*ALICE Offline Week , 5.10.2005*

# "Preparation for the User Analysis Phase of PDC06"

**Andreas-Joachim Peters CERN**

Enabling Grids for E-science in Europe

www.eu-egee.org

cern.ch/lcg

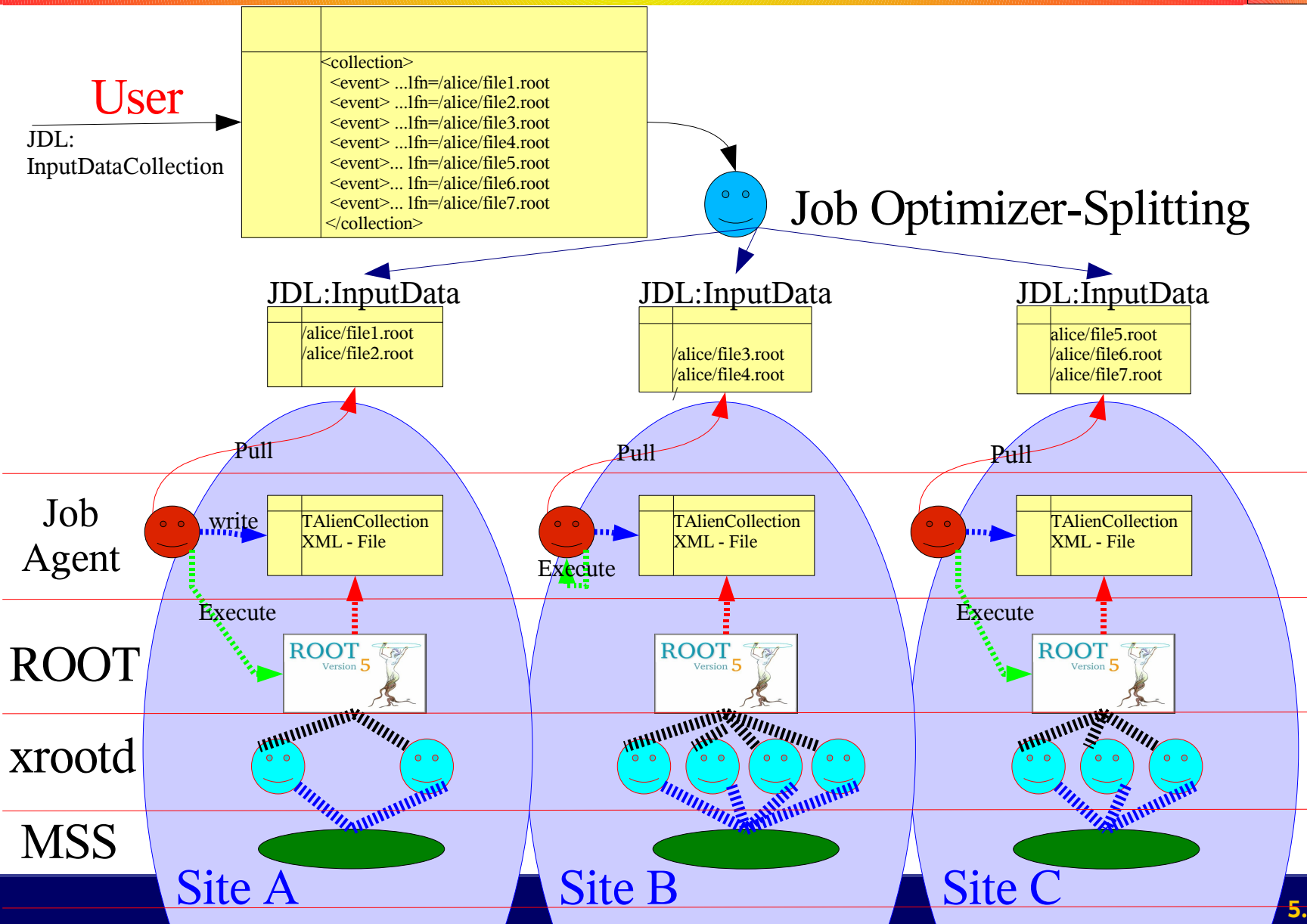
http://cern.ch/arda

EGEE is a project funded by the European Union under contract IST-2003-508833

# Overview

- Overview Analysis Model
- Planning for PDC06 Analysis Phase
- Possible Extensions
- Summary

# Batch Analysis Model
# What we need to setup ....



# Planning for Analysis in PDC06

- Phases (they are not all sequential)
  - I Commissioning of End-User Tools / DM components
  - II Site Configuration/Installation
  - III Site validation
  - IV Analysis in validated Sites
  - V Bug Fixes
  - VI Extensions (DM access, Rules, Quotas etc.)

- We need to fix a timescale for all phases in PDC06

# Phase I - Commissioning

- many improvements/fixes in analysis tools
  - **API service v.2.1.0**
    - moved to gSoap 2.7. and solved connectivity problems
    - now absolutely stable
    - per API machine 20-40 cmds/ s [ file open/s ]
      - » 0-commands 200 cmds/s
      - » f.e. ESD analysis: every 6s one file opening:
        1000 analysis jobs = 166 file open/s = 8 machines
        → scalability problem also in SE service → see Phase IV/VI
    - 2 server now with v.2.1.0 – 1 with old protocol v.2.0.8
      - » will be upgraded with installation of AliEn v2.12
  - **ROOT API**
    - reimplementation of TAlienFile
      - » solved memory leaks and SEGV during long processing
      - » no inheritance problems anymore (inherits from TXNetFile)
      - » supports “intelligent” access from the closest to the farest image(replica) of a file
    - Overlap function in TAlienCollection
      - » needed for TAG analysis with AliEn
    - fix of memory leaks in TAlienResult

# Phase I - Commissioning

- **aliensh**
  - possibility to display admin message, even if the api services are down
  - possibllity to centrally direct users to certain api services
  - inclusion of trigger / mirror / meta data commands
  - minor bug fixes

- **xrootd**
  - new version available *under test right now, maybe solves →
    - erratic problem in high load situation (produces SEGV)
      - » xrootd dies in pthread_mutex_lock
    - erratic problem in high load situation (produces kind of dead-lock)
      - » accepts connections, but does not serve anymore
- **token authorization library**
  - portability fix of the coding format
  - use blowfish 128-bit sym. CIPHERS – faster & JAVA compatible
  - decoding exists now also in JAVA (dCache)
    - » these fixes are in AliEn v2.12 and NOT BACKWARD COMPATIBLE !

# Phase II – Site Installation

- Key points for a site participating in the analysis exercise are
  - proper installed and configured DM components
    - user analysis jobs read always from xrootd frontend
    - need an xrootd frontend to the existing MSS backend
      - » optimal: site providing disk server(s) with xrootd setup
      - » no interface for sites with DPM yet
      - » sites with dCache can try soon dCache 1.7.0 test release - has xrootd door + authorization plugin
      - » sites with Castor2 – native xrootd interface available or staging scripts
  - sufficient disk space for files to analyze

- Proposed Procedure
  - configure 'easy candidates' first
    - native AliEn/xrootd sites (like Muenster)
    - xrootd sites (like Lyon)
    - Castor sites (like CNAF)
    - dCache site (FZK, Russia etc.)

# Phase III – Site Validation

- Generate Analysis Data
  - FTS transfer – if existing
  - xrootd mirroring
  - Generation of new files stored at the site
- Analyse with 'standard' selector all available data and verify functionality
- Include in “Analysis” partition
- Overlap Phase III + IV

# Phase IV - Analysis

- Publish collections containing all existent data for end users
- Open user analysis on validated (tag) collections
- Get experience with the analysis in this environment

already visible is the need for:

## Condensation of data

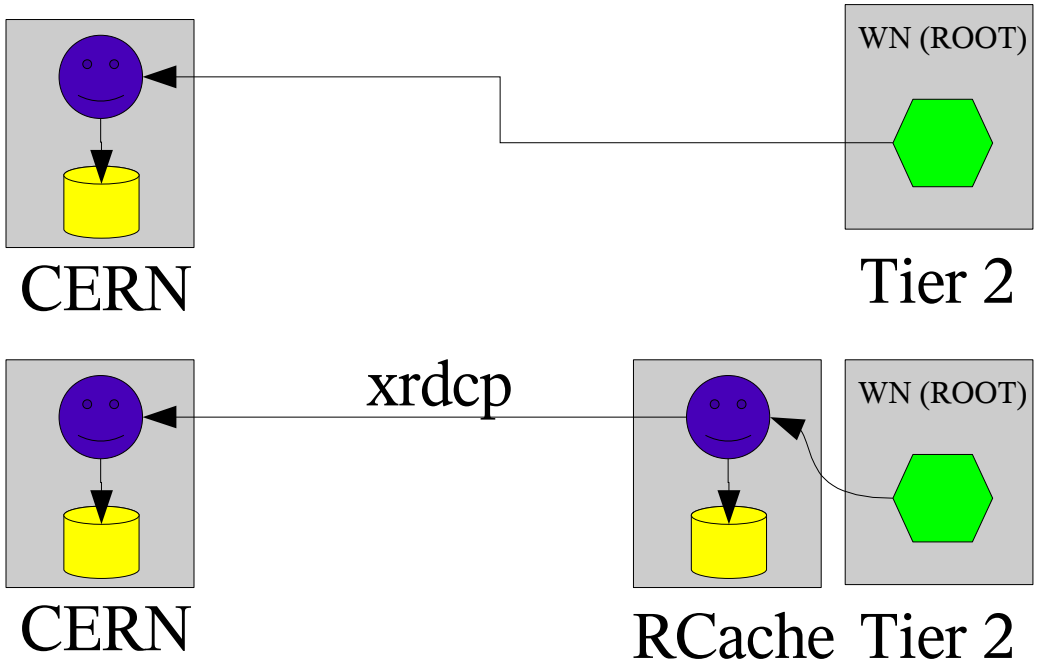
- find a way to reduce the number of files to be analyzed
  - 20 MB ESD files contained in big archives are too small ( too many file opens/catalogue and storage interactions)
  - 1 GB ESD files are read in ~100-200s
    ( 1000 jobs = 5-10 opens/s – relaxes situation, 1 API service is enough )
- create 'trigger' or service for this task → very important !

# Phase V – Bug Fixes

- Fix software according to
  - user feedback
  - experiences during operation
- Fixes can be done easily for
  - API client, since it is used via an AliEn package
  - ROOT, since it is used via an AliEn package
  - central components
- More difficult
  - site services ( needs in principle new release )

# Phase VI - Extensions

- Observation:
  - central file access f.e. to calibration files does not scale for a single disk server to several thousand connecting jobs
  - need to mirror files, but this needs manual intervention
  - proposal:
    - passive xrootd cache

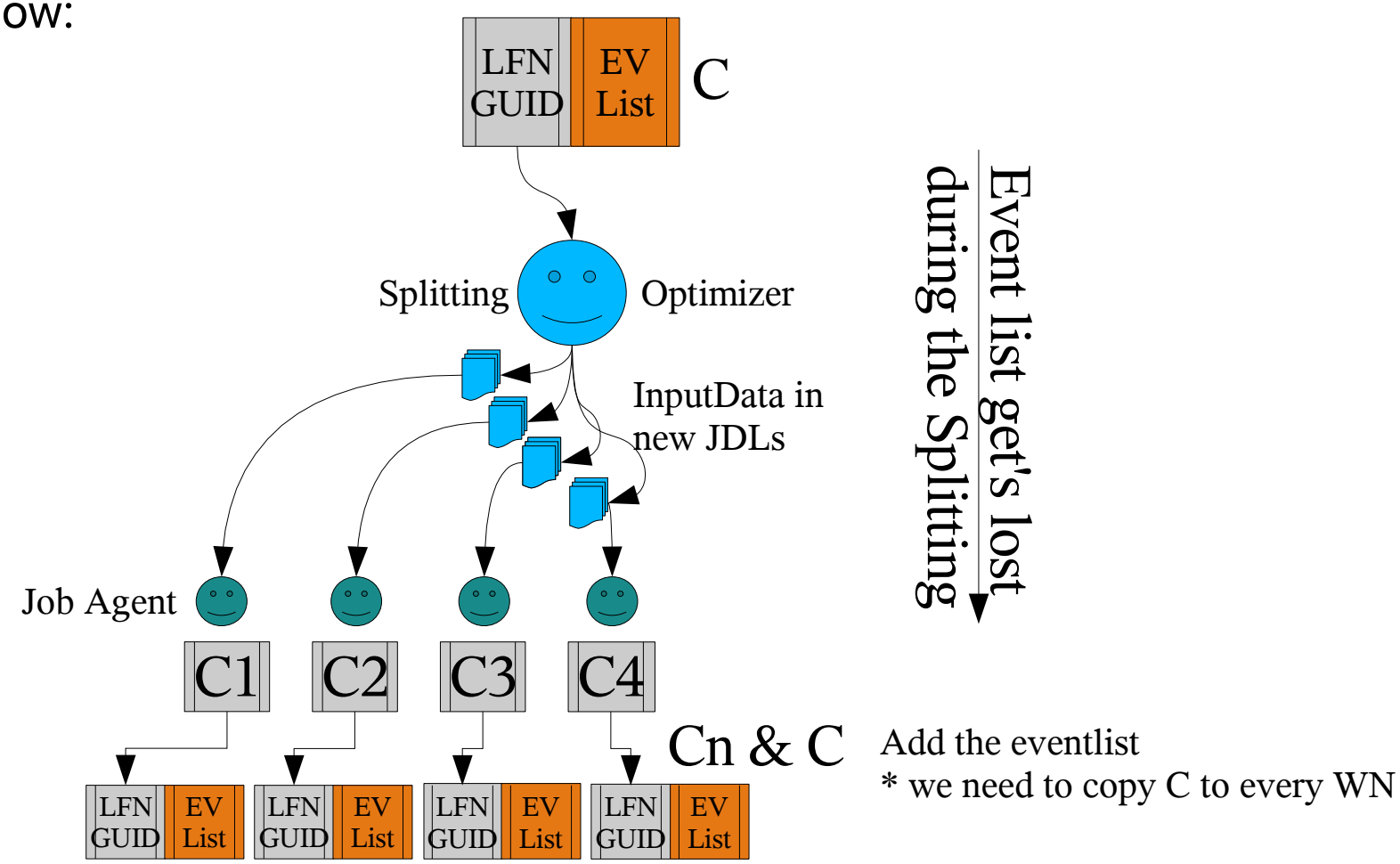


# Phase VI - Extensions

- Read Cache
  - uses simple staging mechanism of xrootd with xrdcp commands
  - uses the client envelope to do the cache copy
  - reduces the number of concurrent clients at the central disk drastically
  - uses simple low/high threshold garbage collection
  - should define for each site the read cache
  - should define meta data in the FC for files, which should be read through caches
  - no development in xrootd (new version supports to pass opaque information to staging scripts)
  - additional entry in LDAP needed
  - modification of the 'access' function in alien, which issues envelopes for file access – don't return the destination disk server URL but the cache disk server URL - trivial

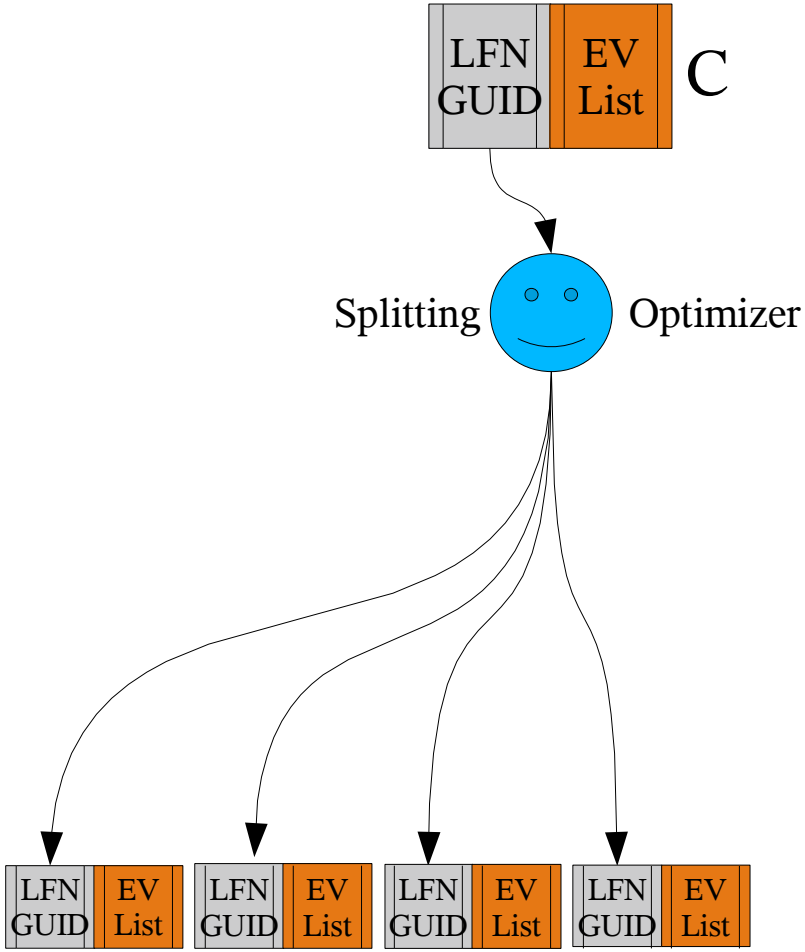
# Phase VI - Extensions

- No full support of XML collections in AliEn
  - now:



# Phase VI - Extensions

- Full support of XML collections in AliEn




Use XML collections directly defining the Job Input Data for each subjob

# Summary

- Phase I (commissioning) is close to the end
- CERN is already used in Phase IV
- Start Phase II-IV with AliEn Release v2.12 in appropriate sites
  - Proposal to start with:
    - Muenster
    - Lyon
    - CNAF
    - one Russian Site to try dCache
    - GSI,FZK ....
- Extensions can be discussed and tested in the next weeks

# Appendix

# AliEn Job defined via JDL

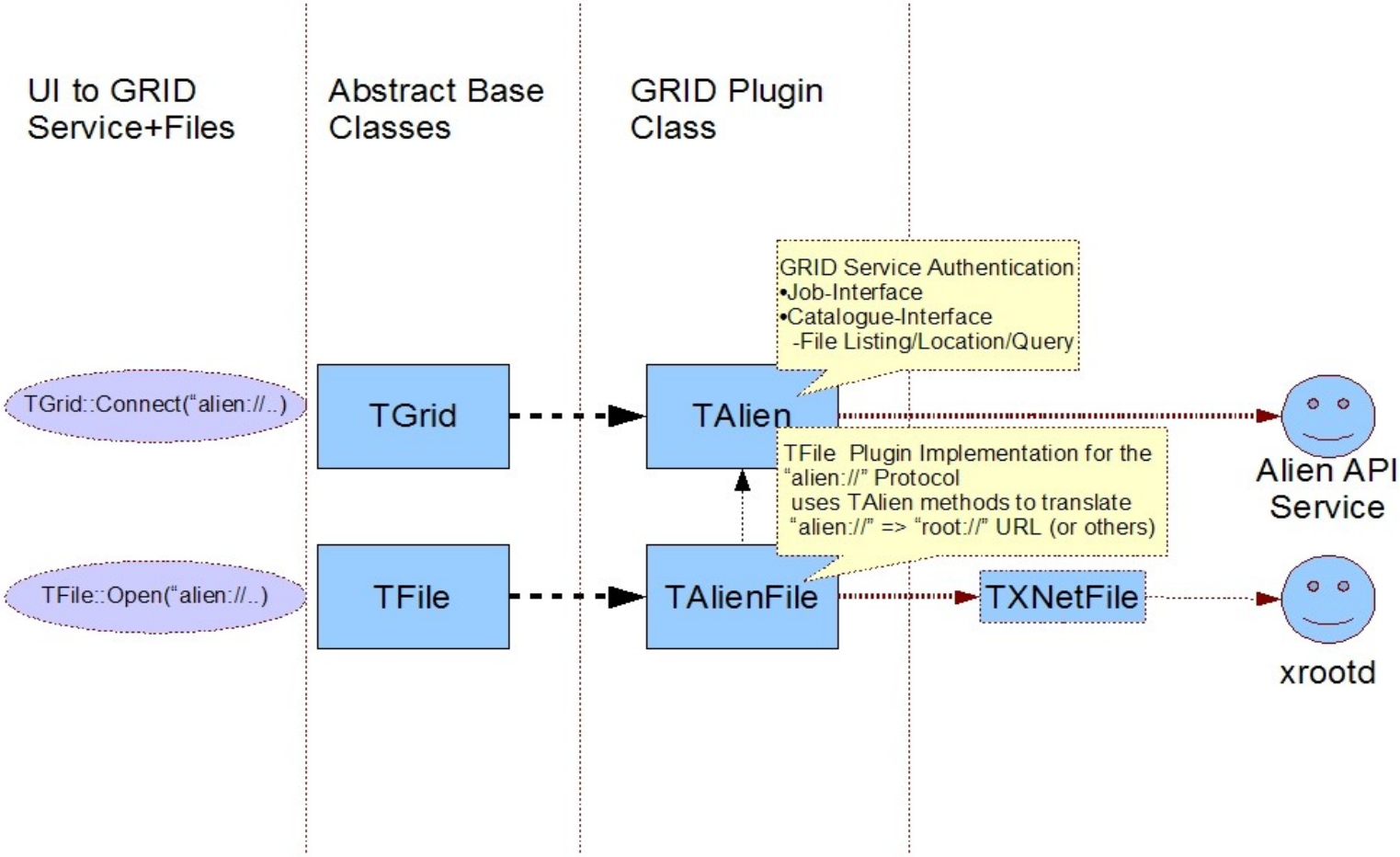
- AliEn uses Classad JDLs:
  - Keywords:
    - Executable
      - what to run ...
    - Jobtag
      - a tag for this job ...
    - Email
      - termination email address
    - InputFile
      - files to be put int the InputSandboxx
    - InputData/InputDataCollection
      - files to be access by the user job (staged or not staged into the users sandbox)
    - Split
      - rule, how to break up a job
    - OutputFiles/OutputArchives aso ....

# AliEn Batch Analysis - Input Files - Input Data

- Analysis Job Input Files:
  - downloaded into the local job sandbox
    - analysis macros, configuration files etc.
- Analysis Job Input Data/Input Data Collection:
  - 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”);

# AliEn Analysis - File Access from ROOT "all files accessible via LFNs"



# AliEn Batch Analysis - Job Splitting

- all jobs (JDLs) submitted to the AliEn task queue pass an optimization and scheduling process
  - Job Optimizer:
    - JDLs can be broken up into several jobs (job splitting)
    - the Input Data list can be split according to several modes:
      - per file: every single Input Data file is submitted as Input Data in seperate jobs
      - per se: Input Data files with an identical storage index (list of SE where the files are available) are submitted as Input Data in a seperate jobs
        - » additionally a maximum number of Input Data files per job can be specified
        - » additionally a maximum size of Input Data per job can be specified#
      - per directory: every Input Data directory is submitted as Input Data in seperate job
  - 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