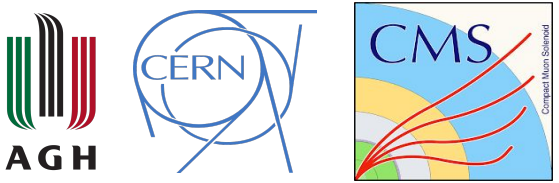
11.01.2024

Environment for particle accelerator conditions data used by the CMS experiment at CERN

Jan Chyczyński
on behalf of CMS

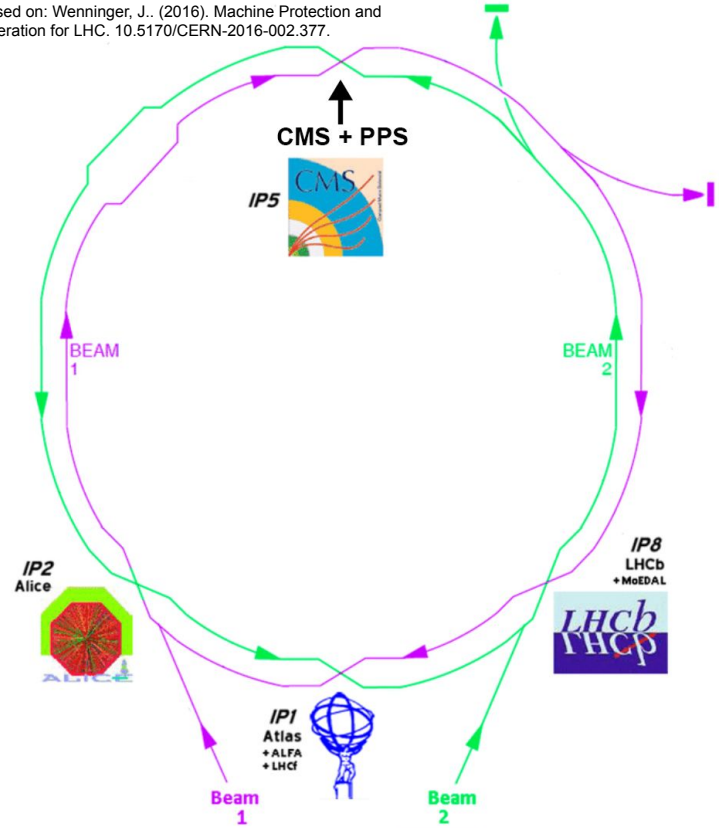


CERN, CMS, PPS

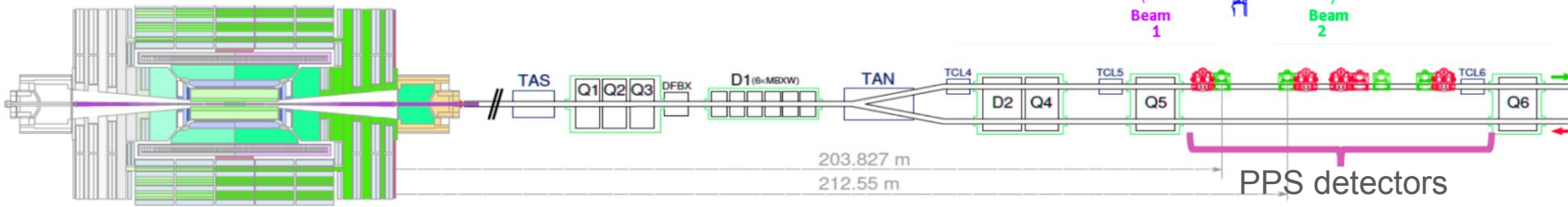
The Precision Proton Spectrometer (PPS) is a part of the CMS experiment at CERN, which is used to conduct research in **forward physics** at the Large Hadron Collider (LHC).

PPS is used to study **protons that survived a collision** in the CMS Interaction Point and exited the CMS cavern with deflected trajectory

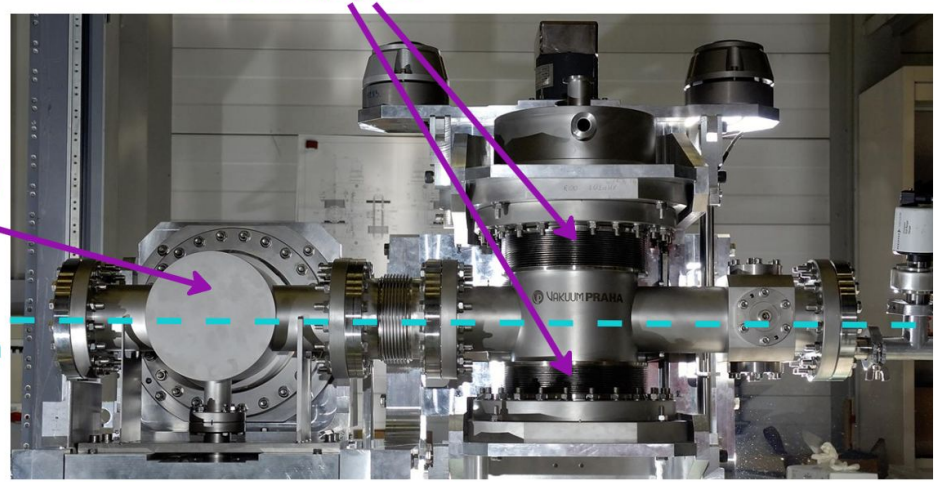
Based on: Wenninger, J.. (2016). Machine Protection and Operation for LHC. 10.5170/CERN-2016-002.377.



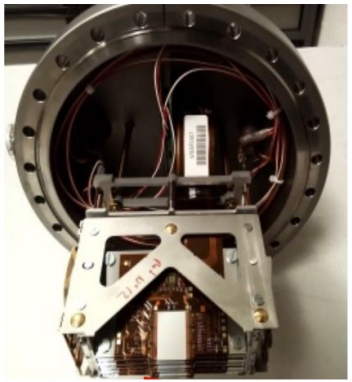
CMS central detector



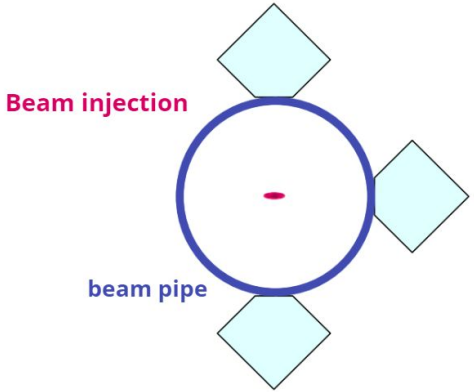
PPS/TOTEM detectors



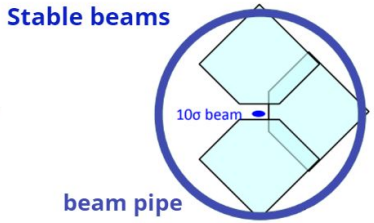
Detector station with 3 Roman Pots: movable detector vessels



3D Silicon Pixel Detectors



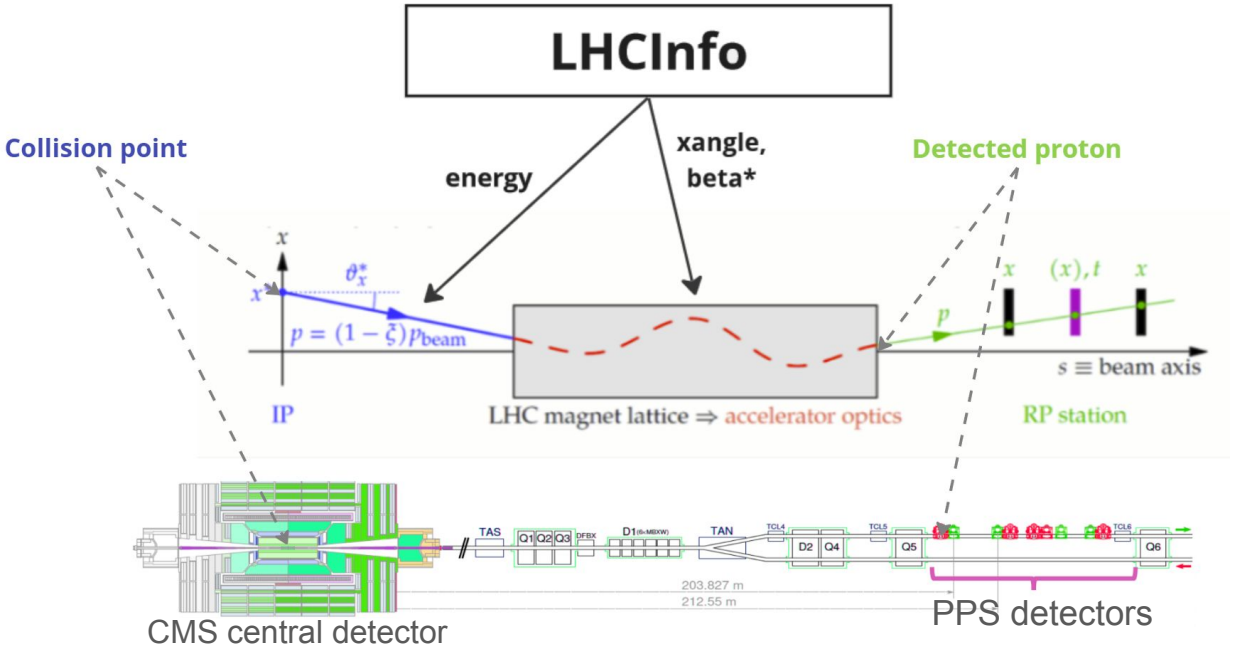
Sensors in "garage" position



Sensors in data-taking position

PPS and LHCInfo

To understand the collision and calculate the kinematics of the colliding proton we need to **reconstruct its trajectory** from the collision point to the PPS detector. For this reconstruction we need to know e.g. the angle of the incoming beams, the energy and shape of the beams - these **conditions data** are stored in the **LHCInfo** class of CMS Software (CMSSW) - the CMS reconstruction framework



$\vec{d}(s) = T(s,\xi) \cdot \vec{d}^*$ $\quad \vec{d} = (x, \theta_x, y, \theta_y, \xi)^T$

T= transport matrix

Includes optical functions O = L,v,D
O = O (β*, crossing angle, Energy)

Main idea of the project

- PROBLEM: original LHCInfo stores lots of redundant data because it contains parameters with different **granularity** and has to be stored for every lumisection:
  - some are changing once or a few times per **fill*** (eg. bunchesInBeam, fillType)
  - some are changing every **Lumisection** (LS, ~23 sec) (eg. betaStar, crossingAngle)

- SOLUTION: Add 2 new classes for these 2 granularities:
  - **LHCInfoPerFill**
  - **LHCInfoPerLS**

- Moreover: LHCInfo contains only one value of beta* and crossing angle but in the future LHC plans to use “flat optics” which require β*x != β*y and both crossing angles (x and y)

* Fill - period of around 2-24 of hours, which corresponds to injecting, accelerating, and maintaining protons (or ions) in stable beams for the collisions.

Main steps of the update

1. **Define the new classes**

   LHCInfoPerLS and LHCInfoPerFill

2. Data aggregation

program managing sources of the data, creating objects of the classes and writing them to the offline database

3. Update usage of LHCInfo
   - number of PPS modules in CMS software (CMSSW) depends on LHCInfo: optics, reconstruction
   - maintaining backward compatibility so both old or new formats can be used

Step 1: Define the new classes

OLD

LHCInfo
fillNumber
collidingBunches
energy
…
lumiSection
crossingAngle
betaStar
delivLumi
…
setLumiPerBX
injectionScheme
ctppsStatus
lhcState

stored for every **lumisection**

NEW

LHCInfoPerFill
fillNumber
collidingBunches
energy
…
delivLumi
…
setLumiPerBX
injectionScheme
ctppsStatus
lhcState

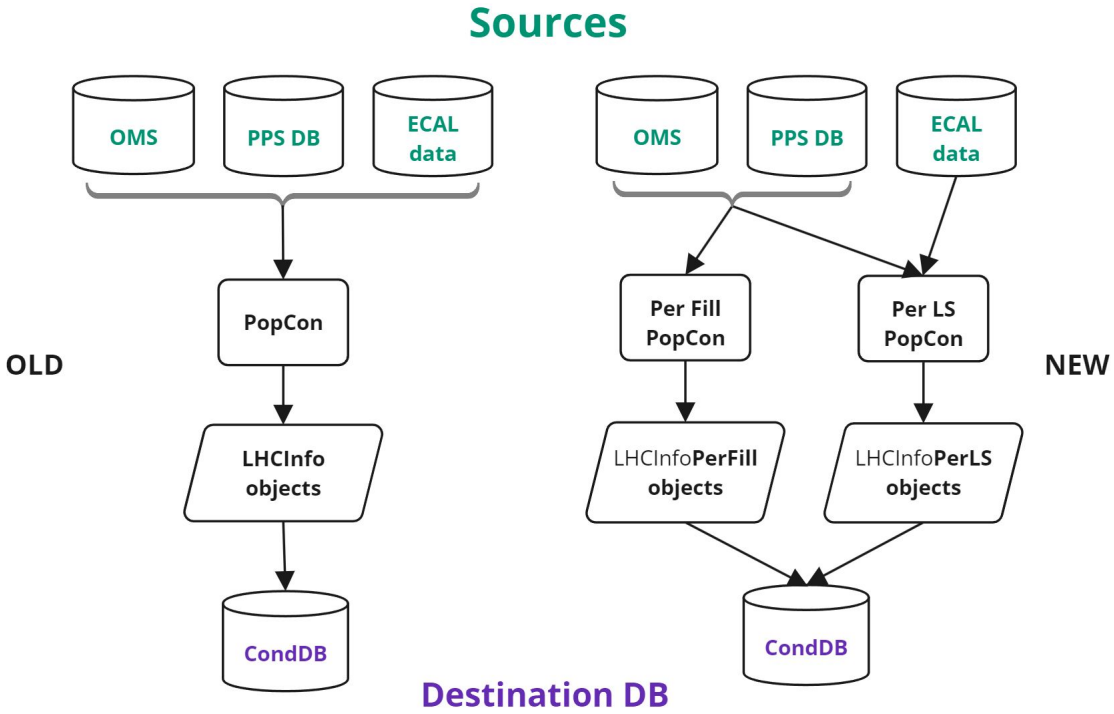
stored once or twice per **Fill**

LHCInfoPerLS
fillNumber
runNumber
lumiSection
crossingAngleX
crossingAngleY
betaStarX
betaStarY

stored for every **lumisection**

Step 2: **Data aggregation**

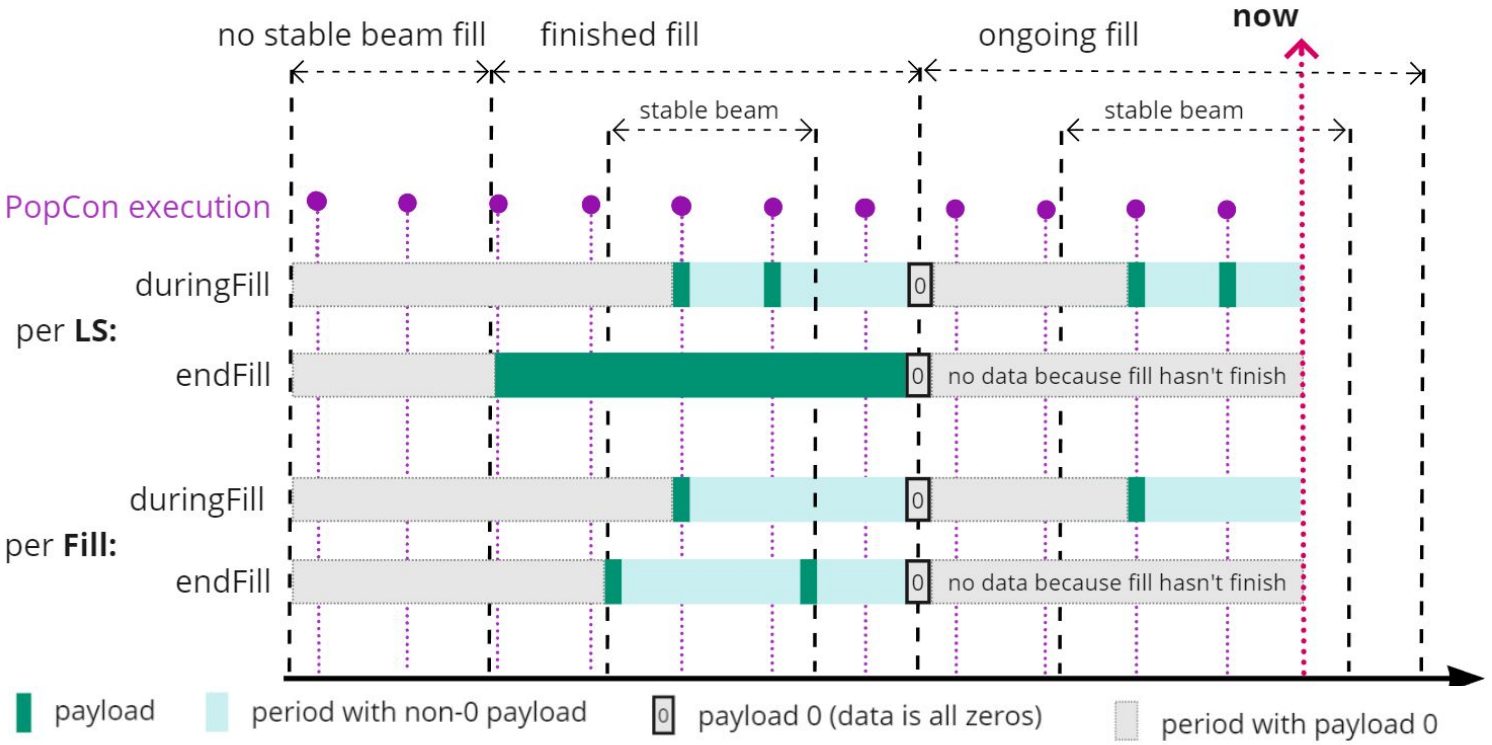
Populator of Conditions (PopCon) manages the data sources, creates condition objects and populates the destination database. It's executed every hour in a cron (time-based scheduler) job



DuringFill and EndFill modes of PopCon

Object generated in the 2 modes are stored separately and are aggregated in a different way

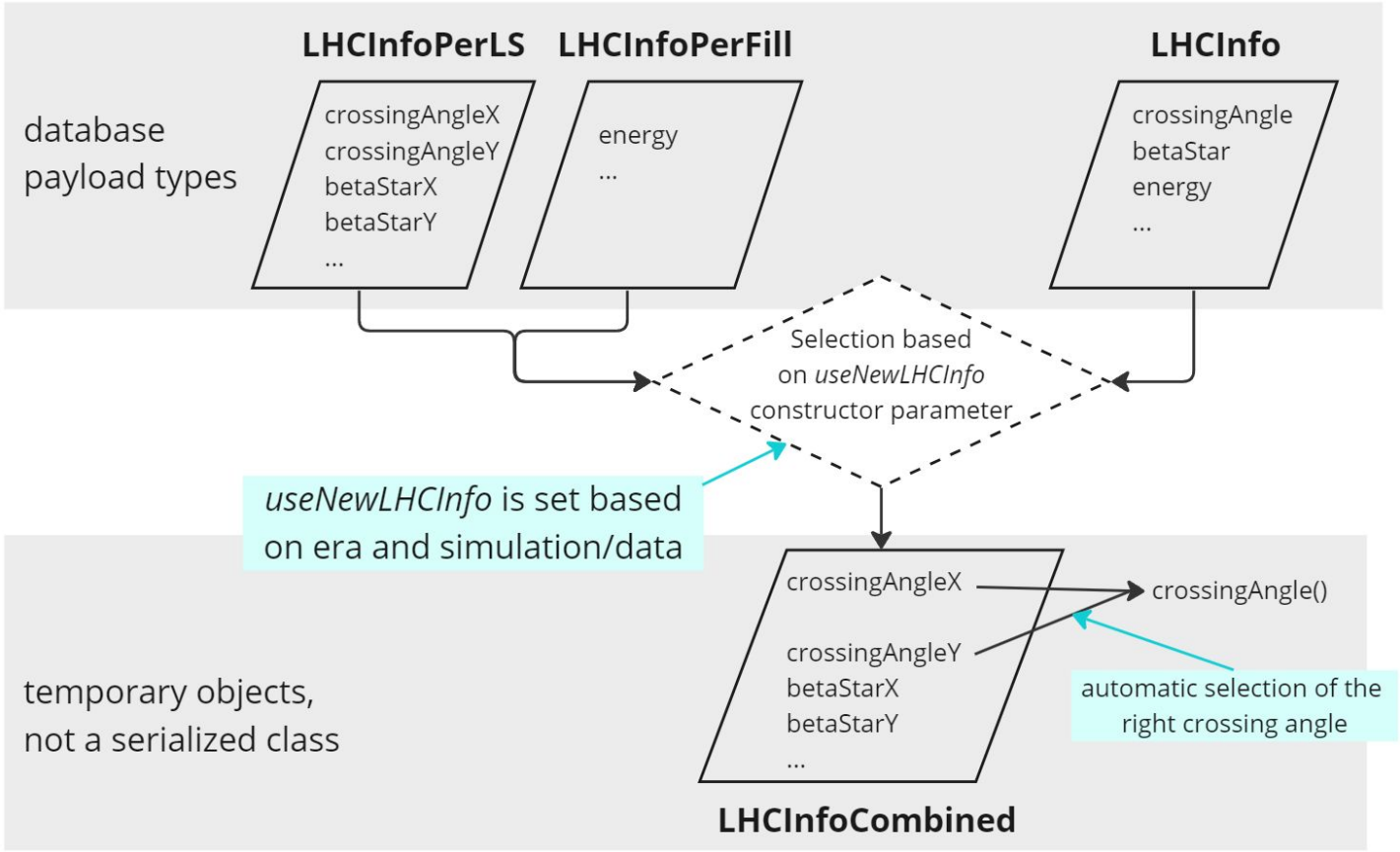
- DuringFill: every hour writes only one payload with the newest available data
- EndFill: processes a fill only once, at the end, writing all its payloads to the tag (data sequence)



Step 3: **Update usage of LHCInfo**

- Old data are stored with old LHCInfo format
- Data after 2022 are using LHCInfoPerLS and LHCInfoPerFill
- Simulation workflows, regardless of the time period, should use old LHCInfo
- How to provide a simple mechanism for updating every old LHCInfo usage to select between new and old classes?

LHCInfoCombined



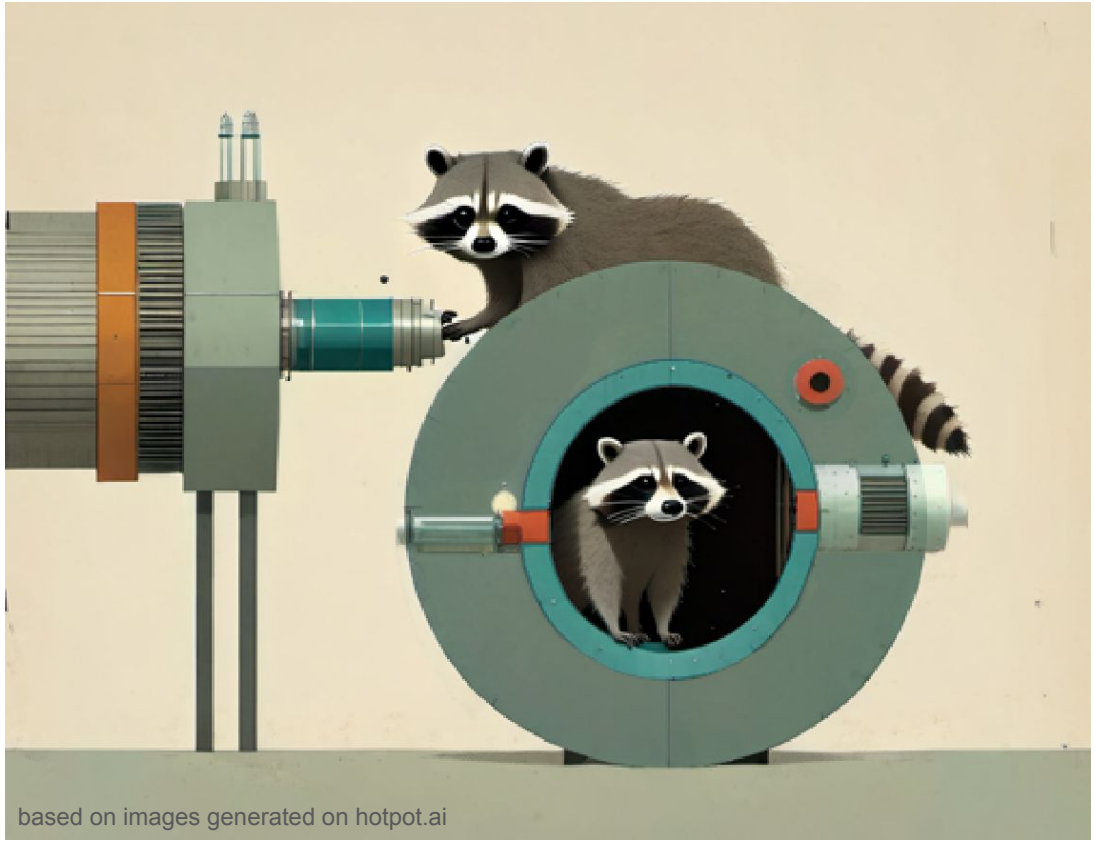
Summary

The task of updating the environment for storing LHC parameters to optimize database load has been finished.

New records and its PopCons were implemented and validated. It’s ready for deployment. LHCInfoPer* payloads for Run3 have been generated and validated.

All modules using LHCInfo were updated to be able to use new LHCInfoPer* data types.

Thanks for your attention!



based on images generated on hotpot.ai