

# EDM/EMB Meeting

---

## ❖ Organizational changes

- Karsten Koeneke has moved on to co-chair PAT
  - Looking for replacement
- Kirill is leaving as InDet rep, important to find replacement

## ❖ Meeting overview:

- Cover current event sizes
- See if we can reduce ESD size
- Some progress reports
- Removing backwards compatibility:
  - First look at which classes are used by the trigger bytestream

# I/O speed optimization

---

- ❖ Ilija has optimized vertex I/O and Kirill added in to rel 16
  - Appears to have validated without any problem
- ❖ Next step will be to simplify the tracking model
  - This will affect `Trk::Track` and `TrackParticle`
  - This could also improve the tracking size
- ❖ Trigger has been working on improves which I expect we will hear from Fabrizio

# Backwards compatibility

---

- ❖ First step is to outline the set of persistent classes that must be maintained
  - List needed from bytestream streaming - see Fab's talk
- ❖ Then need to organized a systematic cleanup of old unused versions
  - Will affect converter code and persistent class dictionaries
    - Dicts are loaded for each ROOT startup!



# Calo ESD size – from Guillaume

---

- ❖ For ttbar MC: calo is 440 kB of 1.25 MB total
  - CaloCell = 280 kB, highly compressed
  - TopoClusters for jets = 120 kB, fairly optimized, could look a bit here
  - Rest (CaloTopoEM) ~ 40 kB
- ❖ We can have a closer look, but don't expect a large gain here

# Event sizes (backup)

---

## ❖ ESD: period G (need to look at later periods)

- Egamma: 1.6 MB

- ID: 45%, Calo: 33%, EMT: 13%, Jet: 11%, Trig: 5%

- Muons: 1.3 MB

- ID: 42%, Calo: 34%, EMT: 11%, Trig: 6%, Muon 5%

- JetTauEtmis: ~2 MB

- ID: 45%, Calo: 30%, EMT: 11%, Trig: 4%

## ❖ AOD: period H

- Egamma: 150 kB

- ID: 32%, Calo: 21%, EMT: 11%, Jet: 11%, Trig: 12%, MET: 4%

- Muons: 150 kB

- ID: 28%, Calo: 18%, EMT: 12%, Muon 9%, Jet: 8%, Trig: 17%, MET: 4%

- JetTauEtmis: 200 kB

- ID: 32%, Calo: 20%, EMT: 13%, Jet: 11%, Trig: 11%, MET: 3%

# Fast file merge

---

- ❖ Would like to improve the merge time for POOL files
- ❖ Two options available:
  - Pers-to-pers: copy event data on persistent side, no transient conversion
    - Still has unzip/rezip of persistent objects
    - Meta data is handled normally, i.e. merged correctly - small fraction of time
    - Improvement seen: ~30% in time
  - poolMerge: avoids zipping, can be 10-20 times faster
    - Metadata is merged each time new file is read
    - Must verify that this works

# Fast file merge (2)

---

## ❖ Considerations:

- Know bug: UserDataSvc is not correctly merged
  - Report just submitted
- File optimization: I see two options
  - Input files are optimized => merge is fast / no optimization
  - Input file optimization is OFF => optimize output of merge
    - This will of course slow down the merge

