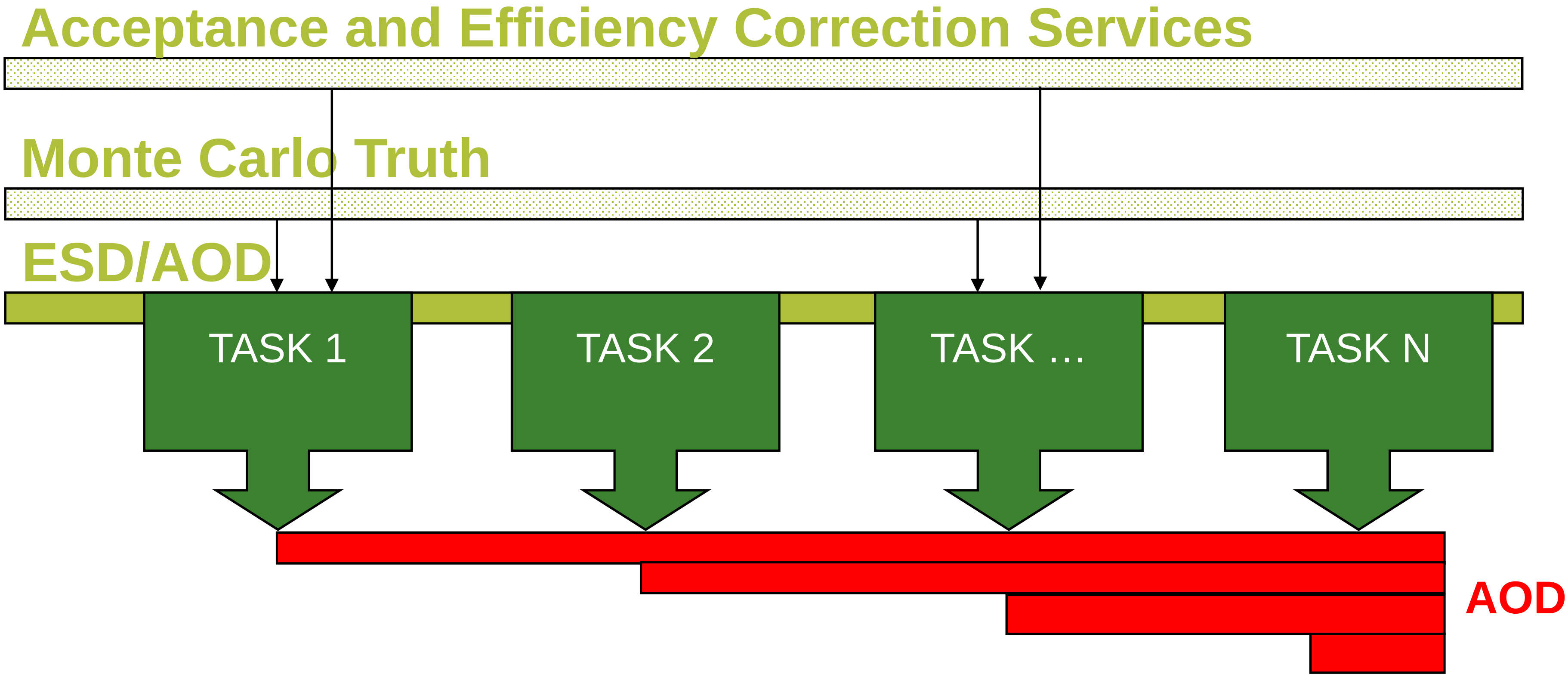
Status of the Analysis Train

A. Gheata, M. Gheata, A. Morsch

CAF-GRID User Forum
27/3/2008

Organized Analysis

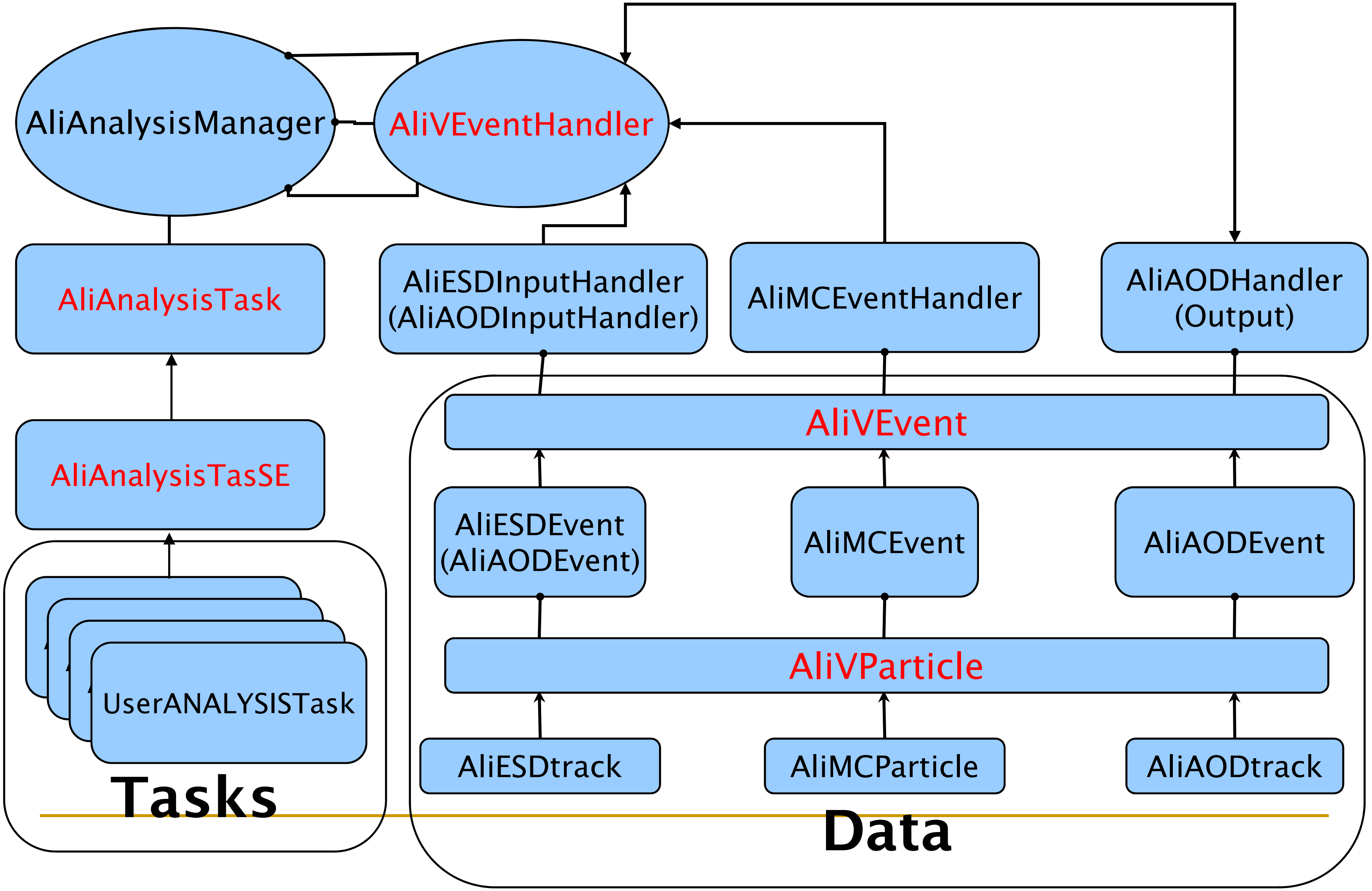
- Organised analysis is the most efficient way for many tasks to read and process the full data set
  - Optimise CPU/IO ratio for distributed resources
  - Common well tested framework
  - Common knowledge base and terminology
  - Document procedure
  - Makes results reproducible
- Will run “sanctified” algorithms and will assess global data quality


Acceptance and Efficiency Correction Services
Monte Carlo Truth
ESD/AOD
TASK 1
TASK 2
TASK …
TASK N
AOD

Usage of the Framework

- Conclusion from Physics Week in Prague
  - All Physics working groups are using the official analysis framework.
- In some cases they come up with new/known requirements.
  - Multi Event analysis
  - AOD updates
  - … we are working on this

The overall picture



Important simplifications

- For single event analysis, user can derive from *AliAnalysisTaskSE* and implement
  - LocalInit()
  - UserCreateOutputObjects()
  - UserExec()
  - Terminate()
- Pointers to input and output events + optional MC information are provided by the framework.

New InputHandler

- *AliESDInputHandlerRP*
  - Gives access to RecPoint Trees (if available)

Memory checking

AliAnalysisManager::SetNSysInfo(Long64_t nevents)

- Monitoring memory&cpu usage each *N* events
  - Using (Marian's) AliSysInfo functionality to spot leaking tasks
  - Only in local mode
  - Writing an ntuple with memUsed and cpuUser and plotting 2 histograms in Terminate()
  - 2 output files: syswatch.log and syswatch.root

Special outputs in PROOF mode

- Files can be written on workers in PROOF, but there is no way to access them
- Merging of files foreseen in PROOF, but what about non-mergeable information
  - For a small subset of events, we may want to copy some files to a known location
- Special outputs will be copied to the specified location using TFile::Cp()

AliAnalysisDataContainer::SetSpecialOutput()

AliAnalysisManager::SetSpecialOutputLocation(const char* dir)

Merging in PROOF via files

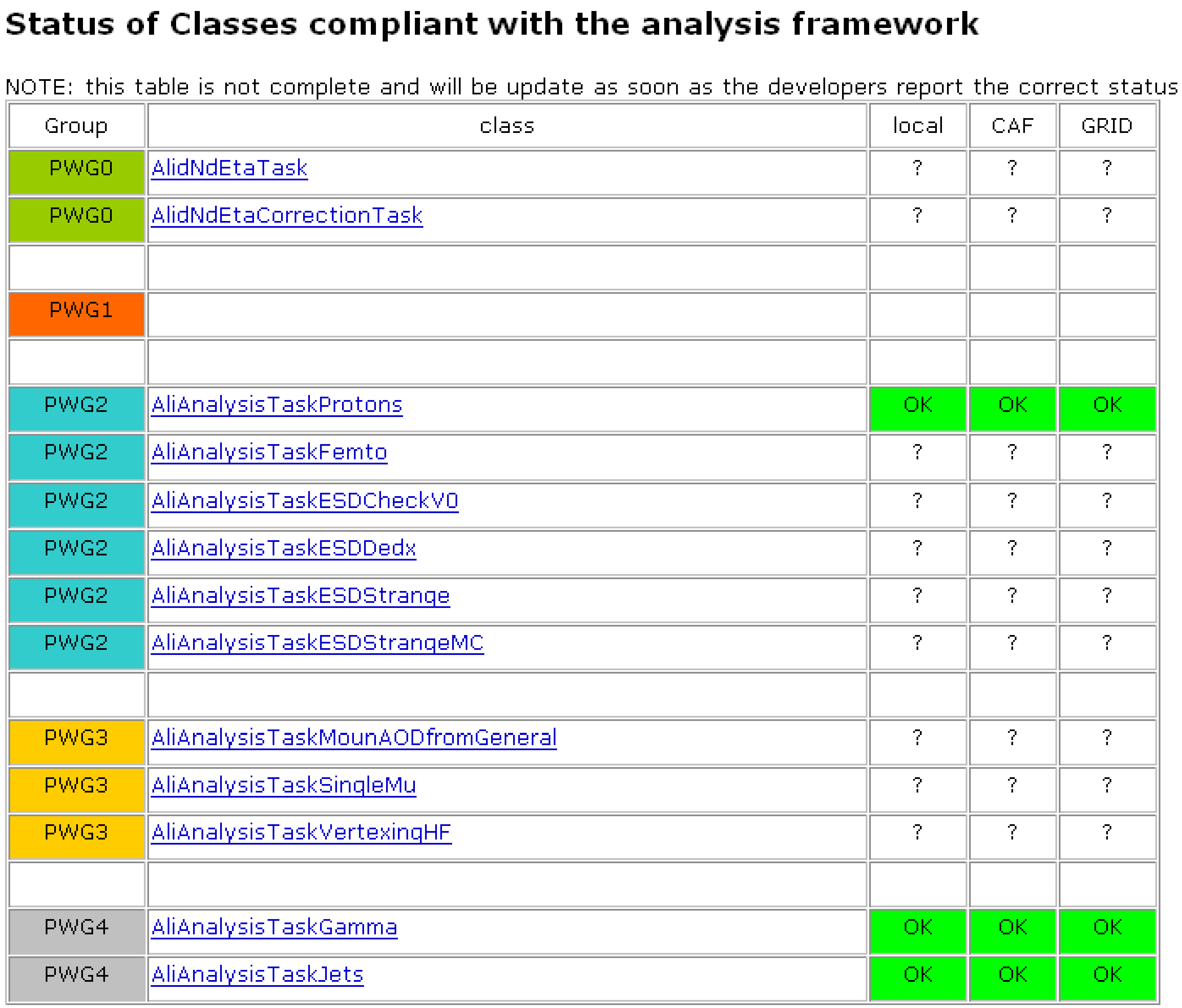
- Implemented in PROOF using *TProofFile*
- Declare the output container as “special outputs” but do not give a location
  - Feature activated only in “proof” mode
  - Files will be merged on the scratch space
  - Waiting for a hook to grab the output from proof team

Documentation

- Web based documentation available thanks to M. Gheata
- http://aliceinfo.cern.ch/Offline/Analysis/AnalysisFramework/
  - Manual
  - Train status

Test train

- Routinely running on GRID and CAF
  - Filter
  - JETAN
  - GAMMA
  - PROTONS
- Running with the Efficiency and Correction framework has been tested and problems fixed



Status of Classes compliant with the analysis framework

NOTE: this table is not complete and will be update as soon as the developers report the correct status

Group	class	local	CAF	GRID
PWG0	AlidNdEtaTask	?	?	?
PWG0	AlidNdEtaCorrectionTask	?	?	?
PWG1				
PWG2	AliAnalysisTaskProtons	OK	OK	OK
PWG2	AliAnalysisTaskFemto	?	?	?
PWG2	AliAnalysisTaskESDCheckV0	?	?	?
PWG2	AliAnalysisTaskESDDedx	?	?	?
PWG2	AliAnalysisTaskESDStrange	?	?	?
PWG2	AliAnalysisTaskESDStrangeMC	?	?	?
PWG3	AliAnalysisTaskMounAODfromGeneral	?	?	?
PWG3	AliAnalysisTaskSingleMu	?	?	?
PWG3	AliAnalysisTaskVertexingHF	?	?	?
PWG4	AliAnalysisTaskGamma	OK	OK	OK
PWG4	AliAnalysisTaskJets	OK	OK	OK

To be done …

- Short Term
  - Fix memory leak in xrootd
- Medium term
  - Filters for V0 and Kinks to be provided by PWG1+2
  - Merging of file resident output objects on CAF
  - Design and implement strategy for updating AODs
  - Multievent analysis

Updating AODs: Current thinking

- Updates will be written to separate files
- Updates will be independent trees and can be connected as friends to the standard AOD
- Header (run, event) information will be copied to each update file.