Proposed information flow for Glance to Gitlab to AMI to DDM

0) We should agree on a convention for naming the AMI hashtag describing the datasets used in an analysis.

AMI proposes the following :
Hashtags are in the form “GlancePublication:<analysis-id>” where

- “GlancePublication” is the hashtag's scope
- “<analysis-id>” is simply the glance id. For example “HDBS-2018-11” for this analysis

AMI will then authorize only the gitlab CI script to set hashtags under the “GlancePublication” scope.
Thanks to view access to the Glance DB, AMI can also

- Authorize individual members of analysis “Analysis-X-Y” to set hashtags in the form “GlancePublication:Analysis-X-Y_anything”. This is to allow analysis team to sub-group easily their datasets
- Prevent the addition/removal of the main hashtag “GlancePublication:Analysis-X-Y” when the analysis is in a “published” (or any other frozen state) in Glance. This will in practice implement the freezing of the dataset list.

1) Glance creates a “<GlanceID>_dataset.txt” (or equivalent) file in the **existing** Gitlab repository of each analysis team.
Questions from glance team :

- Should the datasets file be defined for all analysis? Even the ones that skip the Phase 0 workflow? All leading groups?
    - Answer (Borut): Yes, for all analyses; each Physics analysis uses data and MC containers..
- Should the dataset files be added in all repositories (Papers, CONF note, and PUB notes) or only the Papers one?
    - Wouldn't it be more convenient for analyzers the file is in the code repo ?
        - Answer (Borut): I think the same file, listing containers used should be in both places, good point. The analyzers (or some Gitlab magic) need to make sure the file is synced..

2) Data Prep can create an equivalent structure in Gitlab wherever they think convenient.

- DataPrep creates a Gitlab entry for each case when they want to keep files, this contains only the <ID>_dataset.txt . They bypass Glance and use their own ID instead of the Glance ID, so the Gitlab entry name and the <ID>_dataset.txt are up to DataPrep to define. We need to make sure that the Continuous integration scripts then handle this Gitlab entry in the same way as for the analysis lists.
  (this is meant by ‘equivalent structure’).

3) Every edit to the *_dataset.txt file triggers a cont. integration (CI) Gitlab job that generates a diff file with the previous version.

A part of the merge request could be the trigger for AMI update, or CI or … In any case, the Gitlab should create a push request to AMI, not AMI to browse Gitlab…

AMI also expects the CI scripts will send the add/remove hashtags command for a list of datasets from the diff file (we (AMI) think it's more flexible to let the script decide which dataset needs to be tagged or not)

ACTION: Check with Lukas on Gitlab ways to do this…

4) Glance adds a link from their analysis page to an AMI webpage displaying the dataset list for the analysis hashtag (Glance prefers this solution rather than embedding the dataset table)

5) AMI refreshes daily a list of tagged datasets. This list is used as exclusion list by the DDM lifetime model script, similarly to the exclusion lists that are now generated by people requesting the exclusions. No change is needed in Rucio.

In the convention proposed in 0) the list would simply be all datasets having the hashtag scope "GlancePublication" .

6) After the analysis ends the final hashtag gets 'frozen', so that a hashtag of a published analysis remains unchanged. (CREM can then fully implement the policy what to do with the inputs of published/preserved analyses - e.g. store them in CERN EOS?)

In the convention proposed in 0) the freezing is automatic when the analysis reaches the relevant status.

Other options : This could be done maybe in a way that when an analysis reaches the status "published" in Glance, Glance copies the "<GlanceID>_dataset.txt" to "<GlanceID>_dataset_final.txt" (and empties "<GlanceID>_dataset.txt" ?) or … ? The freezing of the final setup and how to do it needs a bit of discussion).

Actions to implement the workflow

a) Glance must create the file in Gitlab when an analysis repository is created, save the link and take action 6) when the paper or note is published and put the AMI hashtag link in Glance (and maybe use the AMI tools to present the hashtag info embedded into Glance, if desired?)

b) Someone must set up the CI job in Gitlab to create the diff file and send it to AMI (and perhaps Rucio).

c) AMI must provide the global exception file if needed.

d) Rucio must (perhaps) provide the interface to receive the diff files and set the dataset lifetimes.

Misc

DatasetSearch by Hashtag : https://ami.in2p3.fr/app/?subapp=search

Choose for example mc16, there is 2 search parameters hashtag scope and hashtag name -> you can select datasets here

There is also a google like dataset search by hashtag here : https://ami.in2p3.fr/app/?subapp=searchDatasetsByHashtag

If one wants to associate an hashtag to a dataset, one have to select the dataset with AMI search interface and, once get the selected list of dataset, click on the #hashtags links under one logical dataset name.

Of course all these features accessible via the web interface are also available via an API or lightweight client.