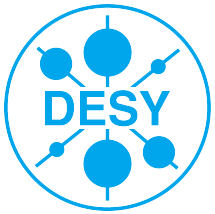
MC16 Dataset Nomenclature

The single tag container model employed in MC Production

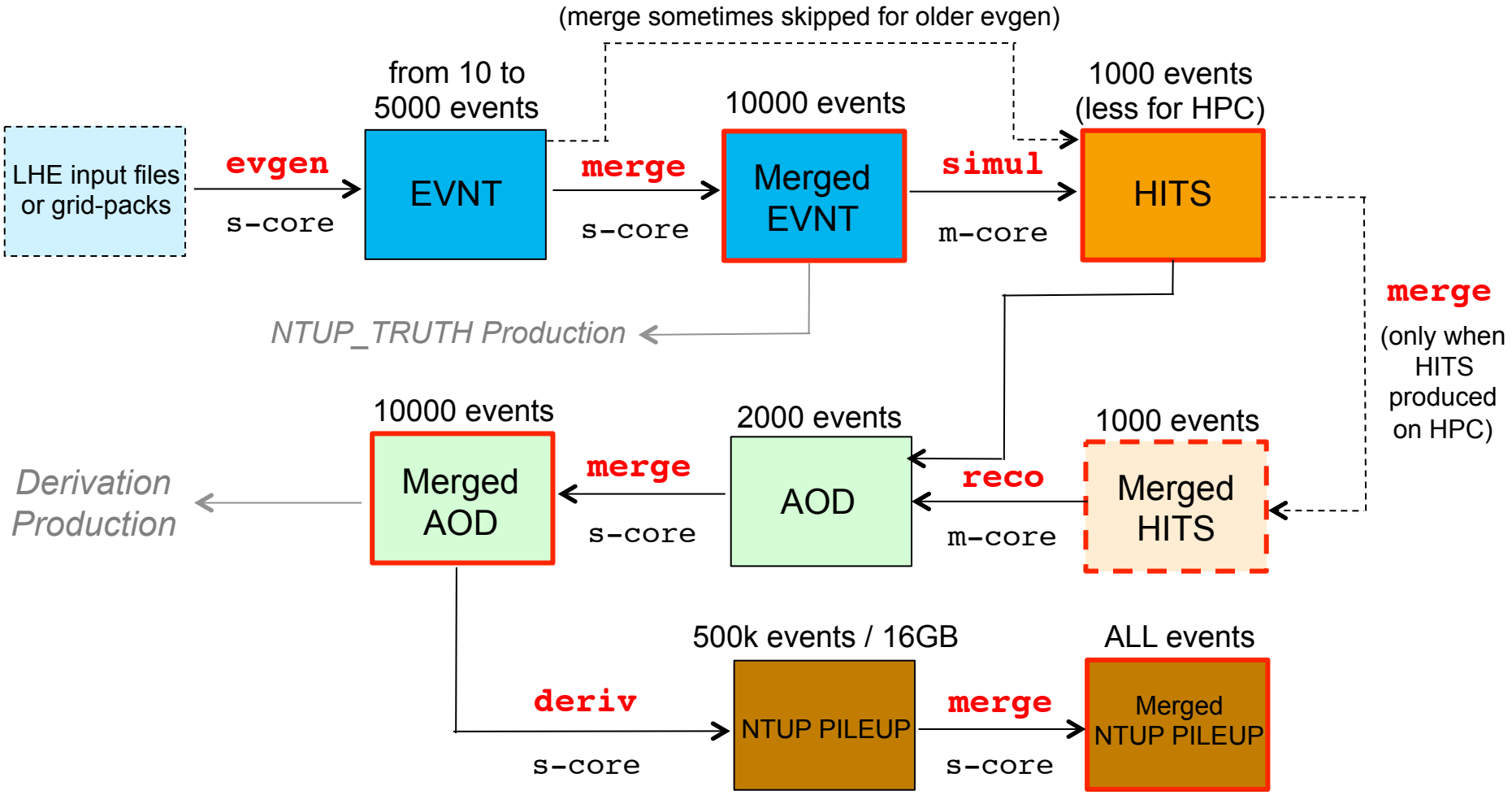
- Explain the rationale behind the model
- Try to answer points raised in the recent mail thread
- Suggest some future working directions

David South (DESY)
Dominic Hirschbühl (Uni. Wuppertal)

ATLAS Software and Computing Week
Data Characterisation and Curation
December 13th, 2018



Current MC production chain



Keep only the (merged) dataset at each step, according to the relevant rules of the lifetime model for each data type

Nomenclature, production steps and formats

- MC Production is wide reaching in the ATLAS computing infrastructure, influencing and fully integrated with ProdSys, rucio, AMI and beyond

- ATLAS dataset nomenclature is defined in (an evolving) document https://cds.cern.ch/record/1070318, section 6 which defines MC dataset names
  project.datasetNumber.physicsShort.prodStep.dataType.AMITag[_tidnnnnnn[_SS]

- In particular section 6.1.3 describes at length the well defined options and names for the **prodStep** and the associations with e.g. the ami tags used

- The production steps used in MC Prod are: **evgen**, **simul**, **recon**, **deriv** where deriv in our case refers only to the NTUP_PILEUP format

- **merge** steps are also described, decision was taken in ATLAS that dataset containers used for analysis, or as input to analysis, should no longer contain "merge" but should have the production step associated to that format
  – e.g. "recon" for AOD

Single-tag containers

- The development of the single tag container model for MC16 was almost inevitable for MC Production, due to:
  - *The variation in the number of merge tags in evgen, from using older HITS productions created before evgen was universally merged – thankfully now much rarer*
  - *The variation in the number of merge tags in simulation, where for HPC the lower number of events/job requires an additional merge step*
  - *Extending samples, where original and extension may involve alternative workflows*
  - *And to some extent, now having tids from multiple sub-campaigns within one container*
- The single-tag concept is applied to evgen, simul, recon and deriv containers
- First concerning evgen and simulation:

`evgen.EVNT.e`
containers have (unique) tids of the format:

`evgen.EVNT.e`
`merge.EVNT.e_e`

`simul.HITS.e_s`
containers have (unique) tids of the format:

`simul.HITS.e_s`
`merge.HITS.e_s_s`
`simul.HITS.e_e_s`
`merge.HITS.e_e_s_s`

An example of a varied HITS container

mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126

Dataset Name	Events Number	SubCampaing	Tasks
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_e5984_s3126_tid12196360_00	1489400	MC16:MC16c	13038485
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_e5984_s3126_tid12592858_00	7431200	MC16:MC16e	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_e5984_s3126_tid13866273_00	2510000	MC16:MC16e	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid10730514_00	1995000	MC16:MC16a	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid10944971_00	3986600	MC16:MC16a	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid11324488_00	5981200	MC16:MC16c	13038493

- Here we have:
  - *Two MC16a tids, where the evgen was not merged, so only one e-tag*
  - *Two MC16c tids (highlighted), one with an evgen merge (two e-tags) and one without*
  - *Two MC16e tids, both with evgen merge steps*

- This is extremely useful for production to collect various tid formats together

- A more complicated example is in the back-up slides

Single tag containers: Reco and beyond

- For merged AOD containers in MC16, the following patterns are therefore relevant (tags detailed on twiki, e.g https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/AtlasProductionGroupMC16e)
  - MC16a: **recon.AOD.e*_s3126_r9364**
    MC16d: **recon.AOD.e*_s3126_r10201**
    MC16e: **recon.AOD.e*_s3126_r10724** (and for AF2, replace **s3126** with **a875**)
  - *Using same DSID again: MC16d AOD container has two tids (1 single e-tag, 1 double e-tag)*:

mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.recon.AOD.e5271_s3126_r10201

mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.merge.AOD.e5271_e5984_s3126_r10201_r10210_tid13038487_00
mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.merge.AOD.e5271_s3126_r10201_r10210_tid13038495_00

- Some additional notes before wrapping up single-tag containers:
  - *The content of **recon.AOD** and **deriv.NTUP_PILEUP** containers is simpler, as we always merge these formats: they only ever have "merge" tids after all merges are done*
  - *Derivations use internal merging, so all tids in other **deriv.NTUPXXX** containers are "deriv"*
  - *Data produces many formats from RAW, so in that case it's not just **recon.AOD** but also **recon.DRAW_RPVLL** etc, so both production step and format are important*

The input container for derivation productions

1) People should not have to care about tids, only containers
2) The tids in one container are unique, i.e. there is no double counting
3) The idea is to have **one** container for all types of tids in one sample, with and without e/s merges
4) Merges are really not very interesting, so the only tags of significance are evgen, simul, recon
5) Therefore the production step for the container should be evgen, simul, recon and there is one container per sample (i.e. per DSID)
6) The rule for the derivation input container nomenclature is simple:

"`recon.AOD.e_s_r`"
Production step: recon
One tag of each type

Now to the problems and issues

- The position of ProdSys, ADC and MCProd is that this is the well established model used in production and it would be very disruptive to change this again
- However, we do indeed want to discuss issues raised, fix problems and naturally also save CPU and person power. Parallel solutions are possible

- There are two issues identified with this model, which may make things difficult when looking for inputs to derivation production
  - *(1) A simple "**`rucio ls recon.AOD.e*_sXXXX_rYYYY`**", may pick up potential production containers of the form **`recon.AOD.e_e_s_r`***
    *Note the **`e*`** allows for different e-tags*
  - *(2) For older samples, there may be production containers of the form **`recon.AOD.e_s_r`** which are intermediately filled with unmerged AOD tid datasets*

- I try to address these, and present some suggestions about moving forward

Real example using MC16d Pythia JZ*W mufilter samples

```
rucio ls --short
"mc16_13TeV.*.Pythia*W_mufilter.recon.AOD.e*_s3126_r10201" | sort
```

```
mc16_13TeV:mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_s3126_r10201
mc16_13TeV:mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427004.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4W_mufilter.recon.AOD.e5660_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427004.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427005.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5W_mufilter.recon.AOD.e5660_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427005.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427106.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ6W_mufilter.recon.AOD.e5839_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427106.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ6W_mufilter.recon.AOD.e5839_s3126_r10201
mc16_13TeV:mc16_13TeV.427107.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W_mufilter.recon.AOD.e5839_e5984_s3126_r10201
mc16_13TeV:mc16_13TeV.427107.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W_mufilter.recon.AOD.e5839_s3126_r10201
```

- The standard "**`rucio ls`**" does pick up the double e-tag containers here
- Note the group of data sets here includes different evgen tags; if all evgen tags had been e.g. **`e5839`**, then trivial to pick out only the single tag containers:

```
rucio ls --short
"mc16_13TeV.*.Pythia*W_mufilter.recon.AOD.e5839_s3126_r10201" | sort
```

Real example using MC16d Pythia JZ*W mufilter samples

```
rucio ls --short
"mc16_13TeV.*.Pythia*W_mufilter.recon.AOD.e*_s3126_r10201" | sort | grep -v "e*_e"
```

```
mc16_13TeV:mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_s3126_r10201
mc16_13TeV:mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427004.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427005.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427106.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ6W_mufilter.recon.AOD.e5839_s3126_r10201
mc16_13TeV:mc16_13TeV.427107.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W_mufilter.recon.AOD.e5839_s3126_r10201
```

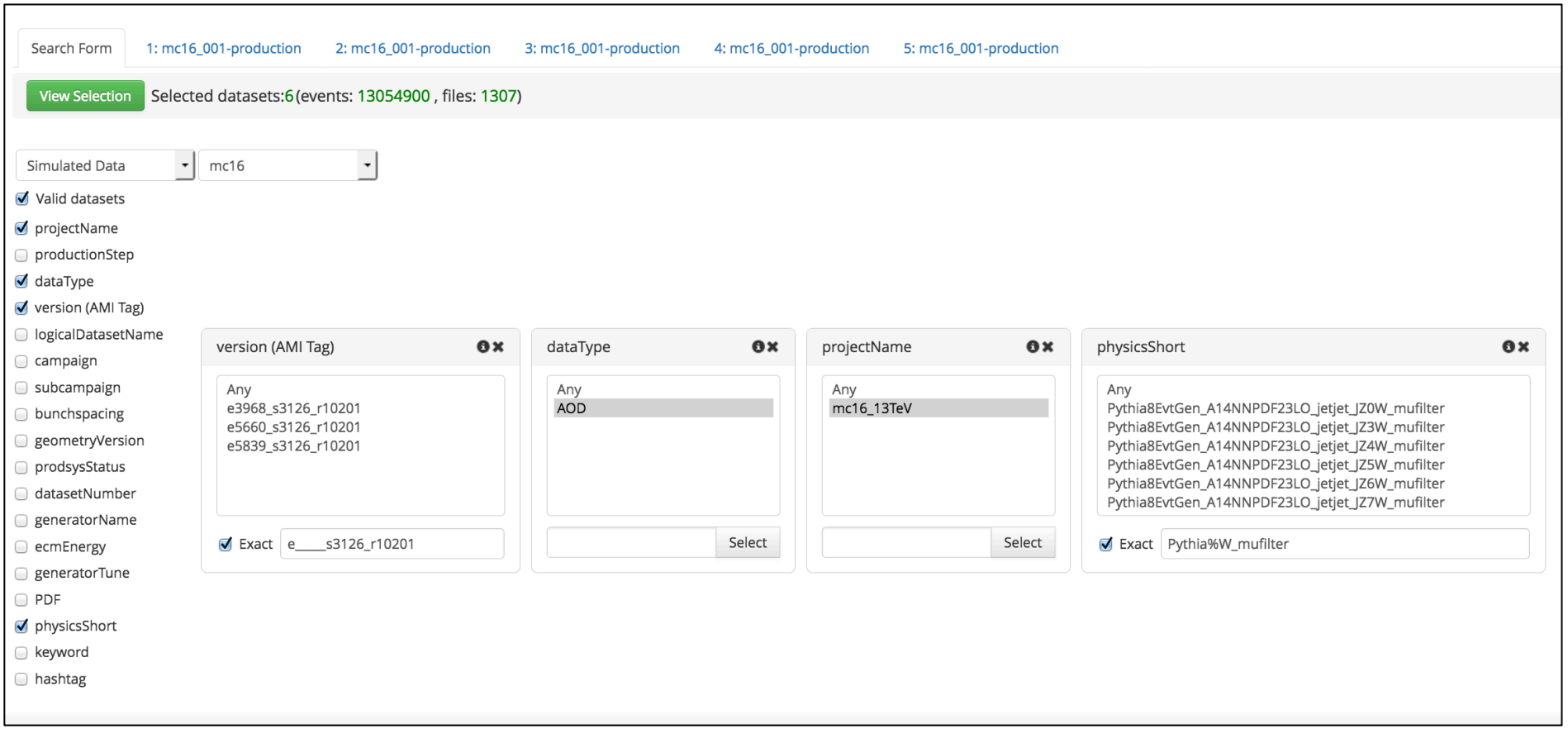
- However using (for example) grep it is fairly trivial to remove such containers - there are other solutions available

- The same is applies when using AMI to get lists of datasets:

```
ami list datasets --physics-short
"Pythia%W_mufilter" --ldn "mc16_13TeV.%.e%s3126_r10201" | sort | grep -v "e*_e"
```

- But what about using the AMI dataset browser?

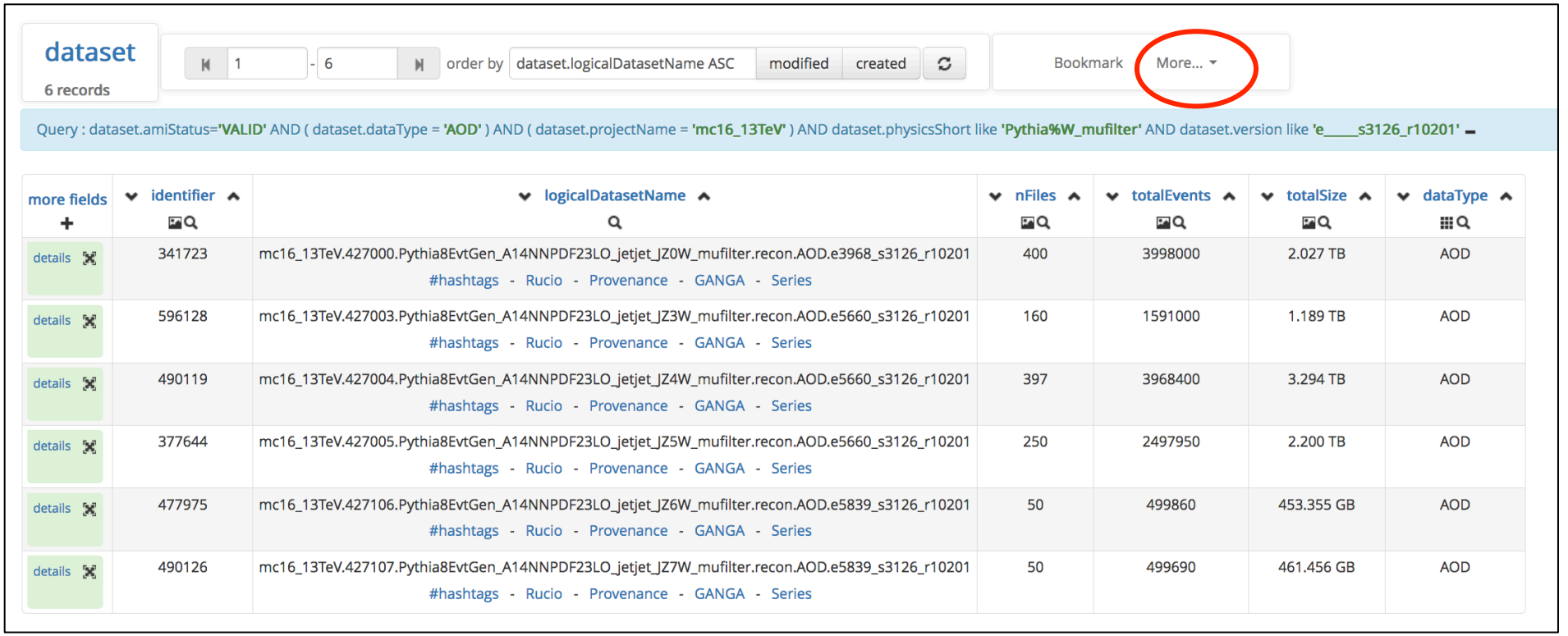
Real example using MC16d Pythia JZ*W mufilter samples




```
Query : dataset.amiStatus='VALID' AND ( dataset.dataType = 'AOD' ) AND
( dataset.projectName = 'mc16_13TeV' ) AND dataset.physicsShort like
'Pythia%W_mufilter' AND dataset.version like 'e____s3126_r10201'
```

5 underscores between e and s

Real example using MC16d Pythia JZ*W mufilter samples



dataset

6 records

1 - 6 order by dataset.logicalDatasetName ASC modified created Bookmark More...

Query : dataset.amiStatus='VALID' AND (dataset.dataType = 'AOD') AND (dataset.projectName = 'mc16_13TeV') AND dataset.physicsShort like 'Pythia%W_mufilter' AND dataset.version like 'e____s3126_r10201'

more fields	identifier	logicalDatasetName	nFiles	totalEvents	totalSize	dataType
details	341723	mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	400	3998000	2.027 TB	AOD
details	596128	mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	160	1591000	1.189 TB	AOD
details	490119	mc16_13TeV.427004.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4W_mufilter.recon.AOD.e5660_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	397	3968400	3.294 TB	AOD
details	377644	mc16_13TeV.427005.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5W_mufilter.recon.AOD.e5660_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	250	2497950	2.200 TB	AOD
details	477975	mc16_13TeV.427106.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ6W_mufilter.recon.AOD.e5839_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	50	499860	453.355 GB	AOD
details	490126	mc16_13TeV.427107.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W_mufilter.recon.AOD.e5839_s3126_r10201 #hashtags - Rucio - Provenance - GANGA - Series	50	499690	461.456 GB	AOD

- It works – the query finds the six datasets!
- AMI then provides an option to export this list as text, xml or csv under "More"

Real example using MC16d Pythia JZ*W mufilter samples

```
AMI#
command : AMIBrowseSQLQuery
time    : 2018-12-12  at 01:39:14 PM CET
query   : SELECT
dataset.identifier ,dataset.logicalDatasetName ,dataset.nFiles ,dataset.totalEvents ,dataset.totalSize ,dataset.dataType ,dataset.prodsysStatus ,dataset.completion
,dataset.ecmEnergy ,dataset.physicsComment ,dataset.PDF ,dataset.version ,dataset.AtlasRelease ,dataset.crossSection ,dataset.genFiltEff ,dataset.datasetNumber ,dat
aset.principalPhysicsGroup ,dataset.physicsShort ,dataset.requestedBy ,dataset.generatorName ,dataset.geometryVersion ,dataset.conditionsTag ,dataset.lastModified ,
dataset.created ,dataset.generatorTune ,dataset.amiStatus ,dataset.beamType ,dataset.productionStep ,dataset.projectName ,'mc16_001' as PROJECT,'production' as
PROCESS, 'dataset' as AMIENTITYNAME, dataset.identifier as AMIELEMENTID ,TO_CHAR(dataset.created,'yyyy-mm-dd hh24:mi:ss') as AMICREATED ,
TO_CHAR(dataset.lastModified,'yyyy-mm-dd hh24:mi:ss') as AMILASTMODIFIED , TO_CHAR(CURRENT_TIMESTAMP - INTERVAL '48' HOUR,'yyyy-mm-dd hh24:mi:ss') as AMISYSDATE
FROM dataset WHERE ( ( ( ( ( (dataset.identifier IN (SELECT dataset.identifier FROM dataset WHERE dataset.amiStatus = 'VALID') AND dataset.identifier IN (SELECT
dataset.identifier FROM dataset WHERE dataset.dataType = 'AOD')) AND dataset.identifier IN (SELECT dataset.identifier FROM dataset WHERE dataset.projectName =
'mc16_13TeV')) AND dataset.identifier IN (SELECT dataset.identifier FROM campaign,dataset WHERE campaign.subcampaign = 'MC16d' AND dataset.identifier =
campaign.datasetFK)) AND dataset.identifier IN (SELECT dataset.identifier FROM dataset WHERE dataset.physicsShort LIKE 'Pythia%W_mufilter')) AND dataset.identifier
IN (SELECT dataset.identifier FROM dataset WHERE dataset.version LIKE 'e_____s3126_r10201')) ) ORDER BY dataset.logicalDatasetName ASC LIMIT 0,15
result  :
  -> rowset Element_Info
    -> row 1
      -> identifier = 341723
      -> logicalDatasetName = mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_s3126_r10201
      -> nFiles = 400
      -> totalEvents = 3998000
      -> totalSize = 2027150786785
      -> dataType = AOD
      -> prodsysStatus = EVENTS PARTIALLY AVAILABLE
      -> completion = 100.00
      -> ecmEnergy = 13000000
      -> physicsComment =
      -> PDF =
      -> version = e3968_s3126_r10201
      -> AtlasRelease = Athena_21.0.53
      -> crossSection = 78420000
      -> genFiltEff = 0.00044719
      -> datasetNumber = 427000
      -> principalPhysicsGroup = gen-user
      -> physicsShort = Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter
      -> requestedBy = dsouth
      -> generatorName = Pythia8(v8.186)+EvtGen(v1.2.0)
      -> geometryVersion = ATLAS-R2-2016-01-00-01
      -> conditionsTag = OFLCOND-MC16-SDR-20
      -> lastModified = 2018-05-15 13:27:22
      -> created = 2017-12-29 08:39:32
      -> generatorTune = A14 NNPDF23LO
      -> amiStatus = VALID
      -> beamType = collisions
      -> productionStep = recon
      -> projectName = mc16_13TeV
      -> AMICREATED = 2017-12-29 08:39:32
      -> AMILASTMODIFIED = 2018-05-15 13:27:22
      -> AMISYSDATE = 2018-12-10 13:41:12
    -> row 2
      -> identifier = 596128
      -> logicalDatasetName = mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_s3126_r10201
      -> nFiles = 160
      -> totalEvents = 1591000
```

- But what is really exported is the whole output of the query
- And we're back to grep (and a bit of cat and awk as well)..

Real example using MC16d Pythia JZ*W mufilter samples

- What would be really useful is for the AMI interface to provide an additional option to export just the list of dataset containers identified by the query. This is what is useful to the users:

```
mc16_13TeV:mc16_13TeV.427000.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ0W_mufilter.recon.AOD.e3968_s3126_r10201
mc16_13TeV:mc16_13TeV.427003.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ3W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427004.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ4W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427005.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ5W_mufilter.recon.AOD.e5660_s3126_r10201
mc16_13TeV:mc16_13TeV.427106.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ6W_mufilter.recon.AOD.e5839_s3126_r10201
mc16_13TeV:mc16_13TeV.427107.Pythia8EvtGen_A14NNPDF23LO_jetjet_JZ7W_mufilter.recon.AOD.e5839_s3126_r10201
```

- Then we could also imagine greater things, like a way to export a list of dataset container directly to ProdSys to create a request
- Or, using an AMI query as input to request creation in ProdSys ..

Addendum after talk: this option is already there!!
More – EXPORT – GANGA
gives you a file called "datasetList.txt"
Request to AMI that this "GANGA" label be renamed to "datasetList" ?

What about that other problem?

- For older MC16 samples where the evgen was not merged, there may be production containers of the form **`recon.AOD.e_s_r`** which are intermediately filled with the unmerged AOD tid datasets
- As an example: this container:

  `mc16_13TeV.364100.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV0_70_CVetoBVeto.recon.AOD.e5271_s3126_r9781`

  initially had this unmerged AOD tid:

  `mc16_13TeV.`
  `364100.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV0_70_CVetoBVeto.recon.AOD.e5271_s3126_r9781_tid11925041_00`

  which, once the AOD was merged, was replaced by this merged AOD tid:

  `mc16_13TeV.`
  `364100.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV0_70_CVetoBVeto.merge.AOD.e5271_s3126_r9781_r9778_tid11925045_00`

- Such cases are relatively rare, but if a derivation production starts on this input container before the AOD merge has completed, it may be a problem
- The solution for this is to only allow "**`merge.AOD`**" tids as input to derivations tasks – this was part of our original plan, but was not yet implemented

Summary of single-tag containers

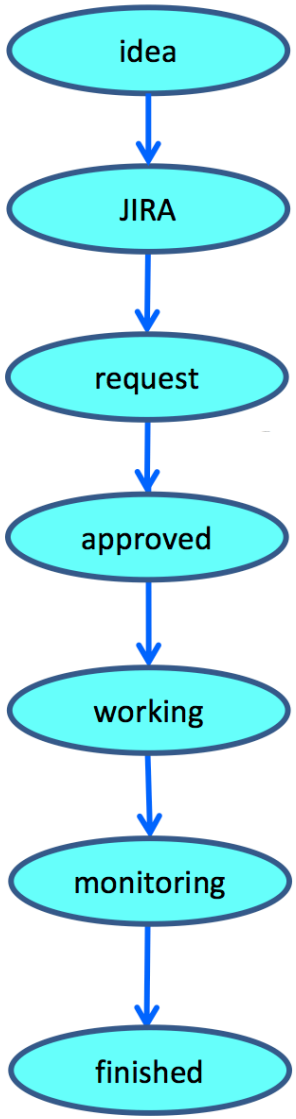
- Single tag container model is well established in the MC Production system and many components rely on the strict nomenclature rules introduced
- The rule for the derivation input container nomenclature is simple: "**`recon.AOD.e_s_r`**", i.e. production step: recon, one tag of each type
- Although the AOD level is somewhat simpler than the previous steps, there are nevertheless complications arriving at a list of datasets
- There are ways to remove unwanted entries in such lists when created on the command line by rucio or AMI
- Suggest expanding functionality of AMI dataset browser to export lists of datasets and to explore automatic creation of derivation tasks in ProdSys
- Additionally, ProdSys should only allow "**`merge.AOD`**" tids as input to derivations tasks

Beyond what we have, some suggestions

- Make new additional containers, with identical tid content to the **`recon.AOD.e_s_r`**, called for example "**`final.AOD.e_s_r`**"
  - *The new **`prodStep`** would need to be agreed by (probably) OAB, nomenclature doc etc*
  - *Would require regular syncing to deal with new sub-campaigns, productions (e.g. extensions)*
- Could also remove all tags from the container name and have something like "**`recon.AOD.MC16eFS`**" (or **`final`**), based on the sub-campaign metadata
  - *You would need the FS to distinguish between full simulation and AF2 productions*
  - *This labelling would still be under strict nomenclature rules, decided by derivations*
- AMI hashtags. Created/maintained by who, what structure, how regular, if at all? May end up being chaotic, can always use original **`recon.AOD.e_s_r`**
- Some Rucio/AMI cl options for picking only one tag (the first) of each type?
- Only limited, registered people run derivations, not an ATLAS free for all
  - *May solve more problems and save more CPU and person-power compared to anything else*

Extra

MC Production Workflow



- Requester talks to their sub-group convenor about MC needs and to their MC contact person about samples
- ATLMCPROD JIRA ticket is created, usually by MC contact
- Production request is created in ProdSys by MC contact, via input spread-sheet
- PMG convenor approves request
- MC production team member takes over the request (becomes manager), modifying it if necessary, before submitting it for production
- MC production manager monitors the request until finished
- MC production manager puts notification in JIRA when done

MC Production: Tags

- All tags maintained in AMI, main ones: `e-tag`, `s-tag`, `a-tag`, `r-tag`, `p-tag`

- `e-tags`: EVNT (EVGEN) production and merging
  - *We run MC event generators, for example Pythia6/8, Herwig++, Powheg, Sherpa, MadGraph, Alpgen, etc. Sometimes using LHE files or grid-packs as inputs*
  - *e-tag must contain tar.gz file of relevant Job Options and as this is different for each request there are many e-tags in the current model*

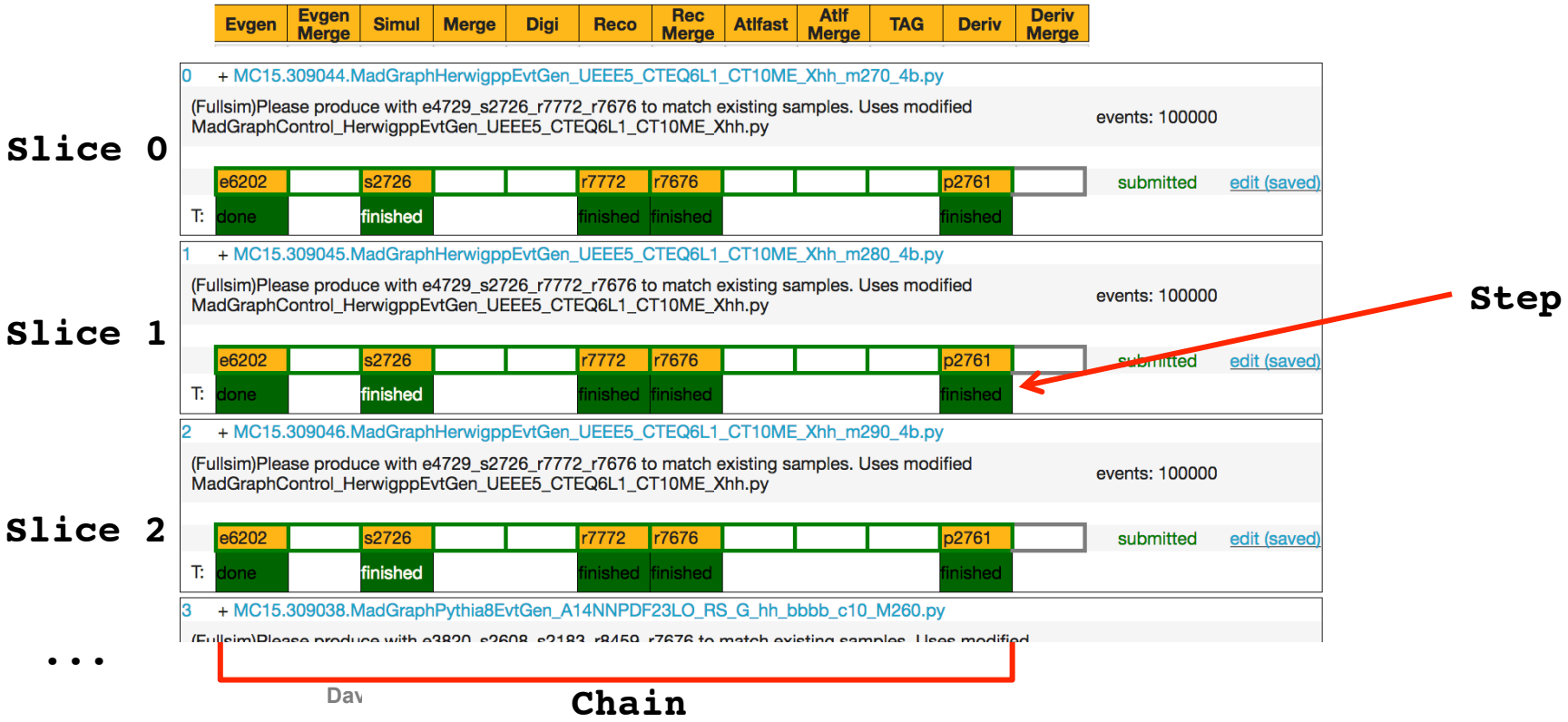
- `s-tags`: Geant4 simulation to produce HITS and merging
- `a-tags`: Simulation tag when running faster, less detailed AFII simulation

- `r-tags`: Digitisation and reconstruction, as well as AOD merging

- `p-tags`: Production of NTUP_PILEUP format and merging
  - *Used by analysis in conjunction with merged AOD, contains same events*

MC Production: Requests and Slices

- Each MC production request consists of a logical set of samples
  - *Request made up of many (up to 300) slices, typically grouped in physics not computing!*
  - *There may also be sub-slices within a slice (see later)*
- Each chain consists of different production steps, each with its own tag
  - *Example of MC15 production, much simpler workflow than MC16: submitted in two clicks!*



ATLAS Monte Carlo for Run 2: The MC16 Campaign

Sub-campaign	Data match	Processing	Description
MC16a	2015+2016	HITS+digi+reco	matches 2015+2016 dataset + mu profile
MC16b	none	digi+reco	uses higher mu (30-70) for trigger and CP studies for 2017 data
MC16c	2017	HITS+digi+reco	uses expected 2017 mu profile, geometry and trigger
MC16d	2017	digi+reco	uses true 2017 mu profile
MC16e	2018	HITS+digi+reco	uses expected 2018 mu profile and trigger
MC16f	2018	digi+reco	uses true 2018 mu profile if required

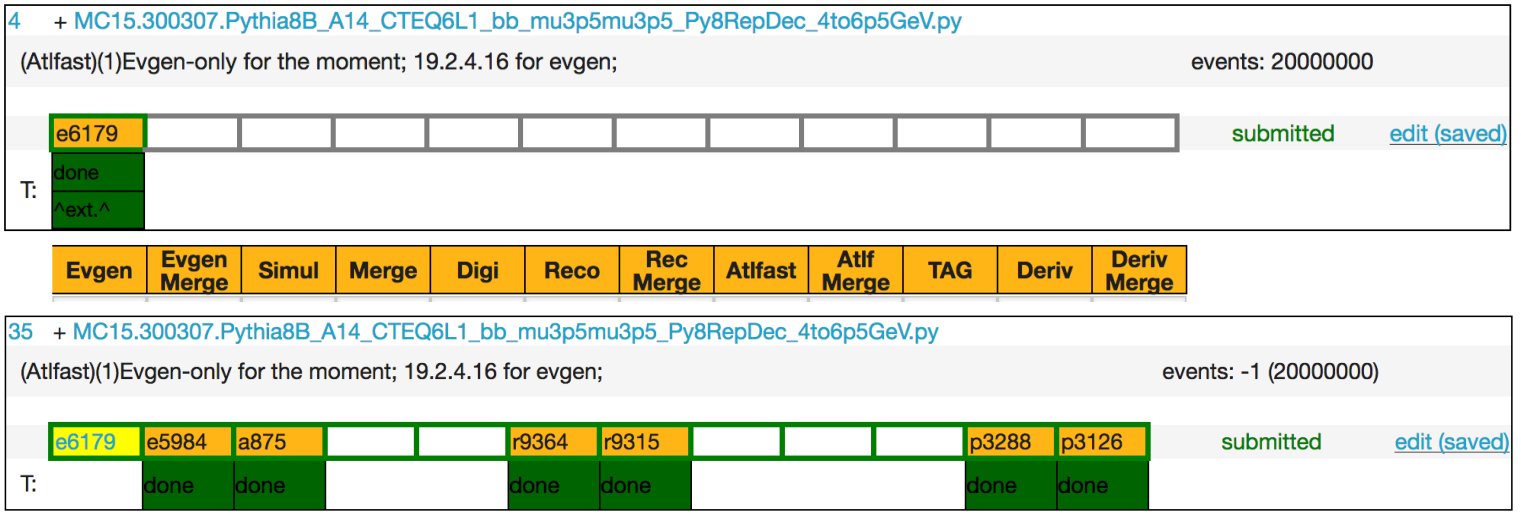
- There are six **sub-campaigns** foreseen in MC16
- MC16a, MC16c and MC16e have to use statistically different EVGEN events but the same EVGEN configuration, so they can be combined for analyses using all run 2 data
- The simulation configuration for HITS production is the same for all sub-campaigns
- MC16c and MC16e are initial versions for 2017 and 2018 with the initial mu profile for that year; MC16c is now superseded by MC16d, which has the updated pile-up distribution for 2017
- MC16d and MC16f samples may also use different conditions, e.g. if part of some detector got disabled during the run and is thus masked in the new MC version

What does this mean for production?

Physics	evgen		simul		recon	
Sub-campaign	Project	Campaign	Project	Campaign	Project	Campaign
MC16a	mc15_13TeV	MC16:MC16a or MC15:MC15* or None	mc16_13TeV	MC16:MC16a	mc16_13TeV	MC16:MC16a
MC16b	mc15_13TeV	MC16:MC16b	mc16_13TeV	MC16:MC16b	mc16_13TeV	MC16:MC16b
MC16c	mc15_13TeV	MC16:MC16c	mc16_13TeV	MC16:MC16c	mc16_13TeV	MC16:MC16c
MC16d	mc15_13TeV	MC16:MC16c	mc16_13TeV	MC16:MC16c	mc16_13TeV	MC16:MC16d
MC16e	mc15_13TeV	MC16:MC16e	mc16_13TeV	MC16:MC16e	mc16_13TeV	MC16:MC16e
MC16f	mc15_13TeV	MC16:MC16e	mc16_13TeV	MC16:MC16e	mc16_13TeV	MC16:MC16f

- We have multiple configurations to cover the six MC16 sub-campaigns
- Project is the same for a given step across all sub campaigns
  - *Only the evgen step has project mc15_13TeV*
  - *All others, including evgen merge, are mc16_13TeV*
- In the case of MC16d and MC16f, only the reco (with the updated pile-up distribution) has that sub-campaign
- Some examples…

MC16a Workflow



Request 1

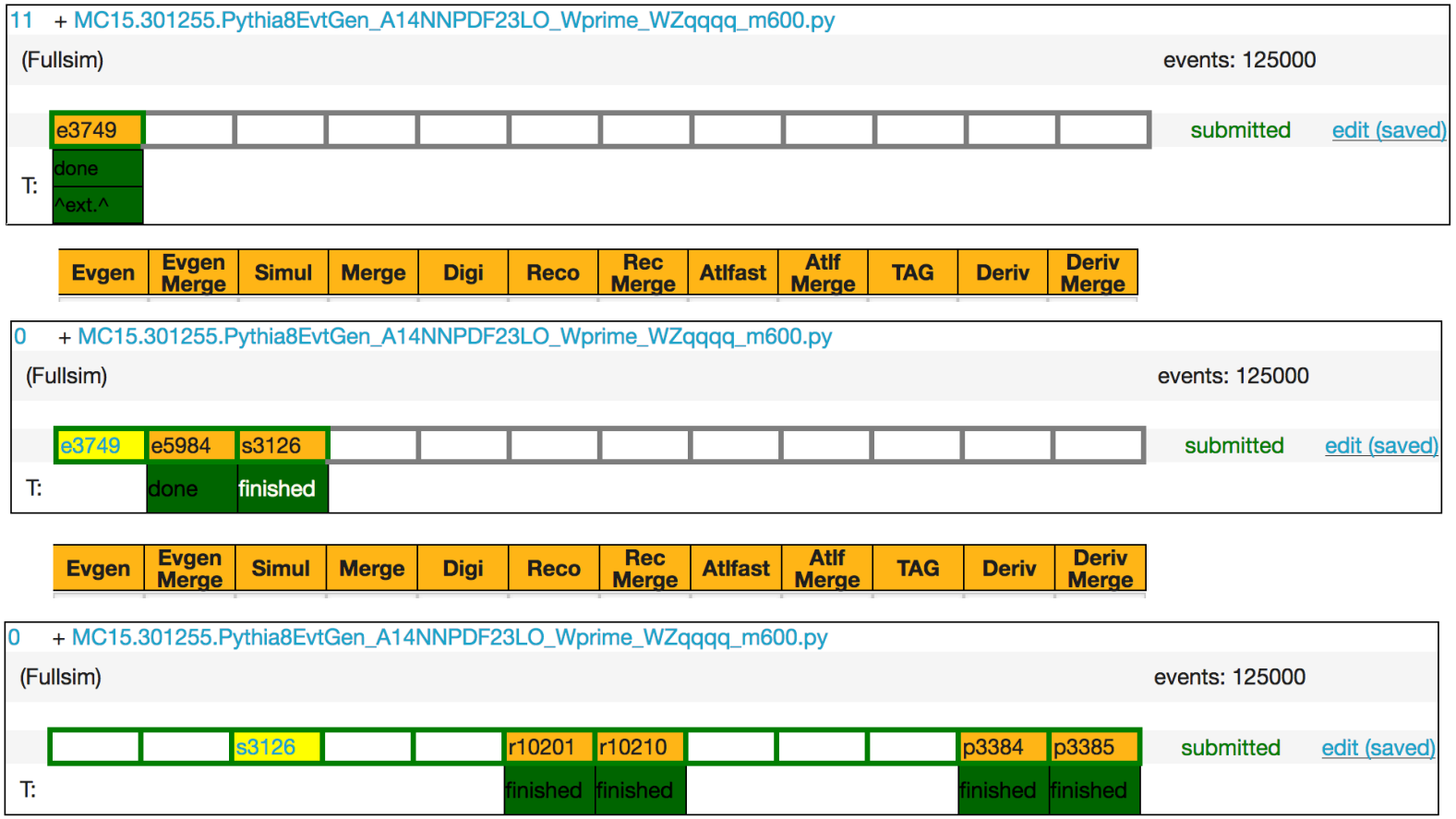
Project:
mc15_13TeV
Sub-campaign:
MC15c/MC16a/None

Request 2

Project:
mc16_13TeV
Sub-campaign:
MC16a

- For both MC16a and MC16c, the standard workflow is 2 requests:
  - *First request performs the Evgen step*
  - *Second request to do the Evgen merge, (Fast)Simul, Reco+Merge and Deriv+Merge steps*
- This example is for MC16a: the workflow is similar for MC16c (and MC16e), which uses the MC16c (MC16e) sub-campaign for both request 1 and 2

MC16d Workflow



Request 1

Project:
mc15_13TeV
Sub-campaign:
MC16c

Request 2

Project:
mc16_13TeV
Sub-campaign:
MC16c

Request 3

Project:
mc16_13TeV
Sub-campaign:
MC16d

- The standard MC16d workflow for new requests necessarily involves three requests:
  1) Evgen
  2) Evgen merge and Simul
  3) Reco + merge and Deriv + merge

MC16d Workflow using existing MC16c HITS

Evgen	Evgen Merge	Simul	Merge	Digi	Reco	Rec Merge	Atlfast	Atlf Merge	TAG	Deriv	Deriv Merge
9 + MC15.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.py											mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmum
(Fullsim)											events: -1
e5271		s3126			r10201	r10210				p3384	p3385
T:					finished done	finished done				finished done	finished done

- Only run steps from reco, using HITS produced in an earlier MC16c production
- Here you see sub-slices, so there are two input MC16c HITS tids
- In fact there multiple tids from different sub campaigns available as inputs:
  Here we have 2 x MC16a, 2 x MC16c and 1 xMC16e HITS tids:

Dataset Name	Events Number	SubCampaing	Tasks
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_e5984_s3126_tid12196360_00	1489400	MC16:MC16c	13038485
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_e5984_s3126_tid12592858_00	7431200	MC16:MC16e	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid10730514_00	1995000	MC16:MC16a	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid10944971_00	3986600	MC16:MC16a	
mc16_13TeV:mc16_13TeV.364105.Sherpa_221_NNPDF30NNLO_Zmumu_MAXHTPTV70_140_BFilter.simul.HITS.e5271_s3126_tid11324488_00	5981200	MC16:MC16c	13038493

- Note that for historical reasons not all evgen datasets have been merged, so there are some tids with one e-tag and some with two e-tags

Another example of a varied HITS container

mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.simul.HITS.e5340_s3126

Dataset Name	Events Number	SubCampaing
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483117_00	1999900	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483123_00	1996300	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483128_00	1995800	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483136_00	1995900	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483142_00	1999850	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483149_00	1999950	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483154_00	1999800	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483161_00	1999950	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483166_00	1999950	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483172_00	1999700	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483177_00	1999800	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483184_00	1999950	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.merge.HITS.e5340_e5984_s3126_s3136_tid11483204_00	990000	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.simul.HITS.e5340_e5984_s3126_tid12197119_00	6217000	MC16:MC16c
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.simul.HITS.e5340_e5984_s3126_tid12944773_00	31098000	MC16:MC16e
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.simul.HITS.e5340_s3126_tid10730390_00	8330000	MC16:MC16a
mc16_13TeV:mc16_13TeV.364156.Sherpa_221_NNPDF30NNLO_Wmunu_MAXHTPTV0_70_CVetoBVeto.simul.HITS.e5340_s3126_tid10944745_00	16469000	MC16:MC16a

> This container also including merged HITS