Analysis Tools

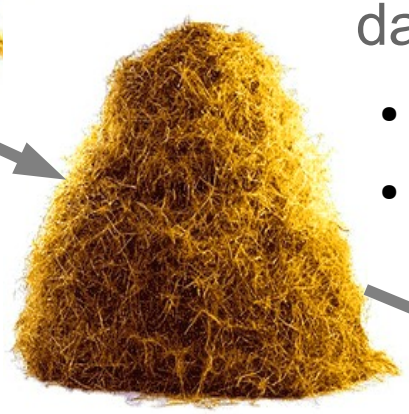
Petar Maksimovic (JHU)

- How the information flows through an analysis
  - skimming, slimming, thinning
- Our common language: Physics Analysis Toolkit (PAT)
- “Interactive” use (*vs*. “batch” -- cmsRun)
- Options for “Interactive” use
  - Bare ROOT, FW Lite in ROOT[], FW Lite standalone, Full FW
- Other stuff to be mindful of (RooFit, RooStats, etc.)

Data flow from detector to analysis

Primary Dataset
(on trigger bits)

- We will be flooded with data – access to data is a **big** issue
- Must optimize: filter events into smaller and smaller data samples, so that you only need to run over a small portion of data.
  - S/B improves with tighter cuts
  - This is called *skimming*



Secondary Dataset

Group Skim
(e.g. Higgs PAG)

Final Sample
(interactive analysis)

(Also need to drop unneeded parts of each event – more later.)

More on the Art of Skimming

- All the skimming is done by cmsRun jobs (i.e. Full Framework) running on various computer clusters
- Each skimming step needs to reduce sample size by one or more orders of magnitude to make it feasible
- User skims & final sample production: your chance to apply `almost final' analysis cuts:
  - cuts are tight enough to make the final sample small (otherwise the interactive use is slow!)
  - at the same time, cuts must be sufficiently loose to allow for enough wiggle room, further cut studies, etc.

Reduction in event size: RECO, AOD

- Important: remove unneeded information *from each event*
- EDM stores event data as collections of objects (Data Formats)
  - collections of unneeded objects can be `dropped'

    ==> could make output sample significantly smaller

    (so we can trade looser cuts vs tigher content...)

    (or trade both for faster “interactive” use)
  - this is called *slimming*
  - it's tricky – hard to know up front what will not be needed later
- RECO: collections which are output in reconstruction
- AOD: data needed for doing an analysis – a subset of RECO
- RECO and AOD are *Data Tiers*

  ==> their content is fixed and uniform for all CMS

Next step in data reduction: PAT

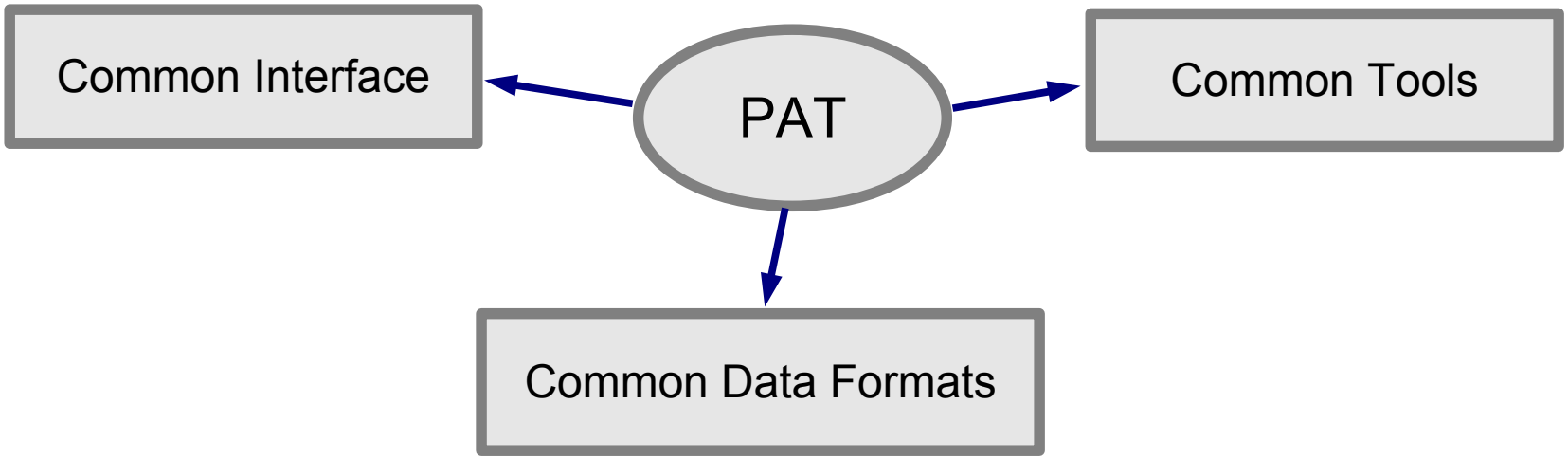
- RECO and AOD contain all the needed collections
- EDM stores these collections efficiently
  - objects in various collections are inter-linked, saves disk space, optimizes I/O, etc...
  - however, this makes it harder for humans to access it (requires non-trivial C++ gymnastics)
- What most users need is something which is **simple** to access:
  - objects which *aggregate* all the related information = PAT objects

    (PAT electron, PAT jet, etc.)
- User also needs to control what goes into these objects
  - your PAT jet may be larger (contain more info) than mine!

==> *PAT is not a data tier!* -- it's content is fluid

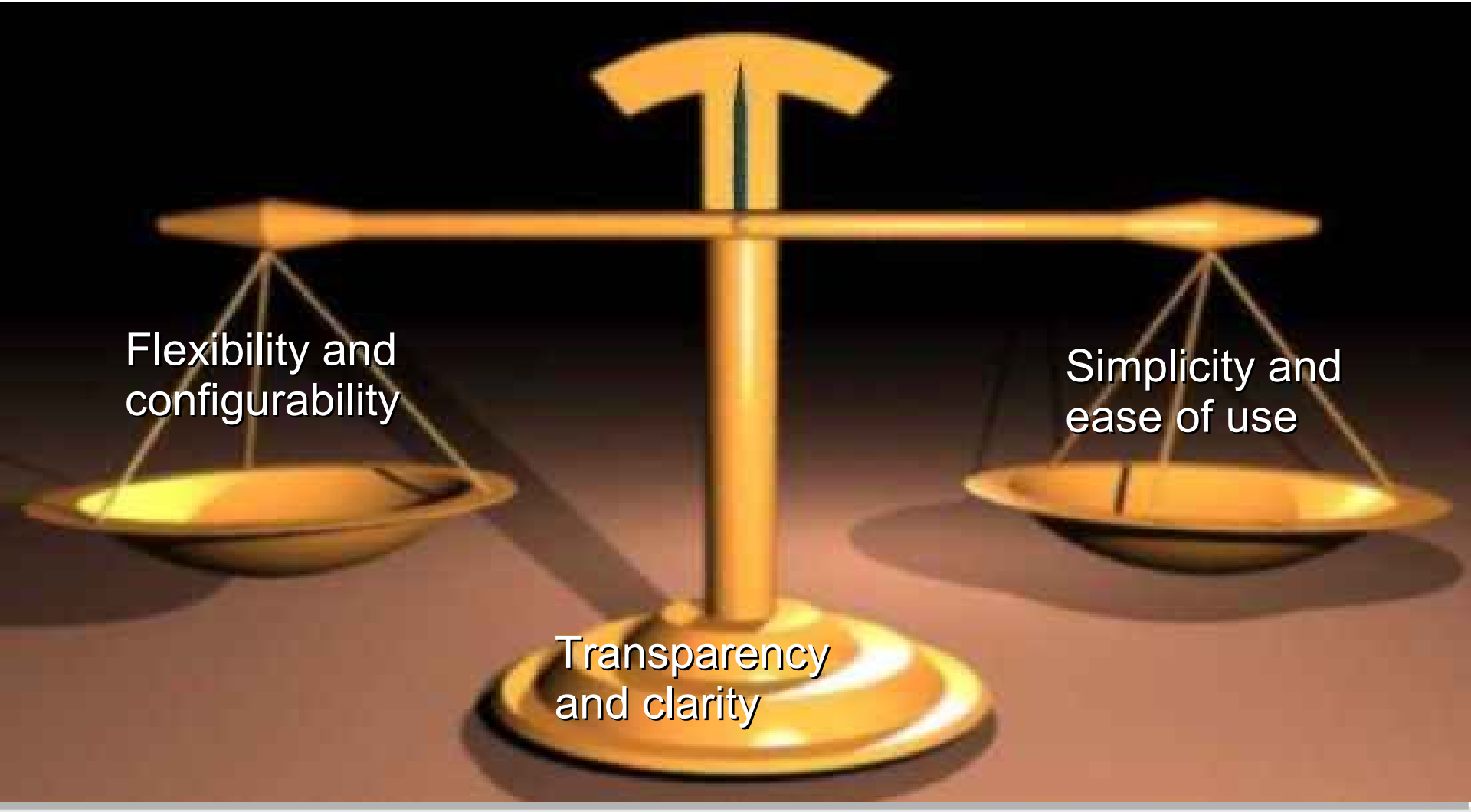
PAT = Physics Analysis *Tookit*

- provides ways of making and storing information (algorithms and data members in PAT objects)
  - algorithms provided and vetted by POG and other experts
- make use of modular structure of CMS Framework
- provide easy access via member functions in DataFormats
- serve ~80% analyses in CMS (likely more, has enough flexibility)

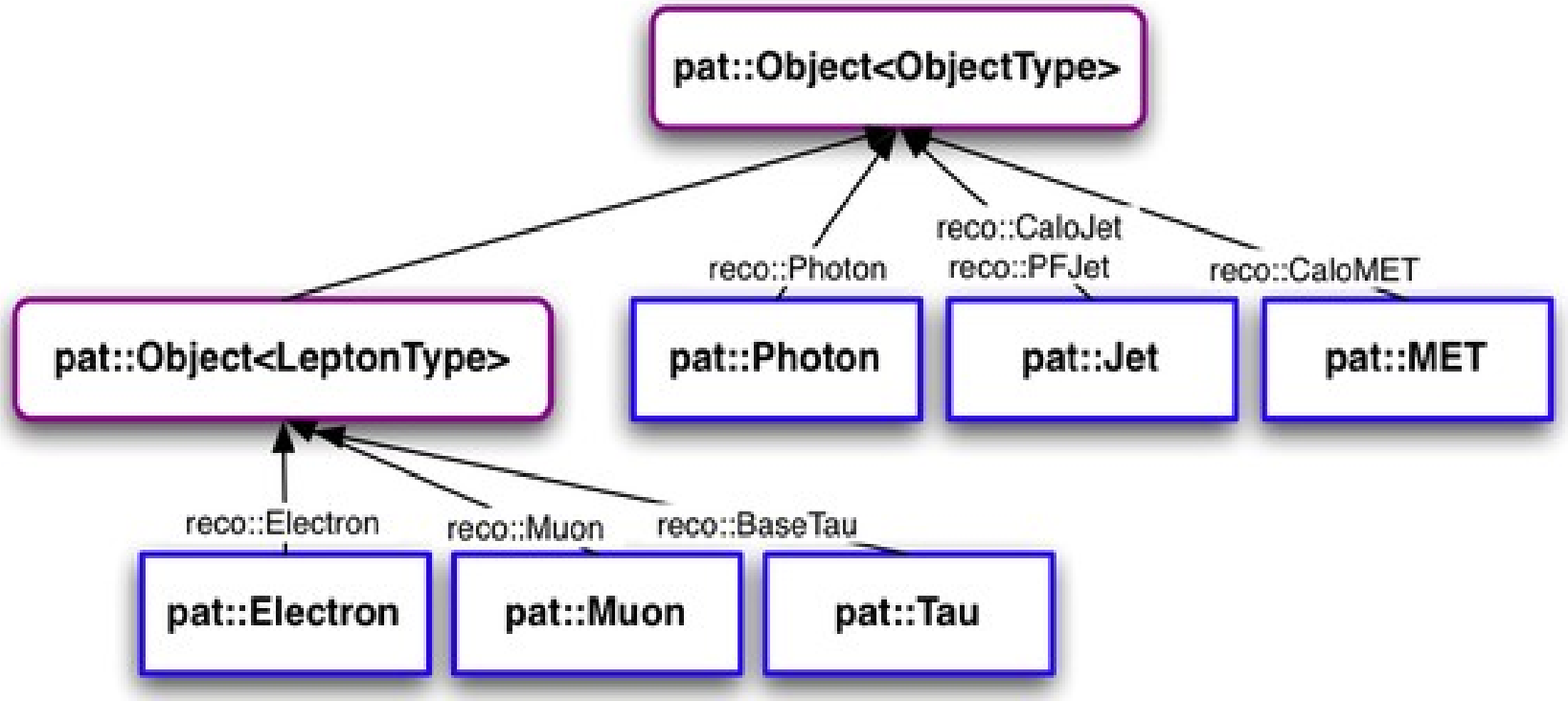


PAT Objectives

- PAT must balance competing needs



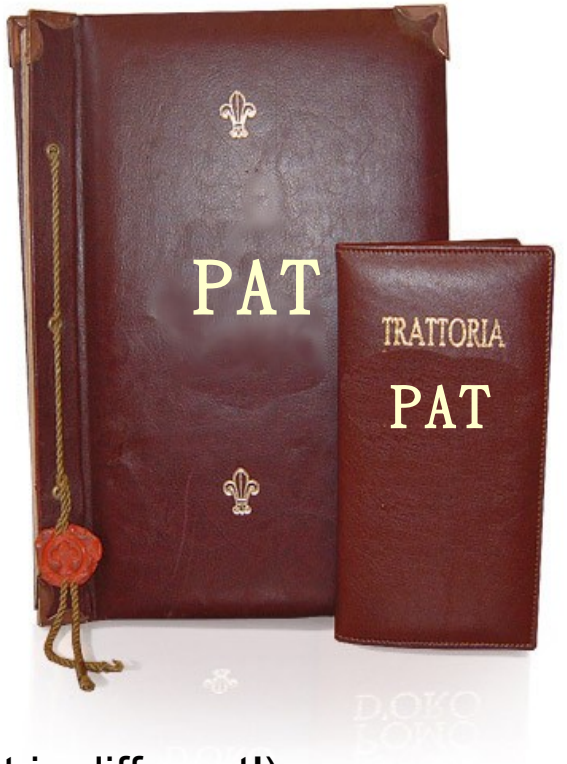
PAT Object = reco::Candidate + extra info



- PAT Objects inherit from their corresponding reco::Candidates
- Everything in DataFormats/PatCandidates

PAT – our common language

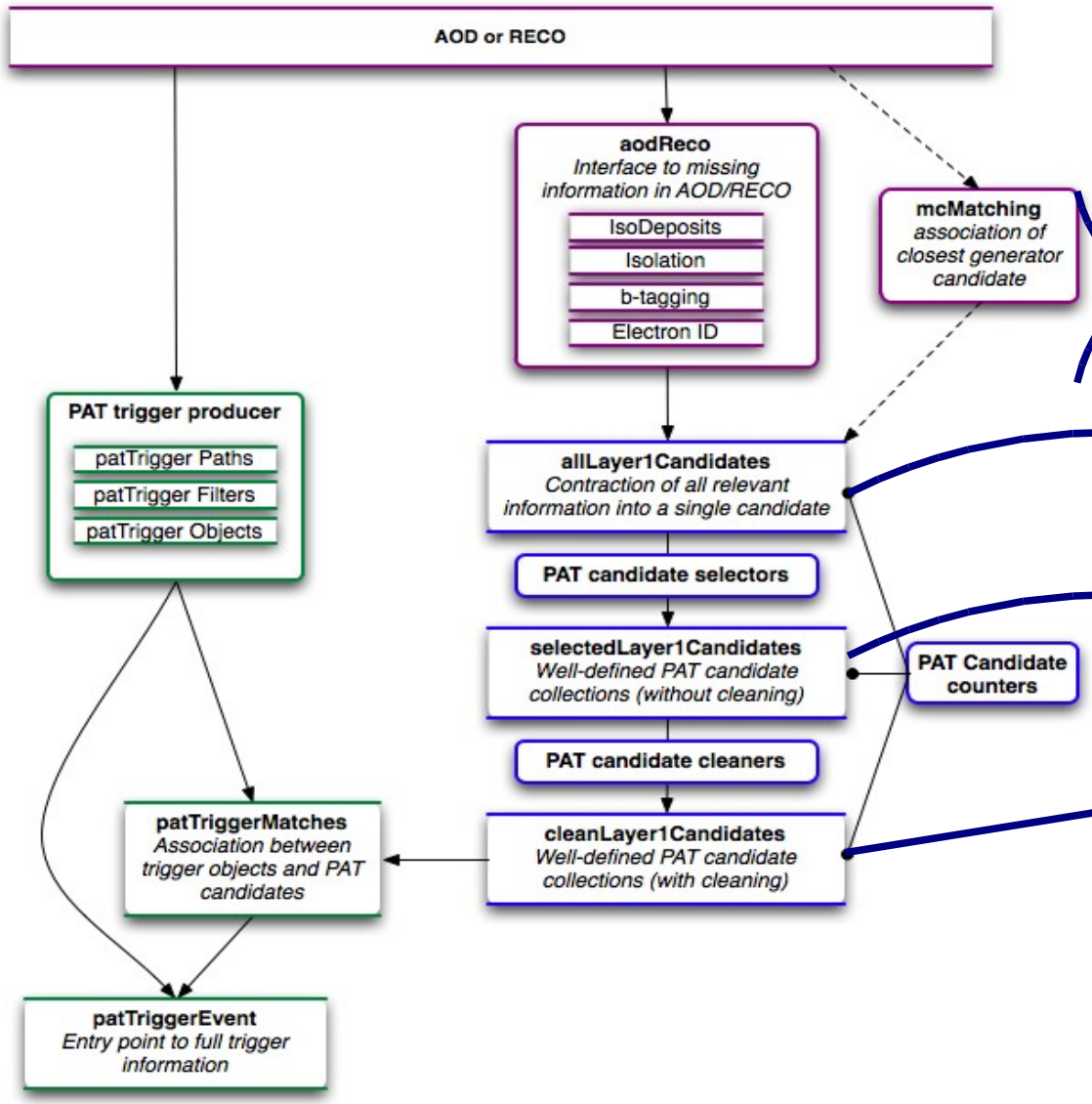
- PAT is *a toolkit!*
  - you can choose **which piece** of each PAT object you'd like to keep
  - each piece is made according to a standard recipe (provided by experts)



- The "Chez PAT" metaphor – PAT is like a restaurant menu:
  - you can order what you want: choose to add or not add pieces of information to your PAT objects
  - but, *we all are ordering from the same menu!* -- the asparagus on your plate is made in the same way as on mine

    (and if not -- I have a way of knowing exactly what is different!)
  - the chef is an expert

PAT Workflow



The 2nd iteration! We have already simplified it once. (PAT "V2.0")

Pre-production steps

Basic input collection

Output collection (w/o cleaning)

Output collection (with cleaning)

NB: some steps can be skipped, it's your call!

Benefits from PAT

Tecnological:

- you need to write less C++ code
  - smaller chance of mistakes (procedure) & bugs (coding)
- code is provided / vetted / maintained by experts
- common validation (by POGs and AT group)
- common tools to validate yourself

Sociological / Organizational:

- shared development
- sensible defaults
- common language:
  - you can move to another analysis or PAG and quickly be productive
  - helps analysis review: you'll know **exactly** what others have done
- provenance
  - you can find out how data sample was made (yours or somebody else's)

Interactive use

- Interactive analysis = anything that can sustain the cycle

1) start running the macro
2) wait for it to finish
3) stare at plot(s), ponder your next move
4) realize that changes are needed
5) edit macro, save it, compile it
6) GOTO 1

- everything except (3) and (4) should not take much time

Interactive Analysis – the options

- Somebody gave you a PAT-tuple, or you made it yourself. Now what? You can run

1. Bare ROOT on CMS data
   - clumsy, tricky, complicated 
2. FW Lite = ROOT + CMS DataFormats + few helper classes 
3. FW Lite standalone executable:
   - same as (2) but without a root[] prompt 
4. cmsRun:
   - CMS Framework is very fast; if no geometry and calibrations are loaded, event processing is approaching #2 and #3 
5. Bare ROOT on TTrees made from CMS data
   - no benefit in portability or speed (same as #2 or #3)
   - one extra step (filling a TTree) = more jobs, more code to write
   - loss of provenance + not a common language !!! 

FW Lite on PAT

- PAT objects are monolithic – perfect for FW Lite!

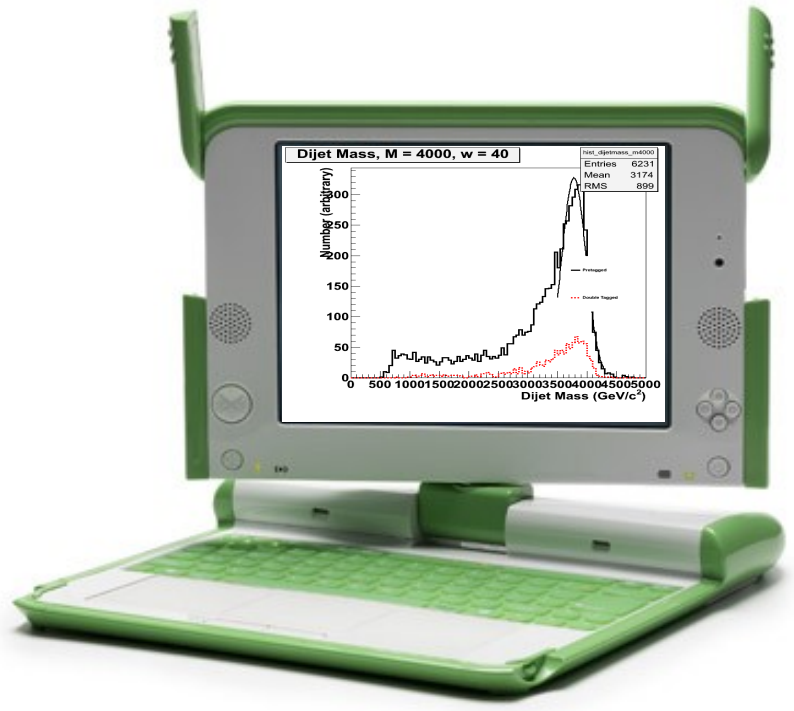
  (unlike RECO and AOD, where the associations make life harder)

- *FW Lite on PAT-tuples = bare ROOT on TTees*
  plus many other advantages since PAT-tuples are CMS data!
  - you can view the event with Fireworks
  - you can use edmProvDump to look into provenance
  - you can use other EDM tools
  - you can use code both in the full Framework and FW Lite
    - but can't use edm::Event, edm::Handle<>, ParameterSet, etc.
    - we recommend writing your event selection as such a function; this will save you a lot of grief later

- NB: don't trust CINT and *always compile* your macro

FW Lite status

- Used for analyses already
- Distribution:
  - you of course want to analyze your PAT-tuples on your laptop
  - SL4.6 supported with CMSSW
  - other GNU/Linuces supported via apt-get or tarball
  - Mac native built available
  - Windows via CERNVM
- Needed for collisions:
  - access to full trigger info
  - more complete access to provenance



Tutorials later today

- There will be two tutorials at 1 pm, running in parallel
- Beginners tutorial:
  - will focus on FW Lite + very basic PAT
  - will be held in Hornets Nest (WH 8X)
- Advanced tutorial:
  - CRAB
  - manipulation of PAT workflows to clean, cross-clean, and define your own PAT-tuple content
  - will be held in Sunrise (WH 11NE)

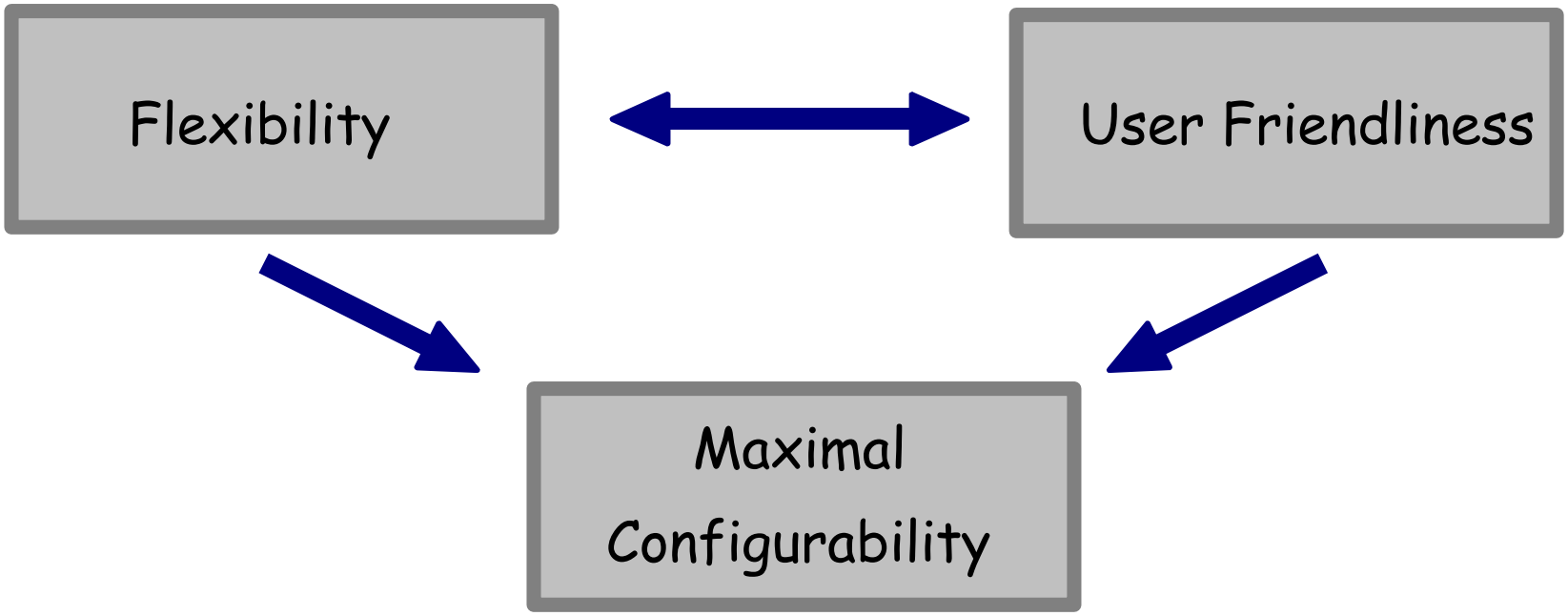
Summary

- Skimming is a vital part of all CMS analyses
- You need to understand your Physics Groups skimming plan
- If needed, make your own skims and final sample (PAT-tuples)
- FW Lite ideal for analyzing PAT-tuples on your laptop
  - will work on most OSes
- Other tools for fitting (RooFit), statistics (RooStatsCms), and little helper classes (e.g. sideband subtraction, histogram management) also available for ROOT and FW Lite
  - All these will be a part of the “CMS laptop kit” ...
- Give us feedback! We take it **very** seriously
  - Complaints on the original PAT workflow resulted in PAT v2.0, ...

BACKUP

PAT Concepts

- Balance competing needs



Other tools for ROOT / FW Lite

- RooStats = inter-experiment statistics tools
  - built on top of RooFit (thus greatly favoring RooFit)
  - included in ROOT 5.22 → into 3.1.0
- RooStatsCms = additional CMS tools
  - in PhysicsTools/RooStatsCms package
  - also included in 3.1.0
- Other helper classes available
  - sideband subtraction (uses RooFit), histogram management, etc.

Your “Laptop Kit”

- Copy your PAT-tuples onto your laptop
- Install
  - FW Lite, Fireworks, edmProvDump, etc.
    - RooFit and RooStats come with ROOT (brought by FW Lite)
  - check out code from CVS
    - most notably RooStatsCms
    - other useful tools as needed (POG algorithms, PAG event selection, sideband subtraction, histogramming, etc...)
  - build your own shared libs locally with scram/ACLiC
  - load them into FW Lite or compile standalone executable
- Go to a coffee shop, and work away!