Simplifying the Utilization of Grid Computation using Grid Wizard Enterprise

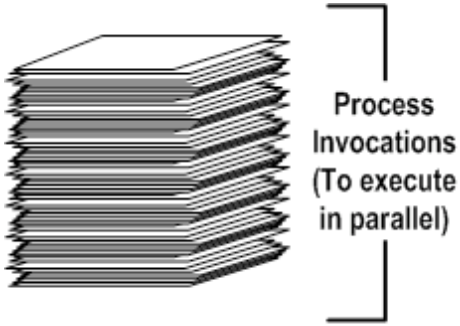
Introduction

- Typical computation intensive problems in research in computation sciences:
  1. Refinement of computational protocol.

     Iteratively improve computational protocol by testing each round of the applications against different algorithmic parameters. (*Parameter exploration*).
  2. Usage of released computational protocol applications.

     Process large amounts of pathological inputs using the particular application. (*Dataset processing*).
- Both of these are embarrassingly parallel problems.

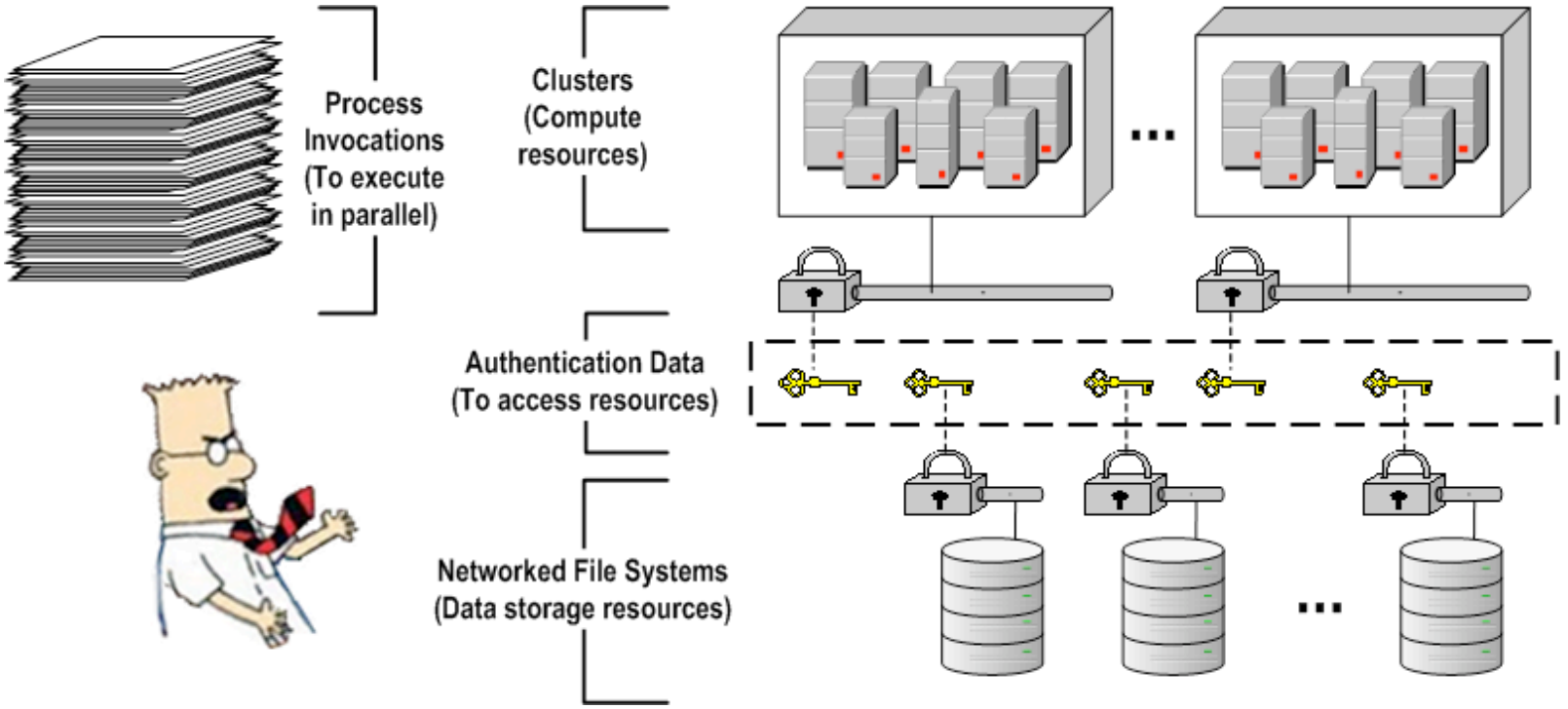
Embarrassingly Parallel Problem



- Embarrassingly parallel problem (EPP) is the one faced when trying to execute in parallel a collection of inter-independent process invocations.
- Inter-independent processes are those which don't have any execution related dependencies from each other.
- These processes are ideally suited to execute in parallel by distributing their execution across multiple processing units such as clusters of computers.
- EPP is also known as "embarrassingly parallel workload".

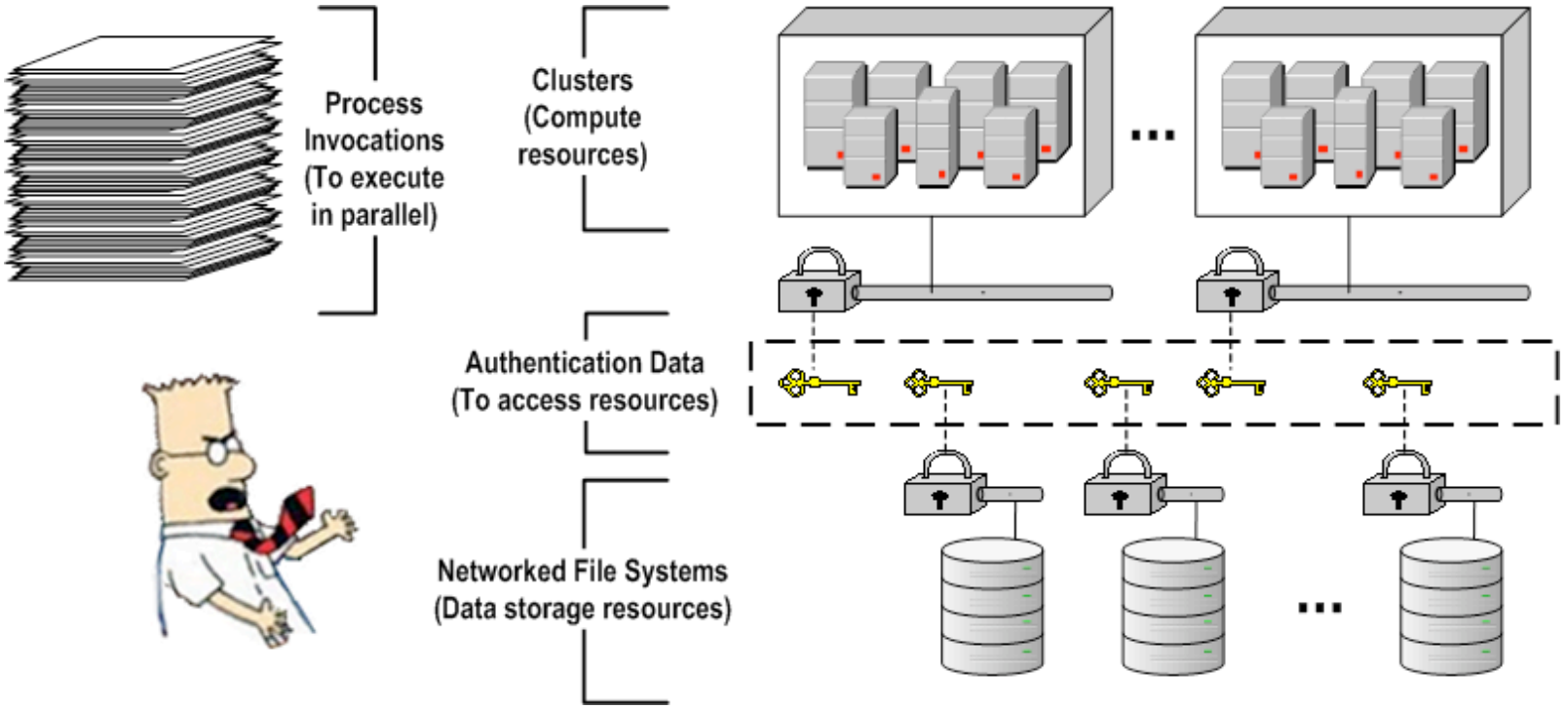
Distributed Solution for EPP

- Solution: Distribute the execution of processes over an infrastructure consisting of cluster(s) of computers, their resource managers (Condor, PBS, SGE) and networked file systems (where inputs/outputs are/will be stored).
- To use this infrastructure, researchers required programming and system administrator skills; which most of the time they don't posses.



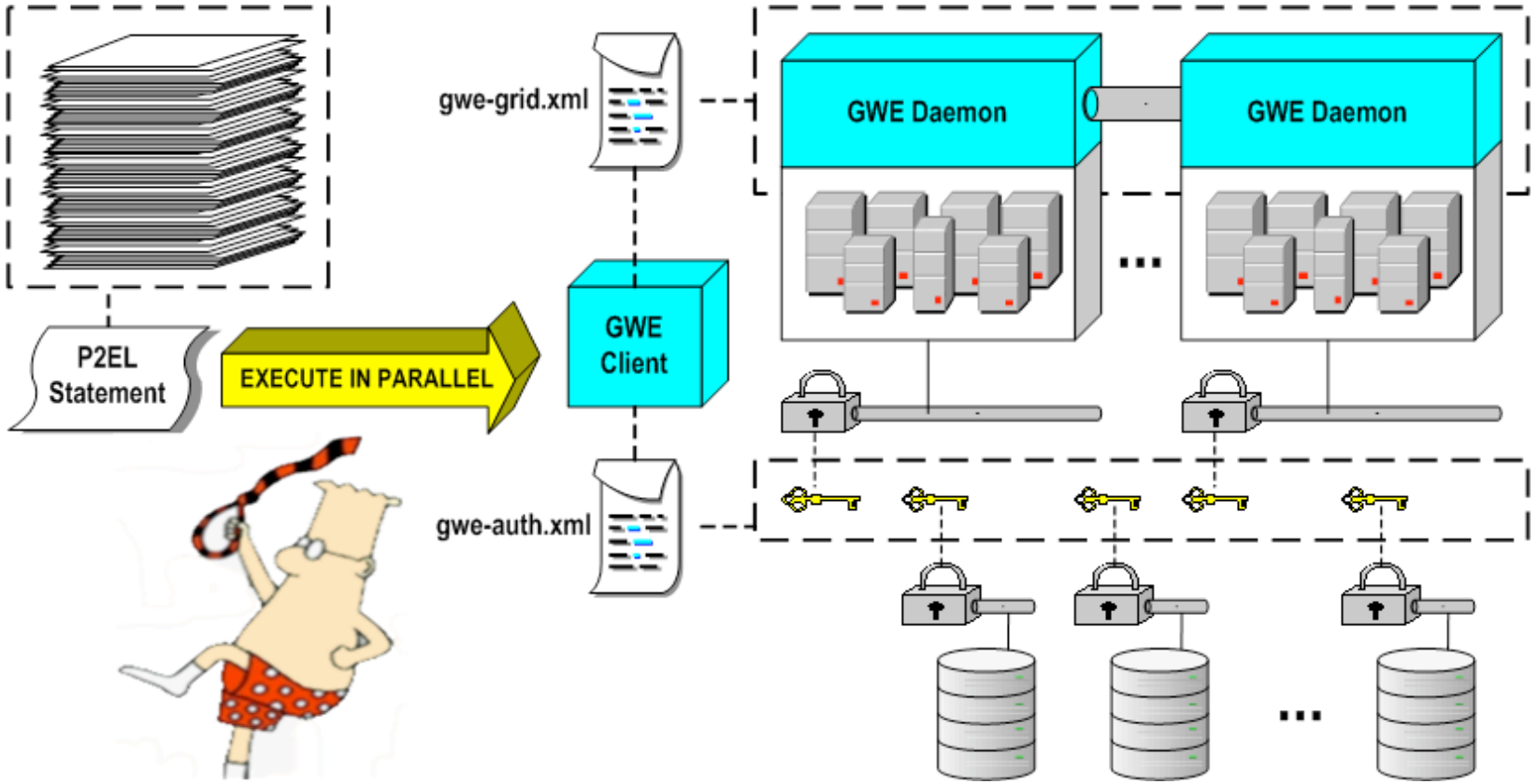
Distributed Solution for EPP

- Even with such skills the implementing this solution is non-trivial.
- Common tasks: describe processes, queue them for execution, prepare them, monitor their progress, collect and consolidate their results, wrap them up.
- Users can take advantage of an easy to use solution that provides generic, cohesive strategies to address common tasks.



GWE's Solution

- GWE: Distributed system intended to ease the effort of executing in parallel inter-independent processes across clusters.
- Low requirements! Only SSH enabled clusters and Java 1.5.



GWE Usage

- Quick Start Guide:
  1. Install GWE on your machine.
  2. Configure GWE installed with:
     - Authentication information to access clusters and file systems.
     - Description of computational grid as a collection of clusters.
  3. Run “GWE daemons” installer utility.
  4. Launch a GWE client.
  5. Interact with your defined grid using your GWE client!
- Interaction features:
  1. Queuing a set of process invocations described through P2EL.
  2. Real time and on demand progress monitoring and result status.
  3. Execution control: pause, resume, abort.

P2EL

- P2EL = Processes Parallel Execution Language.
- Language especially designed to allow a single statement to describe a collection of inter-independent process invocations.
- Semantics to allow versatile permutations to generate process invocations.
- P2EL statement composition:
    1. Variables. Set of variables each associated with a particular value set (evaluated through a value set generator function invocation).
    2. Process Invocations Template. Process invocation with variable to value substitution expressions.
- Permutation of the variables values. Creates a set of all the unique variable to value resolution combinations of a statement's variables, respecting the variables semantics (multidimensionality, co-dependency, etc).
- The full language specification (syntactic and semantic rules) is described in the P2EL guide on the GWE's project site.

P2EL Sample: Dataset Processing

- **"Free Surfer" Subject Cases Processor**:

```
${PATH}=sftp://sourceHost/subjectsPath
${FILES}=$dir(${PATH},.*)
${SUBJ_ID}=$regExp(${FILES}, /, [^/]*, $)
${INPUT_DIR}=$in(${FILES})
${OUTPUT_DIR}=$out(${PATH}/results/${SUBJ_ID})

${SYSTEM.USER_HOME}/RunFreesurfer.sh ${INPUT_DIR} ${OUTPUT_DIR}
```

- This command instructs GWE to download all remote directories that match a given pattern and execute the RunFreesurfer.sh script against each one of them in parallel. That same command instructs GWE as well to upload the directory generated by the script, to a remote host with the given, parameterized name.

P2EL Sample: Parameter Exploration

- **Slicer's BSpline Deformable Image Registration**:

```
${ITER}=$range(10,50,5)
${HIST}=$range(20,100,010)
${SAM}=$range(500,5000,0750)
${OUTPUT}=$out(sftp://destinationHost/path/out-${ITER}-${HIST}-${SAM}.nrrd)
${FILES_DIR}=http://www.na-mic.org/ViewVC/index.cgi/trunk/Libs/MRML/Testing/TestData
${FIXED}=$in(${FILES_DIR}/fixed.nrrd?view=co,fixed.nrrd)
${MOVING}=$in(${FILES_DIR}/moving.nrrd?view=co,moving.nrrd)

${SYSTEM.USER_HOME}/Slicer3/Slicer3 --launch
    ${SYSTEM.USER_HOME}/Slicer3/lib/Slicer3/Plugins/BSplineDeformableRegistration --
    iterations ${ITER} --gridSize 5 --histogrambins ${HIST} --spatialsamples ${SAM}
    --maximumDeformation 1 --default 0 --resampledmovingfilename ${OUTPUT} ${FIXED}
    ${MOVING}
```

- This command instructs GWE to execute in parallel 700 BSplineDeformableRegistration parameter exploration type of invocations and, upon completion, upload each result image to a remote host with a given parameterized name.

GWE Client API

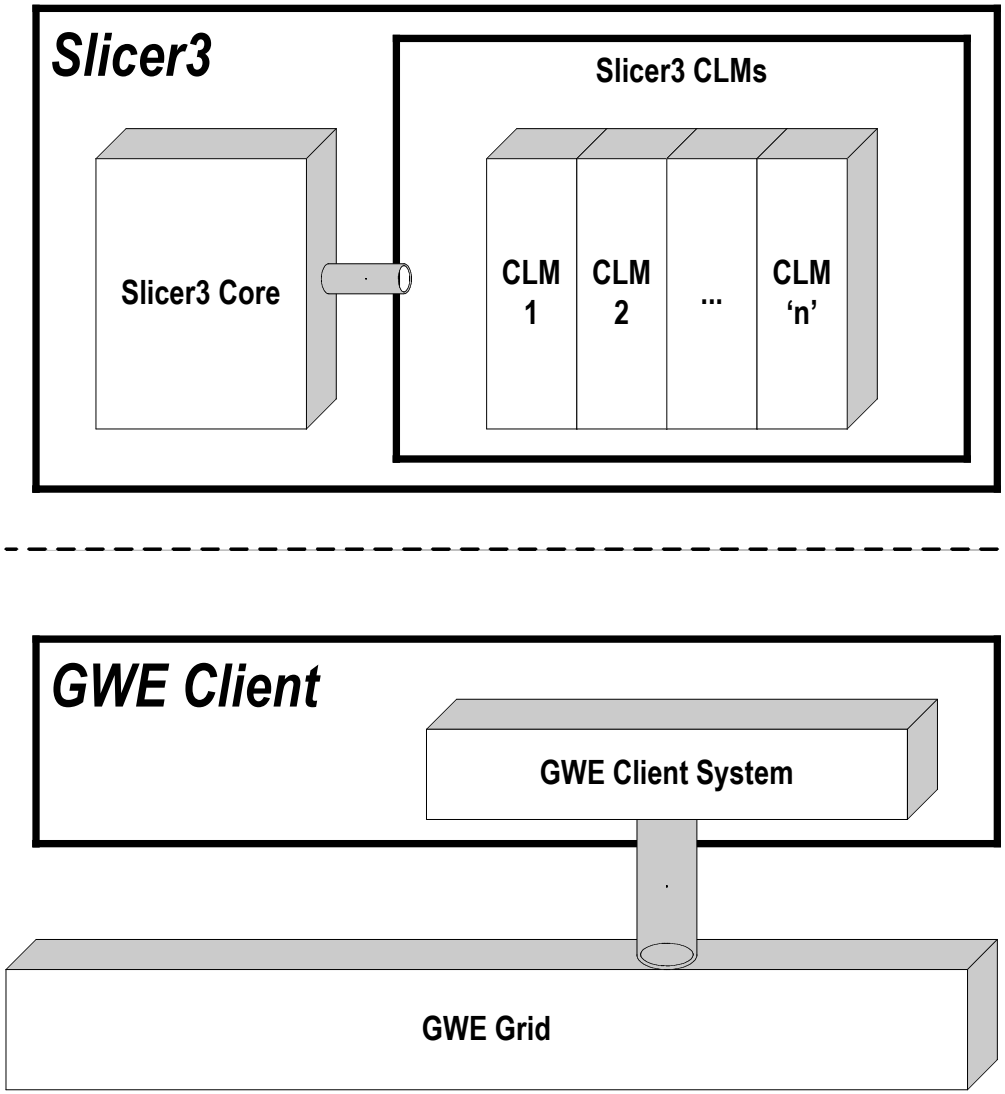
- Programmatic, full featured, API to access “GWE Grid”s services (interact with “GWE daemons”).
- Secured RPC communications layer using RMI over SSH Tunnels.
- “GWE Client”s are applications built on top of this API.
- Samples: GWE Terminal GWE Commands and GSlicer3.

Remote GWE Daemon Setup Sample

```
[mruiz@MarcosMac]/usr/local/gwe:$ gwe-terminal.sh
================================================

       Welcome to GWE Terminal.
       You are connected to mruiz@birn-cluster0.nbirn.net:1099.

================================================


GWE [mruiz@birn-cluster0.nbirn.net:1099] > list-orders
Id      Owner   Submitted               Completed               Progress
1       admin   2008-02-09 02:45:42.658 2008-02-09 02:47:08.76  30 / 30
2       admin   2008-02-09 02:51:03.206 2008-02-09 02:51:04.15  30 / 30
3       admin   2008-02-09 02:51:55.127 2008-02-09 02:51:55.942 30 / 30
4       admin   2008-02-09 02:58:07.427 2008-02-09 02:58:08.534 30 / 30
5       admin   2008-02-09 02:59:11.495 2008-02-09 02:59:21.947 525 / 525
6       admin   2008-02-09 03:00:32.77  2008-02-09 03:00:33.245 30 / 30
7       admin   2008-02-09 03:01:05.249 2008-02-09 03:01:06.174 30 / 30
8       admin   2008-02-09 03:03:18.024 2008-02-09 03:03:18.601 30 / 30

GWE [mruiz@birn-cluster0.nbirn.net:1099] > queue-order ${i}=[1..5];${j}=[1..10] echo permutation i=${i} with j=${j}
Order queued with id 9
GWE [mruiz@birn-cluster0.nbirn.net:1099] > list-orders
Id      Owner   Submitted               Completed               Progress
1       admin   2008-02-09 02:45:42.658 2008-02-09 02:47:08.76  30 / 30
2       admin   2008-02-09 02:51:03.206 2008-02-09 02:51:04.15  30 / 30
3       admin   2008-02-09 02:51:55.127 2008-02-09 02:51:55.942 30 / 30
4       admin   2008-02-09 02:58:07.427 2008-02-09 02:58:08.534 30 / 30
5       admin   2008-02-09 02:59:11.495 2008-02-09 02:59:21.947 525 / 525
6       admin   2008-02-09 03:00:32.77  2008-02-09 03:00:33.245 30 / 30
7       admin   2008-02-09 03:01:05.249 2008-02-09 03:01:06.174 30 / 30
8       admin   2008-02-09 03:03:18.024 2008-02-09 03:03:18.601 30 / 30
9       admin   2008-02-09 03:04:53.419 2008-02-09 03:04:54.138 50 / 50

GWE [mruiz@birn-cluster0.nbirn.net:1099] > view-order 9
Order Id: 9
Description: Order from null
Jobs Descriptor:
 #foreach(${i} in [1..5])
#foreach(${j} in [1..10])
 echo permutation i=${i} with j=${j}
#end
#end

Owner: admin
Submitted at: 2008-02-09 03:04:53.419
Completed at: 2008-02-09 03:04:54.138
Progress: 50/50

GWE [mruiz@birn-cluster0.nbirn.net:1099] > view-result 9 1
permutation i=1 with j=1
```

Tool Integration - GSlicer3: Architecture

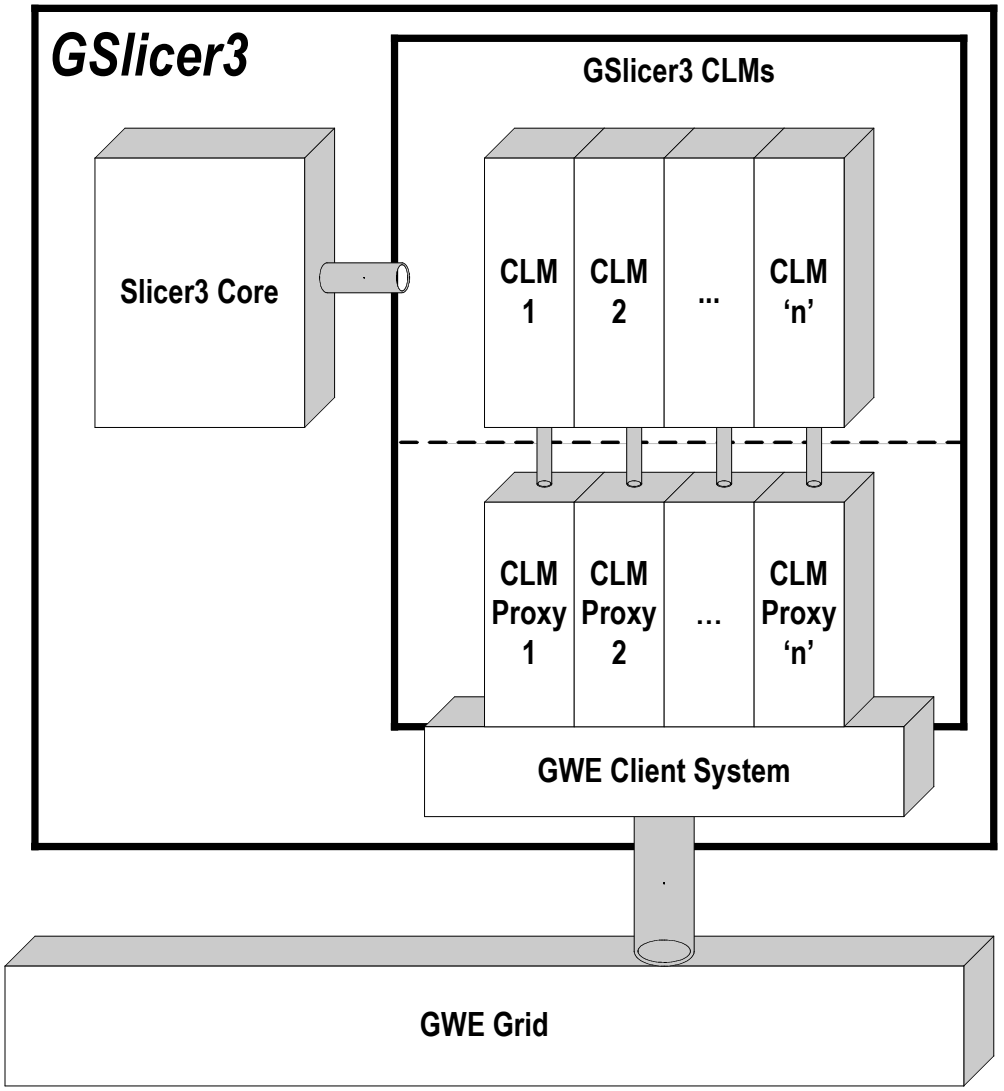
- “Slicer3” and “GWE Client API” are two independent products.
- The goal of the integration effort is to provide Slicer3 with grid computing capabilities out of the box through GWE.
- This effort consists on merging a Slicer3 distribution, a “GWE Client API” distribution and “GWE CLM Proxys” (CLMP).
- The result is a “GWE Client” application we call GSlicer3.
- The integration effort also includes a utility that generates GSlicer3 bundles out of Slicer3 and GWE distributions.



Tool Integration - GSlicer3: Architecture

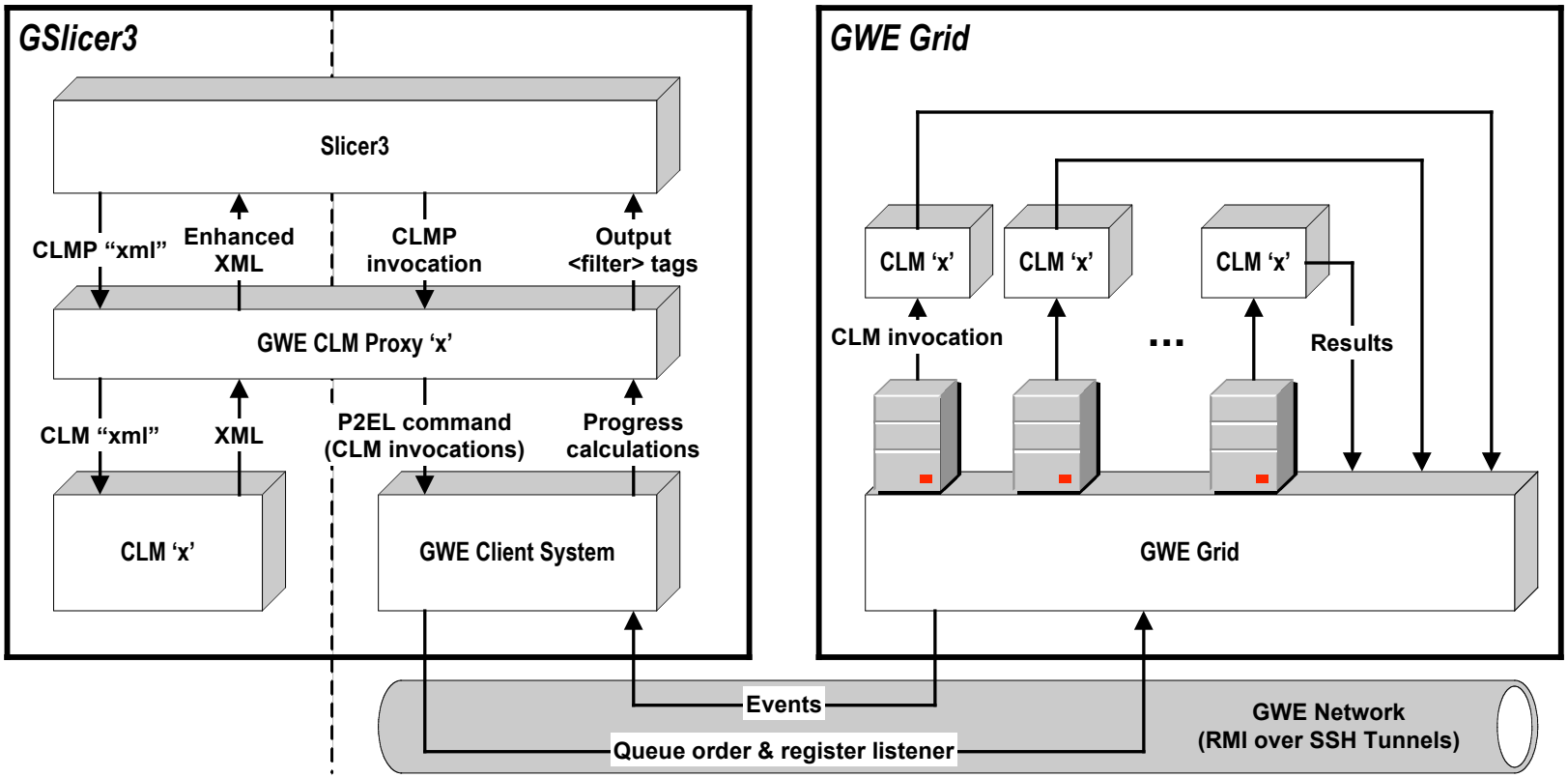
- ***GWE CLM Proxys (CLMP):***
  Slicer3 CLMs which will proxy into another (proxied CLM) to provide a “GWE Powered” version of the proxied CLM.

- ***Technology Requirements:***
  Out of all CLMs discovered in a Slicer3 distribution; only those complying with the “Standard Execution Model” specification will be able to have an automatic CLMP created for them.



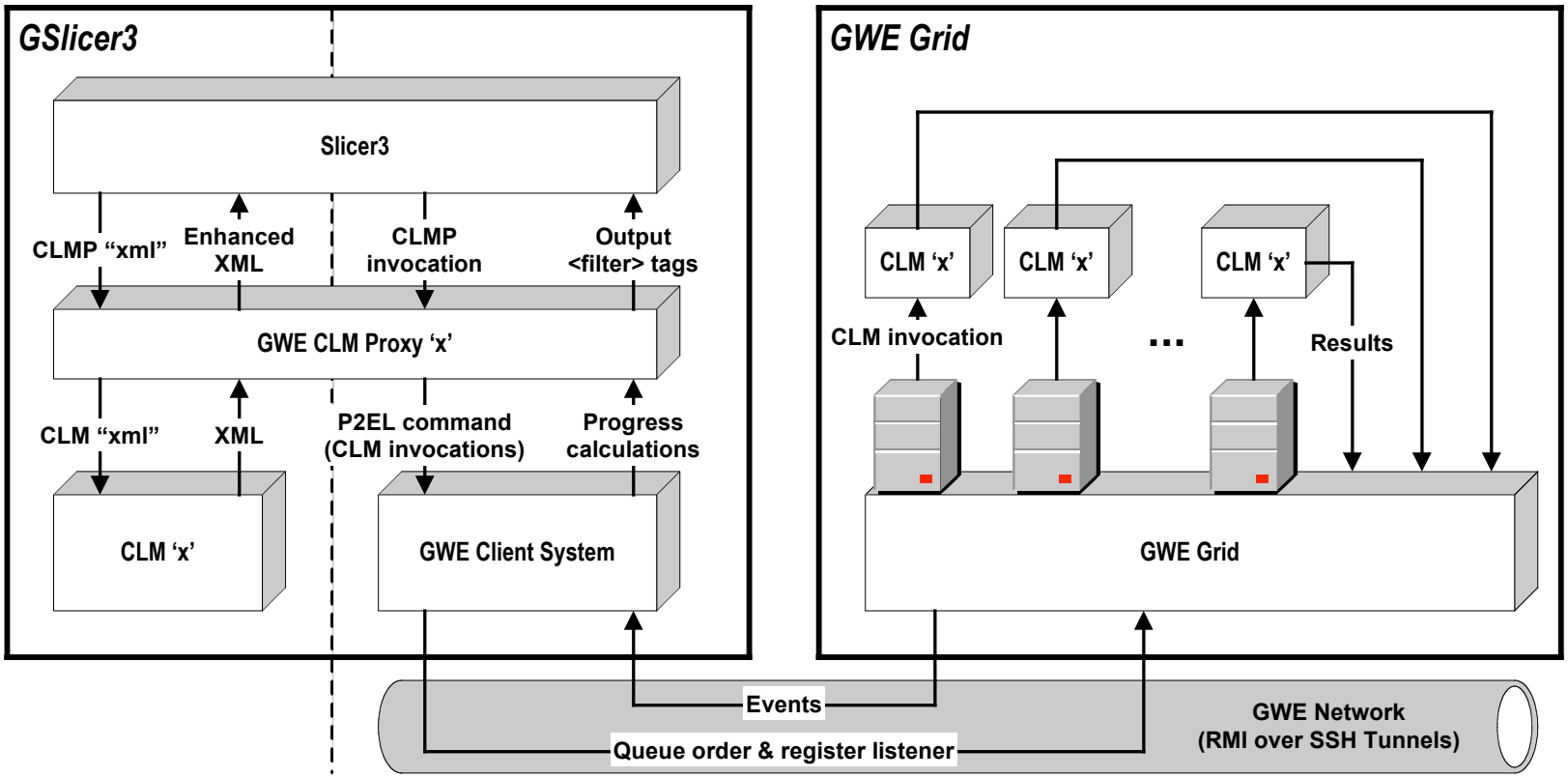
Tool Integration - GSlicer3: CLM Proxy Flow

- Gathers proxied CLM “xml” and enhance it to add GWE support.
- Generate P2EL commands based on GUI input and meta parameter values.
- Submit GWE order representing the group of proxied CLM invocations (P2EL).

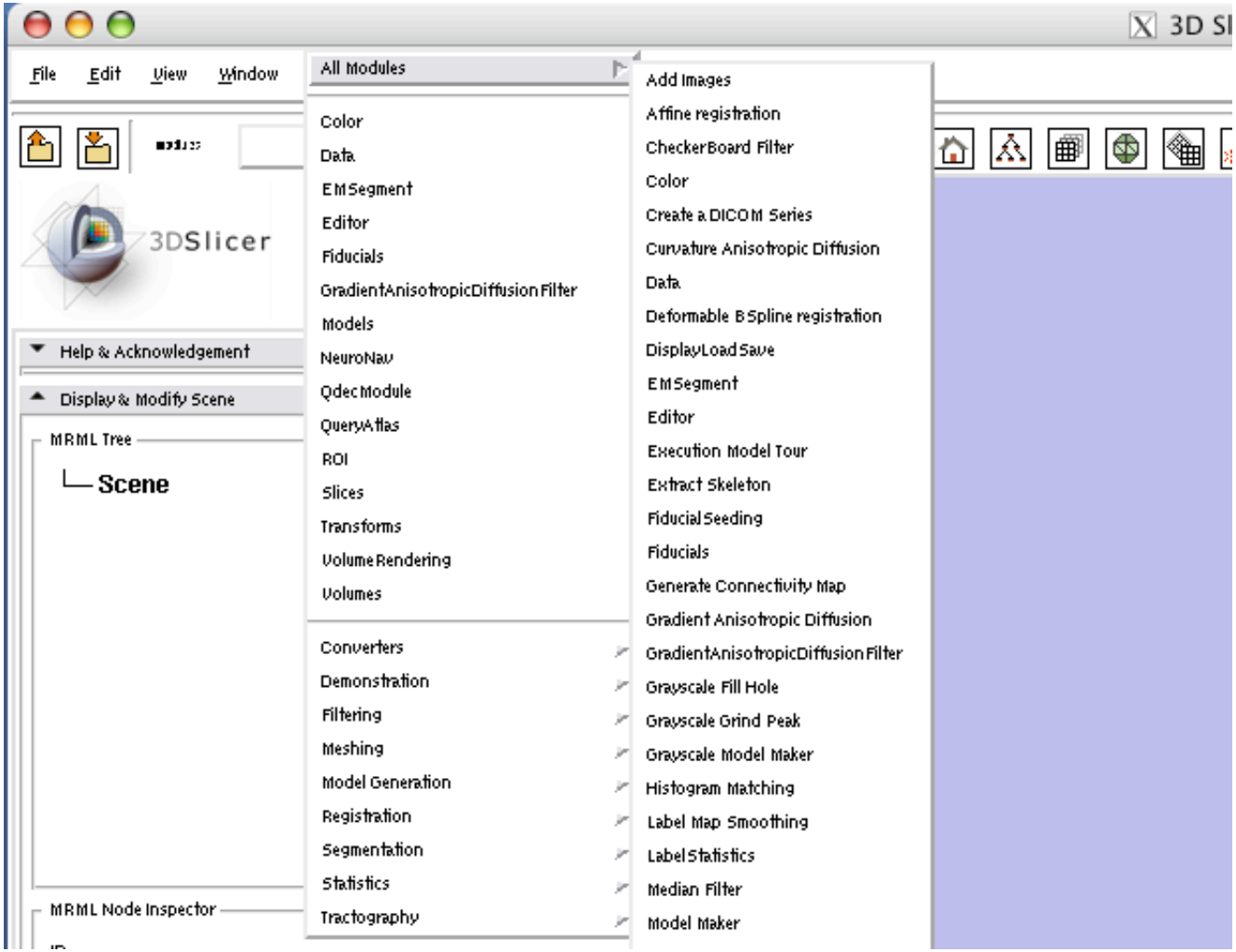


Tool Integration - GSlicer3: CLM Proxy Flow

- Monitors the execution on the user's grid of the localized proxied CLM invocations.
- Keeps track of the CLMP progress as the percentage of invocations executed.
- Notifies Slicer3 of the CLMP progress using Slicer3's XML based progress API.



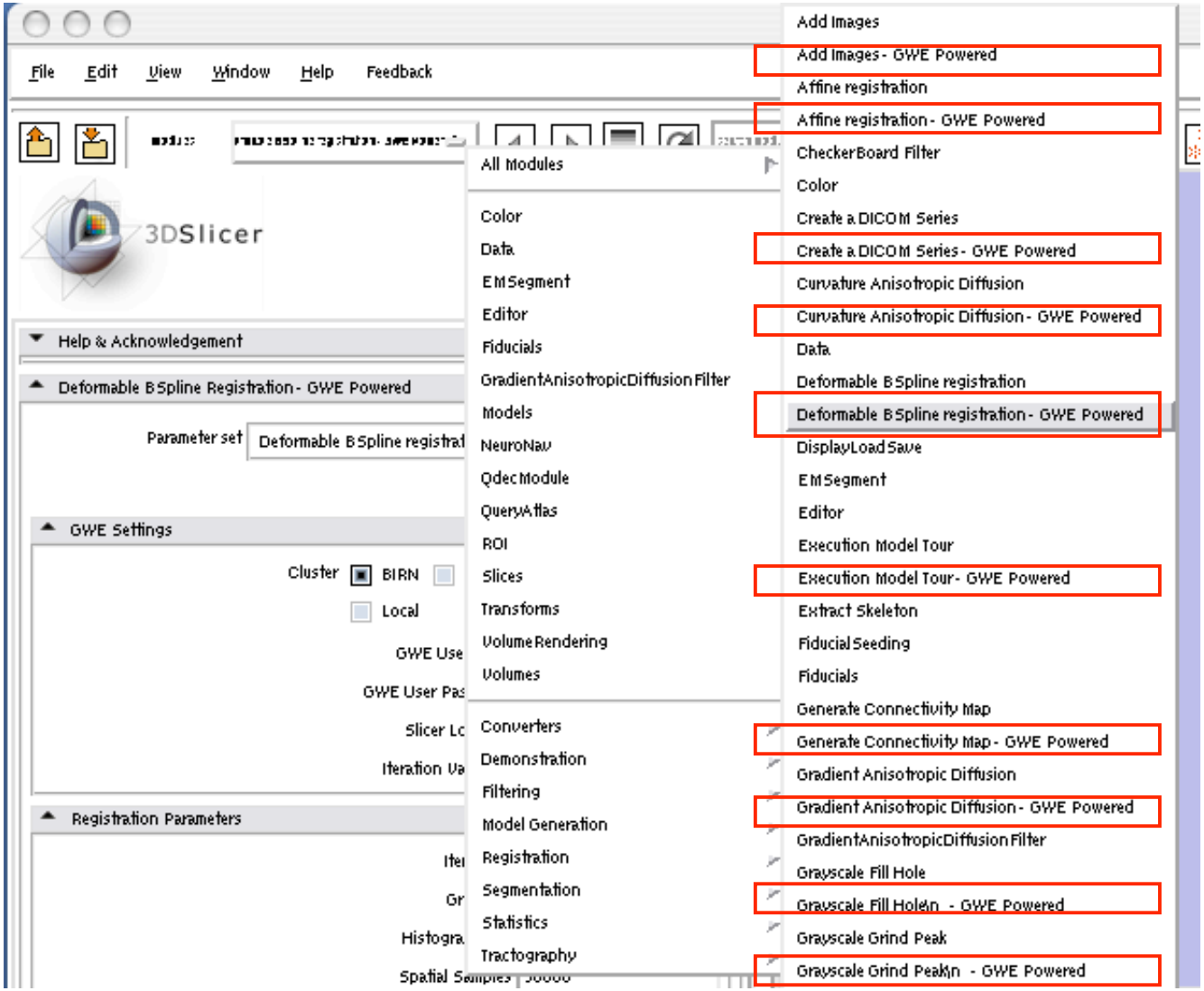
Tool Integration - GSlicer3: Registered Modules



Slicer3

- Standalone CLMs.

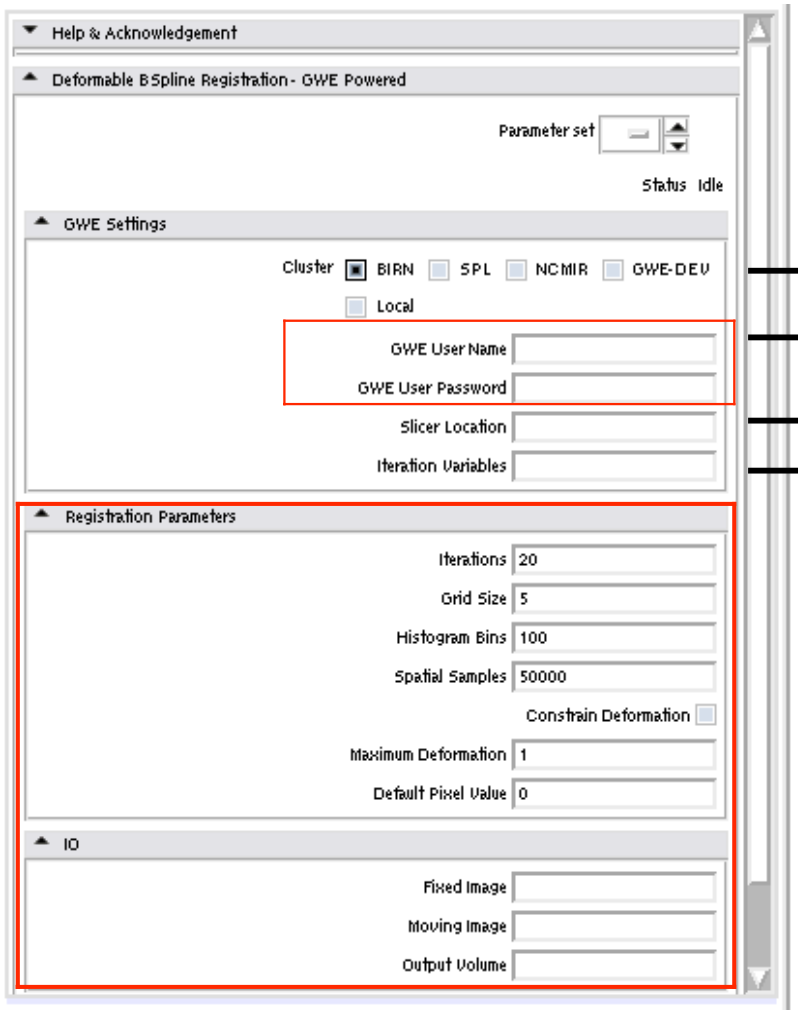
Tool Integration - GSlicer3: Registered Modules



GSlicer3:

- Standalone CLMs.
- 1 autogenerated GWE CLM Proxy for each standalone CLM discovered (which complies with the Standard Execution Model).

Tool Integration - GSlicer3: CLM Proxy Parameters



New section. Captures GWE parameters to learn how to execute invocations of this module on the grid.

Proxied CLM specific arguments tweaked to accept P2EL semantics

Clusters described in ${SLICER_HOME}/gwe/conf/gwe-grid.xml

GWE level authentication

Location of Slicer in the grid (soon to be deprecated)

P2EL iteration variables

More Information

- **Project site with a great wealth of information including detailed guides and GWE's source code:**

  http://www.gridwizardenterprise.org/

- **Users mailing list to receive project news and announcements:**

  gwe-users@nbirn.net

- **Project community forum:**

  http://groups.google.com/group/gwe-forum?hl=en

- **Project team email address (questions, requests and/or feedback):**

  gwe-support@nbirn.net

Thanks!