CAP requirements: update & plan for next steps

CMS:

- Data model is more or less finished and currently under discussion within the collaboration: http://simko.home.cern.ch/simko/tmp/cms-analysis.png
    - some parts are still somewhat high-level (regarding physics information)
- Next steps:
    - Adapt model to collaboration feedback
    - Create prototypes and test them with researchers to see how the submission best fits their workflow
    - Check APIs of DAS, CRAB, CADI etc. for auto-filling fields
    - Integrate statistics questionnaire

ALICE:

Overview of the ALICE LEGO train system:
https://indico.cern.ch/event/359440/contribution/1/5/material/slides/0.pdf

- there's a possibility to run jobs/tasks outside the LEGO system, but if an analysis should get approved, there has to be a train iteration connected to it
    - about 60% analysis are using central system; interest in driving others that do not use it yet
- all code used for a 'wagon' is stored/linked in a ALICE code repository
    - macros are stored in ALIRoot indeed (not linked)
- once per day (4pm CERN time) there's a snapshot taken of the whole train including all the macros providing information on parameters and software version used
- output of the train run is one big root file including the results of all the wagons, researcher has to look for his/her histogram etc. in there
    - importance of "internal" merge step
- all output file are stored permanently on the grid, nothing is deleted (LEGO system provides path to the root output file)
    - which means effectively long-term preservation; we only need to store links
- to rerun an analysis, one needs iteration of the train which is connected to information such as software version used, datasets used and the macros used for the wagons
- each Aliroot version contains the database needed for analysis
- ALICE LEGO train system doesn't have an API as there's no need for it

Next steps:

- develop data model based on the submission form + the analysis relevant information from LEGO
- investigate w/ ALICE LEGO team how to push/pull the analysis relevant information as JSON from LEGO to CAP (will require some thinking/discussion within ALICE when one is sure to have the correct iteration of a train and it is the correct moment to take the snapshot)
    - it was not wanted to do it systematically due to large noise (not needed to store every testing and debugging run)

- added value of CAP will be the linking of train iterations to published papers and thus getting statistics on which datasets are important (for clean-up and keeping less copies in the grip e.g.)
- there might be a summer student being able to help with some example analysis

LHCb:

Bookkeeping database:

- access with LHCb grid certificate only (most do it via terminal, though there is a web interface; can't find a link that doesn't give me an error because I don't have access rights though)
- no search box to query
- query through navigating through the tree structure
- tree with the list of subdetectors, referenced all through the file path
- further grouping by conditions of the magnet, beam data, collision data etc.
- also exclusions are sometimes included in the file naming
- separate run database that stores metadata for the runs
- tree navigation then follows the processing steps of an analysis: RAW - > RECO -> Stripping (=selection criteria) -> streams -> files/events
- this all is defined by the physics groups
- the LHCb stripping lines are self-explanatory python modules that include the metadata of the selection; they are documented at http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/stripping/
- so all queries are converted into these path strings, such as /LHCb/Collision11/Beam3500GeV-VeloClosed-MagDown/RealData/Reco09/Stripping13/90000000/CHARMCOMPLETEEVENT.DST
- for an analysis, one takes a list of these lines and writes the analysis code around it
- currently, there is no way to search and find out if data was used already, datasets are used by certain working groups only and the group convener will know about the usage and point newcomers to "their stripping lines"

- n-tuples created in the working groups are not indexed/uploaded anywhere for the whole collaboration but stay within working groups and their tiers
- most n-tuples created for analysis are root files or at least compatible with/convertible into root

Conclusion:
- path strings such as /LHCb/Collision11/Beam3500GeV-VeloClosed-MagDown/RealData/Reco09/Stripping13/90000000/CHARMCOMPLETEEVENT.DST could be chopped into their single elements which are then connected to python modules that contain context metadata
- LHCb could do that on their end so that we don't have to adapt in case they change anything on their side
- in addition, they are currently working on automatically including (provenance) metadata into their root files (the final n-tuples) which we could extract if people upload and store these

files on CAP (which then also could be the central storage for the experiment for this kind of datasets)

Next steps:

- develop data model based on S.'s slide and feedback + A.'s efforts
- investigate how A can help with automatic extraction and pre-filling of metadata
- adapt the submission form according to S.'s suggestions in the slides and then do a next round of testing/feedback

Overarching task:

- DASPOS workshop to try and "harmonise" data models/create a "HEP analysis ontology" during May 18-20th @Universty Notre Dame