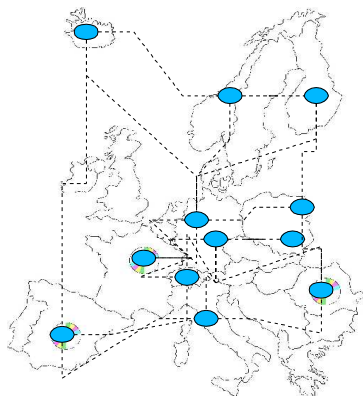
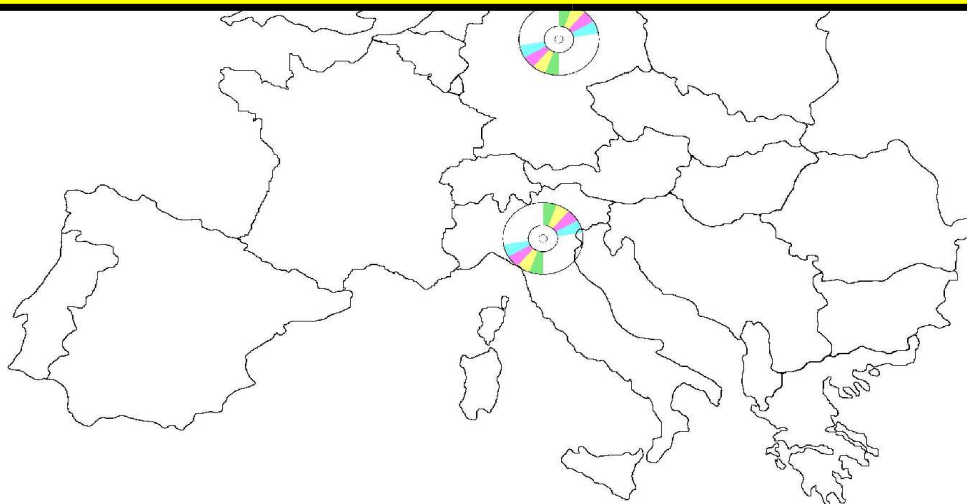
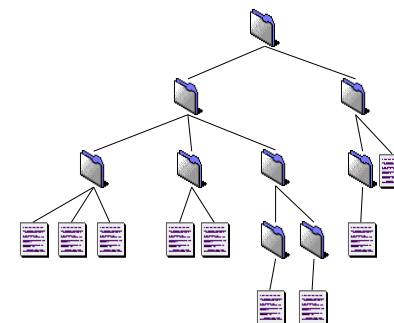


- The **ALICE** Analysis Approach -
&
Expectations of **ALICE** from
ARDA



A.J.-Peters for the ALICE
Collaboration



The **ALICE** Analysis Approach

- The **ALICE** software framework is built on top of one base:
 - **ROOT** - most popular HEP analysis toolkit
 - => **AliRoot** framework

- **ALICE** uses **AliEn** as an GRID analysis platform for distributed computing within the **ROOT** framework:
 - => **AliEn** GRID functionality is integrated into the **ROOT** framework!

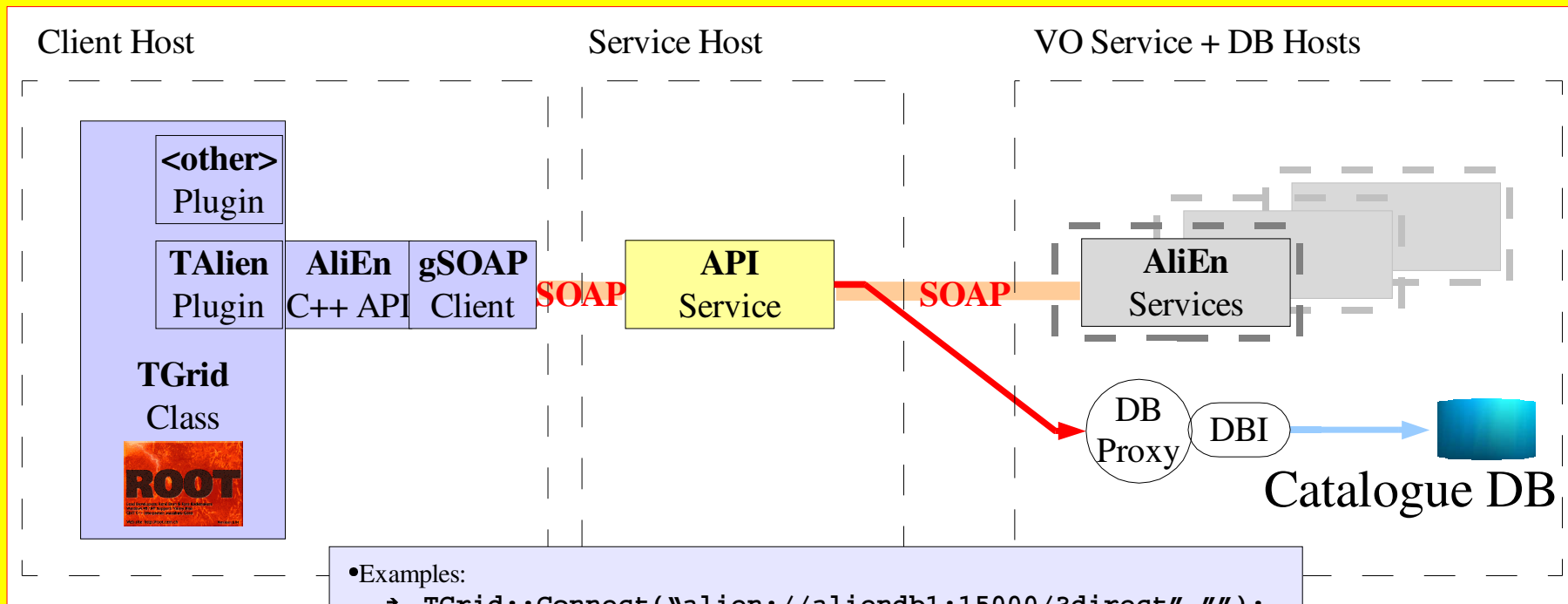
=> One framework for the end-user: GRID *enabled* **ROOT**

Key features of **ROOT** & **AliEn**

- **ROOT** physics analysis toolkit:
 - Interactive analysis work using familiar C++ style syntax
 - data visualisation, an object-oriented I/O system
 - is successfully used within the AliRoot framework of the ALICE experiment as an all in one solution
- **PROOF** extends workstation based concept of ROOT to the 'parallel ROOT facility'.
 - user procedures are kept identical during an analysis session
 - tasks are distributed automatically and executed in parallel
- **AliEn** as a GRID analysis platform:
 - a global namespace for GRID files
 - files are indexed and tagged in a virtual file catalogue and everywhere globally accessible
 - a global queue system for batch and interactive jobs
 - global job scheduling according to resource requirements

Jointventure of **ROOT** & **AliEn**

- ROOT uses the **SOAP based** AliEn C++ API to communicate with the AliEn API service



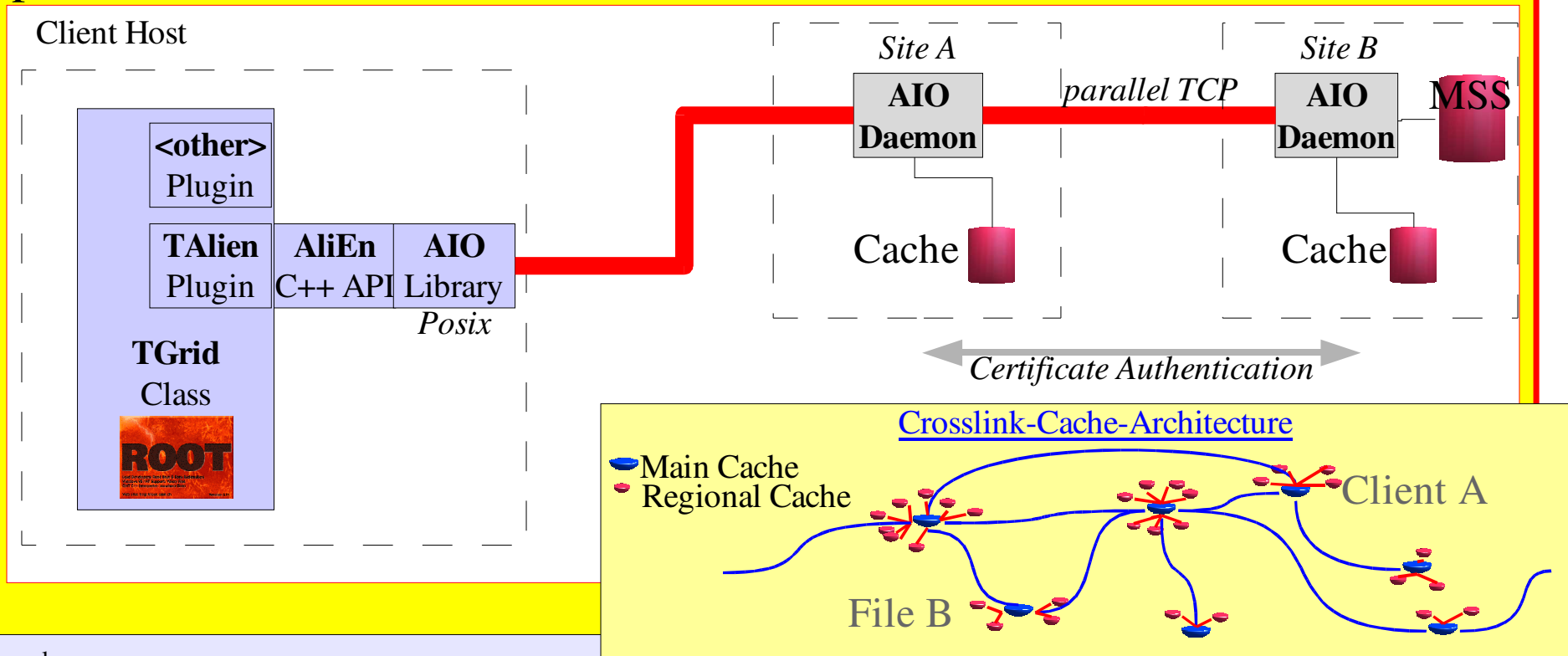
```

•Examples:
  → TGrid::Connect("alien://aliendb1:15000/?direct", "");
    // initiate gGrid with AliEn plugin (API server at aliendb1, port 15000)
  • gGrid->mkdir("/alice/acat03");
    // create directory in virtual file catalogue
  
```

- AliEn Services + Catalogue are accessible via the **TAlien(TGrid)** class

File Access with **ROOT** & **AliEn**

- For direct file access ROOT uses implementation of a **POSIX I/O interface** provided in the AliEn C++ API



• Examples:

```

→ TGrid::Connect("alien://aliendb1:15000/?direct","");
// initiate gGrid with AliEn plugin (API server at aliendb1, port 15000)
• gGrid->GridOpen("/alice/production/histogram.root");
// open a ROOT file using <aio> access
    
```

ALICE Analysis model with **AliRoot** & **AliEn**

- **Analysis Data Structures stored as events in trees:**
 - Analysis Event Objects (DS) are wrapped up by **ROOT** trees
- **Set of data** are produced with catalogue queries using:
 - pathname to initiate search
 - file/suffix pattern match
 - meta data conditions
- **Workload distribution using**
 - *batch* analysis model
 - *interactive* analysis model

Batch Analysis model with **AliRoot & AliEn**

needs:

=> **Job Splitting** “relation data/ressources”

=> **Result Merging** = “merge local” + “merge global”

=> **Job Dependencies**

1st Job Splitting 2nd “merge local” 3rd “merge global”

Job Split Procedures:

1 Job Spawning:

- a production job is spawned <n> times by the Job Optimizer

2 File based Splitting:

- a task is split into one job for each input data file

3 Storage Location Splitting:

- a task is split into one job per storage system

4a/b File size/number based Splitting:

- a task is split into one job for a group of input data files with a maximum input data size or maximum number of files

5 Combinations of 3+4a & 3+4b & 3+4a+4b

Job Description Language:

=> **splitting/merging**

```
InputData={"/alice/*galice.root"};
Split="SE";
MaxInputFileSize="100000000";
MaxInputFileNumber="10";
```

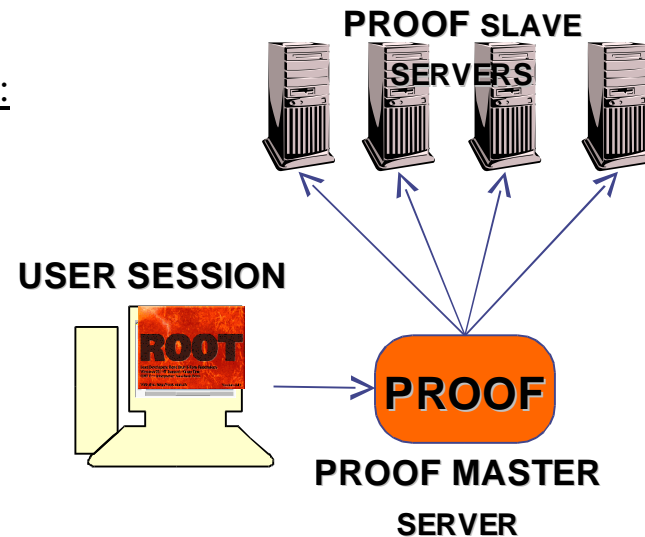
=> **job dependencies**

```
JobPredecessor = <jobId>
```

Interactive Analysis model with **PROOF**

→ PROOF allows interactive analysis on local clusters with a **static** master/slave **configuration**

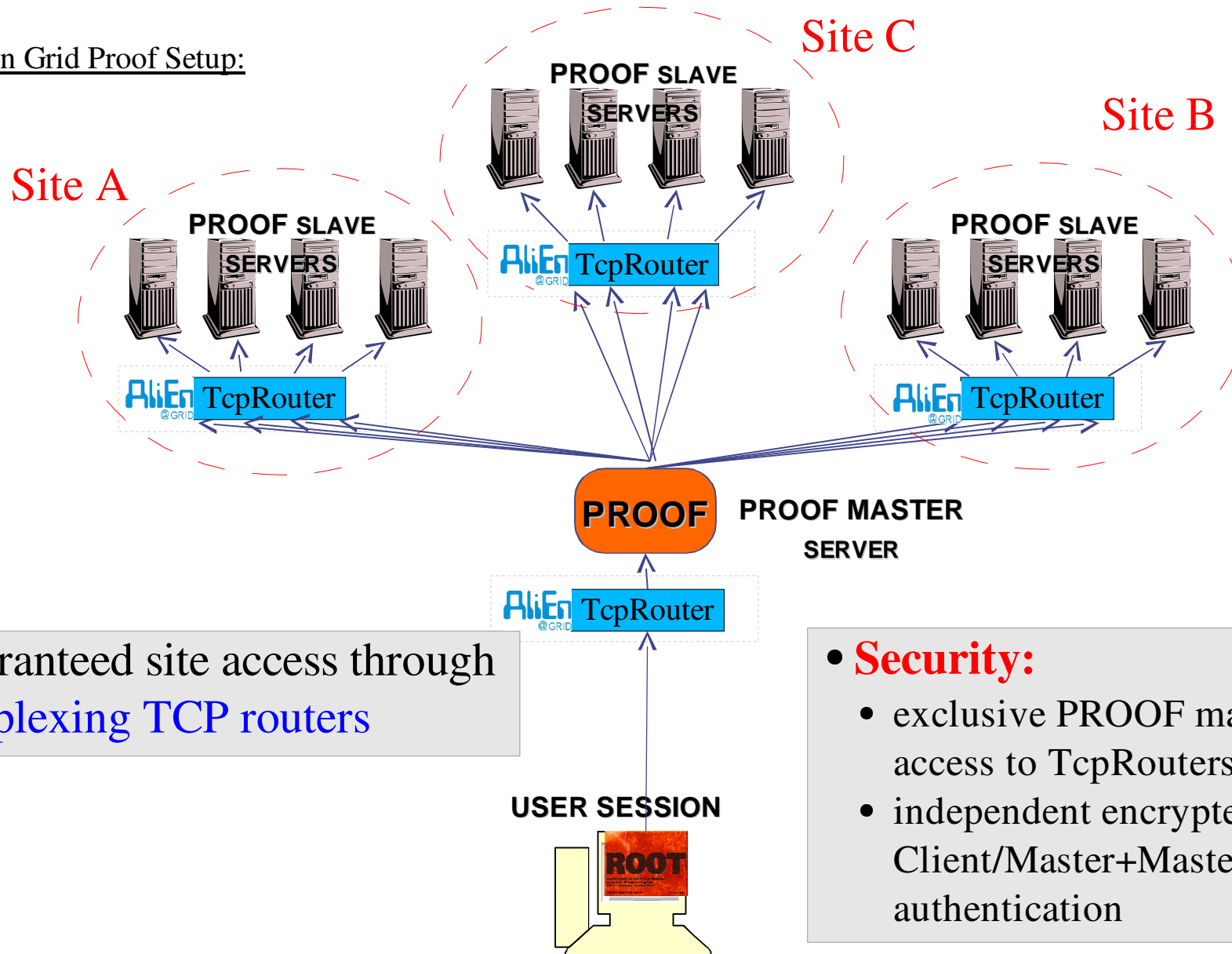
Classical Proof Setup:



- in the GRID Environment: **resources** are **distributed** and **dynamically allocated**
- no guaranteed incoming network access to every possible slave

Interactive Analysis model with **PROOF** & **AliEn**

AliEn Grid Proof Setup:



- Guaranteed site access through **multiplexing TCP routers**

- **Security:**
 - exclusive PROOF master access to TcpRouters
 - independent encrypted Client/Master+Master/Slave authentication

Interactive Analysis model with **PROOF** & **AliEn****Work Distribution**

- interactive analysis with **PROOF** steered by a data packetizer

- in a local cluster:
 - cluster wide accessible data can be processed by all slaves
 - packet takeover by **all slaves!**

- in a GRID environment:
 - site wide accessible data can be processed by all slaves
 - packet takeover by **all slaves close to a MSS !**

Reservation and Booking System

- interactive daemons are (pre-) started on demand by AliEn
- interactive resources are assigned corresponding to the requested set of data and availability

Demonstration!

Unification of Batch & Interactive Analysis with AliEn + ROOT/PROOF

current implementation:

- datasets are represented by objects of the type **TDataSet** in ROOT
- a **GRID data query** assigns data files to TDataSet Objects
- the “**process**” method **initiates** the interactive processing on the assigned GRID proof cluster

to come:

- the same “**process**” method **initiates** the batch processing of the same data set and the automatic merging of results.

ALICE will evolve and test analysis facilities soon during the physics data starting 2/2004

Summary

The traps and difficulties in running analysis on distributed systems have been presently explored within AliEn.

A lot of expertise has already been collected within a short time with minimal manpower,

- **PROOF** in the AliEn GRID environment enables distributed interactive analysis
- AliEn has a modular architecture which allows extensions
=> can also satisfy the ALICE needs in the future!

but also problems and ideas for improvements:

- interactive systems need to deal with all kind of resource failures and unexpected interruptions
- “How to effectively deploy and manage GRID services”
- “How to do the tradeoff between wasting of resources using prestarted interactive daemons and starting them on demand”
- dataset queries could be processed like queue jobs and stored as objects for re-use =><Grid Dataset Objects>

Expectations of **ALICE** from **ARDA**

- A prototype along the lines of the report, i.e. re-factored ~~OCSI~~ compliant AliEn-like GRID that could provide *at least* the current functionality including *interactive* analysis support
 - If possible the prototype should be *backward compatible* with the AliEn file catalogue
 - the prototype should be done *as soon as possible* – a lot of time has been already lost in discussion
 - the prototype should be the *basis for the EGEE/LCG* future developments
 - ALICE experience in web service oriented GRID developments and usage should be taken into account!
- ALICE is willing to support and contribute to development of a project aiming at these objectives.