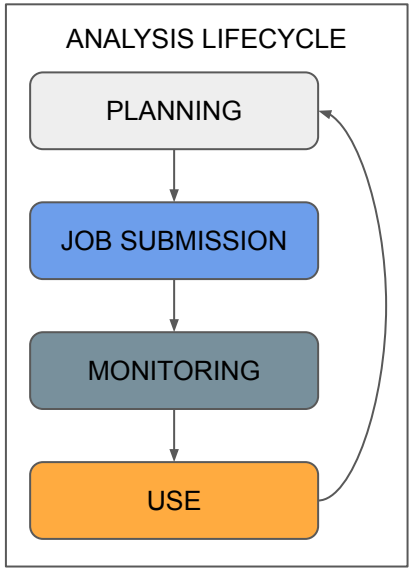
Event processing and bookkeeping in CMS

Physics Data And Monte Carlo Validation Team's (PdmV) Group Analysis Sample Page

Presented by:
Justinas Rumsevicius
on behalf of PdmV/PPD

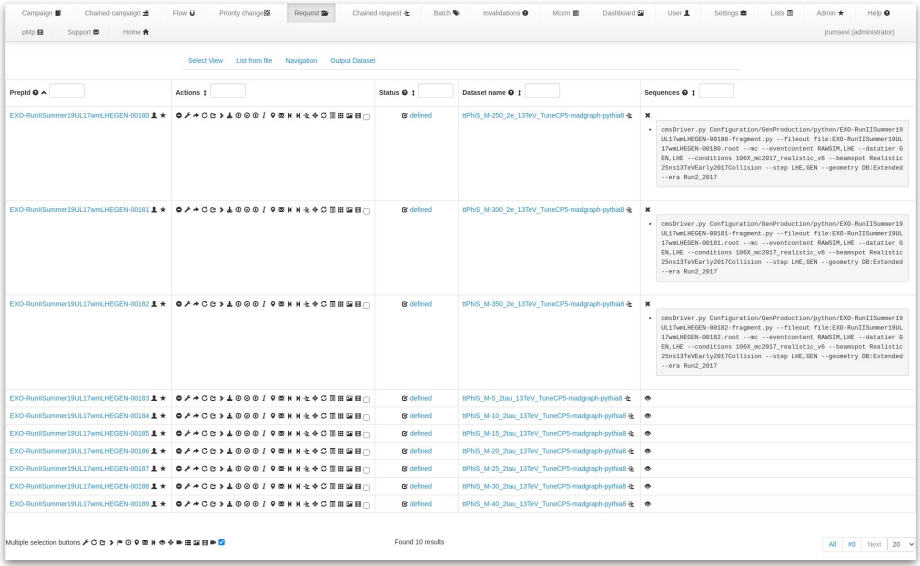
Analysis lifecycle

- Planning
  - Production managers collect requirements for analysis as well as estimate needed resources and draw a plan of upcoming campaign
- Production job submission
  - Requirements are translated into defined requests which are validated with small test runs and submitted to computing for central production
- Monitoring
  - Running jobs are observed in order to have a picture about each request's as well as whole campaign's progress
- Use
  - Produced samples are looked up and used by analyzers
- *Retrospective*
  - *Knowledge gained in each step is used in new campaign planning and hopefully leads to better estimation and more efficient use of resources*



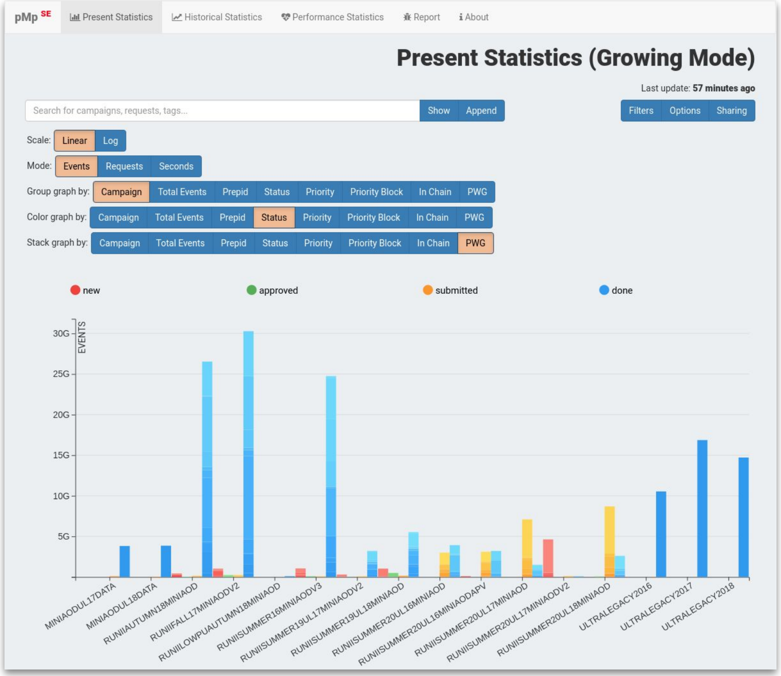
Monte Carlo Management (McM)

- Main PdmV's Monte Carlo production management tool
- It is used to create, validate (small scale test), submit, manage, reuse and bookkeep Monte Carlo requests
- Can sometimes be overwhelming
- Backend:
  - Python 2.7 Flask application
  - Newest CouchDB as database
  - couchdb-lucene wrapper for search
- Frontend:
  - Angular.js
- Around 1.07M JSONs in the database and growing every day



Production Monitoring Platform (pMp)

- Main monitoring tool in PdmV, could be called graphical representation of McM
- Powerful tool, but quite technical, allows a detailed and customizable picture of current production and it’s progress
- Backend:
    - Python 3 Flask application
    - Locally running Elasticsearch as storage and search engine
- Frontend:
    - Angular.js
    - Custom plots are drawn using d3.js library



Group Analysis Sample Page (GrASP)

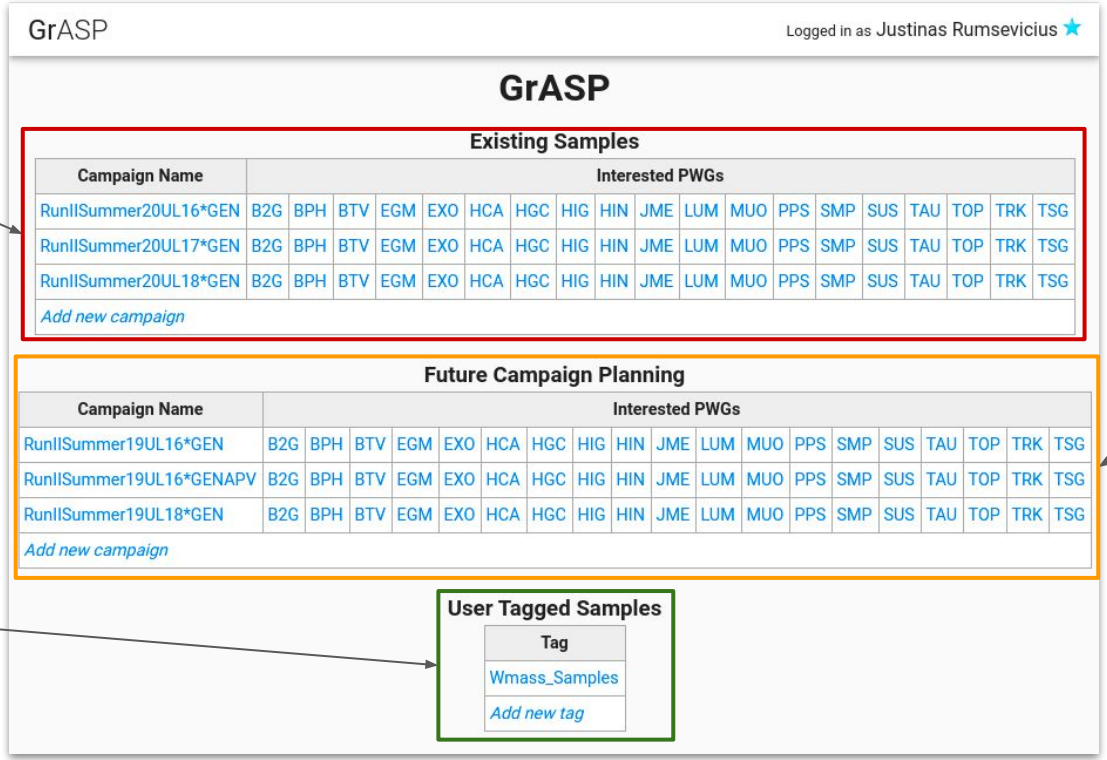
- A simplified view of ongoing Monte Carlo production
- Source of information - McM, but less technical than pMp
- It allows PWGs to mark their interest in certain datasets as well as easily track production progress
- It is aimed at analyzers and non-expert users
- Goal is to have a central and already-available tool for analysts where each PWG can search for and track samples that they are interested in and replace collection of arbitrary Google Docs
- Can be found here (behind CERN SSO): https://cms-pdmv.cern.ch/grasp

Important acronym

PWG (Physics Working Group) - group of analyzers focused on a specific topic

Main page of GrASP

Existing samples grouped by campaign and interested PWG for monitoring



Planning page for upcoming sample production per planned campaign or even interested PWG

Existing samples that are “tagged” (manually or automatically selected) by users

Existing samples page

- Samples that are either already produced, being produced or will be produced in near future
- At any point of sample lifetime any PWG can mark their interest in the sample
  - Easier tracking for the PWG - they can view only those samples that they are interested in
  - Enables PdmV to know which PWGs are interested in which samples and plan accordingly
- Table that shows main steps of sample production from event generator to analysis ready MiniAOD or NanoAOD datatiers
  - The usual computing model in CMS is GEN⇒SIM⇒DIGI⇒RECO⇒AOD⇒MiniAOD⇒NanoAOD
- Links to external services - McM, pMp or Data Aggregation Service (DAS) to get more details about steps, datasets or track progress there

Existing samples page

Share or bookmark currently visible table including search terms

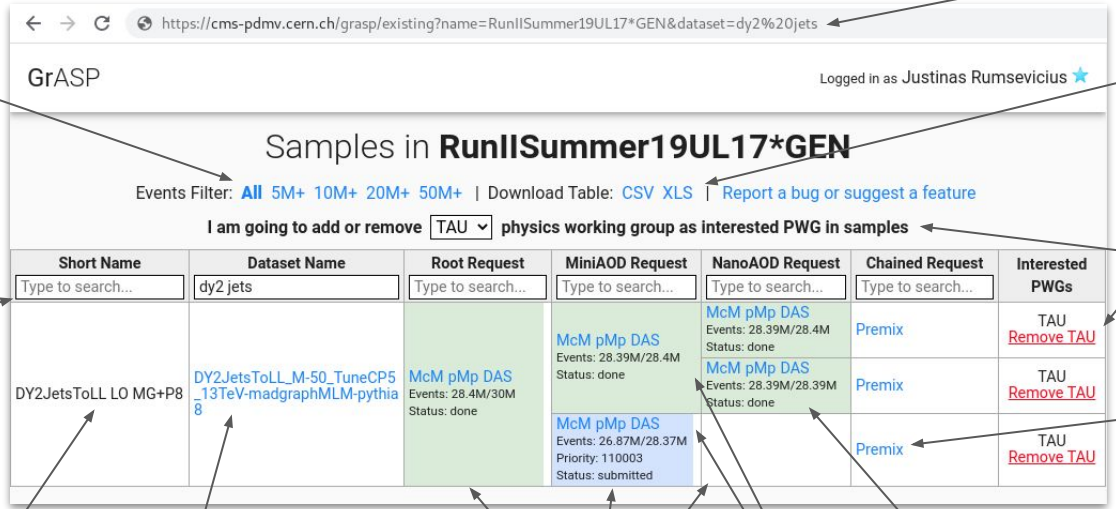
Filter table entries by number of events in GEN request

Generate and download CSV or XLS file from currently visible table

Search for samples by filtering table entries (case insensitive regex)

One click addition/removal of interested PWG

Type of digitization in sample processing chain



Automatically generated short version of dataset name - physics process and generator

Full dataset name

Three steps of sample production - GEN, MiniAOD and NanoAOD

Multiple versions of MiniAOD for the same GEN request

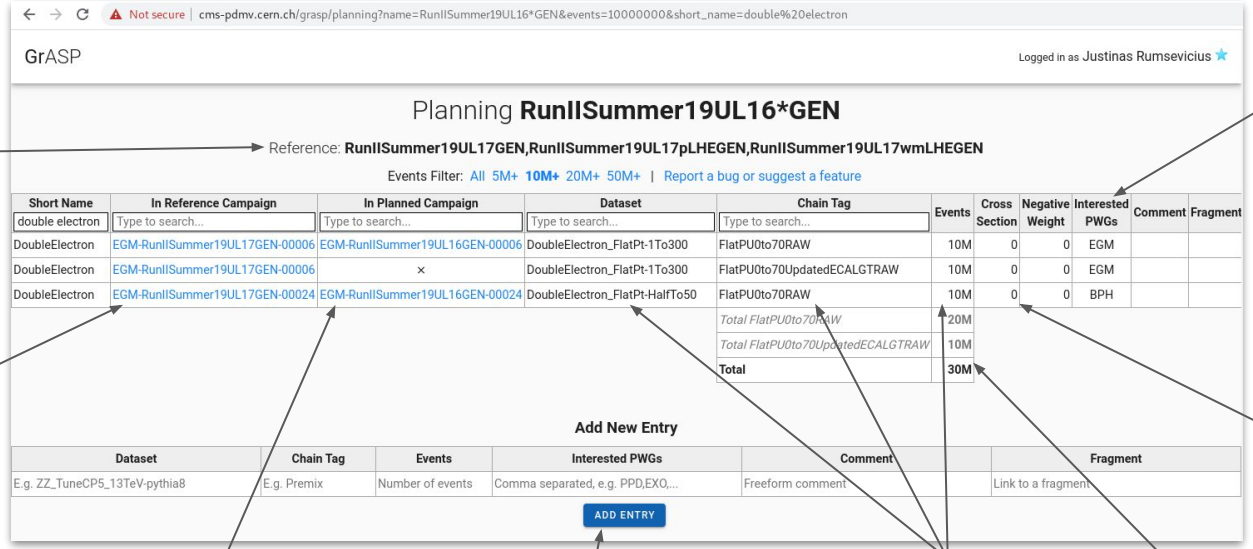
Multiple versions of NanoAOD for the same MiniAOD request

Future campaign planning page

- Replaces currently used manually updated Google Spreadsheet
- Used to plan total number of events “budget”
- Automatically finds and fills IDs of requests in McM if they exist
- If new campaign is created based on some existing campaign, it can be prefilled with entries of existing campaign

Future campaign planning page

New campaigns can be planned using already existing campaigns as a reference



PWGs can express their interest in certain dataset even when campaign is still being planned

If request is present in the reference campaign that planned campaign is based on, it's ID is automatically shown here

If planned campaign exists in McM and request is already materialized in it, it's ID is automatically shown here

Manually add new entries

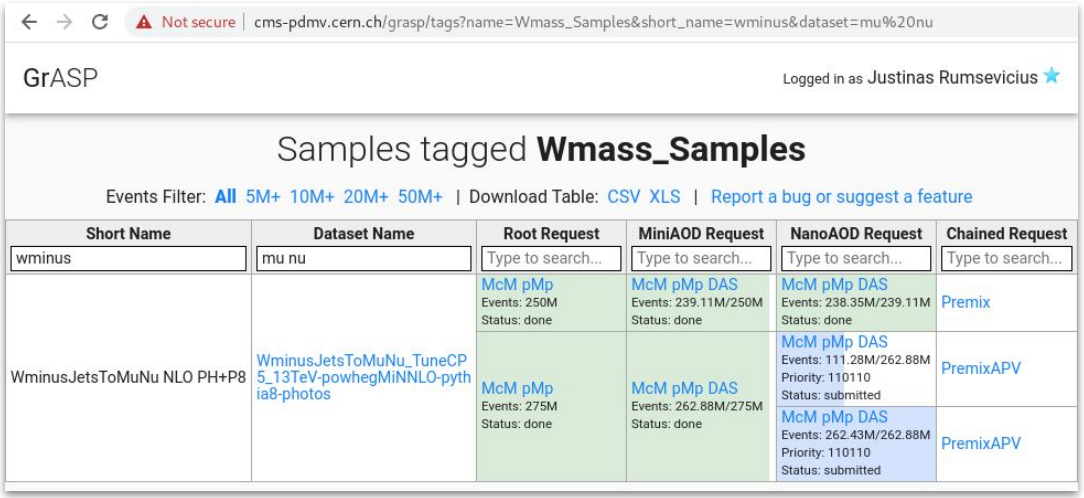
Values are editable in a Google Spreadsheet style - double click a cell, edit it and click outside it to save

Fetched from Cross Section Database (if available)

Sum of events of currently shown table

User-tagged samples page

- Same type of page as “Existing samples”, but users can choose and “tag” samples that they want to see in the table
- Finer selection that is completely in user’s hands and leads to any desirable subset of samples for monitoring or sharing
- Samples can be tagged either manually or using a script in McM



Technical side of GrASP

- GrASP is split into three parts - web frontend, web backend and update scripts (synchronization with McM)
- Frontend is Vue.js based application that fetches data from GrASP API and renders the page on the client side. Search is also performed on client side
- Backend is a Python 3 Flask application with a local SQLite database
- Update scripts are several Python 3 scripts periodically run by Jenkins that pull data from McM and store it in local database as well as push PWG interest in samples (updates to Interested PWGs column) back to McM
- Two instances - production and development (“testing”)
- GitHub repo: https://github.com/cms-PdmV/grasp

GrASP API

- If user wants to fetch data or perform actions in GrASP not in the usual web browser way, but, for example using a script, they can use the GrASP API
- It is also used by Vue.js frontend, so all features and possibilities are present
- Documentation is automatically generated from comments (python docstrings)
- Requires CERN SSO cookie, just like the page itself
- API documentation can be found here https://cms-pdmv.cern.ch/grasp/api

EXISTING

- Endpoint for creating a new existing campaign
  /api/existing/create
  - **PUT** (Requires "production_manager" role in McM): Create an empty existing campaign with the provided JSON content
- Endpoint for getting an existing campaign or campaign group
  /api/existing/get/<string:campaign_name>
  /api/existing/get/<string:campaign_name>/<string:interested_pwg>
  - **GET**: Get a single existing campaign with all entries inside

The end

PdmV homepage: http://cms-pdmv.cern.ch/