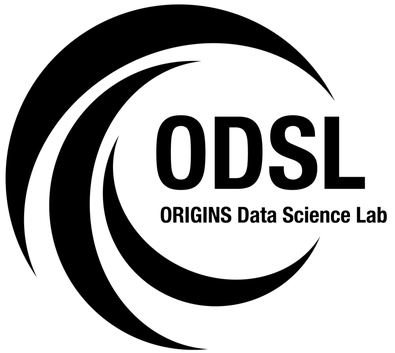
Augmenting PHYS(LITE)

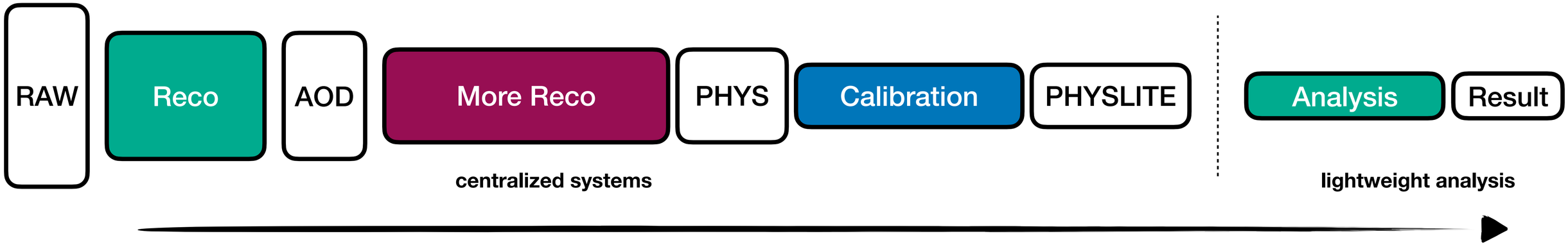
Analysis Ecosystem Workshop II, Orsay

Lukas Heinrich, TUM



Common Formats

User Experience and Storage Constraints encourage us to push much of preprocessing to **centralized upstream systems**



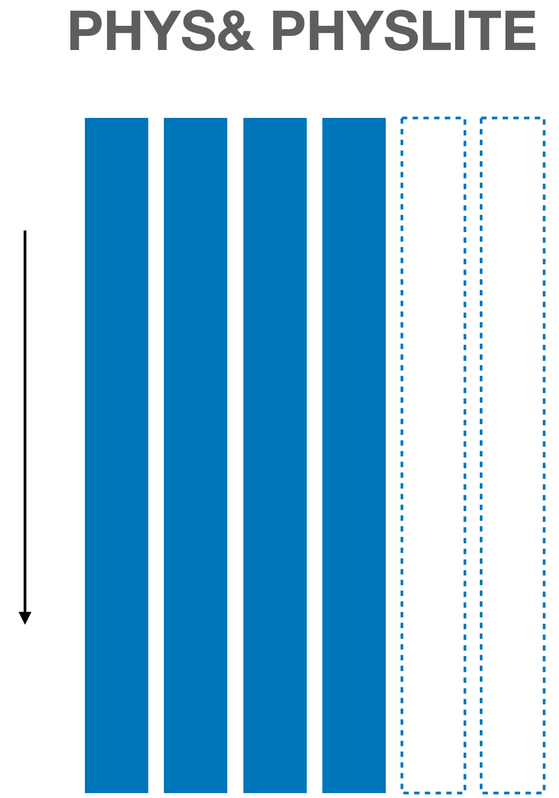
- **more streamlined production (+ flex. for centralized systems e.g. tape)**
- **ability to do common tasks once for everyone**
- **reduced workload on AFs, analyzers, book-keeping, …**
- **central campaigns (for PHYS & PHYSLITE) multiple times per year**

What's not to like?

Missing Flexibility for Users

The price of admission centralized production is more “process” to go through to change upstream work

In ATLAS, it’s planned to produce central formats like PHYSLITE unkimmed, i.e. for every event

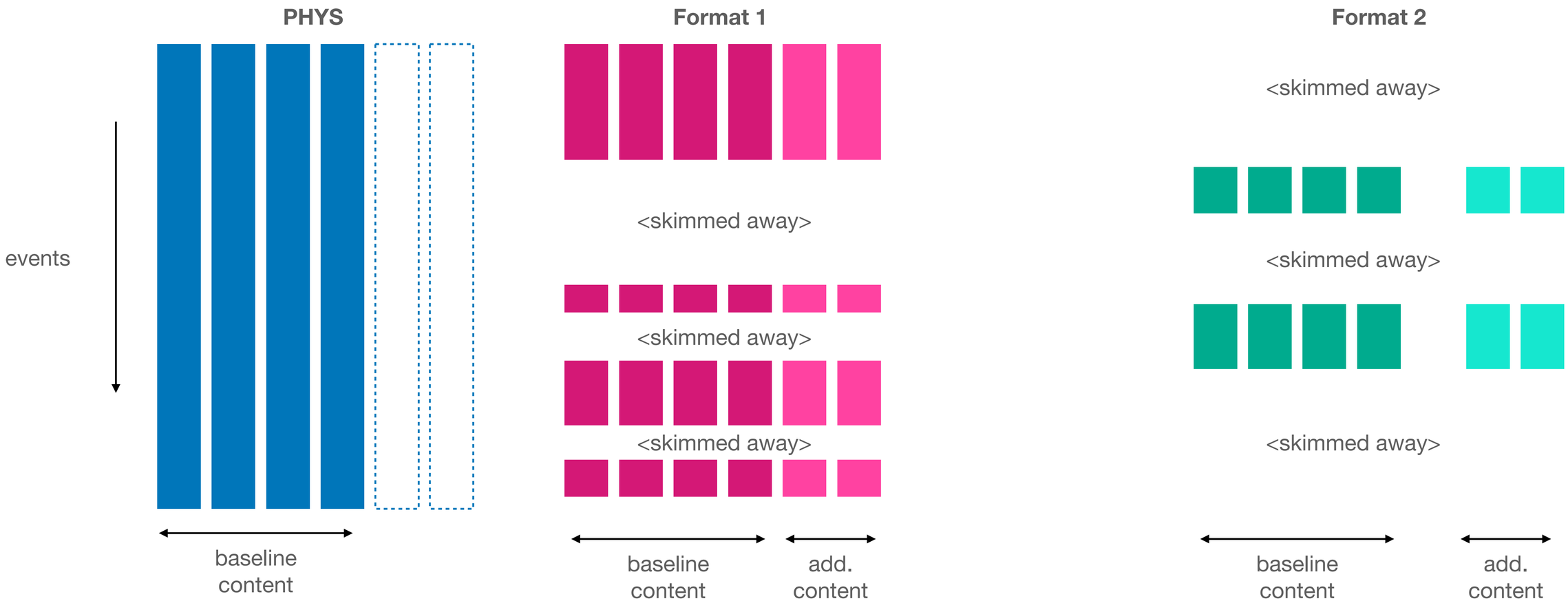


- each change has potentially a big effect storage wise
- want to avoid “ntuple creep” where we just progressively add the kitchen sink
- avoid people not adopting the format b/c it doesn’t have the data they need

Skimmed, but bigger Formats

The traditional reaction is to create **custom formats:** skimmed, but more vars

- but creates lots of overlap & wastes storage,



Having the Cake & Eating it, too

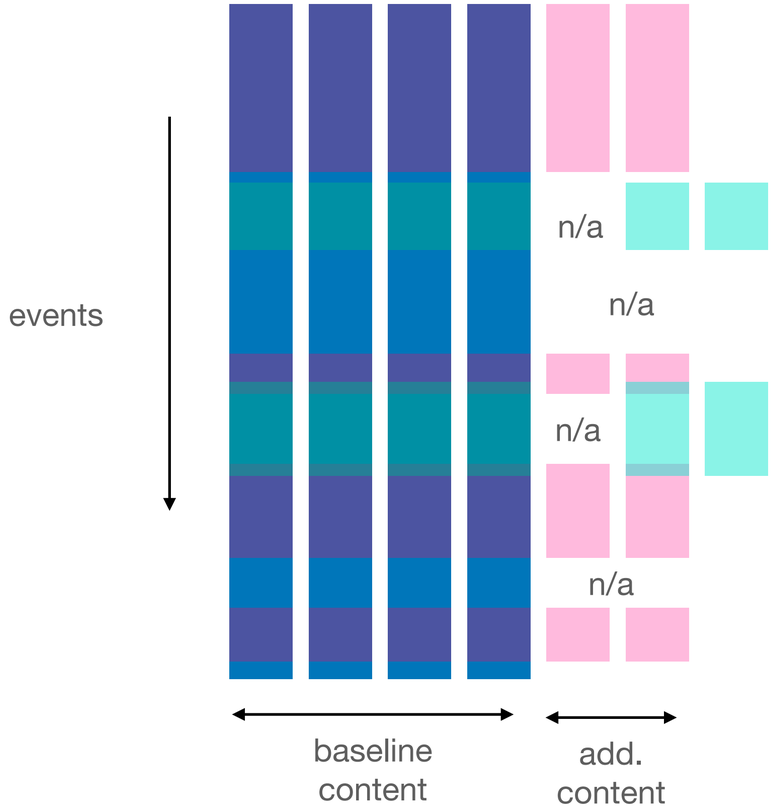
Ideally we want to avoid the overlap & need for additional format.

Can we have a **“fused” format,** that provides

- a baseline content for **every event**
- additional columns depending on the event
  - if event passes **selection A:** add X
  - if event passes **selection B:** add Y

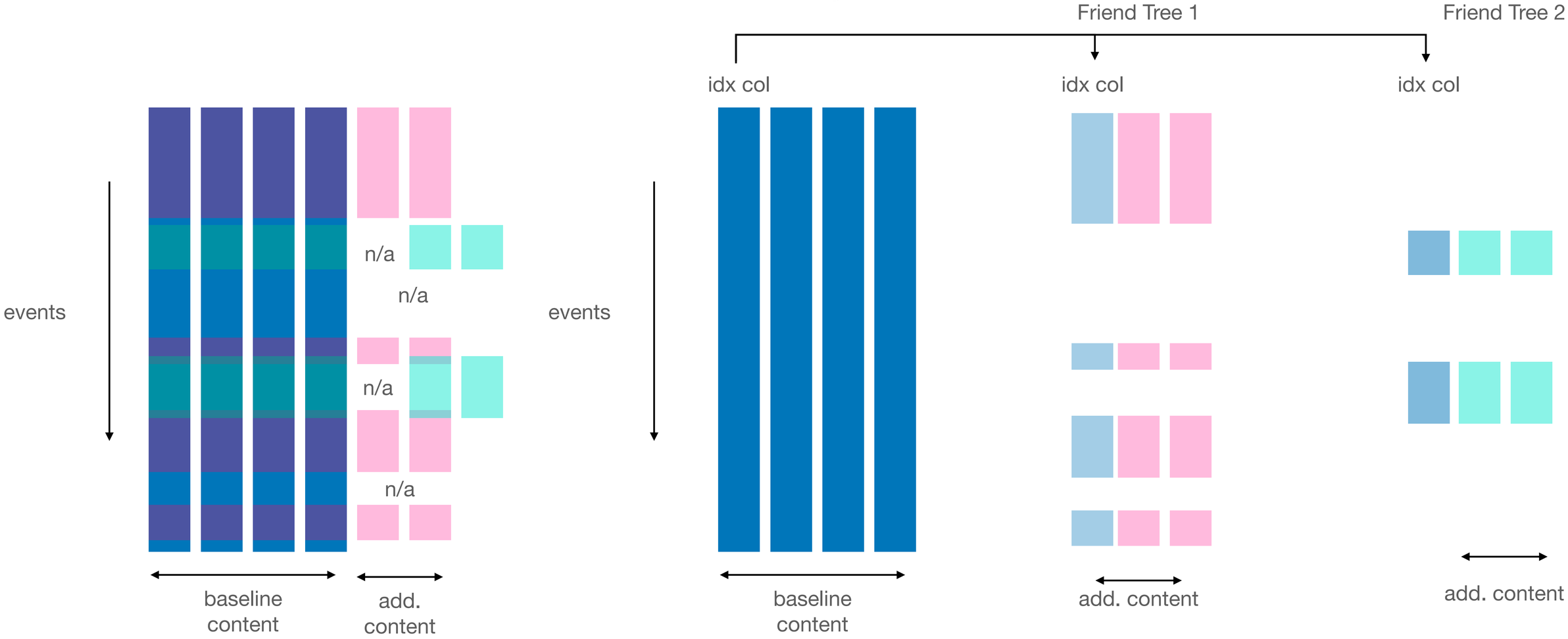
Still provide everything in a single file

- simplifies book-keeping & grid operations, etc..



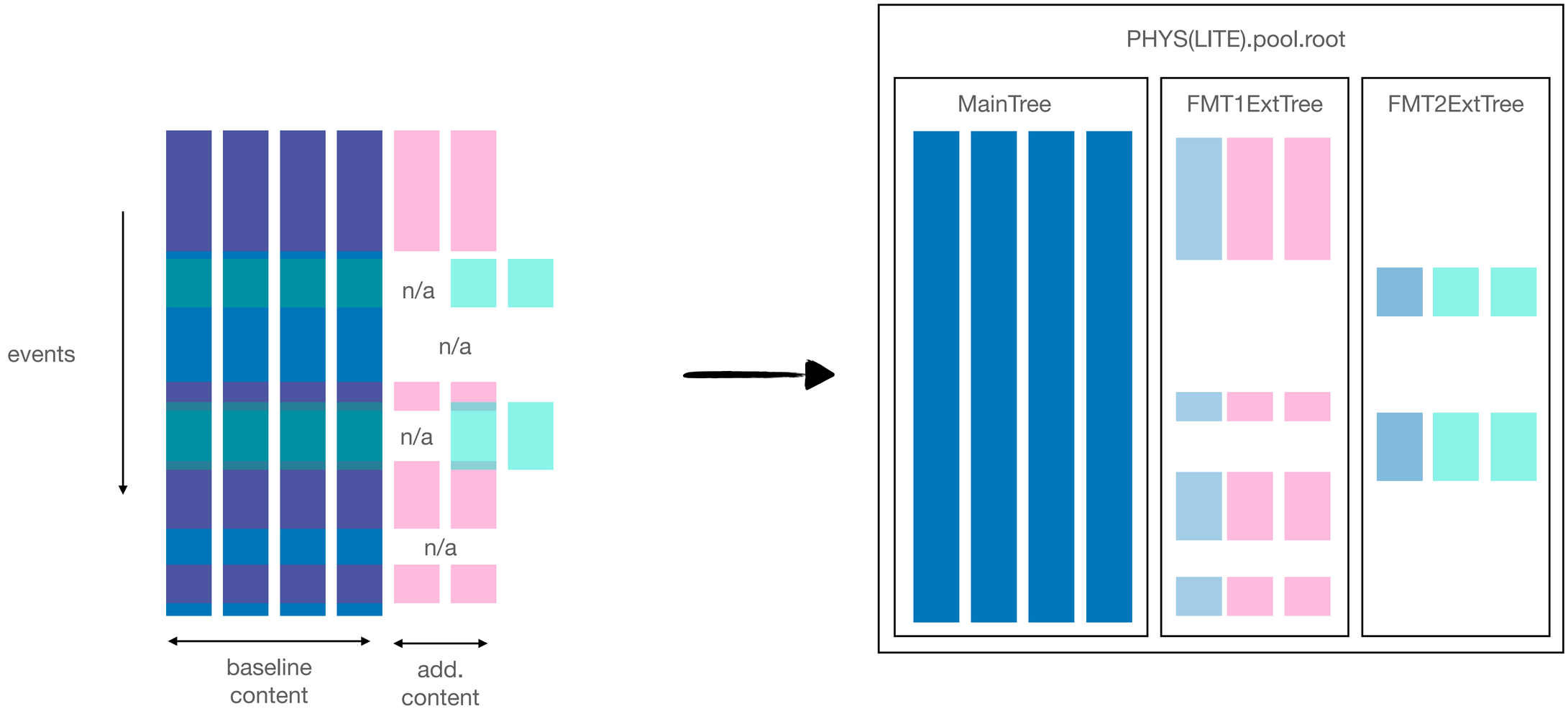
Splitting Into Pieces

Doable by additional skimmed trees with “delta columns” for future use as friend trees. **Overhead:** a few columns to “join” on.



Packing it in a file

To keep it in a single file: prepare output trees in a single pass over events



Re-surfacing an old Athena feature

Athena had a lot of advanced event & data processing features (e.g. “back-navigation”) including writing out additional skimmed trees

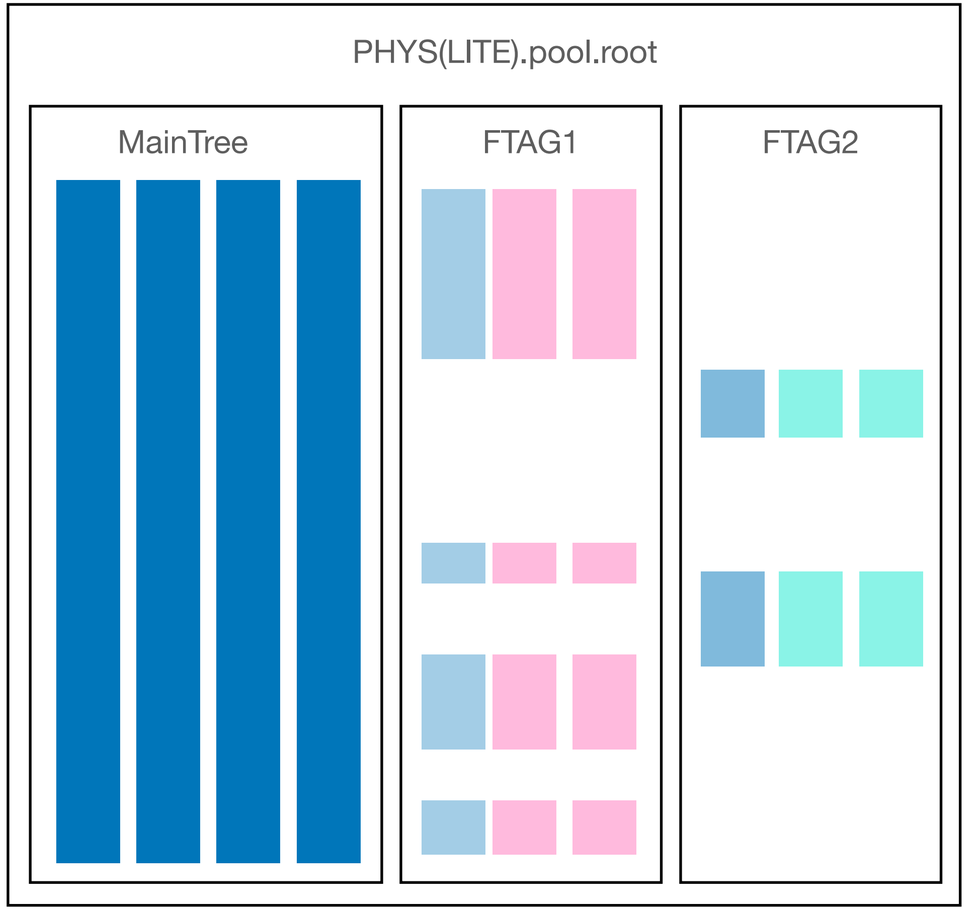
- reactivated after many years of non-use
- seems to “just work” after a bit of cleanup
- even for non-simple tree (i.e. at Athena level)

```
Reco_tf.py …
--reductionConf PHYS FTAG1 FTAG2 …
```

```
…
root [1
TFile**
…
  KEY: TTree    CollectionTree;1        CollectionTree
  KEY: TTree    POOLContainer;1 POOLContainer
  KEY: TTree    POOLCollectionTree;1    POOLCollectionTree
  KEY: TTree    CollectionTree_StreamDAOD_FTAG1;1       CollectionTree_StreamDAOD_FTAG1
  KEY: TTree
  KEY: TTree
  KEY: TTree
  KEY: TTree
  KEY: TTree
…
```

```
COLLECTIONTREE->SCAN("INDEX_REF",
"", "COLSIZE=30")


*****************************************
*    Row   *                  index_ref *
*****************************************
*        0 *             494561189167104 *
*        1 *             494561189167105 *
*        2 *             494561189167106 *
*        3 *             494561189167107 *
*        4 *             494561189167108 *
*        5 *             494561189167109 *
*        6 *             494561189167110 *
*        7 *             494561189167111 *
```

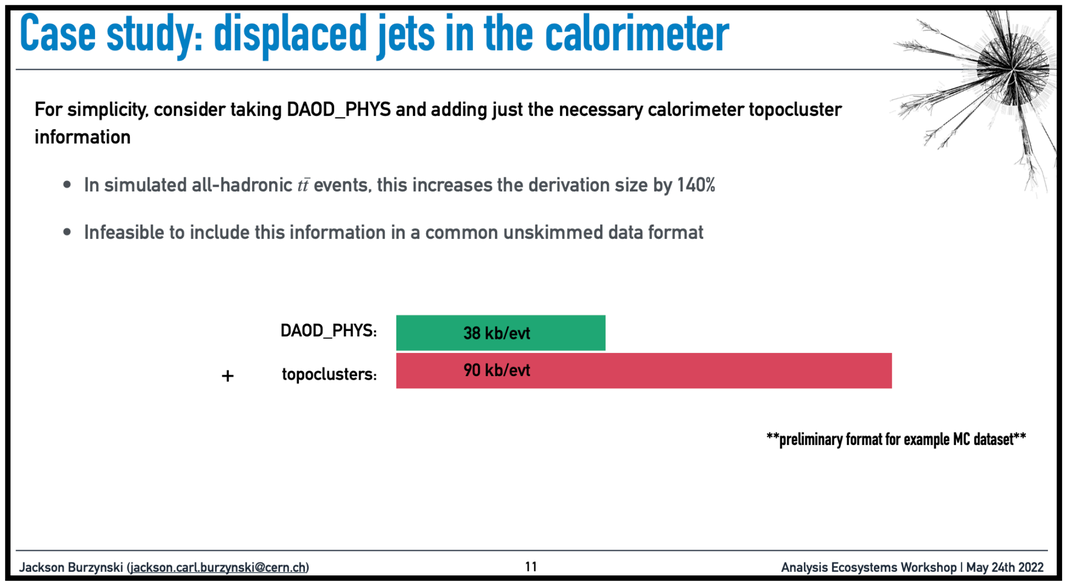
```
COLLECTIONTREE_STREAMDAOD_FTAG2-
>SCAN("INDEX_REF", "", "COLSIZE=30")


*****************************************
*    Row   *                  index_ref *
*****************************************
*        0 *             494561189167107 *
*        1 *             494561189167110 *
*        2 *             494561189167112 *
…
Type <CR> to continue or q to quit ==> q
*****************************************
```

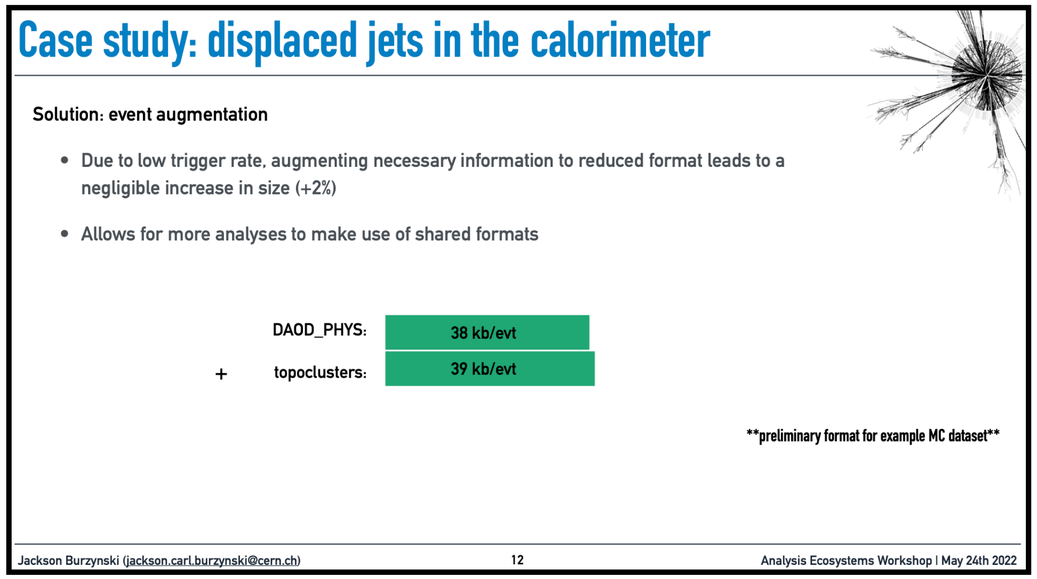


Re-surfacing an old Athena feature

First Tests with LLP formats seem promising (see Jackson's Talk)

- able to replace 5 formats by fusing them into the common PHYS format



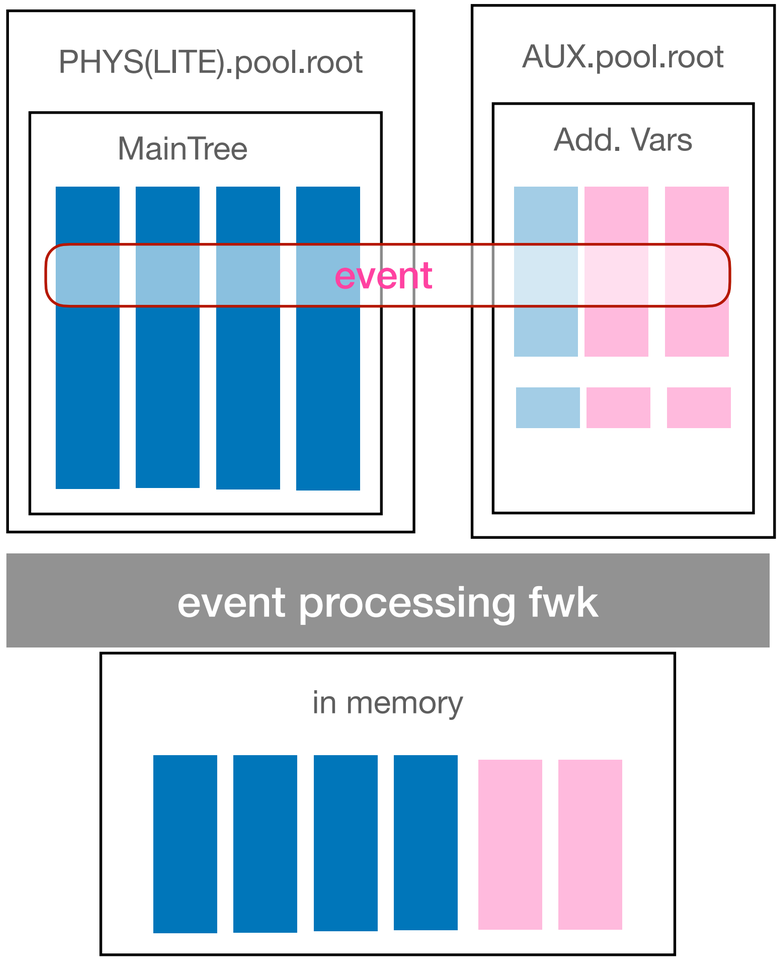


After-the-fact columns

Situation may arise where a few more column would have been useful for a given analysis (“back-navigation”: go back to source and extract add. variables)

Could use infrastructure to allow users to request “on-demand” add. columns for skimmed selections

- event processing framework would need to be able to synchronize / find / load event data from N sources
  - present a unified view in-memory
- otherwise wait for next prod. round.



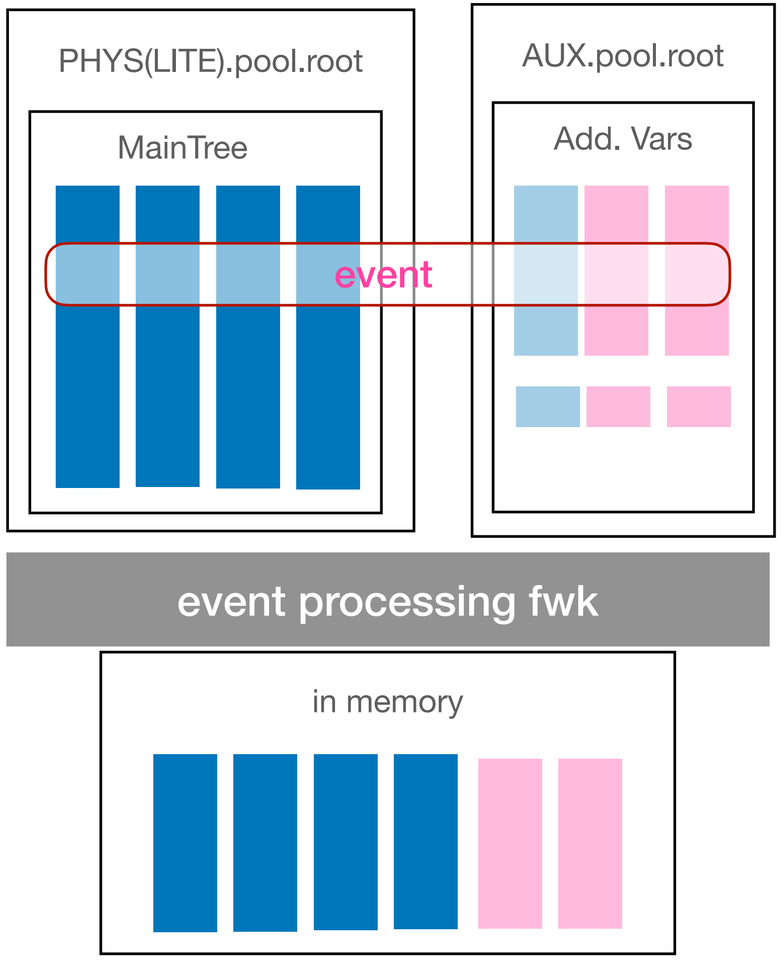
After-the-fact columns

Situation may arise where a few more column would have been useful for a given analysis (“back-navigation”: go back to source and extract add. variables)

Big worry: single event is split across many files

- book-keeping nightmare?
  - need to be absolutely sure the “join columns” work in all cases & **are consistent** across runs
  - versioning (which delta goes with which bulk?)

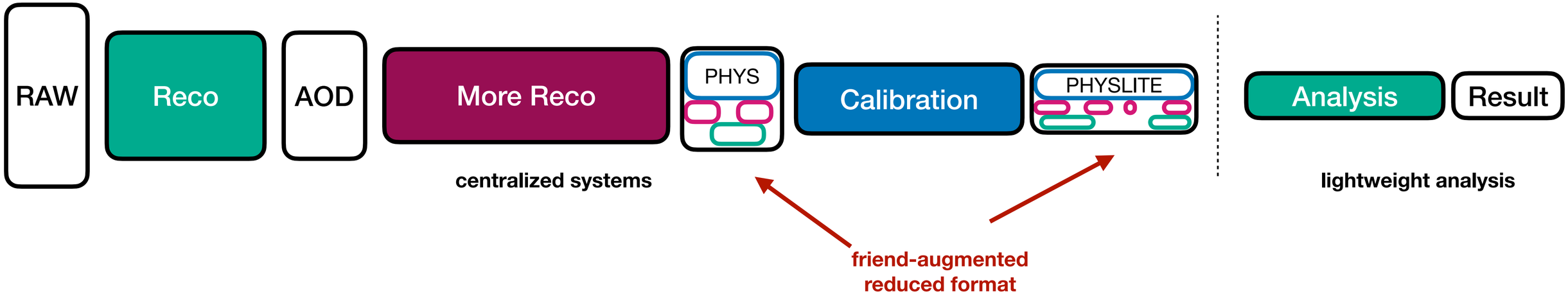
- stability of event processing
- more workable for AF-style processing than distributed grid with per-job stage-in?



Operation Ideas

Would be fairly straight-forward to friend-tree-augmented reduced formats

- reduce burden to add new variable (if skim is tight enough, can add a lot)
- still have central tracking of many formats, still centralized production, but can keep larger fraction of analysis on “common datasets”
- on-demand “deltas” would be a bit more involved



Summary

Centralized Formats are a good idea: hoping for streamlined production experience, reduced size, leaner analysis

But can easily introduce inflexibility for analyzers (missing variables, high threshold to get new ones added to unskimmed format etc.)

Investigating “fused formats” with in-file skimmed friend trees to provide targeted formats without losing central processing & reduced overlap

Reactivated old Athena tricks to make it work nicely, and first tests with LLP use-case looks promising