

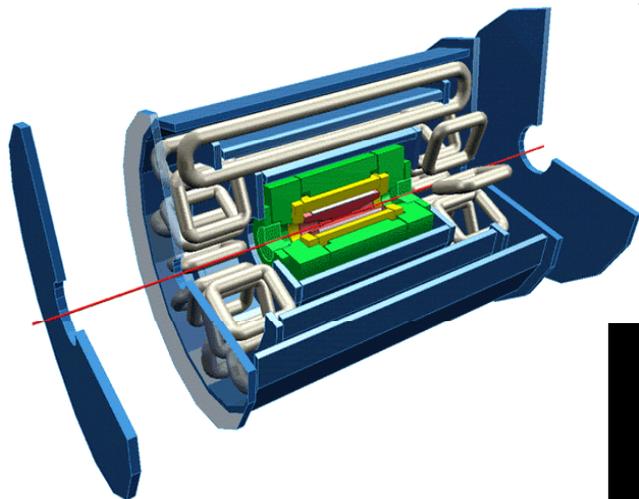
# Tecnologie software per l'analisi offline



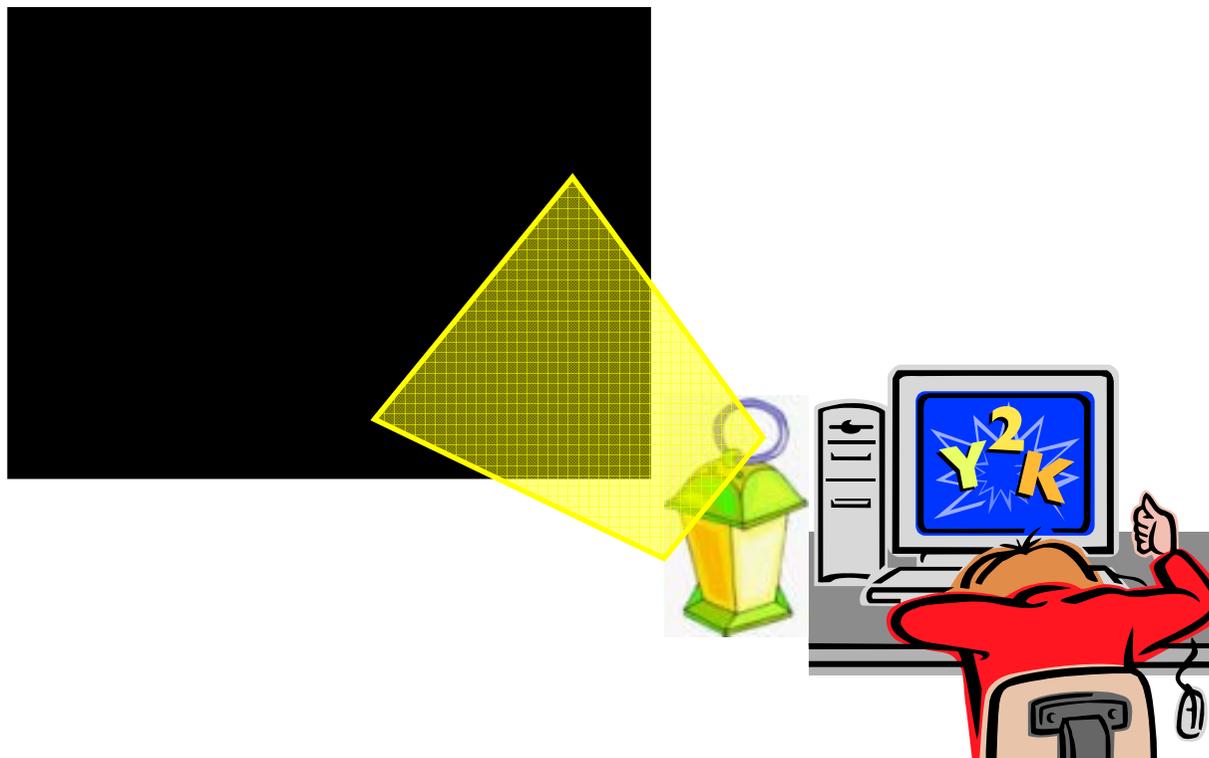
Carminati Leonardo  
Universita' e sezione INFN di Milano



## CM, EDM, AM e altro ancora...



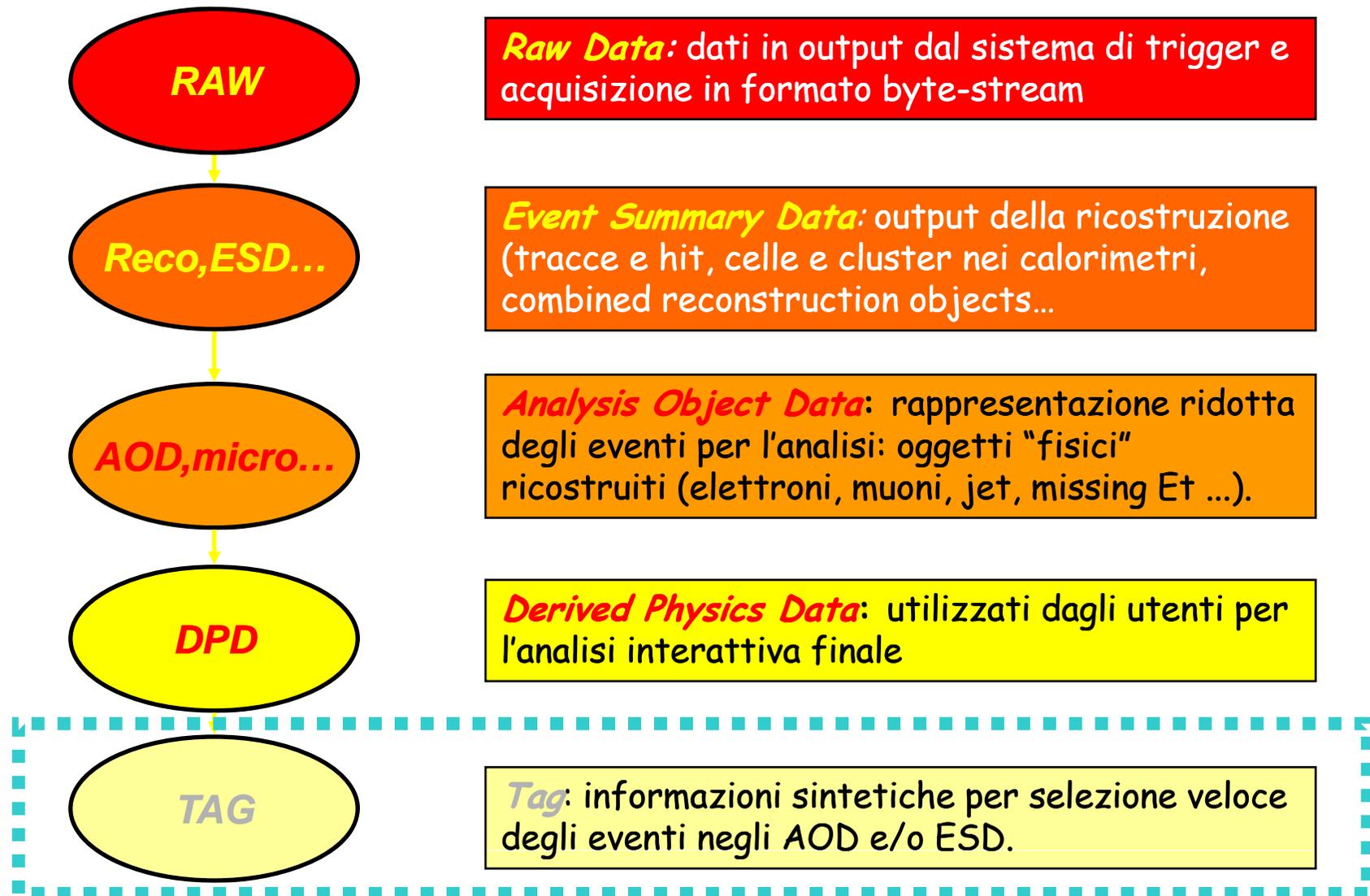
- Cerchero' di affrontare il problema di come fare l'analisi dal punto di vista dell'utente
- Occorre fare luce su quel tanto che basta del CM per capire come muoversi



## Sommario & disclaimer(s)....

- ❑ Overview sul data flow, formati e tecniche di analisi degli esperimenti che stanno prendendo dati
  - ❑ CM1 e CM2 di Babar, CDF e D0
- ❑ Cosa si e' capito dall'esperienza dagli esperimenti attualmente in funzione
- ❑ Analysis model di ATLAS e CMS: soluzioni, pro/cons...
- ❑ Impossibile una review completa: mi limitero' a segnalare gli aspetti che mi sembrano piu' significativi
- ❑ Non sono un esperto di software e computing:
  - ❑ Chiaramente un male per gli esperti presenti in sala
  - ❑ spero un bene per i non addetti ai lavori...

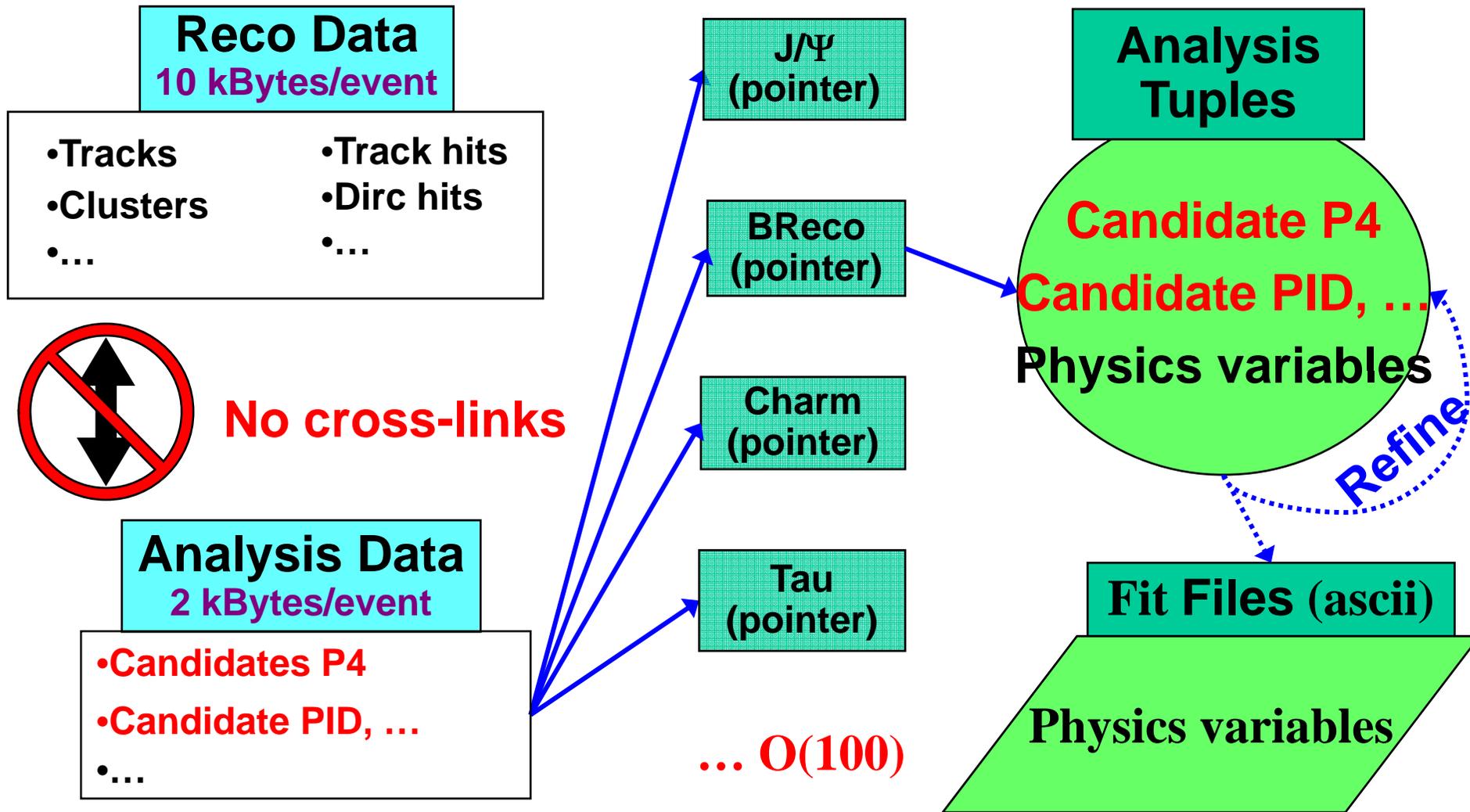
## Tipico data-flow



## Analogie, differenze e keypoints

- ❑ Central vs distributed computing:
  - ❑ running experiments nascono prevedendo un central computing e si muovono verso forme piu' distribuite
  - ❑ gli esperimenti LHC investono (e scommettono!) molto sul calcolo distribuito
  
- ❑ Formato di analisi, accessibilita' dei dati ai vari livelli e possibilita' di aggiungere user data.
  
- ❑ Frameworks di analisi vs analisi privata ROOT - based: portabilita' del codice di analisi e uso di tools comuni
  
- ❑ Alcune parole chiave da tenere a mente
  - ❑ Skimming : "tenere solo gli eventi interessanti"
  - ❑ Thinning : "tenere solo gli oggetti interessanti (ex. Elettroni, muoni..)"
  - ❑ Slimming : "tenere solo subset di informazioni relative agli oggetti selezionati"

# running experiments : il CM1 di Babar

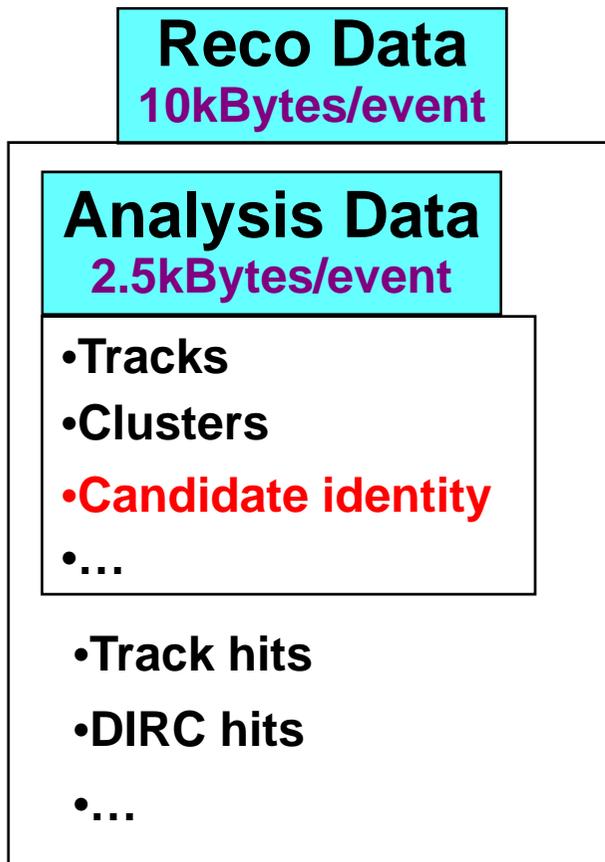


## running experiments : limiti del CM1 di Babar

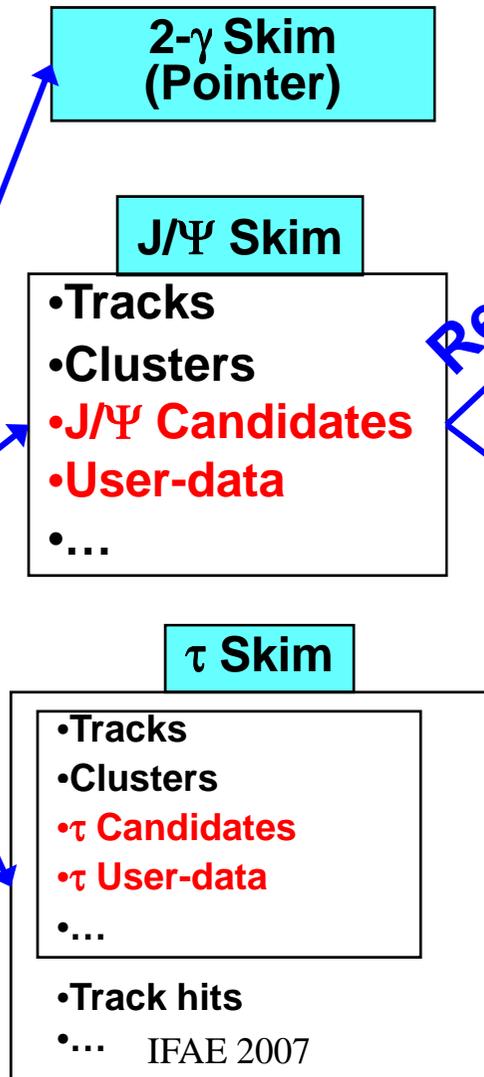
- ❑ Necessaria una massiccia duplicazione dei formati di analisi
  - ❑ Consumo di spazio disco
  - ❑ La produzione di NTuple domina l'occupazione delle code di analisi
  - ❑ Proliferazione di formati di Ntuple per ogni analisi
    - ❑ Nessuna supervisione o supporto alla produzione di ntuple
    - ❑ Spreco di manpower
- ❑ Pointer skims
  - ❑ Data servers agli originali spesso overloaded
  - ❑ Skimmed data non possono essere esportati
  - ❑ Algoritmi 'pesanti' (combinatori) ri-eseguiti leggendo gli skims
- ❑ Nessuna connessione tra analisi e ricostruzione:
  - ❑ Nuovi algoritmi e/o costanti necessitano un full reprocessing
  - ❑ Il codice di analisi non puo' essere back ported alla ricostruzione
  - ❑ Detector-level analysis quasi impossibile

running experiments : il CM2 di Babar

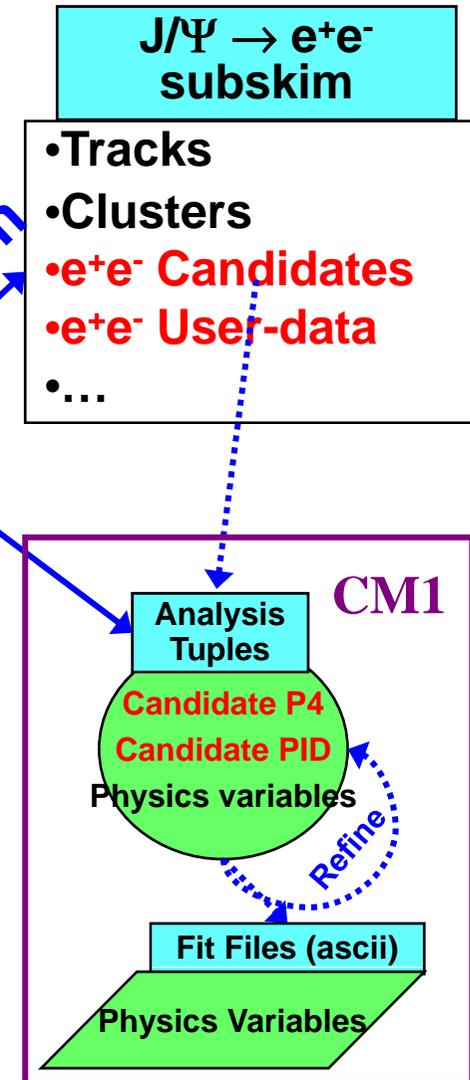
# Reconstruction



# 'Skim'



# Analysis



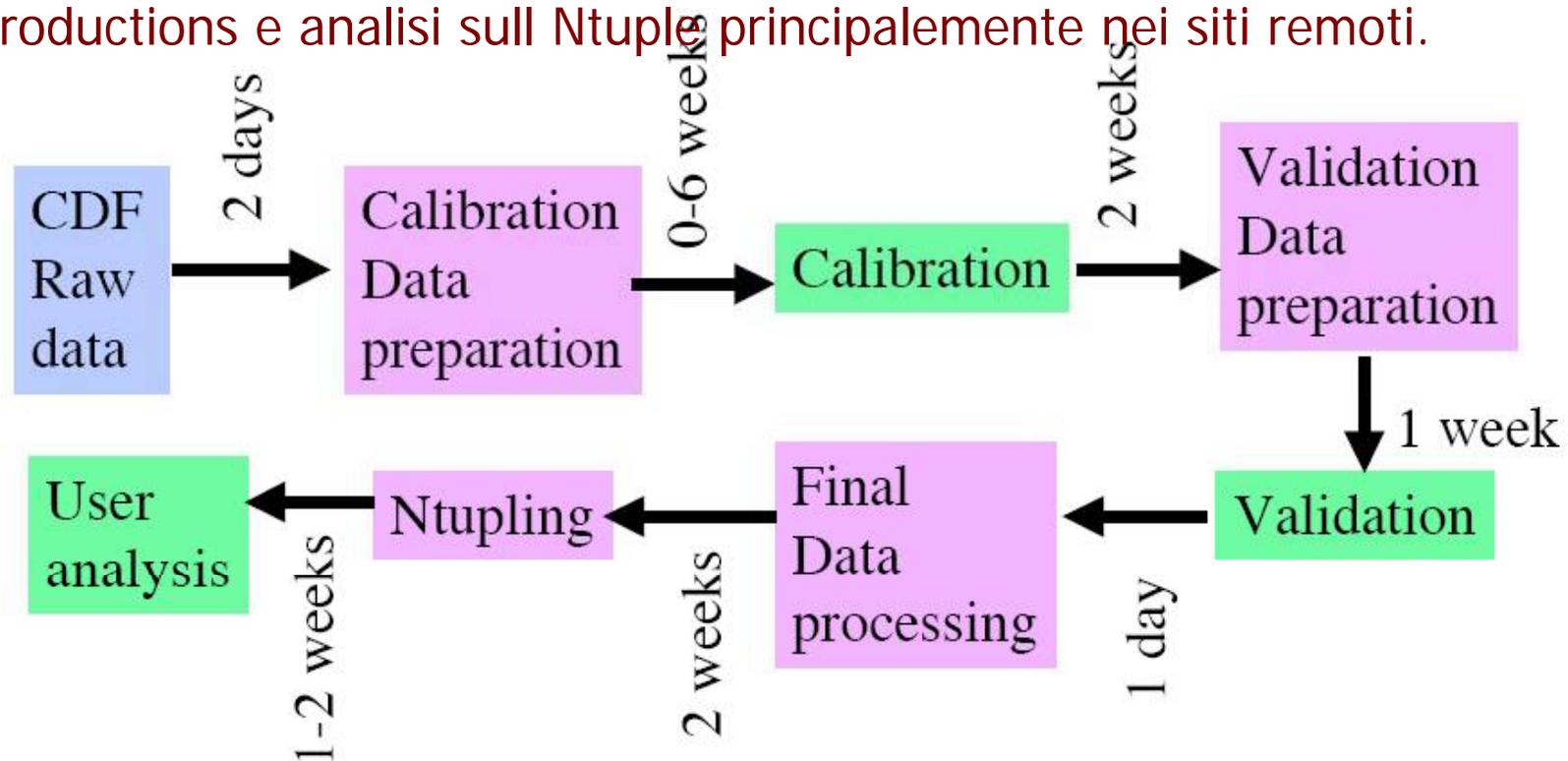
Reskim

## running experiments : goals del CM2 di Babar

- ❑ Consolidare la ricostruzione e I formati di analisi
- ❑ Supportare il deep-copy skimming
  - ❑ La 'profondita' della copia specificata dall'utente nell'analisi
  - ❑ Componenti non copiati ancora accessibili per referenza
- ❑ Permettere agli utenti di customizzare l'output
  - ❑ Storare oggetti compositi e user data
- ❑ Provvedere varie opzioni di accesso ai dati:
  - ❑ Permettere all'utente di scegliere il livello di dettaglio
  - ❑ Supportare l'accesso diretto in interattivo ai dati di produzione
- ❑ Mantenere la piena compatibilita' con il codice di analisi esistente:
  - ❑ Introdurre gli improvements senza compromettere le analisi esistenti

## running experiments : CDF

- ❑ Sistema centralizzato di ricostruzione, Ntupling e storage dei raw data (FNAL)
- ❑ Il formato dei dati e' ROOT based a tutti gli stage di processamento.
- ❑ Sistema centralizzato: i dati sono disponibili per l'utente dopo 6-12 settimana dal recording.
- ❑ MC productions e analisi sull Ntuple principalmente nei siti remoti.



## running experiments : CDF

- ❑ Produzione delle 'master ntuples' per l'analisi avviene in maniera centralizzata e coordinata
  - ❑ Il contenuto delle ntuple e' negoziato tra i vari gruppi di fisica
  - ❑ Ottimizzazione risorse di storage e computing: i task piu' comuni (vertexing, b-tagging, ulteriori jet algos...) sono effettuati durante l'ntupling
  - ❑ re-ntupling time tipico 6-12 mesi
  
- ❑ Le 'master ntuples' sono la base per l'analisi
  - ❑ gli utenti estraggono sotto-ntuple (thinning/slimming) dalle main ntuples per la 'laptop analysis'
  - ❑ Raramente gli utenti tornano ai raw data: nuovo processing centrale
  - ❑ Esiste un framework per l'analisi dentro ROOT
  
- ❑ Il sistema centralizzato di produzione ha aspetti positivi:
  - ❑ elimina la duplicazione degli sforzi e beneficia dello sharing del codice
  - ❑ Assicura la riproducibilita' delle analisi!!

## running experiments : D0

Solo 2 hints dall'esperienza di analisi e computing di D0:

- ❑ Esempio di formato non efficiente: il "thumbnails"
  - ❑ 2001 primi dati: ntuple PAW copmpatibili prodotto con un eseguibile comune d0analyze
  - ❑ 2002/2003 : thumbnails come formato ufficiale per l'analisi. Accesso lento. I gruppi di fisica producono private ntuples
    - ❑ Proliferazione di formati di analisi e di codice per produrre le ntuple
    - ❑ Circa impossibile confrontare diverse analisi
  
- ❑ Formato di analisi comune e produzione centralizzata:
  - ❑ fine 2004: si decide di adottare un formato ROOT-based ("everybody wants to use ROOT at the end")
  - ❑ La produzione di ntuple viene organizzata centralmente
  - ❑ Sviluppo di un framework di analisi comune (cafe) utilizzabile anche in standalone : permette la condivisione di codice comune!

## La lezione degli esperimenti

- ❑ La velocità di accesso ai dati è il fattore guida nei modelli di analisi adottati dagli utenti
  - ❑ indipendentemente dalle indicazioni di management e developers
- ❑ I tasks basati su analisi al livello "ESD" o il reprocessing (eg: calibrations, allineamenti, tracks fits, re-clustering) sono di norma eseguiti al livello più alto dell'analisi
- ❑ Man mano che l'Analysis model evolve
  - ❑ L'ESD si gonfia e diventa di difficile accesso → dropped
  - ❑ "AOD" viene aumentato con alcune grandezze dell' "ESD" (eg: hits in roads, calo cluster cells) per avere più possibilità in fase di analisi
  - ❑ In generale si è osservato un proliferare di formati di ntuple e l'impatto di ciò sul CM è contrastato da:
    - ❑ Produzione centralizzata di "Ntuples"
    - ❑ Permettendo l'accesso via ROOT agli "AOD"-equivalenti

## La lezione degli esperimenti

### ❑ Dropping degli ESD, incremento degli AOD

- ❑ Chiaramente questo processo funziona a patto di mantenere la size degli AOD ad un livello tale che gli AOD stessi siano facilmente disponibili.
- ❑ Molti tasks di calibrazione e allineamento richiedono dettagli in ogni caso impossibili da inserire negli AOD: gli ESD sono necessari per subsets di dati.

### ❑ In-Framework vs Out-of-Framework Analysis

- ❑ FW e' necessario per alcune ricalibrizioni (DB access e tools) e tasks complessi (eg Jet finding, b-tagging).
- ❑ FW fornisce un environment comune per i tools di analisi
- ❑ La familiarita' con ROOT e la velocita' di accesso spinge gli utenti verso una l'ntuple analysis.

### ❑ Central vs Private Derived Physics Data (DPD) production

- ❑ Le strategie verranno determinate con l'esperienza dentro la comunita' dei fisici.
- ❑ Tools comuni per la produzione di DPD garantiscono l'uniformita' tra i diversi DPD.

## Diversi punti di vista....

- Gli utenti sono preoccupati dalla complessita' del framework. In genere vogliono soltanto:
  - Un'ntuple con la quale possano fare tutto cio' che vogliono in ROOT.
  - Deve essere il piu' veloce possibile
  - Deve essere semplice
- I Developers e il management sono preoccupati
  - Varie copie delle Ntuple possano diventare un peso per il CM.
  - Gli utenti a volte sottostimano la complessita' delle analisi: con il tempo arrivano a ri-creare da se il framework ri-scrivendosi tools che gia' esistono nel framework.
    - Senza framework non si puo' avere accesso a tools comuni debuggati e ottimizzati
  - Senza framework diventa quasi impossibile mantenere codice comune e garantire la riproducibilita' delle analisi.

## L'analysis model di CMS

CMS ha raccolto appieno l'esperienza di BaBar e CDF : la parola d'ordine e' "standardizzazione":

❑ tutte le collezioni, dai RAW data agli AOD e anche le collezioni definite dall'utente sono accessibili sia in batch che in interattivo con ROOT.

❑ L'EDM puo' essere usato come formato finale di analisi, no ntuples!

❑ Interfacce uniformi agli oggetti ricostruiti ovunque (reco/aod) : dovunque si definisce pt() o getPt() per accedervi

❑ L'uniformita' permette di scrivere algoritmi generici (selettori, filtri...) validi per oggetti diversi via templates in modo molto semplice.

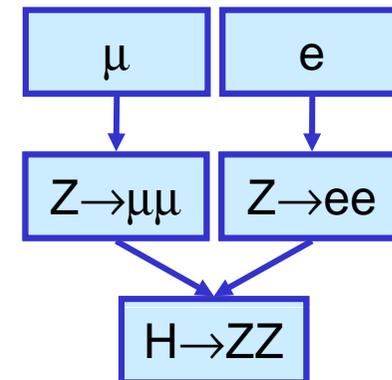
```
> gSystem->Load("libFWCoreFWLite")
> AutoLibraryLoader::enable()
> TFile f("reco.root")
> Events.Draw("tracks.phi() -
  tracks.outerPhi() :
  tracks.pt()",
  "tracks.pt()<10",
  "box")
```

## L'analysis model di CMS: il modello a particle candidates

- Modello a 'particle candidates':
  - Stabilire un linguaggio comune per le analisi
  - Interfaccia comune a tool di analisi (fitters, combiners..)
- Un processo di analisi viene decomposto in steps intermedi:
- Ogni step produce una collezione intermedia di *Candidates*

– Es.:  $H \rightarrow ZZ \rightarrow \mu\mu ee$ :

- Scegliere le collezioni di muoni ed elettroni standard
- Ricostruire la  $Z \rightarrow \mu\mu$  da una collezione di  $\mu$
- Ricostruire la  $Z \rightarrow ee$  da una collezione di  $e$
- Ricostruire l' $H \rightarrow ZZ$  da  $Z \rightarrow \mu\mu$  e  $Z \rightarrow ee$



## L'analysis model di CMS: modello a particle candidates

```

process Analysis = {
  source =
    PoolInputService {
      int32 maxEvents = 50
      string fileName = "aod.root"
    }

  module allTracks =
    TrackCandidateProducer {
      InputTag src = "ctfWithMaterialTracks"
    }

  module goodTracks =
    PtMinCandSelector {
      InputTag src = "allTracks"
      double ptMin = 3.0
    }

  module ZCandidates =
    CandCombiner {
      string decay =
        "goodTracks@+ goodTracks@-"
      string cut = "86.0 < mass < 96.0"
    }

  module HiggsCandidates =
    CandCombiner {
      string decay =
        "ZCandidates ZCandidates"
      string cut = "mass < 600.0"
    }

  module out =
    PoolOutputModule {
      string fileName = "canddst.root"
      untracked vstring productsSelected = {
        "drop *",
        "*_ctfWithMaterialTracks_*_*",
        "*_goodTracks_*_*",
        "*_ZCandidates_*_*",
        "*_HiggsCandidates_*_*"
      }
    }

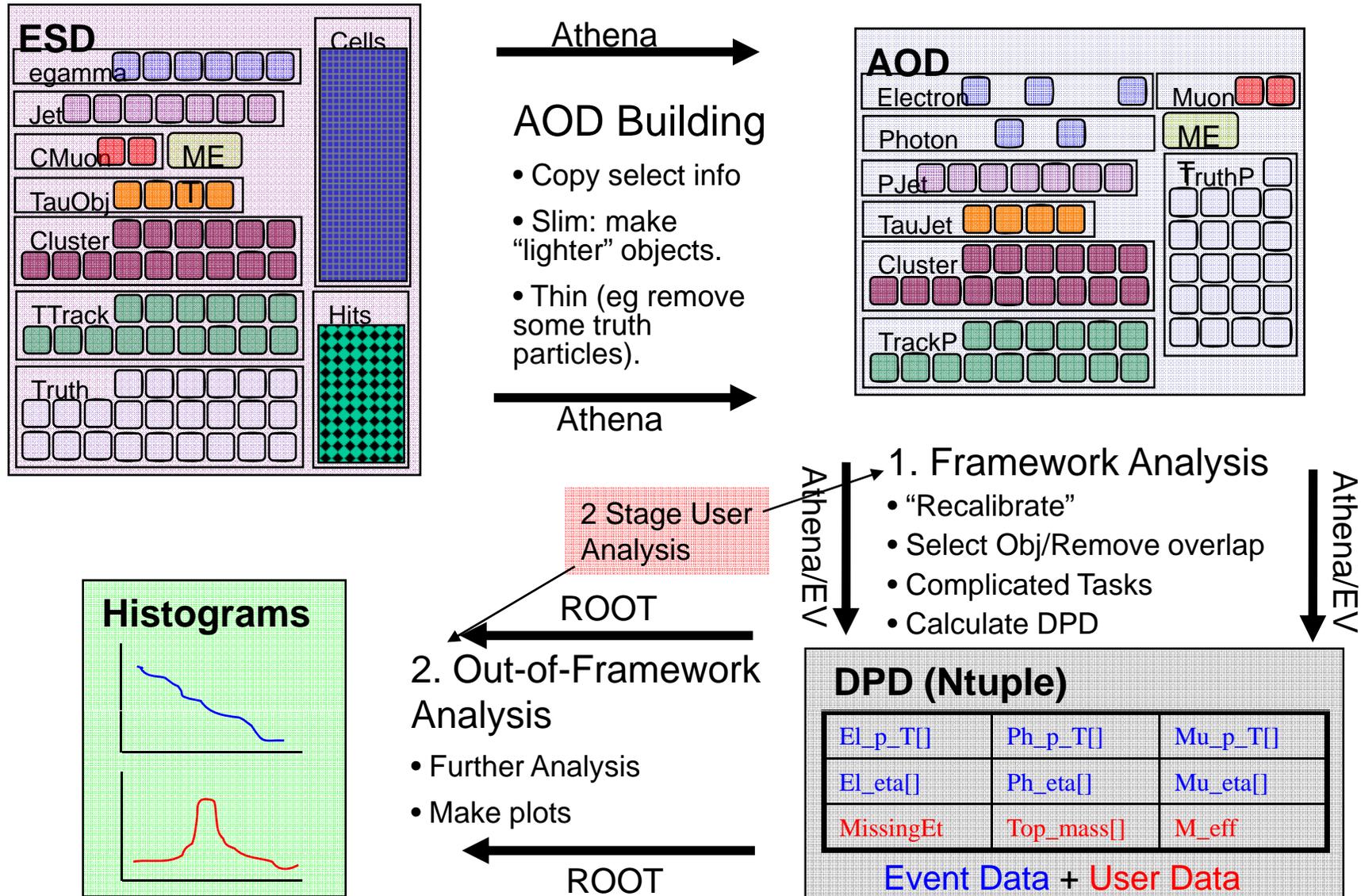
  path p = {
    allTracks, goodTracks,
    ZCandidates, HiggsCandidates
  }
  endpath o = { out }
}

```

## Analysis model di CMS: il CSA06

- ❑ Esercizio di test del flow della ricostruzione e dei dati su un set to 50 milioni di eventi
  - ❑ Un test al **25%** della capacita' richiesta nel 2008i
- ❑ Flow del CSA06:
  - ❑ Simulazione dei datases con HLT-tags)
  - ❑ Ricostruzione prompt al Tier-0:
    - ❑ Ricostruzione a 40 Hz (su 150 Hz) usando CMSSW (software ufficiale)
    - ❑ Applicazione delle calibrazioni dal database dell'offline
    - ❑ Generazione di Reco e AOD
    - ❑ Streaming in physics datasets (5-7)
  - ❑ Distribuzione di tutti gli AOD & alcuni FEVT a tutti i Tier-1s partecipanti
  - ❑ Test di ricostruzione di alcuni FEVT al Tier-1s
  - ❑ Re-reconstruction al Tier-1s (per testare la calibrazione)
  - ❑ Jobs di skim runnati al Tier1 con dati propagati ai Tier2
  - ❑ Jobs di fisica ai Tier-2s su AOD e Reco

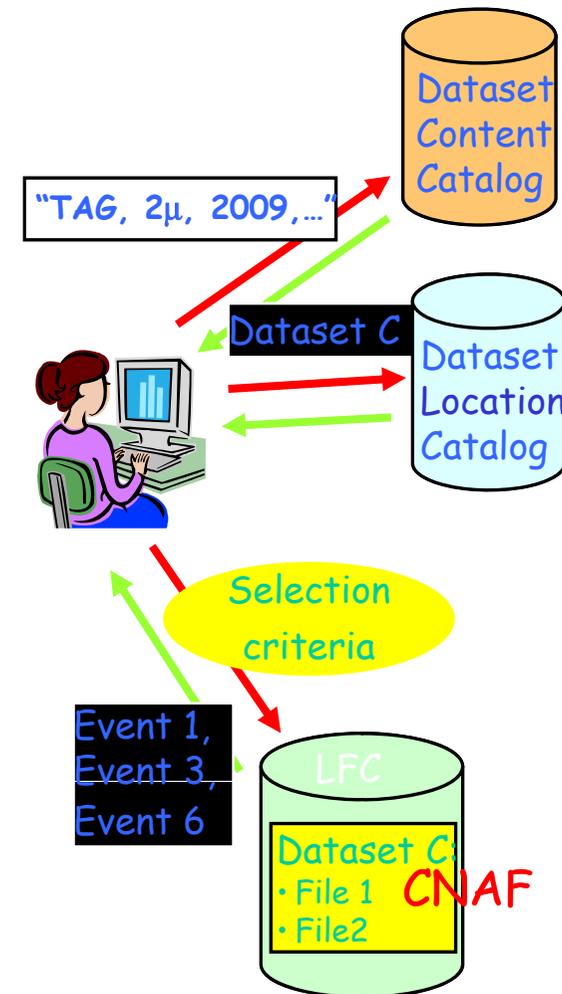
## Il computing model di ATLAS:



## L'analysis model di ATLAS: l'analisi distribuita

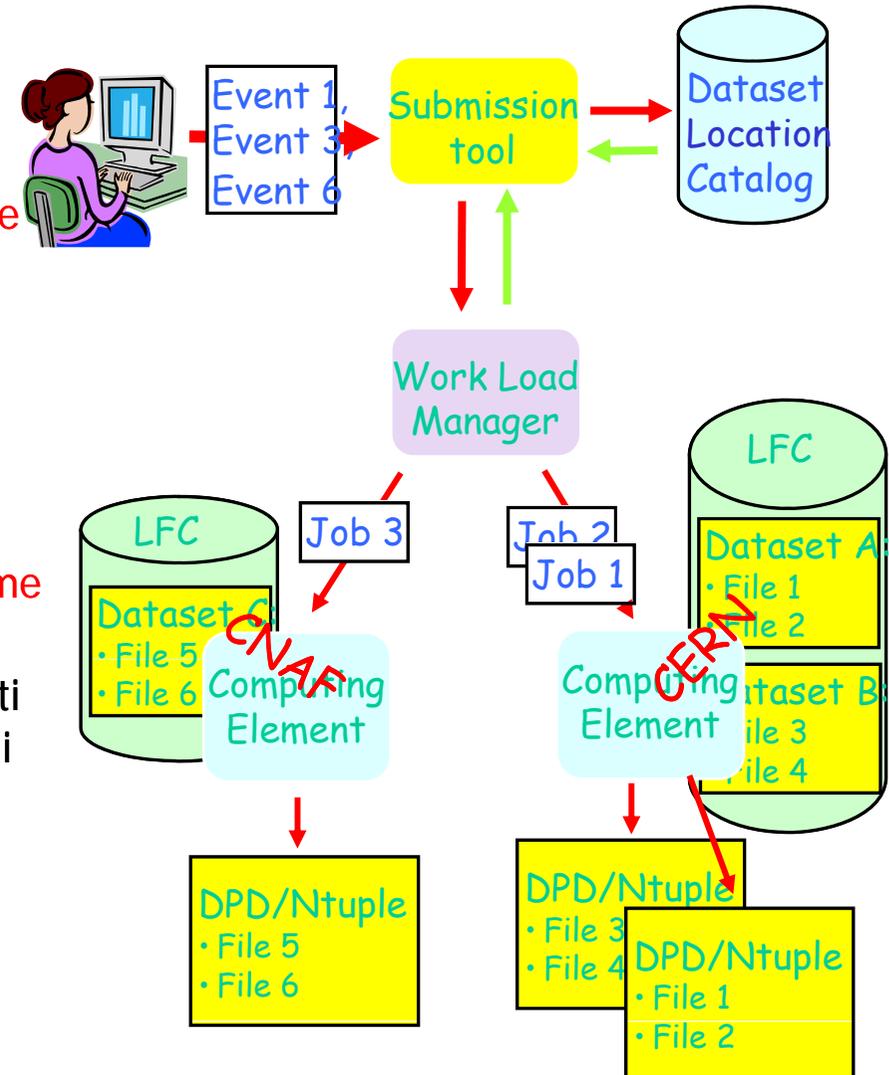
### Selezione TAG based

- Apertura di una sessione tramite un'interfaccia (GANGA) che permette di eseguire tutte le operazioni d'analisi:
  - Job configuration, submission, splitting, merging, monitoring, output retrieval
- Interrogazione del **Dataset Content Catalog** che contiene i metadata per ogni dataset del tipo desiderato per trovare quelli che gli interessano
  - Esempio di query: dammi la lista dei dataset con trigger  $2\mu$  del 2009, versione x.y.z del software etc....
- Localizzazione tramite il **Dataset Location Catalog** del sito (*cloud*) dove risiede il dataset
- Passaggio dai dataset ai singoli files tramite il **Local File Catalog** presente in ogni Tier1 della *cloud*
- Applicazione dell' algoritmo di selezione sui dataset scelti e produzione di una lista di eventi accettati
  - In Athena è disponibile il tool che per ogni dataset permette di processare solo gli eventi specificati via TAG list



## L'analysis model di ATLAS: l'analisi distribuita

- Selezionati gli eventi, sottomissione alla Grid dei job di analisi tramite il Work Load Manager (WLM), nei siti dove risiedono gli eventi
  - Il Dataset Location Catalog localizza i siti dove risiedono i dataset contenenti gli eventi accettati e con il Local File Catalog, si passa ai singoli files residenti sugli SE
- Un job può dare come output una nuova collezione di eventi, che può essere registrata come nuovo dataset nei cataloghi
  - In DQ2 è possibile sottoscrivere uno o più siti come destinazione del dataset
  - La possibilità di generare nuovi dataset consistenti sarà ristretta (almeno inizialmente) ai responsabili delle produzioni dei gruppi di fisica
- Estrazione dall'insieme dei dati analizzati (in formato AOD) dei file di Derived Physics Data (tipo n-tupla) che potrà poi essere analizzare interattivamente in locale.





---

## ATLAS: il modello ideale

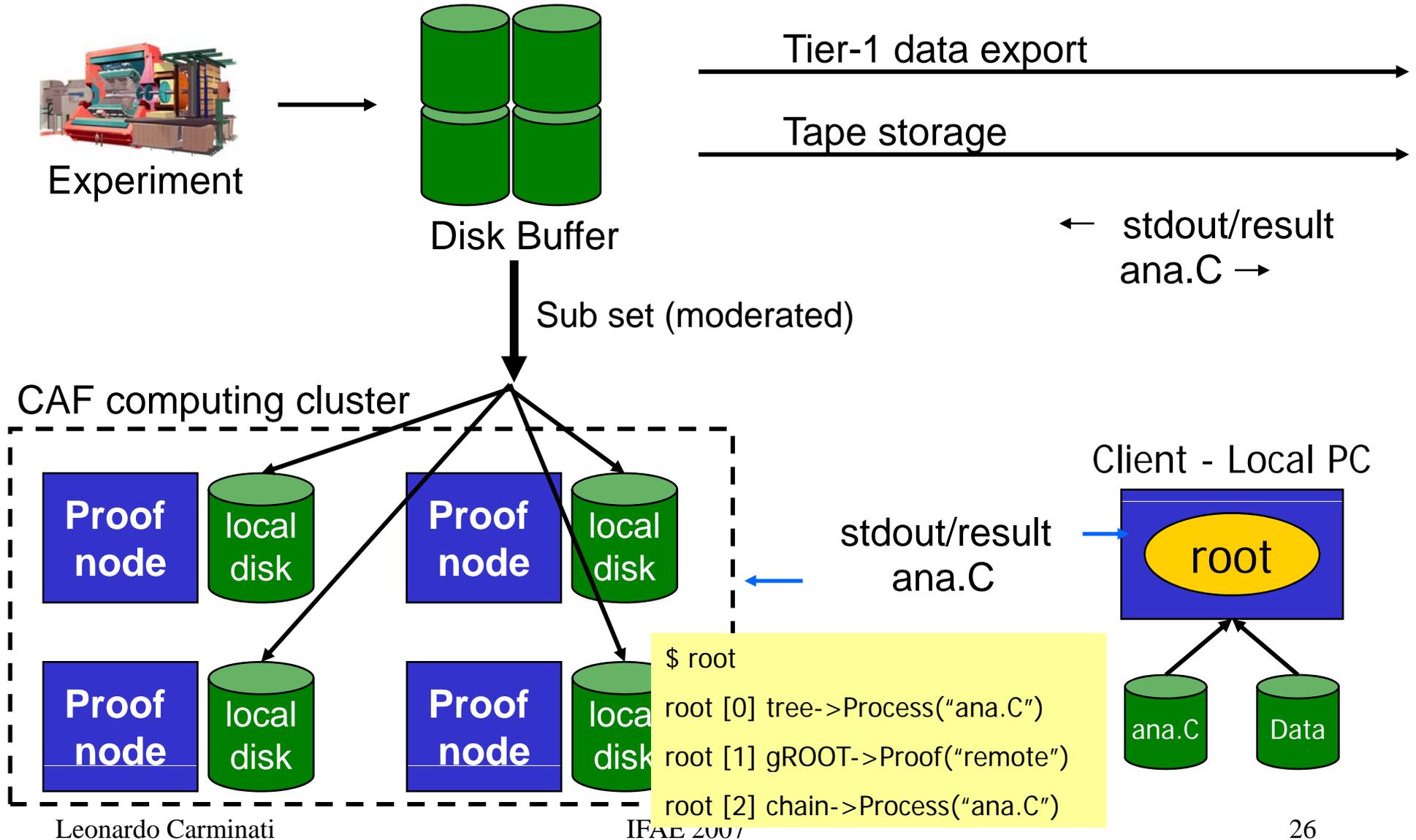
- ❑ Nel modello di analisi attuale AOD non sono leggibili da ROOT
  - ❑ ATLAS ha scelto un modello per sostenere la schema evolution basato sulla separazione Transient&Persistent difficile da esportare in ROOT
- ❑ Possibilita' di leggere gli AOD da ROOT: accesso alle classi transienti a partire da quelle persistenti.
- ❑ I vantaggi sono evidenti:
  - ❑ No ntuple centralizzate: l'AOD e' gia' ROOT readable (CM e' al sicuro).
  - ❑ L'utente puo' usare direttamente ROOT senza usare il framework...
  - ❑ Oppure usare il framework e avere molti benefici:
    - ❑ Costruire DPD piu' semplici e usare un set di tool comuni
  - ❑ Modello di analisi elegante dove input/output sono unificati simile a CMS e BaBar mantenendo la schema evolution:
    - ❑ ESD → Athena → AOD
    - ❑ AOD or DPD → ROOT → Plots
    - ❑ AOD → Athena → DPD
    - ❑ DPD → Athena → DPD

## Alice: soluzioni PROOF-based

Dati e framework di ricostruzione e analisi (AliROOT) tutti ROOT based. Modello a Tiers e analisi distribuita simile ad ATLAS e CMS

- ❑ Alice utilizzerà PROOF su una **CERN Analysis Facility (CAF)** per i tasks da runnare su una time scale corta
  - ❑ Prompt analysis di dati pp e pilot analysis di dati PbPb
  - ❑ Calibrazioni e allineamenti
- ❑ Design goals
  - ❑ 500 CPUs + 200 TB di dati selezionati disponibili localmente
- ❑ L'uso di PROOF e' trasparente per l'utente: lo stesso codice puoà essere runnato in locale o in un PROOF system
- ❑ La soluzione PROOF non e' legata all'uso di GRID
  - ❑ Puo' accedere ai files su grid
  - ❑ In fase di studio se/come i Tier possano venir utilizzati come CAFs

## Alice: soluzioni PROOF-based



---

## Conclusioni e prospettive

❑ Non e' facile proporre conclusioni univoche vista la messe di soluzioni adottate dai vari esperimenti (e talvolta dai vari gruppi negli exp)

- ❑ What went wrong with 'thumbnails'? 3 main reasons: speed, speed, speed"
- ❑ People who had not used the D0 offline framework would avoid it whenever possible, i.e. if *any* alternative was around.
- ❑ Framework perceived as too complex (despite tutorials etc.)

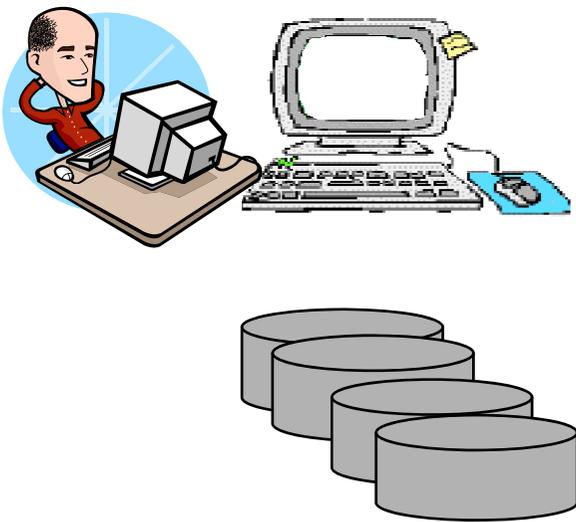
R. Hauser, "The D0 analysis model", Analysis Model Workshop (CERN25/10/2006)

❑ Standardizzazione dei formati di dati e la conseguente modularita' del framework sono elementi chiave

- ❑ accesso ai dati in modo simile e semplice ad ogni livello: viene limitato il ricorso a inventarsi formati di dati per l'analisi.
- ❑ equivalenza e intercambiabilita' dell'analisi in-framework e off-framework e portabilita' del codice: rende il framework piu' "amico"
- ❑ facilita il code-sharing e conseguentemente la riproducibilita' delle analisi

---

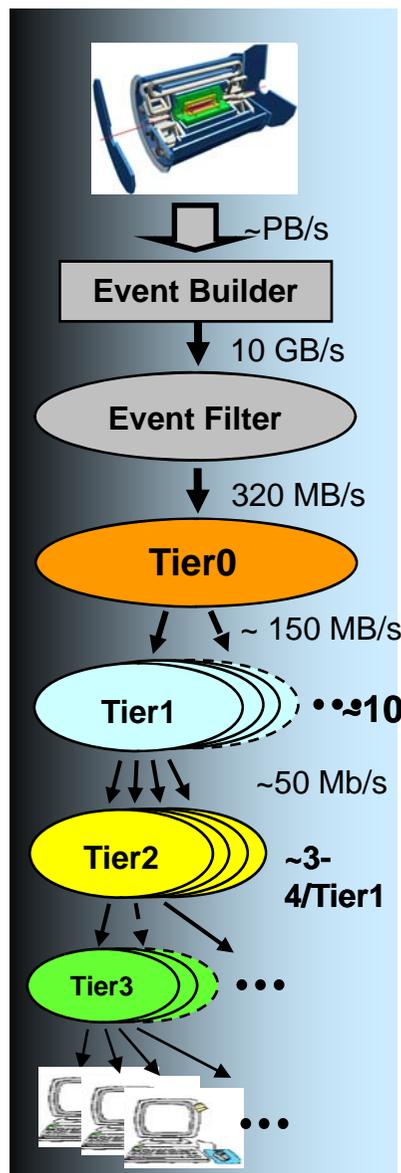
# Conclusioni e prospettive



## Computing model

Il modello generale di calcolo per l'offline e l'analisi di ATLAS è quello gerarchico multi-Tier.

**Modello a *cloud*:** ad ogni Tier-1 sono associati alcuni (3 o 4) Tier-2 spesso i base a considerazioni geografiche.



### Tier-0 (CERN)

- Archivio dei RAW data ricevuti dall'EF e distribuzione ai Tier1
- Prompt Reconstruction delle calibration e express stream
- Prompt Reconstruction dell'event stream
- Distribuzione output ricostruzione (ESD,AOD,TAG) ai Tier-1

### Tier-1 (10)

- Accesso a lungo termine e archivio di un subset di RAW data
- Copia dei RAW data di un altro Tier-1
- Reprocessing della ricostruzione dei propri RAW data con parametri di calibrations e allineamenti finali e distribuzione AOD ai Tier-2
- Archivio dati simulati MC prodotti nei Tier-2

### Tier-2

- Simulazione Monte Carlo
- Analisi

## L'analysis model di ATLAS: l'analisi distribuita

### ■ Uso delle risorse per l'analisi

- I Tier-2 ospitano job di analisi sia individuale che di gruppo
  - ✓ Il CM prevede che il 50% delle risorse di CPU siano dedicate all'analisi
- I Tier-1 possono accogliere job di analisi di gruppo

### ■ Analisi distribuita:

1. Selezione degli eventi da TAG e analisi sugli AOD degli eventi selezionati
2. Determinazione dei siti dove i dati sono memorizzati
3. Invio in questi siti (tramite Grid tools) dei jobs ed estrazione delle informazioni più rilevanti:
  - ✓ nella forma AOD piu' leggeri che consentano un'ulteriore analisi nel fw
  - ✓ nella forma di DPD (ntuple) da usare localmente in modo interattivo