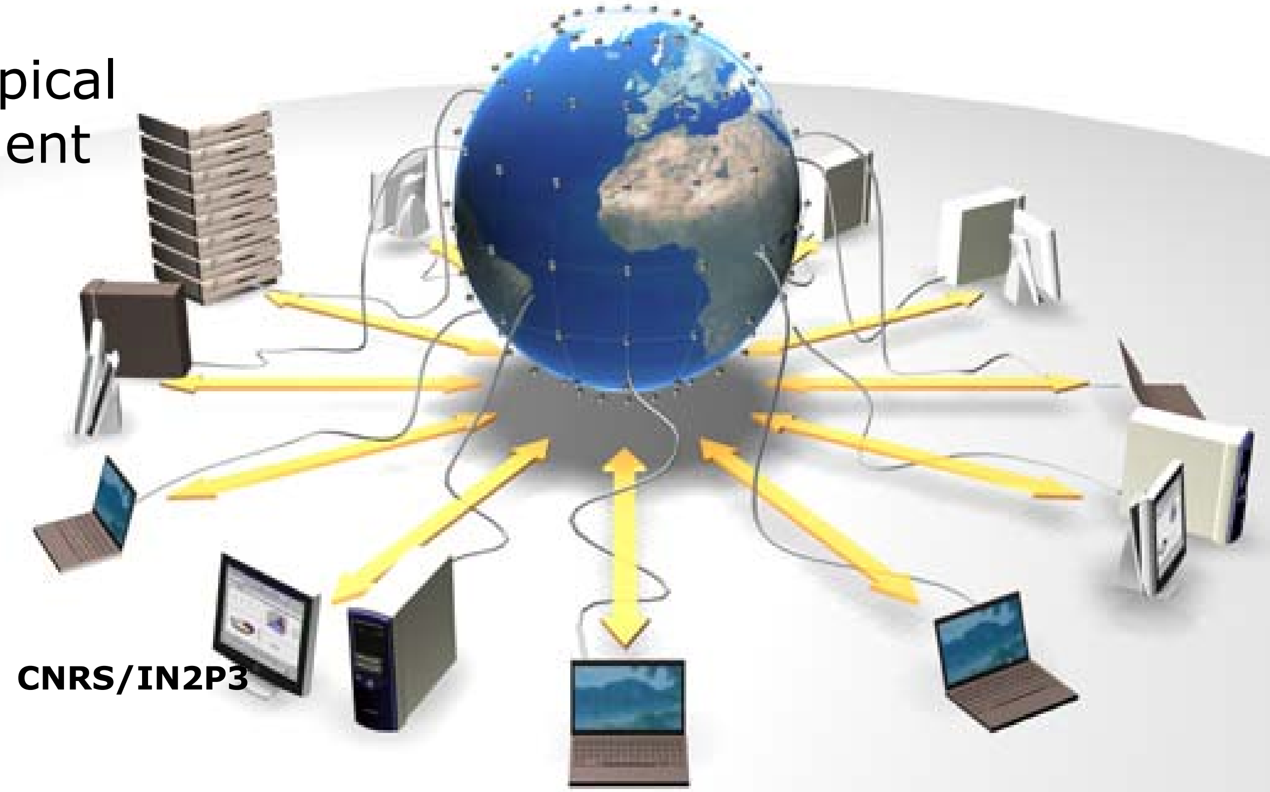
Parallelization of Monte Carlo simulations GATE for medical applications

The scenario of a typical radiotherapy treatment

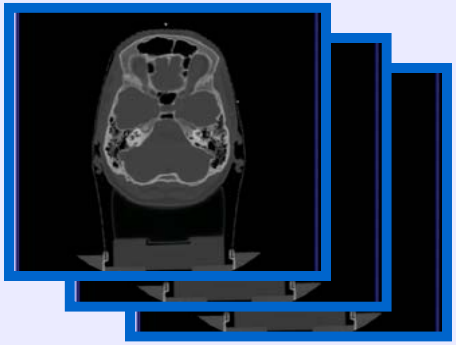


WP10

Lydia Maigne, Yannick Legré **CNRS/IN2P3**
maigne@clermont.in2p3.fr
legre@clermont.in2p3.fr

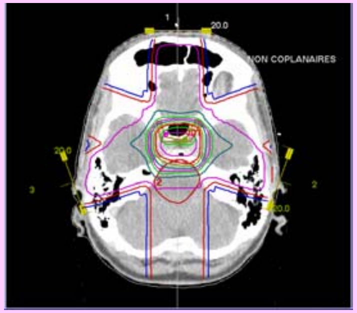
Radiotherapy is widely used to treat cancer

1°) Obtain scanner slices images



The head is imaged using a MRI and/or CT scanner

2°) Treatment planning



Calculation of deposit dose on the tumor (~1mn): A treatment plan is developed using the images

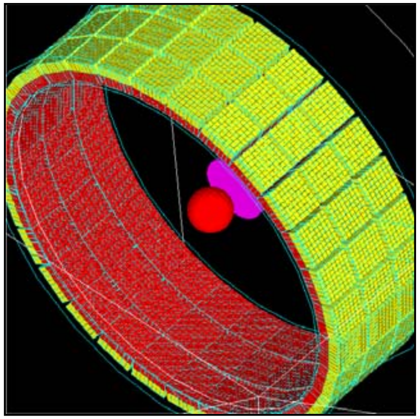
3°) Radiotherapy treatment

Irradiation of the brain tumor with a linear accelerator

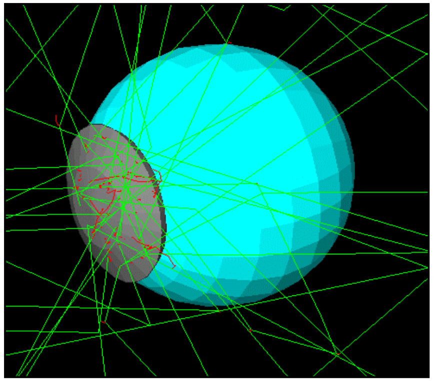
Better treatment requires better planning

- Today: analytic calculation to compute dose distributions in the tumor
  - For new Intensity Modulated Radiotherapy treatments, analytic calculations off by 10 to 20% near heterogeneities
- Alternative: Monte Carlo (MC) simulations in medical applications
- The GRID impact: reduce MC computing time to a few minutes

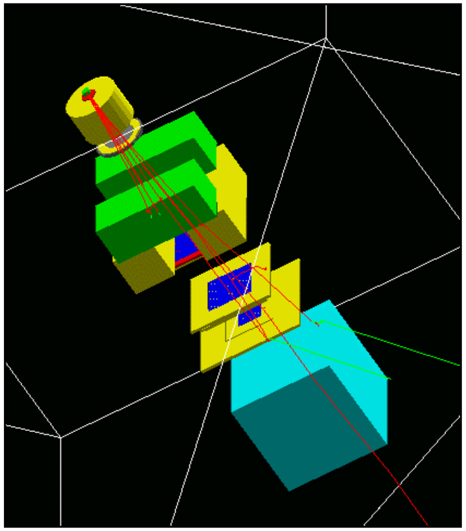
WP10 Demo: gridification of GATE MC simulation platform on the DataGrid testbed



PET camera



Ocular brachytherapy treatment



Radiotherapy

Creatis

LIMOS

Data GRID

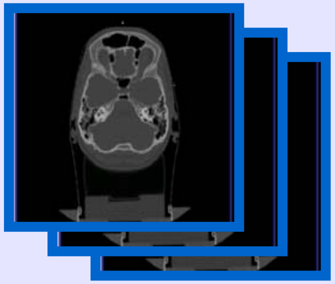
Computation of a radiotherapy treatment on the Datagrid: Let’s go….

Centre Jean Perrin

Centre Régional
de Lutte Contre le Cancer
Clermont-Ferrand - Auvergne

1°) Obtain the medical images of the tumor:

- 38 scanner slices of the brain of a patient are obtained

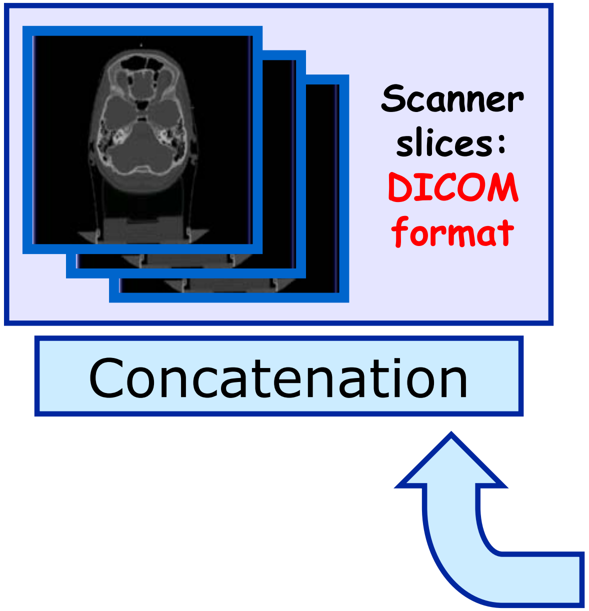


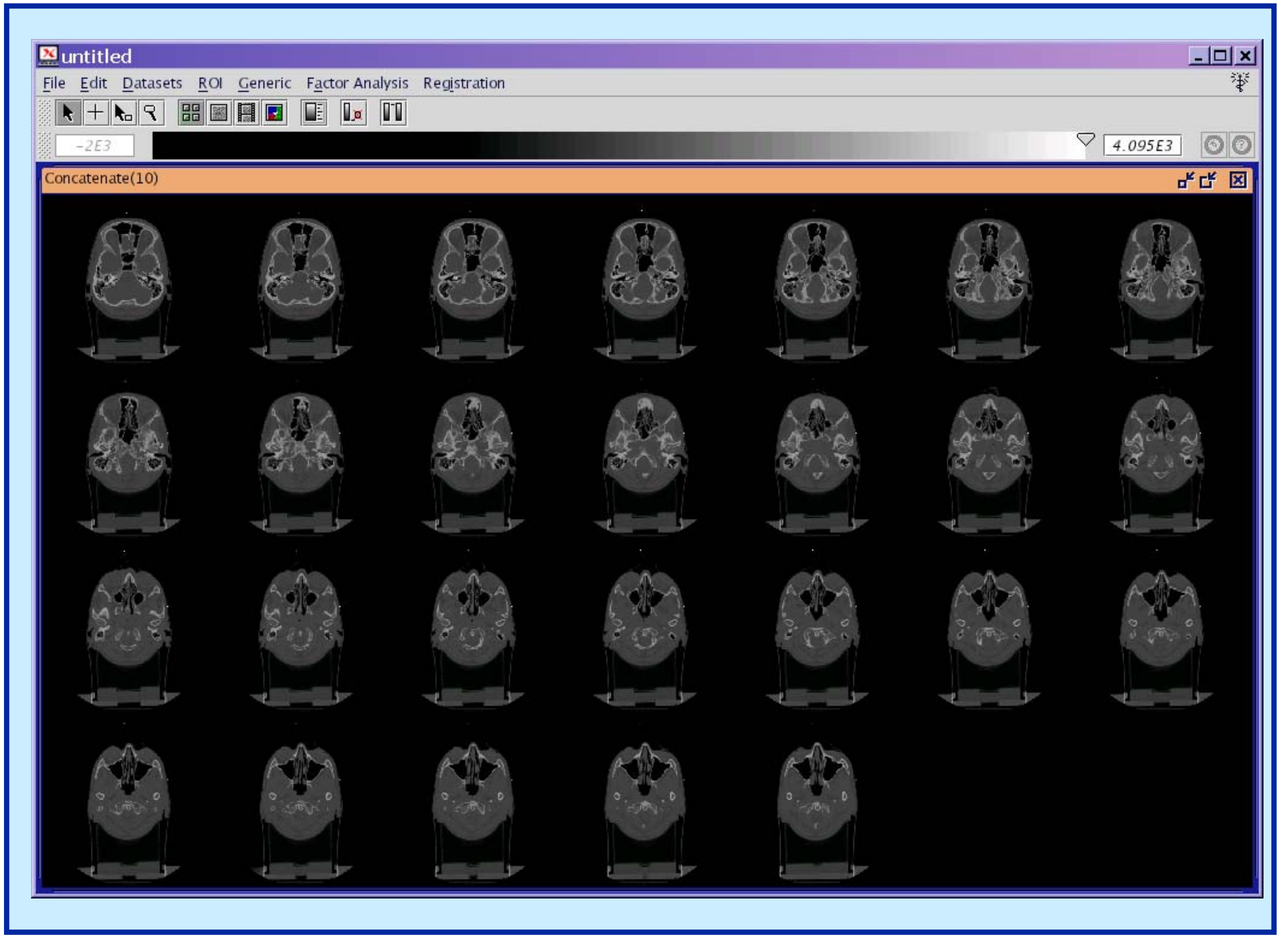
Scanner slices: DICOM format

- **Format of the slices:**
  512 X 512 X 1 pixels
- **Size of a voxel in the image:**
  0,625 X 0,625 X 1,25 mm

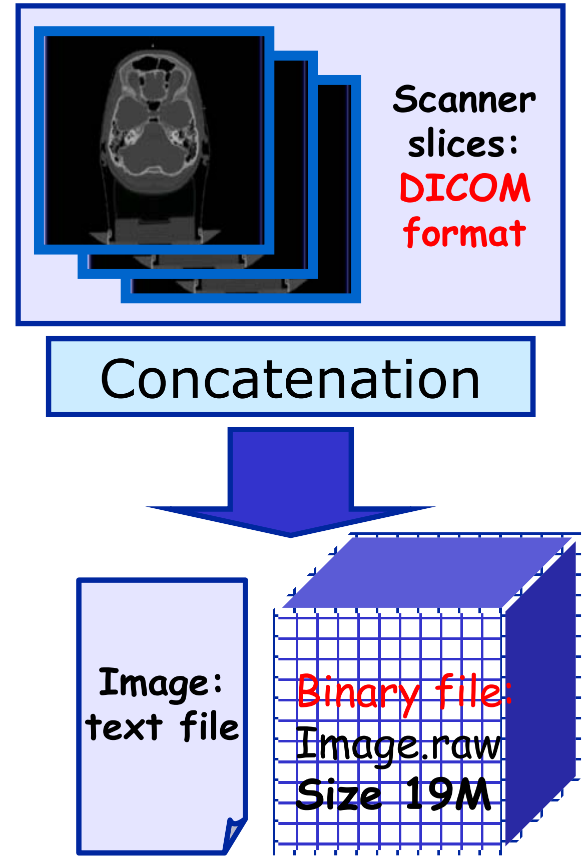
2°) Concatenate these slices in order to obtain a 3D matrix:

Pixies software

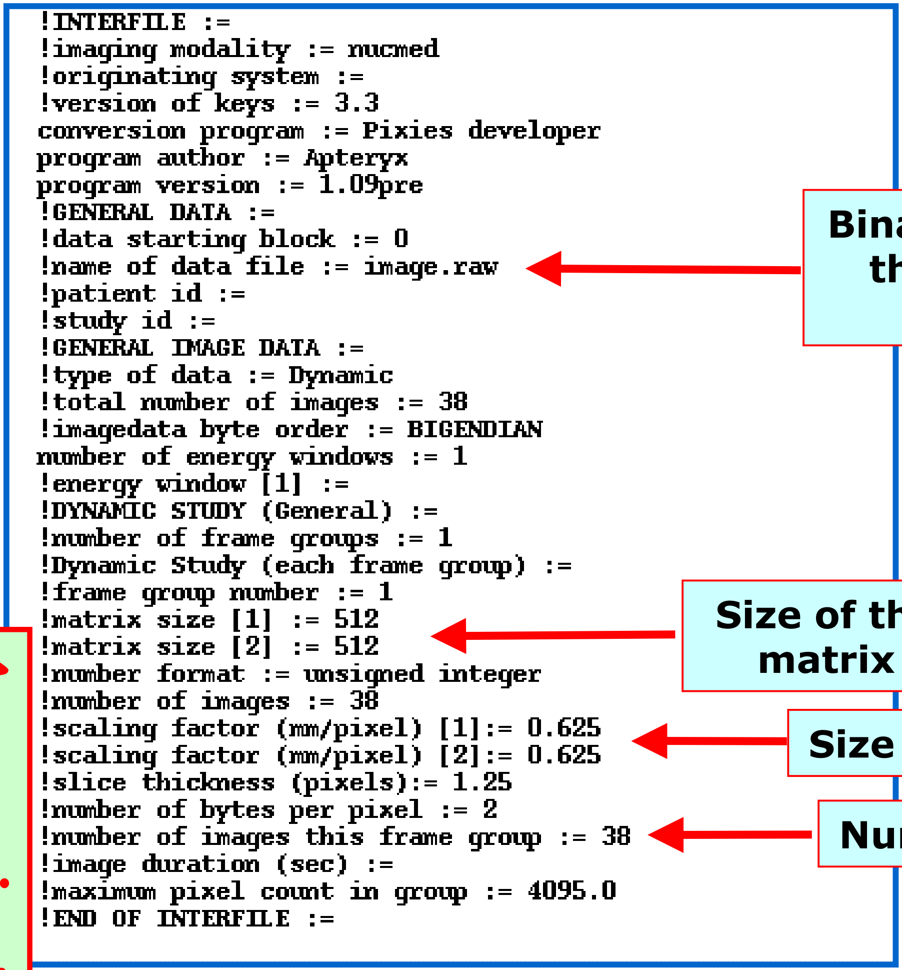




3°) Transform the DICOM format of the image into an Interfile format



Pixies software



```
!INTERFILE :=
!imaging modality := nucmed
!originating system :=
!version of keys := 3.3
conversion program := Pixies developer
program author := Apteryx
program version := 1.09pre
!GENERAL DATA :=
!data starting block := 0
!name of data file := image.raw
!patient id :=
!study id :=
!GENERAL IMAGE DATA :=
!type of data := Dynamic
!total number of images := 38
!imagedata byte order := BIGENDIAN
number of energy windows := 1
!energy window [1] :=
!DYNAMIC STUDY (General) :=
!number of frame groups := 1
!Dynamic Study (each frame group) :=
!frame group number := 1
!matrix size [1] := 512
!matrix size [2] := 512
!number format := unsigned integer
!number of images := 38
!scaling factor (mm/pixel) [1]:= 0.625
!scaling factor (mm/pixel) [2]:= 0.625
!slice thickness (pixels):= 1.25
!number of bytes per pixel := 2
!number of images this frame group := 38
!image duration (sec) :=
!maximum pixel count in group := 4095.0
!END OF INTERFILE :=
```

Binary image of the scanner slices

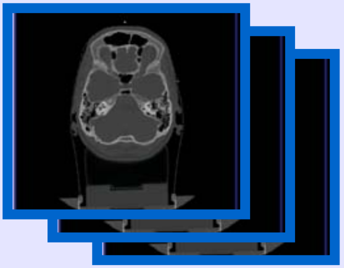
Size of the matrix

Size of the pixels

Number of slices

Anonymisation

4°) Register and replicate the binary image on SEs:



Scanner slices: DICOM format

Concatenation

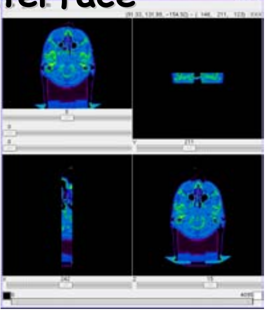
Image: text file

Binary file: Image.raw Size 19M

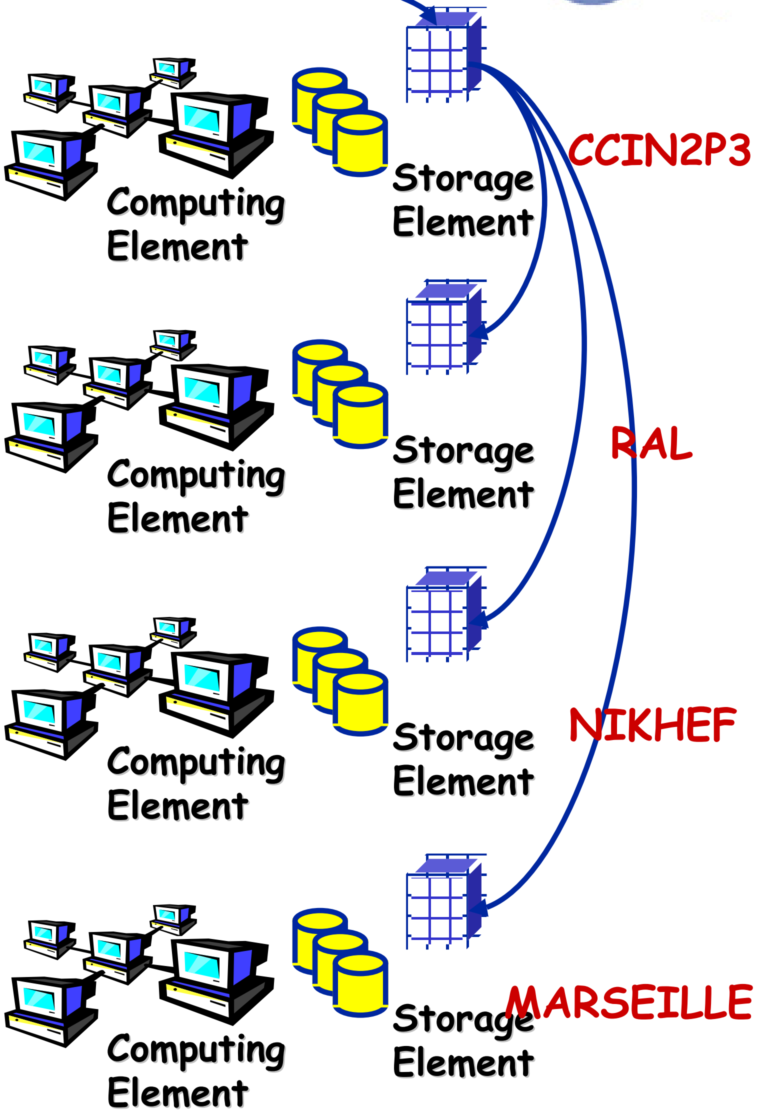
Anonymisation

Replicate lfns on the other SEs

User interface

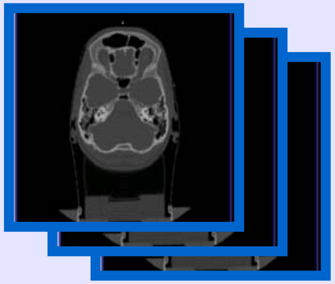


Visualization



5°) Register the lfn of an image:

- WP2 spitfire or local database



Scanner slices: DICOM format

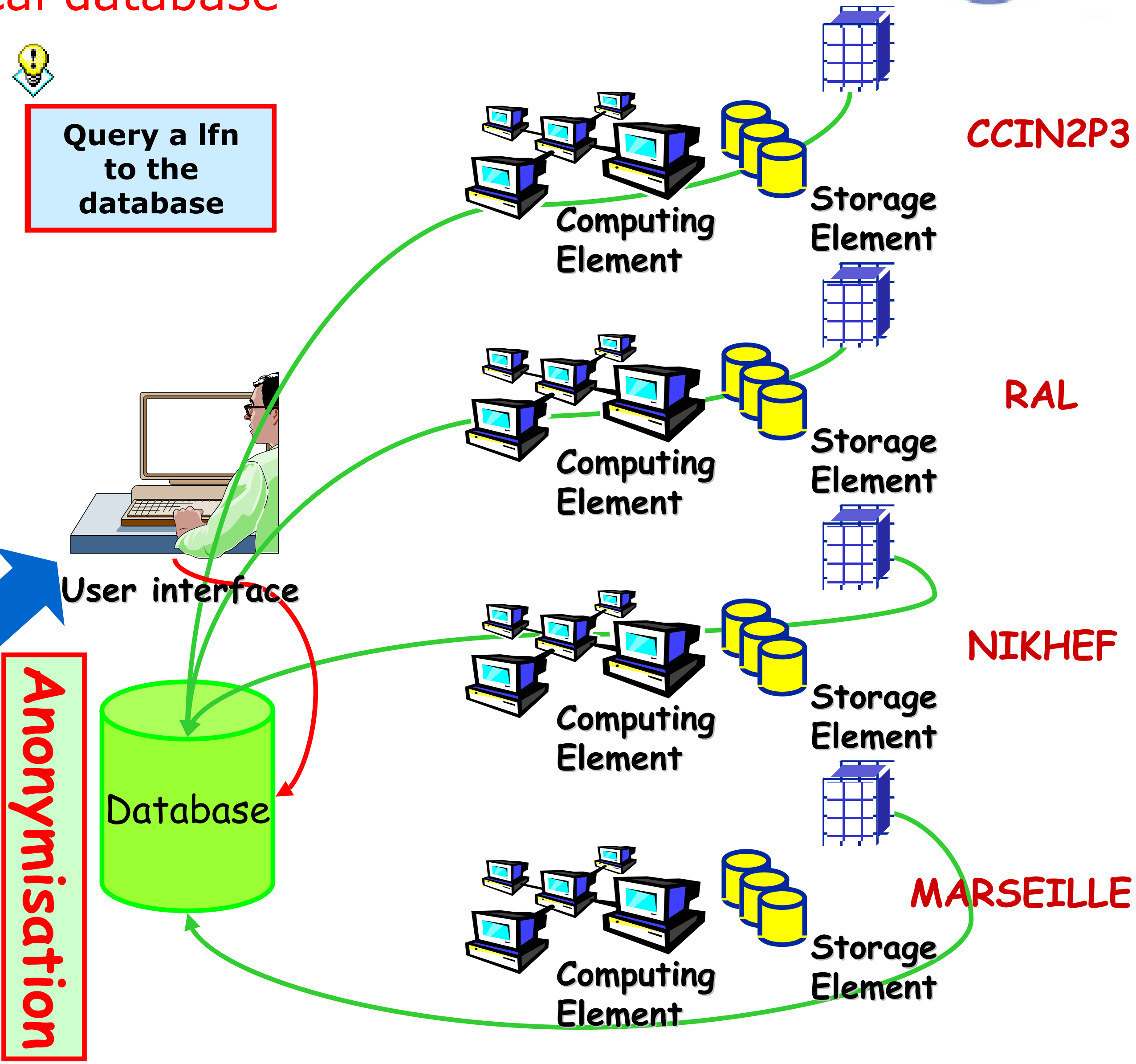
Concatenation

Image: text file

Binary file: Image.raw Size 19M

Anonymisation

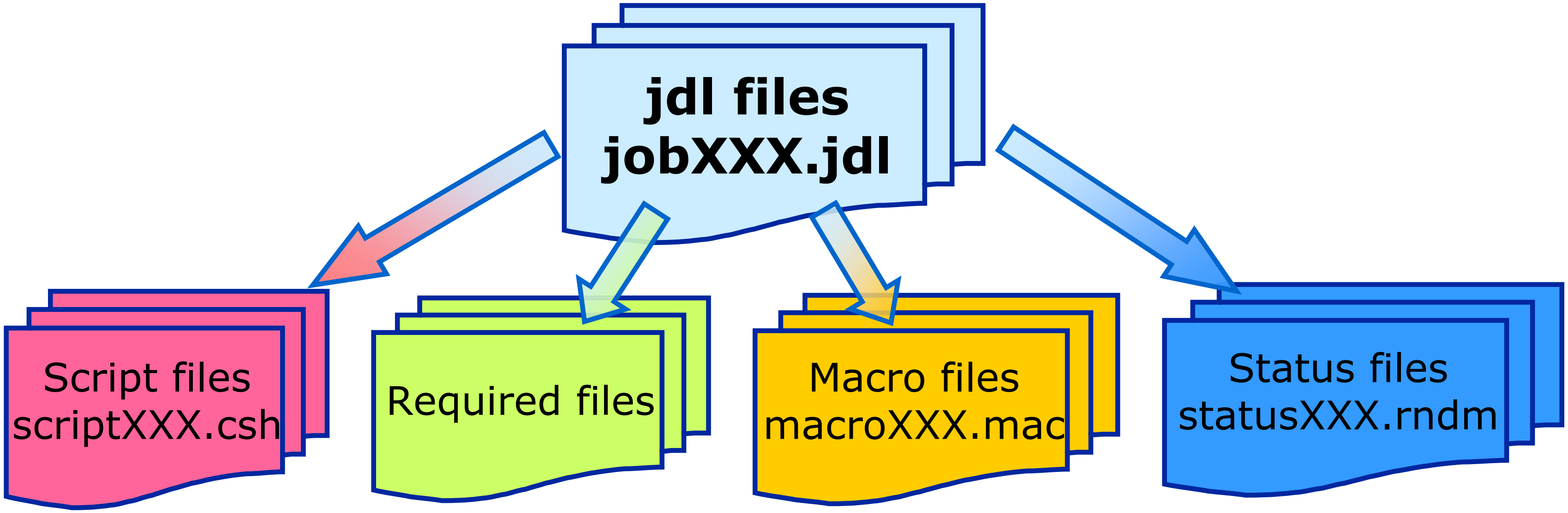
Query a lfn to the database



6°) Split the simulations:

JobConstructor C++ program

- A GATE simulation generating a lot of particles in matter could take a very long time to run on a single processor
  - So, the big simulation generating 10M of particles is divided into little ones, for example
    - 10 simulations generating 1M of particles
    - 20 simulations generating 500000 particles
    - 50 simulations generating 200000 particles ……
  - All the other files needed to launch Monte Carlo simulations are automatically created with the program.



jdl files
jobXXX.jdl

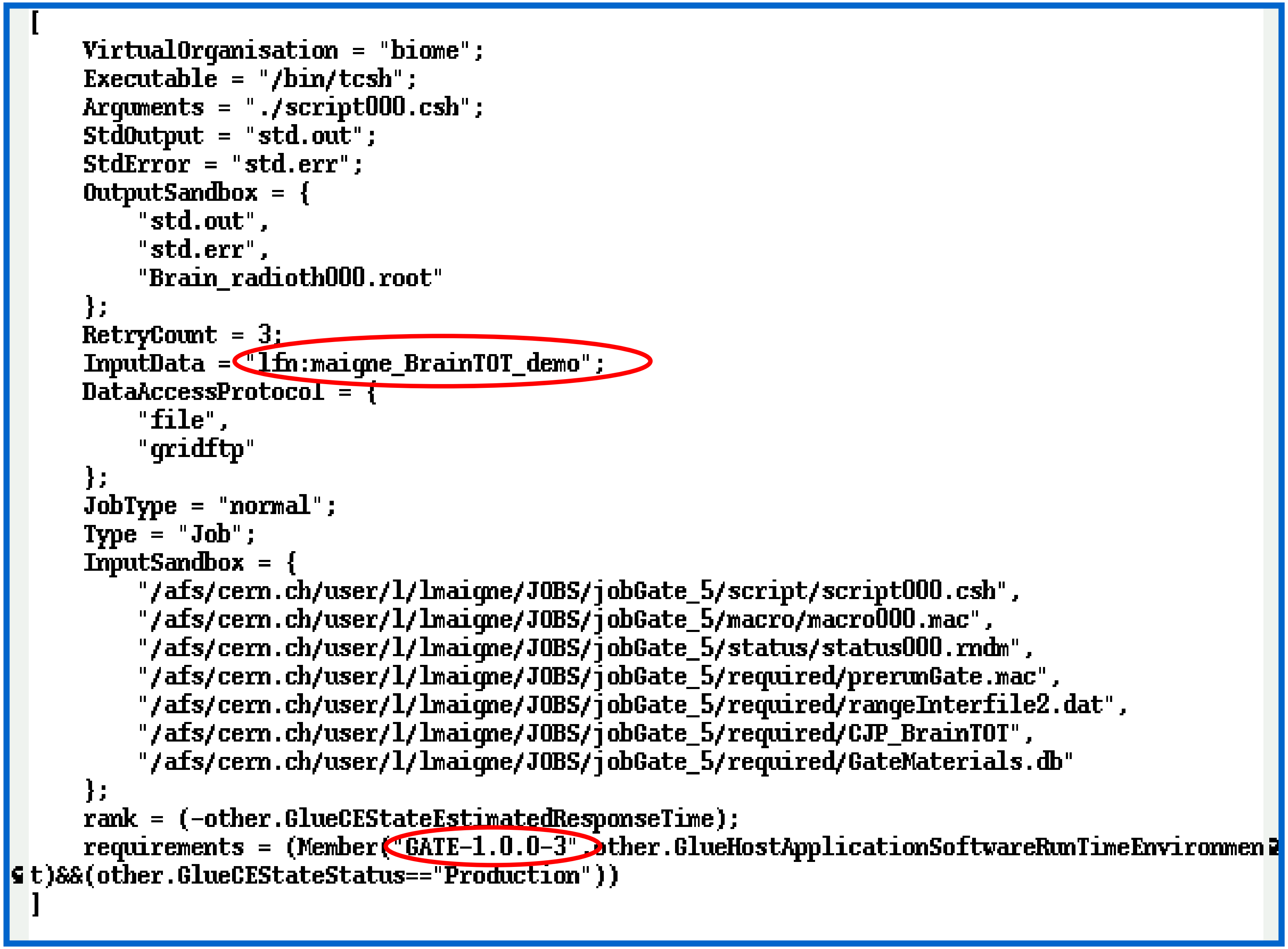
Script files
scriptXXX.csh

Required files

Macro files
macroXXX.mac

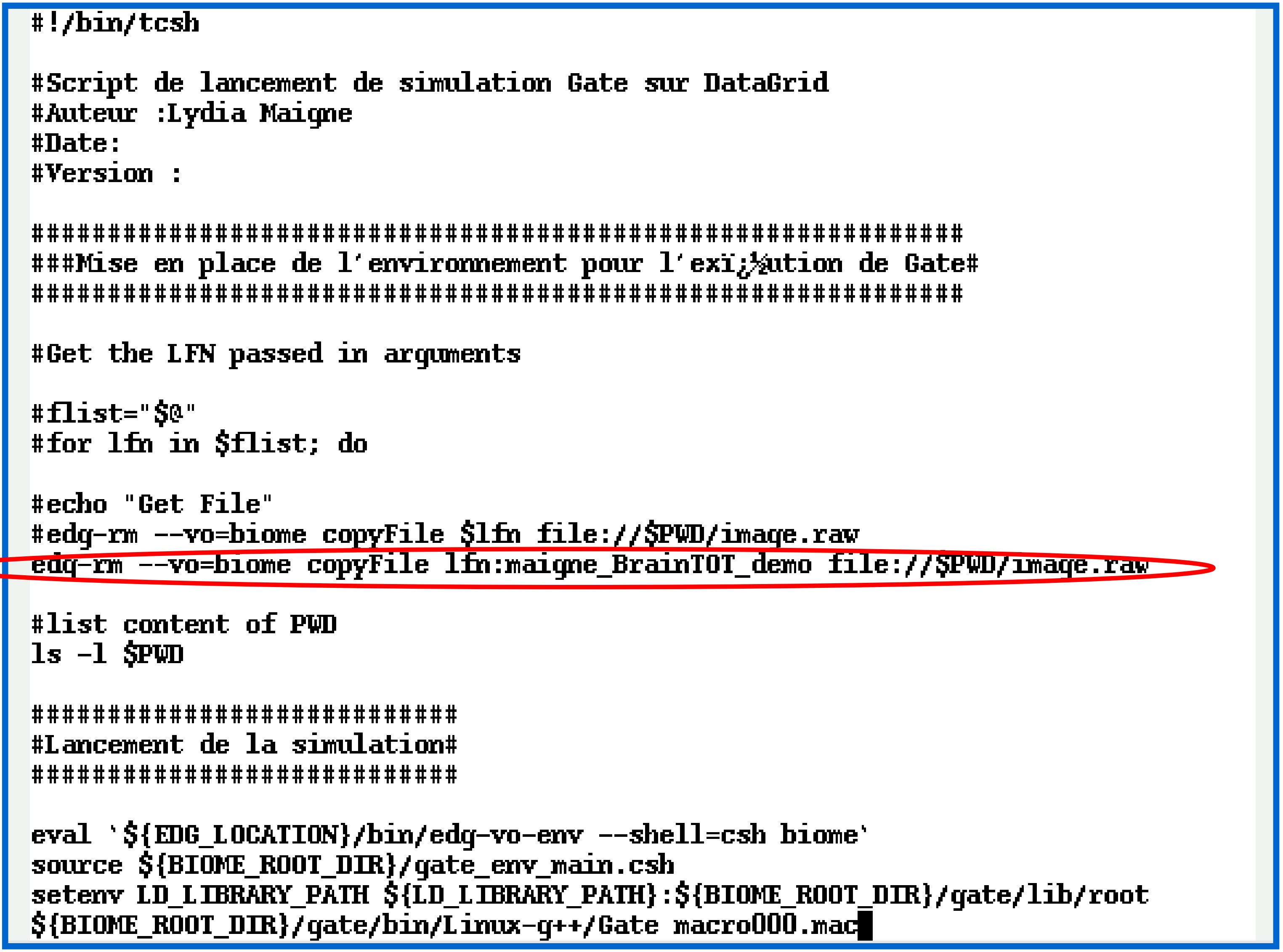
Status files
statusXXX.rndm

A typical jdl file:



```
[
    VirtualOrganisation = "biome";
    Executable = "/bin/tcsh";
    Arguments = "./script000.csh";
    StdOutput = "std.out";
    StdError = "std.err";
    OutputSandbox = {
        "std.out",
        "std.err",
        "Brain_radioth000.root"
    };
    RetryCount = 3;
    InputData = "lfn:maigne_BrainTOT_demo";
    DataAccessProtocol = {
        "file",
        "gridftp"
    };
    JobType = "normal";
    Type = "Job";
    InputSandbox = {
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/script/script000.csh",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/macro/macro000.mac",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/status/status000.rndm",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/required/prerunGate.mac",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/required/rangeInterfile2.dat",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/required/CJP_BrainTOT",
        "/afs/cern.ch/user/l/lmaigne/JOBS/jobGate_5/required/GateMaterials.db"
    };
    rank = (-other.GlueCEStateEstimatedResponseTime);
    requirements = (Member("GATE-1.0.0-3",other.GlueHostApplicationSoftwareRunTimeEnvironmen
t)&&(other.GlueCEStateStatus=="Production"))
]
```

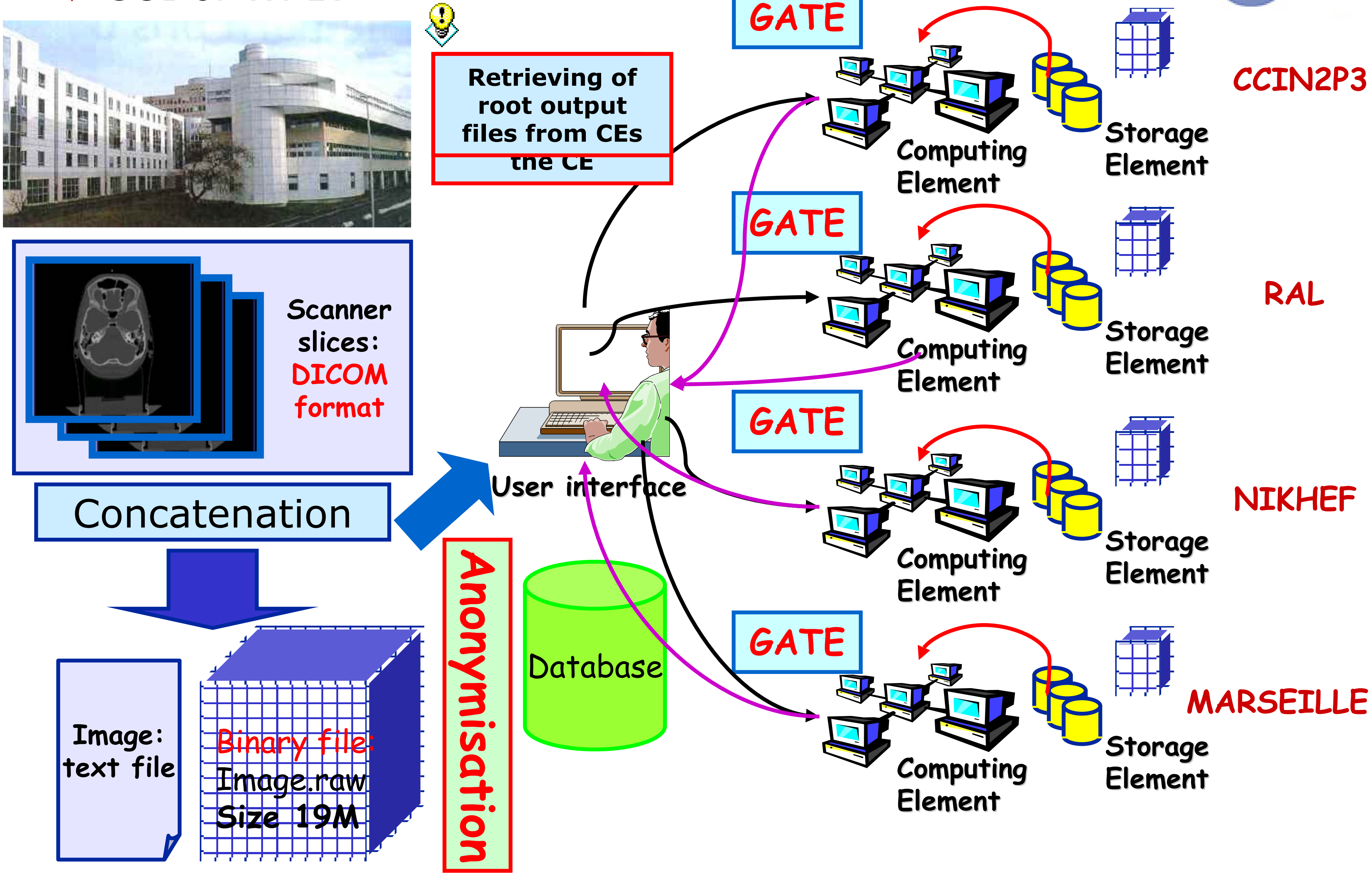
A typical script file:



```
#!/bin/tcsh

#Script de lancement de simulation Gate sur DataGrid
#Auteur :Lydia Maigne
#Date:
#Version :

#############################################################
###Mise en place de l'environnement pour l'exï¿½ution de Gate#
#############################################################

#Get the LFN passed in arguments

#flist="$@"
#for lfn in $flist; do

#echo "Get File"
#edg-rm --vo=biome copyFile $lfn file://$PWD/image.raw
edg-rm --vo=biome copyFile lfn:maigne_BrainTOT_demo file://$PWD/image.raw

#list content of PWD
ls -l $PWD

############################
#Lancement de la simulation#
############################

eval `${EDG_LOCATION}/bin/edg-vo-env --shell=csh biome`
source ${BIOME_ROOT_DIR}/gate_env_main.csh
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${BIOME_ROOT_DIR}/gate/lib/root
${BIOME_ROOT_DIR}/gate/bin/Linux-g++/Gate macro000.mac
```

7°) Submission on the DataGrid:

- GUI of WP1:



Retrieving of root output files from CEs
the CE

Scanner slices: DICOM format

Concatenation

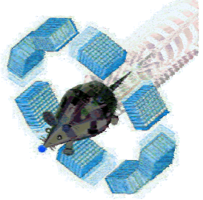
Image: text file

Binary file: Image.raw Size 19M

Anonymisation

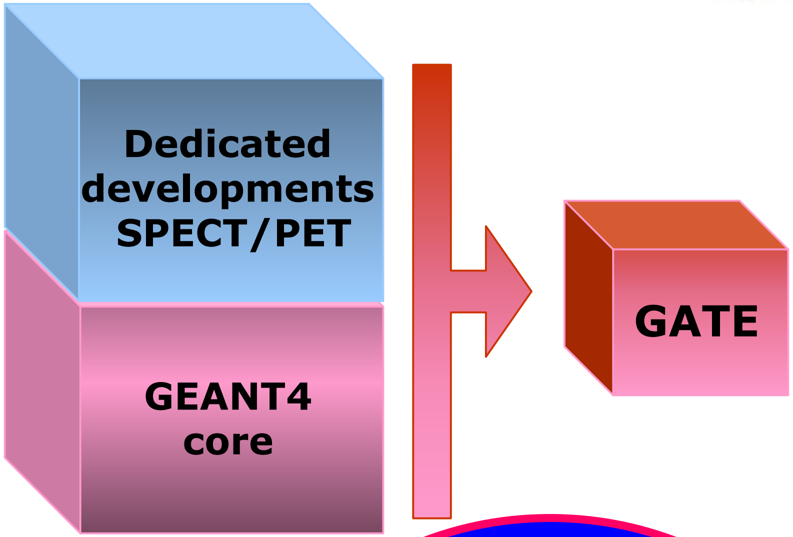
Database

User interface

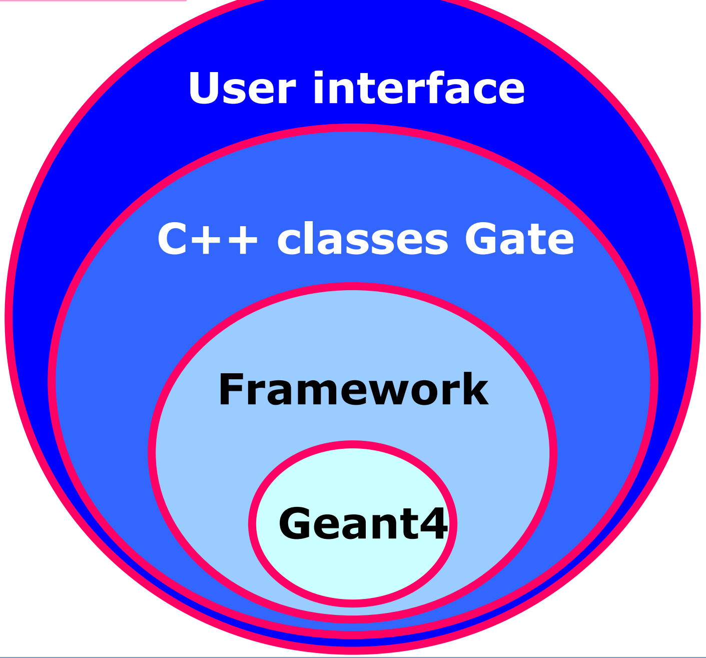


GATE: Geant4 Application for Tomographic Emission

- **Develop a simulation platform for SPECT/PET imaging**
  - **Based on Geant4**
  - **Enrich Geant4 with dedicated tools SPECT/PET**
  - **User friendly**
- **Ensure a long term development**
  - **Effort of shared development**
  - **Collaboration: OpenGATE**

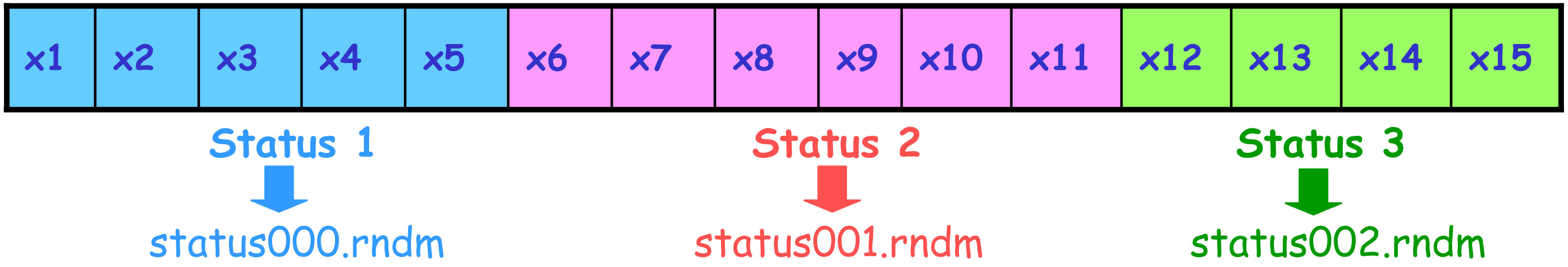


- **Based on Geant4**
  - **C++ object oriented langage**
  - **Reliable cross sections**
- **Framework: interface**
- **GATE development**
  - **modelisation of detectors, sources, patient**
  - **movement (detector, patient)**
  - **time-dependent processes (radioactive decay, movement management, biological kinetics)**
- **Ease of use**
  - **Command scripts to define all the parameters of the simulation**



Parallelization technique of GATE

- The random numbers generator (RNG) in GATE
  - CLHEP libraries: HEPJamesRandom (deterministic algorithm of F.James)
    - Characteristics:
      - Very long period RNG: 2^{144}
      - Creation of 900 million sub-sequences non overlapping with a length of 10^{30}
  - Pregeneration of random numbers
    - The Sequence Splitting Method



x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15

Status 1 → status000.rndm

Status 2 → status001.rndm

Status 3 → status002.rndm

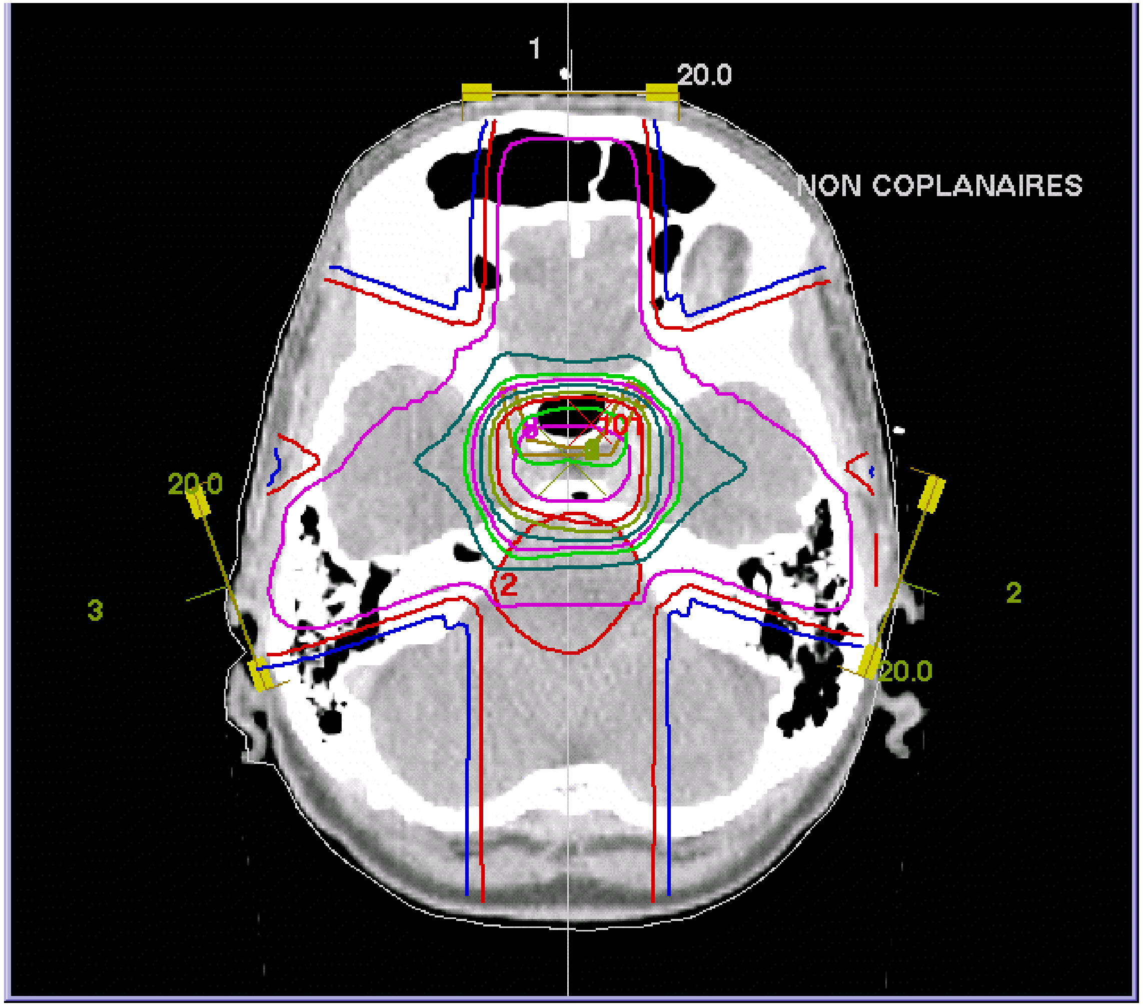
- Until now, 200 status files generated with a length of 3.10^{10}

Each status file is sent on the grid with a GATE simulation

8°) Analysis of output root files

- Typical dosimetry:
  - Merging of all the root files
  - Computation of the root data

Brain_radioth000.root:	**20 MB**
Brain_radioth001.root:	**20 MB**
Brain_radioth002.root:	**20 MB**
Brain_radioth003.root:	**20 MB**
Brain_radioth004.root:	**20 MB**
Brain_radioth005.root:	**20 MB**
Brain_radioth006.root:	**20 MB**
Brain_radioth007.root:	**20 MB**
Brain_radioth008.root:	**20 MB**
Brain_radioth009.root:	**20 MB**

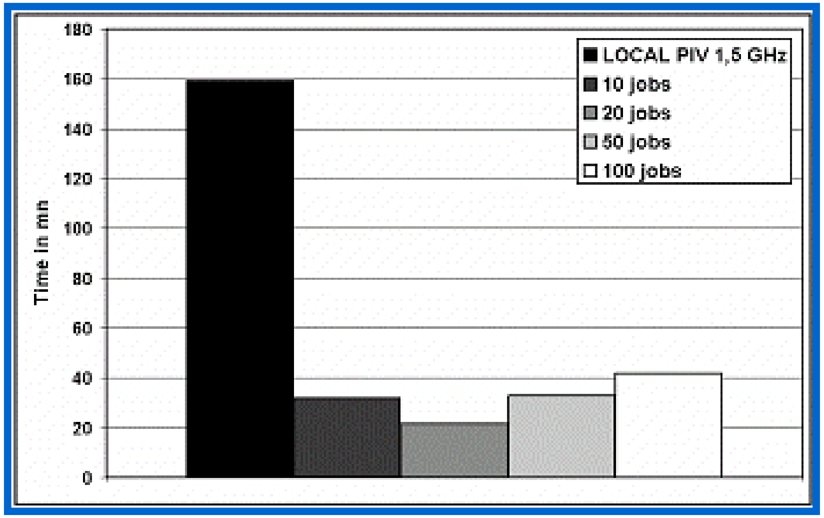


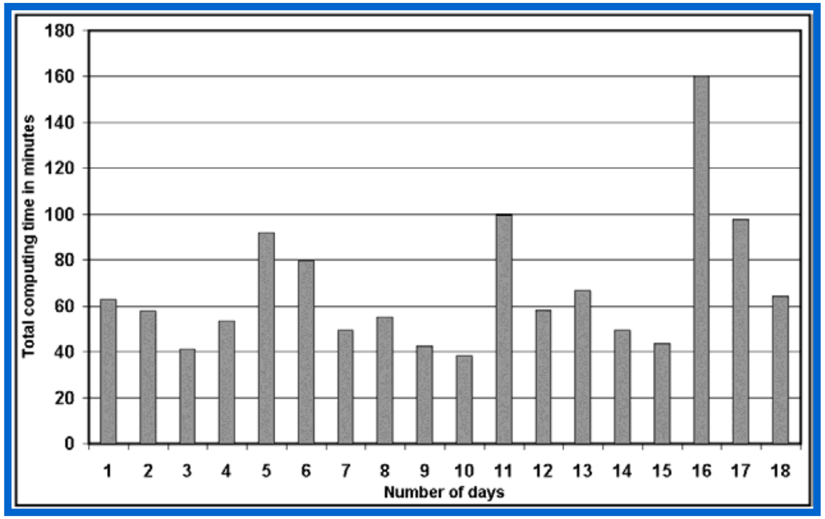
transversal view

Centre Jean Perrin
Clermont-Ferrand

Conclusion and future prospects

- The parallelization of GATE on the DataGrid testbed has shown significant gain in computing time (factor 10)





- It is not sufficient for clinical routine
- Necessary improvements
  - Dedicated resources (job prioritization)
  - Graphical User interface

Aknowledgements

- WP1:
  - Graphical User Interface, JobSubmitter
- WP2:
  - Spitfire
- WP6:
  - RPMs of GATE
- WP8
- WP10:
  - 4D Viewer (Creatis)
  - Centre Jean Perrin
  - LIMOS
- System administrators
  - Installations on UIs