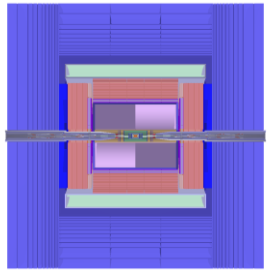
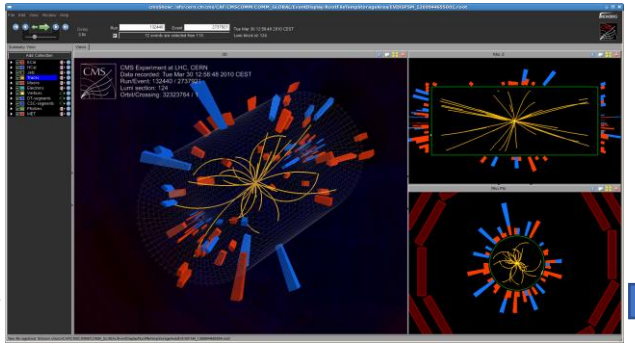
Analysis Tools and Infrastructure

G. Watts (UW/Seattle)

Analysis Software





Production System
Analysis Files

Detector

Trigger

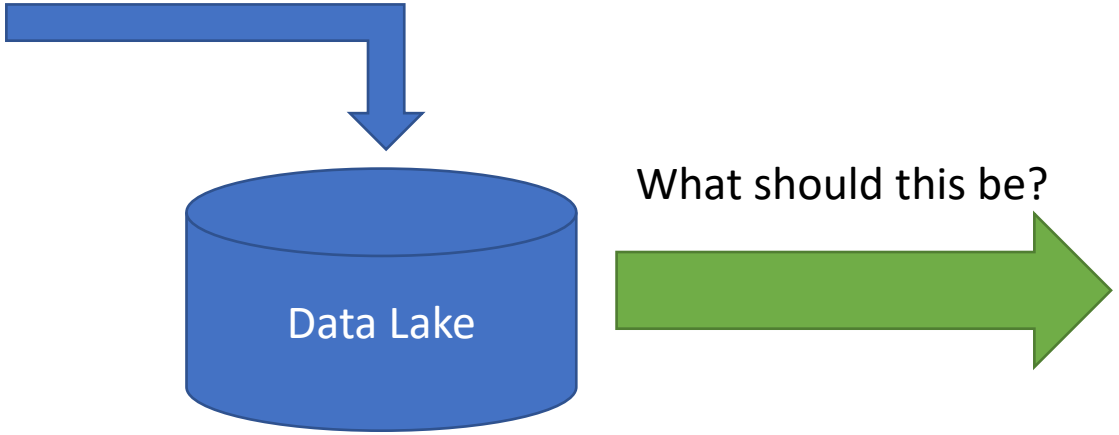
Reconstruction

DataLake storage

Analysis Software

Production System Analysis Files

- File-Based
- ROOT format
- Meta Data (calibrations, lumi, in-file, etc.)



What should this be?

For many reasons the current answer is rucio

This will be the first point of contact for analyzers.

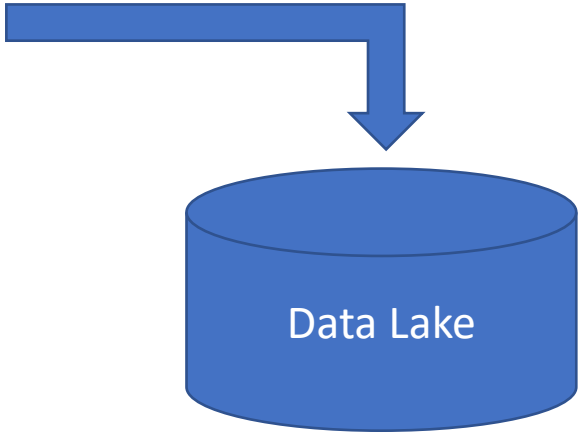
- Users, analysis groups, etc.
- Reduction/etc. runs by analysis groups.

Data-Lake storage

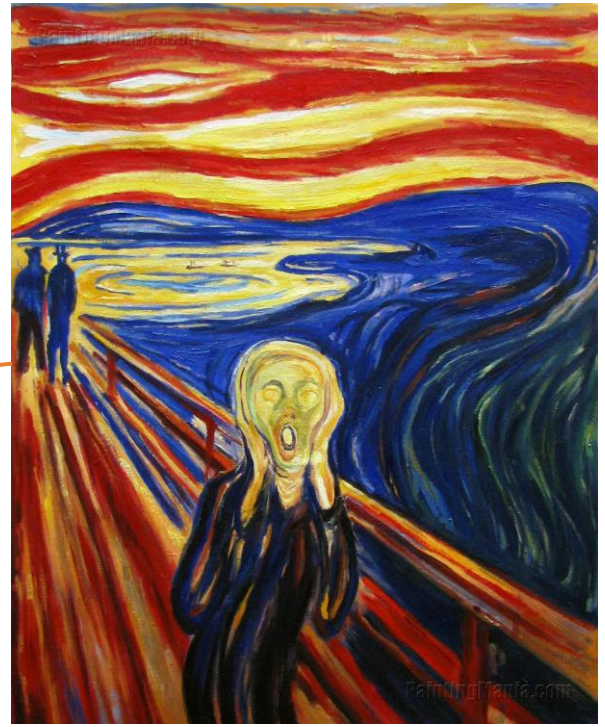
Analysis Software

Production System Analysis Files

- File-Based
- ROOT format
- Meta Data (calibrations, lumi, in-file, etc.)

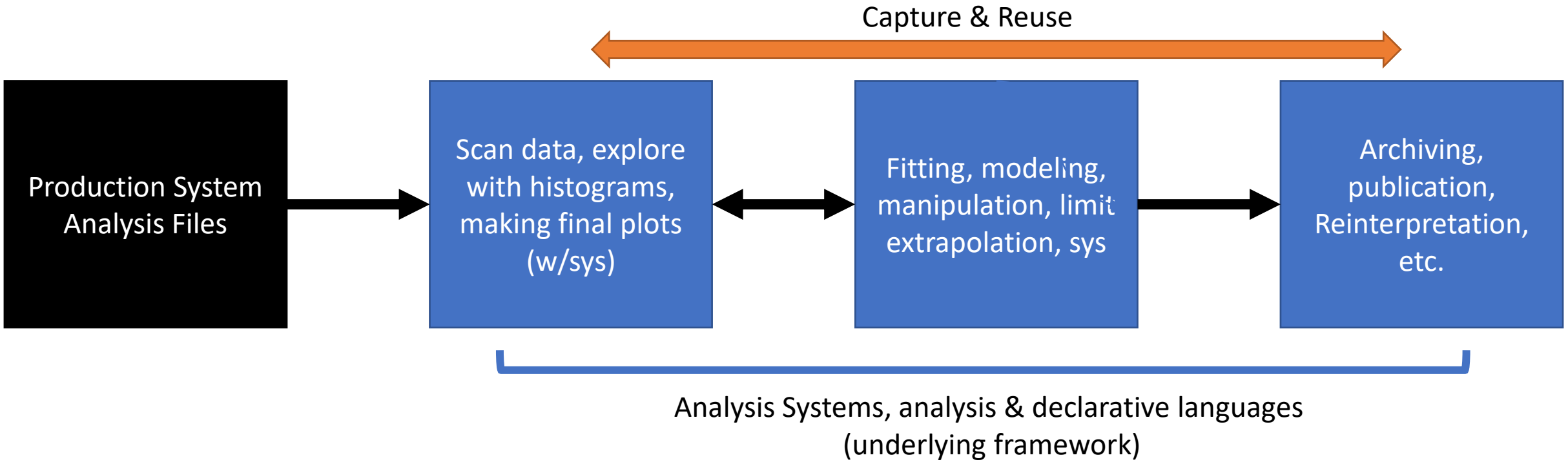


Data-Lake storage

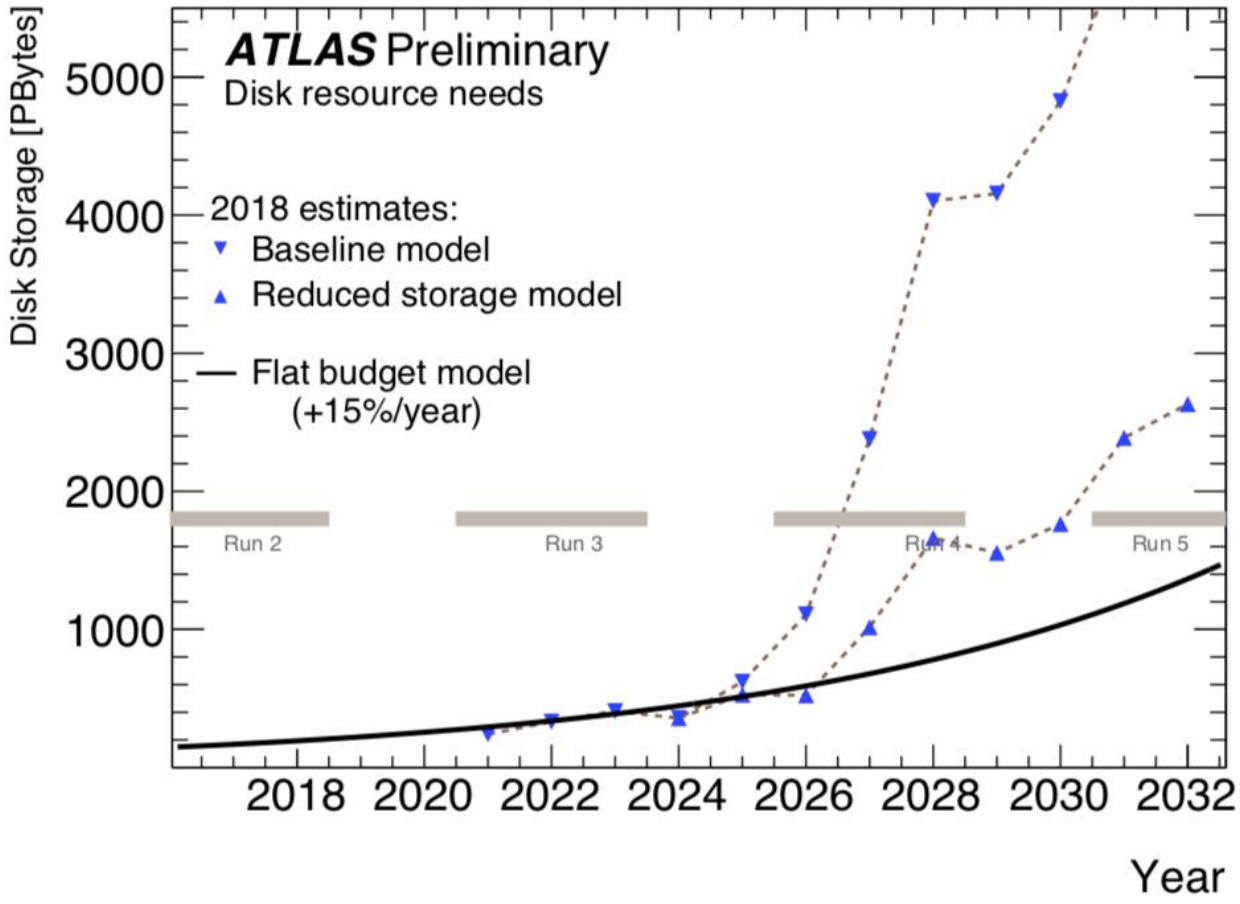


(how to do them “right”?)

Analysis Software From 30K

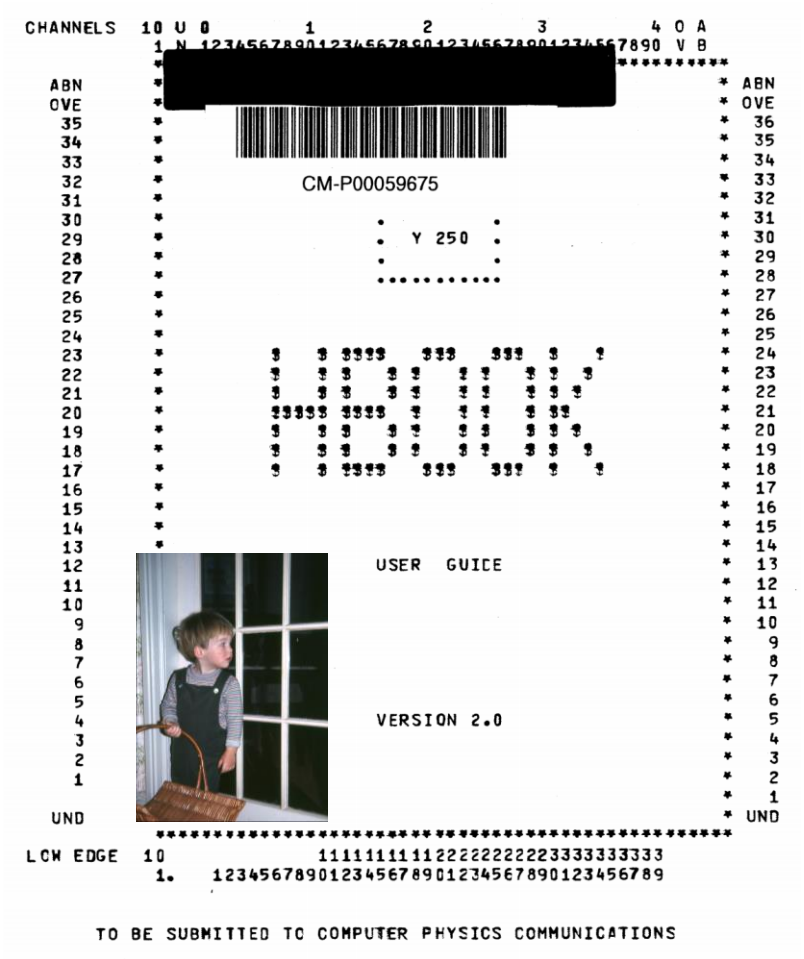


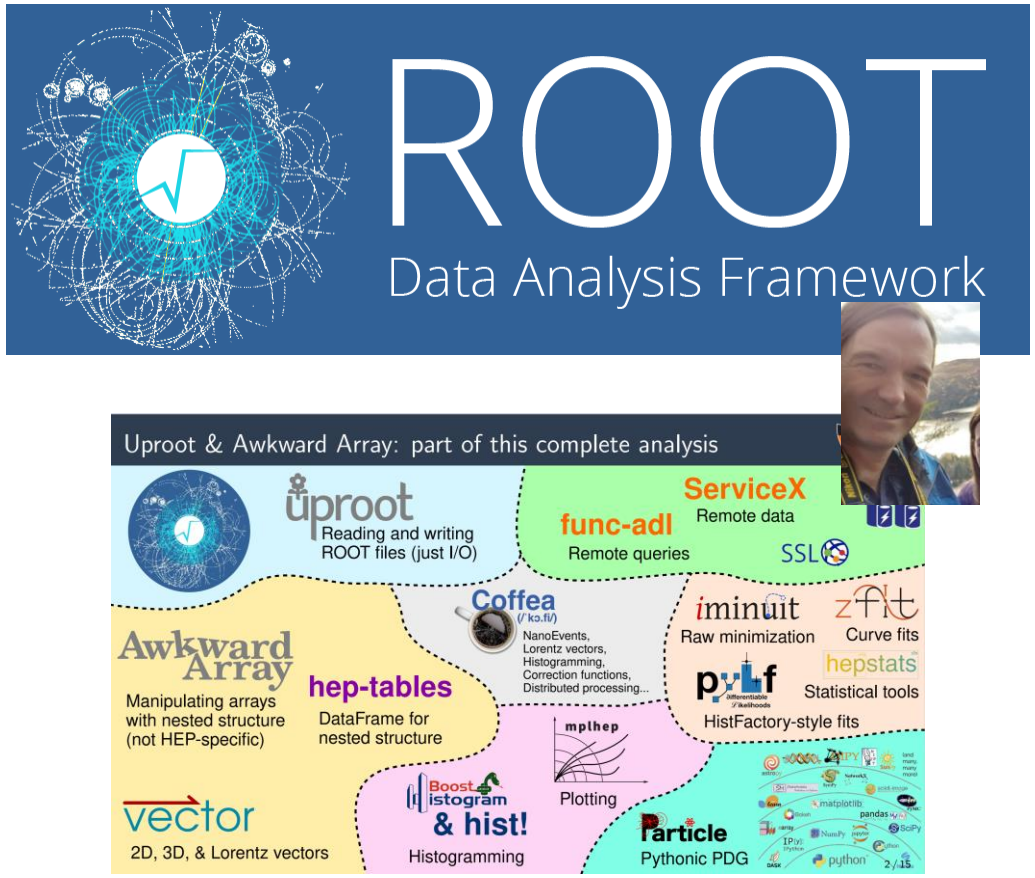
Production Decisions do Affect Analysis!



This will change the way we do analysis in ATLAS

How it started… How it is going…





OpenData
Also plays a roll here

There is a lot new…

Reproducibility

Differentiable Analysis

Python Eco-system, C++ Eco-system

Analysis Facilities

Continuous Integration*

Declarative Analysis, Procedural Analysis

Distributed Analysis, Large Analysis Datasets

Columnar Analysis, Event Analysis

GPU's, CPU's, TPU's, FPGA's

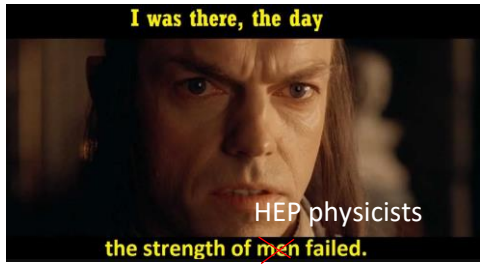
High Level

Low Level

Software professionals vs physicists as software developers

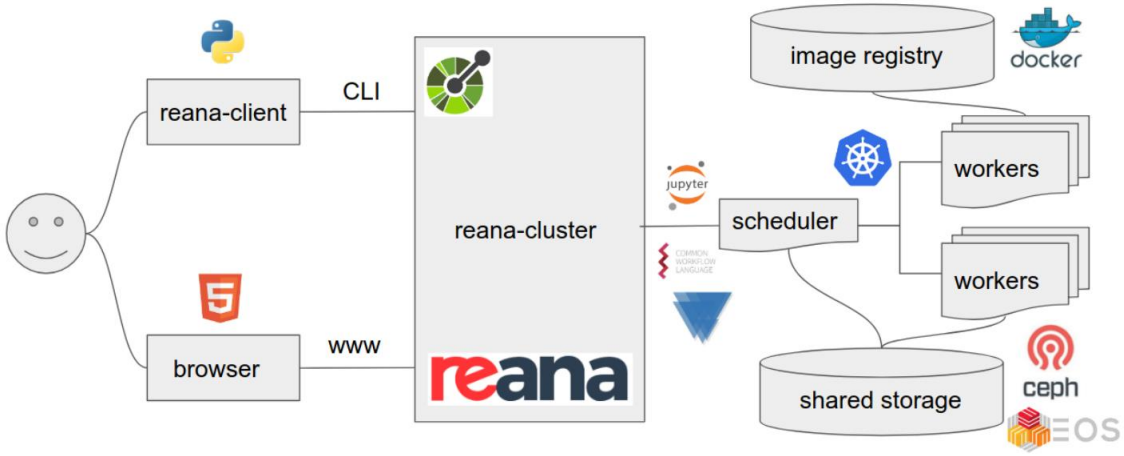
I didn't put ML on here.

Reproducibility



- An analysis can be re-run on new signal models with little work
- Automated system to run an analysis end-to-end

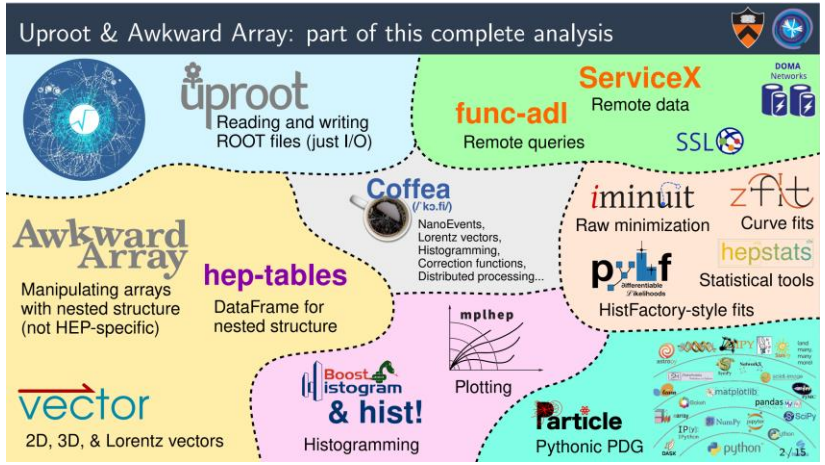
- Can track down that pesky question from the competing detector 2 years later
- Funding agencies are starting to demand this



- Production tools exist for current analysis models
- This touches every part of the analysis pipe-line!
  - Every tool must ask this question if you are going to add it.

Eco Systems

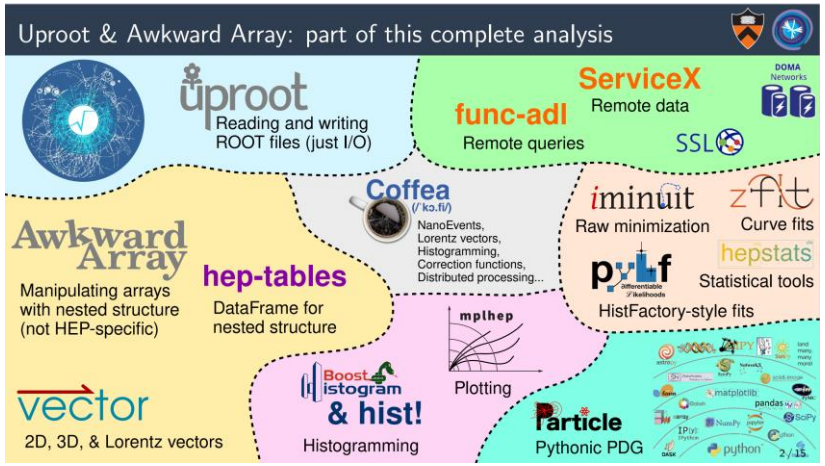
- C++ And ROOT well understood in the community
- Full range of tools customized for our use cases
- 20 years of solid development & debugging
- New tools (like RNTuple, RDataFrame, ROOT7 Histograms etc.) are being added.
- Best in class storage
- Only place we'll do reconstruction & simulation
- Does need support from other labs
  - Don't leave as a service – community participation is desired and welcomed!



- Research and Industry are pouring their resources
- ML frameworks, tools, etc.
  - Someone else did the GPU work
- Almost everything you want to do can be found on stack overflow
- Likely known by your students better than you

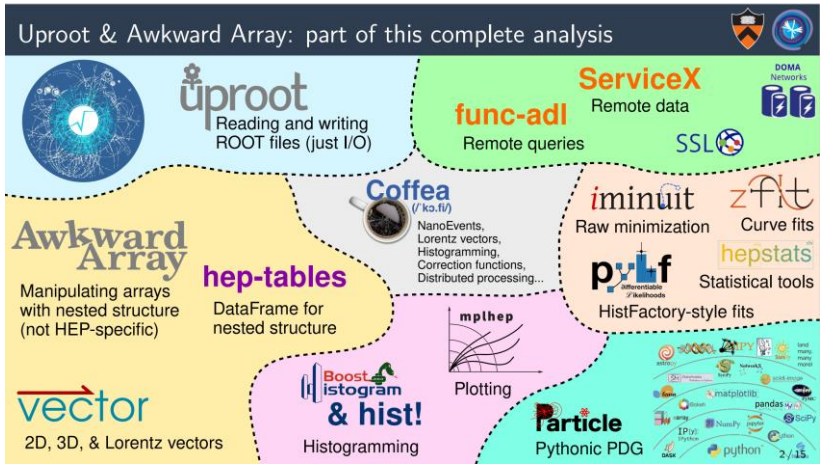
Eco Systems

- C++ And ROOT not well understood by others, and have a significant learning curve
- Moving innovation from industry into C++ world is difficult
- Storage formats in industry
  - Designed to be distributed
  - Rivaling ROOT's abilities



- Industry is not solving the same problems we are
- Industry might move onto a new eco-system
- Python, by-itself is not fast for actual inner-loop processing
  - Much of the code is JIT'ed or C/C++

Eco Systems



Community is finding ways to connect the two eco-systems

What can we do to take the best of both worlds?

e.g. Write your ROOT files to be efficiently ready in both eco-systems

Declarative Analysis

As infrastructure gets more complex you get more and more infrastructure mixed with your physics: it is too hard to do physics!

Procedural Analysis:

```
Loop over events:
  Loop over electrons
    Loop over electrons
      Keep track of pair with mass closest to 91 GeV
  Plot mass
```

Tell the software how to do the job

➡ Fine-tune control, but mixing of control and physics

Declarative Analysis:

```
In every event:
  Find all pairs of electrons
    Best pair with mass close to 91
  Plot mass
```

Tell the software what you want to do

➡ Less boiler plate, more physics

Downside: Computer needs to translate into high-speed analysis

Distributed Analysis

HL-LHC datasets will be too large to spin through in a reasonable amount of time on a single PC

Filling histograms, skimming data for ML training, etc. will become a distributed activity

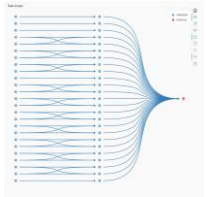
We have non-interactive solutions

- Batch – lsf, slurm, etc
- Batch – htcondor
- Workflow – BigPanda
- Etc.

Building a fault-tolerant distributed system is non-trivial

Industry has created a number of excellent pipe-line processing workflow tools

- parsl
- dask
- ray
- Apache tools (kalfka, etc.)
- Spark, etc.

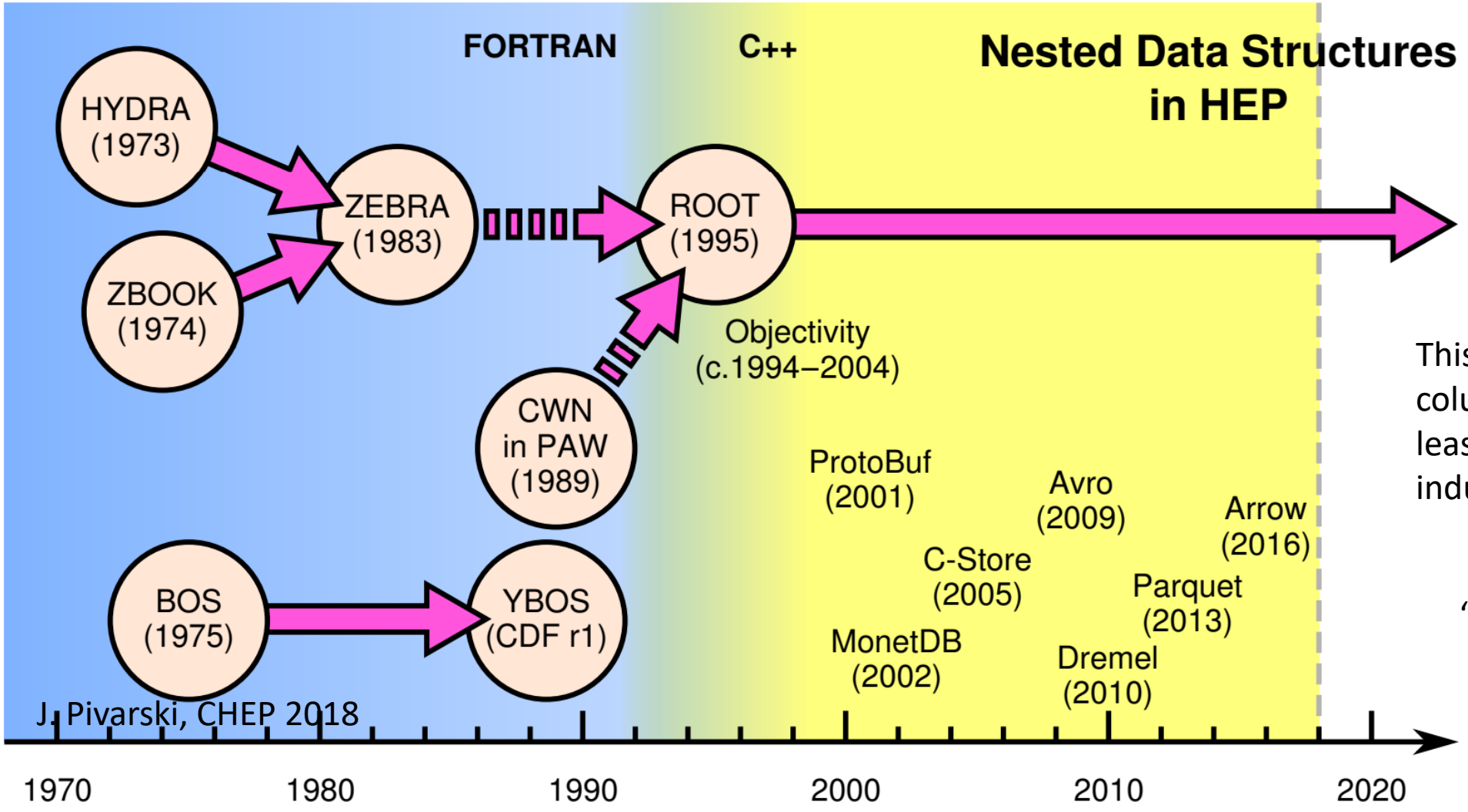


- Some are well suited to running C++ code and collecting the results
- Some can be used to ship data to processors
- All are cluster suitable

Wrappers to make them easy for physicist use?

(e.g. coffea)

Columnar Analysis



This is not new: HEP had column wise data stores at least 10 years before industry

"missed opportunity"

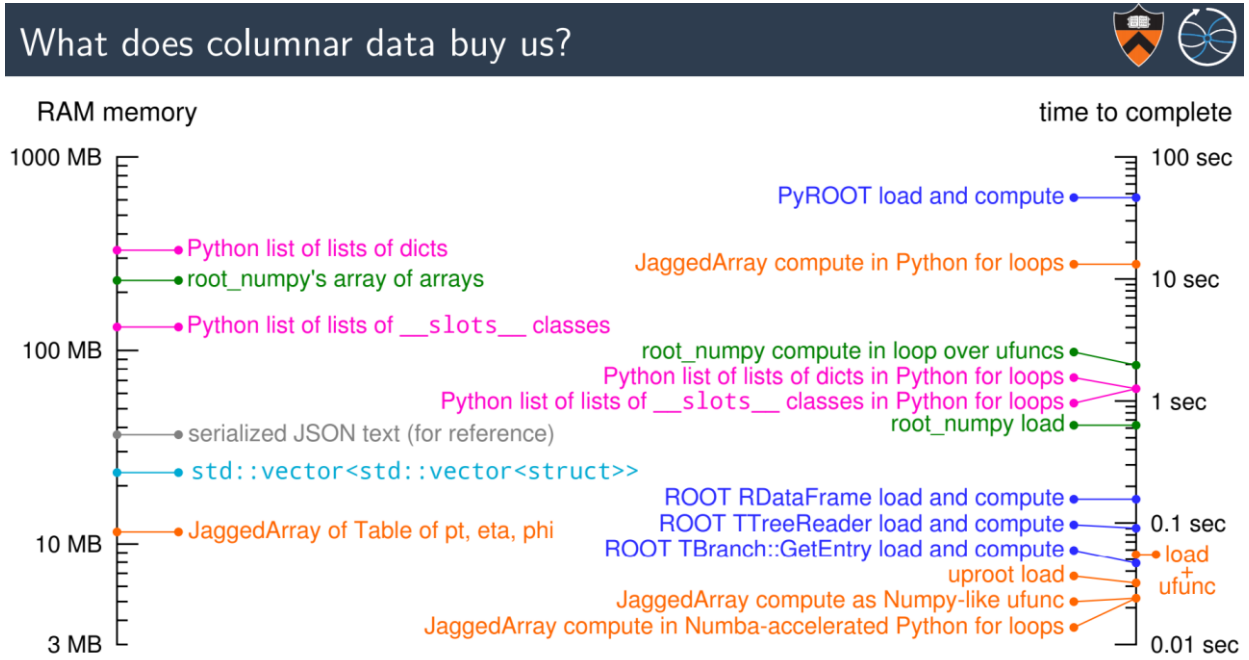
Columnar Analysis and Event Analysis

The physics says process event-by-event

- The computer is much more comfortable using vector operations to process many events at once

Electron1.pt + Electron2.pt

- Add the p_T of the first and second electrons for all 2 billion events



Conceptually works well until you have to ask “how many electrons in an event”

Co-Processors

This is super-hard

Co-Processors

Unless some one else has done the work for you…

Where will they benefit Analysis? → Mostly in AI/ML

Other low-level tools are building it in (like awkward array)

Co-Processors

Where will they benefit Analysis? → Mostly in AI/ML

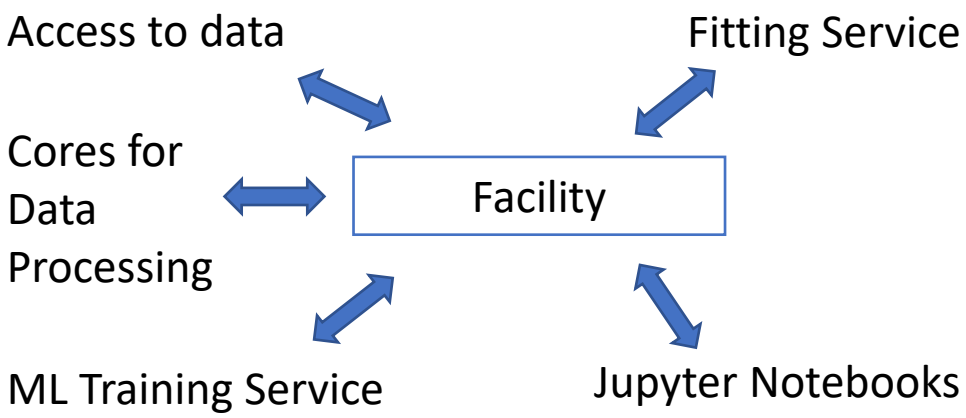
These are tools! I used TF to write an alignment program for the MATHUSLA test stand

Looking Further Out Over The Edge

Analysis is where innovation likely happens first

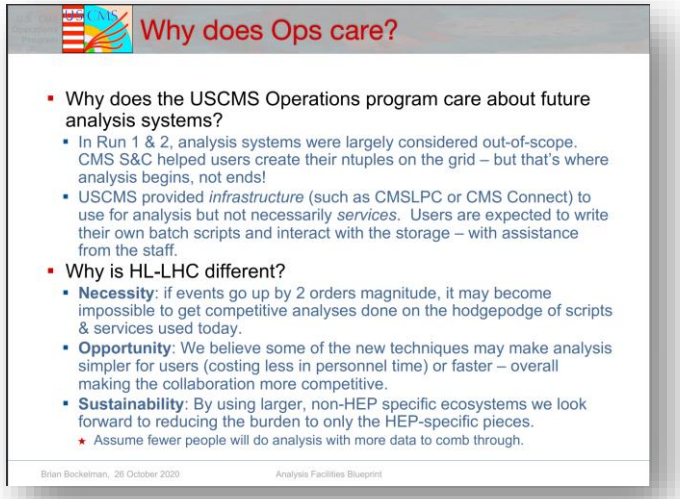
Analysis Facilities

Large datasets mean you can no longer run analysis anywhere!



- Skim data
- Apply corrections
- Build histograms
- Fit complex statistical models
- Train and apply discriminators
- Develop core software
- Submit “GRID” jobs
- Visualize large amounts of data

[See the recent blueprint meeting]



Differentiable Programming

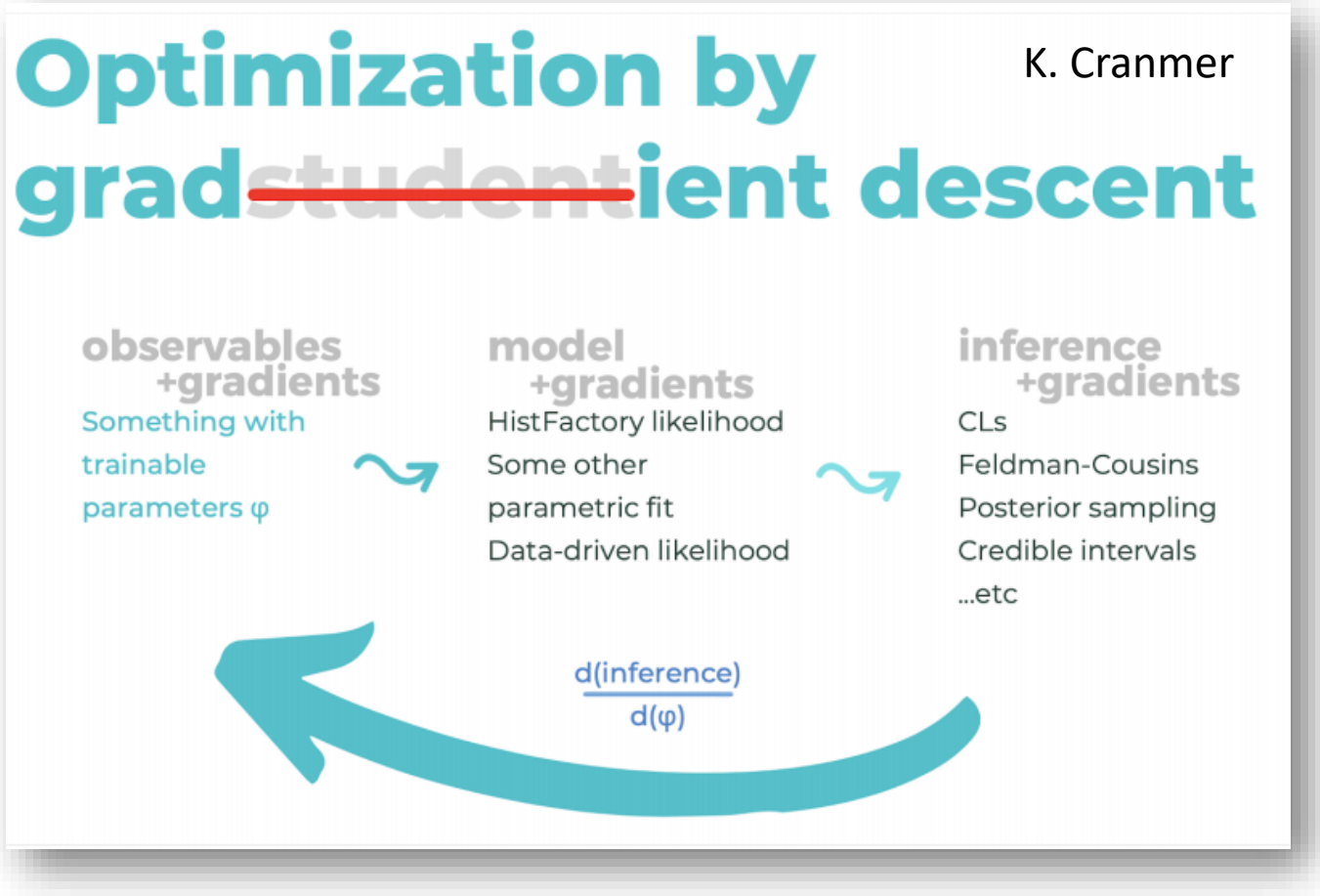
Bring the techniques from ML all the way into the analysis!

Need two jets above a set of p_T cuts? How to choose those p_T cuts?

Use gradient descent to optimize the cuts

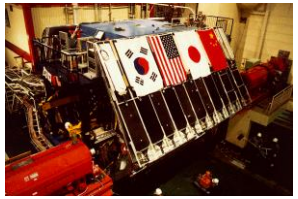
Use final likelihood fit to drive the optimization

Requires auto-differentiation across systems!



Cautions?

Don’t try to make the analysis framework the same as the reconstruction’s



Do not let 1000 analysis frameworks bloom either…

Many people have good ideas

- Getting people to work together to combine frameworks makes things stronger!
- Design process so no one is “losing”.

Conclusions

- CEPC is a different machine
  - The pressures on it are very different than on currently running ones
- Analysis is one of the fastest moving parts of the HEP eco-system
  - What I now call speculative will either be mainstream by the time the CEPC is running or will be consigned to the dustbin of history
- There is a very active community discussing this
  - The HEP Software Foundation
  - In the USA, IRIS-HEP
  - ROOT workshops
  - And many more I'm neglecting here

A lot of people work on this: many solutions are cross-experiment!