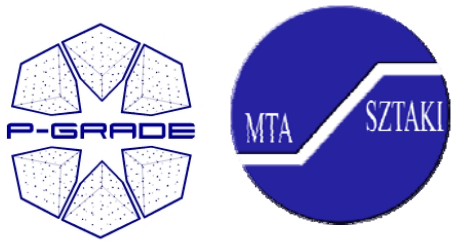


Grid application support by the P-GRADE Portal

Peter Kacsuk

kacsuk@sztaki.hu

MTA SZTAKI

Hungarian Academy of Sciences

www.lpds.sztaki.hu/pgportal

pgportal@lpds.sztaki.hu

Contents

- Motivation of creating P-GRADE portal
- P-GRADE Portal in a nutshell
- Application development with the Portal
- Application execution with the Portal

Context

Graphical interface
P-GRADE Portal and GEMLCA

Command line tools
Grid middleware

Current situation and trends in Grid computing

- Fast evolution of Grid systems and middleware:
  - **GT2, OGSA, GT3 (OGSI), GT4 (WSRF), LCG-2, gLite, …**
- Many production Grid systems are built with them
  - EGEE (LCG-2 → gLite), UK NGS (GT2), Open Science Grid (GT2 → GT4), NorduGrid (~GT2)
- Although **the same set of core services** are available everywhere, they **are implemented in different ways**
  - Data services
  - Computation services
  - Security services (single sign-on)
  - (Brokers)

E-scientists' concerns

- How to concentrate own **my own research** if the tool I would like to use is in continuous change?
- How can I learn and understand **the usage of the Grid**?
- How can I **develop Grid applications**?
- How can I **execute grid applications**?
- How to **tackle performance issues**?
- How to **use several Grids at the same time**?
- How to **migrate my application** from one grid to another?
- How can I **collaborate with fellow researchers**?

The P-GRADE Grid Portal gives you the answers!

P-GRADE Portal in a nutshell

- **General purpose**, **workflow-oriented computational Grid portal**. Supports the development and execution of workflow-based Grid applications – **a tool for Grid orchestration**
- Based on **GridSphere-2**
  - Easy to expand with new portlets (e.g. application-specific portlets)
  - Easy to tailor to end-user needs
- **Grid services** supported by the portal:

Service	EGEE grids	Globus grids
Job execution	Computing Element	GRAM
File storage	Storage Element	GridFTP server
Certificate management	MyProxy	
Information system	BDII	MDS-2, MDS-4
Brokering	Workload Management System	GTbroker
Job monitoring	Mercury	
Workflow & job visualization	PROVE	

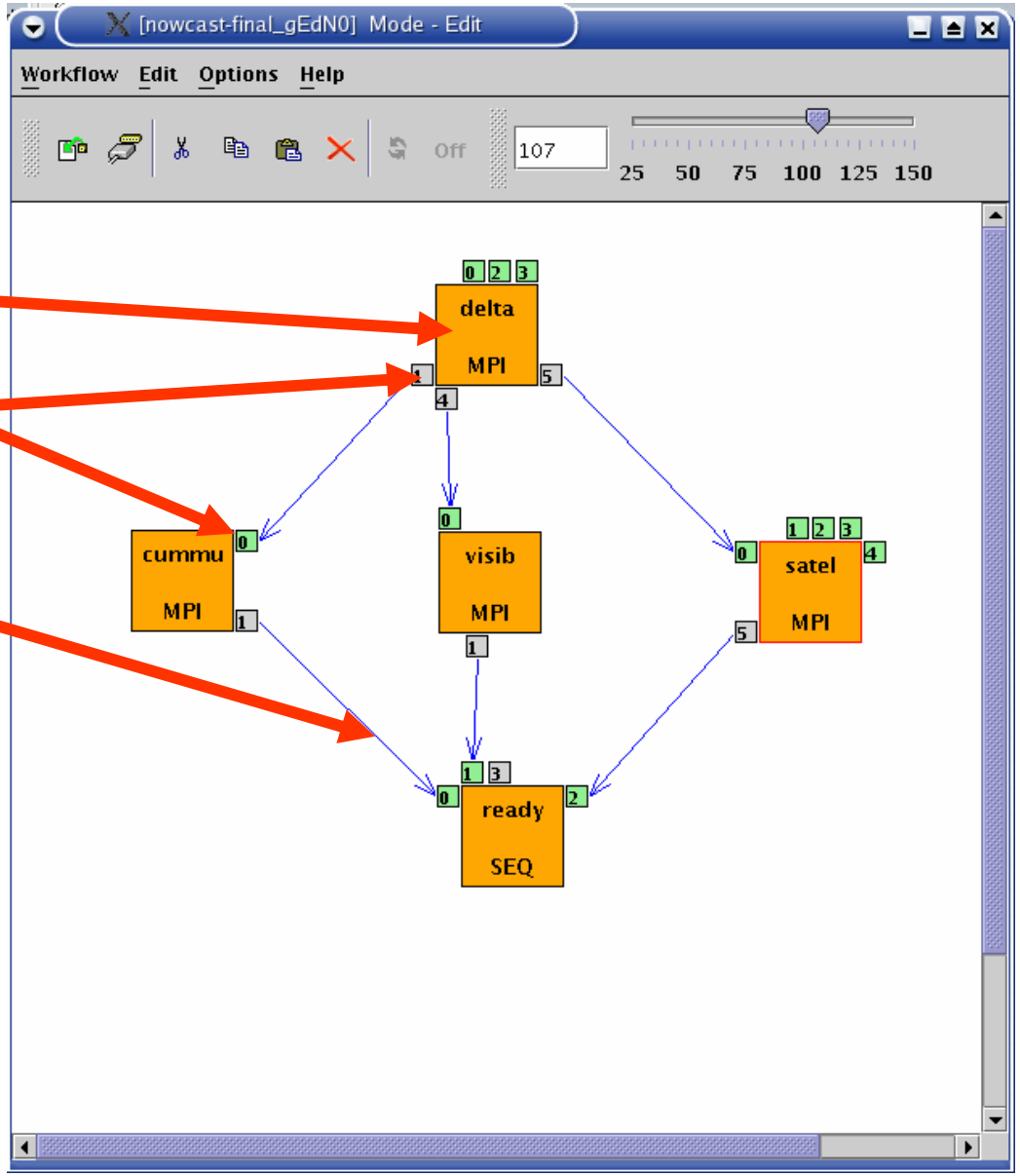
Solves Grid interoperability problem at the workflow level

Related projects

- The development and education of P-GRADE Portal is supported by several projects:
  - **SEE-GRID** *www.see-grid.eu*
    Development, application support
  - **Coregrid** *www.coregrid.net*
    Research, development
  - **EGEE** *www.eu-egee.org*
    LCG and gLite training, application development
  - **ICEAGE** *www.iceage-eu.org*
    Grid training and education

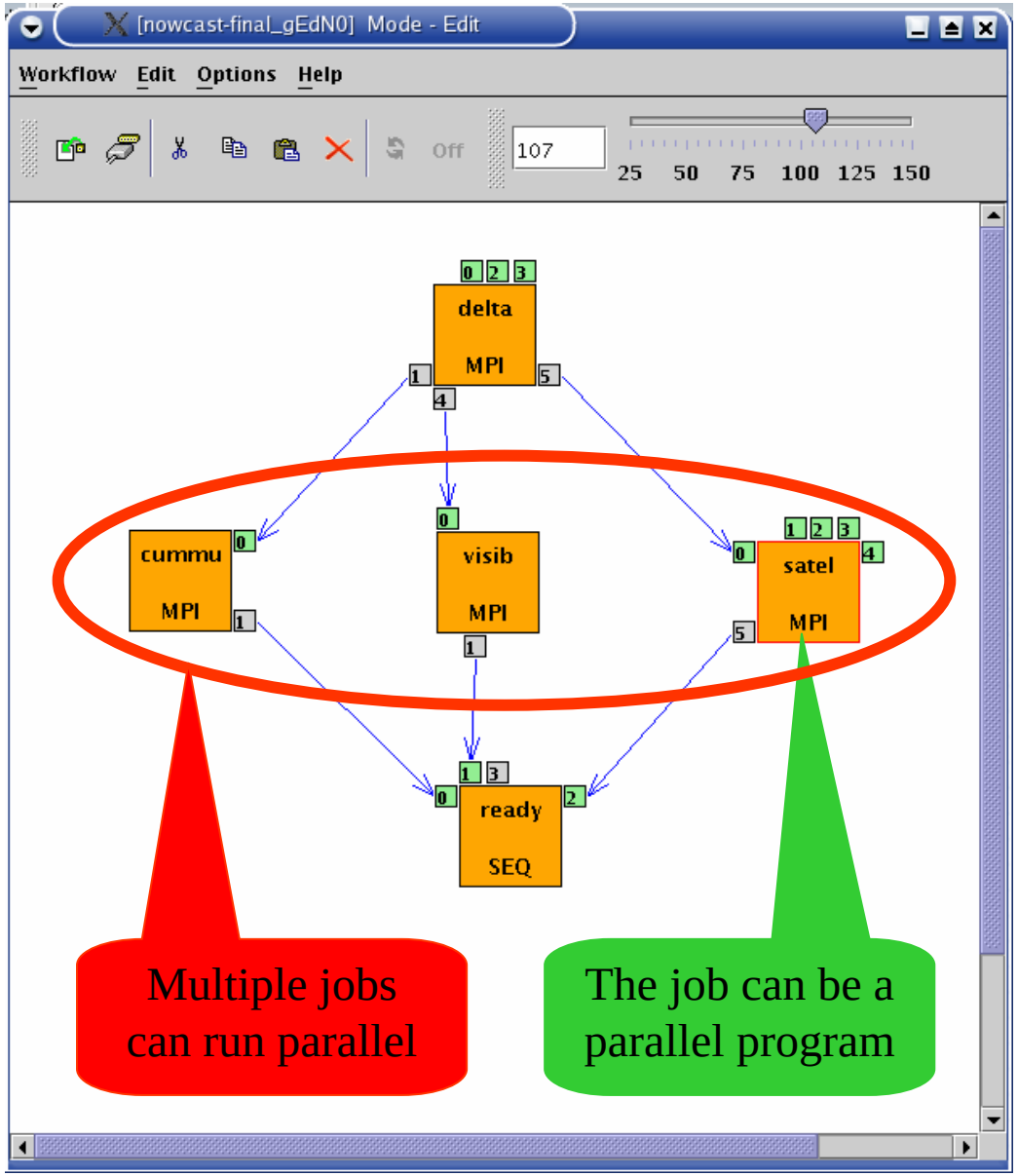
What is a P-GRADE Portal workflow?

- **a directed acyclic graph where**
  - Nodes represent jobs (batch programs to be executed on a computing element)
  - Ports represent input/output files the jobs expect/produce
  - Arcs represent file transfer operations
- **semantics of the workflow:**
  - A job can be executed if all of its input files are available

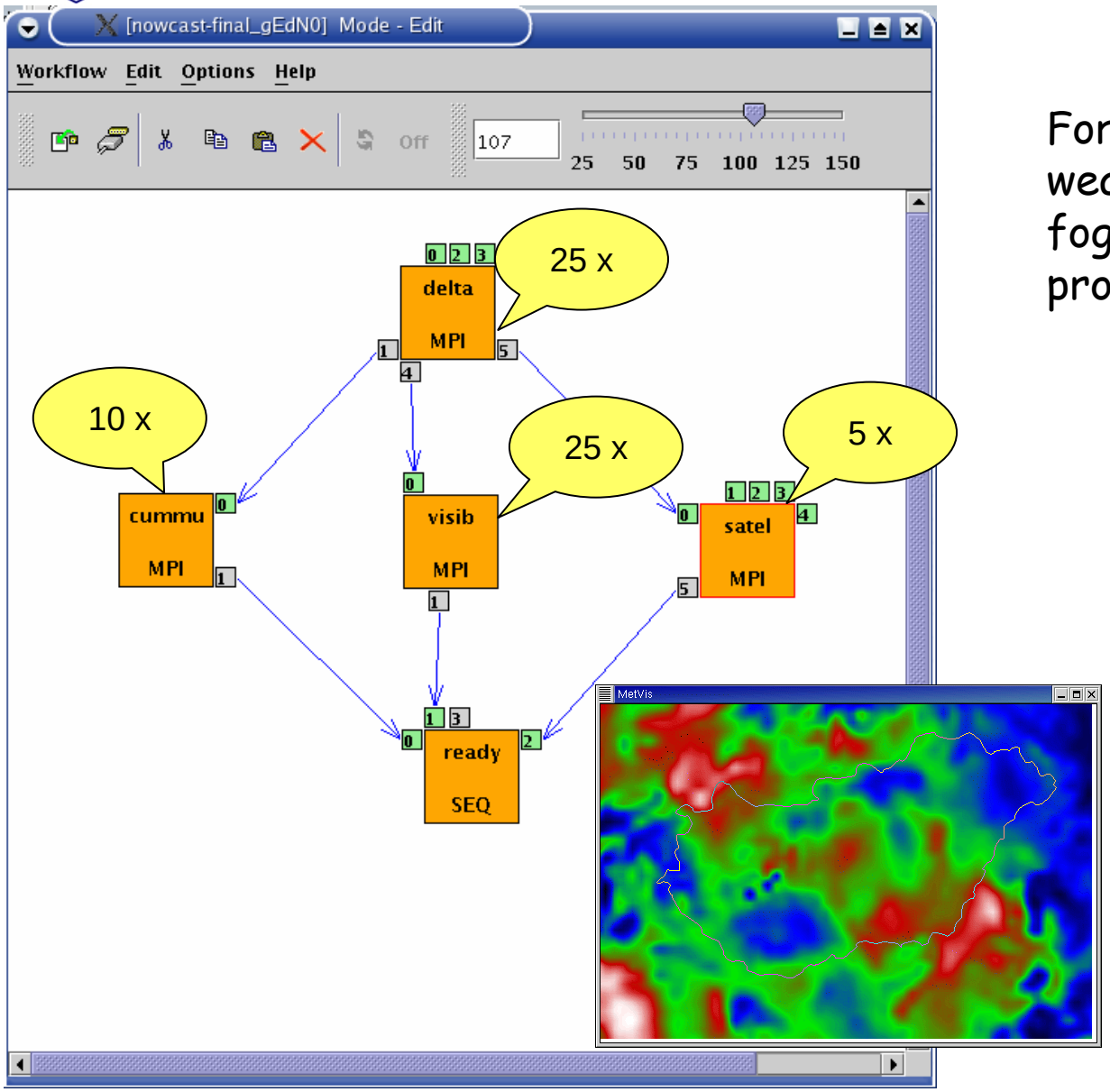


Two levels of parallelism by a workflow

- The workflow concept of the P-GRADE Portal enables the efficient parallelization of complex problems
- Semantics of the workflow enables two levels of parallelism:
  - Parallel execution inside a workflow node
  - Parallel execution among workflow nodes



Ultra-short range weather forecast (Hungarian Meteorology Service)

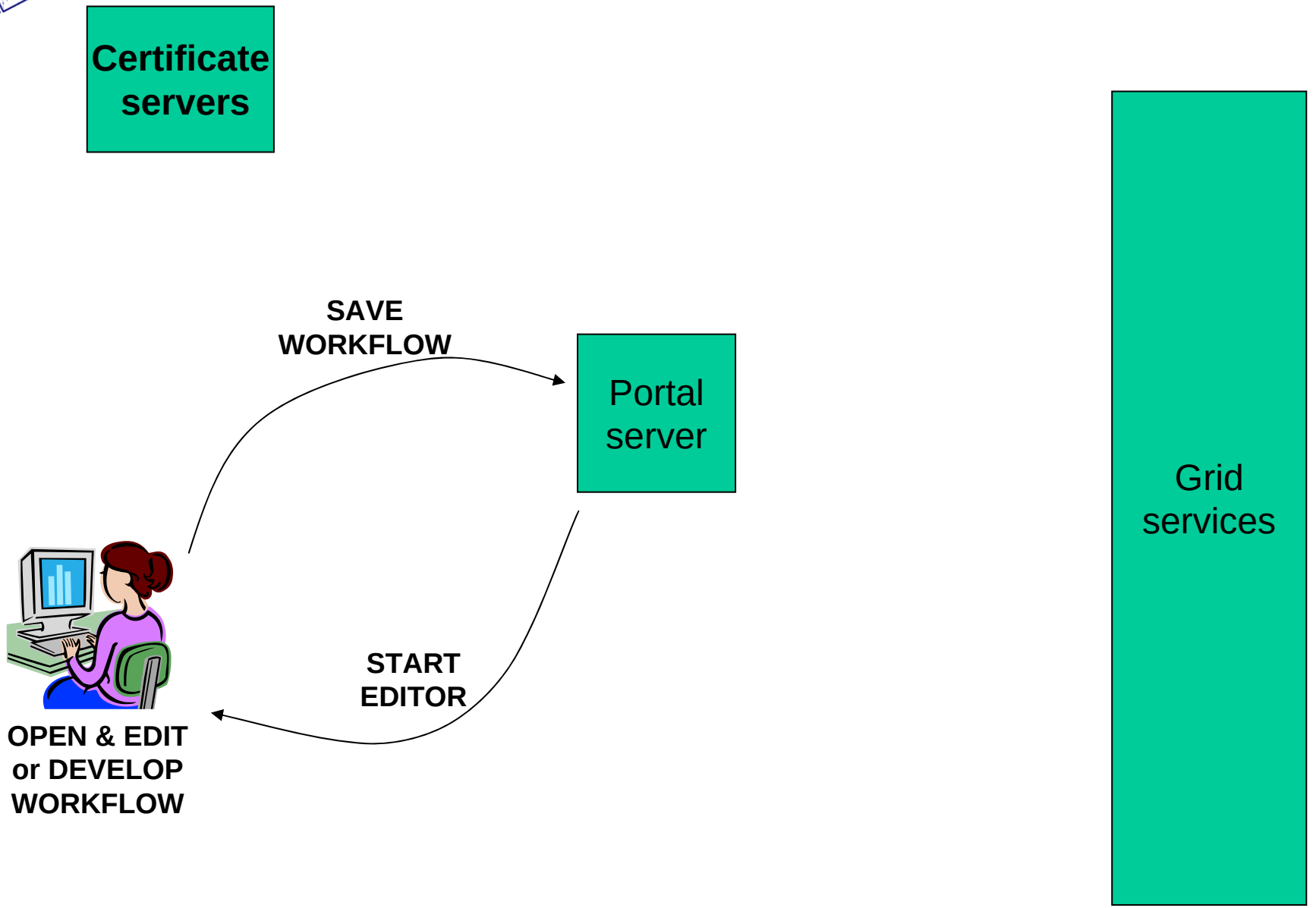


Forecasting dangerous weather situations (storms, fog, etc.), crucial task in the protection of life and property

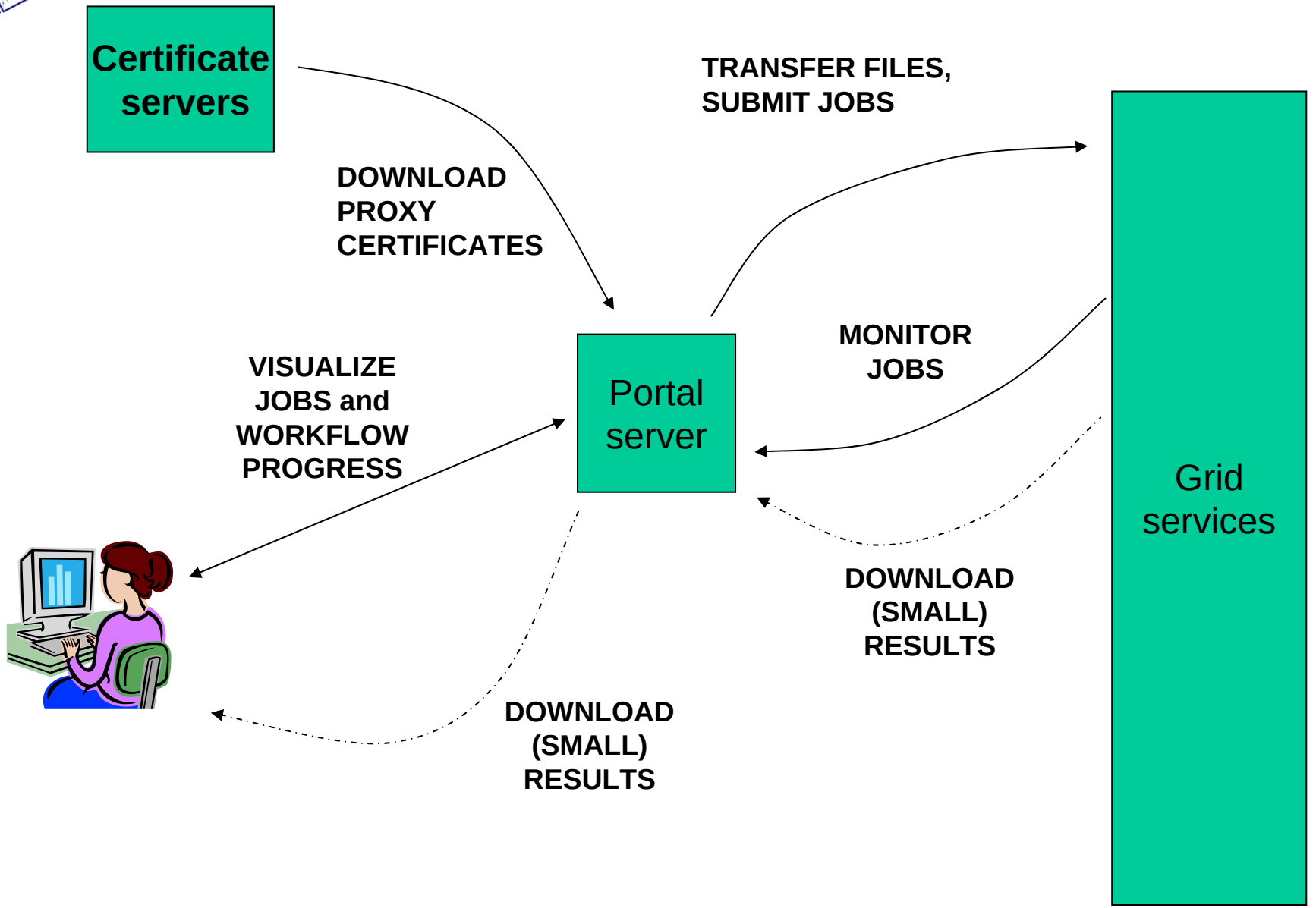
Processed information: surface level measurements, high-altitude measurements, radar, satellite, lightning, results of previous computed models

Requirements:
- Execution time < 10 min
- High resolution (1km)

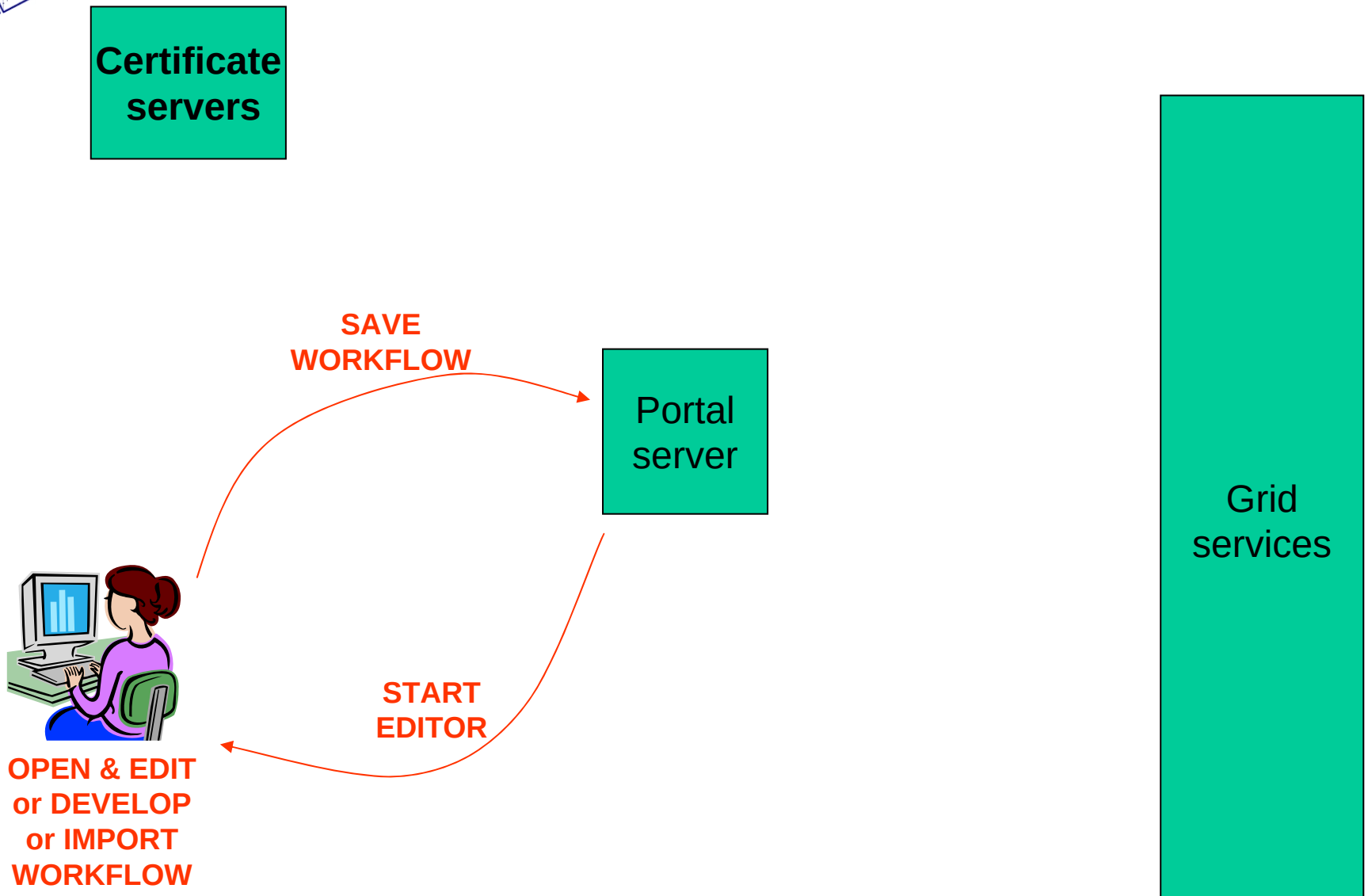
The typical user scenario
Part 1 - development phase



The typical user scenario
Part 2 - execution phase



The typical user scenario
Development phase:



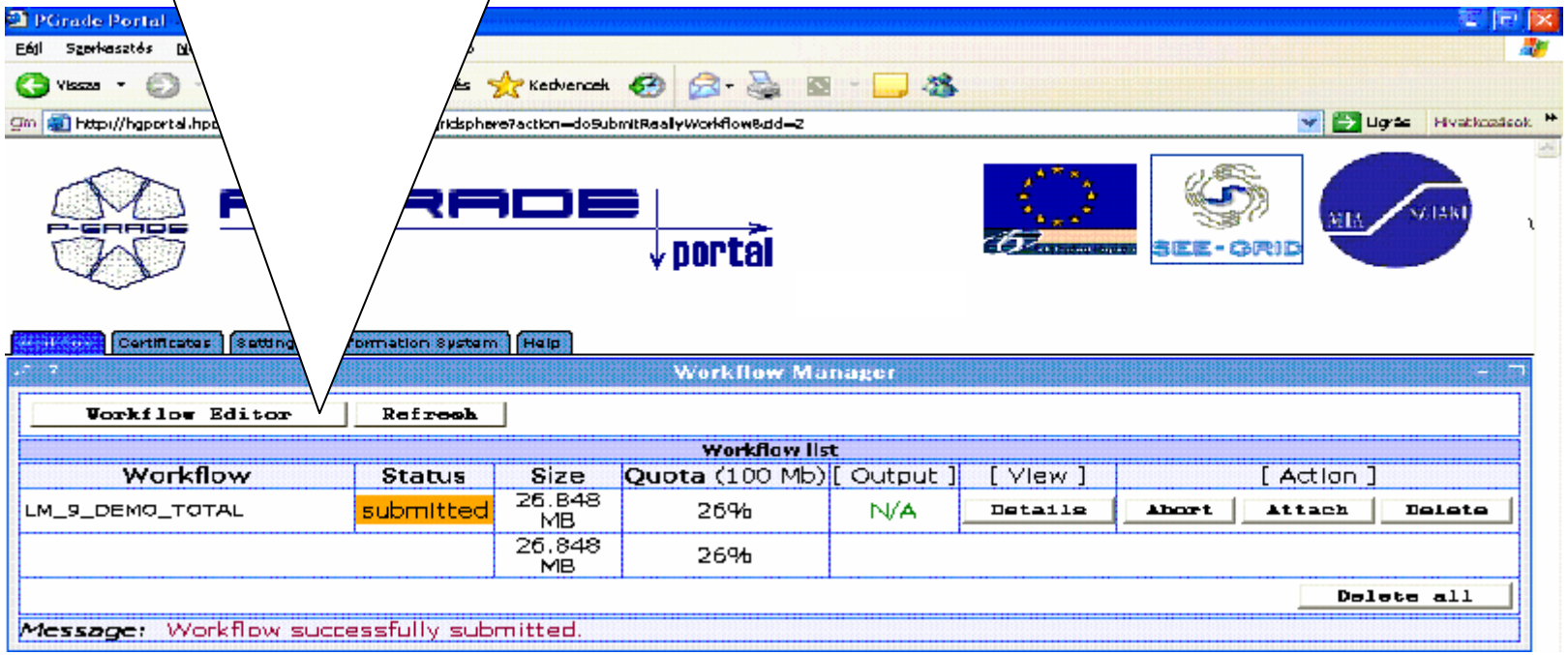
Workflow development

Opening the workflow editor

The editor is a Java Webstart application

↓

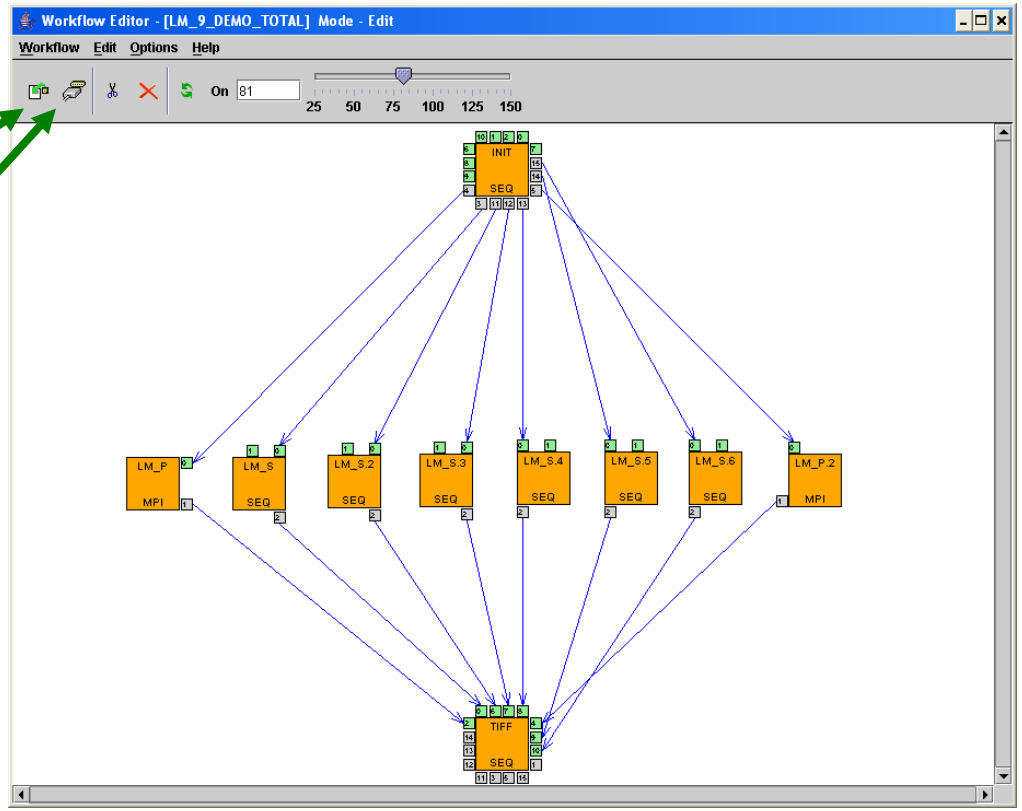
download and installation is only one click!



Workflow Editor

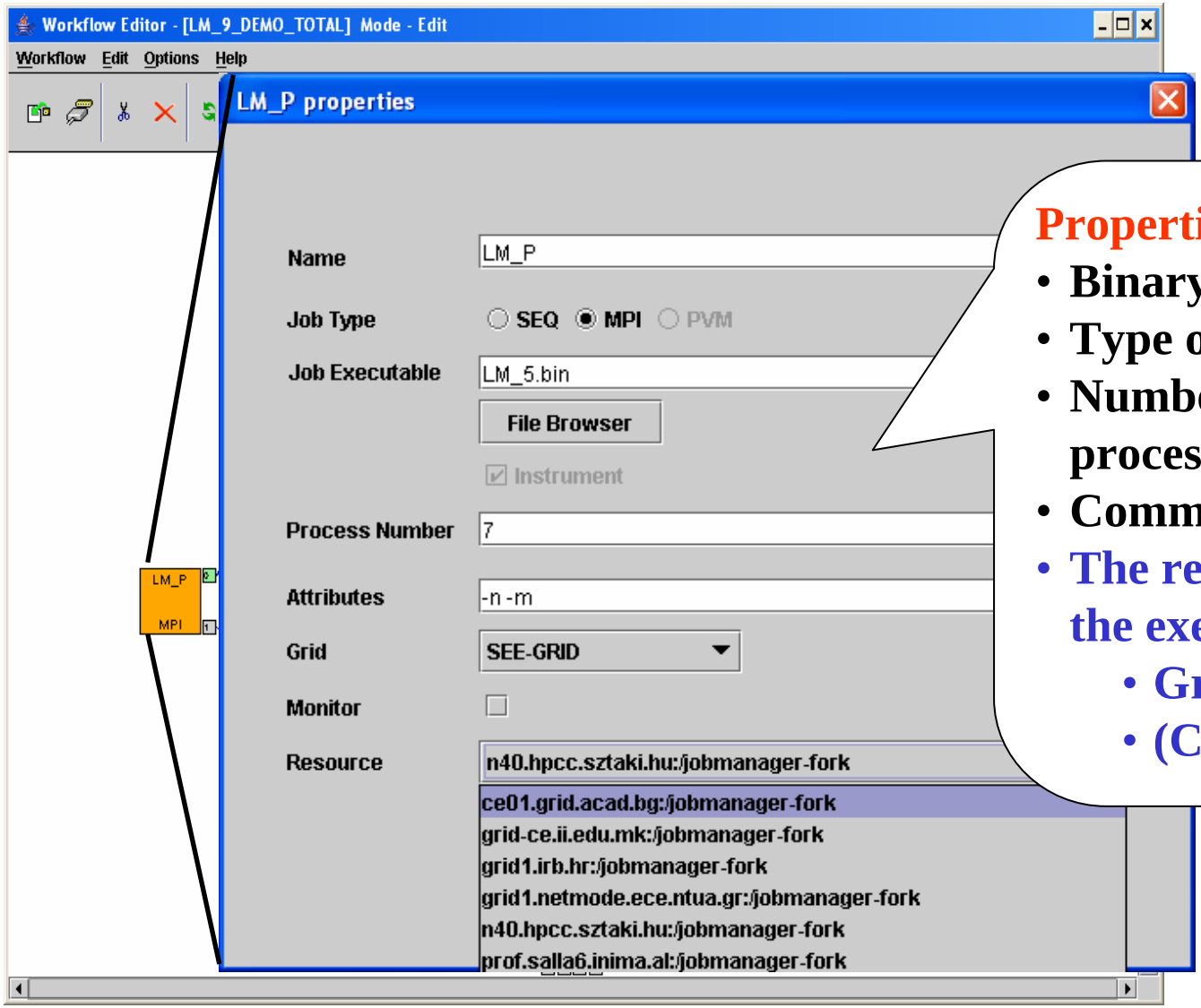
Defining the graph

- The aim is to define a DAG of batch jobs:
  1. **Drag & drop components:** jobs and ports
  2. **Define their properties**
  3. **Connect ports by channels** (no cycles, no loops, no conditions)



Workflow Editor

Properties of a job



Properties of a job:

- **Binary executable**
- **Type of executable**
- **Number of required processors**
- **Command line parameters**
- **The resource to be used for the execution:**
  - **Grid/VO**
  - **(Computing element)**

Direct resource selection: Which computing element to use?

I still don't know which resource to use!

The information system portlet queries BDII and GIIS servers

PGrade Portal - Microsoft Internet Explorer

Grid: SEE-GRID VO: seegrid

Sites

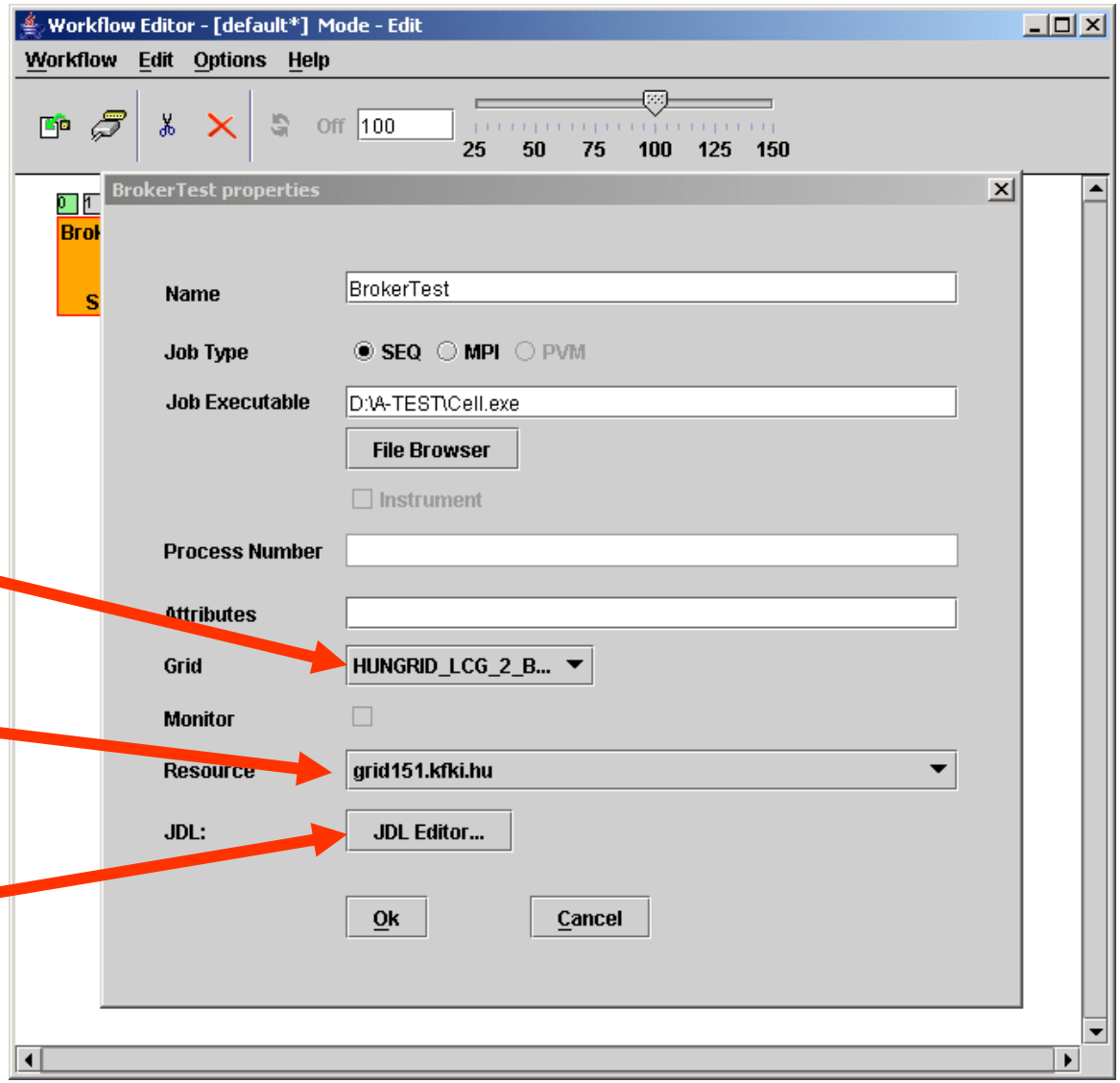
	Computing Element							Storage Element		
		CPU		Job				Space		
		Free	Usage	Running	Waiting	Load		Total	Available	Usage
	112	80	29%	7	0	0%		226.793 GB	216.34 GB	5%
AEGIS02-RCUB	20	20	0%	0	0	0%		398.466 GB	396.58 GB	0%
BG01-IPP	54	18	67%	4	0	0%		609.554 GB	473.543 GB	22%
BG02-IM	20	16	20%	1	0	0%		131.775 GB	79.957 GB	39%
BG03-IPP-N	3	3	0%	0	0	0%		566.608 GB	566.376 GB	0%
BG04-ACAD	48	32	33%	2	5	71%		554.647 GB	475.767 GB	14%
HR-01-RBI	60	12	80%	4	0	0%		78.317 GB	6.271 GB	92%
MK-01-UKIM_II	28	28	0%	0	0	0%		69.709 GB	69.075 GB	1%
RO-01-ICI	54	24	56%	5	36	88%		849.666 GB	828.387 GB	3%
ROGRID-NIPNE-01	24	24	0%	0	0	0%		862.807 GB	848.676 GB	2%
SZTAKI	4	4	0%	0	0	0%		4.566 GB	2.871 GB	37%
tubitaklcg2	35	28	20%	1	0	0%		1.335 TB	1.335 TB	0%

Automatic resource selection

1. Select a broker Grid/VO for the job (e.g. GILDA)
2. (Describe the ranks & requirements of the job in JDL)
3. The portal will use the broker to find the best resource for the job!

Workflow Editor

Defining broker jobs



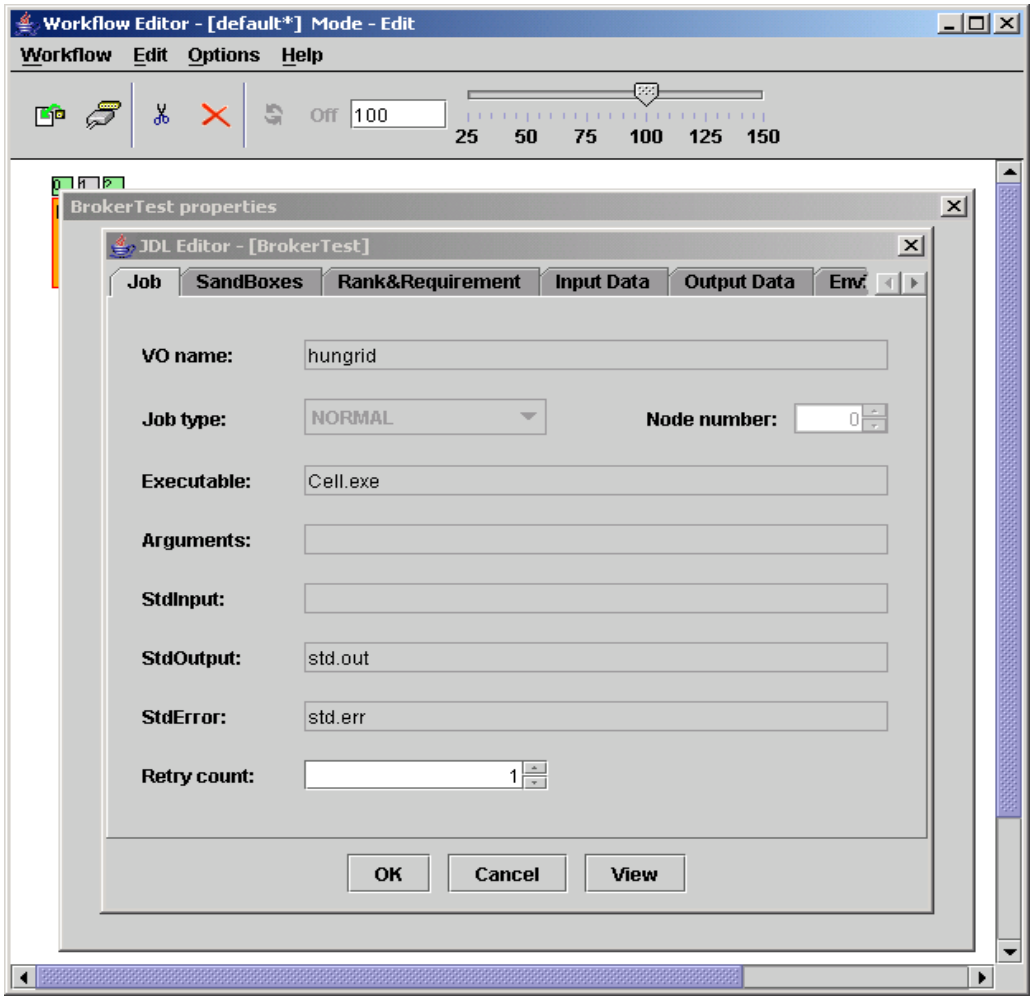
Select a Grid with broker! (*_BROKER)

Ignore the resource field!

If default JDL is not sufficient use the built-in JDL editor!

Workflow Editor

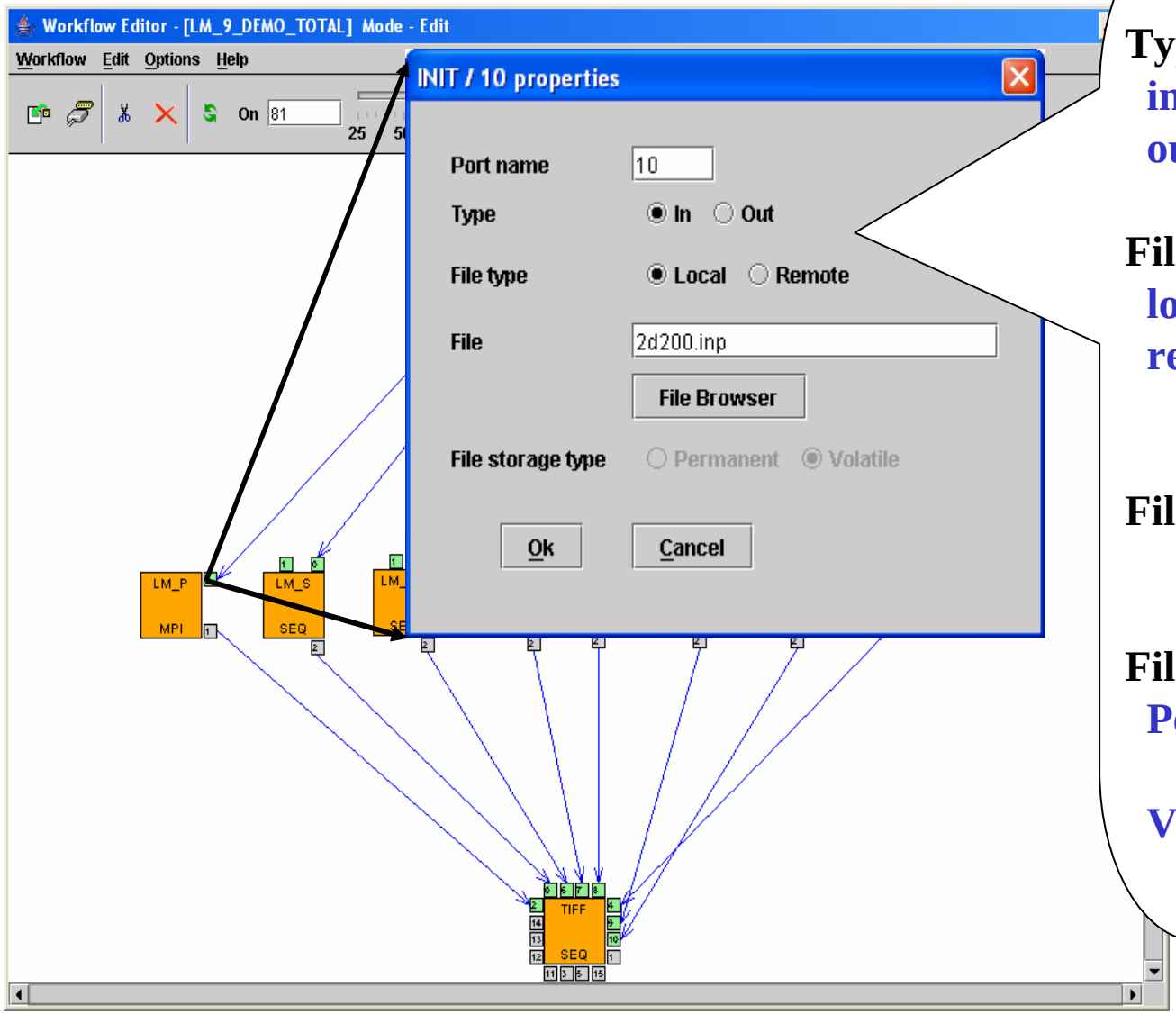
Built-in JDL editor



JDL → look at the LCG-2 Users' manual!

Workflow Editor

Defining ports



Type:
- **input:** *the job requires*
- **output:** *the job produces*

File type:
- **local:** *from/to my desktop*
- **remote:** *from/to a storage resource*

File:
location of the file

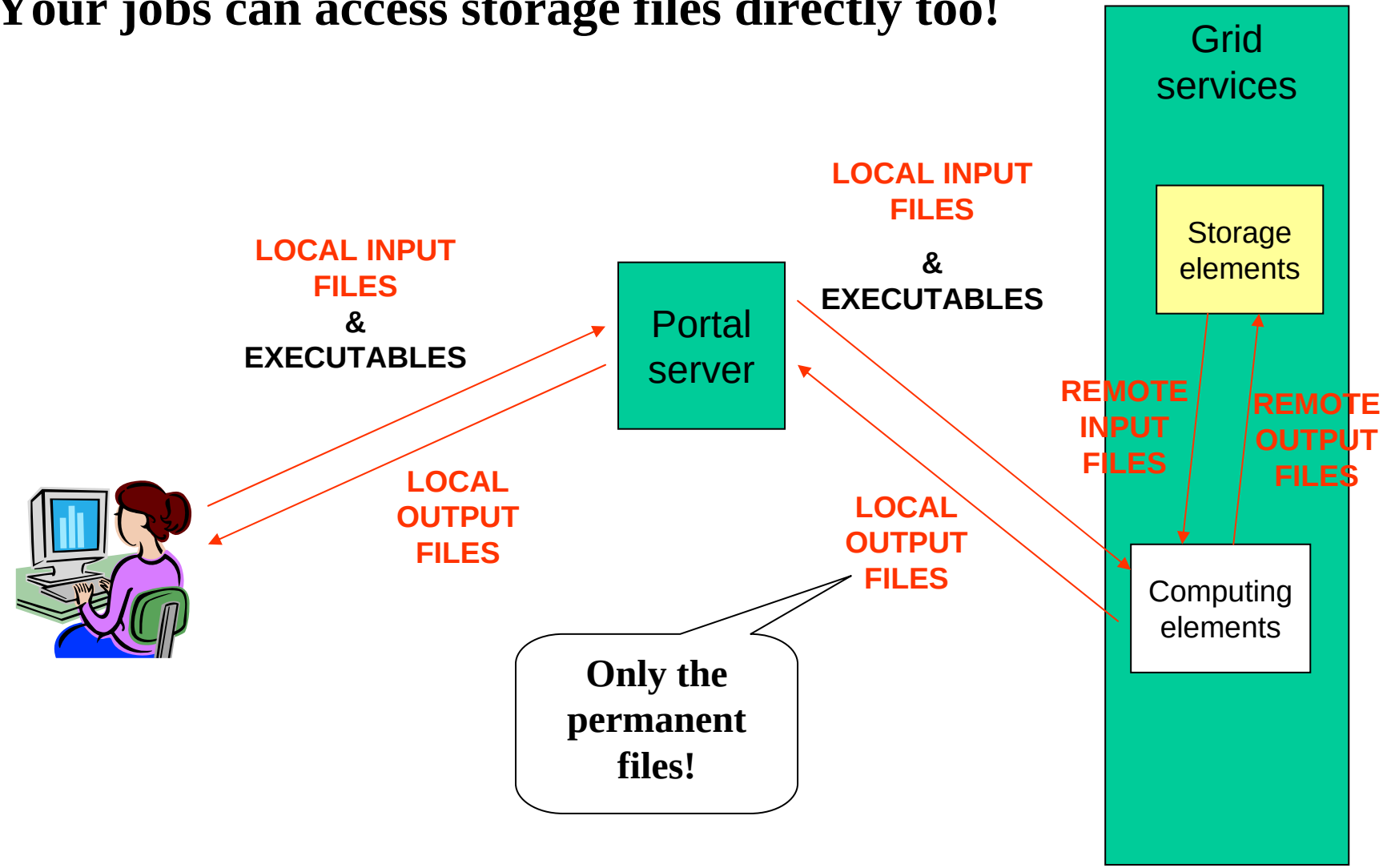
File storage type:
- **Permanent:** *belongs to the final results of the WF*
- **Volatile:** *used only for inter-job data transfer*

Possible file reference values

Input file	Output file
Local file	
• Client side location: c:\experiments\11-04.dat	• Client side location: result.dat
• LFC logical file name *(LFC file catalog is required – eGrid, Hungrid)* lfn:/grid/egrid/sipos/11-04.dat • GridFTP address (in Globus Grids): gsiftp://myhost.com/11-04.dat	• LFC logical file name *(LFC file catalog is required – eGrid, Hungrid)* lfn:/grid/egrid/sipos/11-04_-_result.dat • GridFTP address (in Globus Grids): gsiftp://myhost.com/11-04_-_result.dat
Remote file	

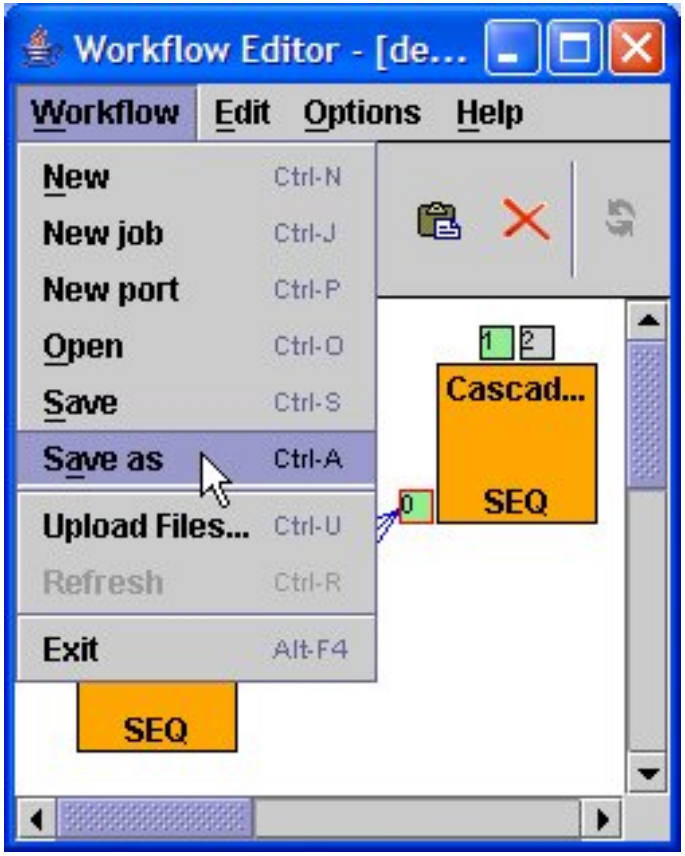
Local vs. remote files

Your jobs can access storage files directly too!



Workflow Editor

Saving the workflow



Workflow is defined!

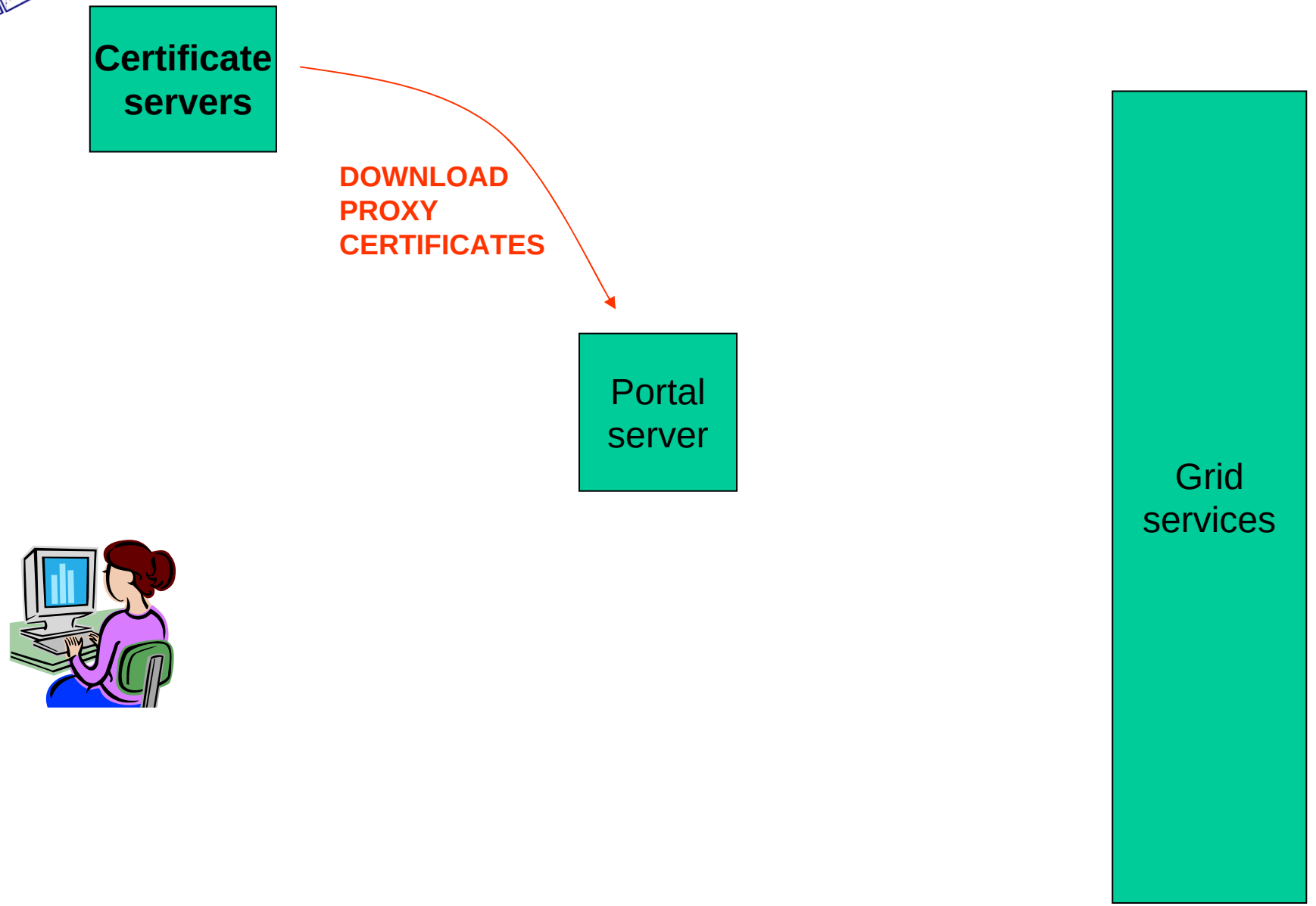
Let's execute it!

Executing workflows with the P-GRADE Portal

Main steps

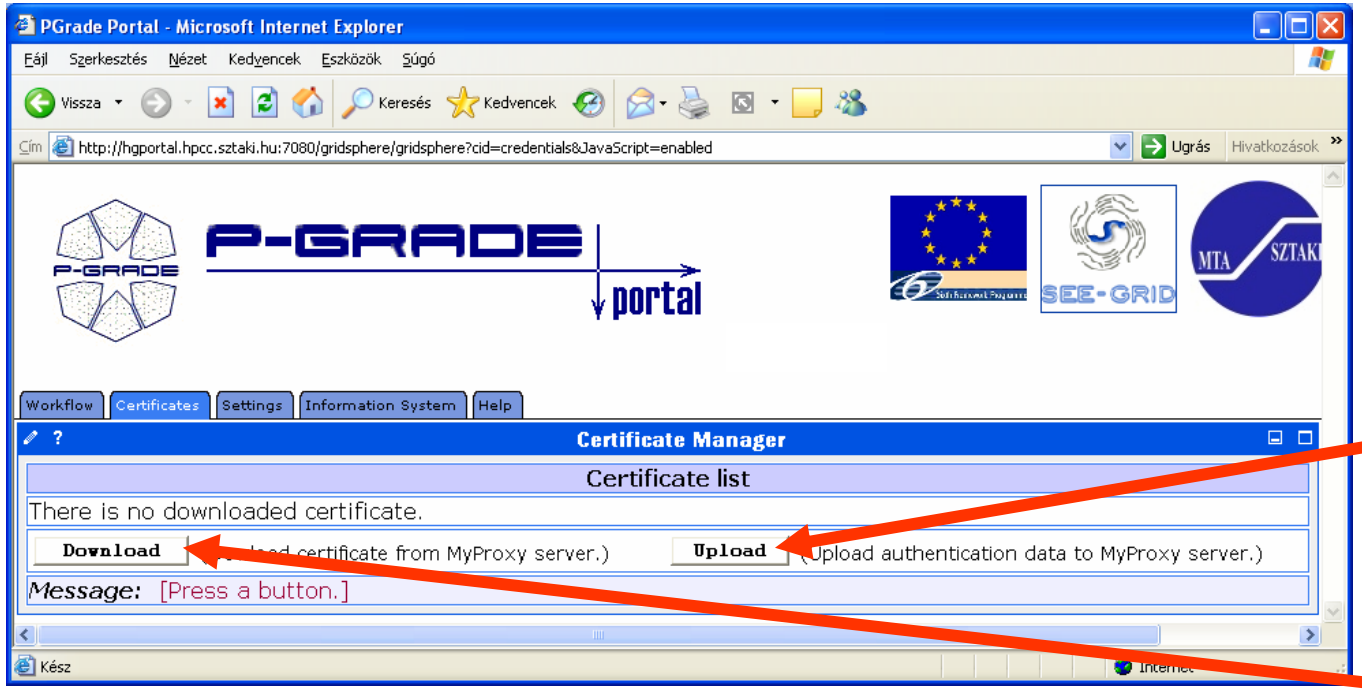
1. Download proxies
2. Submit workflow
3. Observe workflow progress
4. If some error occurs correct the graph
5. Download result

The typical user scenario
Execution phase – step 1:



Certificate Manager

Certificates portlet

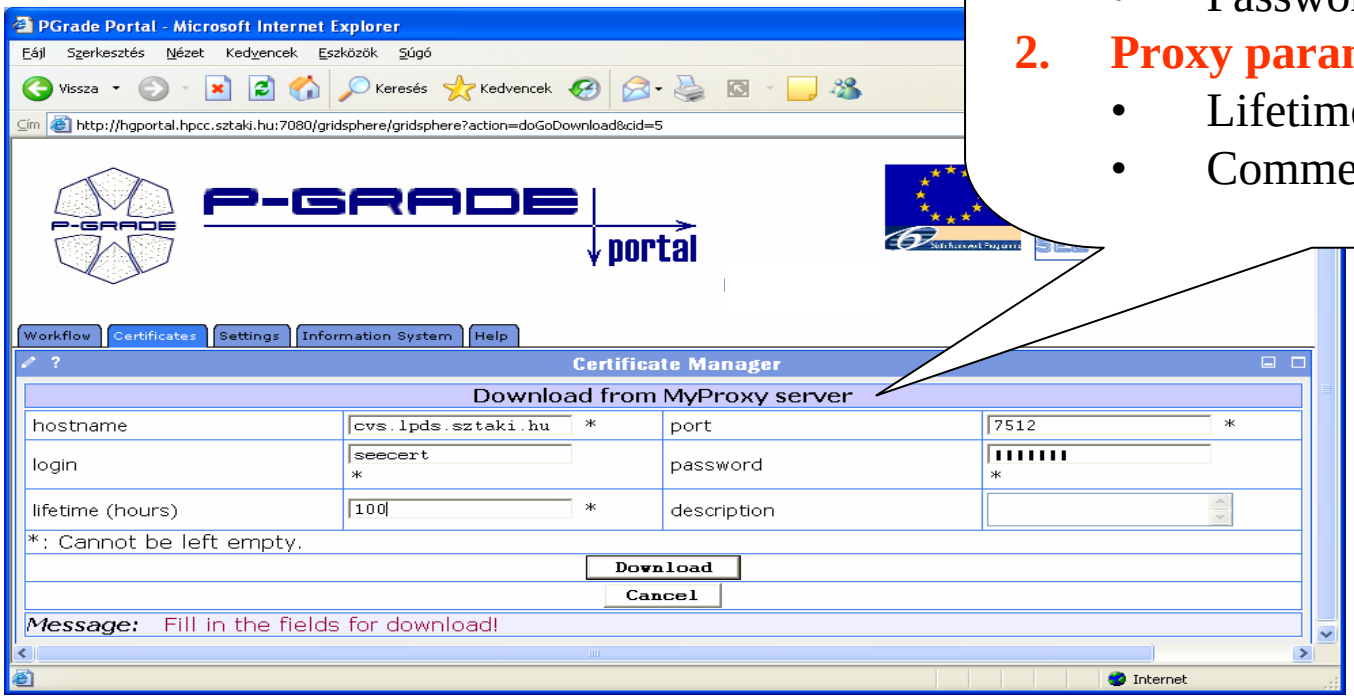


- To access GSI-based Grids the portal server application needs proxy certificates
- "Certificates" portlet:
  - to upload X.509 certificates into MyProxy servers
  - to download short-term proxy credentials into the portal server application

Certificate Manager

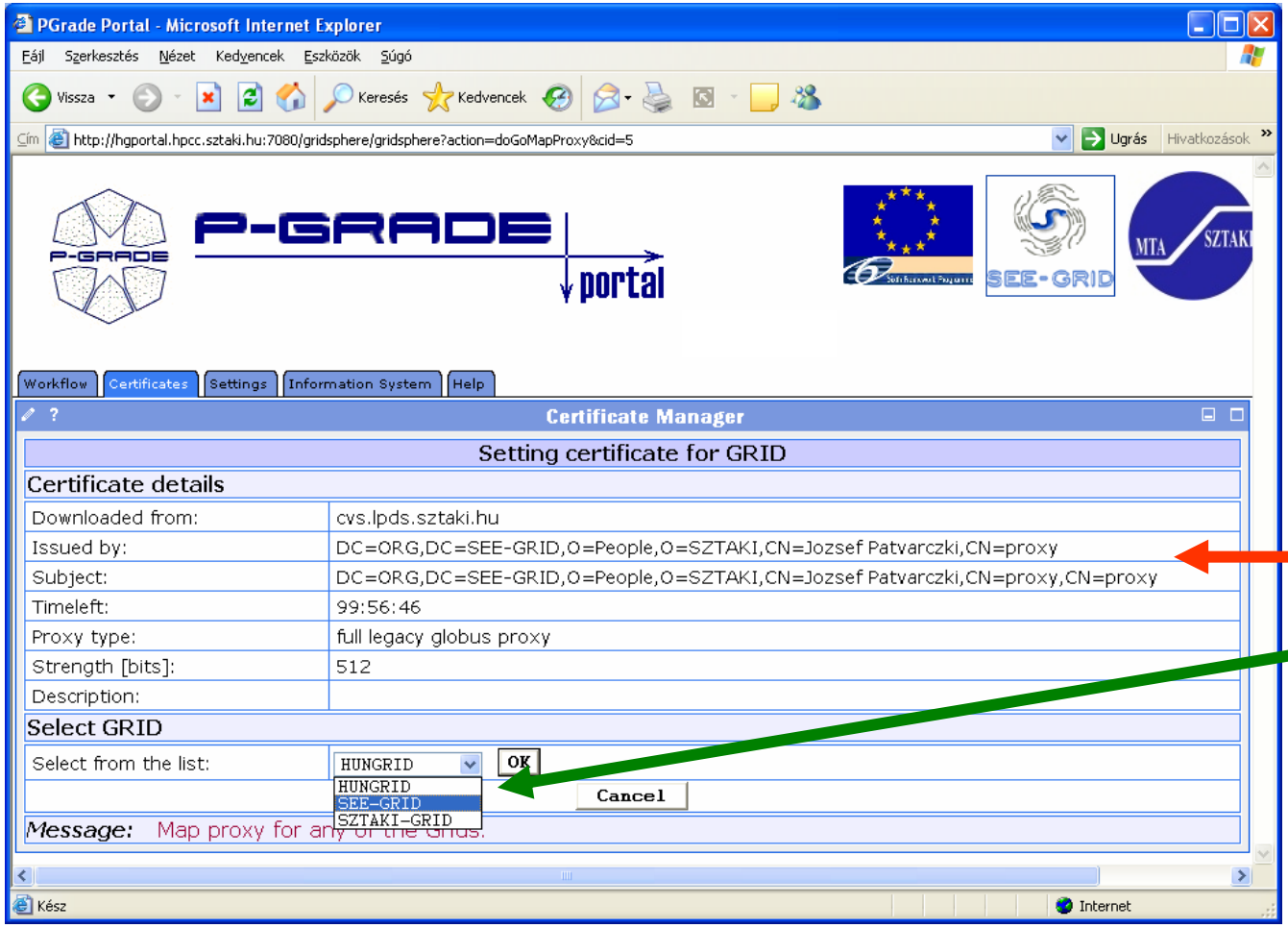
Downloading a proxy

1. **MyProxy server access details:**
   - Hostname
   - Port number
   - User name (from upload)
   - Password (from upload)
2. **Proxy parameters:**
   - Lifetime
   - Comment



Certificate Manager

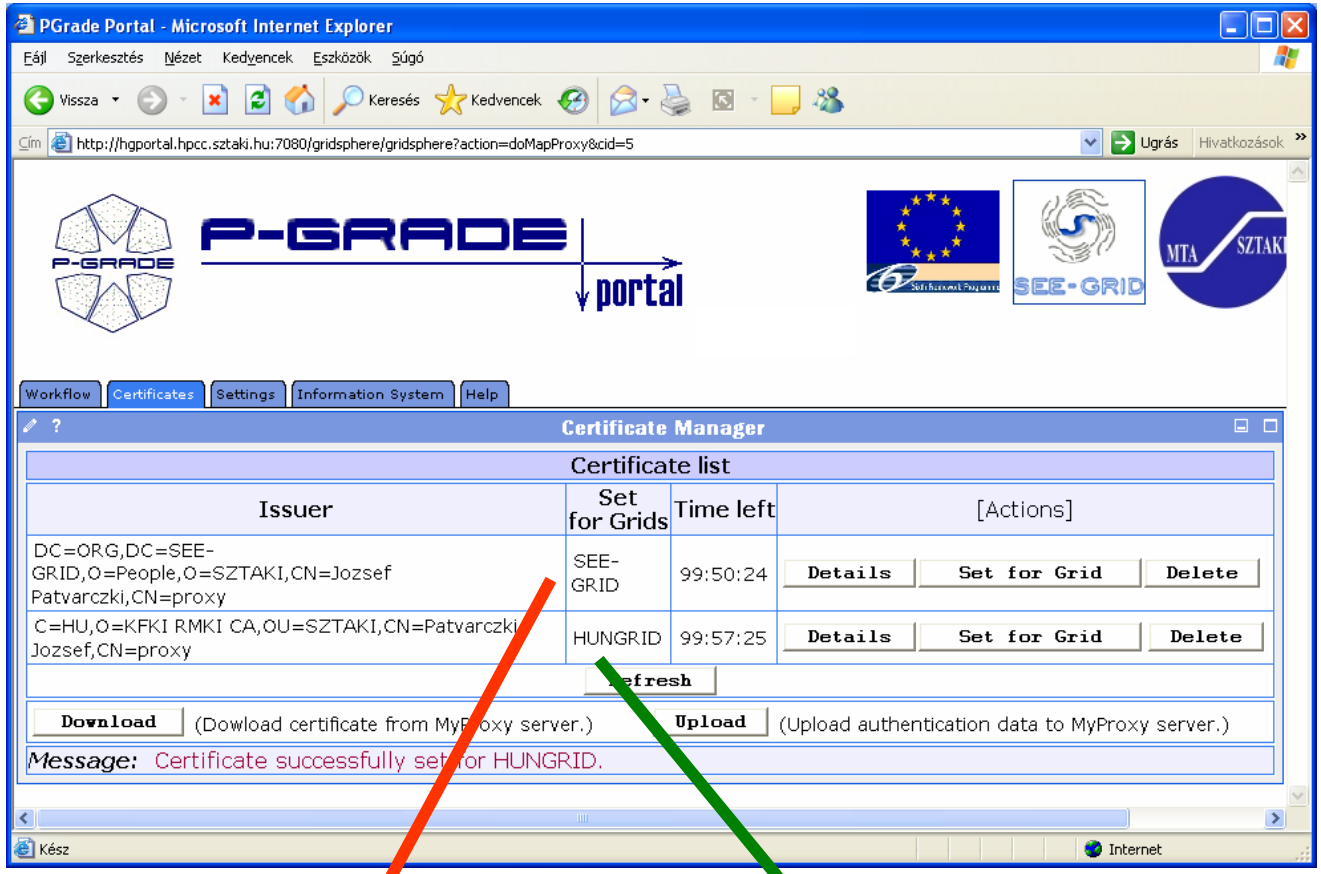
associating the proxy with a grid



This operation displays the details of the certificate and the list of available Grids *(defined by portal administrator)*

Certificate Manager

browsing proxies



SEE-GRID CEs and SEs

HUNGRID CEs and SEs

Multiple proxies can be available on the portal server at the same time!

The typical user scenario
Execution phase - step 2:

Certificate servers

TRANSFER FILES, SUBMIT JOBS

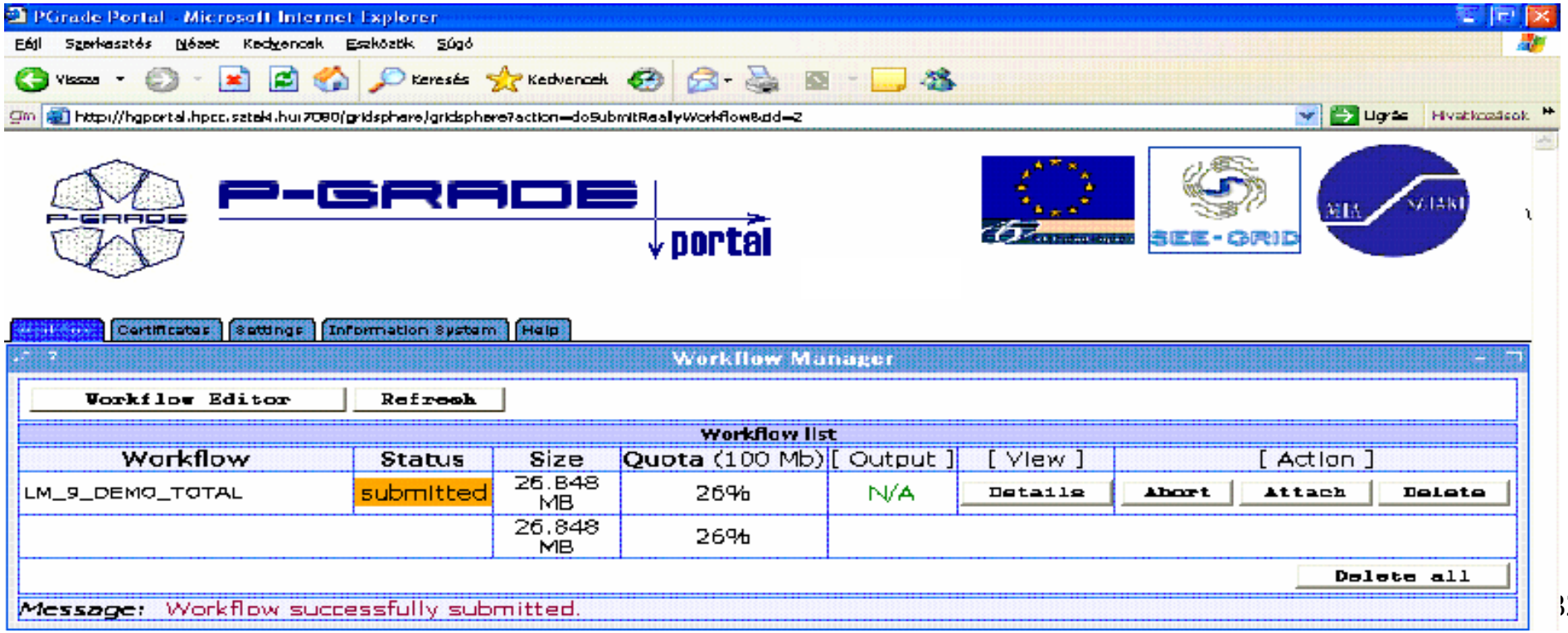
Portal server

Grid services

Workflow Management

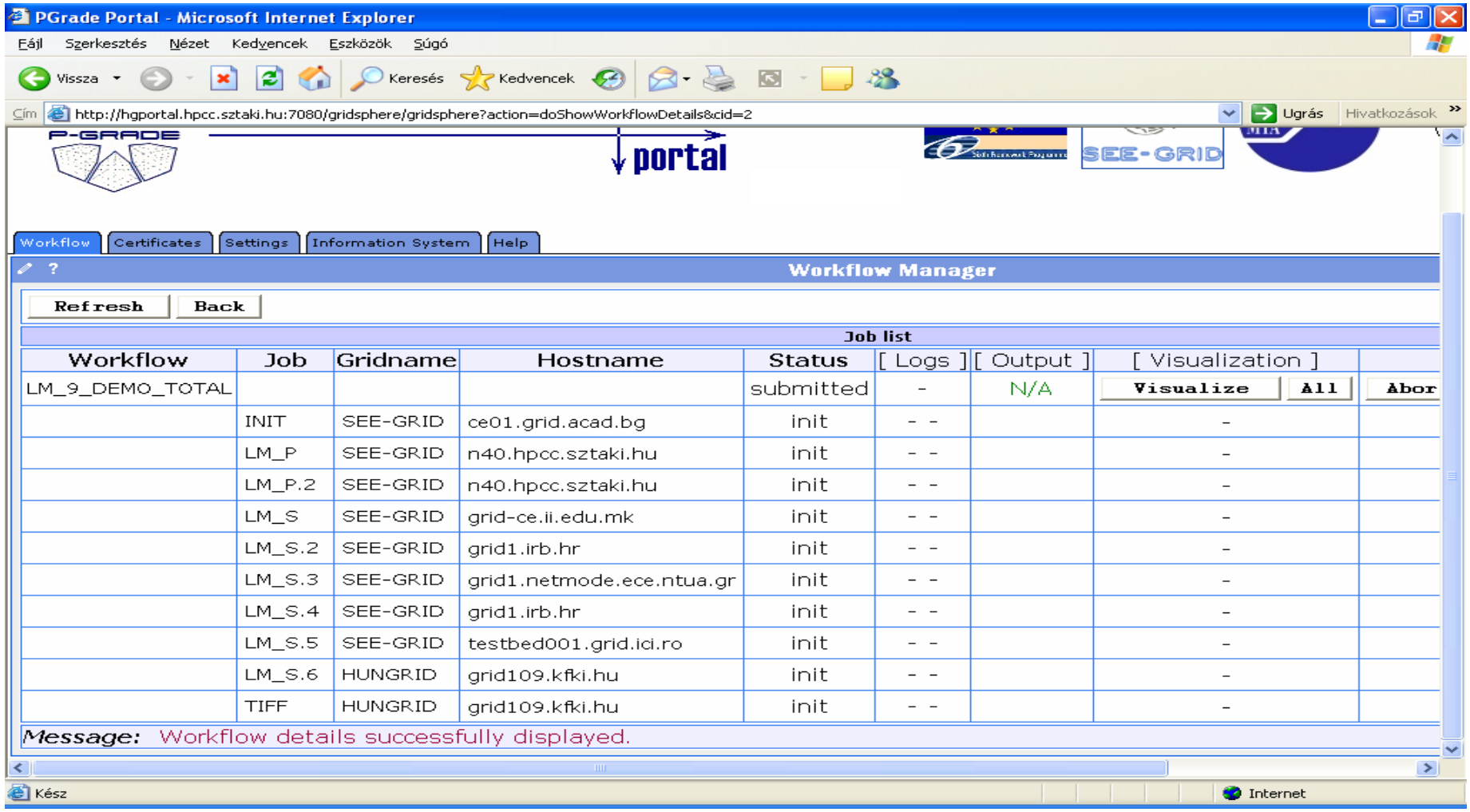
(workflow portlet)

- The portlet presents the status, size and output of the available workflow in the "Workflow" list
- It has a Quota manager to control the users' storage space on the server
- The portlet also contains the "Abort", "Attach", "Details", "Delete" and "Delete all" buttons to handle execution of workflows
- The "Attach" button opens the workflow in the Workflow Editor
- The "Details" button gives an overview about the jobs of the workflow



Workflow Execution

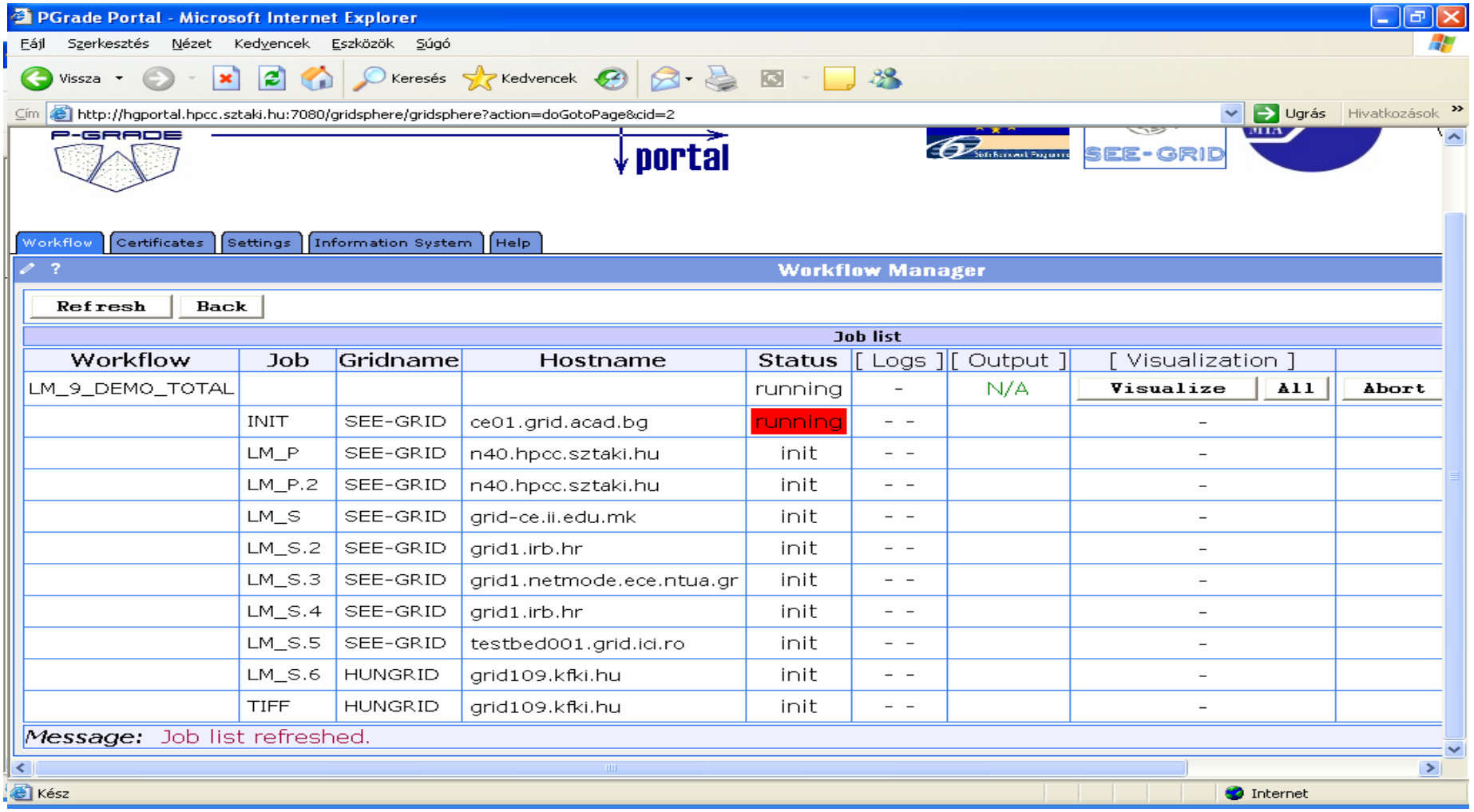
(observation by the workflow portlet)



White/Red/Green color means the job is initial/running/finished state

Workflow Execution

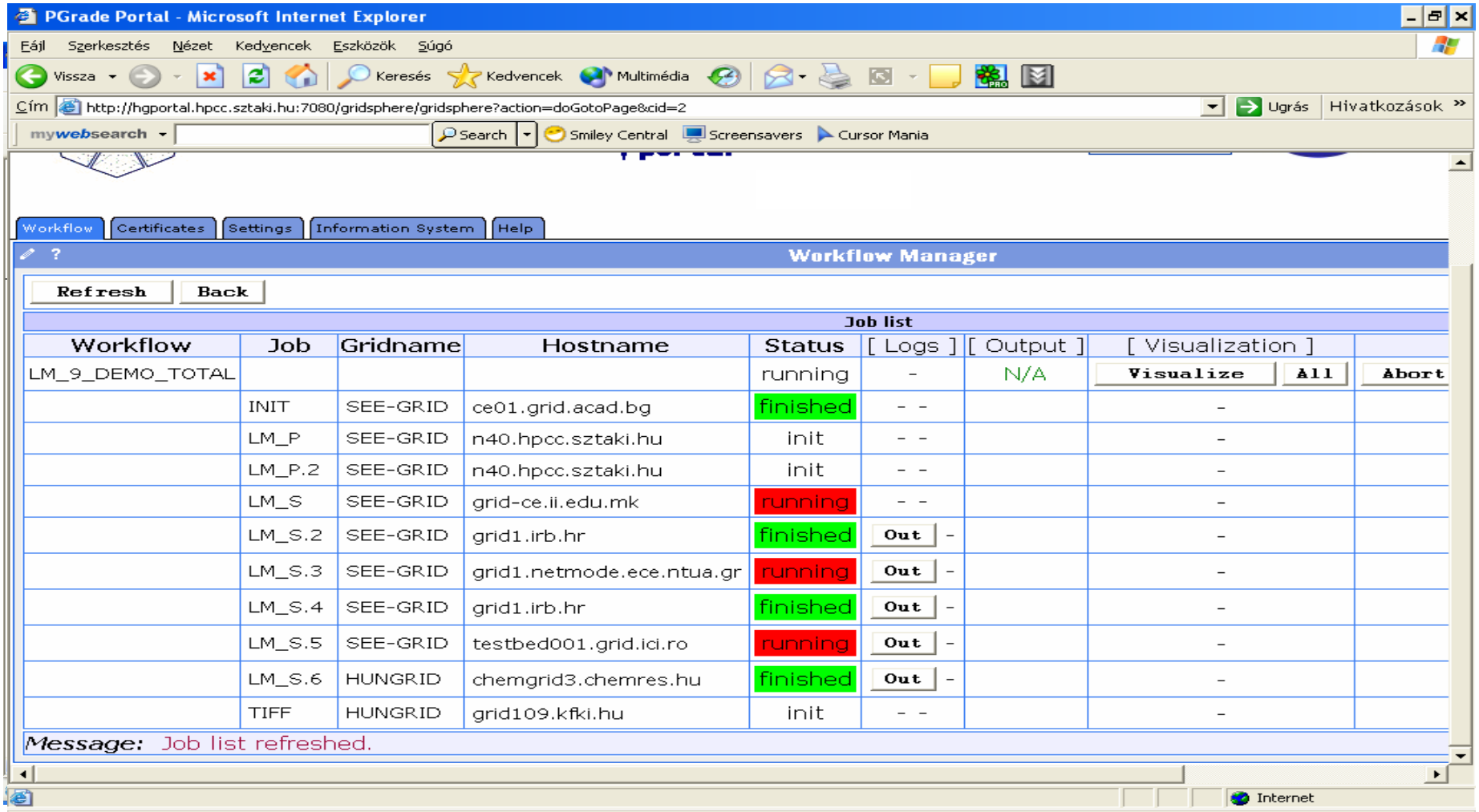
(observation by the workflow portlet)



White/Red/Green color means the job is initial/running/finished state

Workflow Execution

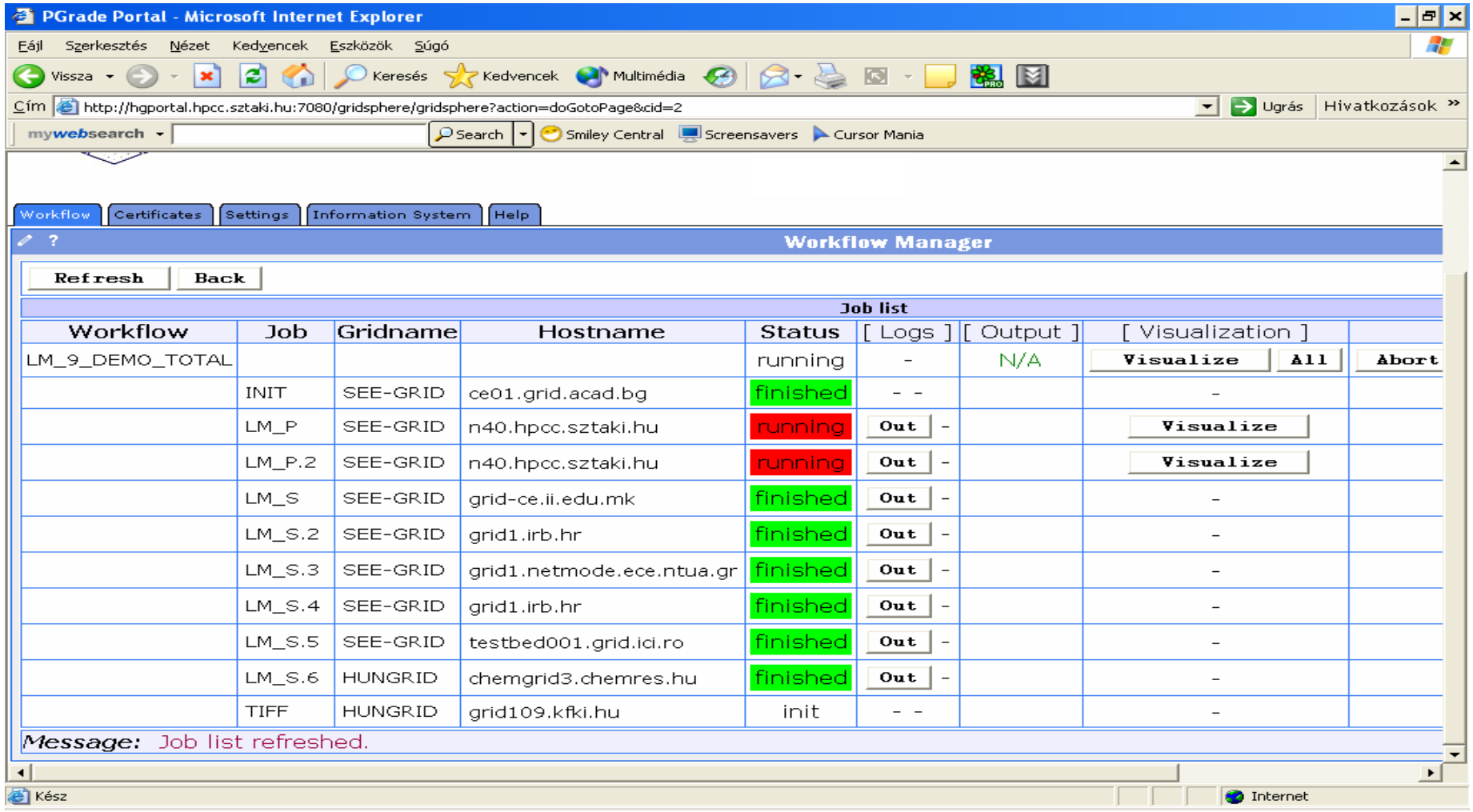
(observation by the workflow portlet)



White/Red/Green color means the job is initial/running/finished state

Workflow Execution

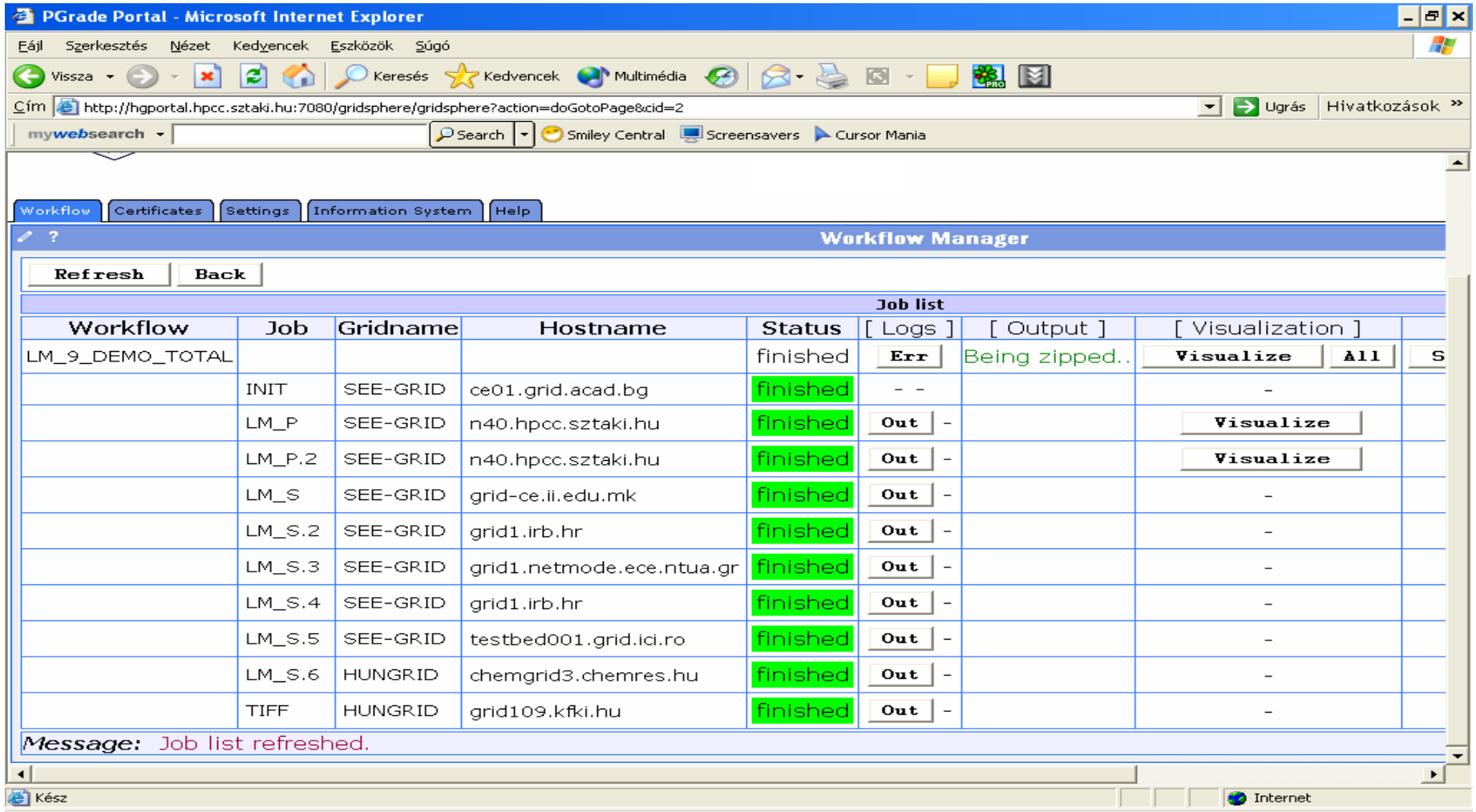
(observation by the workflow portlet)



White/Red/Green color means the job is initial/running/finished state

Workflow Execution

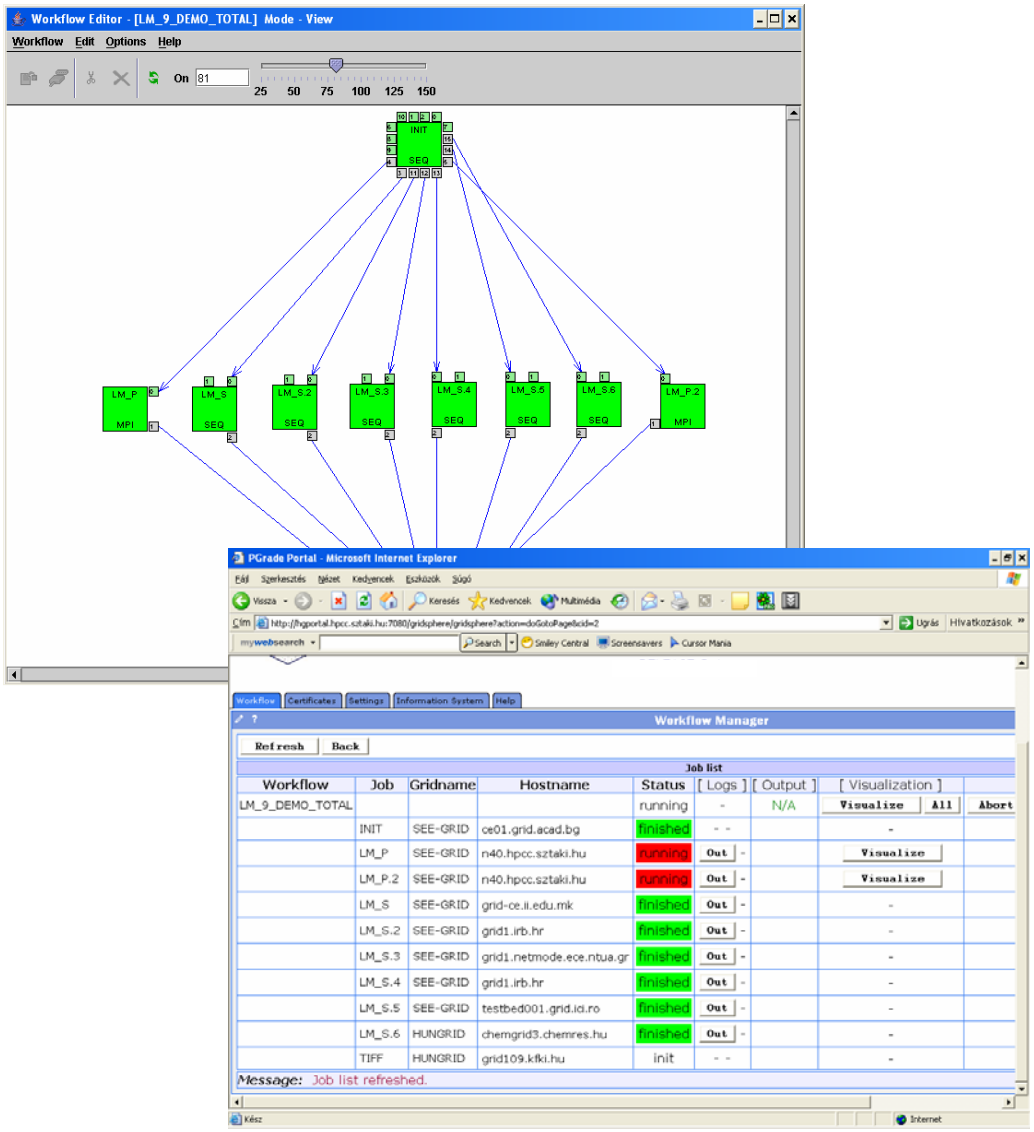
(observation by the workflow portlet)



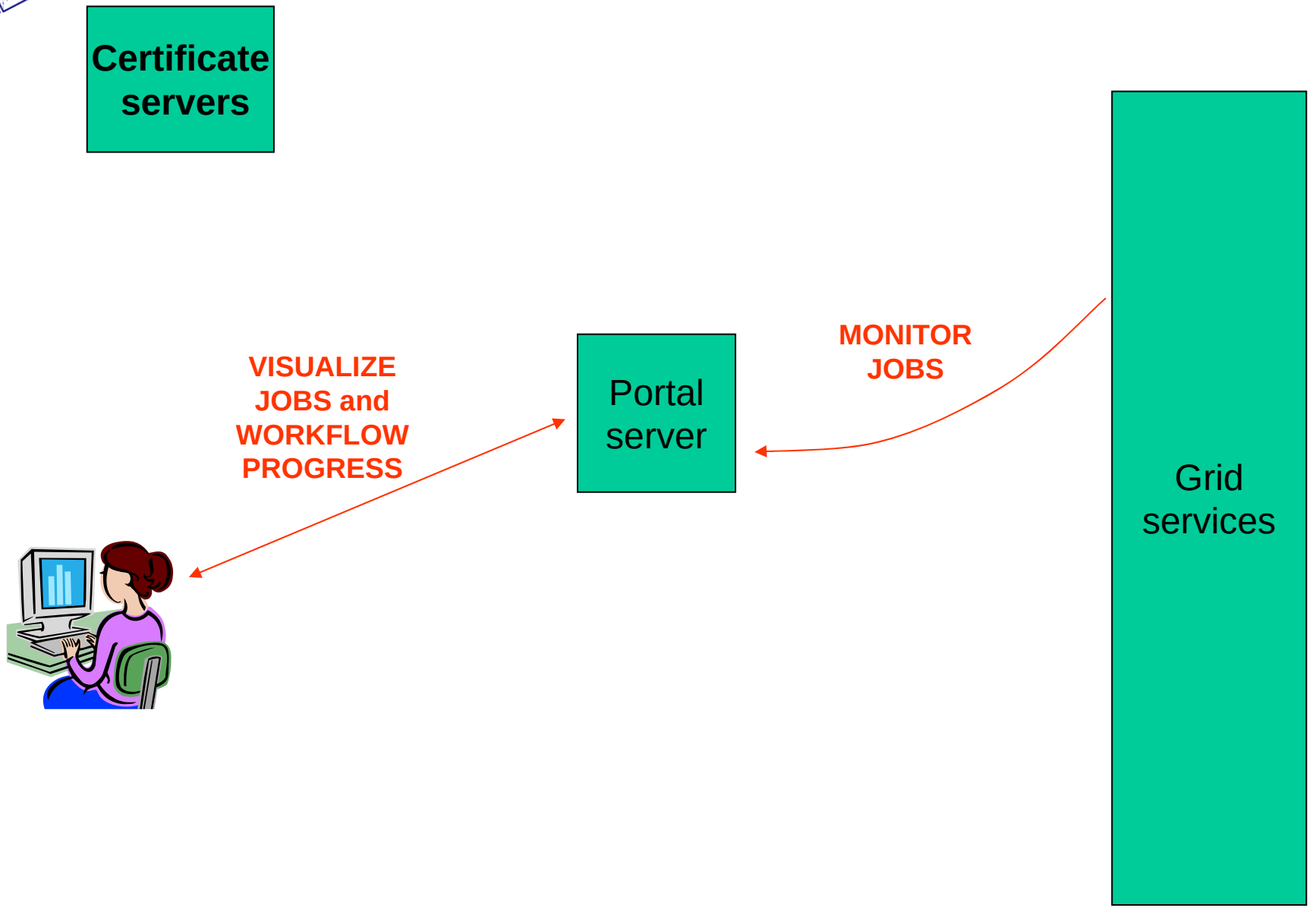
White/Red/Green color means the job is initialised/running/finished

Workflow Execution

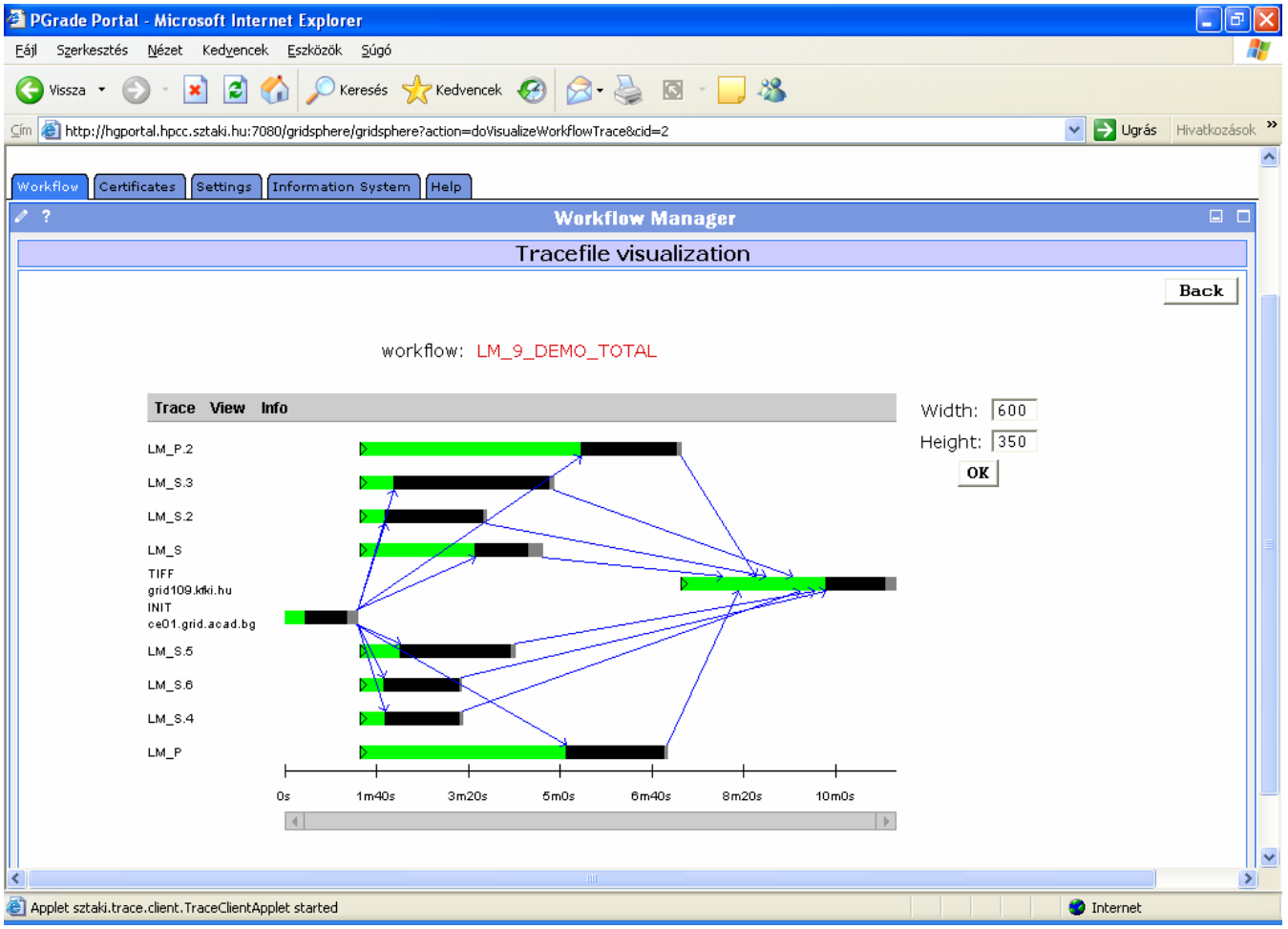
What about data transfers?



The typical user scenario
Execution phase – step 3:



On-Line Monitoring both at the workflow and job levels *(workflow portlet)*

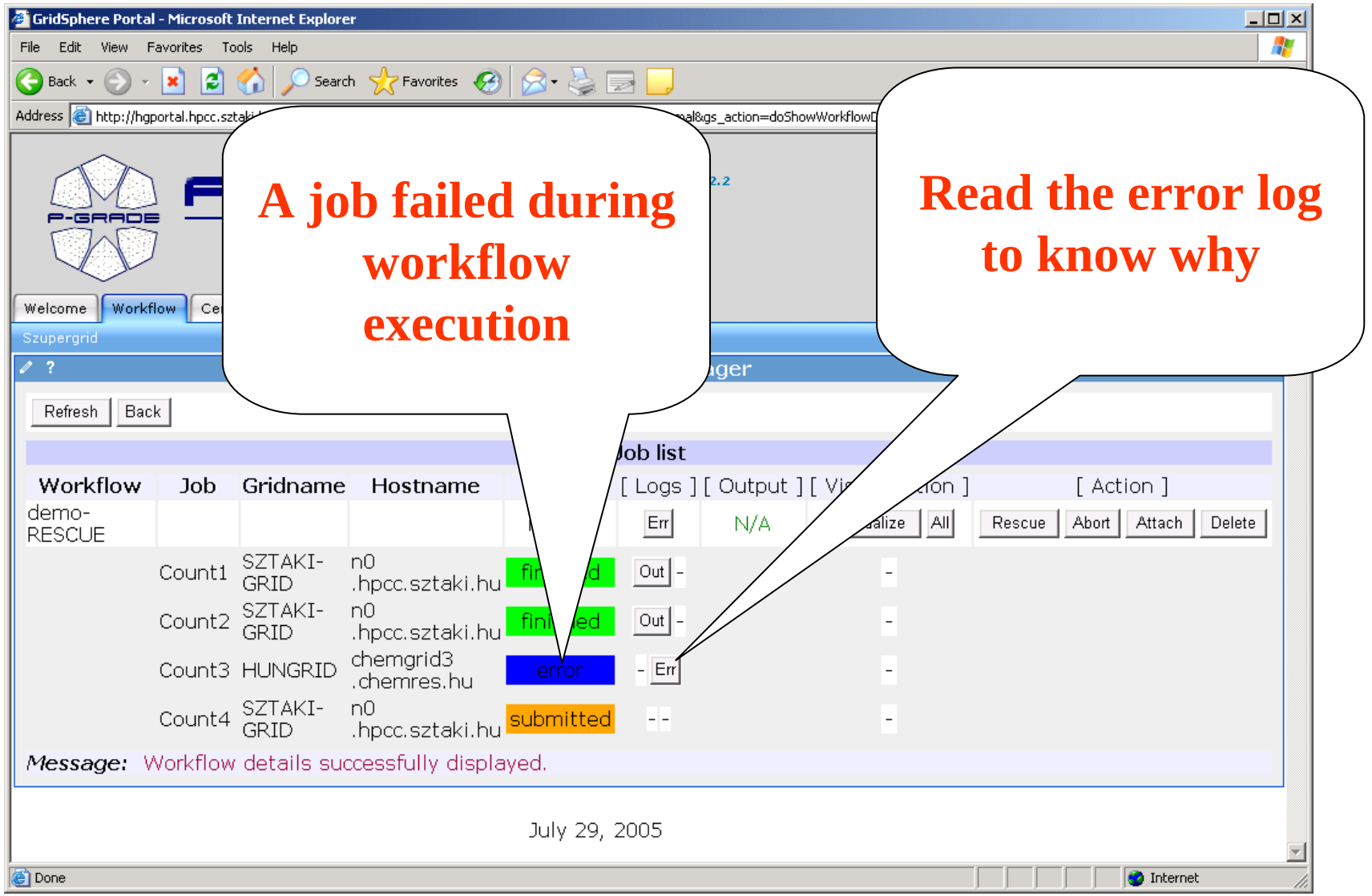


- The portal monitors and visualizes workflow progress

On-Line Monitoring both at the workflow and job levels *(workflow portlet)*

- **The portal monitors and visualizes parallel jobs**
  (if they are prepared for Mercury monitor)

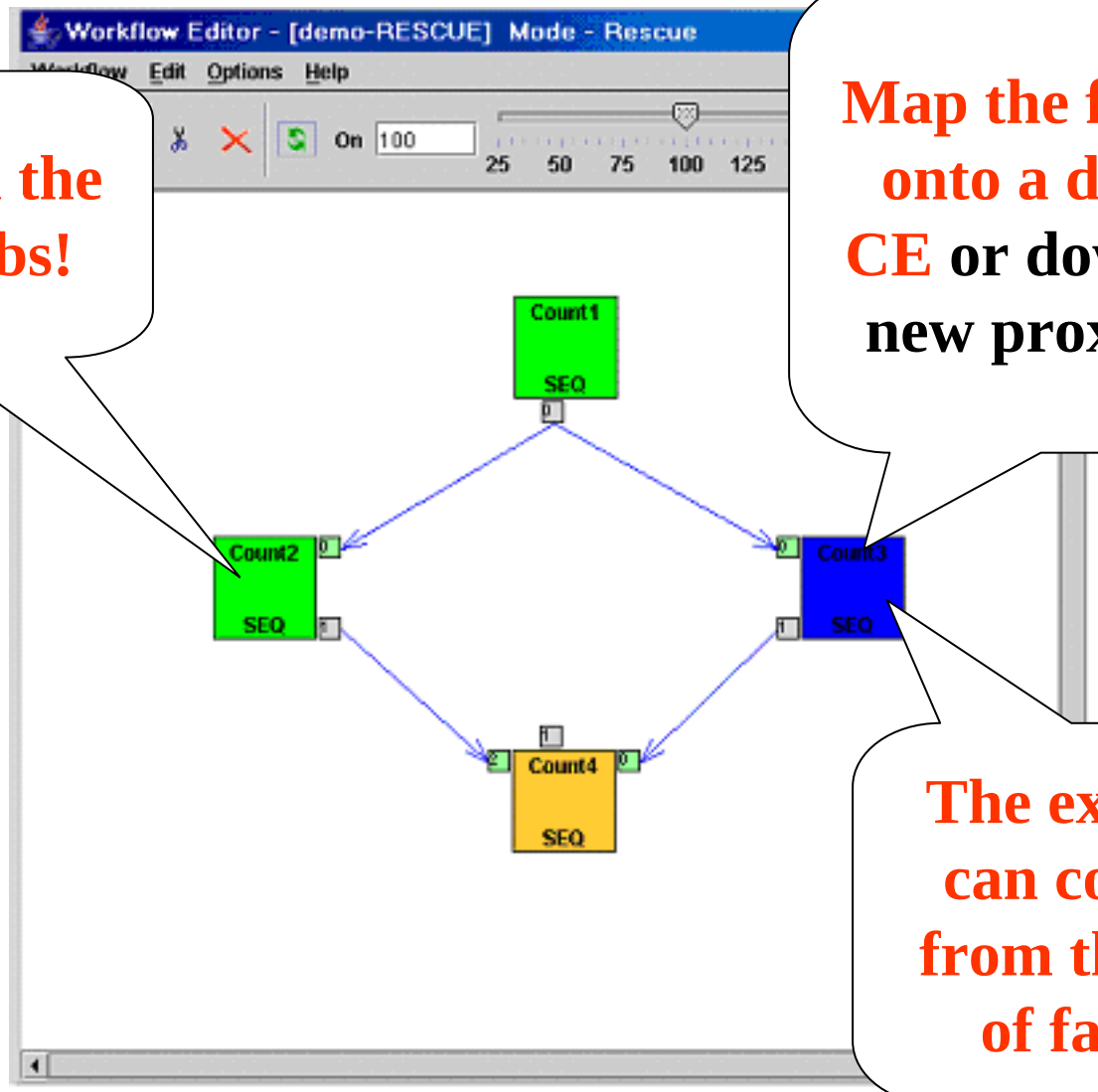
Rescuing a failed workflow 1.



A job failed during workflow execution

Read the error log to know why

Rescuing a failed workflow 2.

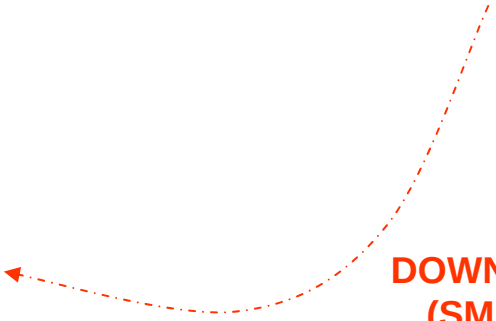


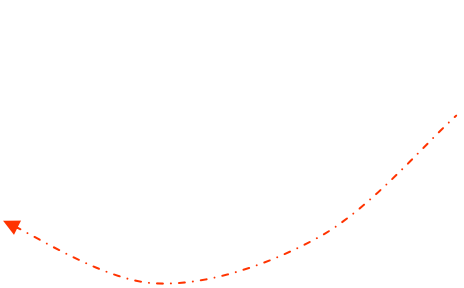
Don't touch the finished jobs!

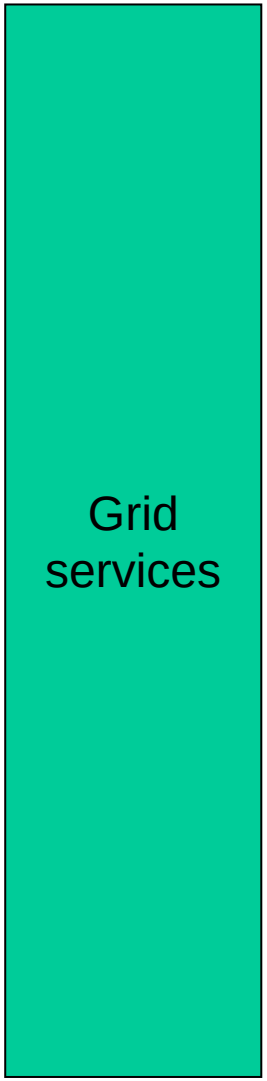
Map the failed job onto a different CE or download a new proxy for it.

The execution can continue from the point of failure!

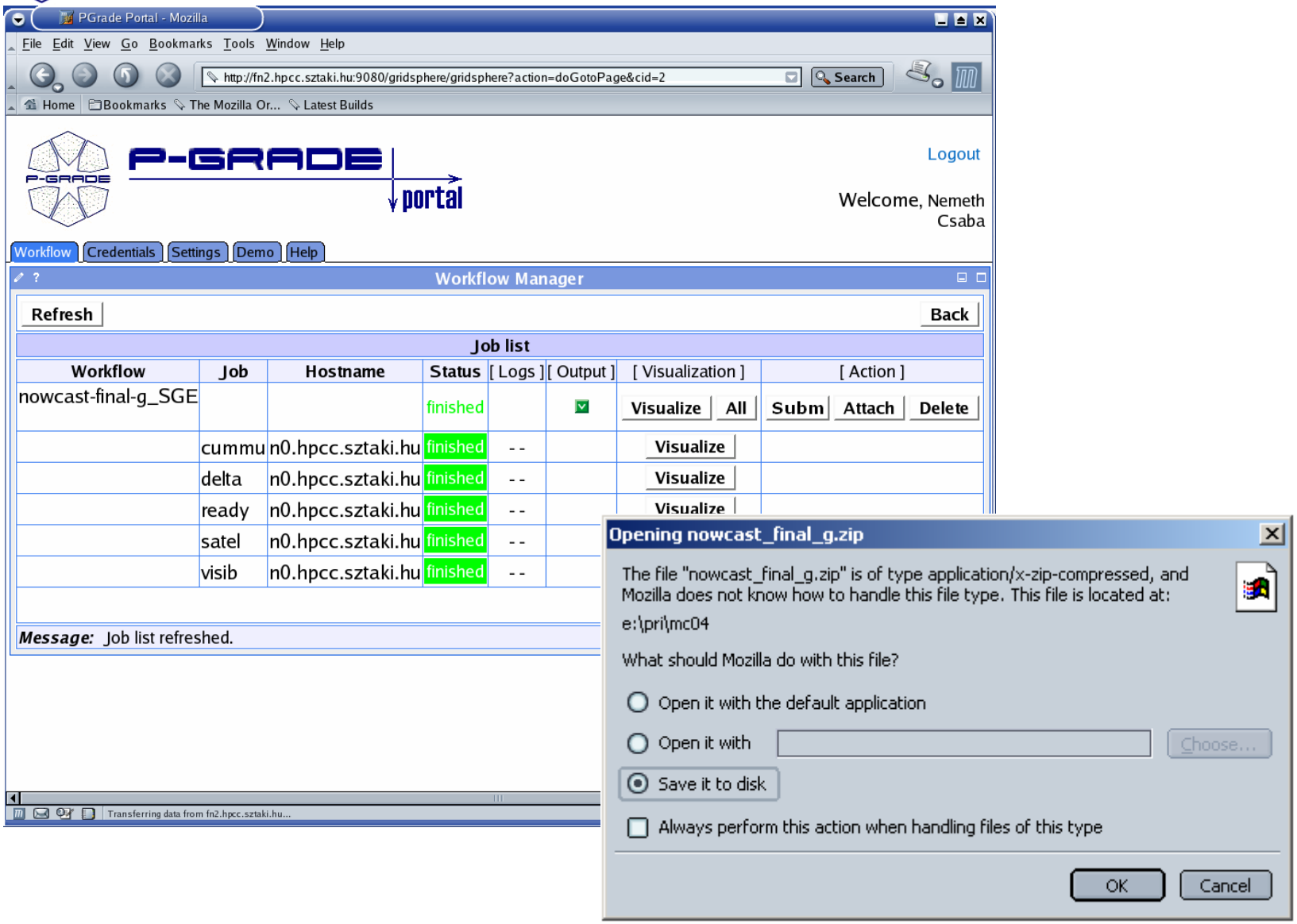
The typical user scenario
Execution phase – step 5







Downloading the results…

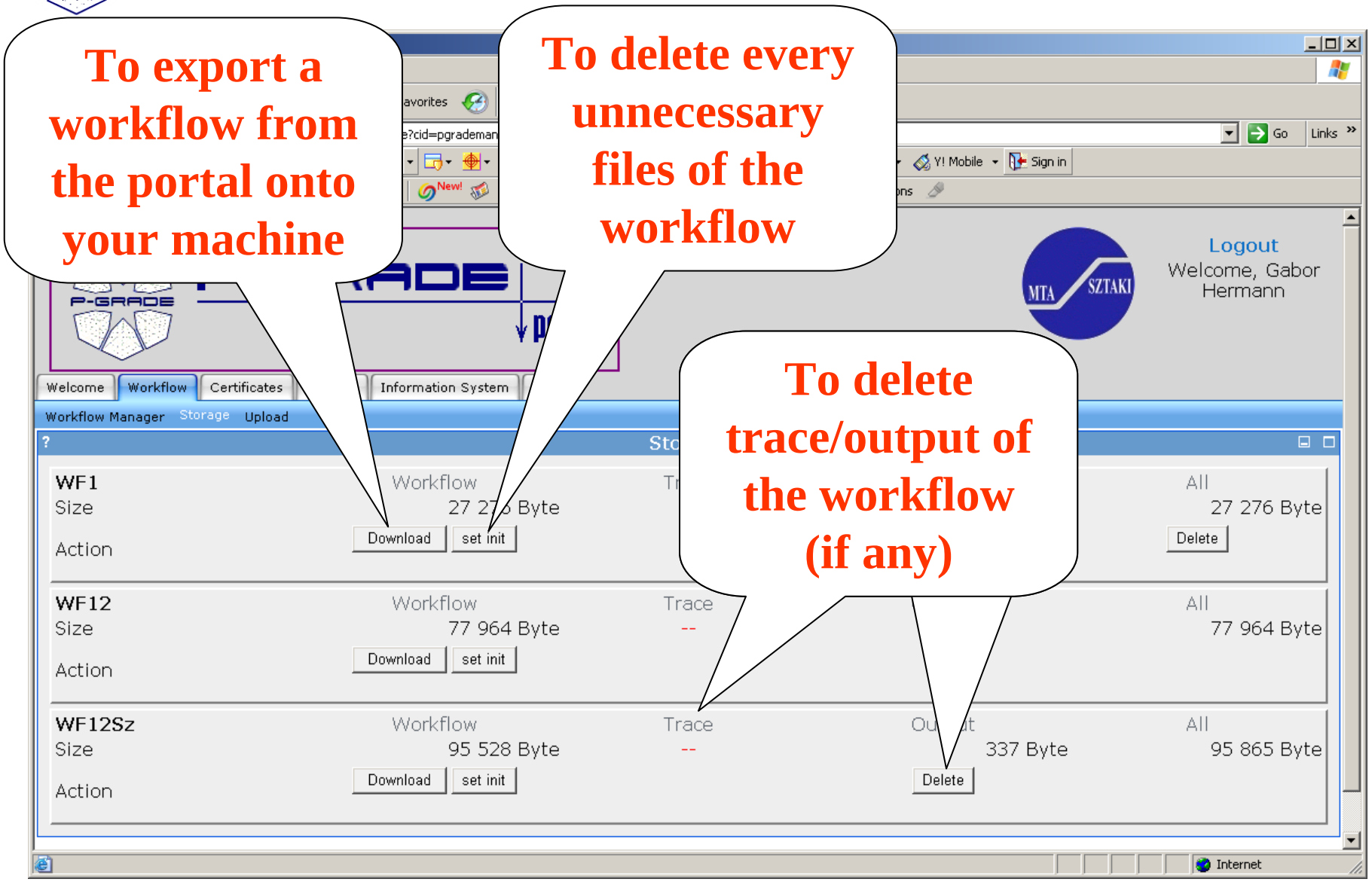


Additional features

- **Workflows and traces** can be exported from the portal server onto your client machine
- **Workflows and traces** can be imported into the Portal

- **Share your workflows or results with other researchers!**
- **Migrate your application from one portal into another!**

Workflow/trace export/import



References

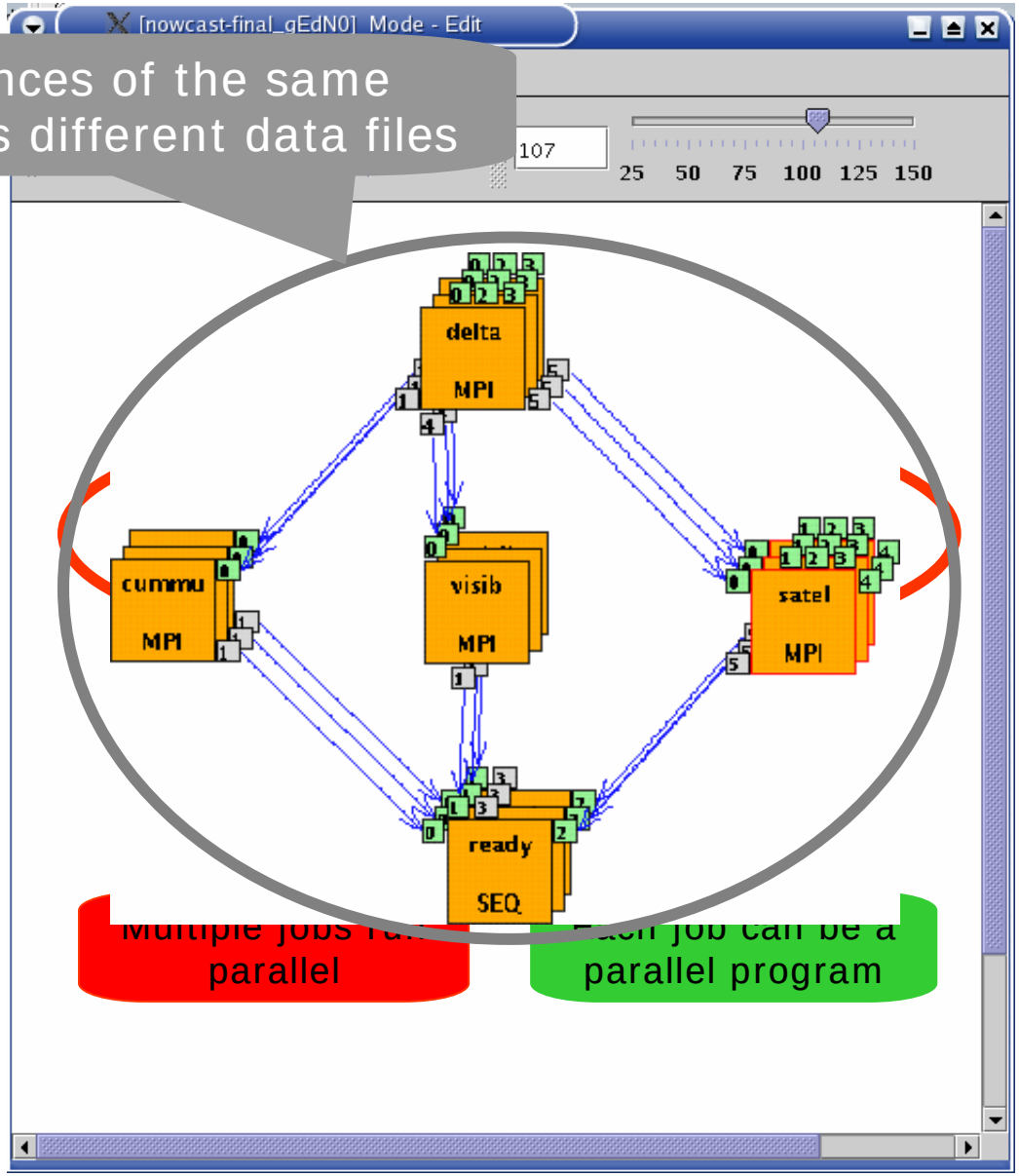
- **P-GRADE Portal service is available for**
  - **SEE-GRID infrastructure**
  - **Central European VO of EGEE**
  - **GILDA: Training VO of EGEE**
  - **Many national Grids (UK National Grid Service, HunGrid, etc.)**
  - **US Open Science Grid, TeraGrid**
  - **Economy-Grid, Swiss BioGrid, Bio and Biomed EGEE VOs, BioInfoGrid, BalticGrid**
  - **GIN VO**

Parameter study extension of the portal

- Users want parameter study (PS) support at workflow level
- It means:
  - If the user has an existing workflow in a repository, he would like to run
    - the same workflow (without any change)
    - with many different parameters

Introducing three levels of parallelism

Multiple instances of the same workflow process different data files



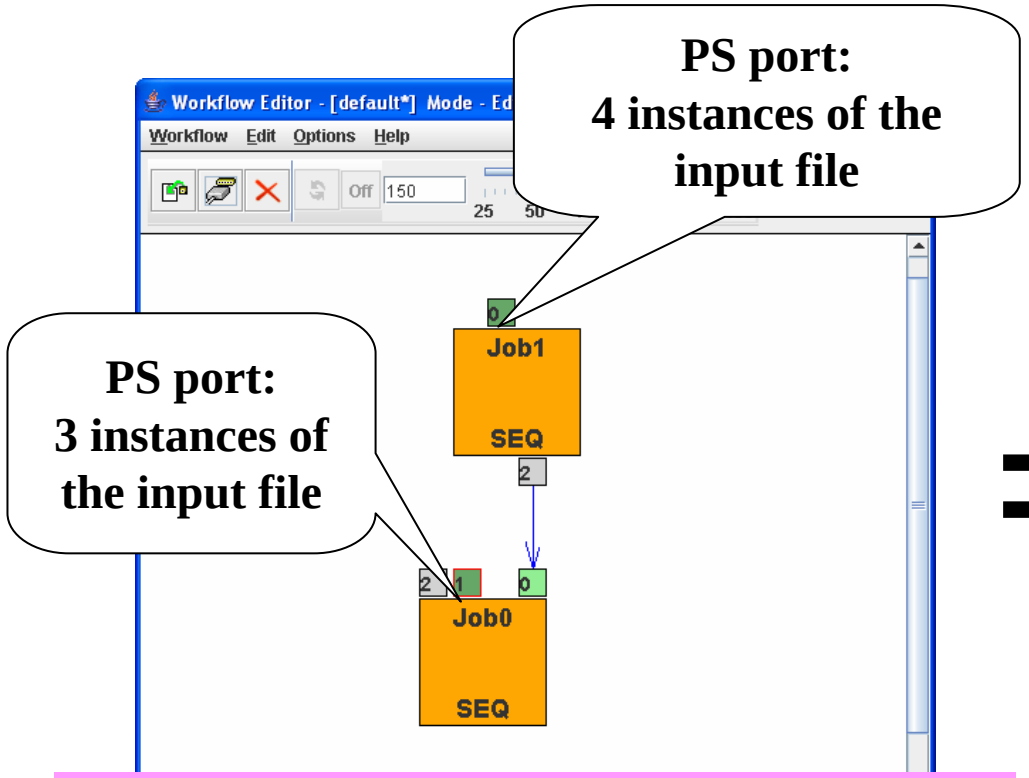
– **Parallel execution inside a workflow node (SIMD/MIMD/MISD)**

– **Parallel execution among workflow nodes (SIMD/MIMD/MISD)**

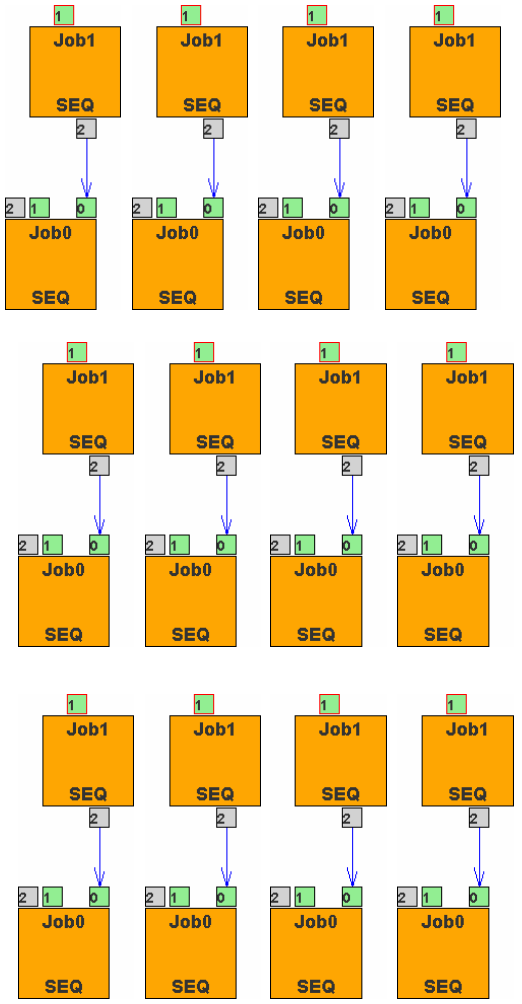
– **Parameter study execution of the workflow (SIMD)**

Parameter sweep (PS) workflow execution in P-GRADE portal

1 PS workflow execution

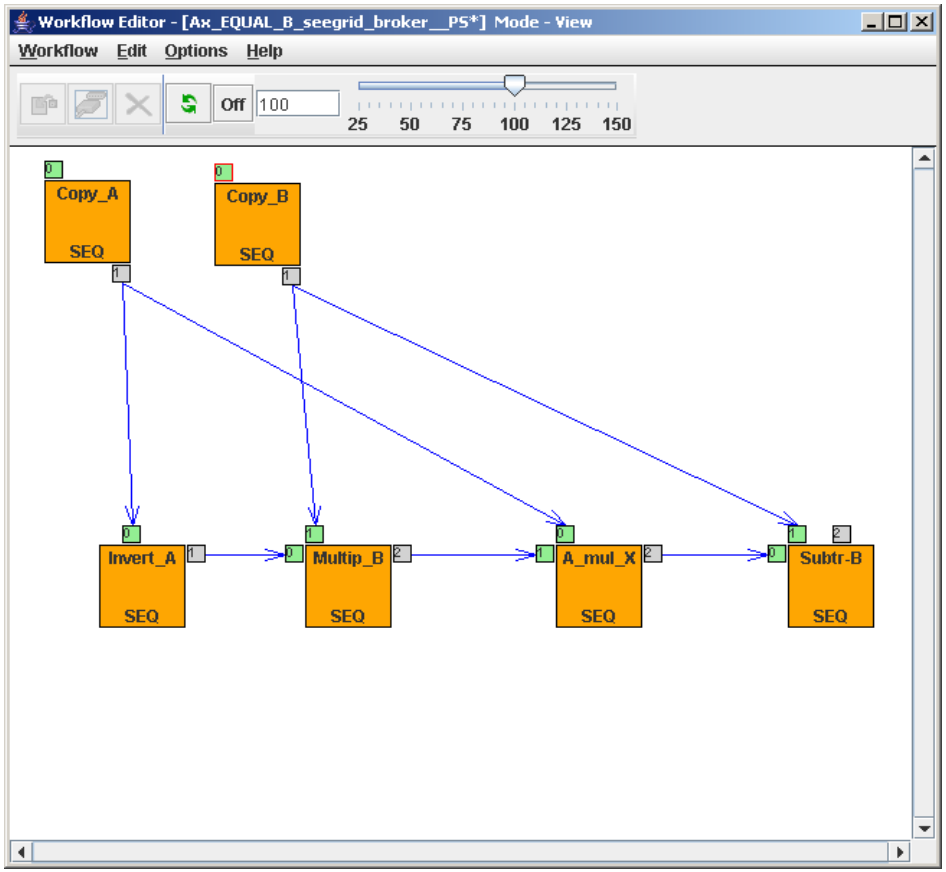


4 x 3 normal workflow execution

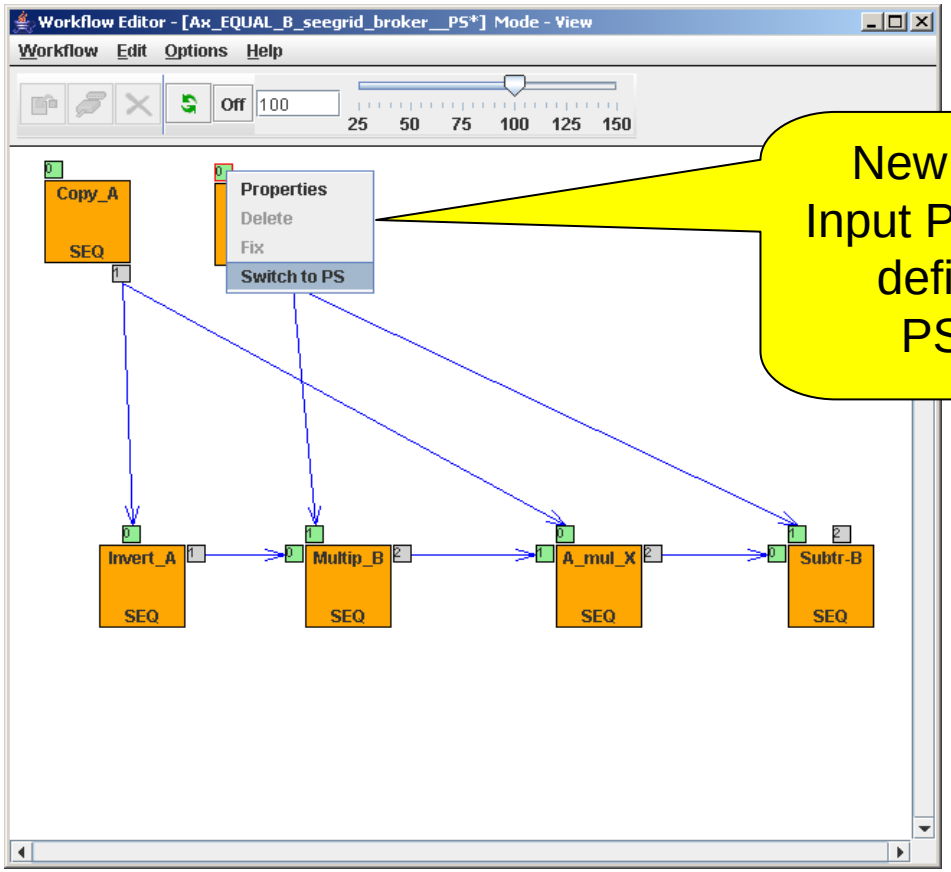


This provides the 3rd level of parallelism resulting a very large demand for Grid resources

Steps of creating a PS-workflow
1. Start from a tested Workflow

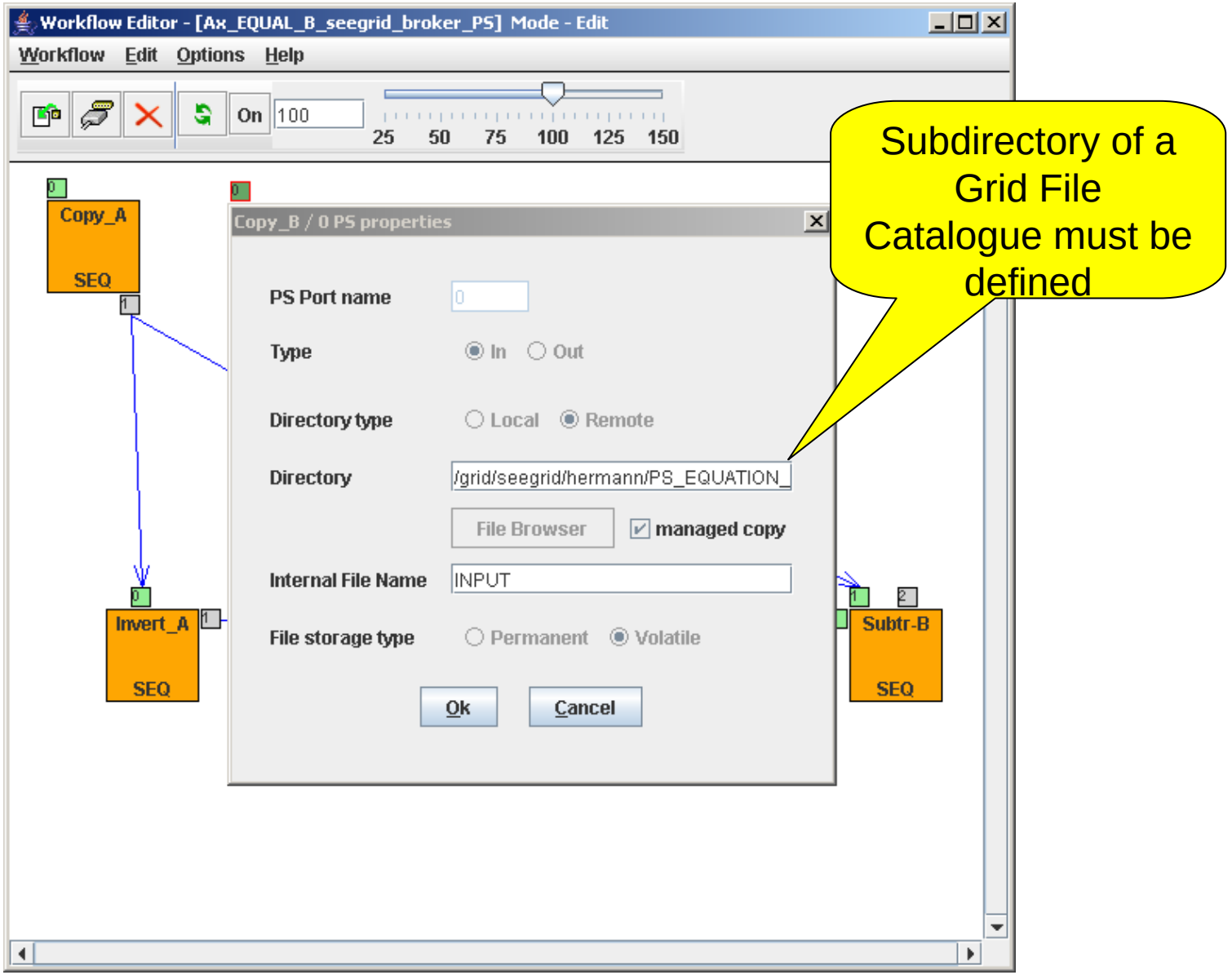


2. Define the PS Port(s)

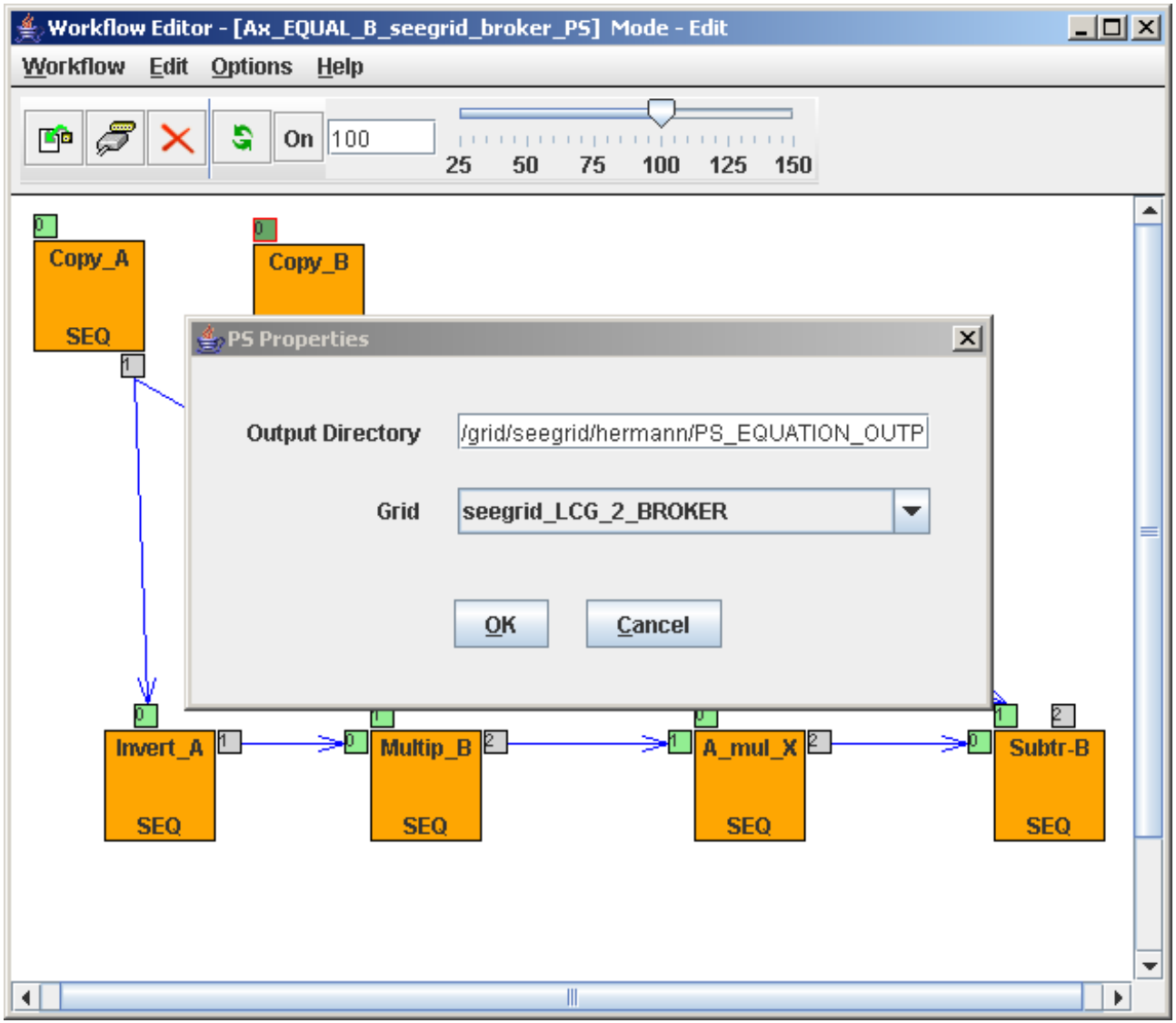


New feature:
Input Port can be defined as PS_port

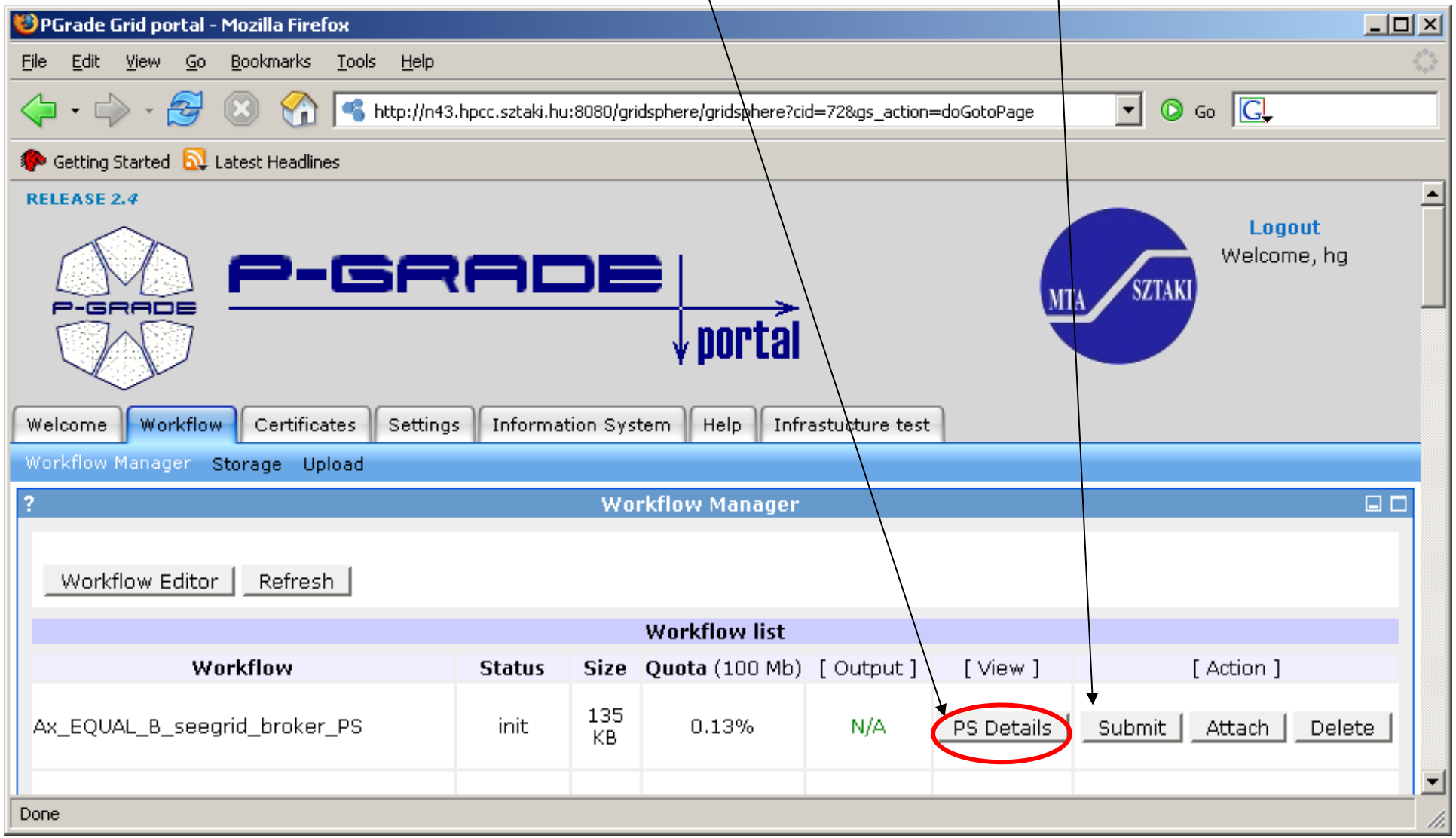
3.Define Subdirectory of Inputs



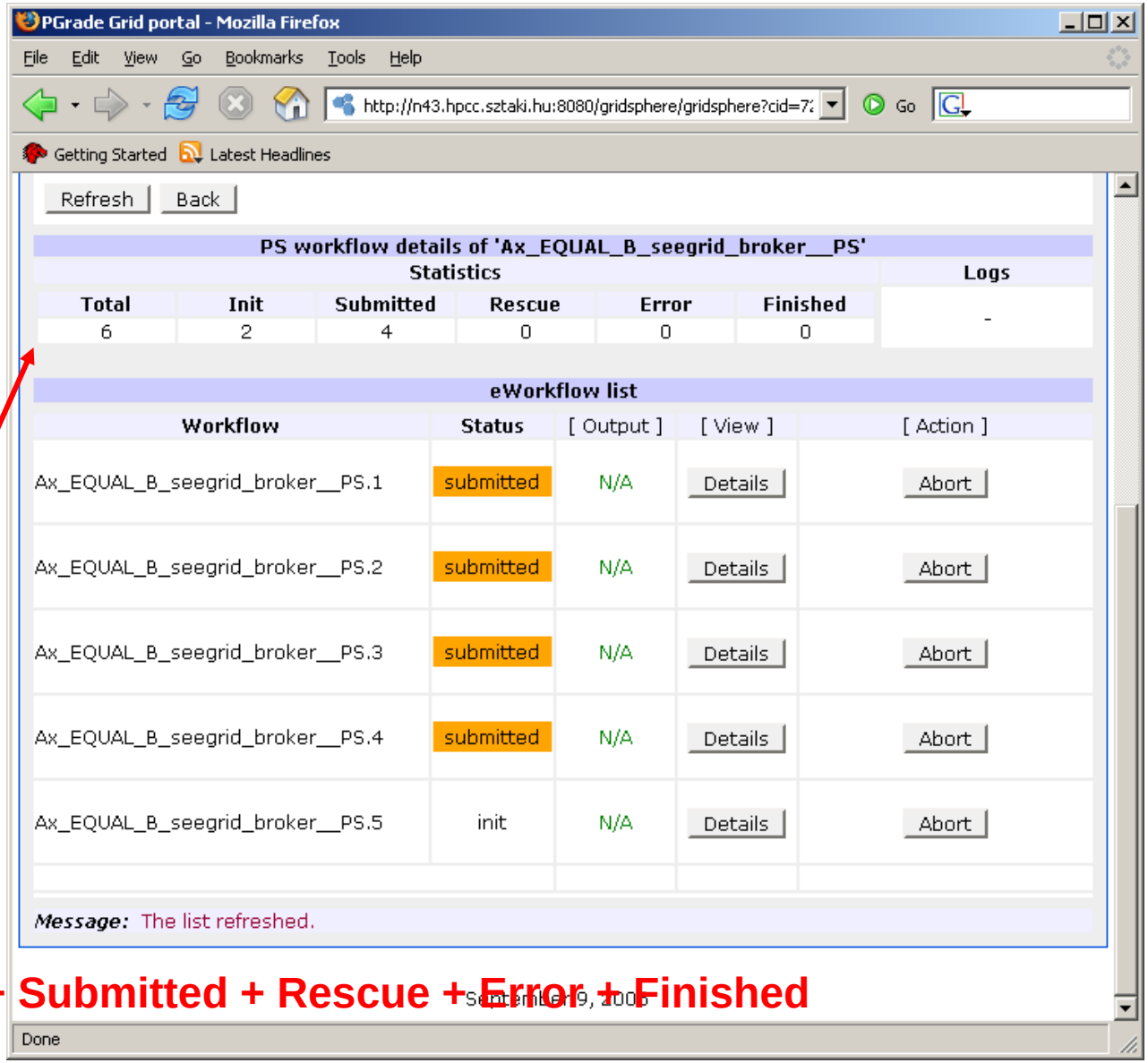
4. Define Subdirectory for the results



5. PS Workflow ready to Submit



Progress of submissions in PS detailed view



Total = Init + Submitted + Rescue + Error + Finished

Conclusions: E-scientists' concerns are resolved!

- The P-GRADE Portal hides the complexity and differences of Grids
  - Globus X – LCG2 – gLite Grid interoperability at the workflow level
  - Switching between Grid technologies will be transparent to the end-user
  - Various components can be integrated into large Grid applications
    - Sequential codes
    - MPI codes
    - Legacy codes (with the GEMLCA-specific P-GRADE Portal)
- You code does not have to include grid specific commands
- Graphical tools for application development, execution and monitoring
- Support for collaborative team work
  - Sharing workflows
  - Sharing jobs (components)
- Built by standard portlet API → customizable to specific application areas, user groups

How to learn the P-GRADE portal?

- Take a look at **www.lpds.sztaki.hu/pgportal**
  (manuals, slide shows, installation procedure, etc.)
- **Visit or request a training event!** (event list also on homepage)
  - Lectures, demos, hands-on tutorials, application development support

- **Get an account for one of its production installations:**
  - VOCE portal - SZTAKI
  - SEEGRID portal – SZTAKI
  - GILDA portal – SZTAKI
  - NGS portal – University of Westminster

- **If you are the administrator of a Grid/VO then contact SZTAKI to get your own P-GRADE Portal!**
- **If you know the administrator of a P-GRADE Portal you can ask him/her to give access to your Grid through his/her portal installation! (Multi-Grid portal)**

Learn once, use everywhere
Develop once, execute anywhere

Thank you!

www.lpds.sztaki.hu/pgportal
pgportal@lpds.sztaki.hu