



Analysis Tools in the 2020's and 2030's

G. WATIS (UW/SEATTLE)



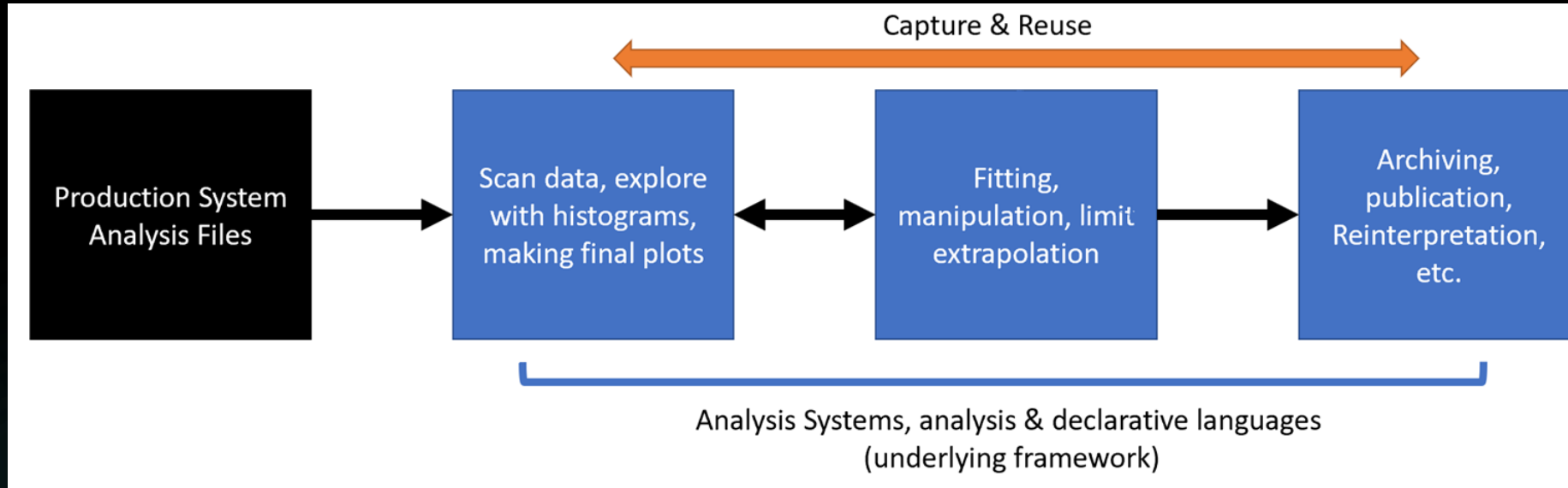
2020's

2030's



What Do We Need To Do?

5



- Build Histograms
- Cut & Count datasets
- Craft good-looking histograms
- Fitting of histograms to complex fractions
- Systematic Error propagation and studies
- Building of complex signal and background models with control regions
- Machine Learning Training & use of ML fits in event selection
- Reuse of an existing analysis w/new Signal
- Archiving of results (HEPData, etc.)

A dark, textured sphere, resembling a planet or moon, is the central focus. It is surrounded by numerous glowing blue lines that curve and intersect around it, creating a sense of motion or orbits. Several text labels are scattered throughout the scene, including "CH 83-0618" in the upper left, "BC 7A-8224" near the sphere, "JC 11-0825" in the upper right, "AC 88-8203" in the lower right, and "TRAFFIC NO. 23" in the bottom left corner. The background is a deep black with faint, concentric circular lines.

The Future We Know

Contents

[arxiv:1804.03983](https://arxiv.org/abs/1804.03983)

Data Analysis

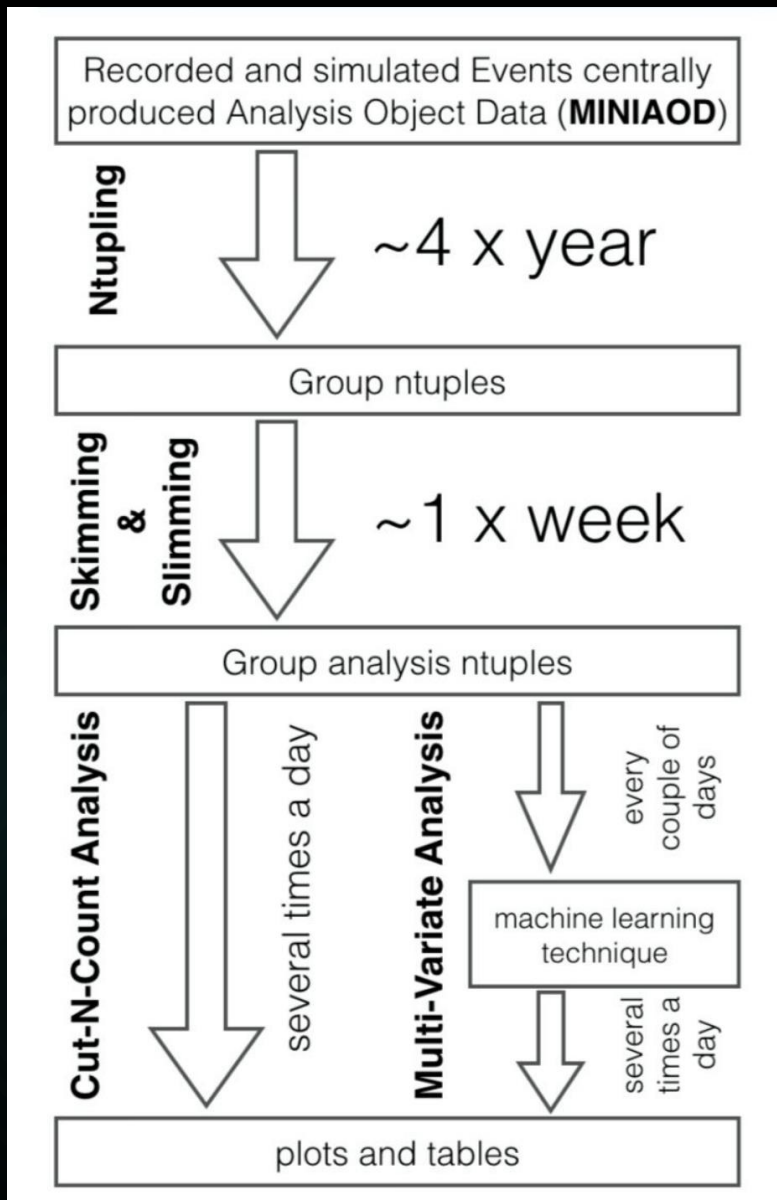
1	Introduction	2
2	HEP Analysis Software Ecosystem	3
3	Analysis Languages	4
3.1	Python	4
3.2	Declarative Languages	4
4	Analyzing Data	5
4.1	File format and compression algorithms	5
4.2	Analysis Facilities	5
4.3	Non-event data handling	6
5	Analysis Models and Future Systems	7
5.1	Sequential Ntuple Reduction	7
5.2	“Spark”-like Analysis	7
5.3	Query-based Analysis	8
6	Analysis Preservation	10
7	Analysis Interpretation	11
8	Analysis Roadmap	12
8.1	1-year Time Frame	12
8.2	3-year Time Frame	13
8.3	5-year Time Frame	13

3 Meeting the Challenges

[arxiv:1712.07959](https://arxiv.org/abs/1712.07959)

Software Sustainability

3	Meeting the Challenges	6
3.1	Effective use of developer effort	6
3.1.1	Design documentation and training	6
3.1.2	Standard development tools	7
3.1.3	Development environment	8
3.1.4	Social coding sites	9
3.1.5	Continuous integration	9
3.1.6	Code review	9
3.1.7	Agile development	9
3.2	Knowledge sharing	11
3.2.1	Documentation	11
3.2.2	Training	11
3.2.3	Seminars and discussion forums	11
3.2.4	Workshops and community reviews	12
3.2.5	Conferences	12
3.2.6	Journal publication	13
3.3	Reusing code effectively	13
3.3.1	Software Licenses	13
3.4	Validation of complex outputs	14
3.5	Data and code preservation	14
3.6	Code deprecation	15
3.7	Hardware and software evolution	15
3.7.1	Effective exploitation of parallel hardware	15
3.7.2	Achieving performance portability	16
3.7.3	Tracking of new software technologies and tools	17
3.7.4	Dealing with hardware diversity	17
3.8	Software deployment	17
3.9	Recognition of software development	18



Example sample sizes:

		MC16e	data18
AOD	logical [PB]	11.2	2.7
	disk [PB]	13.0	4.2
	evt [10^9]	17.178	12.108
DAOD	logical [PB]	9.9	6.1
	disk [PB]	13.4	12.7
	evt [10^9]	91.292	110.139

Top 10 DAOD:

DAOD_TOPQ1	10.10 PB
DAOD_STDM4	3.57 PB
DAOD_TOPQ4	3.40 PB
DAOD_FTAG4	3.27 PB
DAOD_RPVLL	3.10 PB
DAOD_HIGG2D1	2.41 PB
DAOD_JETM6	2.08 PB
DAOD_FTAG1	1.98 PB
DAOD_JETM1	1.97 PB
DAOD_EXOT5	1.80 PB

Run3: Initial assumption resources will be: $1.5 \times$ (resources in 2018) Consistent with "flat budget"

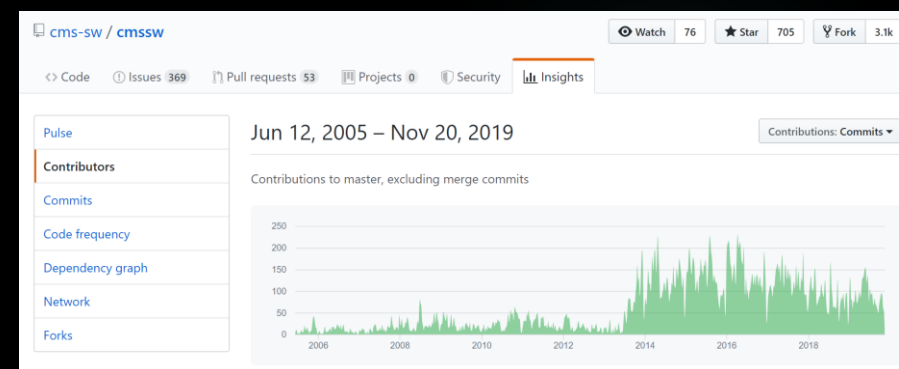
So Run 3 isn't a huge problem...

VERY SIMPLE HL-LHC EXTRAPOLATION FOR DISK

	MC			Data			Sum
	AOD	DAOD	DAOD PHYSLITE	AOD	DAOD	DAOD PHYSLITE	
events (25-28)	$6.4 \cdot 10^{11}$			$1.5 \cdot 10^{11}$			
events / year	$2.13 \cdot 10^{11}$	$1.07 \cdot 10^{12}$	$2.13 \cdot 10^{11}$	$5.0 \cdot 10^{10}$	$2.5 \cdot 10^{11}$	$5.0 \cdot 10^{10}$	
size/event [kB]	1000	100	10	700	50	10	
disk [PB/year]	213.3	106.7	2.1	35.0	12.5	0.5	369.6

This will be a problem, however...

(Taken from Johannes' CHEP talk)





PYTORCH



A person is seated in a futuristic, circular structure that glows with a bright blue light. The person is wearing a dark, form-fitting suit that appears to be made of a material that reflects or emits light, giving it a shimmering, metallic appearance. They are holding a small, glowing object in their hands. The background is dark, and the overall atmosphere is high-tech and futuristic.

What we'd
like in 2026...

All those tools to talk to each other!

The Different ecosystems living comfortably with each other



Time between “Make me a plot of $Z \rightarrow ee$ ” till
your new student gives you the plot measured in
days, not weeks.

Your quick plotting methods are easily
convertible to production quality analysis that is
reusable. Same framework.

Bridges & Ferries

15



Take the best advantage of all the tools out there



Awkward
Array

uproot

RDataFrame can
convert to numpy
arrays, read arrow

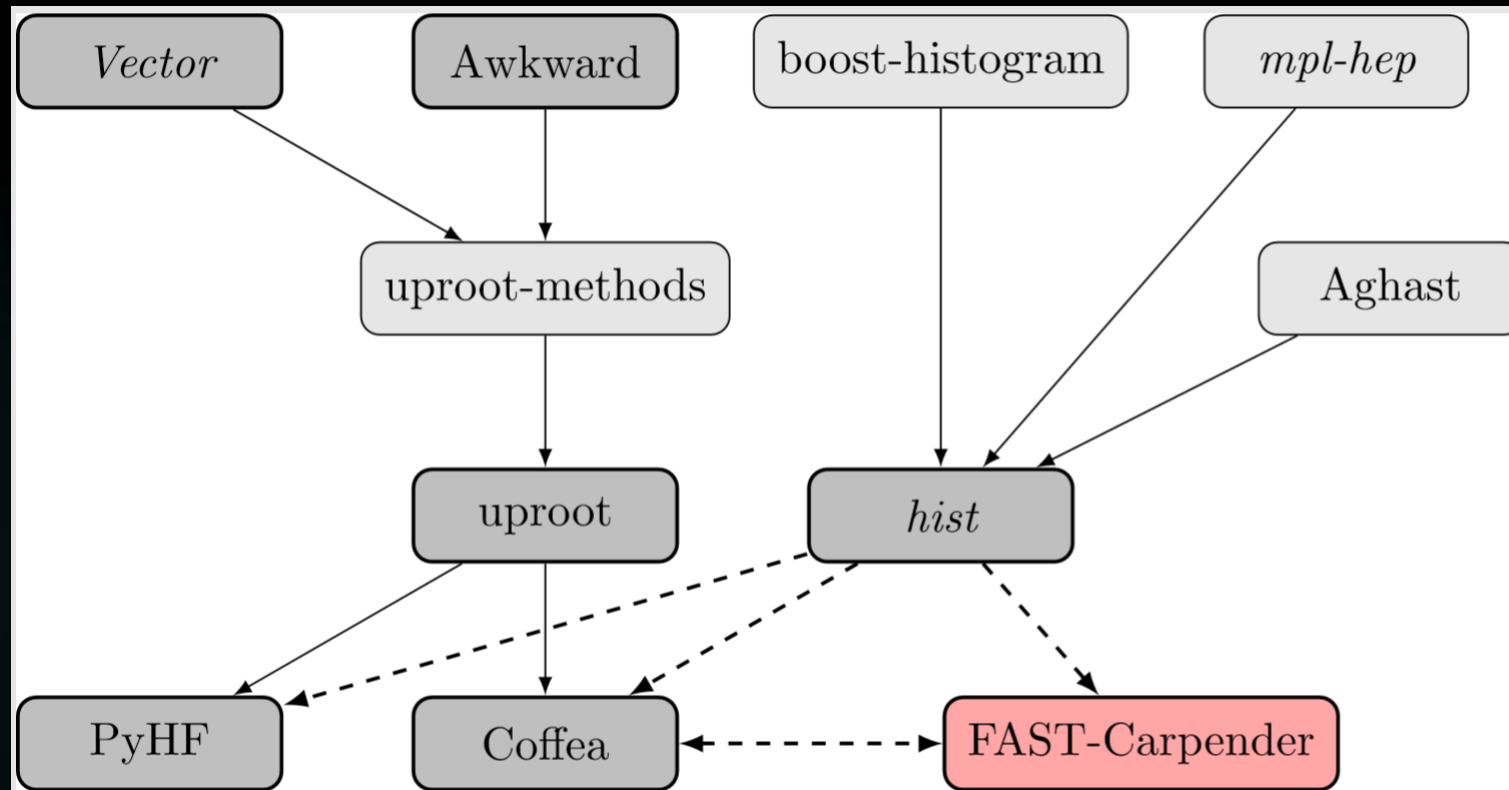


We have to get away from
all the hand-art our analysis
currently requires!

- Training to C++?
- New ROOT NTuple

Histograms in the Python Ecosystem

16



Analysis Facility

17



Can I get all **electrons** from the
`mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.merge.DA
OD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00`
dataset, and plot the $Z \rightarrow e^+e^-$ mass?

- The dataset is an ATLAS xAOD dataset (standard analysis format)
 - Not a flat ntuple
- It is stored in the GRID (as all ATLAS analysis xAOD's are)

BTW, make this
work for ATLAS,
CMS, LHCb, etc...
(using their own
idioms)



Why Not This?

18



Fetching the data

[From a Jupyter Notebook \(on GitHub\)](#)

```
In [6]: ds = EventDataset('localds://mc15_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L1_Zee.merge.DAOD_STDM3.e3601_s2576_s2132_r6630_r6264_p2363_tid05630052_00')
```

```
In [7]: leptons_per_event_as = ds \
        .Select('lambda e: (e.Electrons("Electrons"), e.Muons("Muons"))') \
        .Select('lambda ls: (ls[0].Select(lambda e: e.pt()), ls[0].Select(lambda e: e.eta()), ls[0].Select(
        lambda e: e.phi()), ls[0].Select(lambda e: e.e()), ls[1].Select(lambda m: m.pt()), ls[1].Select(lambda m:
        m.eta()), ls[1].Select(lambda m: m.phi()), ls[1].Select(lambda m: m.e()))') \
        .AsAwkwardArray(('ElePt', 'EleEta', 'ElePhi', 'EleE', 'MuPt', 'MuEta', 'MuPhi', 'MuE')) \
        .future_value(executor=lambda a: use_exe_func_adl_server(a, node=end_point, quiet=False))
```

```
In [8]: leptons_per_event = await leptons_per_event_as
```

Files that were returned:

```
['file:///C:/Users/gordo/Documents/func-adl-cache/40ae9bb8bd6cf8bcb7ae703c715939d7/ANALYSIS_001.root', 'pandas_tree28']
```

```
In [15]: v_particles = uproot_methods.TLorentzVectorArray.from_ptetaphi(
        leptons_per_event[b'ElePt'], leptons_per_event[b'EleEta'],
        leptons_per_event[b'ElePhi'], leptons_per_event[b'EleE'],
        )
```

```
In [17]: v_particles = v_particles[v_particles.counts >= 2]
        diparticles = v_particles[:, 0] + v_particles[:, 1]
```

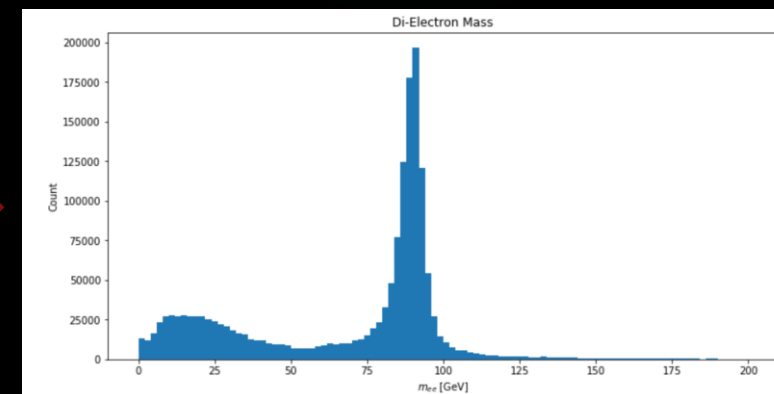
```
In [39]: plt.figure(figsize=(12, 6))
        plt.hist(diparticles.mass/1000.0, bins=100, range=(0,200))
        plt.title('Di-Electron Mass')
        plt.xlabel('$m_{ee}$ [GeV]')
        plt.ylabel('Count')
        plt.show()
```

Dataset

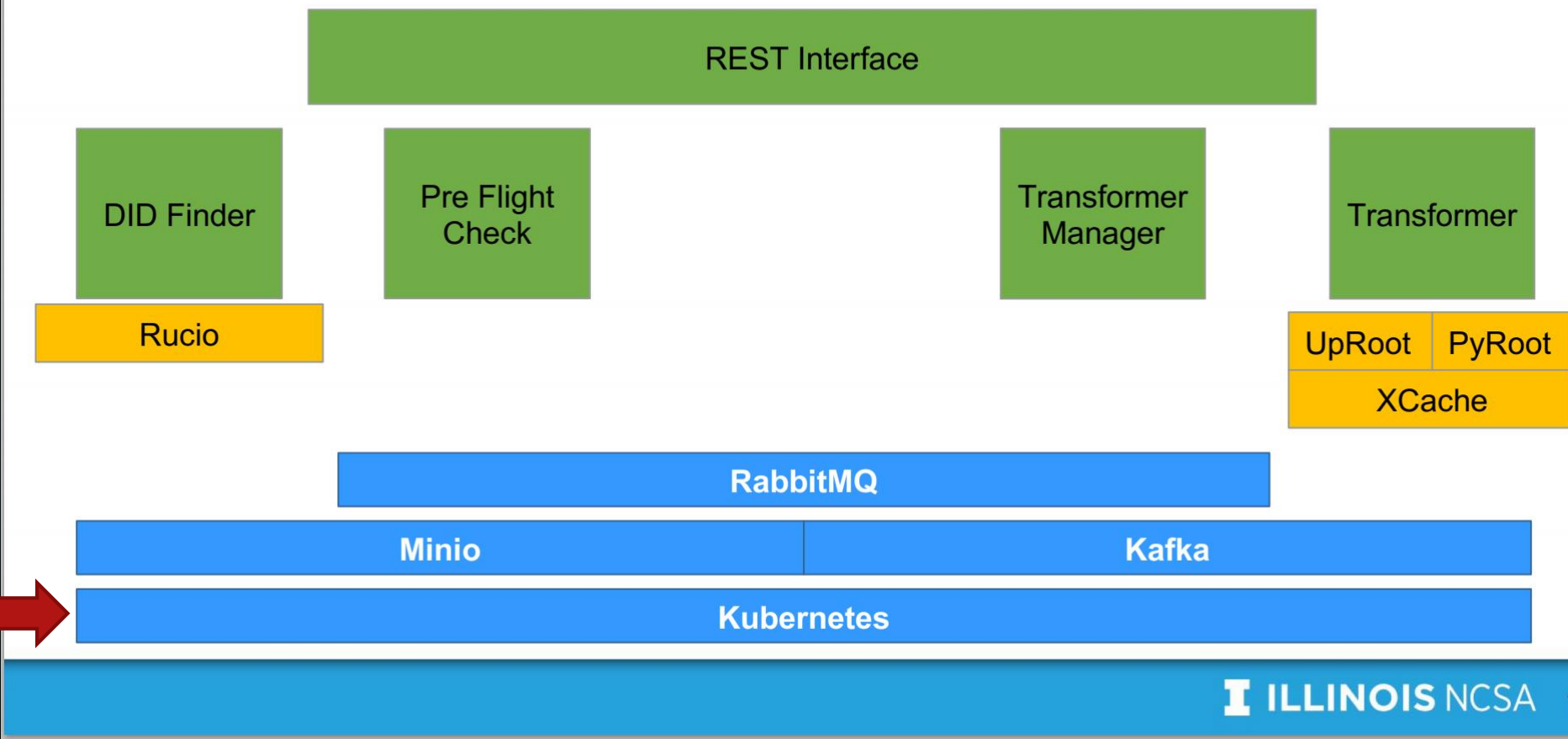
Fetch the data for electrons

Invariant Mass

Plot



Service Architecture V1



This effort is just getting started...

Reproducible Analyses

21



Describe

```
...
steps:
  analyse_data:
    run: analyse_data.cwl
    hints:
      reana:
        compute_backend: slurmcern
    out: [DoubleMuParked2012C_10000_Higgs.root]
  analyse_mc:
    run: analyse_mc.cwl
    hints:
      reana:
        compute_backend: htcondorcern
    out: [Higgs4L1file.root]
  make_plot:
    run: make_plot.cwl
    hints:
      reana:
        compute_backend: kubernetes
in:
  DoubleMuParked2012C_10000_Higgs: >
    analyse_data/DoubleMuParked2012C_10000_Higgs.root
  Higgs4L1file: >
    analyse_mc/Higgs4L1file.root
out: [mass4l_combine_userlvl3.pdf]
...
```



Extensible

Do you have an institutional HPC system? Plug it easily ..



... depending on the compute cluster:



Python bindings
+ Kerberos



Python API client +
token based auth



SSH to headnode



How to connect to
your HPC cluster?

Can we build out of well tested parts?

22



Flexible

Run many computational workflow engines.



Scalable

Support for remote compute clouds.



Reusable

Containerise once, reuse elsewhere. Cloud-native.



Free

Free Software. MIT licence. Made with ❤️ at CERN.



End Game

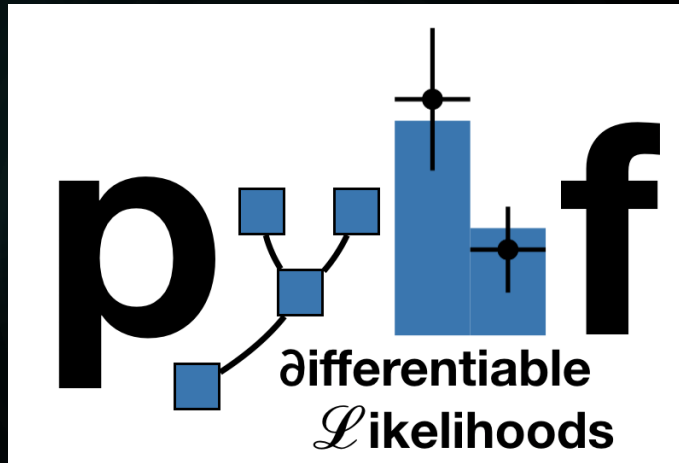
23



Once we have a story for getting our distributions and ML out of the data...



How do we complete the analysis?
And in a way that works with the other components?



RooStats → TF/PyTorch fitting
10 hours → 30 minutes

TRExFitter

High level profile
fitting language

How do we put together an ecosystem of tools that work across languages and experiments?

Proper Role For Jupyter?

24



Tutorials and Training

Quick Studies and cross checks

Full Analysis end-to-end



Conclusions

25

- ▶ I have missed a lot!!
 - ▶ ML MC tools I mentioned yesterday, for example
 - ▶ To get ideas of other projects, see the [Analysis Systems home page](#)
 - ▶ See the white papers for some of the things that were discussed
 - ▶ [Analysis eco-system workshop at AMS](#)
- ▶ Building Software from Existing Components
 - ▶ E.g. Distributed computing is the lifeblood of the real world (Netflix, Microsoft, Amazon, Apple, etc.)
 - ▶ Should we ever write the plumbing again?
 - ▶ Open Source Package managers make a lot of people's work available for us, and us to make it available to them.
- ▶ Building a coherent ecosystem
 - ▶ This is just getting started...
 - ▶ And end-to-end analysis in a coherent way (CI??).