Porting application to Grid

Viet D. Tran

Institute of Informatics, Slovakia

Content

1. **Preparation of your application**
2. **Joining Grid**
3. **First job**
4. **Full computation**
5. **Advanced Grid tools**
   - Portal
   - Data management
   - Workflow

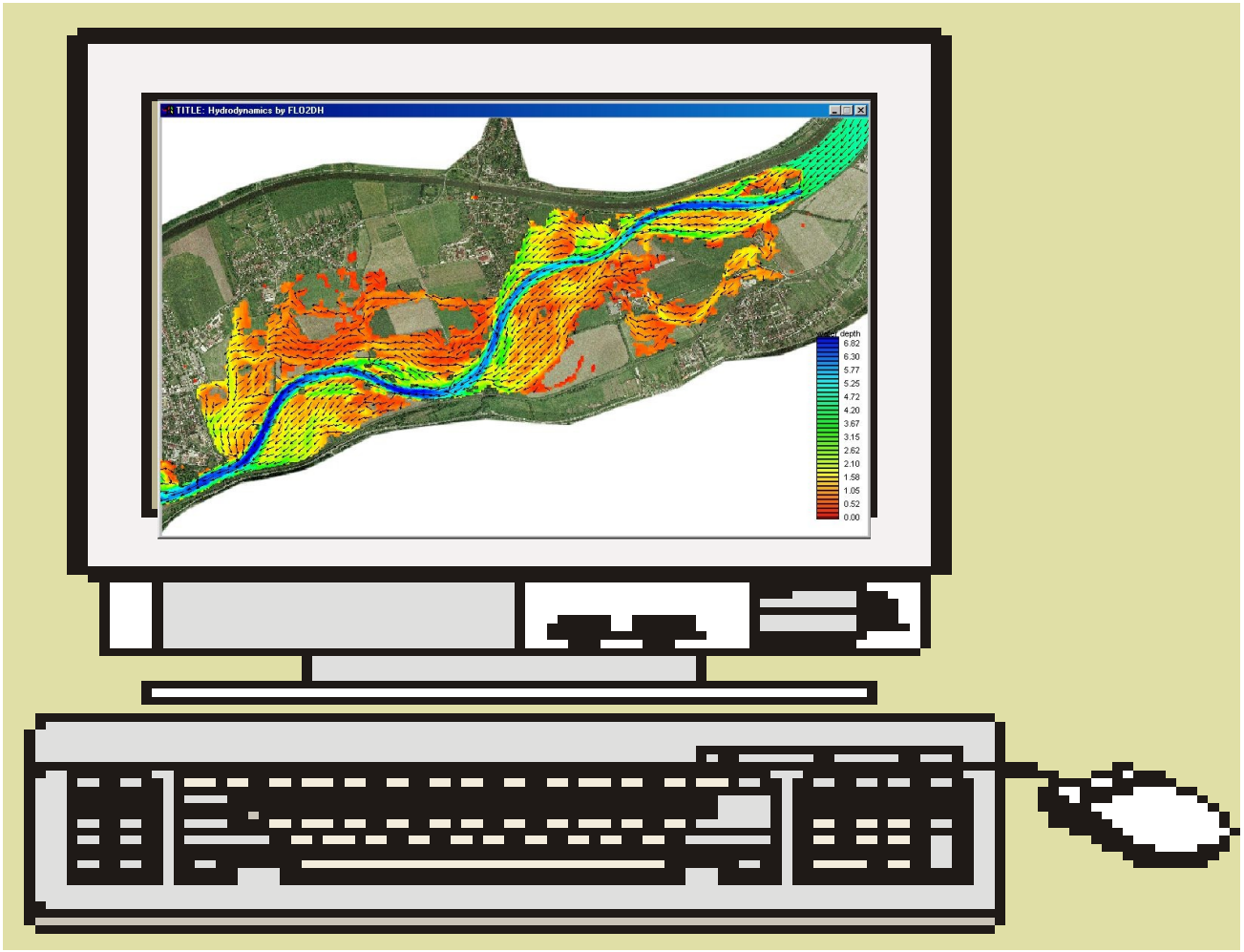
Content

- Preparation of your application
- Joining Grid
- First job
- Full computation
- Advanced Grid tools
  - Portal
  - Data management
  - Workflow

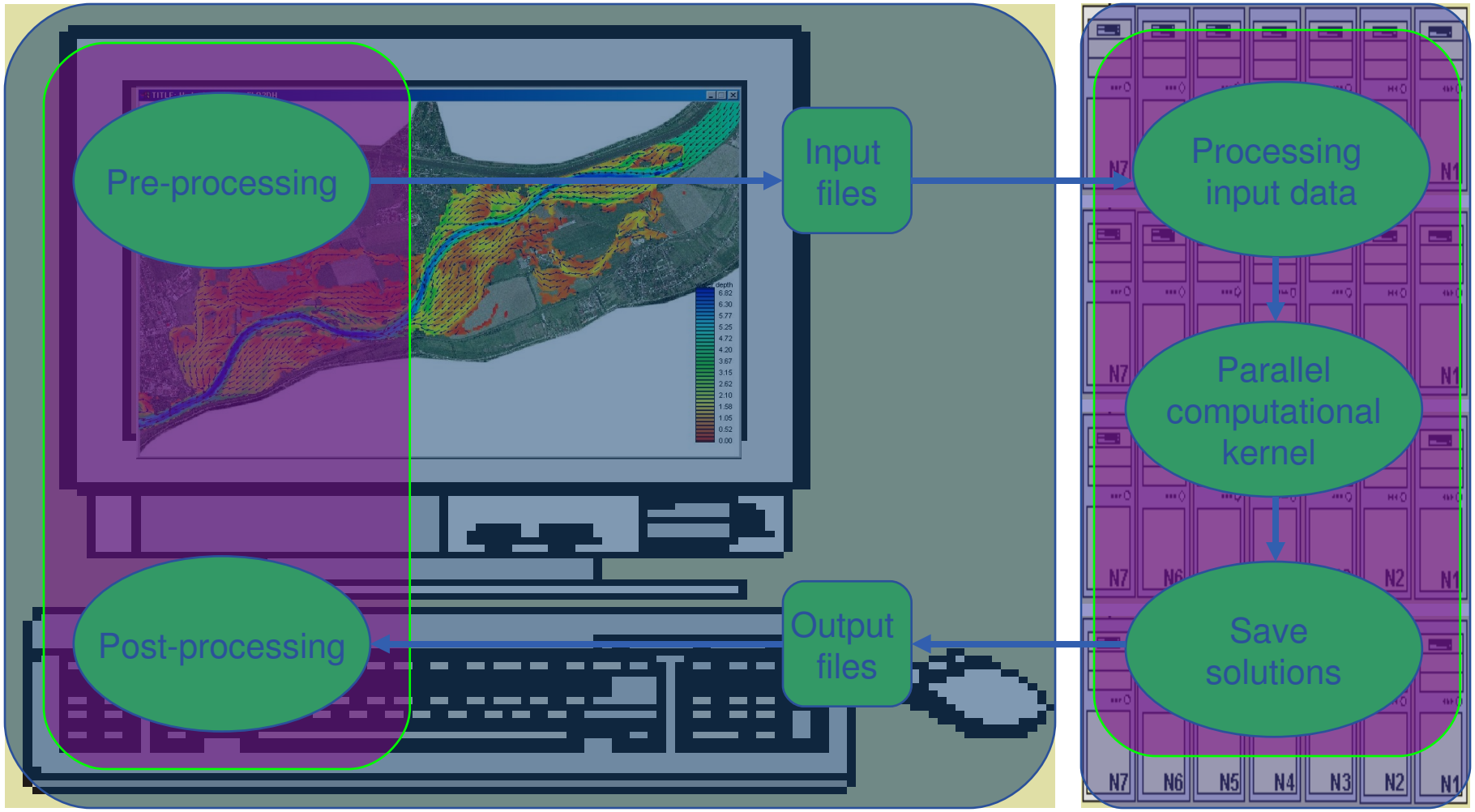
Preparation of application

- **Application should run in batch mode:**
  - Separate computations from graphical interfaces
  - Input and output data from/to files
- **Porting application to Linux**
  - Compile, link and test your application in Linux
  - For MPI parallel program, testing on Linux cluster
  - Non-standard libraries (e.g. LAPACK) should be linked statically
  - TEST: try to copy your application to a new, clean Linux machine and test it there

Example


TITLE: Hydrodynamics by FLO2DH
depth
6.82
6.30
5.77
5.25
4.72
4.20
3.67
3.15
2.62
2.10
1.58
1.05
0.52
0.00

Example


Pre-processing
Input files
Processing input data
Parallel computational kernel
Save solutions
Output files
Post-processing
N7
N6
N5
N4
N3
N2
N1

Content

1. **Preparation of your application**

   Results: Applications running in Linux in batch mode

- **Joining Grid**
- **First job**
- **Full computation**
- **Advanced Grid tools**
  - Portal
  - Data management
  - Workflow

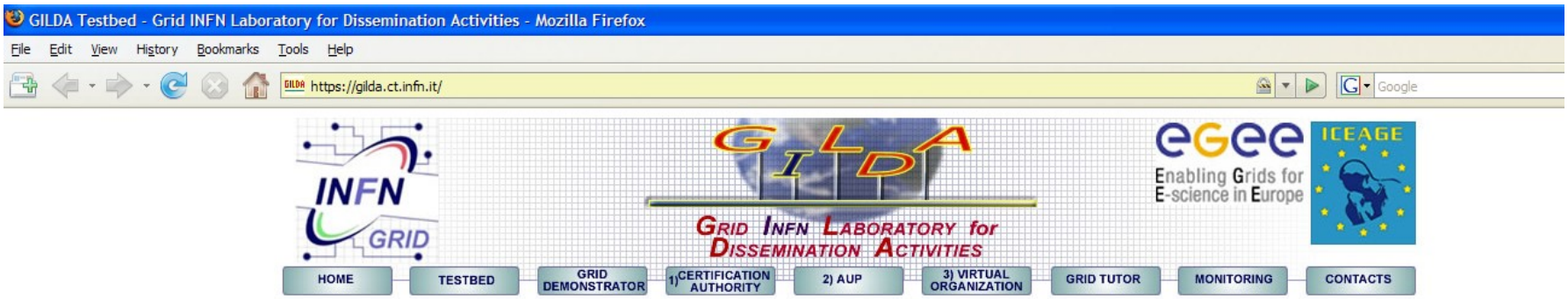
Available Grid infrastructures

- **Choosing the infrastructure for porting**
  - GILDA testing infrastructure
    - Simple registration, available for every ones
    - Suitable for testing, learning and demonstration
    - Should not be used for stress testing (too long or too many jobs), especially during some courses or demonstrations
  - EGEE production infrastructure
    - Suitable for running application in production

Joining GILDA

- **GILDA main web page: https://gilda.ct.infn.it/ with many tutorials, documentations and training materials**
- **Register and get a GILDA certificate https://gilda.ct.infn.it/CA/**
- **Get an account on User Interface (UI) machines**
  - GILDA automatically create an account on glite*.gilda.infn.it during registration but the UI machines have firewalls for SSH
  - Contact with us viet.ui@savba.sk to have an account on our UI dgt03.ui.sav.sk.
- **Start to learn grid commands and test them**

GILDA main page



- News
- List of Tutorials
- Request a Tutorial
- Posters
- Video Tutorials
- User Interface PnP
- Virtual Services
- Instructions for Users
- Instructions for Sites
- Training Material (wiki)
- GILDA Forge
- Support System
- Useful Links

- Sponsors
- Usage Statistics
- Old Usage Statistics

New GILDA usage rule

GILDA (G rid I nfn L aboratory for D issemination A ctivities)

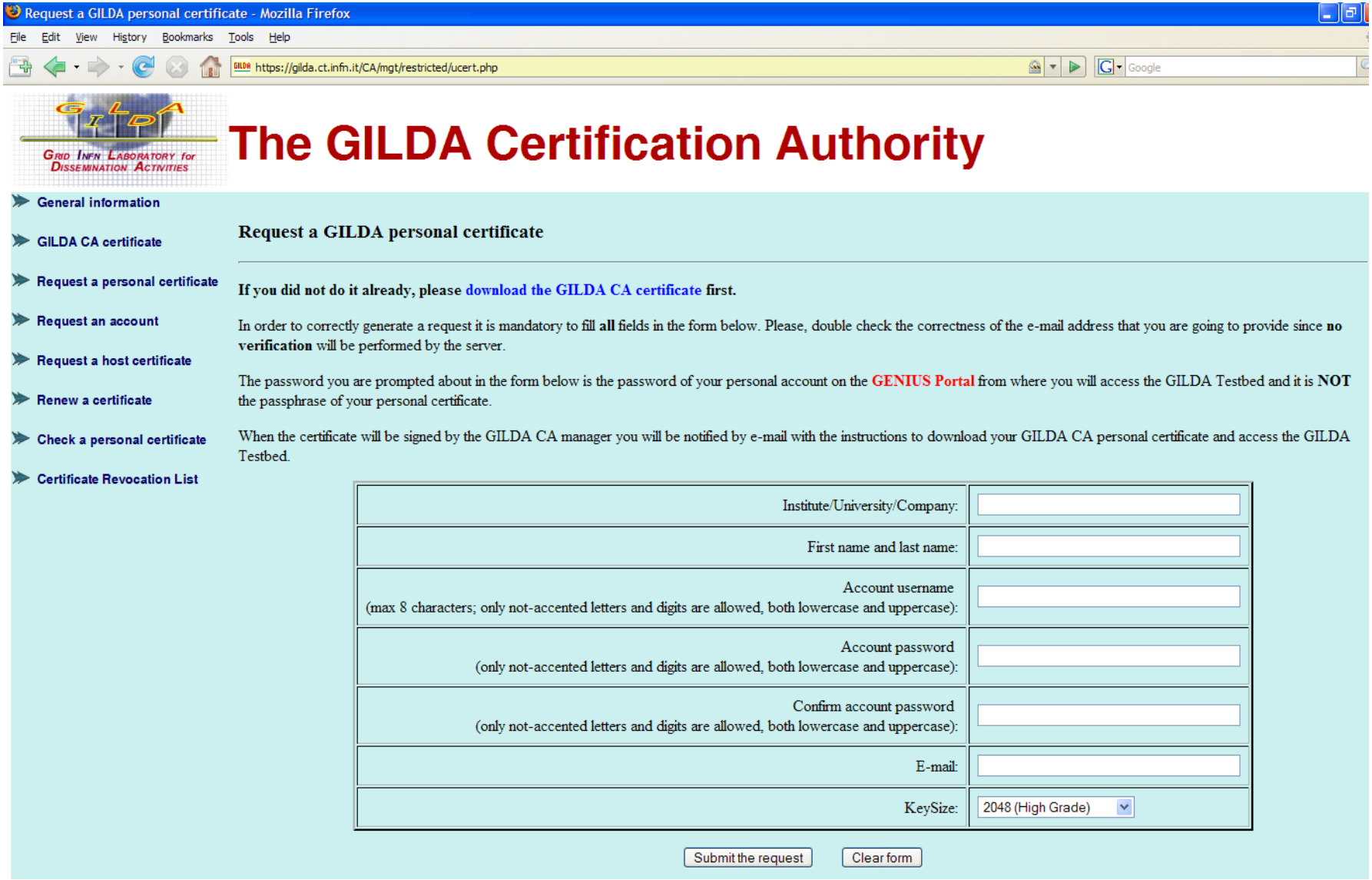
is a virtual laboratory to demonstrate/disseminate the strong capabilities of grid computing.

GILDA consists of the following elements:

- the GILDA Testbed: a series of sites and services (Resource Broker, Information Index, Data Managers, Monitoring tool, Computing Elements, and Storage Elements) spread all over Italy and the rest of the world on which the latest version of the INFN Grid middle-ware (fully compatible with gLite) is installed;
- the Grid Demonstrator: a customized version of the full GENIUS web portal, jointly developed by INFN and NICE, from where **everybody** can submit a pre-defined set of applications to the GILDA Testbed;
- the GILDA Certification Authority: a fully functional Certification Authority which issues 14-days X.509 certificates to everybody wanting to experience grid computing on the GILDA Testbed;
- the GILDA Virtual Organization: a Virtual Organization gathering all people wanting to experience grid computing on the GILDA Testbed; the GILDA Virtual Organization is based on the Virtual Organization Membership Service (VOMS) developed by INFN;
- the Grid Tutor: based on a full version of the GENIUS web portal, to be used only during grid tutorials;
- the GILDA P-GRADE Portal: a Grid workflow developer and executor environment based on the latest release of the P-GRADE Grid Portal provided by MTA SZTAKI;
- the monitoring systems: several tools monitoring the status of the GILDA Testbed and the availability of its grid services;
- the GILDA mailing lists: gilda-sitemanagers@ct.infn.it, for system administrators (both lists are restricted to members only). A web form to subscribe/unsubscribe to/from the lists is available here.

GILDA is an activity of the Italian Istituto Nazionale di Fisica Nucleare (INFN) carried on in the context of the following Projetcs:

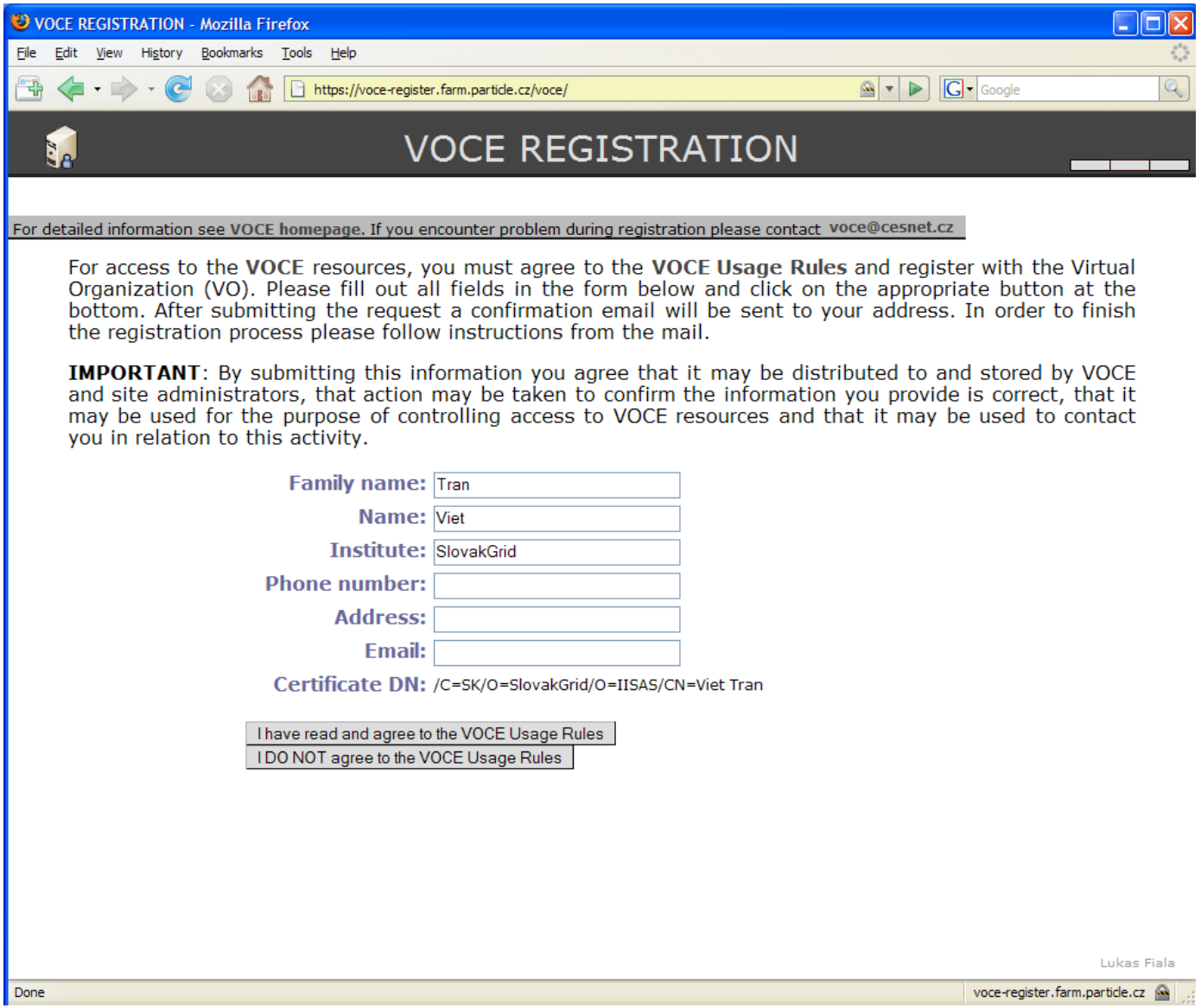
GILDA registration



Joining EGEE infrastructure

- **Register and get an EGEE certificate**
  - Contact with Miroslav Dobrucky (dobrucky.ui@savba.sk)
- **Join to a Virtual Organization (VO)**
  - If your application belongs to areas of some existing VO (e.g. biomedicine, chemistry, earth science, …) you can contact with the VO manager and ask him to add you to his VO
  - Send us short description of your application, we can give contact of the corresponding VO manager
  - If your application does not belong to any VO, you can still join the generic VOCE VO (Central European VO) https://voce-register.farm.particle.cz/voce

VOCE main page



Content

1. **Preparation of your application**

   Results: Applications running in Linux in batch mode

2. **Joining Grid**

   Results: Receive Grid certificate and join VO

- **First job**
- **Full computation**
- **Advanced Grid tools**
  - Portal
  - Data management
  - Workflow

First job

- **Copy your executable program (**myprogram.bin**) and input data (**myinput.dat**) to User Interface (UI) machine (**dgt03.ui.sav.sk**)**
- **Log in to the UI (**ssh dgt03.ui.sav.sk**)**
- **Create a simple JDL file** (vi myfirstjob.jdl)

```
Executable= “myprogram.bin”;
Arguments= “myinput.dat”;
InputSandbox = {“myprogram.bin",“myinput.dat"};
StdOutput = "std.out";
StdError = "std.err";
OutputSandbox = {“myoutput.dat”, "std.out","std.err“};
```

- **Create proxy certificate:** voms-proxy-init
- **And submit the program:** glite-wms-job-submit –a myfirstjob.jdl
- **That is applicable only for smaller data and binary (< 2MB)**

Larger job

- **If you have large data and binary, first upload them to storage elements (**storage.ui.sav.sk**)**
- **Then create a simple script to download the binary, input data and upload the output data** (myscript.sh)

```
globus-url-copy gsiftp://storage.ui.sav.sk/data/myprogram.bin \
                                    file://$PWD/myprogram.bin
chmod a+x myprogram.bin
globus-url-copy gsiftp://storage.ui.sav.sk/data/myinput,dat
   file://$PWD/myinput.dat
./myprogram.bin myinput.dat
globus-url-copy file://$PWD/myoutput.dat \
   gsiftp://storage.ui.sav.skdata/myoutput.dat
```

- **Create a JDL to submit a job with the script**

```
Executable= “myscript.sh”;
InputSandbox = {“myscript.sh”};
```

Content

1. **Preparation of your application**

   Results: Applications running in Linux in batch mode

2. **Joining Grid**

   Results: Receive Grid certificate and join VO

3. **First job**

   Results: Application is running in Grid with simple testing data

- **Full computation**
- **Advanced Grid tools**
  - Portal
  - Data management
  - Workflow

Full computation

- **For parametric study, first create the JDL (**param.jdl**):**

```
JobType = "Parametric";
Executable = "myprogram.bin";
Arguments = “myinput_PARAM_.dat";
StdOutput = “std_PARAM_.out";
StdError = “std_PARAM_.err";
Parameters = 100;
ParameterStart = 1;
ParameterStep = 1;
InputSandbox = {"myprogram.bin", “myinput_PARAM_.dat";
OutputSandbox = {“myoutput_PARAM.dat”, std_PARAM_.out",
   “std_PARAM_.err"};
```

- **Then submit the JDL to Grid:** glite-wms-job-submit param.jdl

Full computation

- **For MPI jobs, just add to the JDL**

```
JobType = "MPICH";
NodeNumber = 16;
Executable= “myprogram.bin”;
Arguments= “myinput.dat”;
InputSandbox = {“myprogram.bin”,“myinput.dat”};
StdOutput = "std.out";
StdError = "std.err";
OutputSandbox = {“myoutput.dat”, “std.out”,“std.err”};
```

Content

1. **Preparation of your application**

   Results: Applications running in Linux in batch mode

2. **Joining Grid**

   Results: Receive Grid certificate and join VO

3. **First job**

   Results: Application is running in Grid with simple testing data

4. **Full computation**

   Results: Application is running in Grid with full data

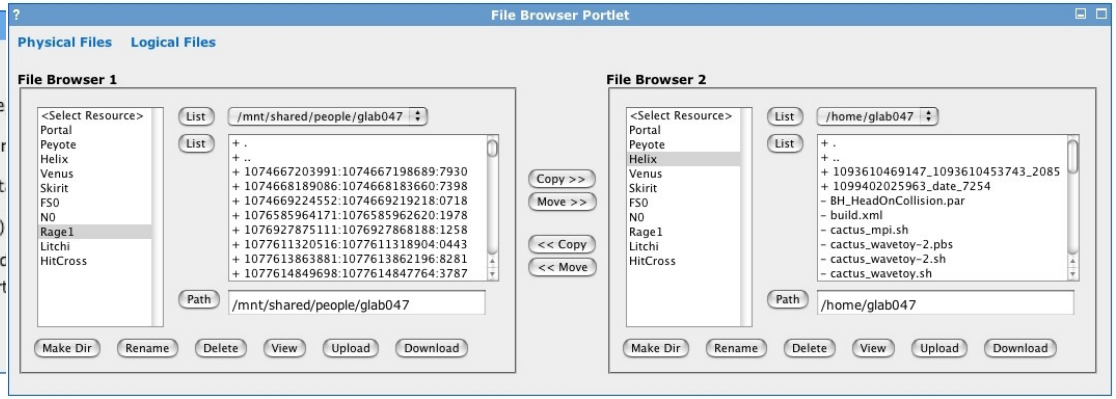
5. **Advanced Grid tools**

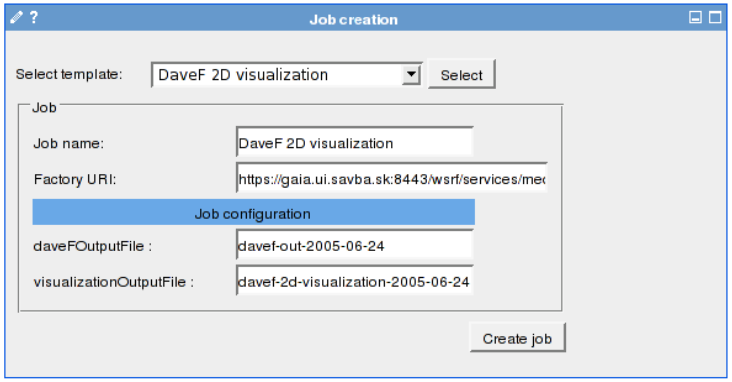
   - Portal
   - Data management
   - Workflow

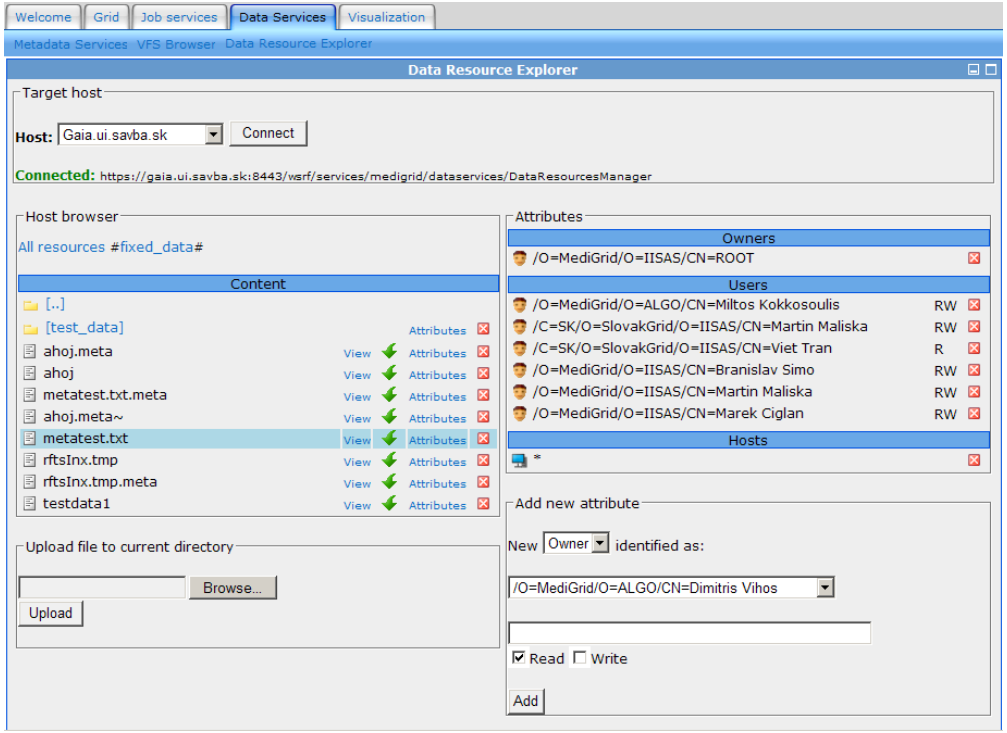
Using portal

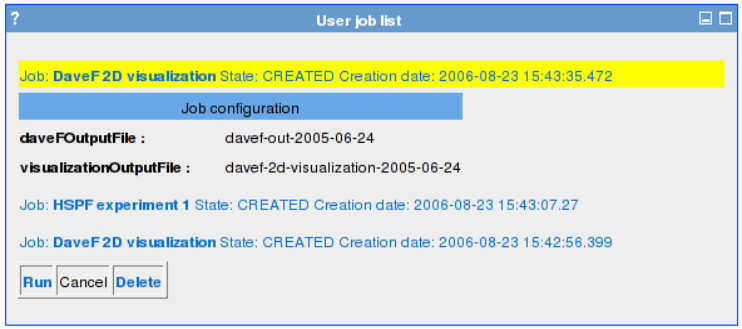
- **Command line interface (**glite-wms-job-* commands**) is powerful but not user-friendly**
- **Portal is useful because:**
  - Accessible from any where
  - Hides complexity: the actual users do not have to learn about gLite or similar things
  - More secure
  - Useful for demonstration, can be combined with visualization
- **Recommended Grid portal framework: Gridsphere**
  - http://www.gridsphere.org/
  - Supporting portlet (JSR168)
  - Easy to use and extend
- **Ready to use portlets:**
  - Certificate management
  - Job submission
  - Data browser

Examples of portlets





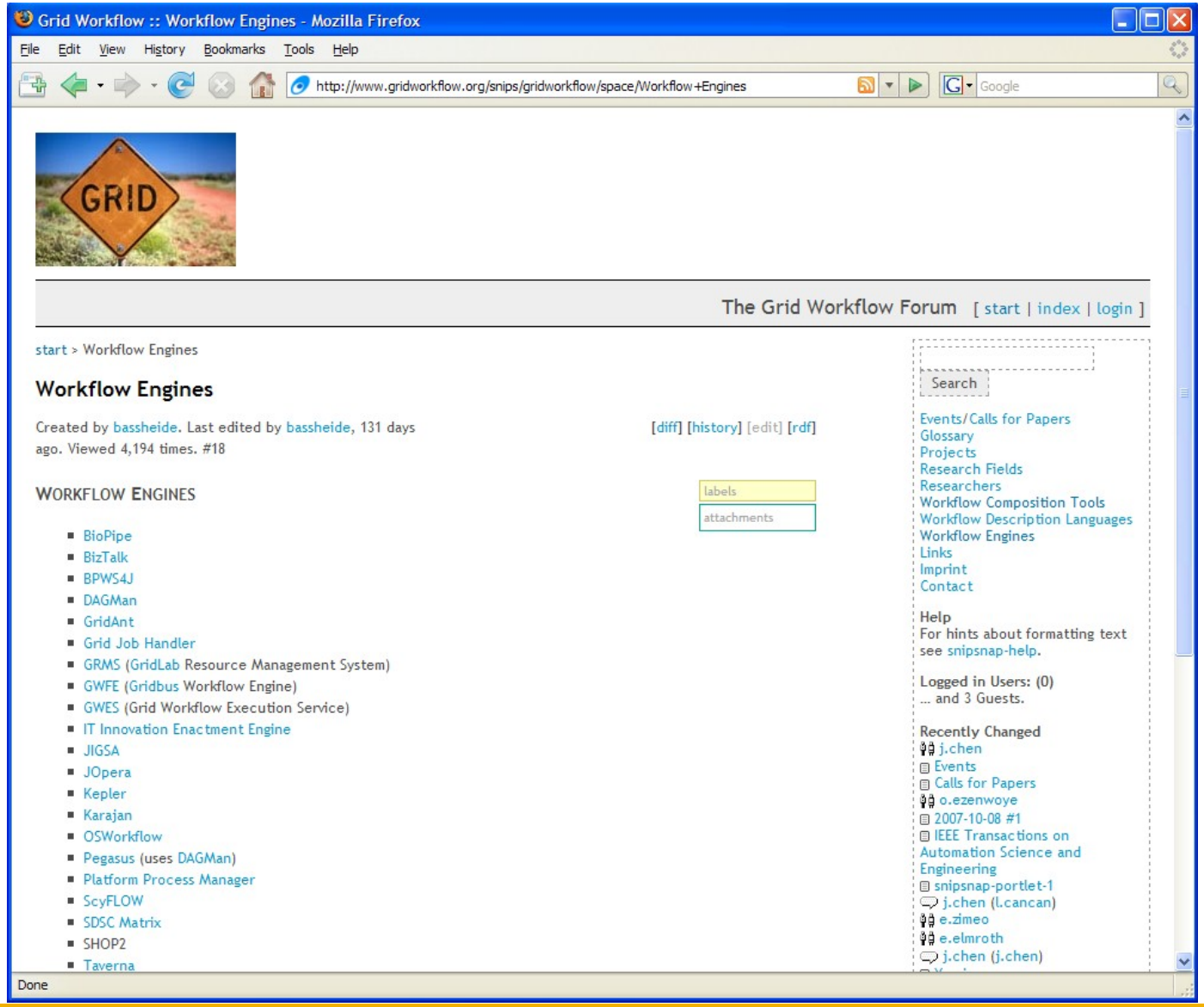




Data management

- **Big applications usually also have big data sets**
- **Some data management tools in Grid**
  - Replica management
    - Each piece of data may have multiple copies (files) to improve the data access
    - Can provide some optimization
  - Metadata catalogs
    - Describe the data in Grid
    - Searching, browsing
  - Access to databases (OGSA-DAI)
    - Access to databases (e.g. Oracle, MySQL) from your program running on the Grid

Workflow management



Content

1. **Preparation of your application**
   - Applications running in Linux in batch mode
2. **Joining Grid**
   - Receive Grid certificate and join VO
3. **First job**
   - Application is running in Grid with simple testing data
4. **Full computation**
   - Application is running in Grid with full data
5. **Advanced Grid tools**
   - User-friendly GUI for applications

Summary

- Porting application to Linux in batch mode
- Getting Grid certificates and joining some VO
- Testing application to Grid with simple testing data
- Full computation
- Portal and data/workflow management

Please contact to our team for additional helps