

# Analysis Grand Challenge

**Alexander Held (UW-Madison)**

**Oksana Shadura (UNL)**

**IRIS-HEP Retreat 2022: <https://indico.cern.ch/event/1196111/>**

# AGC: two components

The AGC has **two components**:

- Defining a **physics analysis task** of realistic scope and scale
- Developing an **analysis pipeline** that implements this task
  - Finding & addressing performance bottlenecks & usability concerns

# AGC: how we envisioned it initially

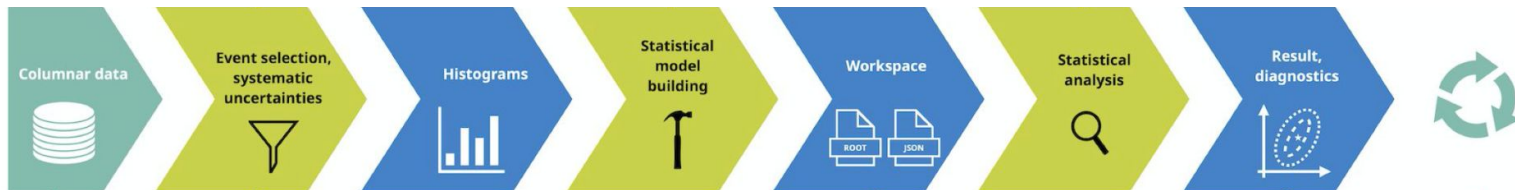
## “Integration exercise” for IRIS-HEP

- Demonstrate method for **handling HL-LHC data pipeline requirements**
  - Large data volumes of  $O(100 \text{ TB})$  + bookkeeping
  - Handling of different types of systematic uncertainties
  - Use of **PHYSLITE** / **NanoAOD** formats, aligned with LHC experiments
- **“Interactive analysis”**: turnaround time of ~minutes or less
  - Made possible by highly parallel execution in short bursts, low latency & heavy use of caching
- **Specify all analysis details** to allow for **re-implementations** and re-use for benchmarking
- Execute on AF

# AGC: analysis task

## Community benchmark

- Main AGC analysis task: **ttbar cross-section measurement** in single lepton channel
  - includes simple top reconstruction
  - setup chosen as it captures relevant workflow aspects and can easily be extended
    - e.g. conversion into a BSM search
  - analysis task prominently features handling of systematic uncertainties
- Analysis is based on **Run-2 CMS Open Data** (~400 TB of MiniAOD available)
  - Open Data is crucial: everyone can participate
  - currently using 4 TB of ntuple inputs (pre-converted, ~1B events before cuts)
- Goal of setup is showing functionality, not discovering new physics
  - want to capture workflow; use made-up tools for calibrations & systematic uncertainties



# AGC: status

## Usability & performance improvements

- **Demonstrated analysis pipeline** at [AGC Tools workshop](#) / [PyHEP 2022](#)
  - Workshop also features details of all tools & services used in pipeline
  - **Pipeline setup**
    - **ServiceX** delivers columns following declarative **func\_adl** request
    - **coffea** orchestrates distributed event processing & histogram production
      - Using **uproot**, **awkward-array**, **hist**
    - Visualization with **hist** & **mplhep**
    - Statistical model construction with **cabinetry** & inference with **pyhf**
- Demonstrator highlighted possible **usability & performance improvements**
  - Everything tracked in [agc#64](#)
  - Main performance bottlenecks are being addressed
- **Everything is openly developed** ([AGC repository](#))
  - Including categorization of datasets in terms of role in AGC demonstrator ([AGC repository](#))

alexander-held commented on Apr 27 • edited • Member

This collects various user experience and performance related aspects that the CMS Open Data pipeline demonstration at the AGC 2022 workshop revealed.

### User experience

#### ServiceX+ coffea

- ☐ schema configuration with ServiceX processors in coffea
- ☐ naming transformations `ssl-hep/ServiceX#407`
- ☐ understand differences between `auto_schema` and AGC schema with similarly named columns `CoffeaTeam/coffea#665`
- ☐ `auto_schema` for non-jagged columns `CoffeaTeam/coffea#664`
- ☐ non-async method to run ServiceX processor for easier debugging (ideally, `NamedEventsFactory.from_rest`-like method)
- ☐ single-letter error messages `CoffeaTeam/coffea#666`, `ssl-hep/ServiceX#408`

#### ServiceX

- ☐ it can take a long time for transforms to report how many files are to be processed in total
- ☐ limiting number of files when querying a rucio dataset `ssl-hep/ServiceX#395`, works via file name suffix (cell 2 in `03_atlas_XAOD.pyynb`)
- ☐ MinIO filling up: automatic cleanup?

#### coffea

- ☐ metadata caching `CoffeaTeam/coffea#662`
- ☐ objects changing in surprising ways in systematic variations `CoffeaTeam/coffea#661`
- ☐ allow attaching per-object systematic variations to the full event (to enable running over copies of events)? not great for performance, but convenient for usability
- ☐ weight-based systematics that use object properties but are attached to events `CoffeaTeam/coffea#667`

#### coffea-casa

- ☐ dask manual scaling settings seem to not be accepted
- ☐ ServiceX dashboard

#### func\_adl

- ☐ find ways to format queries in a way that helps understand the "layer" at which a given operation acts

#### processor design

- ☐ avoid stacking masks of different shapes together (when built after initial filtering), hard to keep track of shapes (perhaps `keepjagged=True` or masking with `None`)
- ☐ improve systematics loop, potentially streamline everything to use the same pattern, or find a way to automatically track which columns change when, and automatically expand observable with systematics dimensions

### Performance

#### ServiceX+ coffea

- ☐ dask scaling `CoffeaTeam/coffea#611`

#### ServiceX

- ☐ DID finder becomes a bottleneck when running over a large amount of files

#### coffea

- ☐ consider splitting out pre-processing `gist / CoffeaTeam/coffea#675`, or merge input files to avoid bottleneck

#### servicex-databinder approach

- ☐ avoid bottleneck with file conversion / copying (feed data straight to Skyhook?)

#### coffea-casa

- ☐ understand issues showing up in dask task stream (file access?)
- ☐ possibility of guaranteeing fixed number of workers for performance benchmarking

#### func\_adl

- ☐ implement full query with proper b-tagging of jets with  $p_T > 25$  GeV

👍 1

# AGC: give it a try!

We are making it easy for you to try out our setup

- One click to get PyHEP notebook in Binder environment

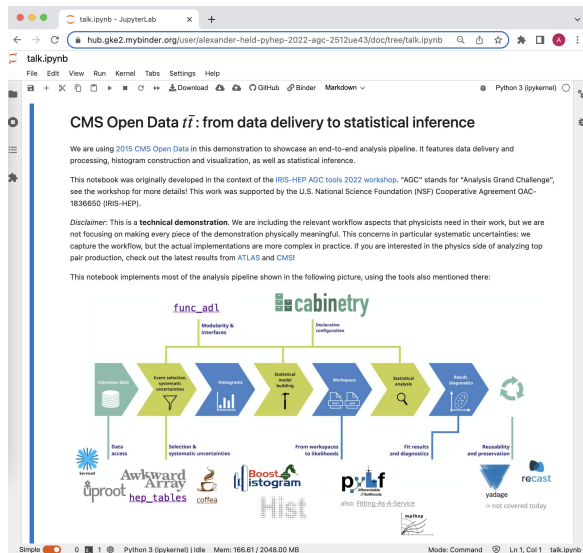
- Try it out today!

- You can also use the UNL Open Data coffea-casa

- Or SSL, or your favorite facility

- This allows you to scale up (limited on Binder)

- Everything is available in the AGC repository



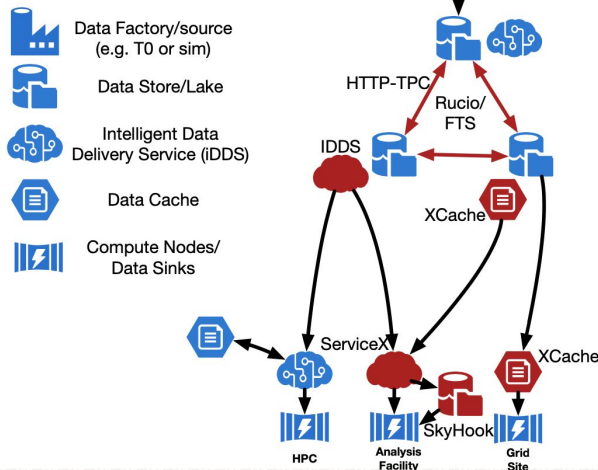
# Essential pieces of the AGC

Data, AS, DOMA & SSL

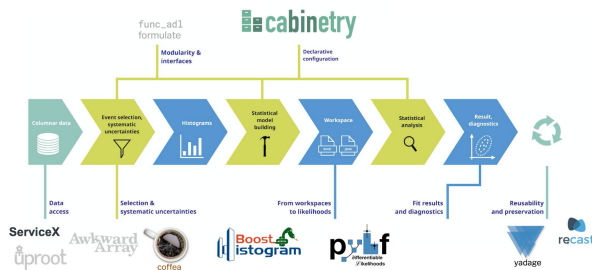
# AGC in IRIS-HEP

## Connecting different areas: done!

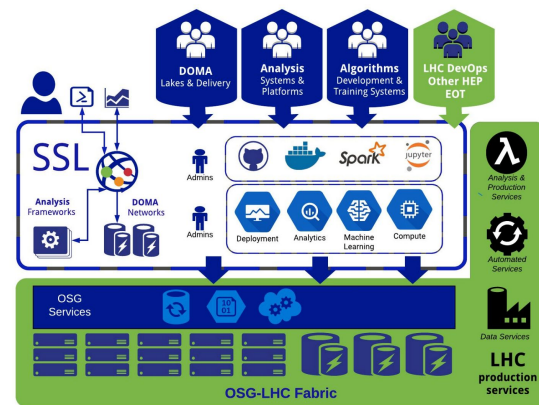
**Data Organization, Management and Access (DOMA): data delivery**



**Analysis Systems (AS): tools**



**Scalable Systems Laboratory (SSL): deployment techniques and resources**





# AGC: dataset

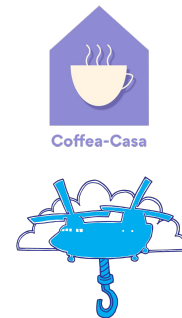
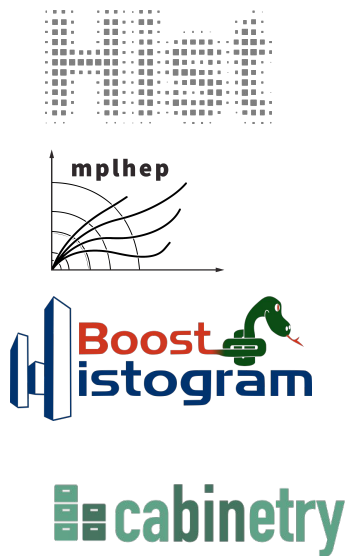
**We rely on Open Data to make the AGC available to anyone**

- Lots of **help** from **CMS DPOA group** with available Open Data
  - CMS also picked up the AGC analysis task & (parts of) our implementation for the [CMS Open Data workshop](#)
- **US CMS fellow project**: Sneha Dixit (supervised by Nick Smith) enabled NanoAOD generation for 2015 CMS data
  - Will allow us to switch input data from custom ntuples to NanoAODs

# AGC: software stack

Involves large number of packages from IRIS-HEP and partners

We are interested to add *correctionlib* to AGC pipeline



Analysis specific frameworks and packages (available in Docker container)

Data delivery  
service (k8s)

Optional  
services (k8s) 10

# AGC: data management tools

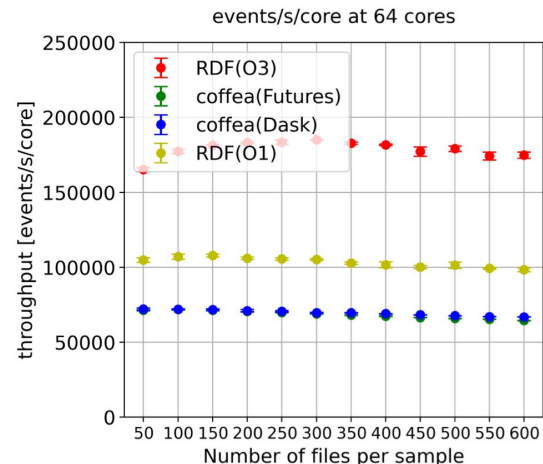
## Relying on DOMA R&D for fast physics analysis turnaround

- First **comparisons** with RDF (fellow project) show **room for performance improvements**
  - No caching / data delivery service used in comparison (uses local inputs)
- Expect **key contributions to improved performance** from three DOMA projects:

XCACHE

ServiceX

Skyhook



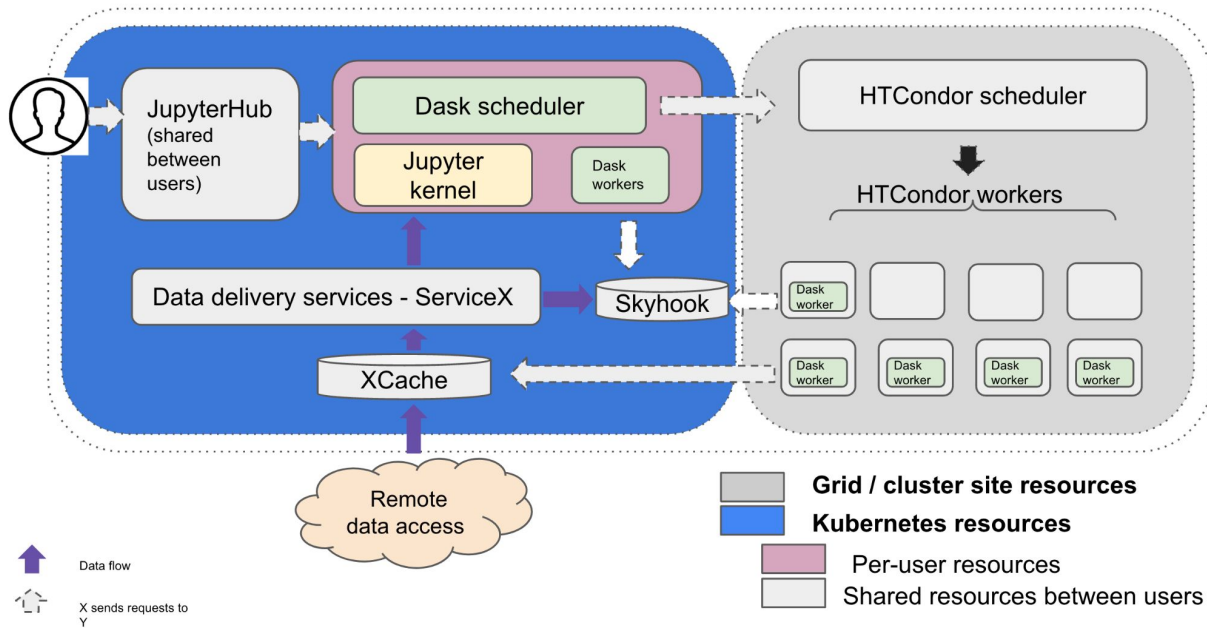
[Andrii Falko's fellow presentation](#)

*We hope to have more contributions from AS/DOMA teams to help us to follow up integration issues*

# R&D on Analysis Facility

## Rapid prototyping on coffea-casa AF

- **Glueing AS, SSL & DOMA together**: an AGC execution environment
- Providing **environment to explore analysis workflows** at scale



**AGC: execution phase**

# AGC: execution phase

## AGC execution workshop

- Have an *AGC Execution Workshop* sometime after CHEP 2023
  - Potentially end of May? Later?
- Inviting everyone who is interested to share their setup and present results
  - Including US ATLAS / CMS AFs
    - Interesting combinations of hardware, network site configurations
  - Any type of “combinatorics” of AGC analysis implementation / the components setup
  - The chance to publicize site to physics analysis community :-)

# AGC execution phase

## Participants

*We would benefit from dedicated people @ facilities to help us to follow up integration issues*

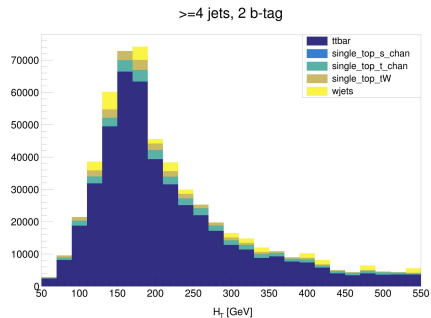
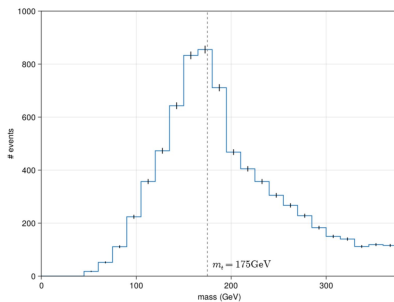
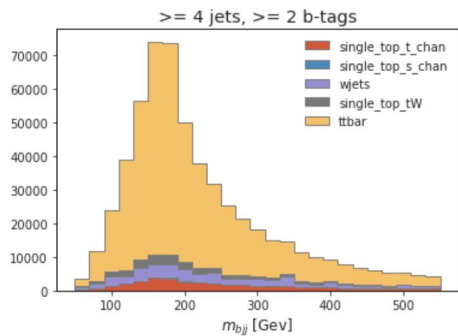
	BNL	FNAL	SLAC	UNL	UChicago
basic <b>coffea</b> (e.g. IterativeExecutor) -> <a href="#">notebook</a> with <code>USE_DASK = False</code>	✓	✓		✓	✓
<b>coffea</b> scaling (e.g. with Dask) -> <a href="#">notebook</a> with default settings*		✓		✓ (using HTCondor @ Tier2, planning to switch to k8s)	✓ (occasional segfaults at scale)
standalone <b>ServiceX</b> -> <a href="#">notebook</a> (no configuration)	✓	✓		✓	✓
<b>ServiceX+coffea+scaling</b> -> <a href="#">notebook</a> with <code>PIPELINE = "servicex_processor"</code>				✓	✓
<b>XCACHE</b> support	✓	✓ (some performance caveats, to be understood)	✓	✓	✓

\* may need site-dependent Dask cluster configuration, see [implementation](#), please get in touch in case of questions

# AGC implementations

## Community effort

- *coffea*: [iris-hep/analysis-grand-challenge/](https://iris-hep.org/analysis-grand-challenge/)
- *ROOT RDF* (Andrii Falko, Enrico Guiraud): [andriiknu/RDF/](https://andriiknu.github.io/RDF/)
- *Julia* (Jerry Ling): [Moelf/LHC-AGC.jl](https://moelf.github.io/LHC-AGC.jl)



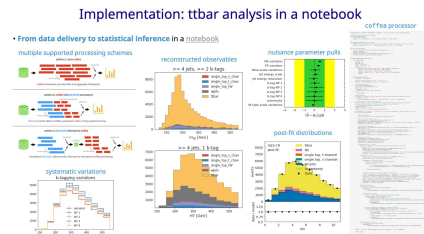


# AGC events

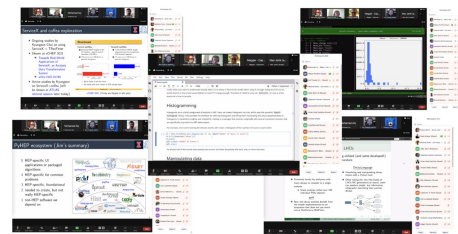
## Community building

- IRIS-HEP AGC Tools 2021 Workshop
  - 102 participants (~90 connected), 9 talks
- IRIS-HEP AGC Tools 2022 Workshop
  - 124 registration (~70 connected), 11 talks
- Co-organisers of HSF AF Forum kick off meeting
- Actively involved in (and co-organisers of) AE2 workshop
- Monthly AGC meetings including US ATLAS / CMS Operations programs
- Stay in touch: [analysis-grand-challenge@iris-hep.org](mailto:analysis-grand-challenge@iris-hep.org) ([sign up here](#))

### ICHEP talk



### AGC tools 2022 workshop



# Plans

Year 5 and beyond

# AGC timeline

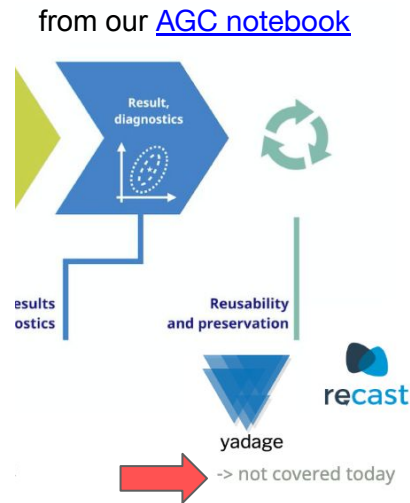
## Key high-level checkpoints

- **Nov 2021:** first **demonstration of toolchain** at AGC workshop ([agenda](#))
- **April 2022:** second iteration of **AGC workshop**, new ttbar analysis pipeline ([agenda](#))
- **Summer 2022:** **benchmarking of system components** in the AGC context
  - Results to be shown at ACAT 2022
- **Spring 2023:** *execution of AGC at full scale*

# AGC components: Year 5 +

## Re-use/preservation & ML Component

- **Analysis re-use / re-interpretation / preservation**
  - Need to develop AGC perspective / solution
  - Minimally: exploring data preservation pipeline (e.g. REANA)
  - Stretch goal: explore user-friendly solutions (AS connections)
- Need to add **ML component** (e.g. ttbar reconstruction)
  - Frequently requested & used in practice
  - How to *efficiently* integrate into pipeline?



# AGC components: Year 5 +

## Stretch goal to improve performance

- Better **integration** of *coffea* & *ServiceX* (could be a **Year 5** goal)
- Integrate **Skyhook** into pipeline
  - Will require to integrate it with *coffea* & *ServiceX*

*We would benefit from having dedicated people from AS/DOMA teams to help us to follow up integration issues*

# AGC pipelines: Year 5 +

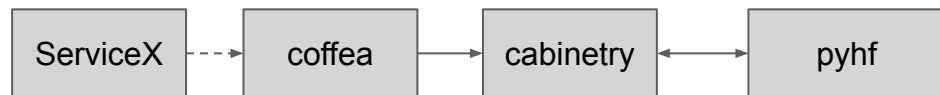
## Run AGC pipeline with ATLAS & CMS data

- **AGC analysis versions with CMS NanoAODs / ATLAS PHYSLITE inputs**
  - *May help adoption of analysis tools in experiments*
    - Useful to evaluate potential experiment-specific issues
  - Possibility to run on larger amount of data
    - For example: CMS has produced 876 TB of NanoAOD in Run 2 ([slides](#))

*Would benefit from dedicated ATLAS / CMS postdoc(s) evaluating AGC pipeline in experiment-specific use cases.*

# AGC pipelines: Year 5 +

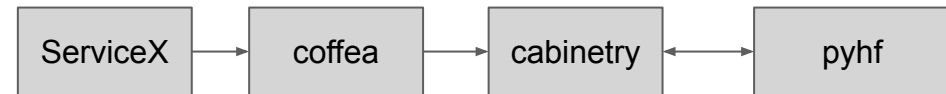
## Stretch goal: ATLAS and CMS specific pipelines



Columns from NanoAOD or flat ntuples from ServiceX or directly through coffea

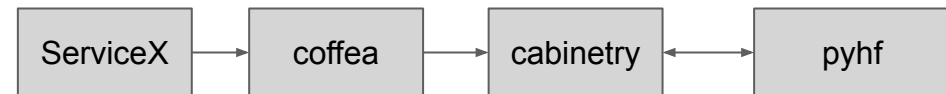
*Generic analysis pipeline based on Open Data dataset allowing to easily port AGC to other analysis frameworks*

This what we have now!



Columns from NanoAOD and **request column from MiniAOD** (only ServiceX)

***CMS specific analysis pipeline** based on Open Data datasets and **CMS datasets***



Columns from PHYSLITE and **request column from PHYS** (only ServiceX)

***ATLAS specific analysis pipeline** based on **ATLAS datasets***

# AGC pipeline: Year 5 +

Stretch goal: differentiable pipeline

- Not much progress recently
  - Had dedicated meeting on [Differentiable Programming for the Analysis Grand Challenge](#)
    - Purpose of this meeting was to finalize [planning document](#) to complete the IRIS-HEP milestone G2.9 *Differentiable programming roadmap across services needed for analysis challenge.*

*We would benefit from having dedicated people from AS team to help us to develop and integrate pipeline*



# Discussion points

# Discussion points and goals

- **Goal of discussion:** concrete list of action items for year 5
  - Including identified leads for individual items
- **Which of the projects mentioned can we achieve in Y5?**
  - Minimal scenario / stretch goals / ideal scenario
  - Create **new associated milestones** for focus areas?
- **-> perspectives from focus areas**

# Year 5: minimal / stretch / ideal scenario

## Minimal / ideal AGC setup for year 5

- **Minimal scenario**

- AGC showcase event in spring 2023 together with AF partners
- Extensive documentation for analysis task to allow re-implementations

- **Stretch goals**

- Performance tuning (DOMA/AS) / extended input size
- AGC in REANA / related work on re-use and re-interpretations (AS)

- **Ideal scenario**

- ML component
  - Including exploring GPU integration at AFs into pipeline
- Processing implementation improvements (systematics handling, use of *correctionlib*)
- Improved ServiceX + coffea integration
- Analysis extension: search for BSM signal in same phase space
- ATLAS / CMS-specific versions

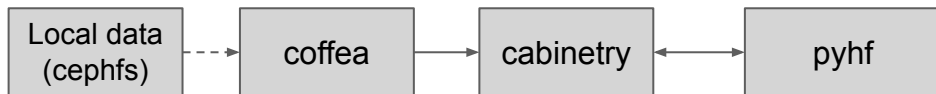
# Backup

# AGC: performance

*We need help from package developers to profile AGC setups*

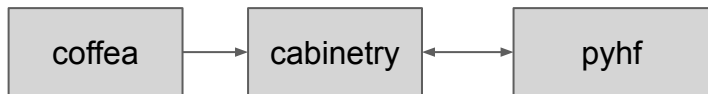
## Test “combinatorics” to investigate bottlenecks

WIP: results expected to be shown on [ACAT 2022](#)



Columns from NanoAOD or flat ntuples  
processed by coffea

Starting out with plain coffea, reading inputs  
through locally stored files



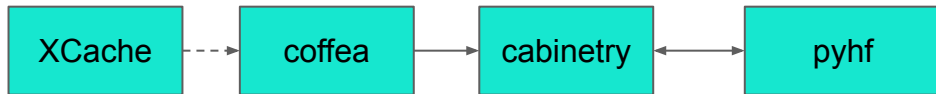
Columns from NanoAOD or flat ntuples  
processed by coffea

Starting out with plain coffea, reading inputs  
available on grid

# AGC: performance

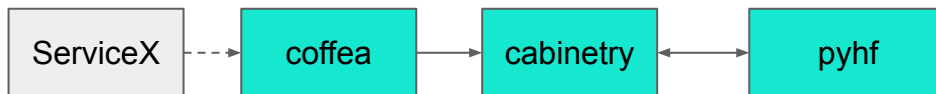
## “Combinatorics” to improve speedup

*We need help from package developers to profile AGC setups*



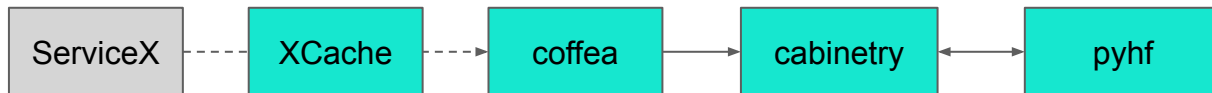
Columns from NanoAOD or flat ntuples  
processed by coffea

Starting out with plain coffea,  
reading inputs through XCache from  
remote place/grid (fast XCache  
deployed as a part of facility)



Columns from NanoAOD or flat ntuples  
processed by coffea

Two possible configurations:  
Starting out with ServiceX (could be  
remote) and coffea ServiceX  
executor adding XCache as a part of  
pipeline available directly on facility



Columns from NanoAOD or flat ntuples  
processed by coffea