

The **T3 Analysis Facility** project aims to exploit Tier 3 (non-grid) computing and storage facilities at Tier 2 sites to run local analysis (see Figure 1). The T2 grid services and facilities are used to download or replicate the data needed from the grid by local physics groups. Once available on the site, the data is then used by the institute's scientific community to perform a local analysis.

Therefore, the basic required services are the compute element, the storage element, the user interface and worker nodes. The compute element is responsible to control the batch system (PBS, LFS, Condor e etc), then computing resources locally accessible (WNs) are used via site batch system queues already in place by the site's middleware installation.

Ganga allows for the user to transparently change on where to submit the job, either the local cluster or the Grid, without the need to change the job description. It is necessary to install Ganga tool at User Interface to be able to submit jobs to batch system and to analyze data.

Site Tier 2 with Tier 3 elements

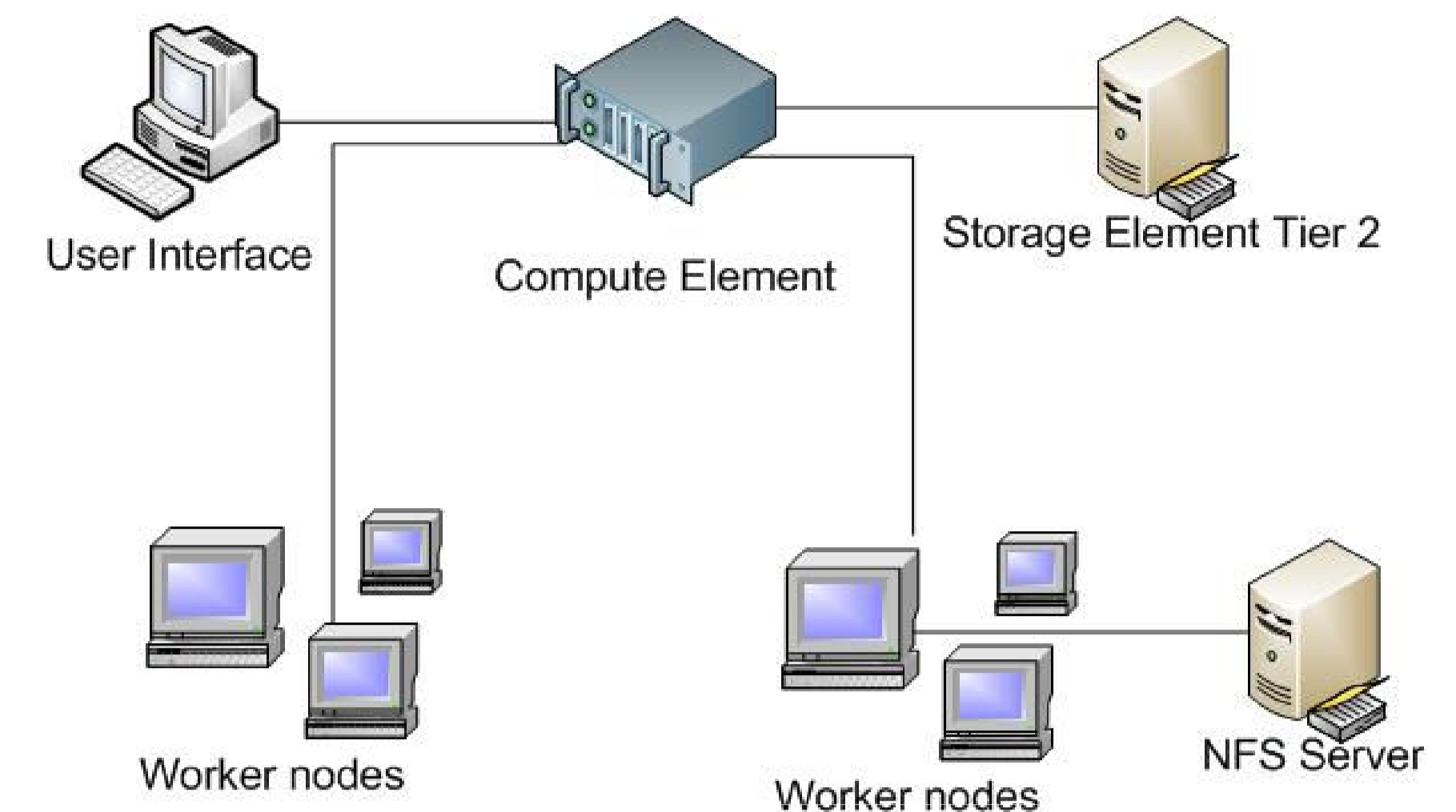


Figure 1: Overview of the T3 Analysis Facility

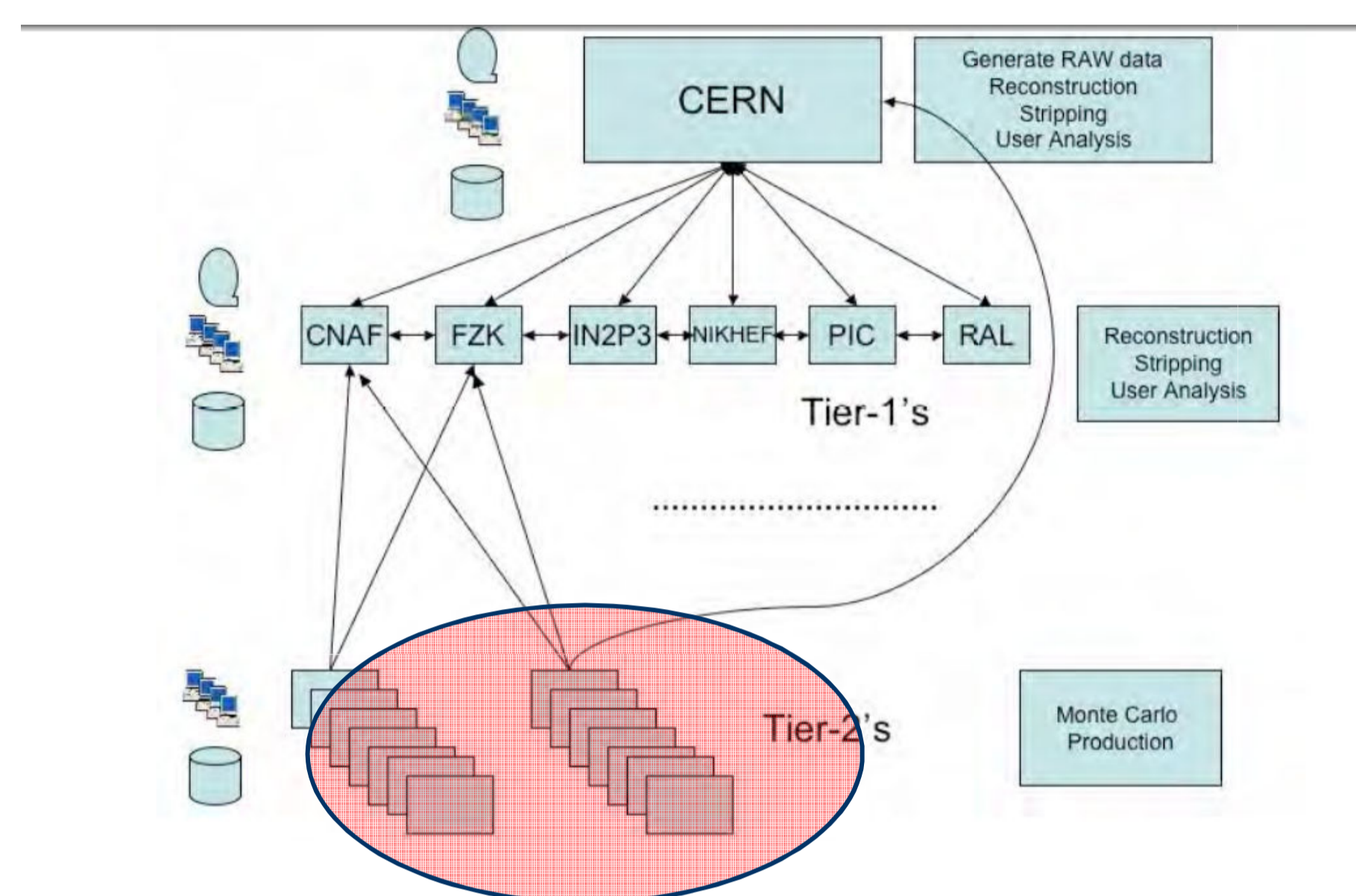


Figure 2: The T3 Analysis Facility targets existing T2 sites to be used for local analysis. In the example the LHCb computing model is used as an example

The LHCb Computing Model divides collaboration affiliated computing centers in three main categories:

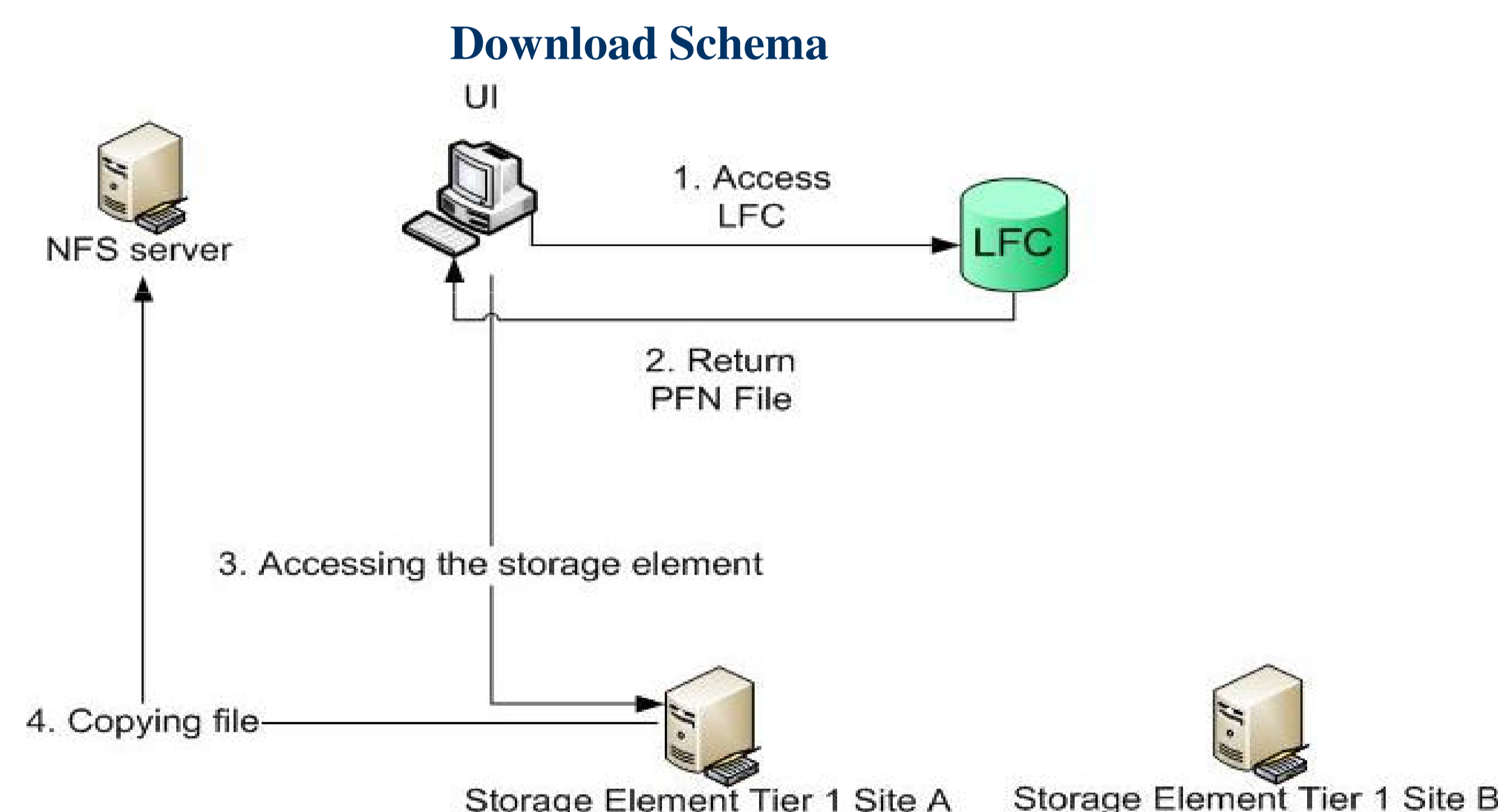
- **Tier0** center : part of CERN infrastructure, targeted for real data acquisition
- **Tier1** (6+CERN) centers: used for data reconstruction/processing /reprocessing and collaboration analysis
- **Tier2** centers: MC generation and - in some particular cases - distributed analysis.

Computing model of LHCb can be found at:

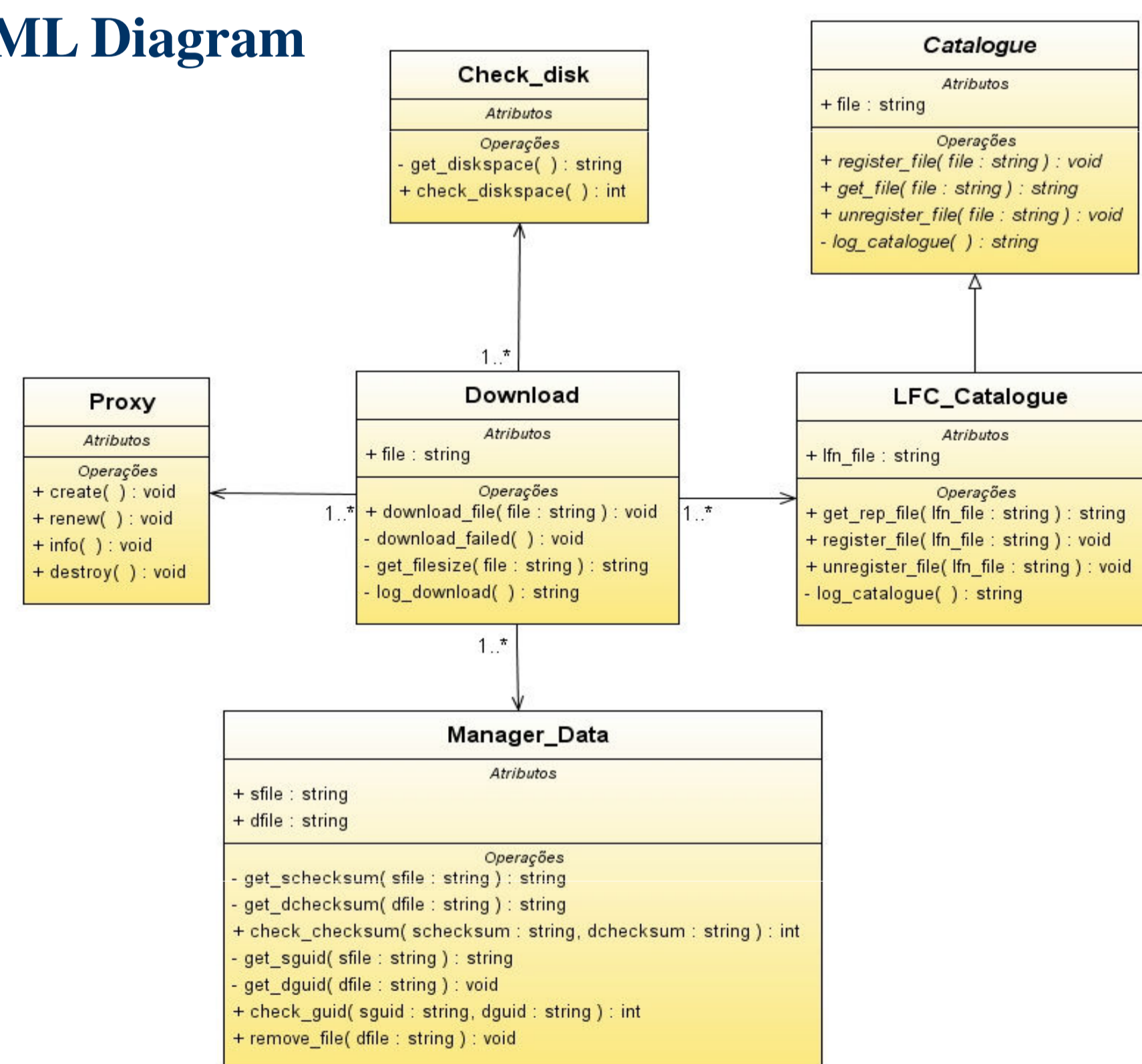
<https://twiki.cern.ch/twiki/bin/view/LHCb/ComputingModel>

T3 analysis Facility

Workflow: A physicist wants to access a dataset of files for analysis. This dataset normally consists of a list of files specified as LFNs (Logical Filenames) The LFC (LCG File Catalog) is then used to retrieve the list of physical replicas of these files and their location. Once the LFC gives the replicas available on the Grid, the file is either copied down to the to storage element or replicated*. Once the download finishes successfully the files are then cached into the local NFS area. Since now the data is available on the local site, jobs can be scheduled to use the data either locally or through the grid. This is best achieved using the Ganga job management and submission tool (ref:Ganga). The use of local NFS area as cache is just an option to by pass issues related with grid proxies management in non grid resources.



UML Diagram



-- Edited By PoolXMLCatalog.py -->

<<POOLFILECATALOG>>

<<File ID="6E9D803C-A596-DD11-B3E6-00304879F95C">>

<<physical>>

<<pfm filetype="ROOT_ALL" name="castor://castorlhc.cern.ch:9002/castor/cern.ch/grid/lhcb/MC/MC09/DST/00005870/00000005870_00000058_1.dst?svcClass=lhcbdata&castorVersion=2"/>>

<<physical>>

<<logical>>

<<File>>

<<POOLFILECATALOG>>

XML Slice produced by T3 tool facility

<<POOLFILECATALOG>>

<<File ID="6E9D803C-A596-DD11-B3E6-00304879F95C">>

<<physical>>

<<pfm filetype="ROOT_ALL" name="/grid/lhcb/MC/2008/DST/00010040/0000/00010040_00000005_5.dst"/>>

<<physical>>

<<logical>>

<<File ID="lhcb/MC/2008/DST/00010040/0000/00010040_00000005_5.dst"/>>

<<File>>

<<POOLFILECATALOG>>

The project was designed in five basic modules to express basic services involved to copy a grid file to a non-grid storage. These modules are named as *download*, *data_manager*, *check_disk*, *proxy* and *catalogue*.

The *proxy* module is responsible for managing proxies such as creating and destroying a proxy.

The *catalogue* acts as an interface between the tool and the generic File Catalog. The main feature of this service is the management of the data replica.

The *data_manager* service is responsible for retrieving file information from the Storage element and to ensure the consistency between the source and the destination file.

The *check_disk* service checks for available disk space in order to store the files.

The *download* service integrated all of these services previously presented. Therefore, this tool offers a friendly user interface to copy a files from the PFN (Physical File Name) to the Storage Element.

Normally, Ganga produces a XML slice to indicate the file's GUID, PFN and LFN. Therefore, through the PFN one can analyze grid files using their protocol to access data.

The proposed project permits to use PFNs of the non-grid files using directly the path of their POSIX directories.

This XML slice is generated at submission time, thus it was necessary to change it. Routines like "RTHUtils.py" - responsible to generate the XML file - has been overloaded for that reason..

* Replicated means that the file is also registered in the file catalog and then potentially made available to the whole community.