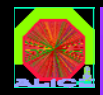
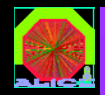


ALICE Data Analysis Model

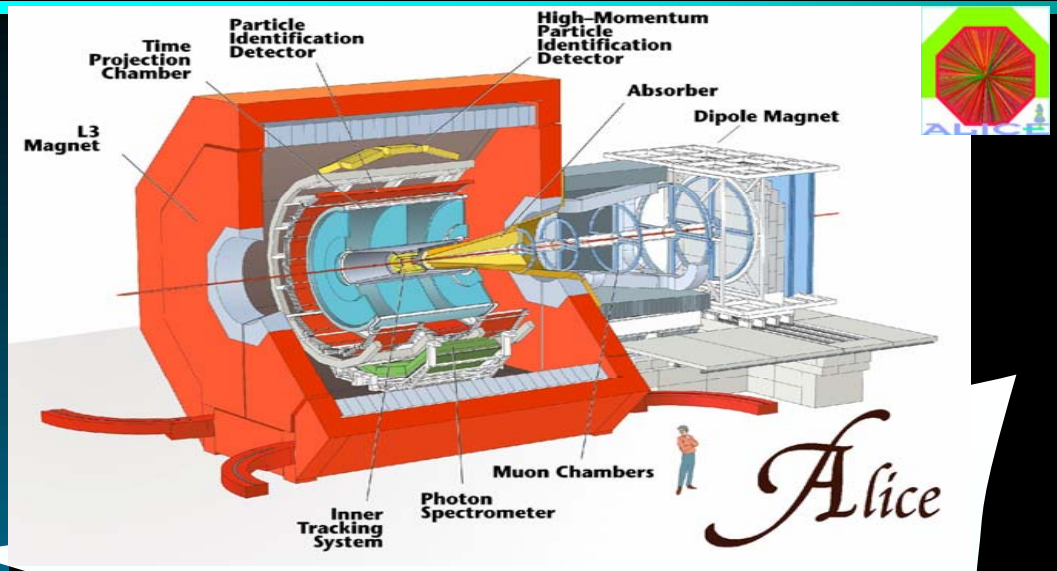
Federico Carminati

Root Workshop 2005

Thanks



- P.Buncic, D.Feichtinger, A.Peters, F.Rademakers, P.Saiz



ALICE Collaboration

- ~ 1/2 ATLAS, CMS, ~ 2x LHCb
- ~1000 people, 30 countries, ~ 80 Institutes

Total weight	**10,000t**
Overall diameter	**16.00m**
Overall length	**25m**
Magnetic Field	**0.4Tesla**

8 kHz (160 GB/sec)

level 0 - special hardware

200 Hz (4 GB/sec)

level 1 - embedded processors

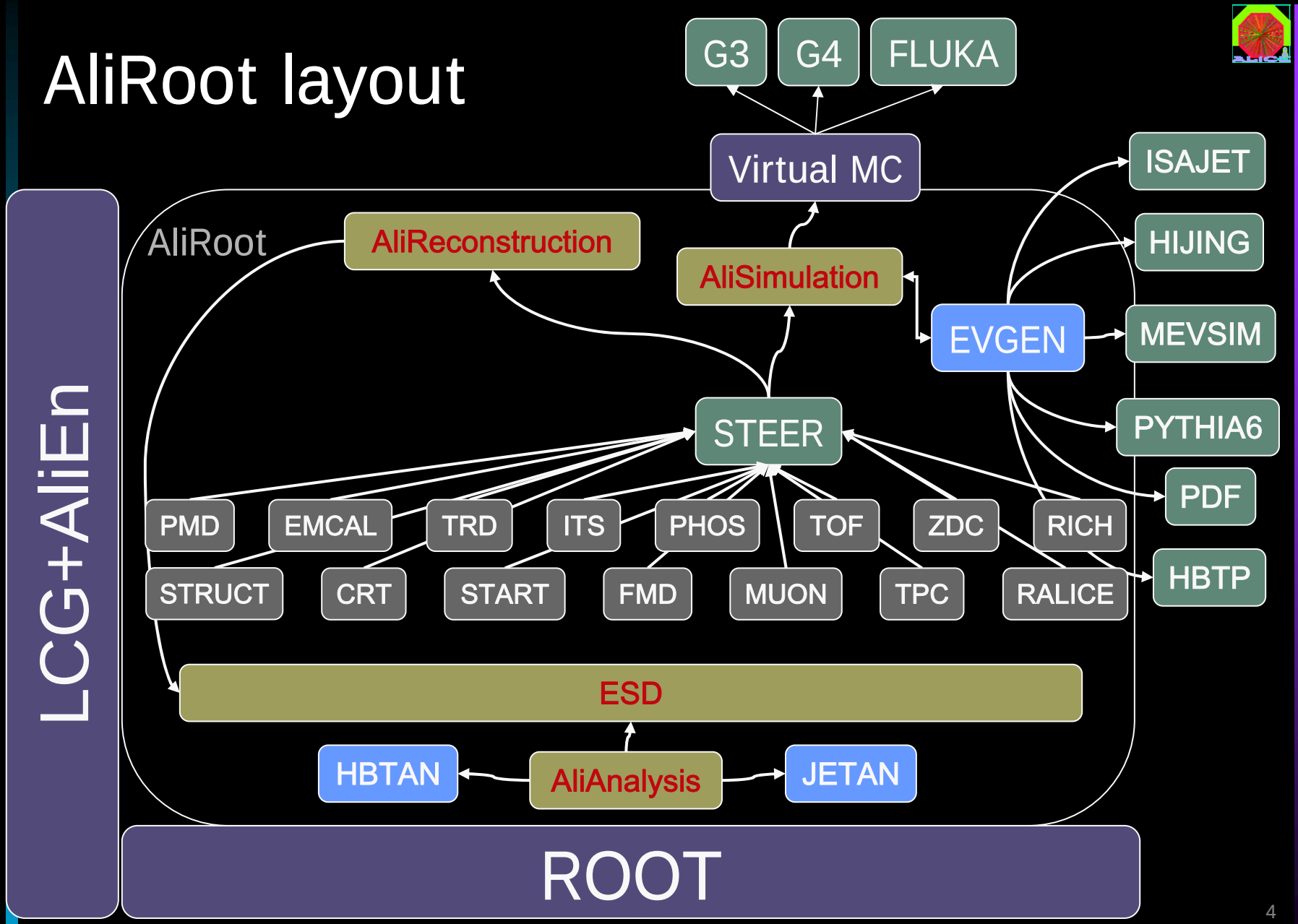
30 Hz (2.5 GB/sec)

level 2 - PCs

30 Hz (1.25 GB/sec)

data recording & offline analysis

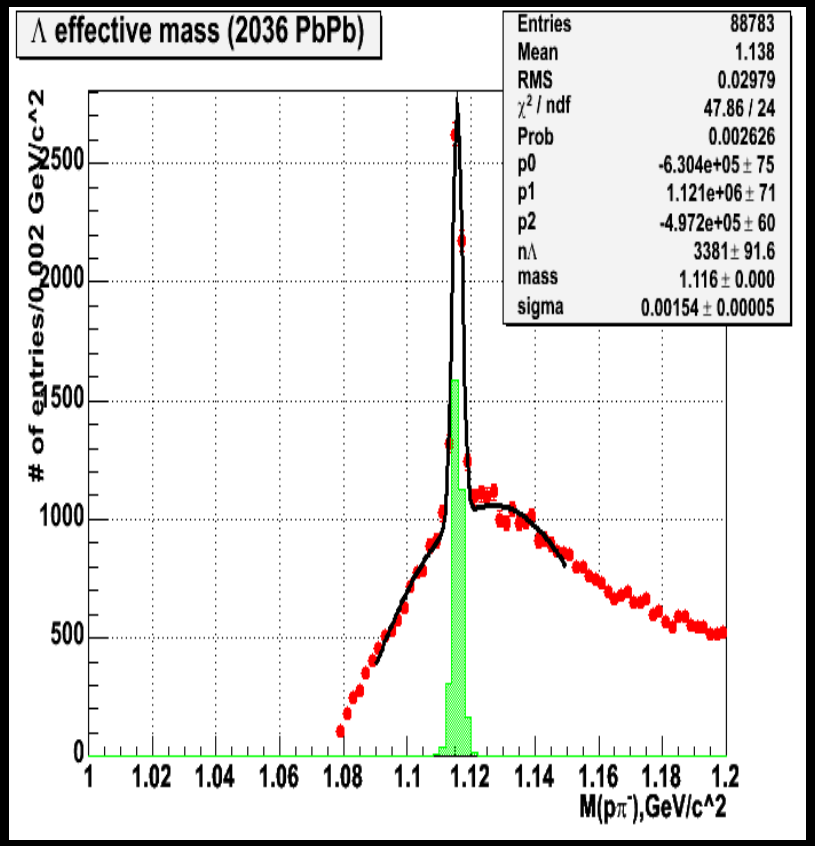
AliRoot layout

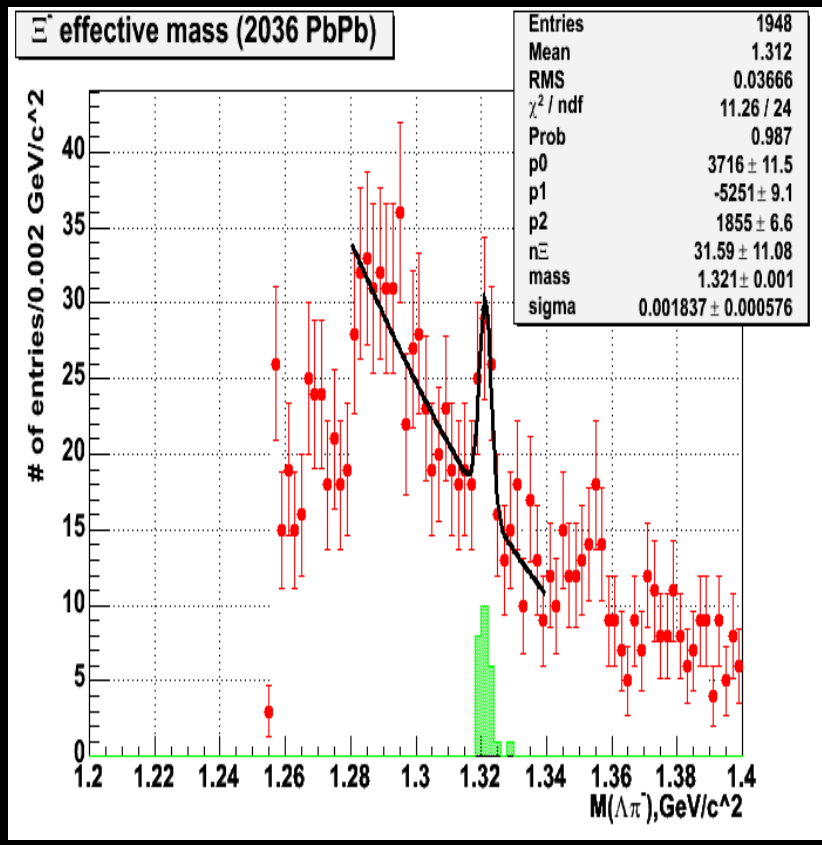


AliRoot: Current Status

- Up-to-date description of ALICE detectors
  - TGeo
- Rich set of event generators, easily extensible
- Possibility to use different transport packages
  - VMC
- User friendly steering classes for simulation and reconstruction
- Efficient track reconstruction
- Combined PID based on Bayesian approach
- ESD classes for analysis and fine-tune calibration
- Analysis examples to explore wide spectrum of heavy-ion and pp physics

$\Xi \rightarrow \pi\Lambda \rightarrow p\pi$





ALICE computing model

- pp
  - Quasi-online data distribution and first reconstruction at T0
  - Further reconstructions at T1's
  - 10 days' buffer
- AA
  - Calibration, alignment and pilot reconstructions during data taking
  - Data distribution and first reconstruction at T0 during four months after AA
  - Further reconstructions at T1's
  - One day's buffer
- One copy of RAW at T0 and one distributed at T1's

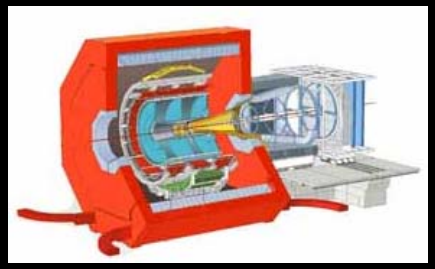
ALICE computing model

- T0
  - Does: first pass reconstruction
  - Stores: one copy of RAW, calibration data and first-pass ESD's
- T1
  - Does: reconstructions and scheduled analysis
  - Stores: second collective copy of RAW, one copy of all data to be kept, disk replicas of ESD's and AOD's
- T2
  - Does: simulation and end-user analysis
  - Stores: disk replicas of ESD's and AOD's
- Three kinds of data analysis
  - Fast pilot analysis of the data "just collected" to tune the first reconstruction at T0 Analysis Facility
  - Ordered "establishment" analysis
  - End-user analysis

ALICE Analysis Basic Concepts

- Analysis Models
  - Prompt analysis at T0 using PROOF(+file catalogue) infrastructure
  - Batch Analysis using GRID infrastructure
  - Interactive Analysis using PROOF(+GRID) infrastructure
- User Interface
  - ALICE User access any GRID Infrastructure via AliEn or ROOT/PROOF UIs
- AliEn
  - Native and “GRID on a GRID” (LCG/EGEE, ARC, OSG)
  - integrate as much as possible common components
    - LFC, FTS, WMS, MonALISA ...
- PROOF/ROOT
  - single- + multitier static and dynamic PROOF cluster
  - GRID API class TGrid(virtual)->TAliEn(real)

Prompt analysis



DAQ

Castor2

AliEn SI

CAF

Prompt analysis

- We estimate a capacity of 7MSI2k to be needed for fast pilot analysis and calibration
- A large PROOF-enabled cluster that will read from the DAQ disk buffer while data waits to be taped
- Limited use of metadata and tags
- This is a “classical” PROOF application that we will start prototyping next year
- Main challenges will be access to data

Distributed analysis

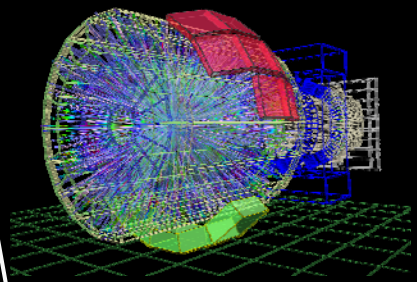
File Catalogue query

User job (many events)

Data set (ESD's, AOD's)

Job output

Job Optimizer



Sub-job 1

Sub-job 2

Sub-job n

Grouped by SE files location

Job Broker

Submit to CE with closest SE

CE and SE
processing
Output file 1

CE and SE
Output file 2

CE and SE
Output file n

File merging job

Tag architecture

Clients

Administrator
- Fetch tag file
- Load subset
- Rollback
- Commit

Analysis
- **New query**

From STAR GC

Index Builder
In: ALICE tag file
Out: Bitmap index

Event Catalog
In: Queries
Out: GUID, ev ID

Servers

SI
In: GUID
Out: List of SE's

Job Splitting
In: GUID
Out: PFN and protocol

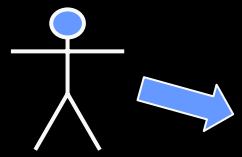
SE's
In: GUID
Out: PFN and protocol

ALICE Analysis Data Distribution

- Sensitivity to errors grows with the size of the system
  - Use local/central concepts, if possible!
  - Keep site/data autonomy, wherever possible
- Analysis Data should be kept in Tier-0-1-2 centres with minimum dispersion within a *set* of data
  - Easier to do bookkeeping for sets of files
- Distributed Production, but focused storage
  - Physics analysis needs a fixed set of data for comparisons
  - Analysis better fail rather than providing results coming from partial data sets

Middleware Services in AliEn

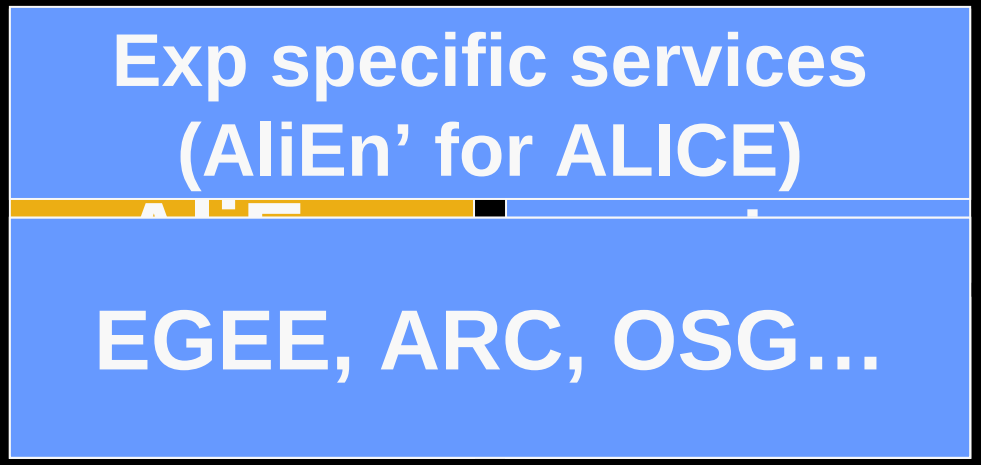
GAS	Grid Access Service
WM	Workload Mgmt
DM	Data Mgmt
RB	Resource Broker
TQ	Task Queue
FPS	File Placement Service
FQ	File Transfer Queue
PM	Package Manager
ACE	AliEn CE (pull)
FC	File Catalogue
JW	Job Wrapper
JA	Job Agent
LRC	Local Replica Catalogue
LJC	Local Job Catalogue
SE	Storage Element
CE	Computing Element
SRM	Storage Resource Mgr
CR	Computing Resource (LSF, PBS,...)



- Many OS components dealing with Grid issues
- Why not build the *minimal GRID* that *does the job*?
  - Fast development of a prototype, no problem in exploring new roads, restarting from scratch etc etc
  - Hundreds of users and developers for the modules
  - Immediate adoption of emerging standards

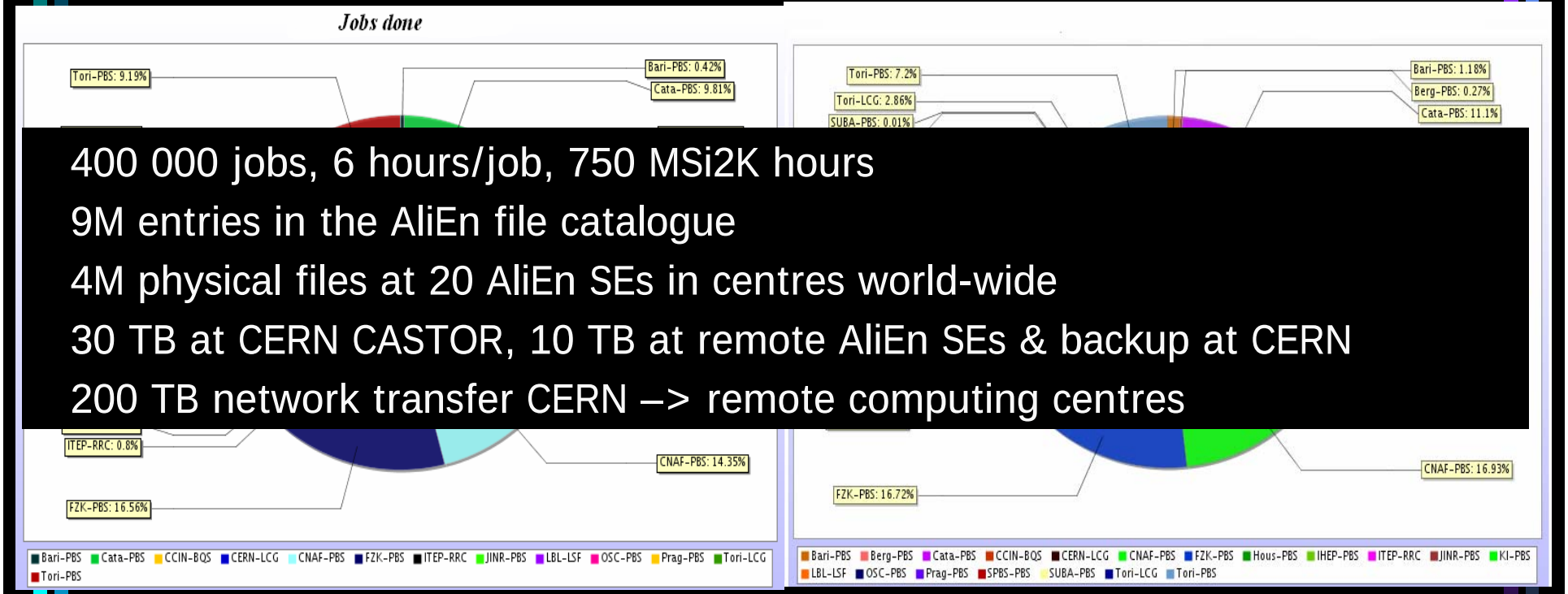
- AliEn (5% of code developed, 95% imported)

ALICE view on the current situation



PDC 04

- Jobs (AliEn/LCG): Phase 1 - 75/25%, Phase 2 – 89/11%
- More operation sites added to the ALICE GRID as PDC progressed



400 000 jobs, 6 hours/job, 750 MSi2K hours

9M entries in the AliEn file catalogue

4M physical files at 20 AliEn SEs in centres world-wide

30 TB at CERN CASTOR, 10 TB at remote AliEn SEs & backup at CERN

200 TB network transfer CERN –> remote computing centres

- 17 permanent sites (33 total) under AliEn direct control and additional resources through GRID federation (LCG)

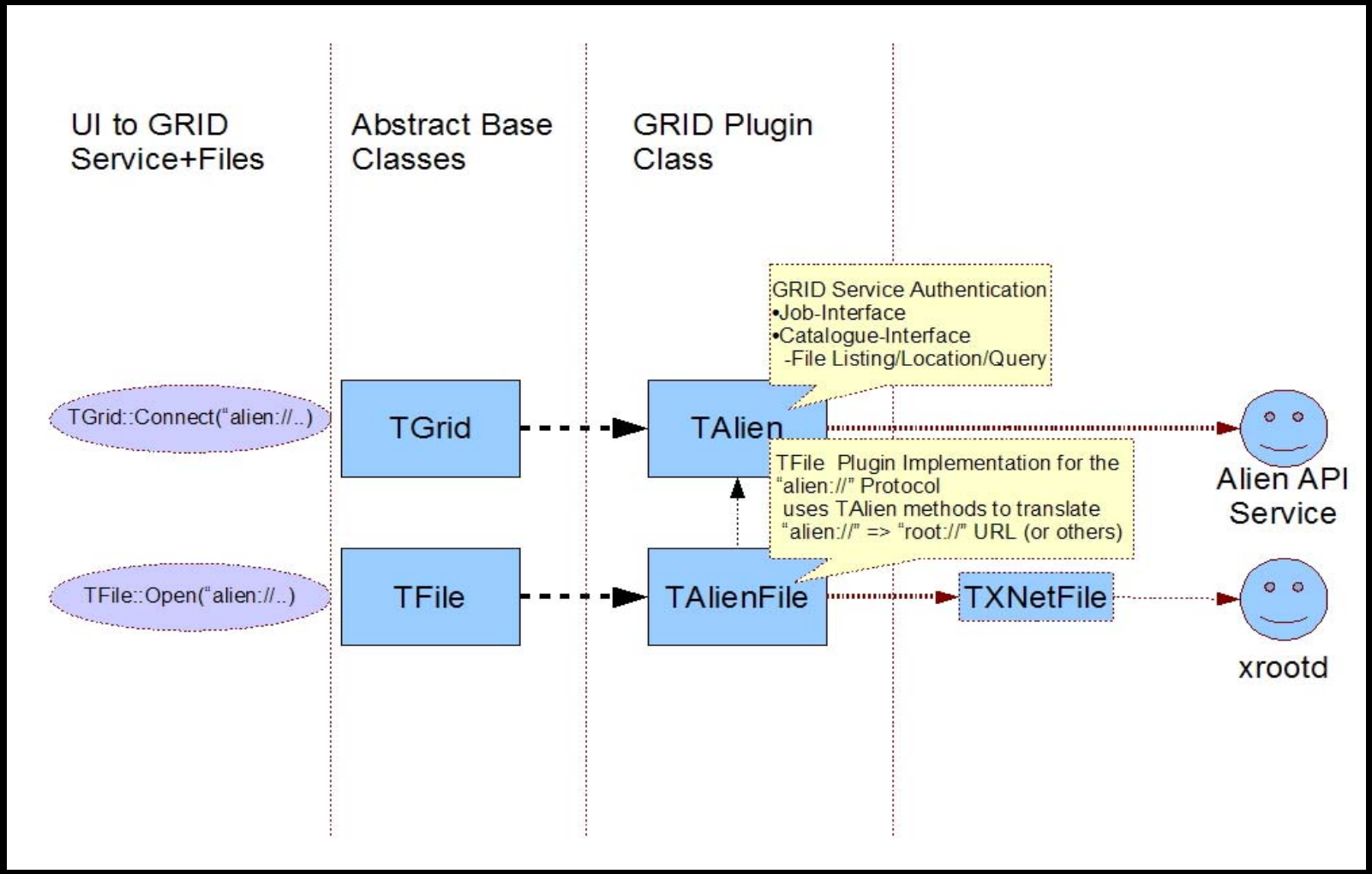
Batch Analysis

Batch Analysis: input

- Input Files
  - Downloaded into the local job sandbox: macros, configuration...
- Input Data
  - Created from Catalogue Queries
  - Stored as ROOT Objects (Tchain, TDSet, TAlienCollection) in a registered GRID file
  - Stored in XML file format in a registered GRID file
  - Stored in a regular AliEn JDL
  - on demand GRID jobs don't stage Input Data into the job sandbox (no download)
    - GRID jobs access Input Data via “xrootd” protocol using the TAlienFile class implementation in ROOT

      TFile::Open(“alien://alice/...../Kinematis.root”);

ALICE Analysis - File Access from ROOT
“all files accessible via LFNs”

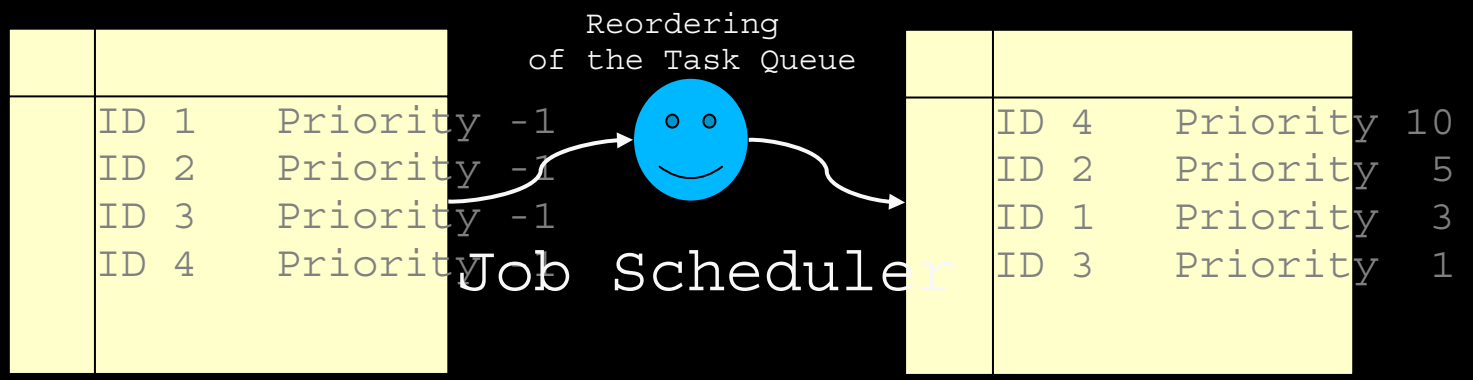


Batch Analysis: Splitting

- Jobs are submitted to the ALICE (AliEn) Task Queue
- JDL's undergo optimization and scheduling
  - Job Optimizer
    - JDLs can be broken up into several jobs (job splitting) based on the Input Data list (file, SE, directory)
  - The AliEn Task Queue implements Masterjobs and corresponding Subjobs
    - Analysis Subjobs can be monitored and referenced via the Masterjob ID
    - Easy handling of hundred or thousands of jobs with a single ID

ALICE AliEn Batch Analysis: Scheduling

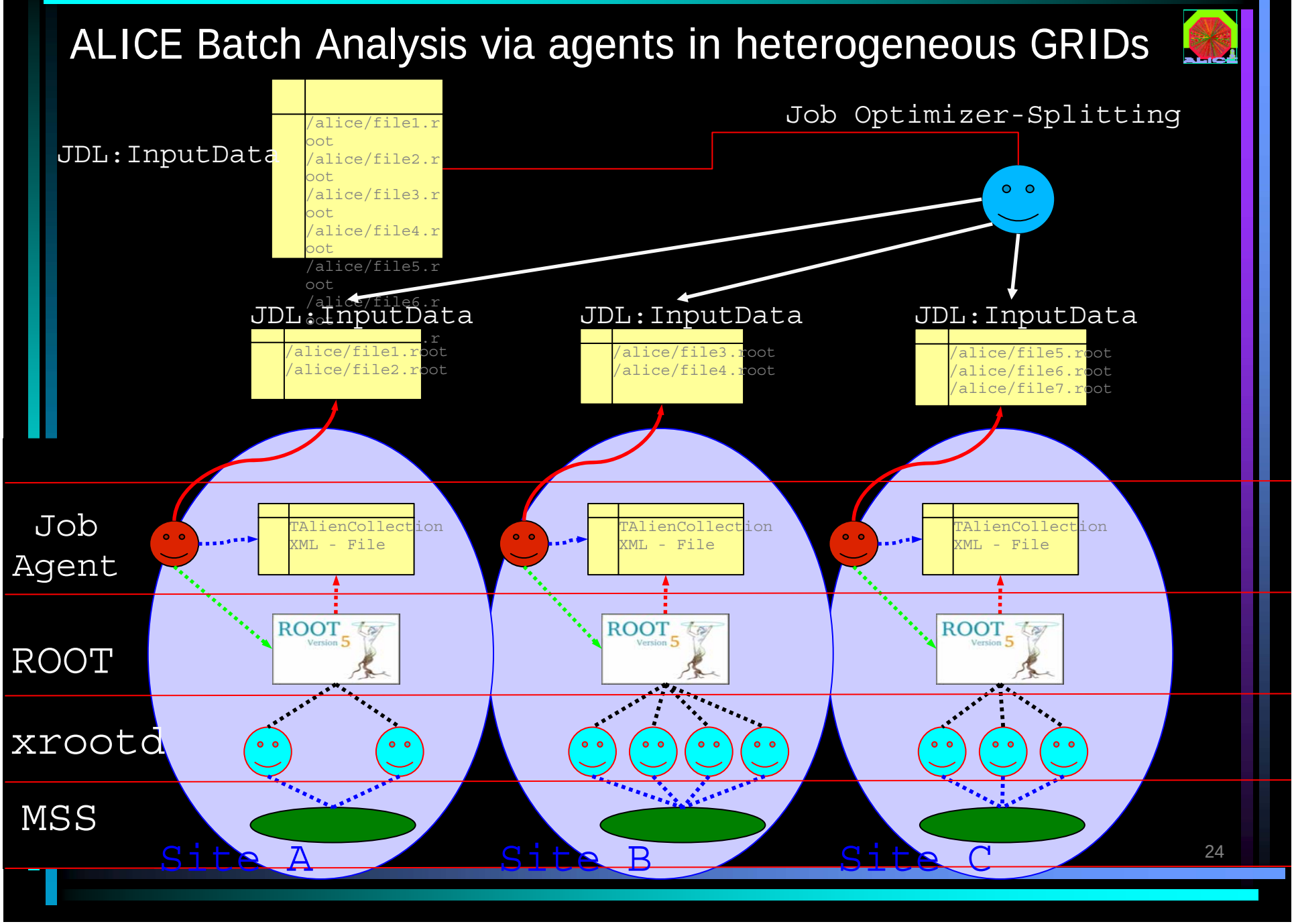
- After optimization, a Job Scheduler periodically assign priorities to jobs in the TQ
  - Scheduling defined by user based on reference and maximum number of parallel jobs
  - Avoids flooding of the TQ and resources by a single user submitting many jobs
  - Dynamic configuration
    - Users can be privileged or blocked in the TQ by a system administrator



Batch Analysis: Agents

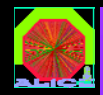
- ALICE CE's do match-making in the AliEn TQ periodically
  - If matching tasks are found, Job Agents are submitted in the local CE queue
- Job Agents do a match-making in the AliEn TQ corresponding to their environment (Job Pull)
  - A running job agent means that the queue is up and running and not a black-hole
  - An unsuccessfully submitted agent does not fetch a job from the Task Queue – no job resubmission need
  - A started job agent knows its running conditions very precisely
  - Same agent can execute more than one job, as long as the time to live of the job does not exceed the time to live of the JA
- The Job Agent Model allows to run with high efficiency even on a less efficient Workload Management/Queue System

ALICE Batch Analysis via agents in heterogeneous GRIDs



Batch Analysis via Agents on heterogeneous GRIDs

- Requirements to run AliEn as a “GRID on an GRID”
  - Provide few (one) User logins per VO
  - Install the Agent Software
  - Startup agents via Queue/Broker systems or run as permanent daemon
  - Access local storage element
    - all data access from the application via xrootd
      - run “xrootd” as front-end daemon to any mass storage system
        - » ideally via the SRM interface, read-write mode
        - » enforce strong authorization through file catalogue tokens
      - run “xrootd” with every JobAgent / WN as an analysis cache
        - » read-only mode
        - » strong authorization only for specific secure MSS paths => “public access SE”



Interactive Analysis

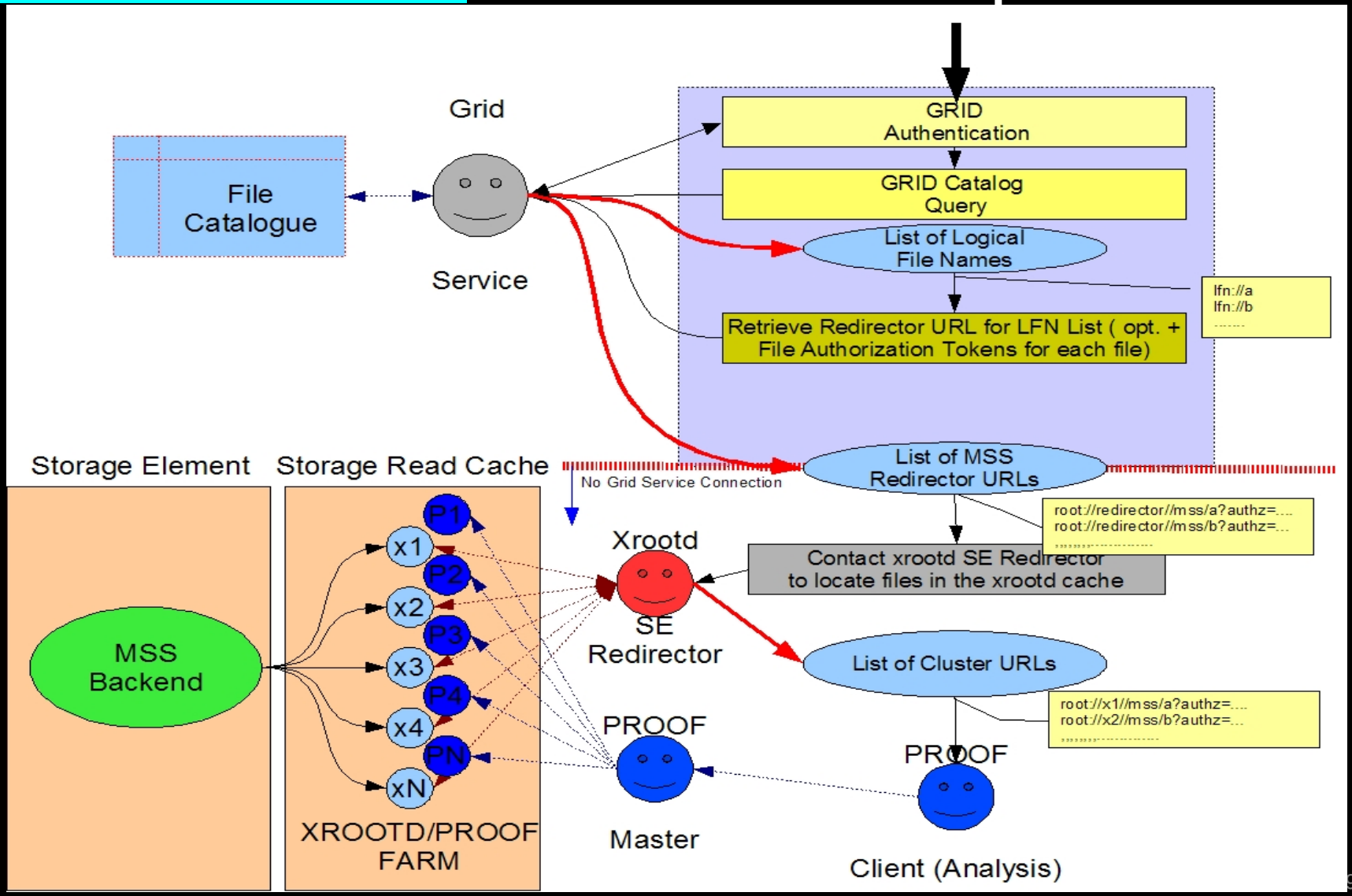
Interactive Analysis Model: PROOF

- Four different use cases to consider
- Local Setups
  - Conventional single-tier PROOF cluster in sites for interactive Analysis (data pre-staged on the cluster disks)
    - site autonomy
    - site policies apply
    - manual work for data deployment, but quite easy to do
  - integrate single-tier PROOF clusters into AliEn
    - a permanent PROOF cluster(proofd+xrootd) is registered as a read-only storage element in AliEn working on a MSS backend
    - PROOF Chains are queried from the AliEn File Catalogue
    - Location of data files in the xrootd cache using the xrootd redirector

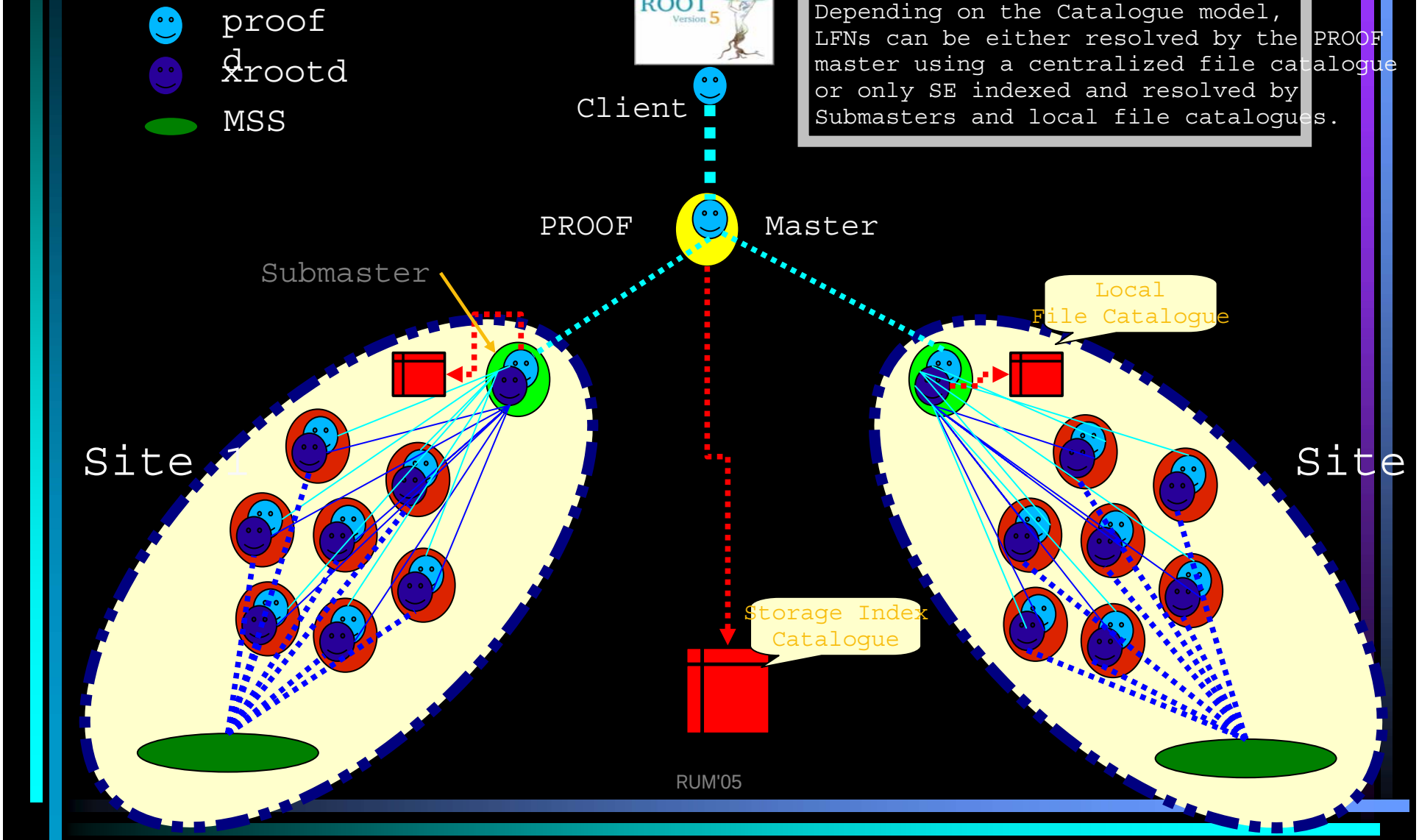
Interactive Analysis Model: PROOF

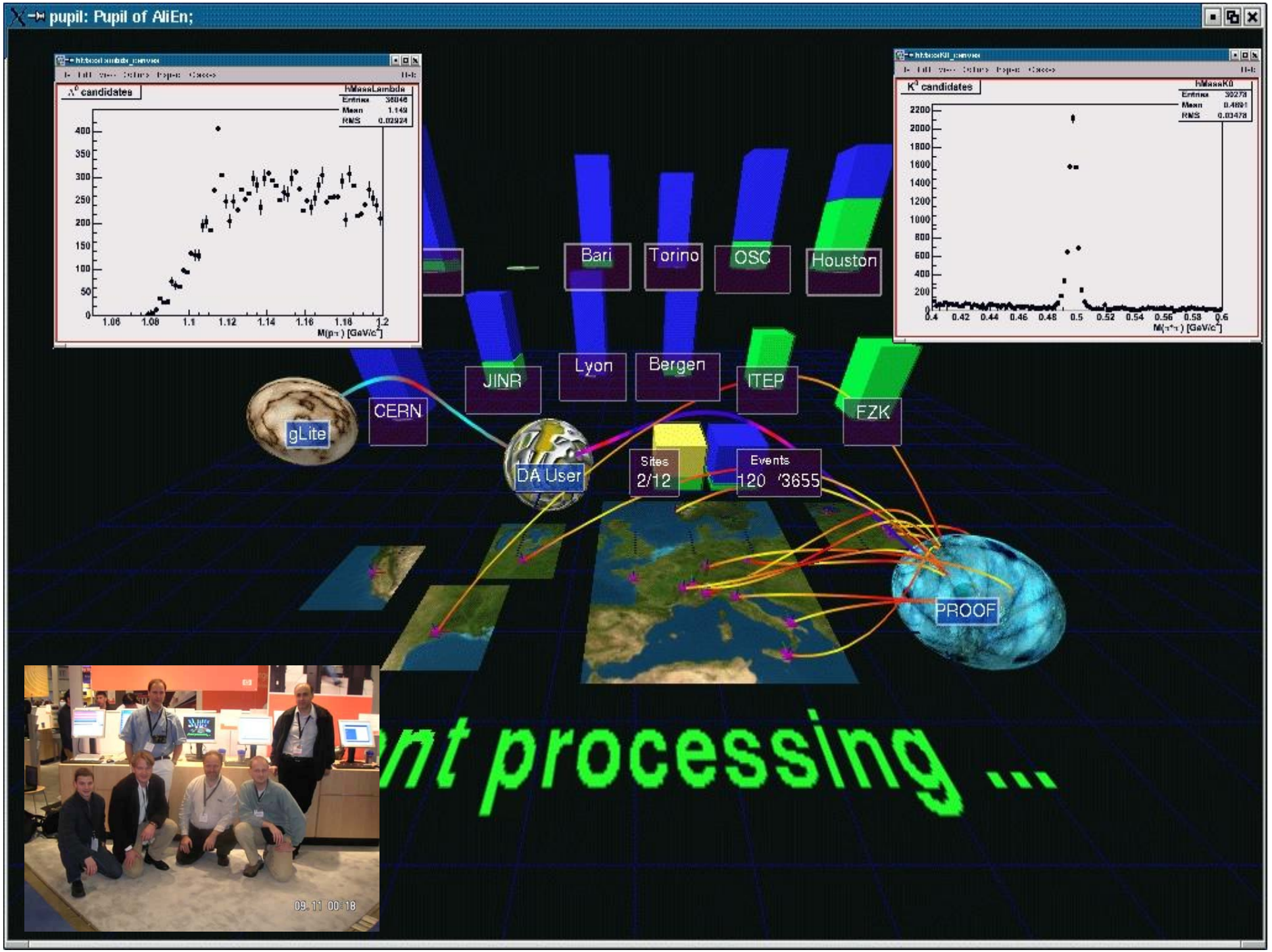
- Multi-tier Static Setup
  - Permanent PROOF clusters are configured in a multi-tier structure
  - A PROOF Analysis Chain is queried directly from the File Catalogue
  - A Chain is looked up by sub-masters using the local xrootd redirectors
  - During an Analysis Query the PROOF master assigns analysis packets to the sub-master -- workers have the *right* (=local) data accessible
- Multi-tier Dynamic Setup
  - all like in the multitier static setup, but
    - proofd/xrootd are started up as jobs for a specific user in several tiers using the AliEn Task Queue
    - ...or... proofd/xrootd are started up as generic agents by a Job Agent – the assignment of PROOFD to a specific user has to be implemented in the PROOF master

PROOF@GRID: 1-Cluster Setup



PROOF@GRID Multitier Hierarchical Setup with xrootd read-cache




pupil: Pupil of AliEn;
Bari
Torino
OSC
Houston
JINR
Lyon
Bergen
ITEP
FZK
CERN
gLite
DA User
Sites
2/12
Events
120 '3655
PROOF
nt processing ...

Conclusions

- Parallelism at different levels offers new opportunity for the analysis of extremely large set of data
- The complexity of the system grows non-linearly and it is subject to the lack of maturity of many of the components
- AliEn and ROOT/PROOF are (rapidly evolving but still) solid foundations to build the system
- However time is very short now
  - Which hopefully will increase our pragmatism!

The AliEn timeline