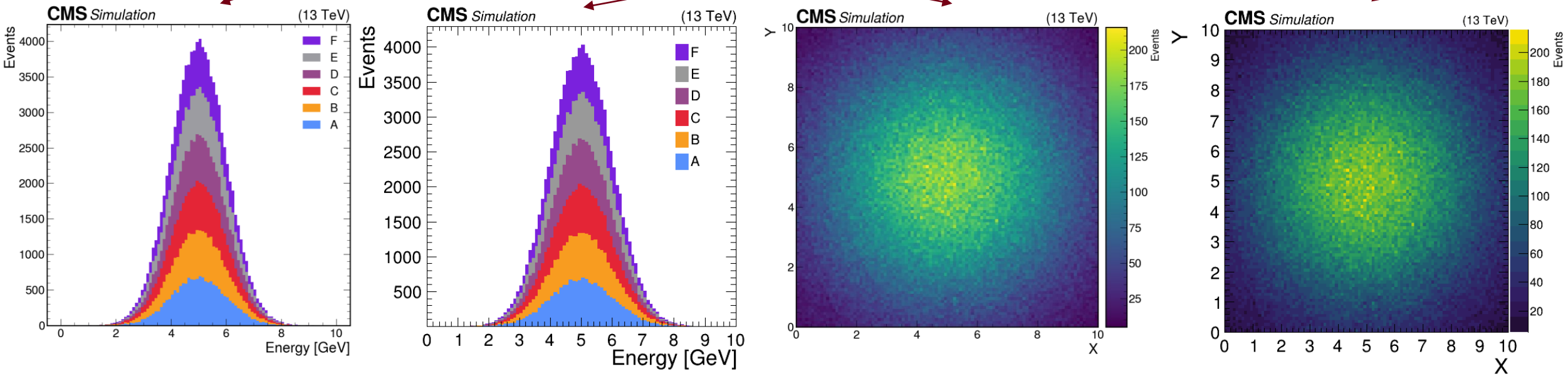
Hex Code	Color Preview
#3f90da	
#ffa90e	
#bd1f01	
#94a4a2	
#832db6	
#a96b59	
#e76300	
#b9ac70	
#717581	
#92dadd	

Hex Code	Color Preview
#5790fc	
#f89c20	
#e42536	
#964a8b	
#9c9ca1	
#7a21dd	

Accessible Color Schemes

For sequential colormaps a number of CVD-accessible options exist - using **viridis**

python **ROOT**



Consistent look regardless of choice of programming language or OS

Code snippets for users provided in docs

Python ROOT

Combine

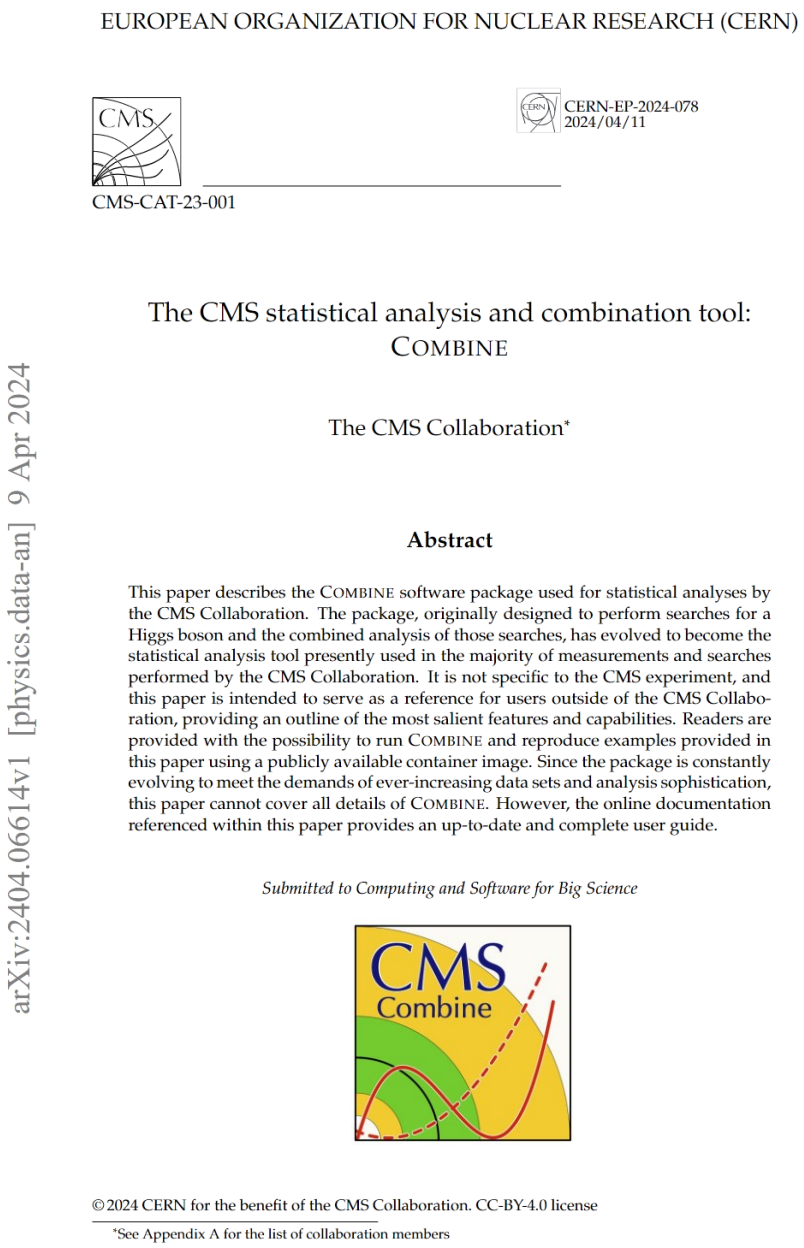
Combine is a high-level tool based on RooFit/RooStats

- Provides CLI to various statistical techniques used in CMS
- Statistical models encoded in **human-readable** *datacards*

```
imax    1 number of bins
jmax    4 number of processes minus 1
kmax    * number of nuisance parameters
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
bin          signal_region
observation  10.0
--------------------------------------------------------------------------------
bin                      signal_region   signal_region   signal_region   signal_region   signal_reg
process                  ttbar           diboson         Ztautau         jetFakes        bbHtautau
process                  1               2               3               4               0
rate                     4.43803         3.18309         3.7804          1.63396         0.711064
--------------------------------------------------------------------------------
CMS_eff_b          lnN   1.02            1.02            1.02            -               1.02
CMS_eff_t          lnN   1.12            1.12            1.12            -               1.12
CMS_eff_t_highpt   lnN   1.1             1.1             1.1             -               1.1
acceptance_Ztautau lnN   -               -               1.08            -               -
acceptance_bbH     lnN   -               -               -               -               1.05
acceptance_ttbar   lnN   1.005           -               -               -               -
norm_jetFakes      lnN   -               -               -               1.2             -
xsec_diboson       lnN   -               1.05            -               -               -
```

De facto **official CMS tool for statistical analysis**

- Support, maintenance, and development organized in CAT
- Recently submitted to CSBS



EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

CERN-EP-2024-078
2024/04/11

CMS-CAT-23-001

The CMS statistical analysis and combination tool: COMBINE

The CMS Collaboration*

Abstract

This paper describes the COMBINE software package used for statistical analyses by the CMS Collaboration. The package, originally designed to perform searches for a Higgs boson and the combined analysis of those searches, has evolved to become the statistical analysis tool presently used in the majority of measurements and searches performed by the CMS Collaboration. It is not specific to the CMS experiment, and this paper is intended to serve as a reference for users outside of the CMS Collaboration, providing an outline of the most salient features and capabilities. Readers are provided with the possibility to run COMBINE and reproduce examples provided in this paper using a publicly available container image. Since the package is constantly evolving to meet the demands of ever-increasing data sets and analysis sophistication, this paper cannot cover all details of COMBINE. However, the online documentation referenced within this paper provides an up-to-date and complete user guide.

Submitted to Computing and Software for Big Science

© 2024 CERN for the benefit of the CMS Collaboration. CC-BY-4.0 license

*See Appendix A for the list of collaboration members

arXiv:2404.06614v1 [physics.data-an] 9 Apr 2024

Likelihood Publishing and HS3

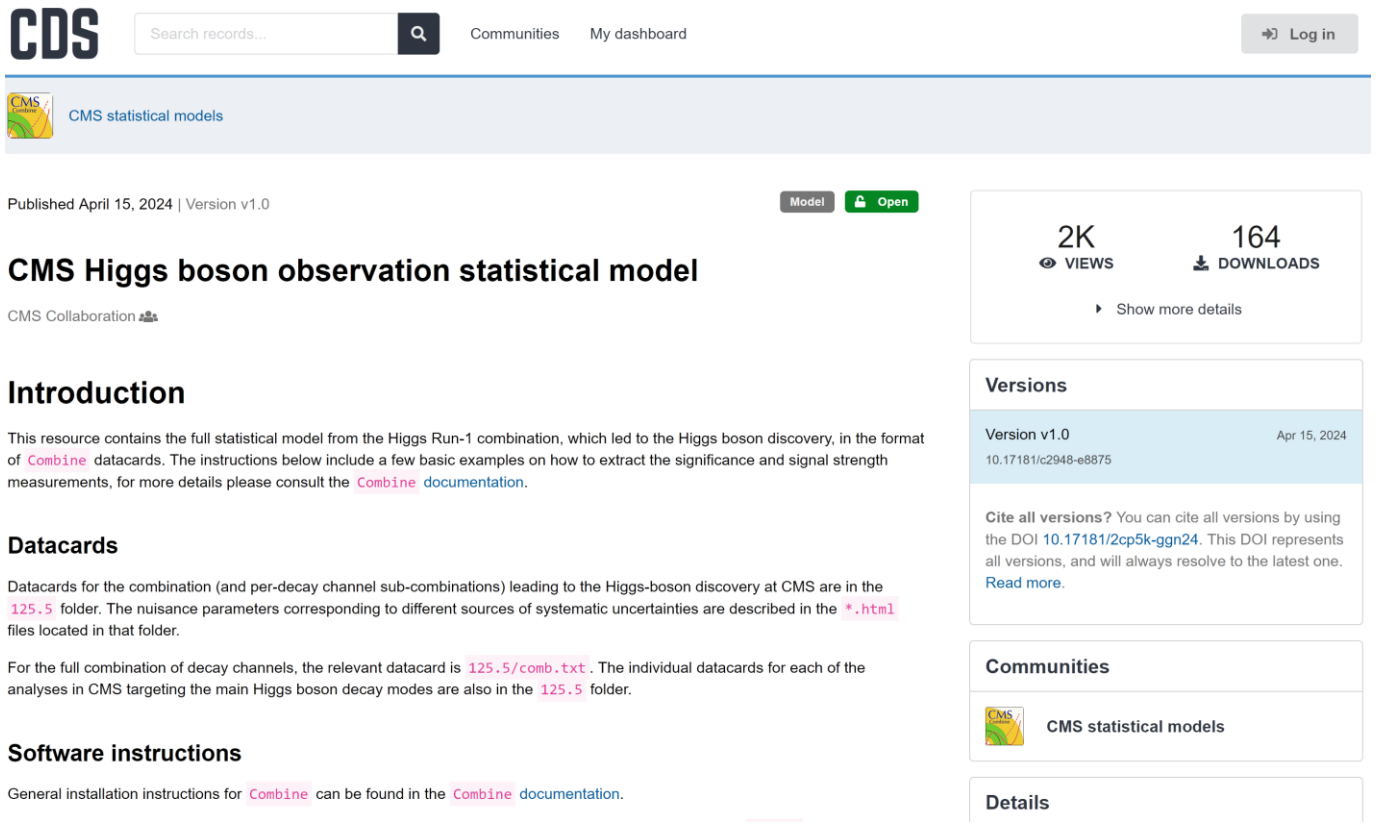
CMS will now publish full likelihood models (docker + combine + cards -> reproducible stats analysis)

- Linked to **HEPData** records and released under CC4 license
- First example already available for CMS Higgs boson observation

More details in Sezen's talk on Saturday 14:30

HEP Statistics Serialization Standard (HS3)

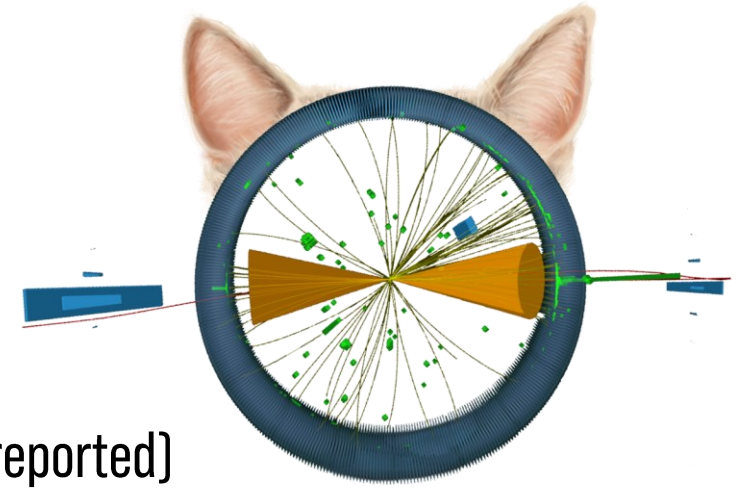
- Ongoing discussion on a new standard for likelihood publishing in HEP independent of particular software across experiments
- https://indico.cern.ch/event/1348309/



Summary of Progress and Results

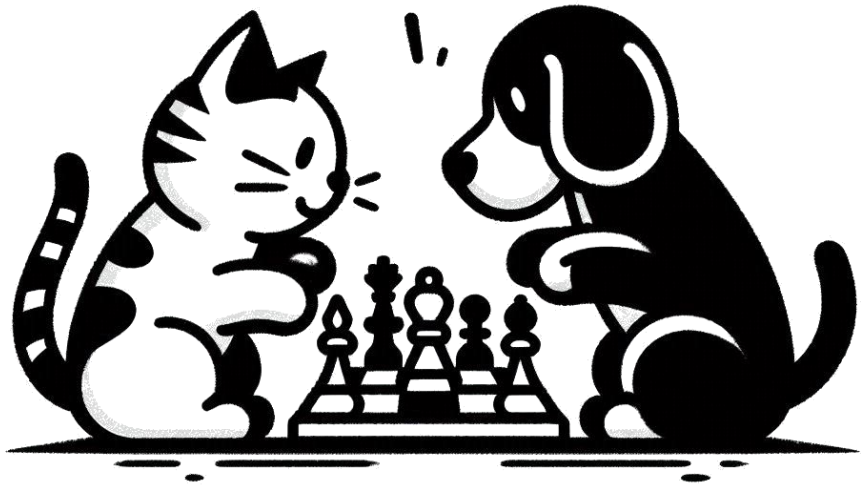
- Substantial progress since CAT group was established 2 years ago
- Established **docs** that should take a junior analyst **from NanoAOD to statistical analysis**
- Coordinate **maintenance** for key internal tools
  - Serve as a channel for feedback for outside packages such as **scikit-hep/coffea/ROOT**
- **Standardized figure style** and provided code to achieve it
- Introduced a recommended **colorblind accessible color scheme**
- Publishing a reference for **Combine** and established process to **release full likelihoods**

Ongoing Efforts and How to Engage



- Metadata scheme standardization and distribution
- Further improvements of automation
  - Ongoing efforts to include CI in analysis development (first positive experiences reported)
- Standardize a likelihood serialization format across HEP experiments
- Better integration with various analysis facilities (like coffea.casa)

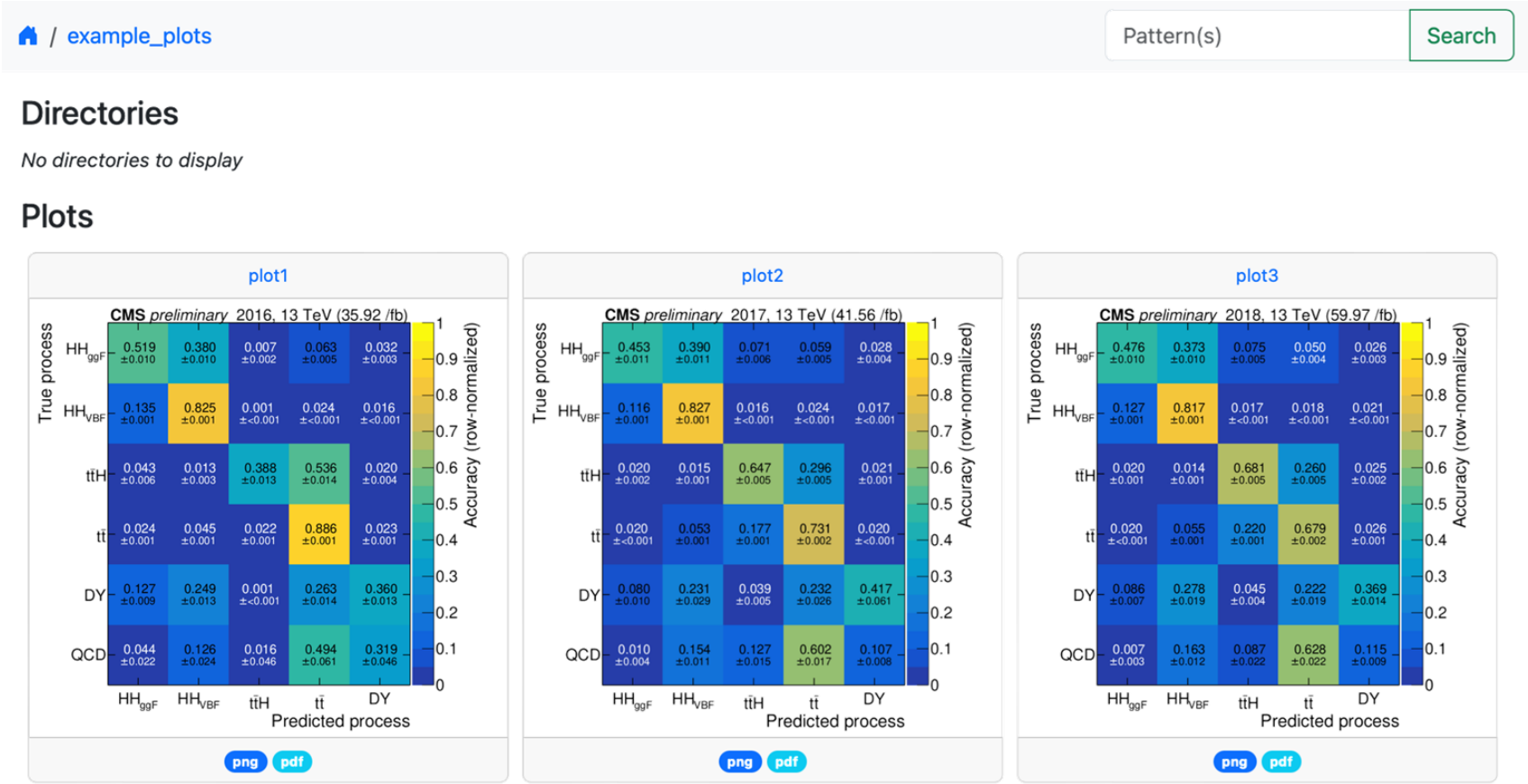
- Reach us at cms-cat-coordination@cern.ch to discuss
- HSF Data Analysis Working Group (DAWG)
  - Inter-experiment forum to discuss needs and developments



Thank You

Other Useful Tools: Updated php-plots

- Easy to set up plot browser for a personal webpages hosted on CERN eos



CAT Supported Tools

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

12 Common CMS Metadata Format: cms-order

1. Significant amount of metadata needed to perform analyses or to extract central recommendations (calibrations, SFs, ...)

2. Complex relations and dependencies render book-keeping highly non-trivial

3. Twiki-as-a-database does not work
   - Information is scattered across various pages with unclear responsibilities and practically no proper versioning
   - High degree of duplication, oftentimes without references to actual sources

4. Work to cope with this is repeated by every member or group
   - This is a very ambitious project & first versions might not be 100% complete, but we intend to release early & often
     - ▻ We see high potential in having a single, common, centralized effort with community-driven input!

- Status update given in [previous CAT meetings]

order

cms-order: Example

- Access a dataset by name

```
In [8]: c.datasets.names()
Out[8]: dict_keys(['wjets', 'ttbar'])

In [9]: %od.show c.datasets.n.ttbar
Dataset(
  id: 1
  name: 'ttbar'
  variations: {
    nominal: DatasetVariation(
      keys: [
        '/TTbb_4f_TTToSemiLeptonic_TuneCP5-Powheg-Openloops-Pythia8/RunIISummer20UL18NanoAODv9-106X_upgrade2018_realistic_v16_L1v1-v1/NANOAODSIM'
      ]
      gen_order: 'nnlo'
      n_files: lazy:das_dataset.n_files
      n_events: lazy:das_dataset.n_events
      lfns: lazy:das_lfns.lfns
    )
    scale_up: DatasetVariation(
      keys: [
        '/TTbb_4f_TTToSemiLeptonic_TuneCP5-Powheg-Openloops-Pythia8/RunIISummer20UL18NanoAODv9-106X_upgrade2018_realistic_v16_L1v1-v1/NANOAODSIM'
      ]
      gen_order: 'nnlo'
```

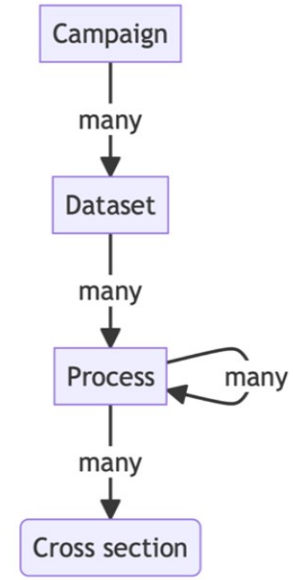
list all datasets

load and show a dataset

nominal variation

lazy fields

scape_up variation




```
n_files: AdapterModel(
  adapter: 'das_dataset'
  key: 'n_files'
  arguments: {
    keys: [
      '/TTbb_4f_TTToSemiLeptonic_TuneCP5-Powh
8_realistic_v16_L1v1-v1/NANOAODSIM'
    ]
  }
)
```

- When accessing a dataset (c.datasets.n.ttbar), its contents are loaded lazily fake values
  - In this case, the order-db is accessed and a yaml file is loaded
- The *materialized* dataset still has lazy fields, e.g. n_files, n_events, lfns
  - So called "adapters" are defined for them that control how values are obtained when fi
  - Fields can share the same adapter if their values can result from the same request
  - Already a handful adapters implemented in order: DAS dataset and LFNs, DBS, order-internal adapters