Organized Analysis
Some Words of Introduction

A. Morsch

ALICE Offline Week

Why this introduction

- Let’s say some important things at the beginning …
- … then let’s hear what the PWGs have to say …
- … and let’s try to wrap up tomorrow during the discussion session.

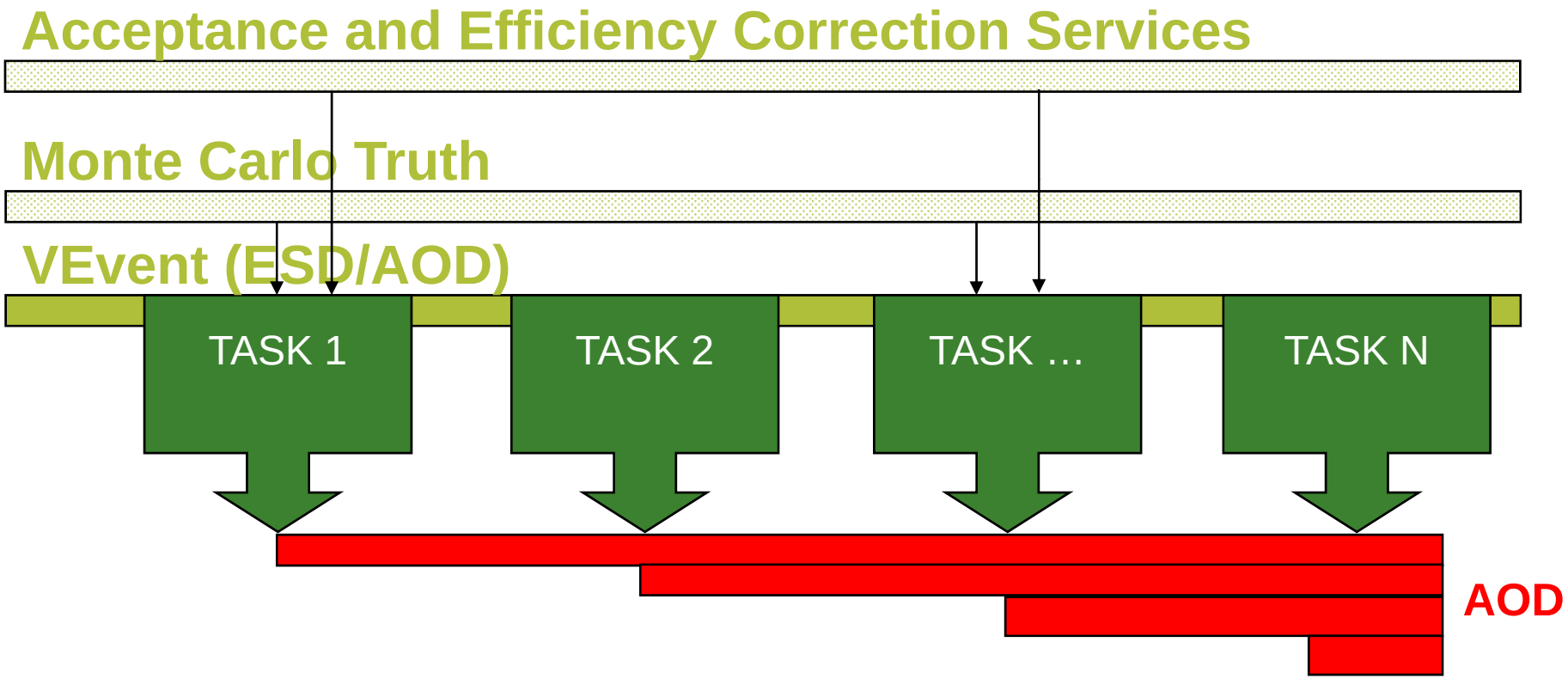
Organized Analysis

- 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

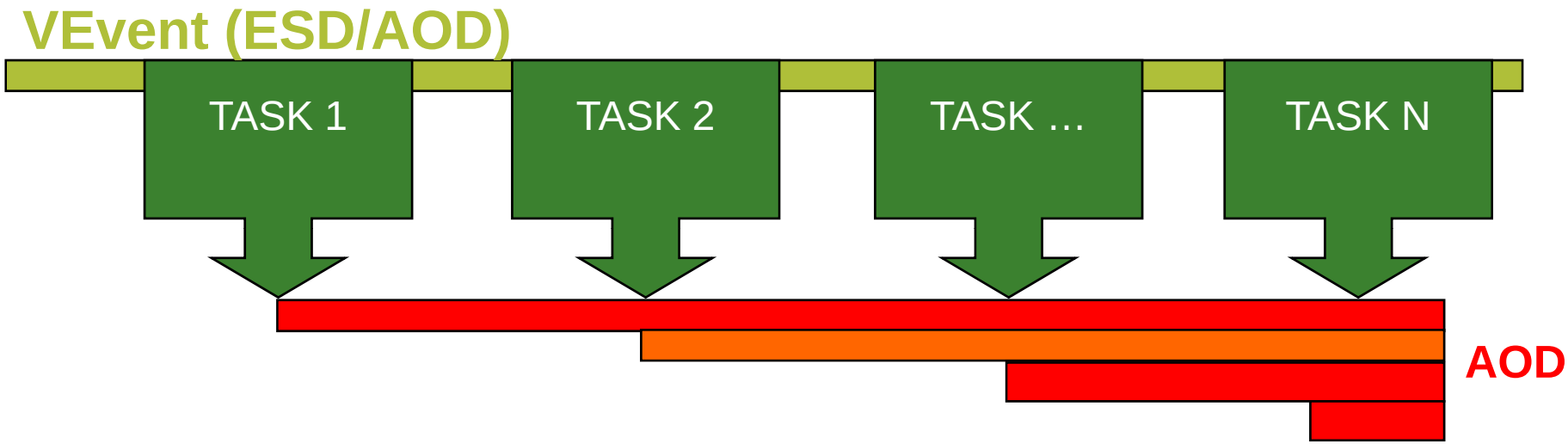
Usage of the Framework

- Conclusion from Physics Week in Prague
  - All Physics working groups are using the official analysis framework.
- While users are adding new wagons we are testing and work on the rails …

Standard AOD production via AliAnalysisTaskSE

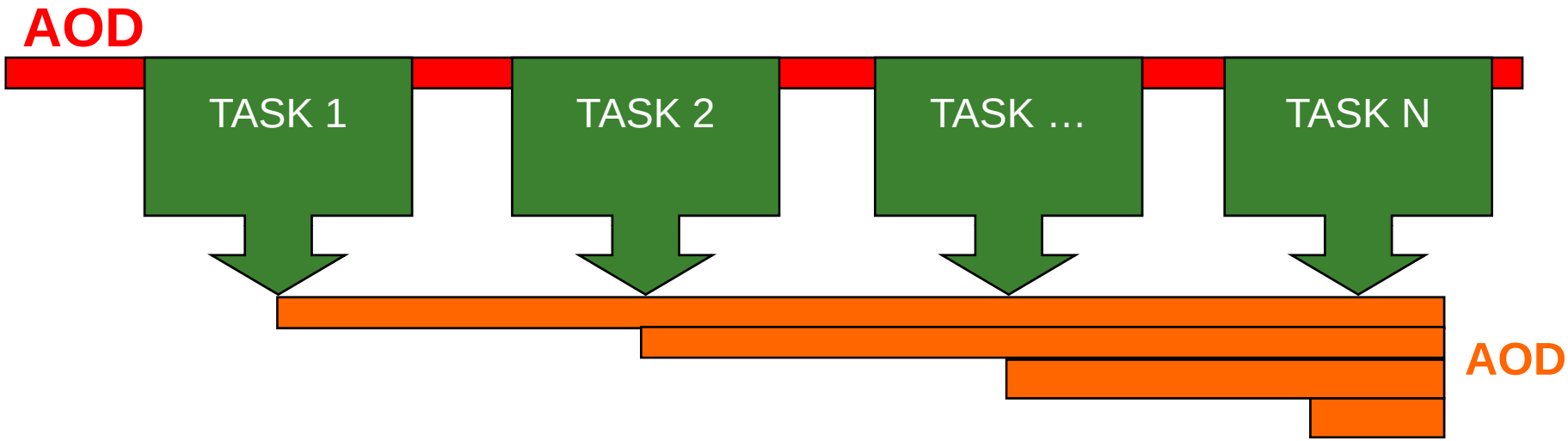


Adding new branches …



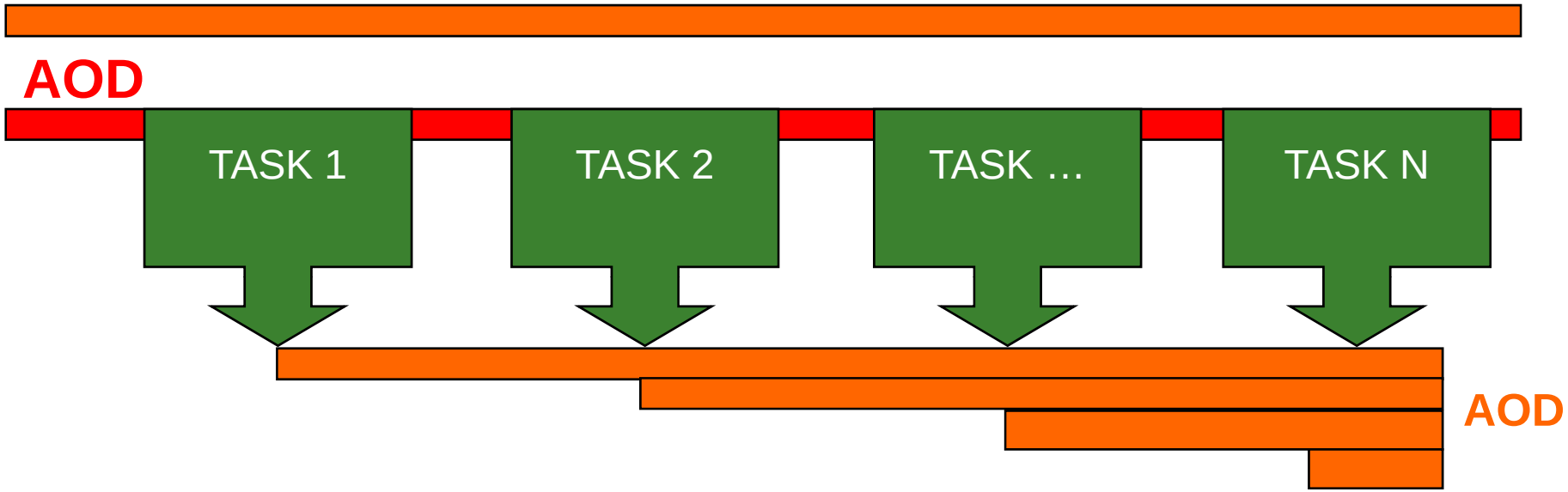
Call
AliAnalysisTaskSE::AddAODBranch(const char* cname, TObject* addobj);
Has to be placed into
AliAnalysisTaskSE::UserCreateOutputObjects()

AOD updates



Call
AliAnalysisTaskSE::AddAODBranch(const char* cname, TObject* addobj);
Has to be placed into
AliAnalysisTaskSE::UserCreateOutputObjects()
In addition
AliAODHandler::SetCreateNonStandardAOD()

Reading Standard and Update AOD



Call
AliAODInputHandler::AddFriend(char* filename);

Some open questions

References: TRref, TrefArray

- Have to work for objects created by different processes
  - Works for *TRref* with a simple trick
    - Enforce that UID is created in process that creates the objects
    - No new UID when object is references (could be another process)
  - Needs some changes in *TRefArray* (updates sent and discussed with Ph. Canal)

Event/Run related information

- AOD updates should be stand-alone
  - Copy header to each update ?
- Is the current contents of AliAODHeader sufficient ?
  - Is there any other run (OCDB) related information you need during analysis ?

Updating AODs: Current thinking

- Updates will be written to separate files
  - Into the same SE as the original
- Updates will be written as independent trees and can be connected as *friends* to the standard AOD
- Two use cases
  - True updates, i.e. updates for almost all events
    - Copy event header information to updated file
  - Filter/selection of rare processes
    - Replicate all necessary information for further processing and analysis
- Need synchronization objects to help to update dependent branches coherently.

How to optimise reading of standard AODs and updates

- On the grid ?
- On CAF
- Current PROOF/CAF bottleneck
  - No output file handling api
  - No file merging

The overall picture