Minimum intrusion Grid Introduction

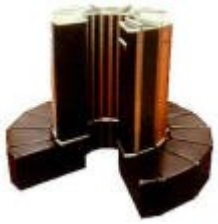
Jonas Bardino
University of Copenhagen
E-Science

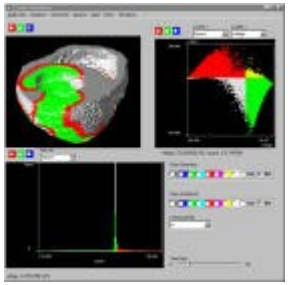
The Grid

- Named after the power-grid
- Sometimes referred to as the information power grid
- Like the power-grid GRID should be powered by large installations
  - not individual generators

Philosophy

- Current ’Internet’ mainly allows access to information
- The Grid should provide access to **any** desired resource
  - CPU/SuperComputers
  - Storage
  - Applications





Why is GRID needed

- Science goes virtual
  - A lot of areas turn to computer simulations
  - A wider range of experiments, and cheap!
- Computer power is consolidated in data centres
  - User maintenance can not be expected
  - Better utilization -> cheaper
  - Users still need easy access
  - Resource owners need access control

Problems found in the existing models? (not every problem in every middleware)

- Single point of failure
- Lack of scheduling
- Poor scalability
- No means of implementing privacy
- No means of utilizing ‘cycle-scavenging’
- Firewall dependency
- Highly bloated middleware

MiG Idea

- MiG should address all the previous concerns
- GRID should be a system
  - not just a protocol between sites
- By having a set of servers that **are** GRID we get full control of GRID
  - we upgrade all machines at the same time
- Users and resources do not have to maintain anything

MINIMUM INTRUSION GRID

The desired user view of MiG

- MiG should be a virtual computer to the user
- Files are stored within MiG – each user has his own home-directory
  - In job descriptions files are relative to the users home-directory
- We even support interactive applications
- Economically MiG should be fair

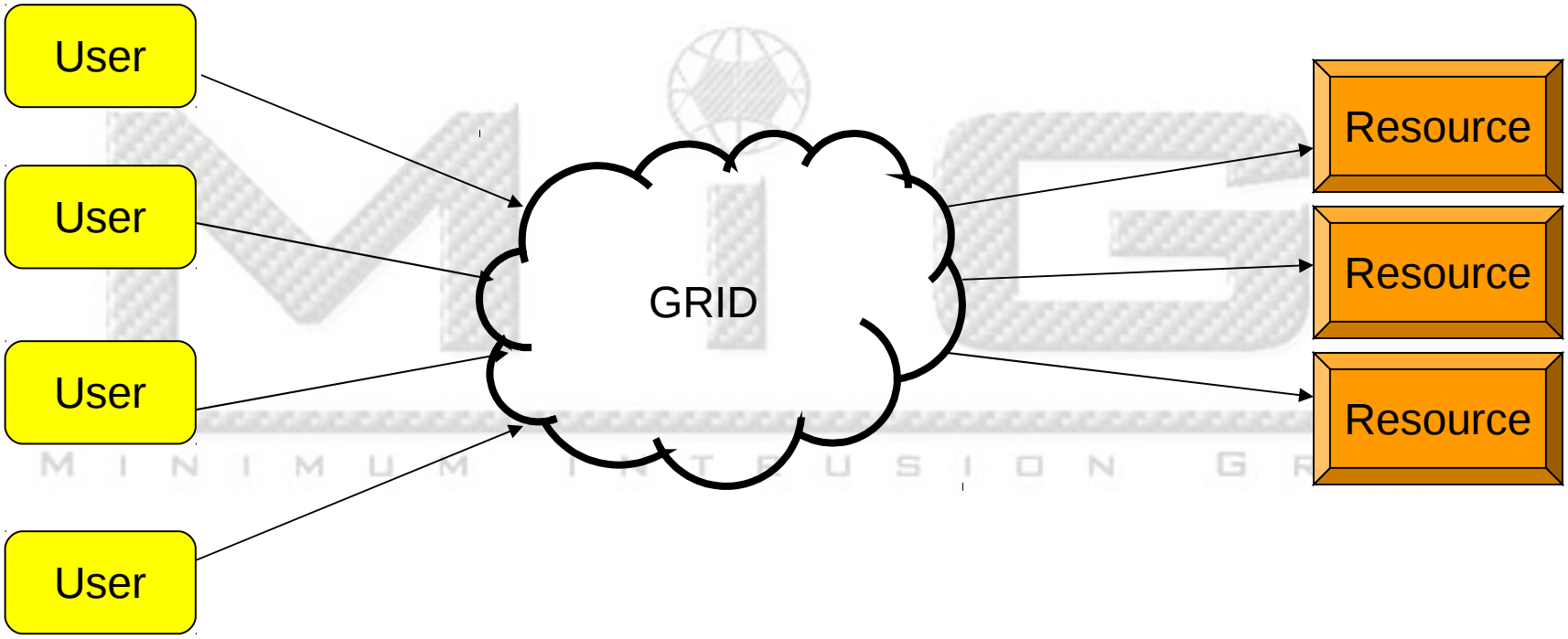
MINIMUM INTRUSION GRID

The desired resource view of MiG

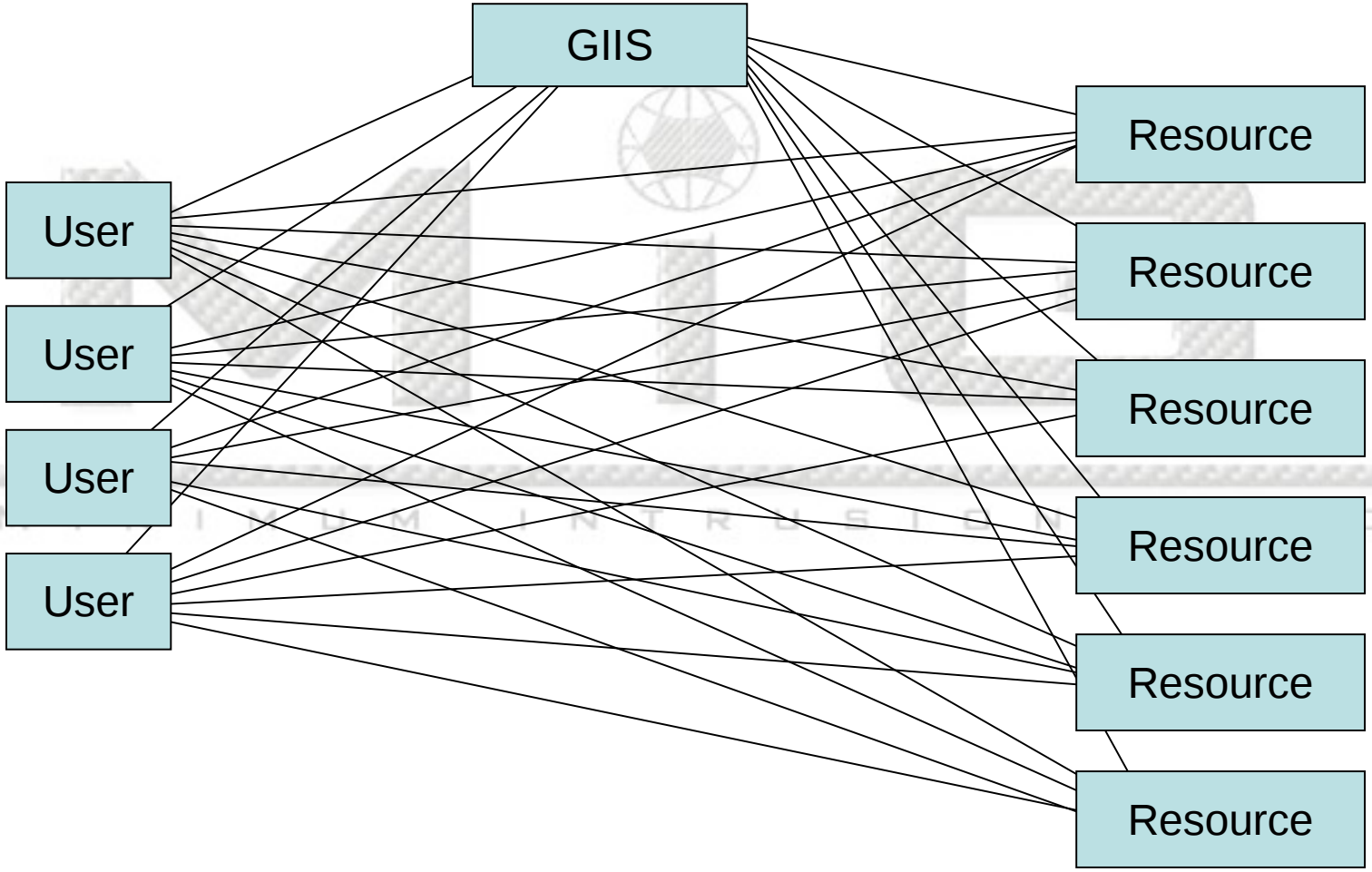
- MiG is just another user
  - or set of users
- Economically MiG should be fair

MINIMUM INTRUSION GRID

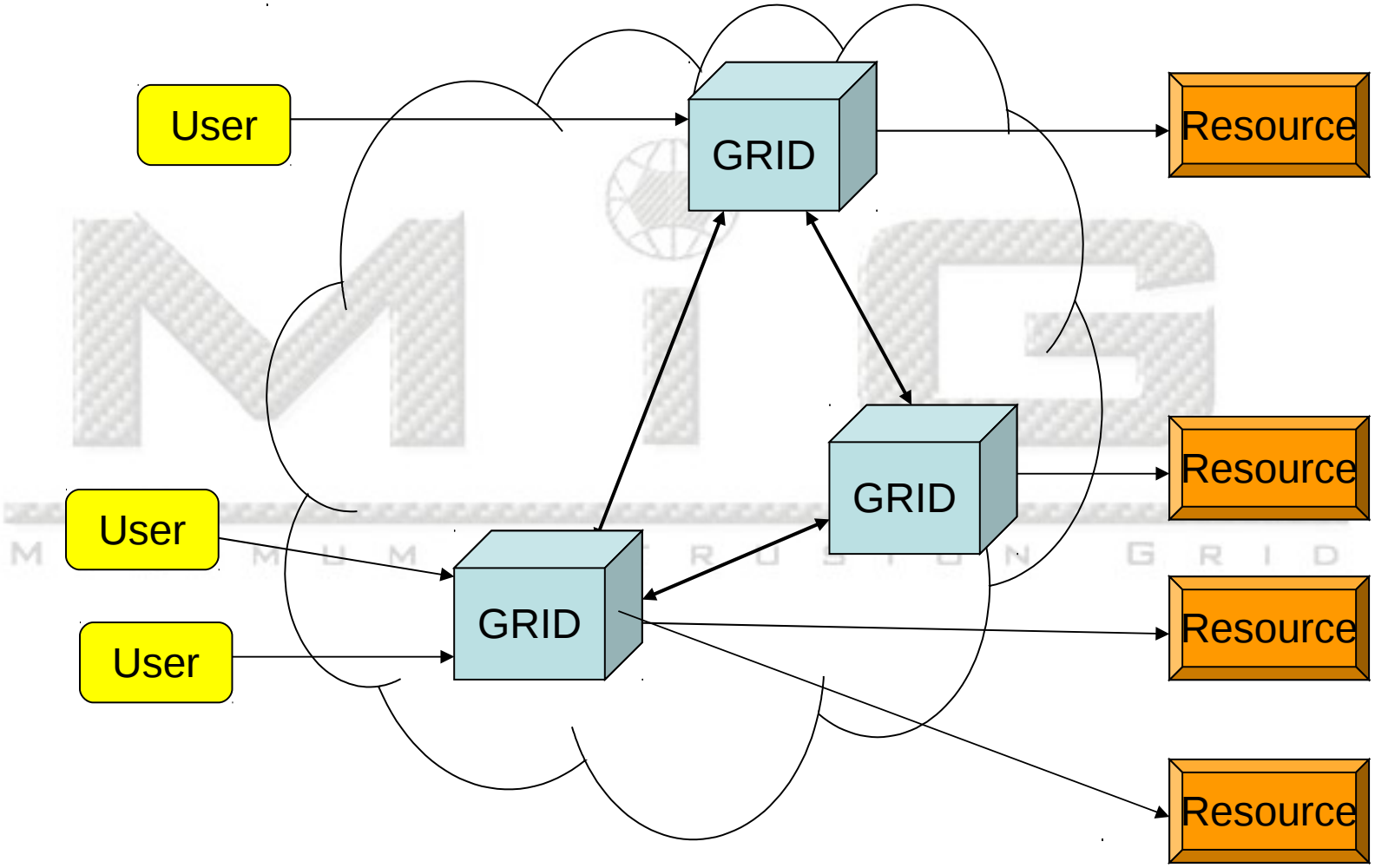
The abstract MiG model



Classic Grid



Minimum intrusion Grid



MiG Rules

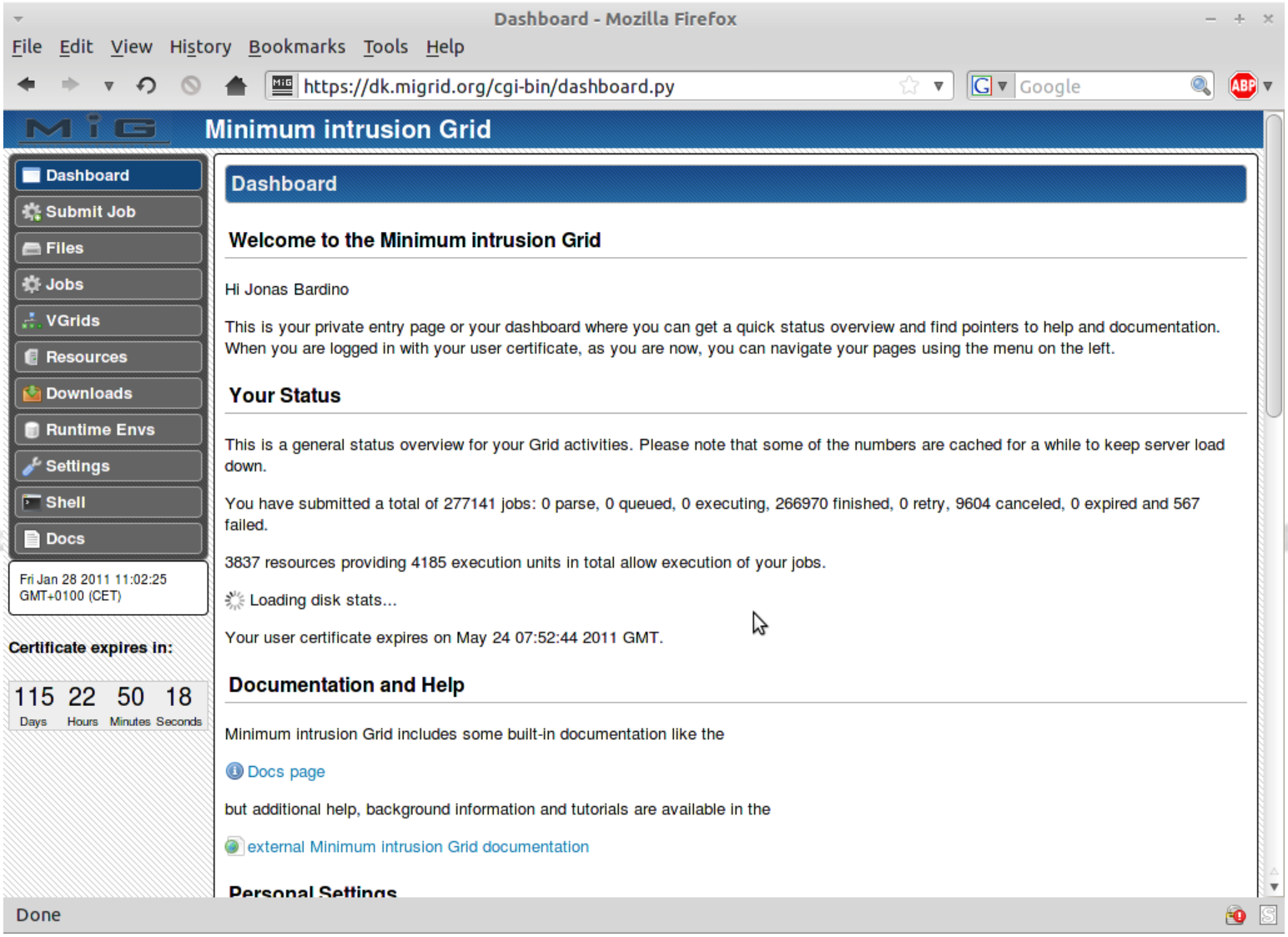
i. **Nothing** produced by MiG can be **required** to be installed on either the resource or client end
ii. Everything within MiG must be implemented in Python unless another language is absolutely required
iii. Any design and implementation decision must optimize towards transparency for the users
iv. Anything that is not right must be thrown away

MiG requirements - User

- Users should depend on **no** installed SW
  - A web-browser should be enough
- In addition to a browser the users need only a certificate
  - A scriptable user-interface is also available – but not required

MINIMUM INTRUSION GRID

User view of MiG



MiG requirements - Resource

- Resources should only create a MiG user
- As few requirements to the resource as possible
  - support ssh inbound
  - support https outbound
- The system administrator also needs a certificate to register the resource
- Sandboxed systems have different requirements

The desired user view of MiG

- MiG should be a virtual computer to the user
- Files are stored within MiG – each user has his own home-directory
  - In job descriptions files are relative to the users home-directory
- Interactive application support

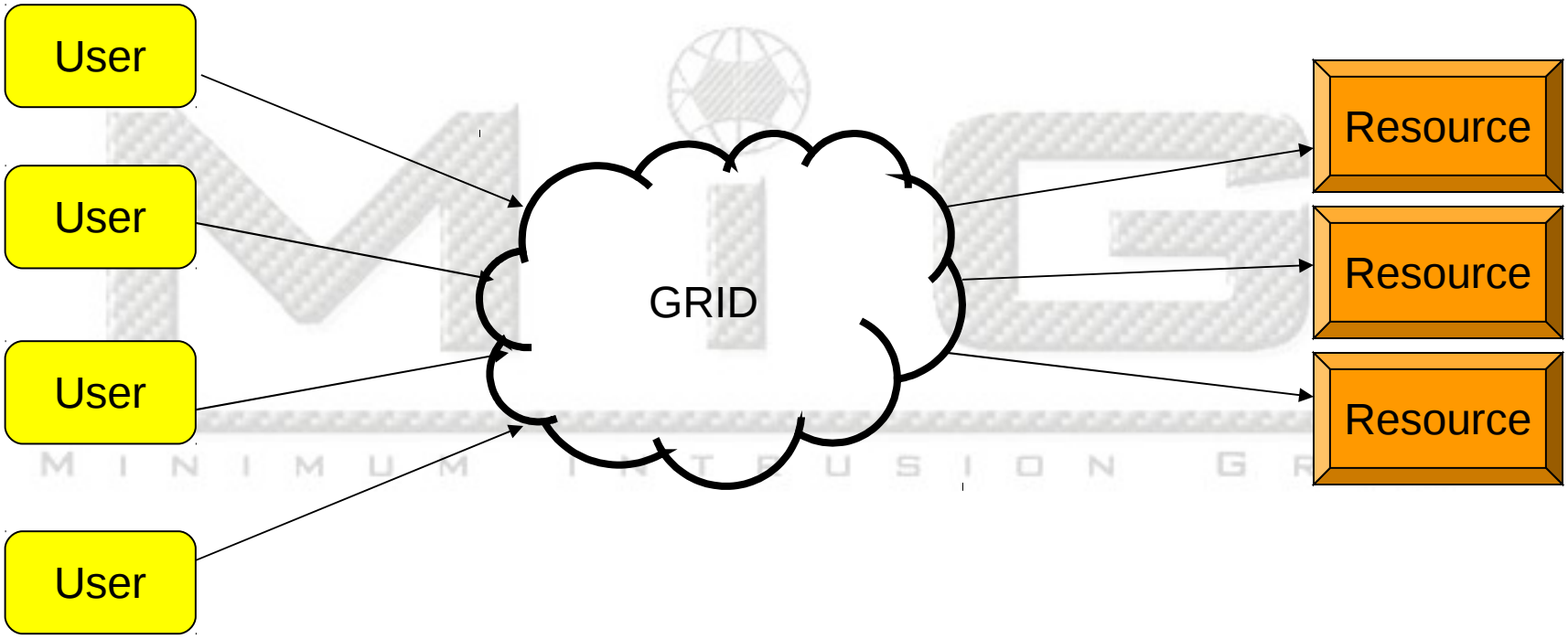
MINIMUM INTRUSION GRID

The desired resource view of MiG

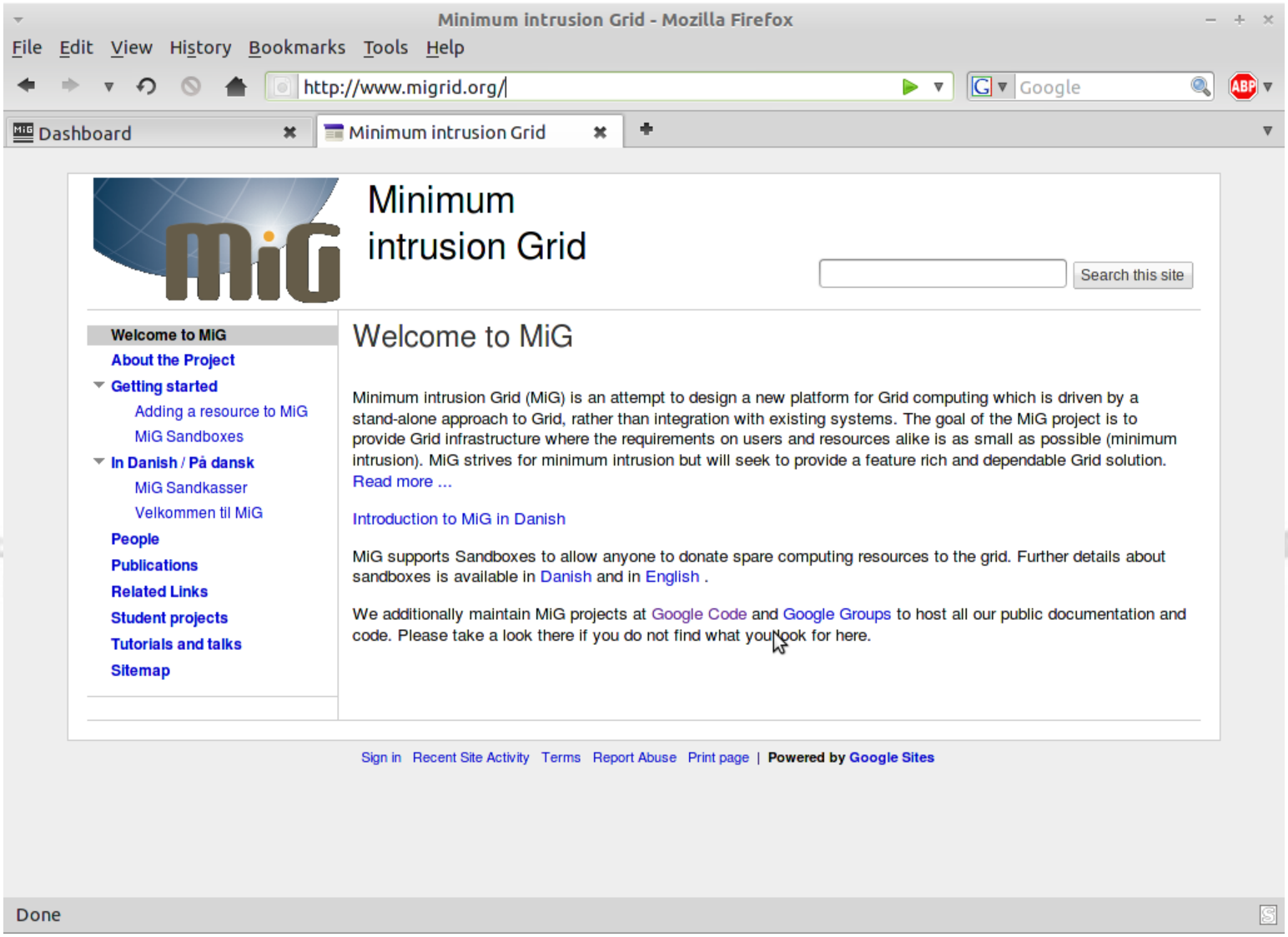
- MiG is just another user
  - or set of users
- Economically MiG should be fair

MINIMUM INTRUSION GRID

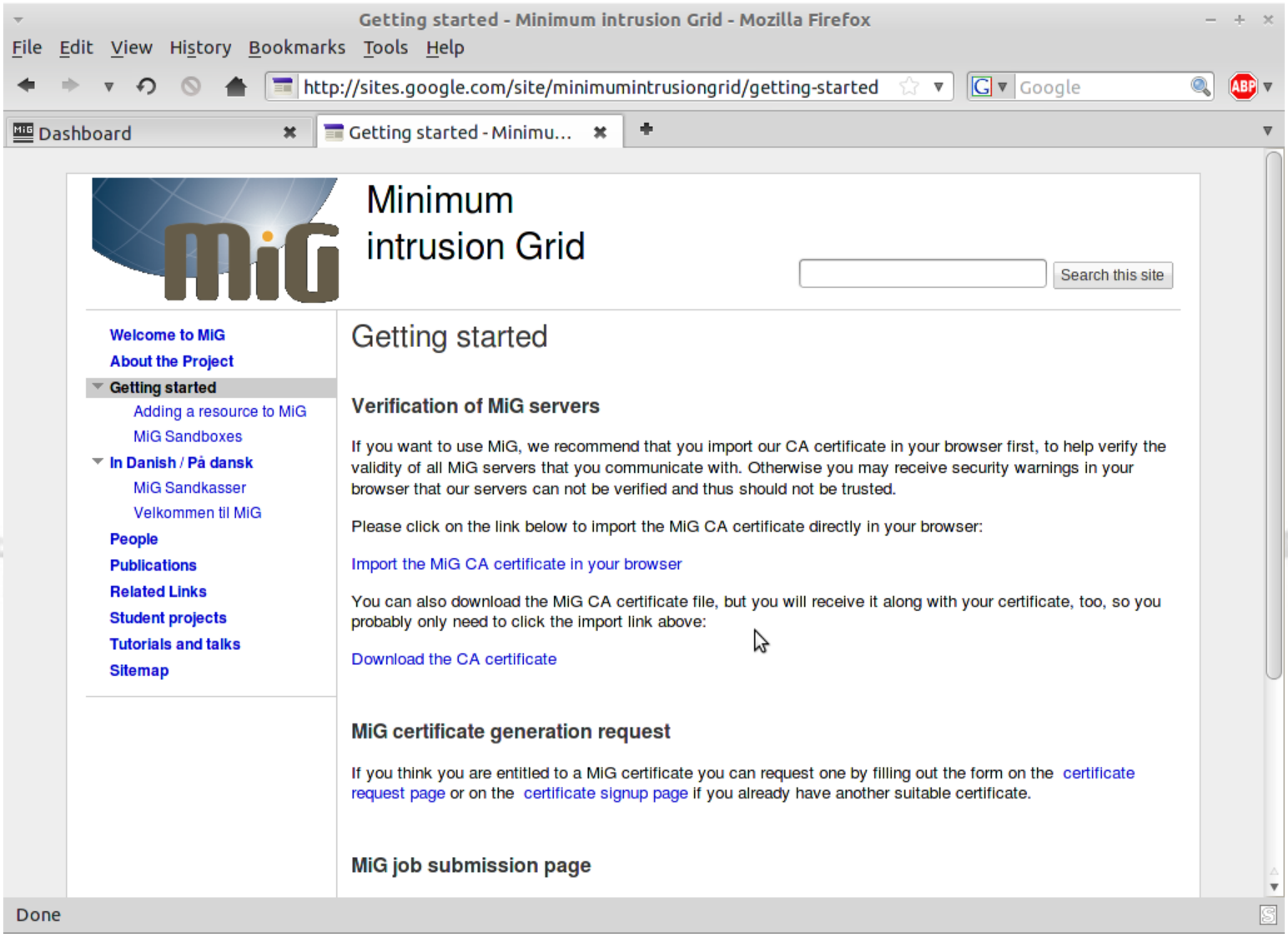
The abstract MiG model



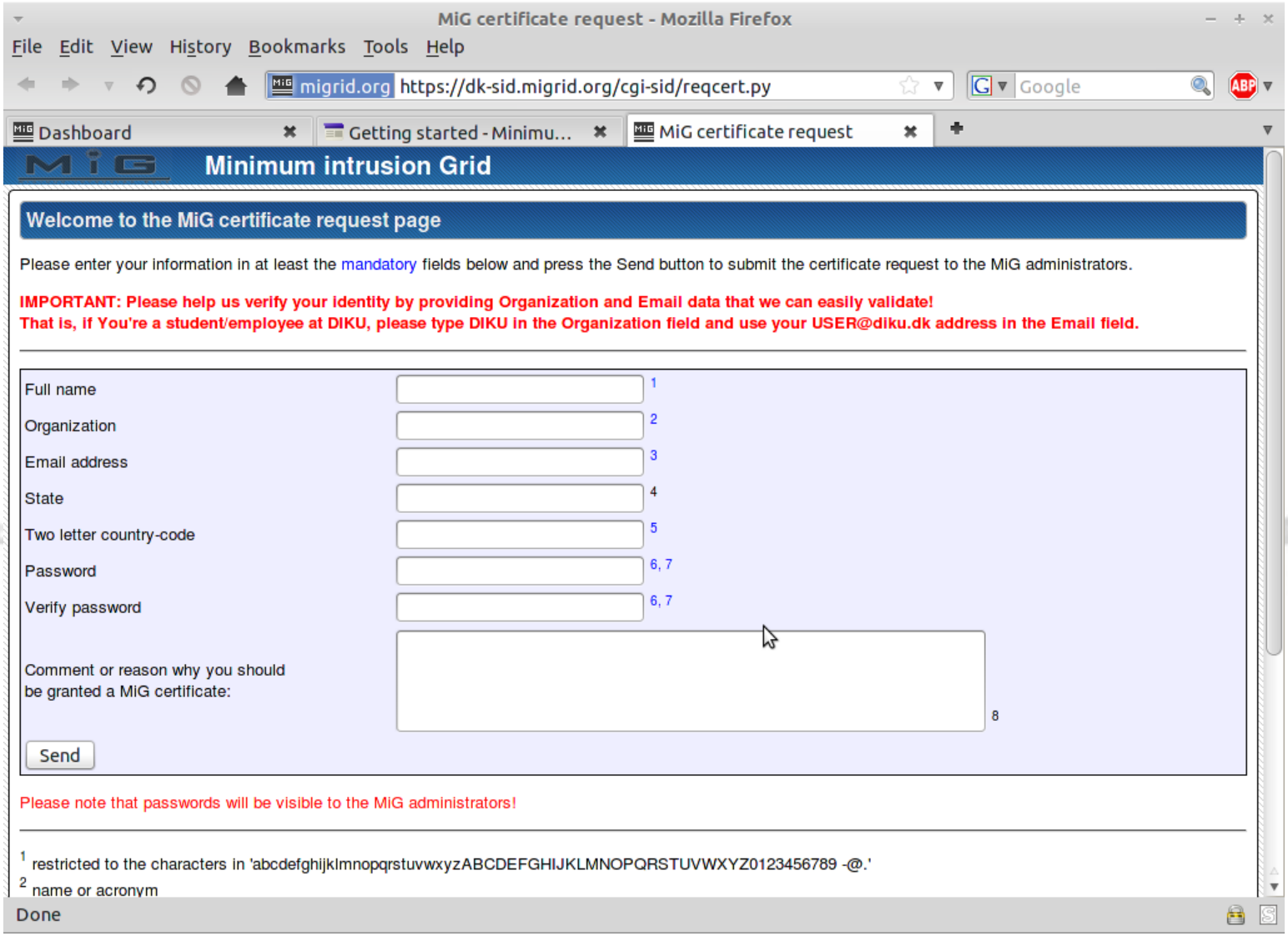
Project page



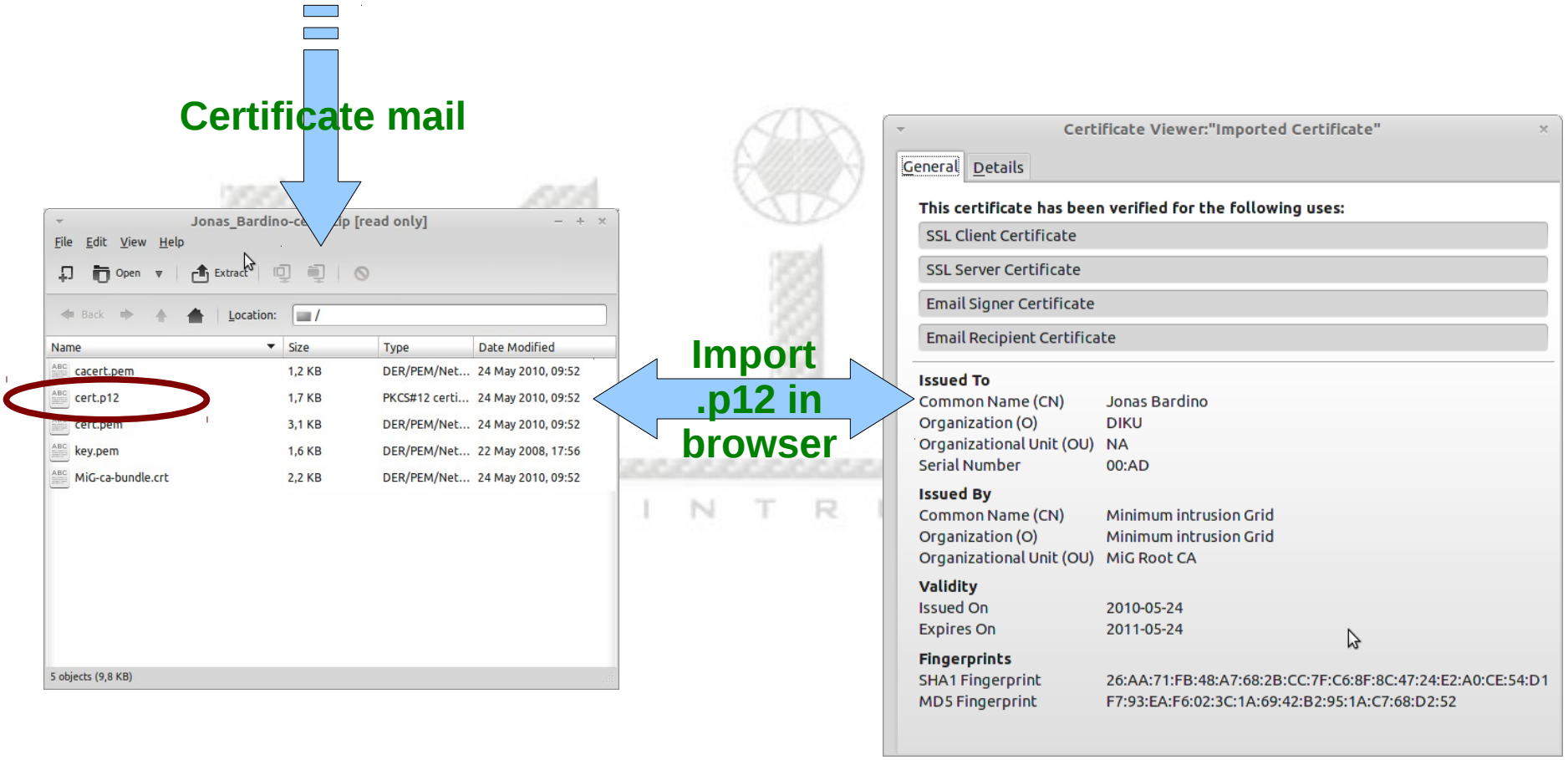
Getting started



Obtaining a certificate



Certificate

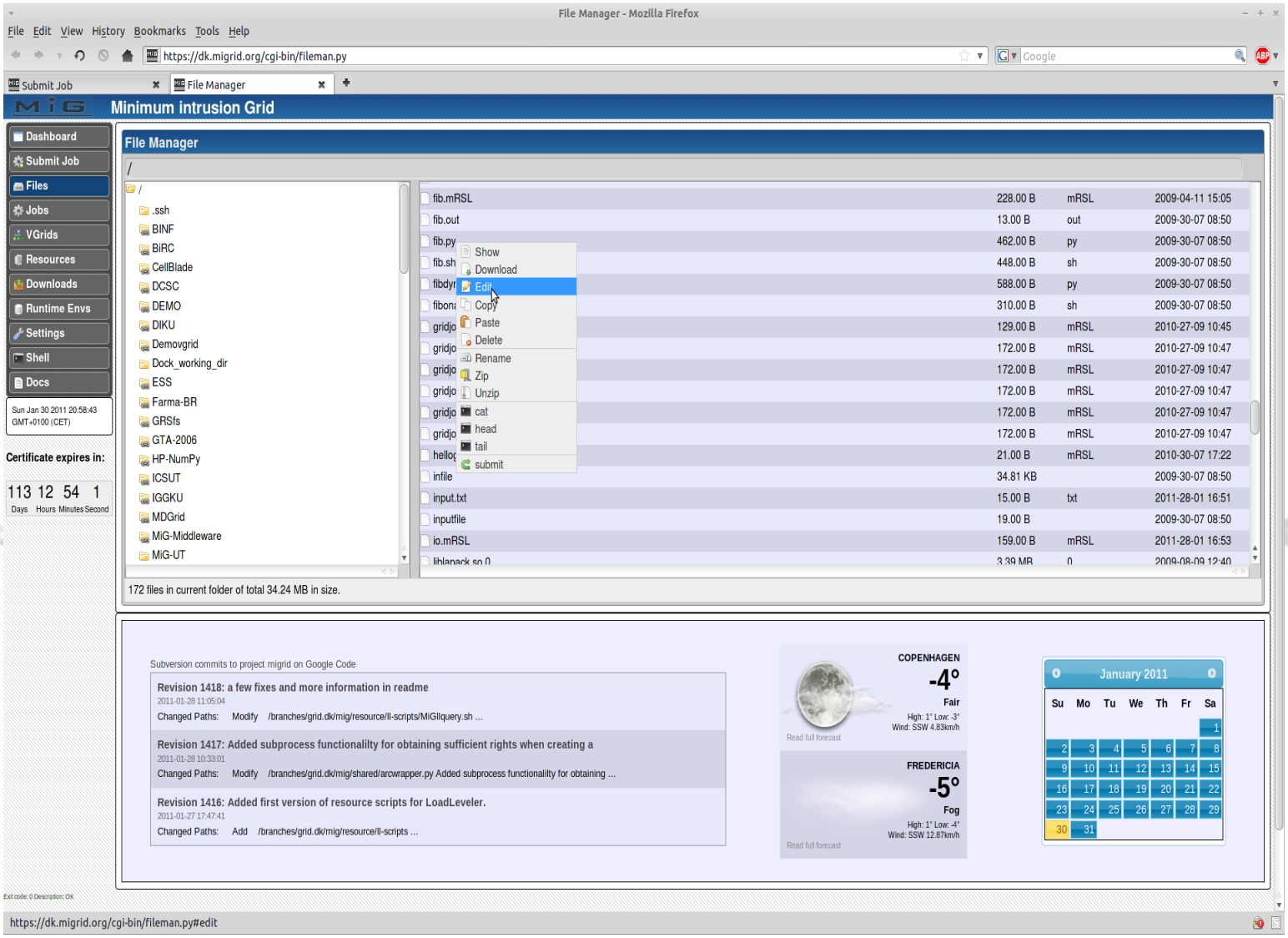


Files

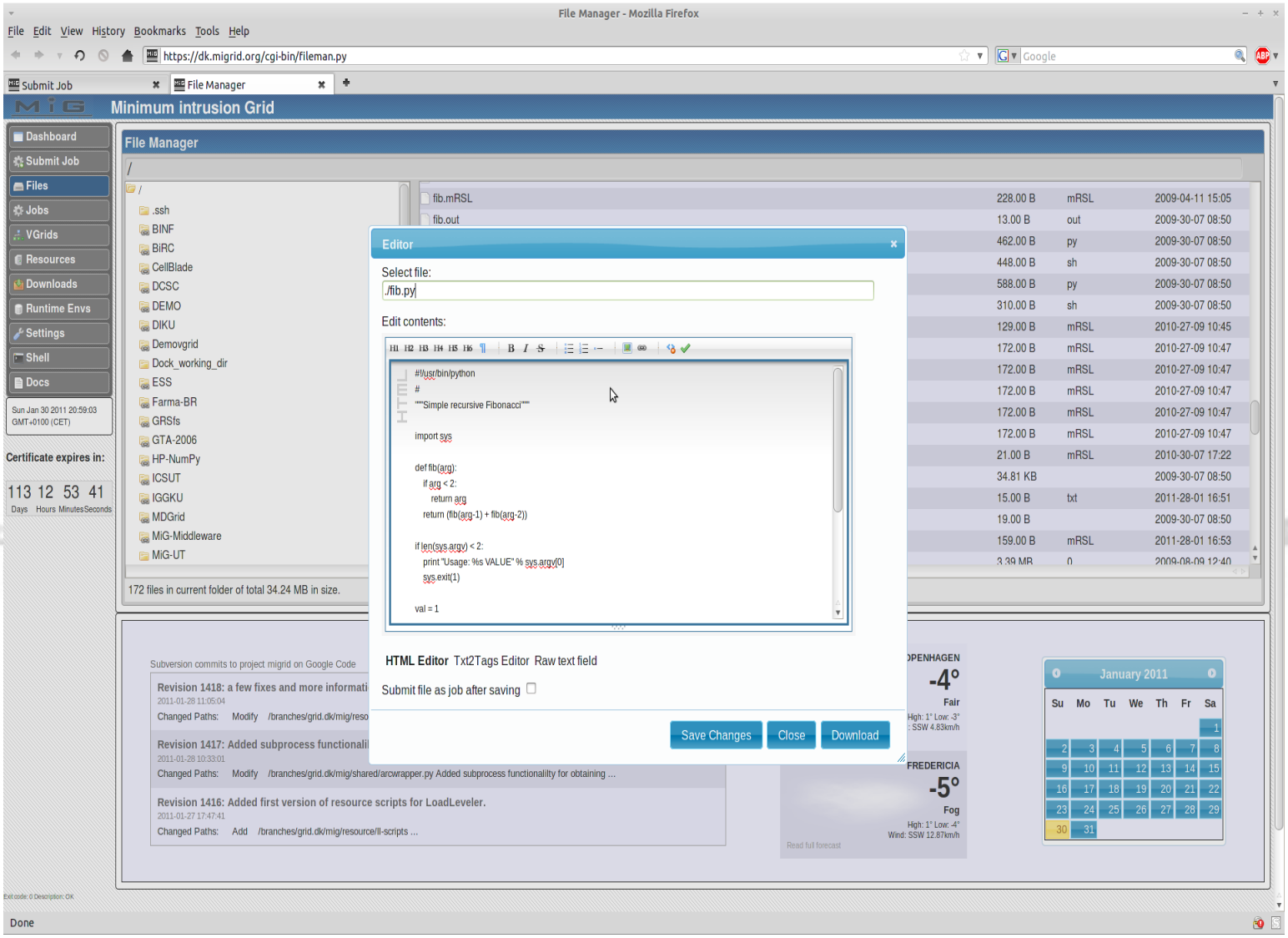
- Each user has a private home on MiG
- Works as a standard UNIX home dir
- GUI file manager and command line acces
- Includes shared dirs from virtual org's
- All file refs in MiG are relative to user home

MINIMUM INTRUSION GRID

File management



File editor


File Manager - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://dk.migrid.org/cgi-bin/fileman.py
Submit Job
File Manager
Minimum intrusion Grid
Dashboard
Submit Job
Files
Jobs
VGrids
Resources
Downloads
Runtime Envs
Settings
Shell
Docs
Sun Jan 30 2011 20:59:03 GMT+0100 (CET)
Certificate expires in:
113 12 53 41
Days Hours Minutes Seconds
File Manager
/
.ssh
BINF
BiRC
CellBlade
DCSC
DEMO
DIKU
Demovgrid
Dock_working_dir
ESS
Farma-BR
GRSfs
GTA-2006
HP-NumPy
ICSUT
IGGKU
MDGrid
MiG-Middleware
MiG-UT
fib.mRSL 228.00 B mRSL 2009-04-11 15:05
fib.out 13.00 B out 2009-30-07 08:50
462.00 B py 2009-30-07 08:50
448.00 B sh 2009-30-07 08:50
588.00 B py 2009-30-07 08:50
310.00 B sh 2009-30-07 08:50
129.00 B mRSL 2010-27-09 10:45
172.00 B mRSL 2010-27-09 10:47
172.00 B mRSL 2010-27-09 10:47
172.00 B mRSL 2010-27-09 10:47
172.00 B mRSL 2010-27-09 10:47
172.00 B mRSL 2010-27-09 10:47
21.00 B mRSL 2010-30-07 17:22
34.81 KB 2009-30-07 08:50
15.00 B txt 2011-28-01 16:51
19.00 B 2009-30-07 08:50
159.00 B mRSL 2011-28-01 16:53
172 files in current folder of total 34.24 MB in size.
Editor
Select file:
./fib.py
Edit contents:
#!/usr/bin/python
#
"""Simple recursive Fibonacci"""
import sys
def fib(arg):
if arg < 2:
return arg
return (fib(arg-1) + fib(arg-2))
if len(sys.argv) < 2:
print "Usage: %s VALUE" % sys.argv[0]
sys.exit(1)
val = 1
HTML Editor Txt2Tags Editor Raw text field
Submit file as job after saving
Save Changes
Close
Download
Subversion commits to project migrid on Google Code
Revision 1418: a few fixes and more informati
2011-01-28 11:05:04
Changed Paths: Modify /branches/grid.dk/mig/reso
Revision 1417: Added subprocess functionali
2011-01-28 10:33:01
Changed Paths: Modify /branches/grid.dk/mig/shared/arcwrapper.py Added subprocess functionality for obtaining ...
Revision 1416: Added first version of resource scripts for LoadLeveler.
2011-01-27 17:47:41
Changed Paths: Add /branches/grid.dk/mig/resource/ll-scripts ...
OPENHAGEN
-4°
Fair
High: 1° Low: -3°
SSW 4.83km/h
FREDERICIA
-5°
Fog
High: 1° Low: -4°
Wind: SSW 12.87km/h
Read full forecast
January 2011
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Done

Jobs

- The core of grid applications is jobs
- Typically execution of batch command(s)
- Optionally working on user files
- Jobs are sent to MiG queue
  - Execution when a suitable resource is ready
  - Job handling is done by MiG
  - User may override jobs: hold, cancel, resubmit
- Advanced apps can wrap jobs

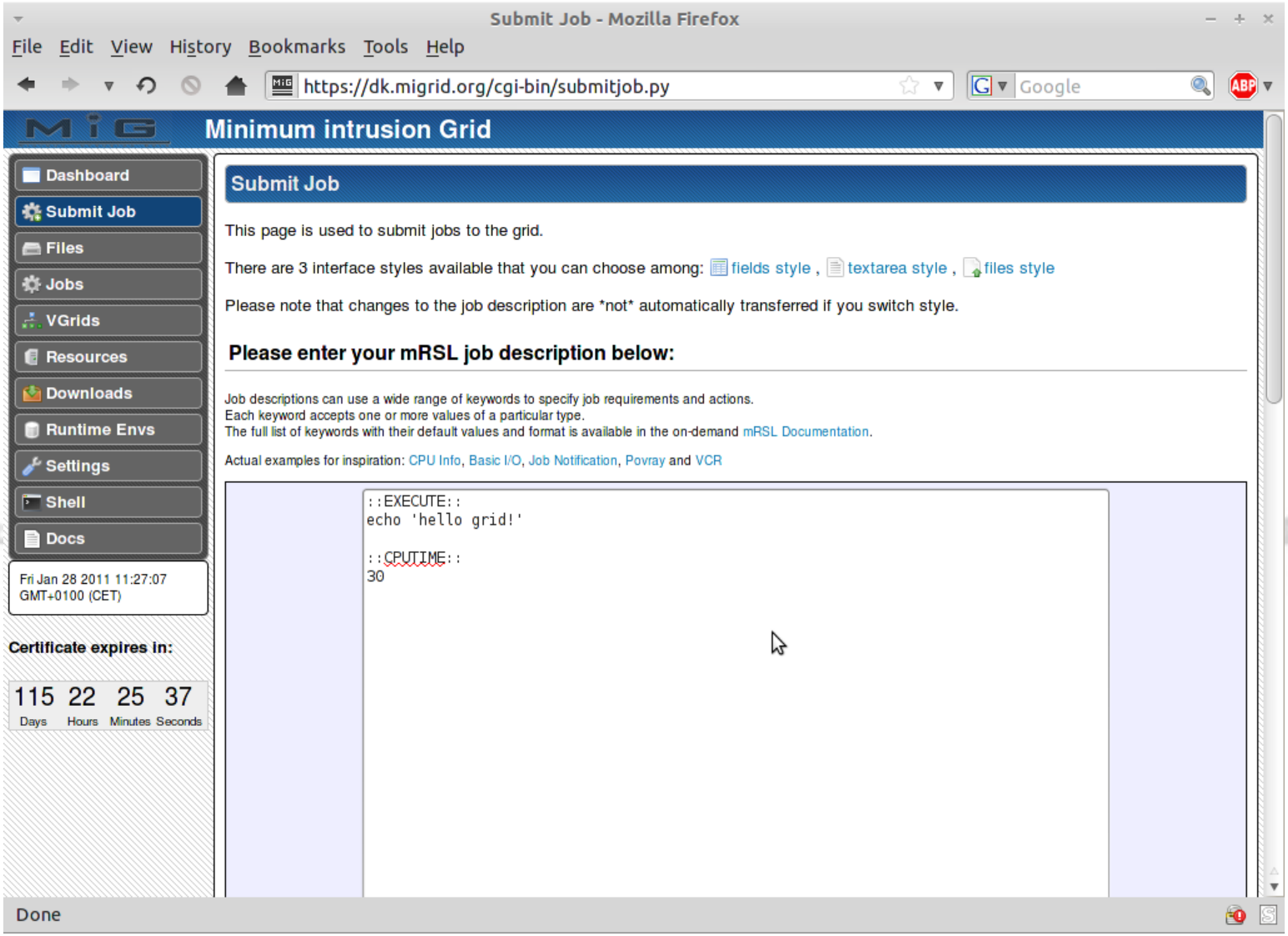
Your first job (mRSL)

- Jobs are executed in a standard UNIX shell environment.

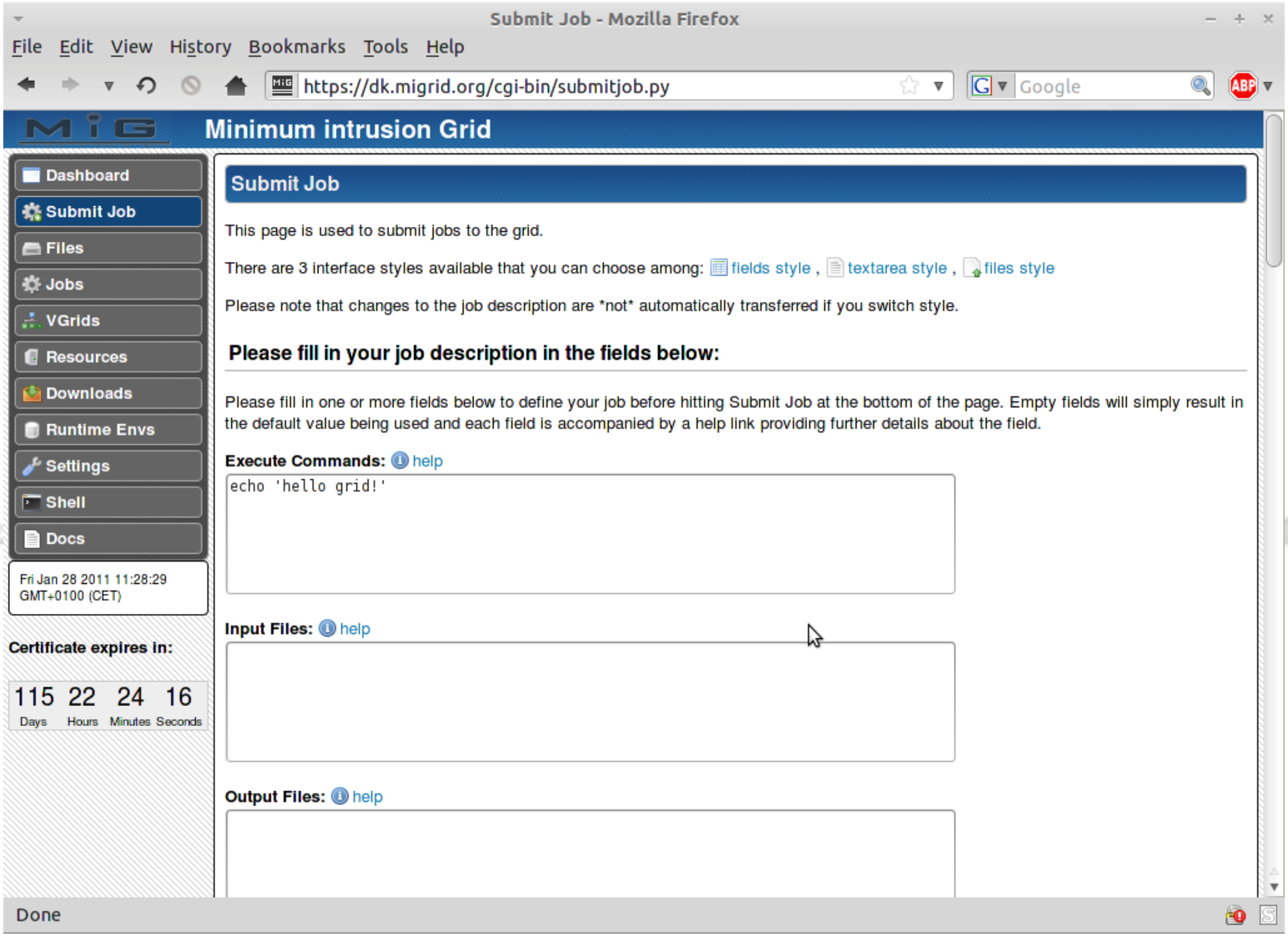
```
::EXECUTE::
echo 'hello grid!'

::CPUTIME::
30
```

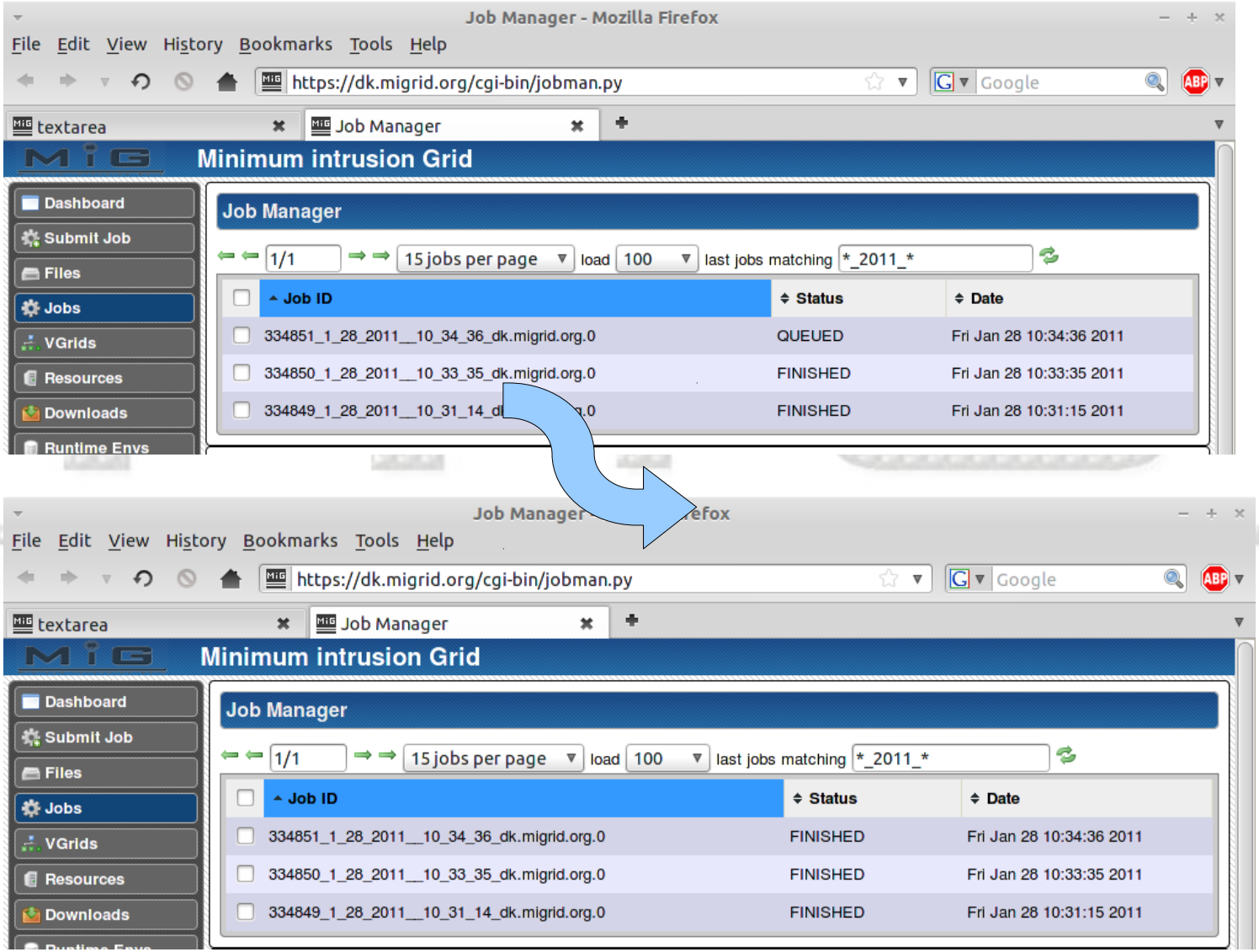
Your first job (submit text)



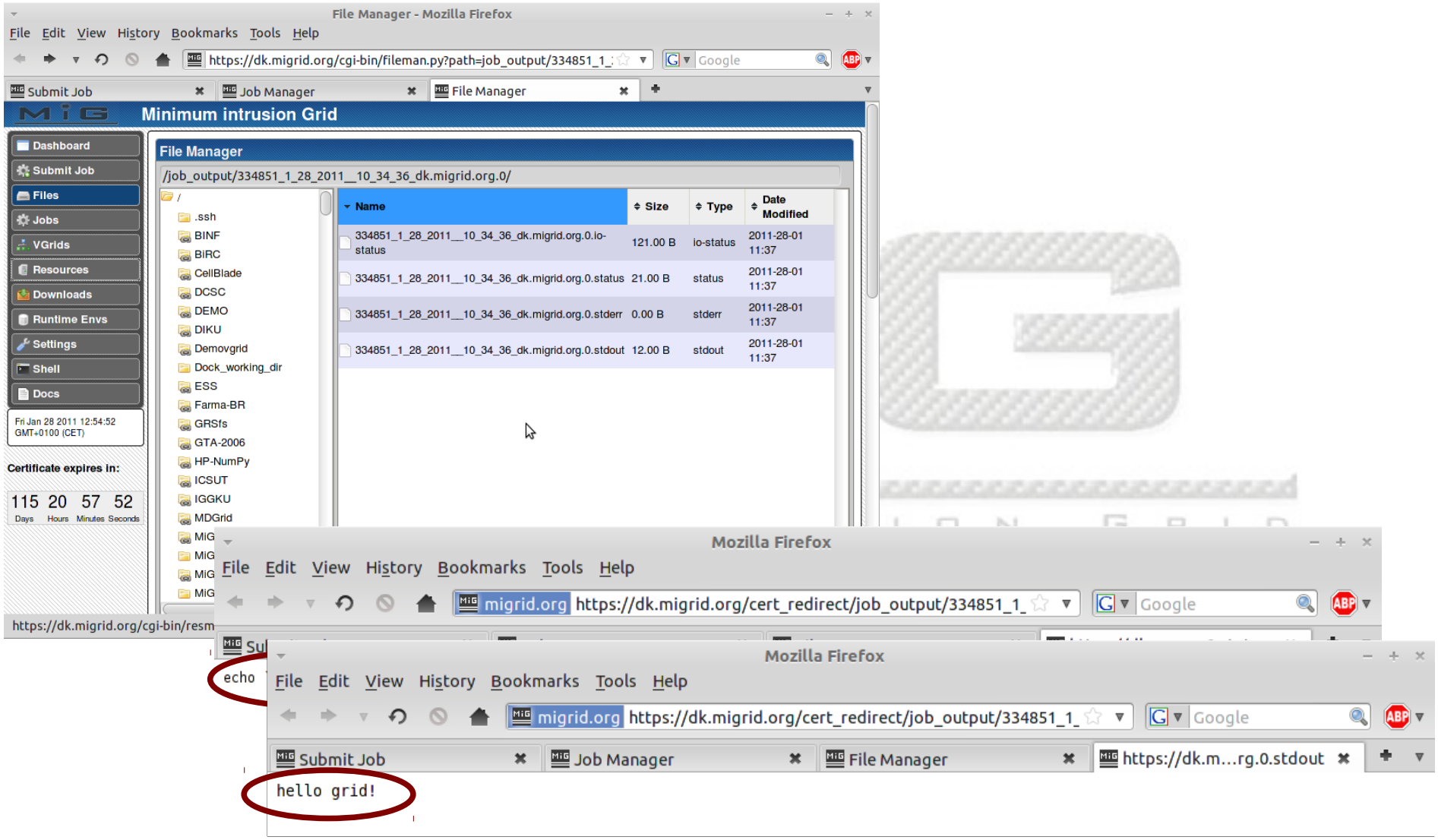
Alternative (submit fields)



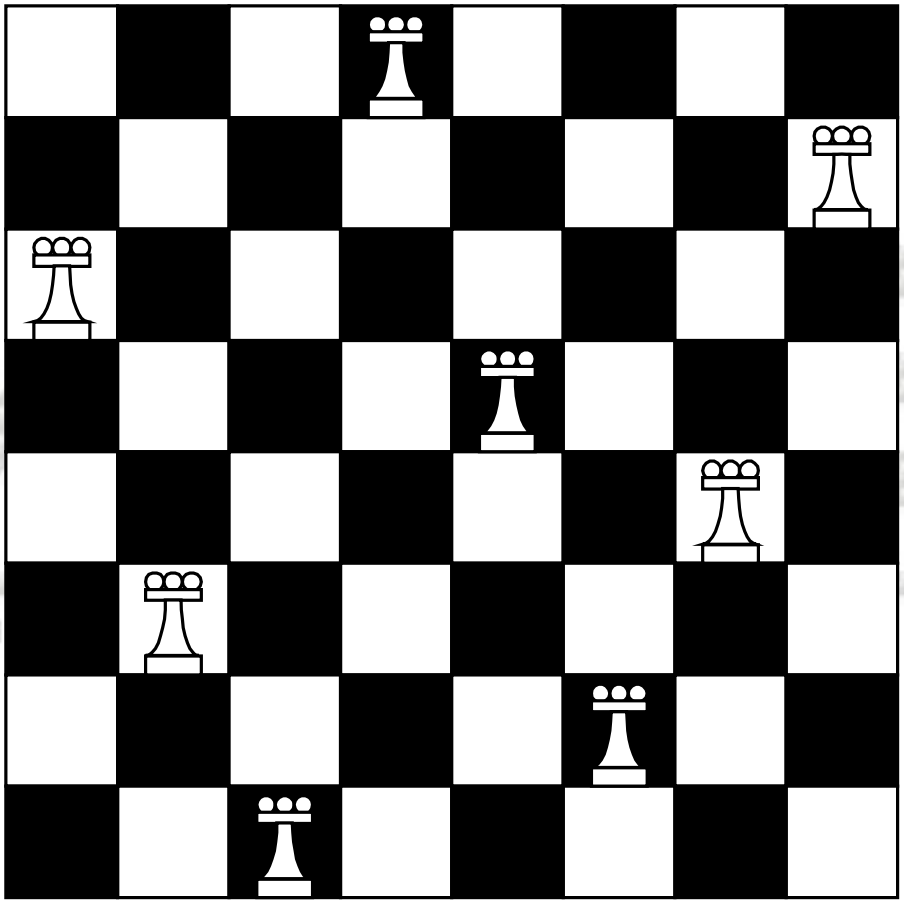
Your first job (status)



Your first job (result)



Your first real job (Problem)



http://dk.migrid.org/public/examples/NQueens.zip

Your first real job (code)

```
class Queens {

  public int n=8;
  public int count = 0;
  int [] all;

  boolean legal(int i){
    for(int x=0;x<i;x++) {
      if(all[i]==all[x]) return(false);
      if(all[x]-(i-x)==all[i]) return(false);
      if(all[x]+(i-x)==all[i]) return(false);
    }
    return(true);
  }

  void make_boards(int i){
    for(int y=0;y<n;y++){
      all[i]=y;
      if(legal(i)){
        if(i+1==n) count++;
        else make_boards(i+1);
      }
    }
  }
// ...continued on the right
```

```
  Queens(int n){
    this.n=n;
    all=new int[n];
    make_boards(0);
  }
}

public class Nqueens_seq {
  public static void main(String args[]){
    int n=8;
    if(args.length>0) n=new Integer(args[0]).intValue();
    System.out.println(new Queens(n).count+" with "+n+" queens");
  }
}
```

M U M I N T R U S I O N G R I D

Your first real job (job)

::EXECUTE::
java Nqueens_seq 15

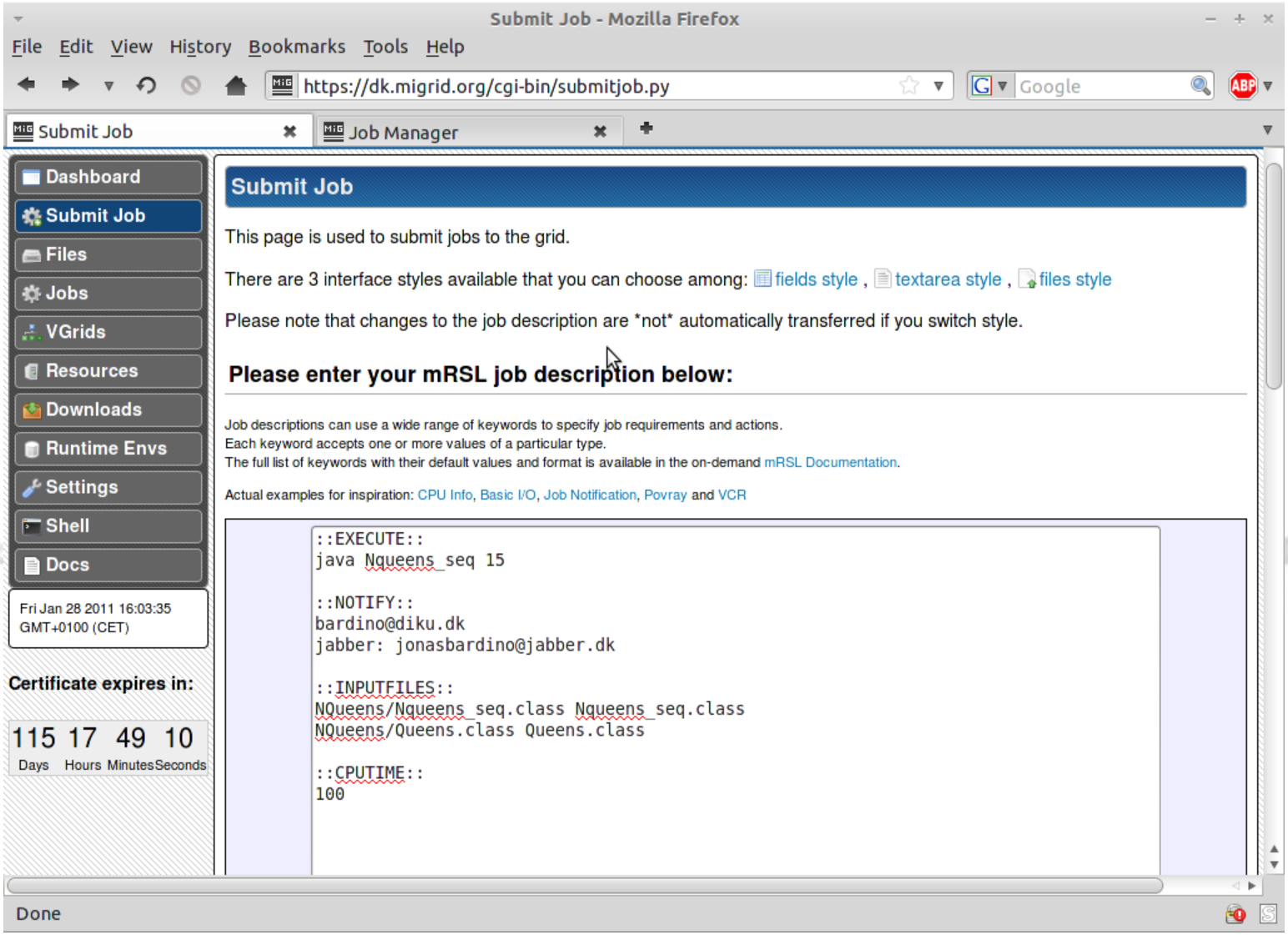
::NOTIFY::
bardino@diku.dk
jabber: jonasbardino@jabber.dk

Please modify to your own preferred notification type(s) and address(es) :-)

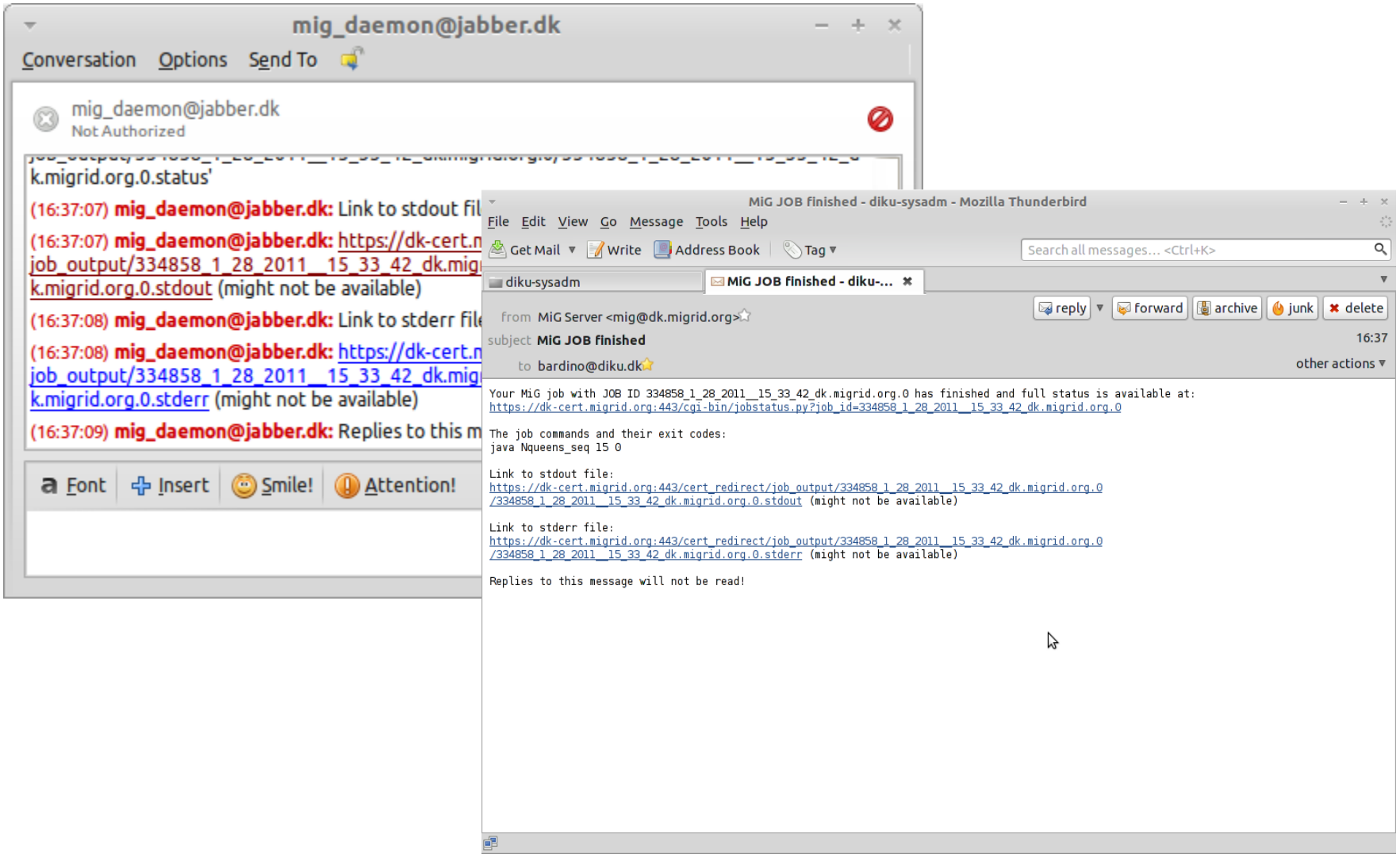
::INPUTFILES::
NQueens/Nqueens_seq.class Nqueens_seq.class
NQueens/Queens.class Queens.class

::CPUTIME::
100

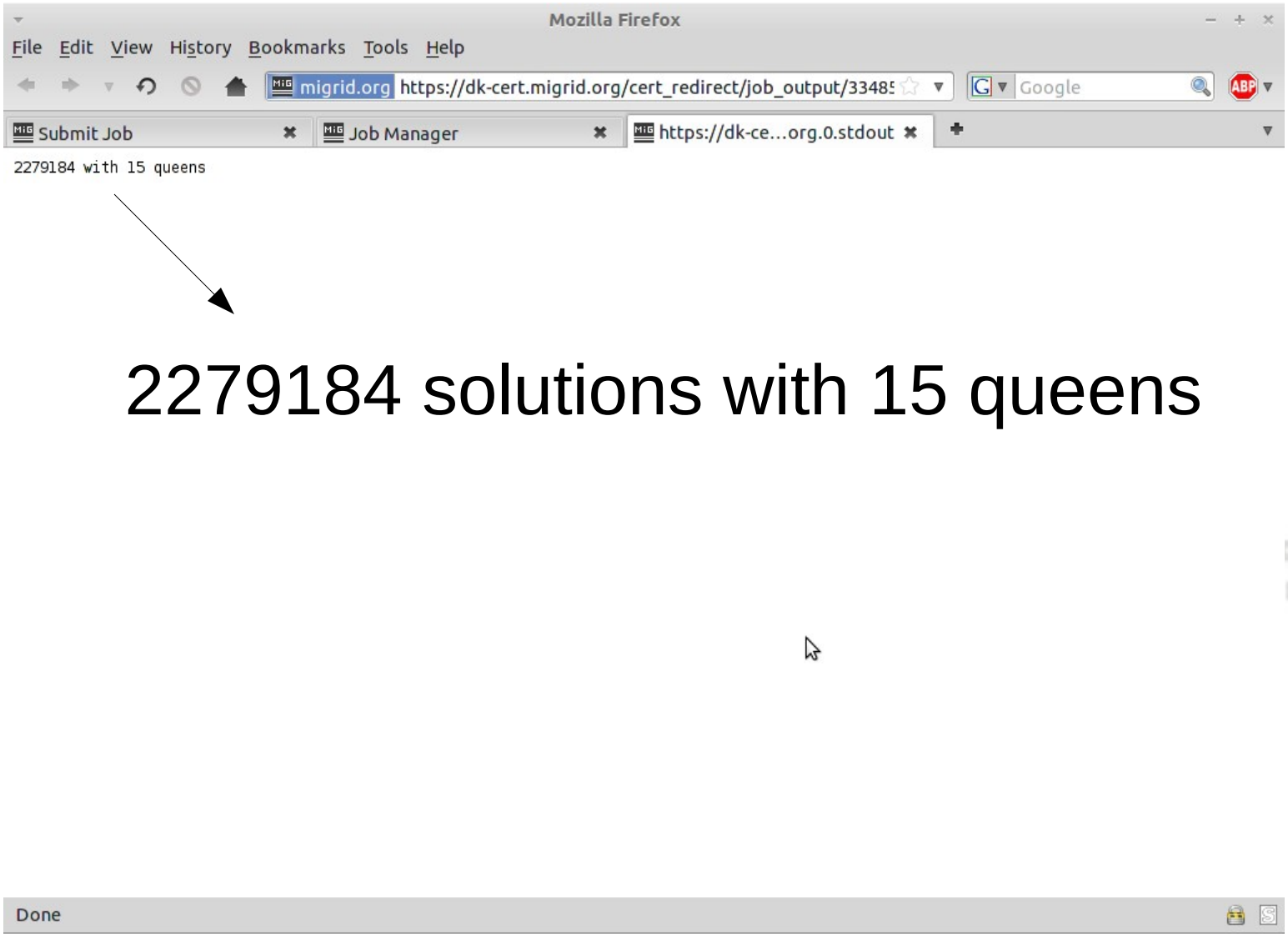
Your first real job (submit)



Your first real job (completed)



Your first real job (result)



2279184 solutions with 15 queens

Your first real job (assumptions)

- implicit resource assumptions
  - Java VM
  - CPU speed vs CPU time
  - memory, disk requirements
  - ...
- more on runtime environments later

MINIMUM INTRUSION GRID

Your first real job (optimizations)

- What did we do
  - Create all possible boards
  - Test each one sequentially
  - N=15 takes a while and run times grow fast
- How could we improve run time on Grid?
  - Symmetry?
  - Parallel jobs?
- With optimizations and Grid we may be able to break the current world record!

MINIMUM INTRUSION GRID

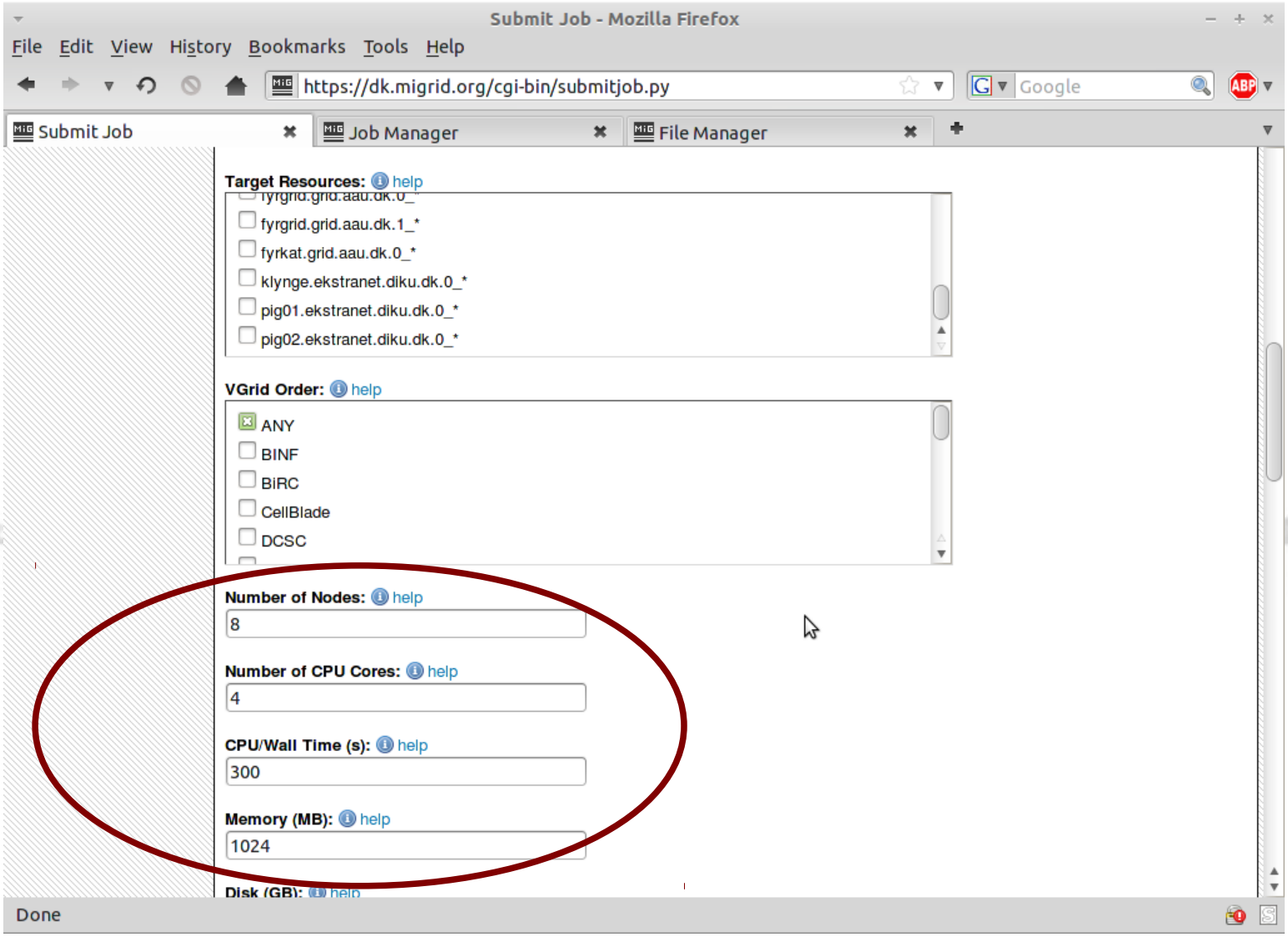
Jobs with specific requirements

- Architecture
  - In mRSL description

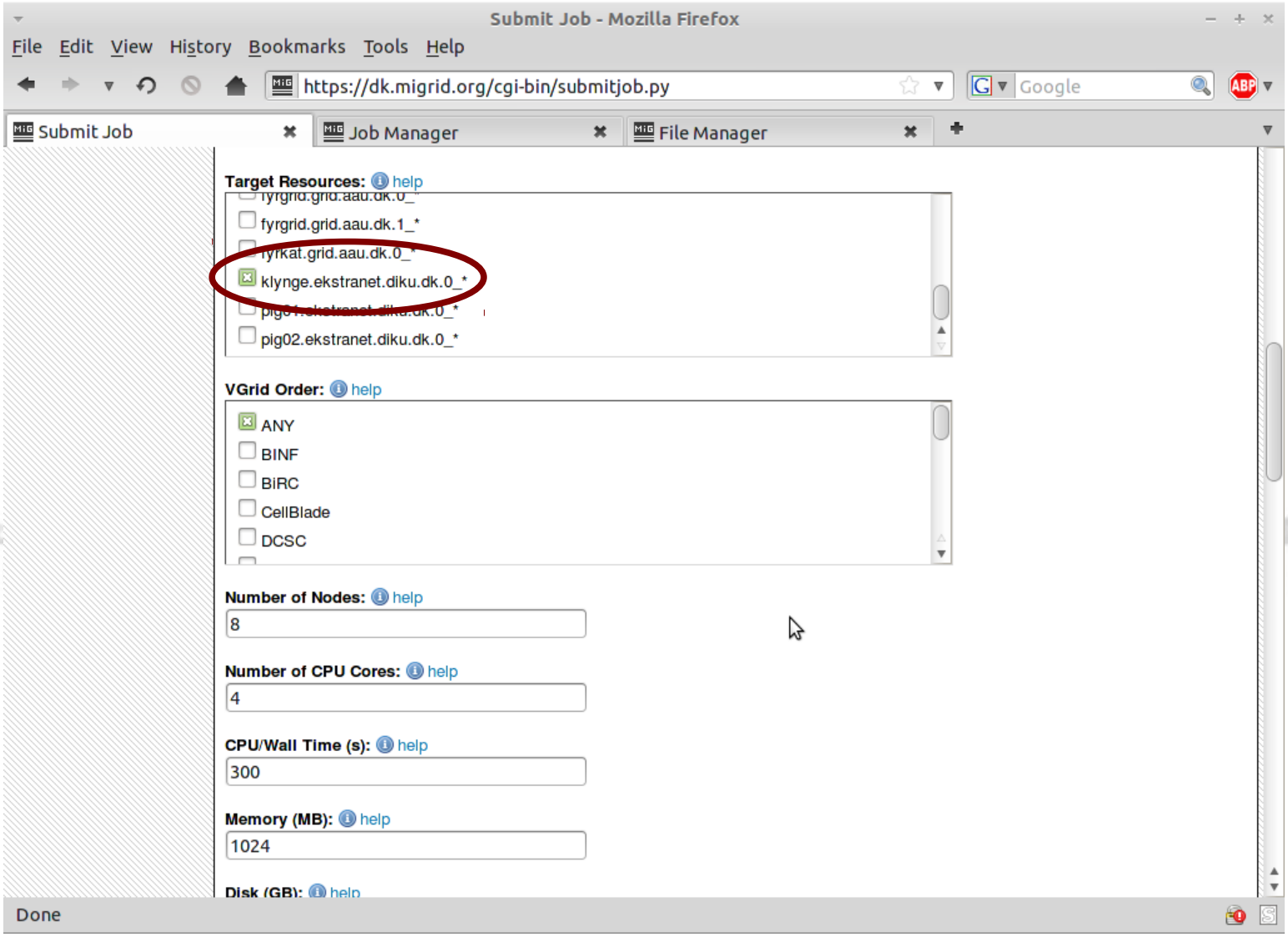
    ::ARCHITECTURE::

    <architecture>
  - or by selecting in drop down box in fields interface
  - x86, AMD64, PS3, ...
- No. of cluster nodes (NODECOUNT)
- No. of CPU's per node (CPUCOUNT)
- MB of memory (MEMORY)
- GB of disk space (DISK)
- VGrid order (VGRID)
- ...

Suitable cluster job (fields)



Specific cluster job (fields)

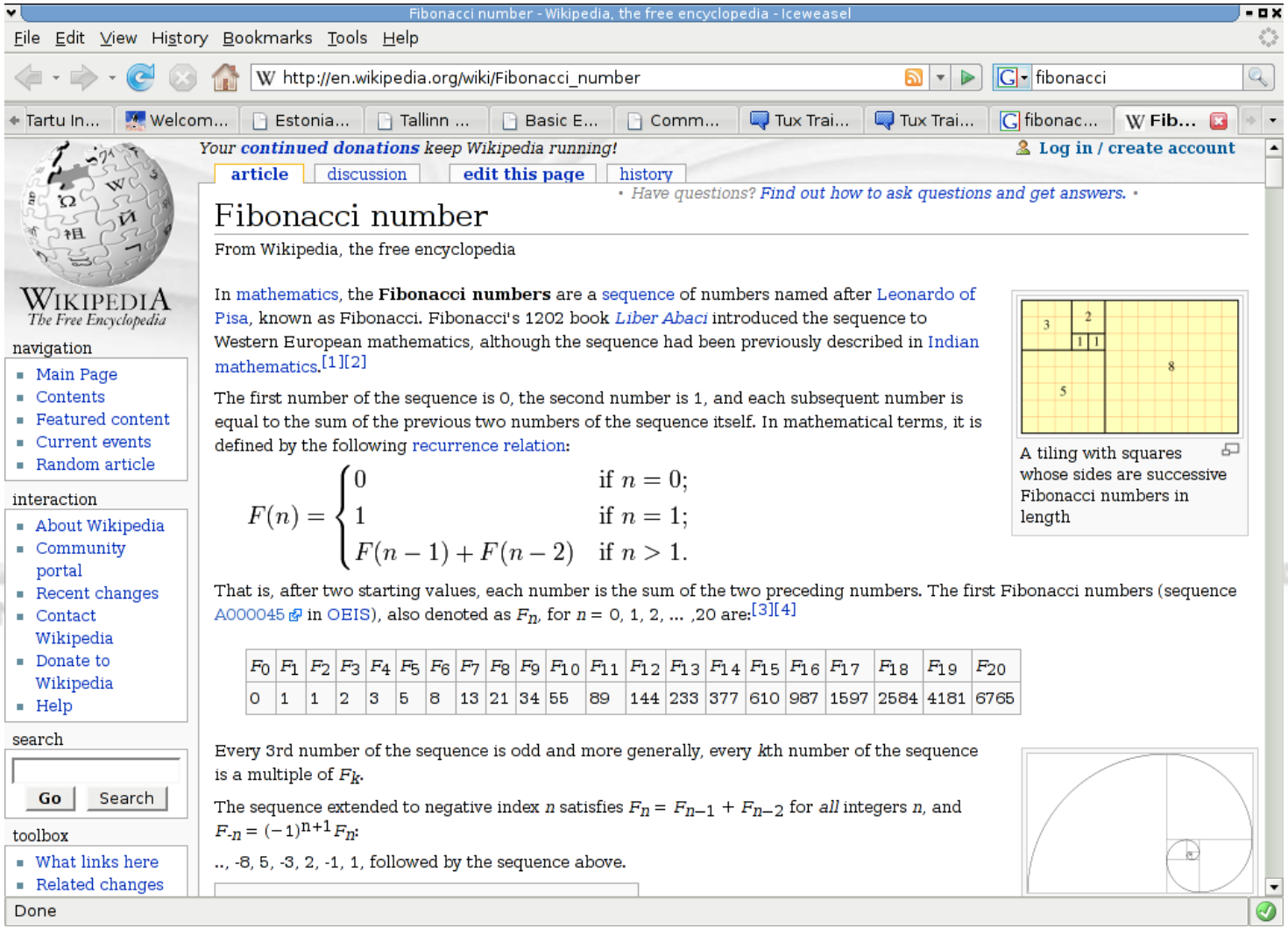


Runtime Environments

- Resource and User contract
  - without them almost nothing is guaranteed
  - used to negotiate apps and libs in a generic fashion
  - anyone can define them, but only resource owners can specify which ones they implement and how
  - no spec update is possible, always create new
- Examples
  - PYTHON-2 (any Python 2.X interpreter)
    - resource owner specifies PYTHON=/path/to/python
    - user knows that $PYTHON contains actual path in job
  - Java support
  - Local files, devices or markers

MINIMUM INTRUSION GRID

Runtime environments (Task)



http://dk.migrid.org/public/examples/SimpleJobs.zip

Runtime Environments (job)

::EXECUTE::
$PYTHON fib.py 26

::EXECUTABLES::
fib.py

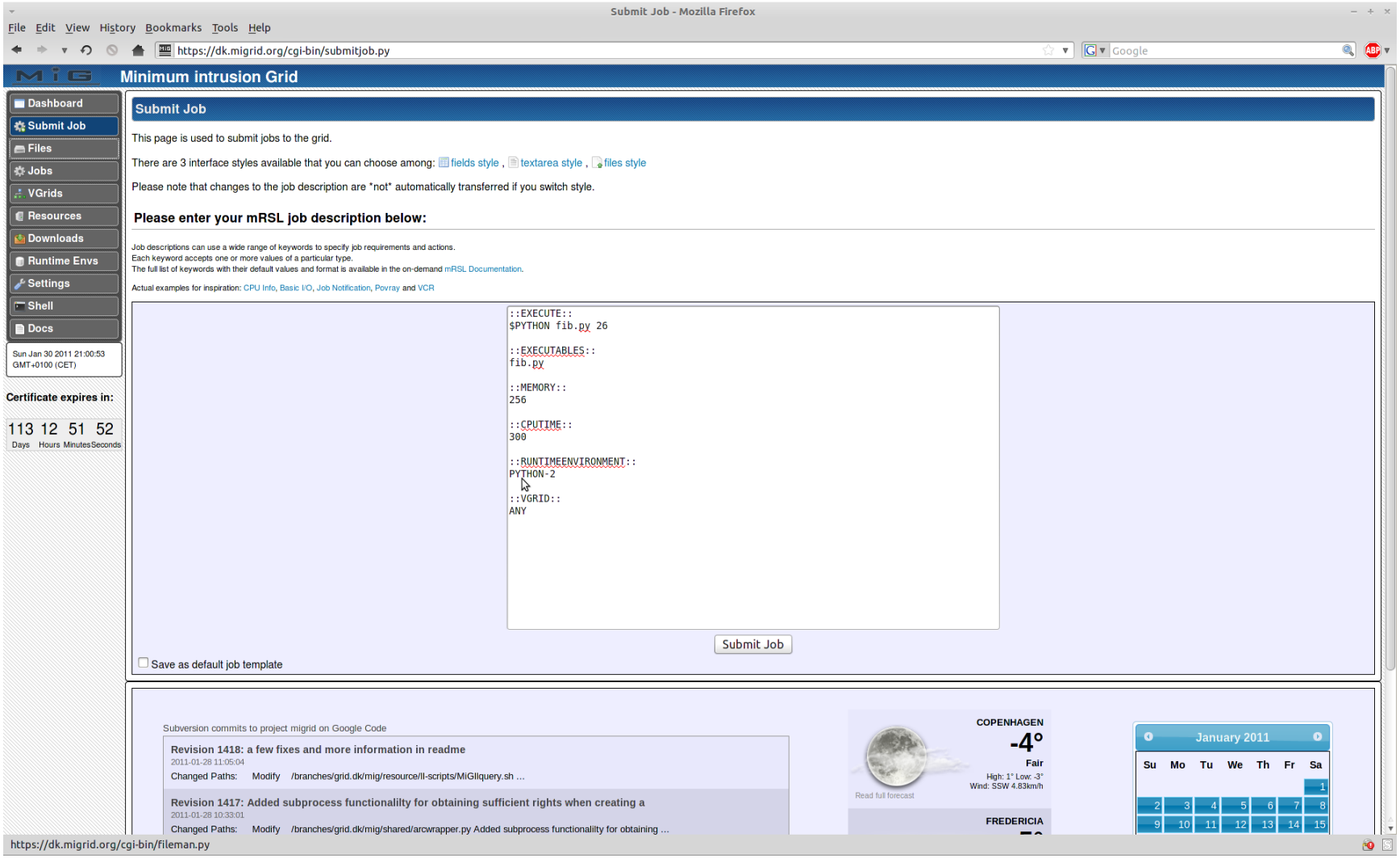
::MEMORY::
256

::CPUTIME::
300

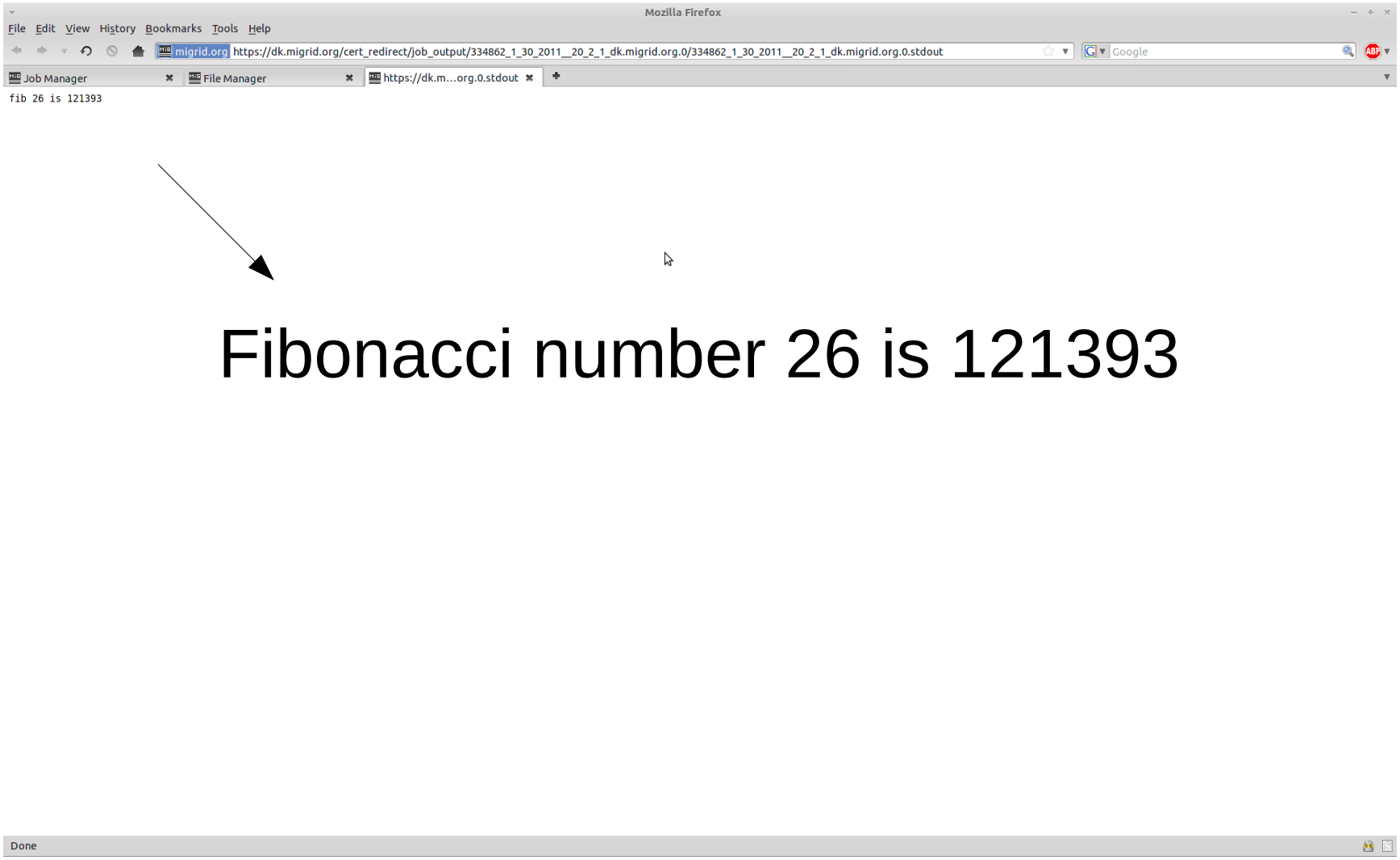
::RUNTIMEENVIRONMENT::
PYTHON-2

MiG
MINIMUM INTRUSION GRID

Runtime Environments (submit)



Runtime Environments (result)

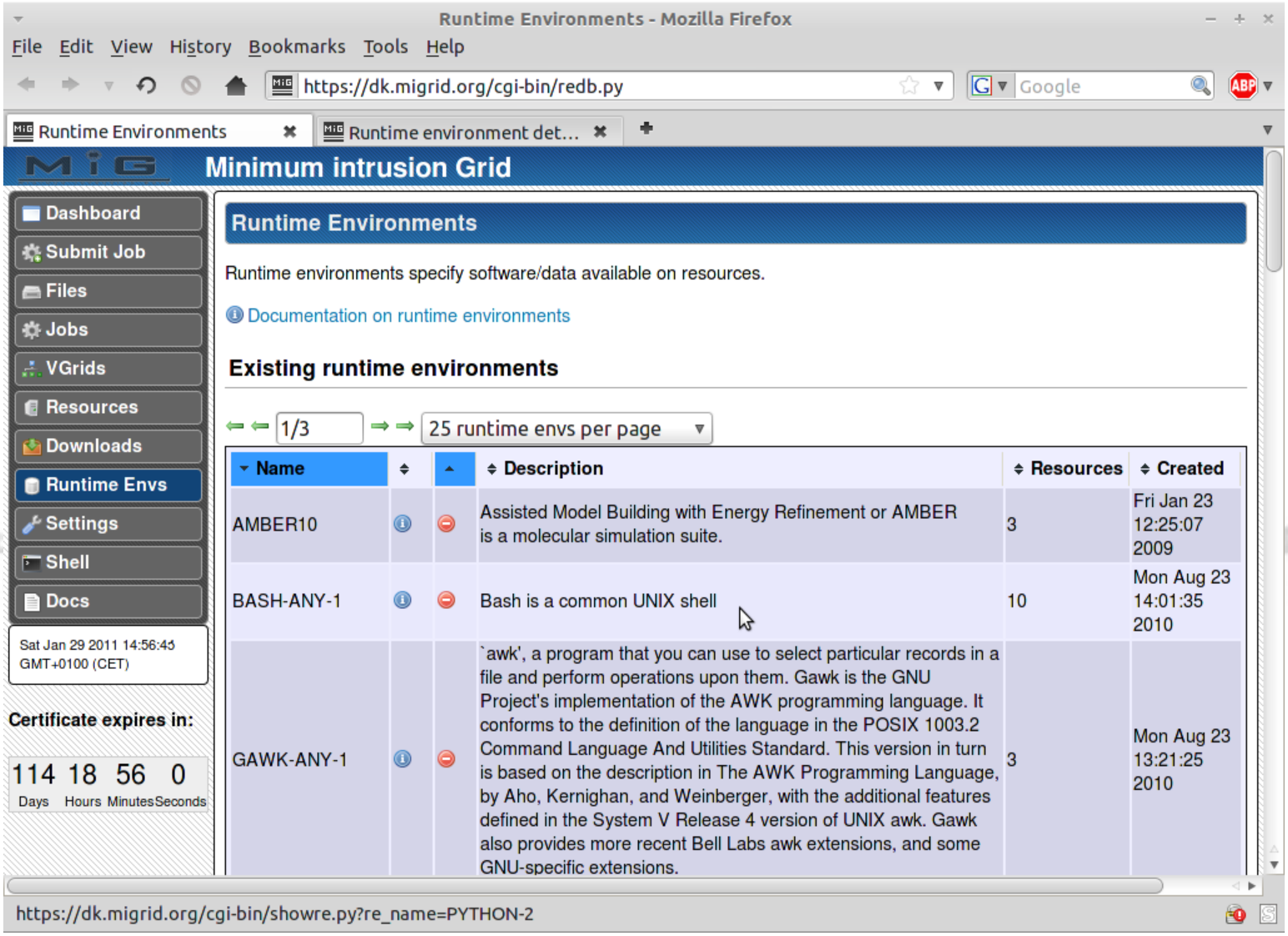


Runtime Environments (optimizations)

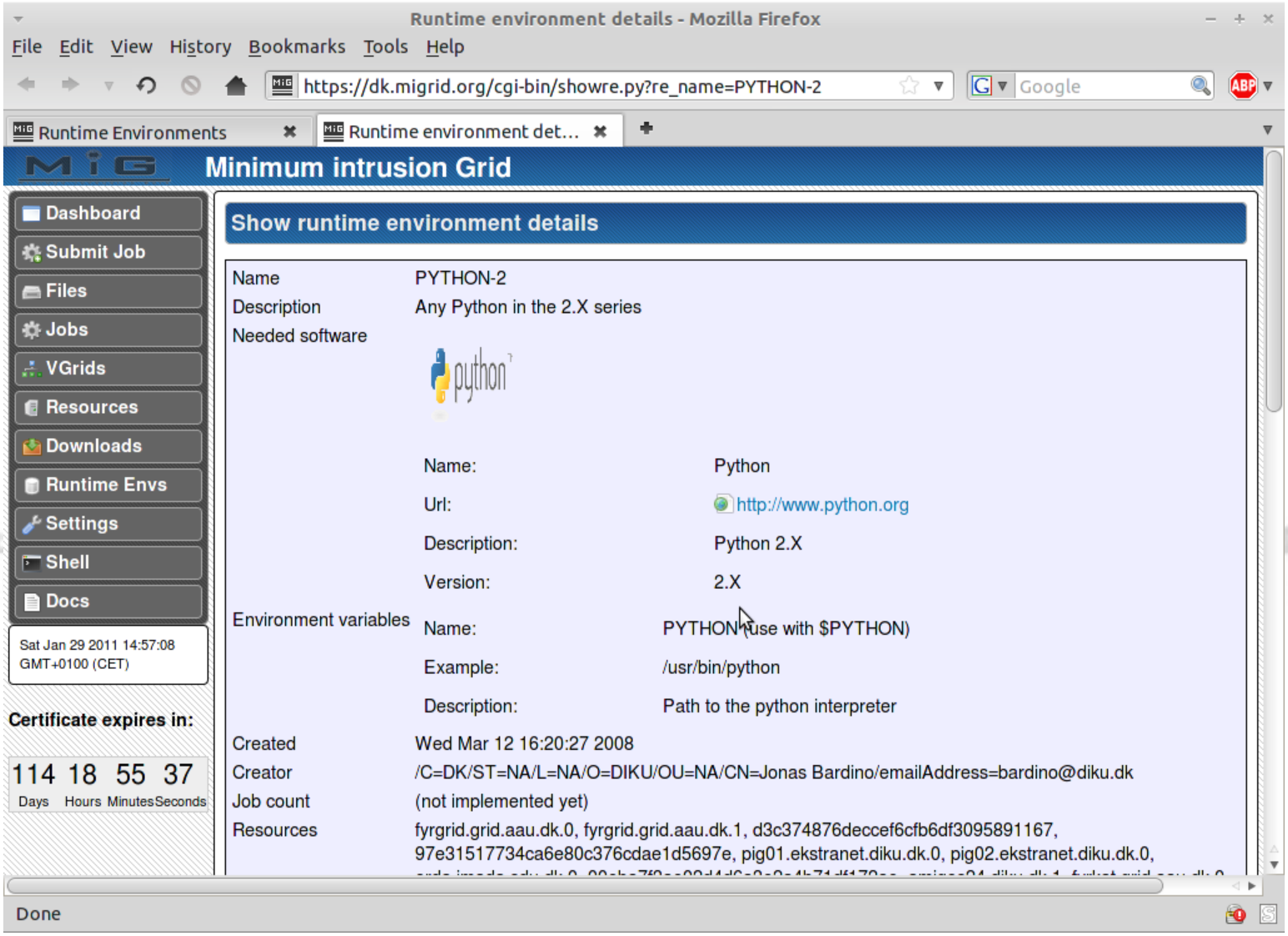
- Horrible performance due to repeated calls
  - recursion calls exact same function many times
- Dynamic programming greatly improves it
  - save result and reuse instead of calling again
- Can we split up in parallel jobs?

MINIMUM INTRUSION GRID

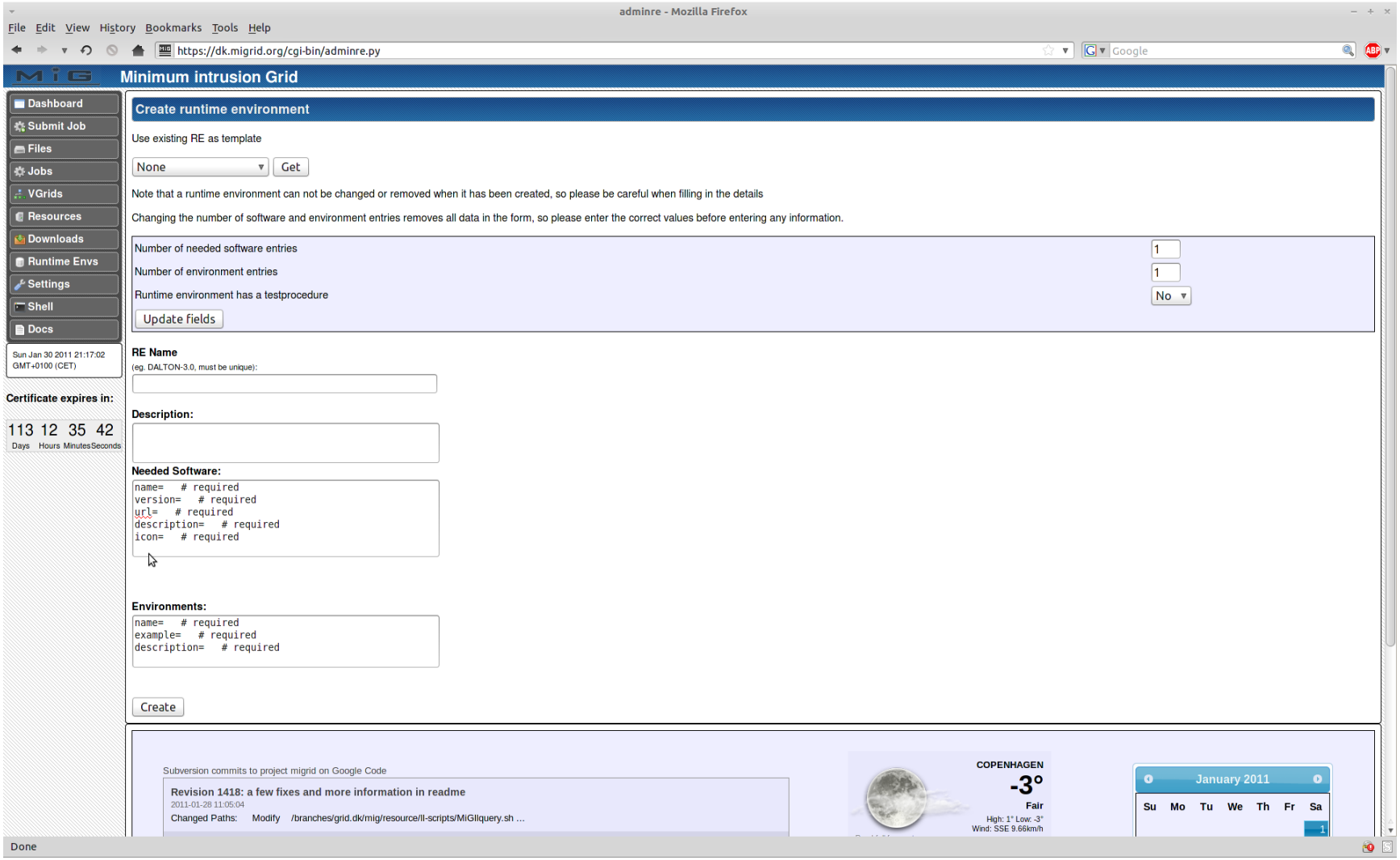
Viewing Runtime Environments



Inspect Runtime Environment



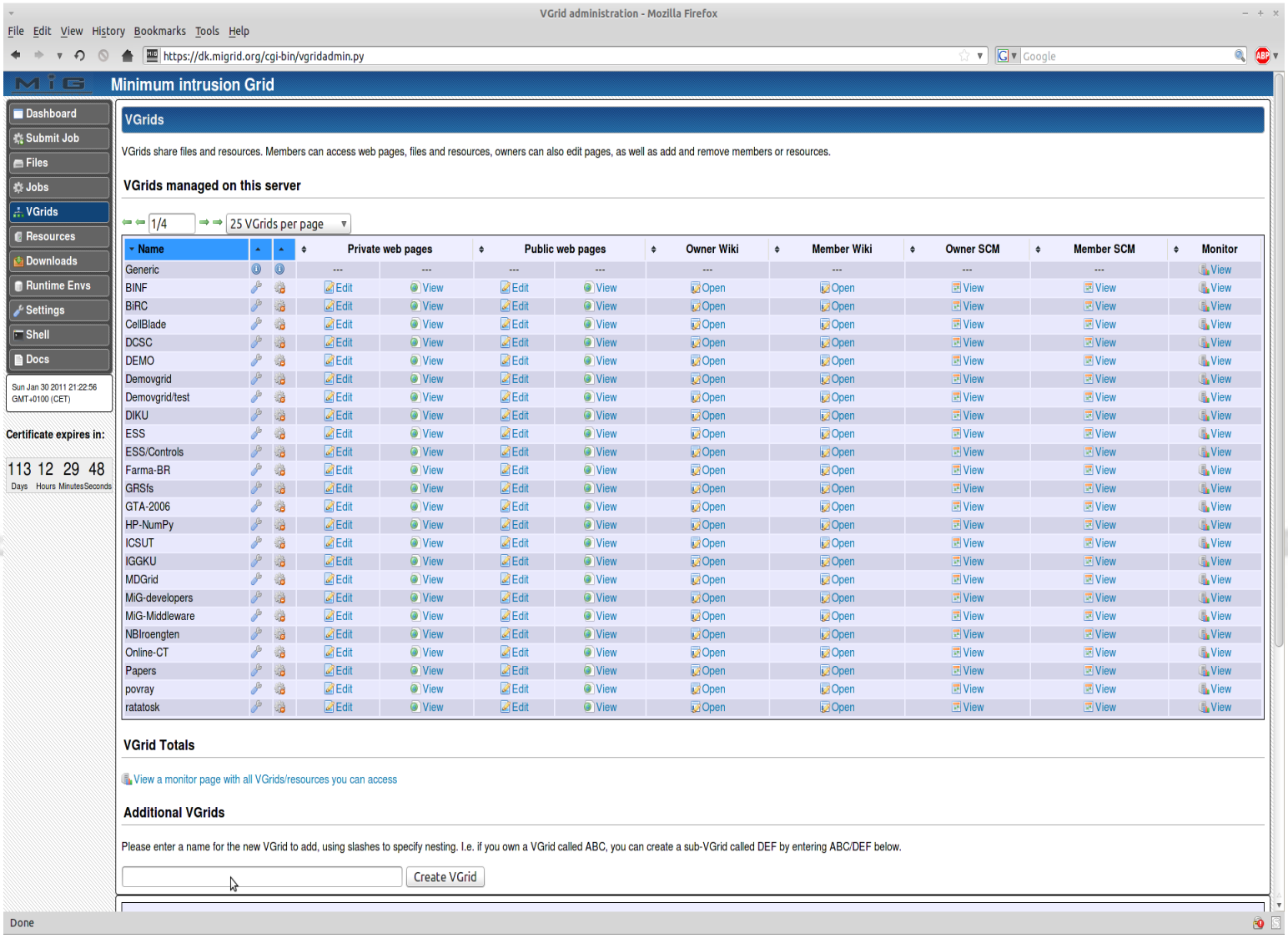
Adding Runtime Environments



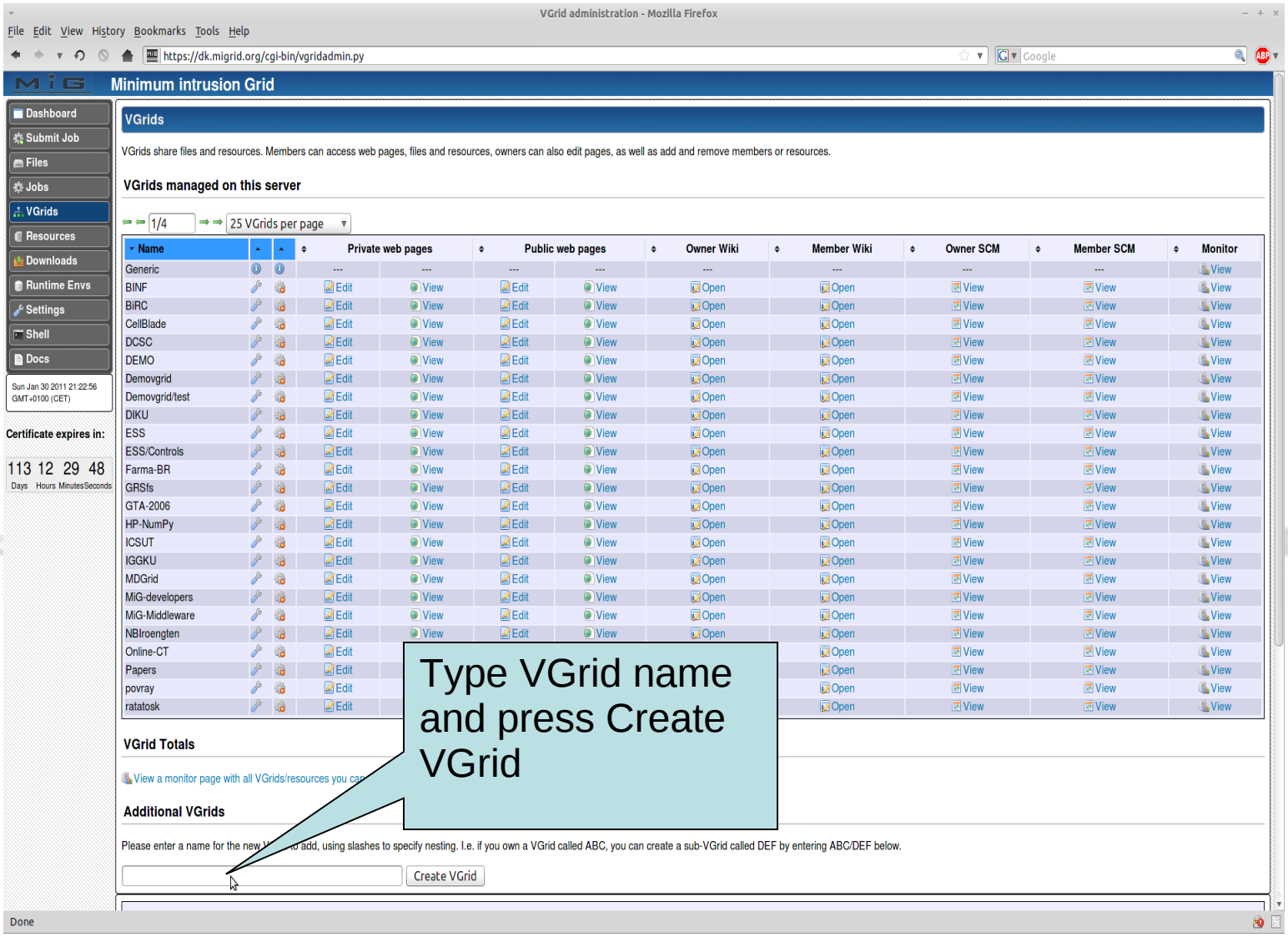
VGrids

- Best thing since sliced bread ☺
- VGrids are Virtual Organizations in MiG
- They are a dead easy way to create collaborations
  - Share files
  - Share resources
  - Private web page / portal
  - Public web page / portal
  - Private Wikis
  - Private repository with revision control (Mercurial)

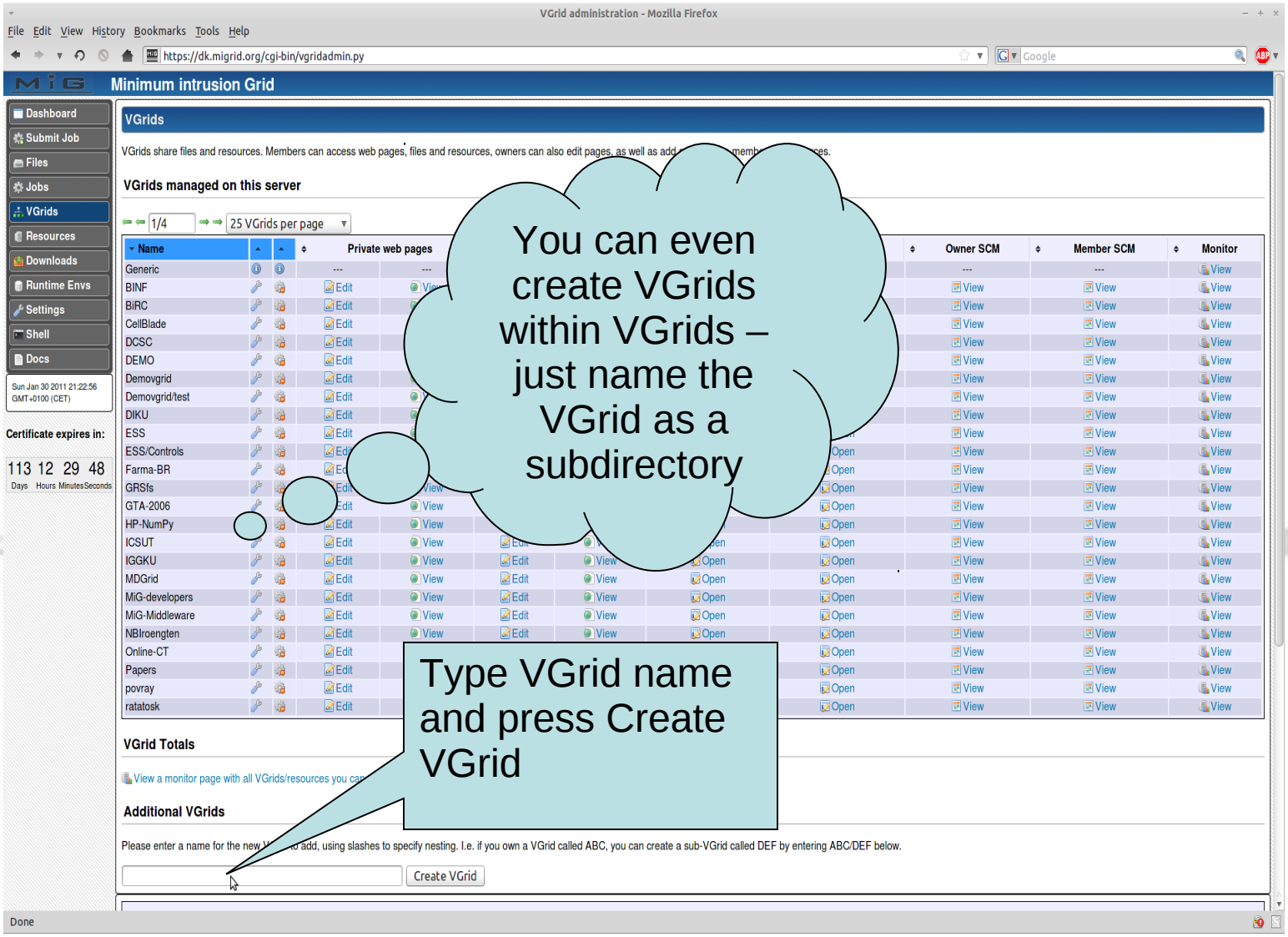
View VGrids


VGrid administration - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://dk.migrid.org/cgi-bin/vgridadmin.py
Minimum intrusion Grid
Dashboard
Submit Job
Files
Jobs
VGrids
Resources
Downloads
Runtime Envs
Settings
Shell
Docs
Sun Jan 30 2011 21:22:56 GMT+0100 (CET)
Certificate expires in:
113 Days 12 Hours 29 Minutes 48 Seconds
VGrids
VGrids share files and resources. Members can access web pages, files and resources, owners can also edit pages, as well as add and remove members or resources.
VGrids managed on this server
1/4
25 VGrids per page
Name
Private web pages
Public web pages
Owner Wiki
Member Wiki
Owner SCM
Member SCM
Monitor
Generic
BINF
BiRC
CellBlade
DCSC
DEMO
Demovgrid
Demovgrid/test
DIKU
ESS
ESS/Controls
Farma-BR
GRSfs
GTA-2006
HP-NumPy
ICSUT
IGGKU
MDGrid
MiG-developers
MiG-Middleware
NBIroengten
Online-CT
Papers
povray
ratatosk
Edit
View
Open
VGrid Totals
View a monitor page with all VGrids/resources you can access
Additional VGrids
Please enter a name for the new VGrid to add, using slashes to specify nesting. I.e. if you own a VGrid called ABC, you can create a sub-VGrid called DEF by entering ABC/DEF below.
Create VGrid
Done

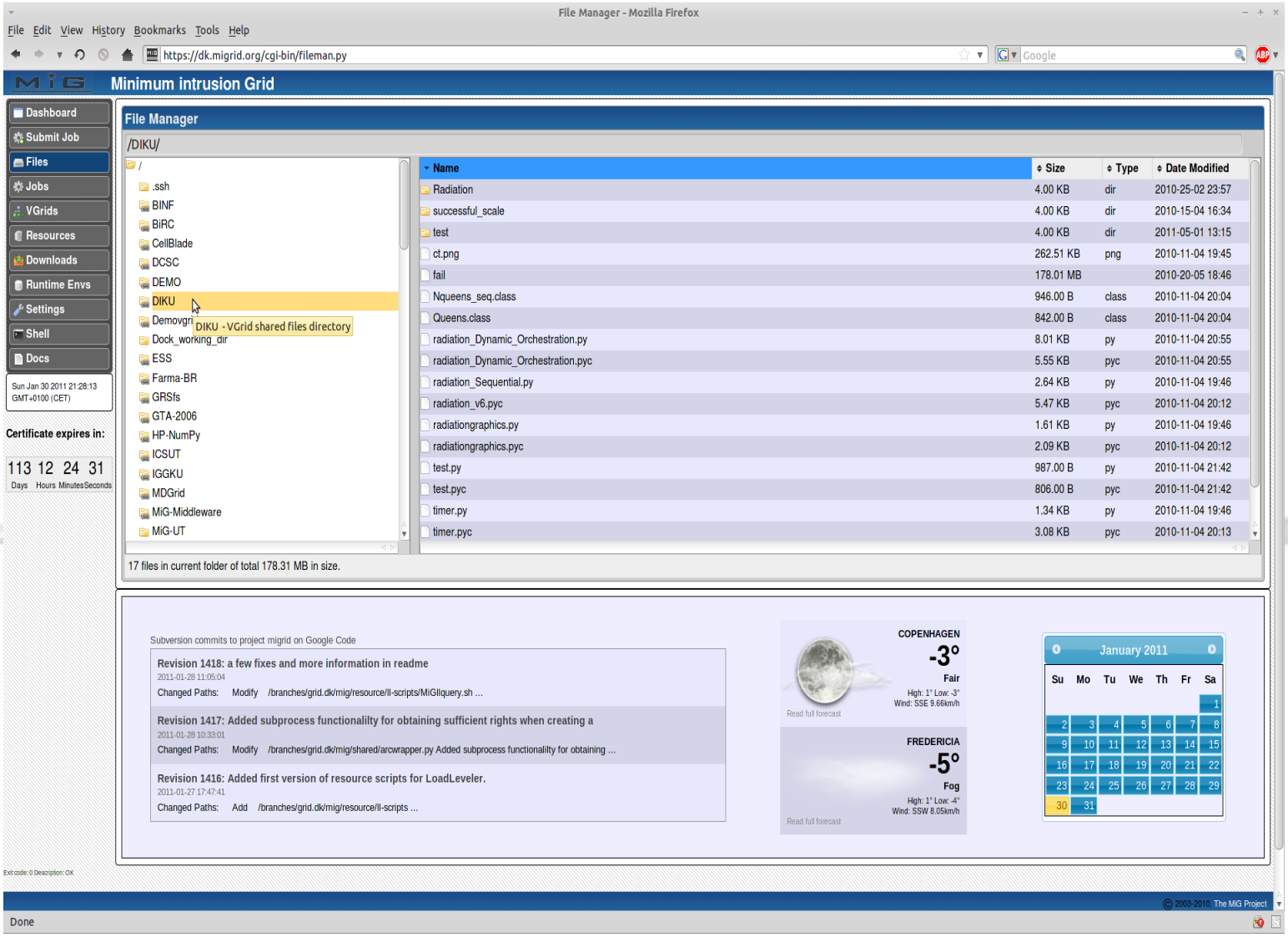
VGrid creation



VGrid creation



VGrids appear as subdirectories



VGrid private homepage I

Private portal for Online-CT - Mozilla Firefox

File Edit View History Bookmarks Tools Help

migrid.org https://dk.migrid.org/cgi-bin/showvgridprivatefile.py?vgri

Google

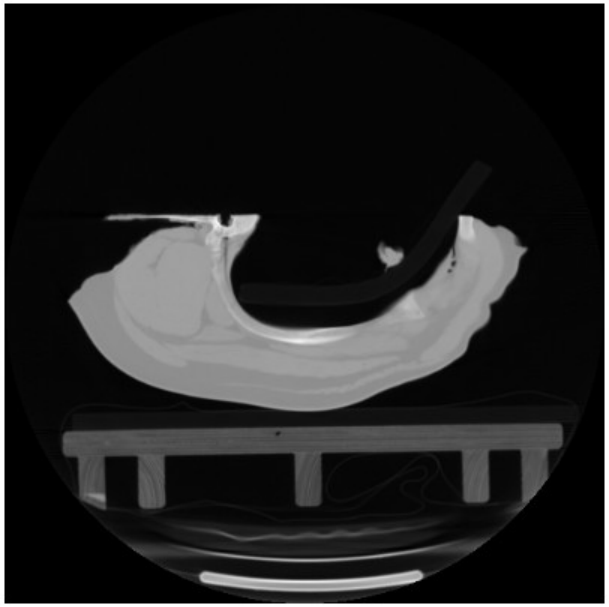
Private p... | Resource ma... | Resource Ma... | Resource Edi... | downloads | Runtime Envi... | Runtime envi...

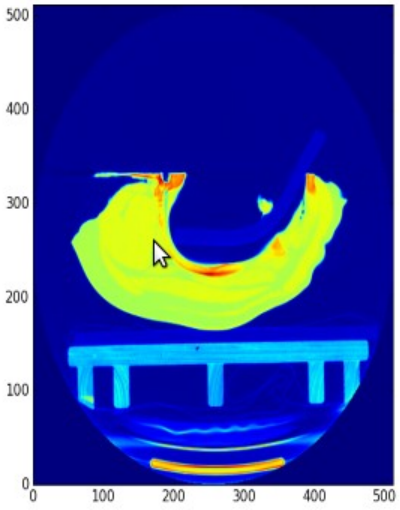
Online CT Member Portal

Online CT Member Portal

This is a work-in-progress portal for Online CT simulations on MiG!

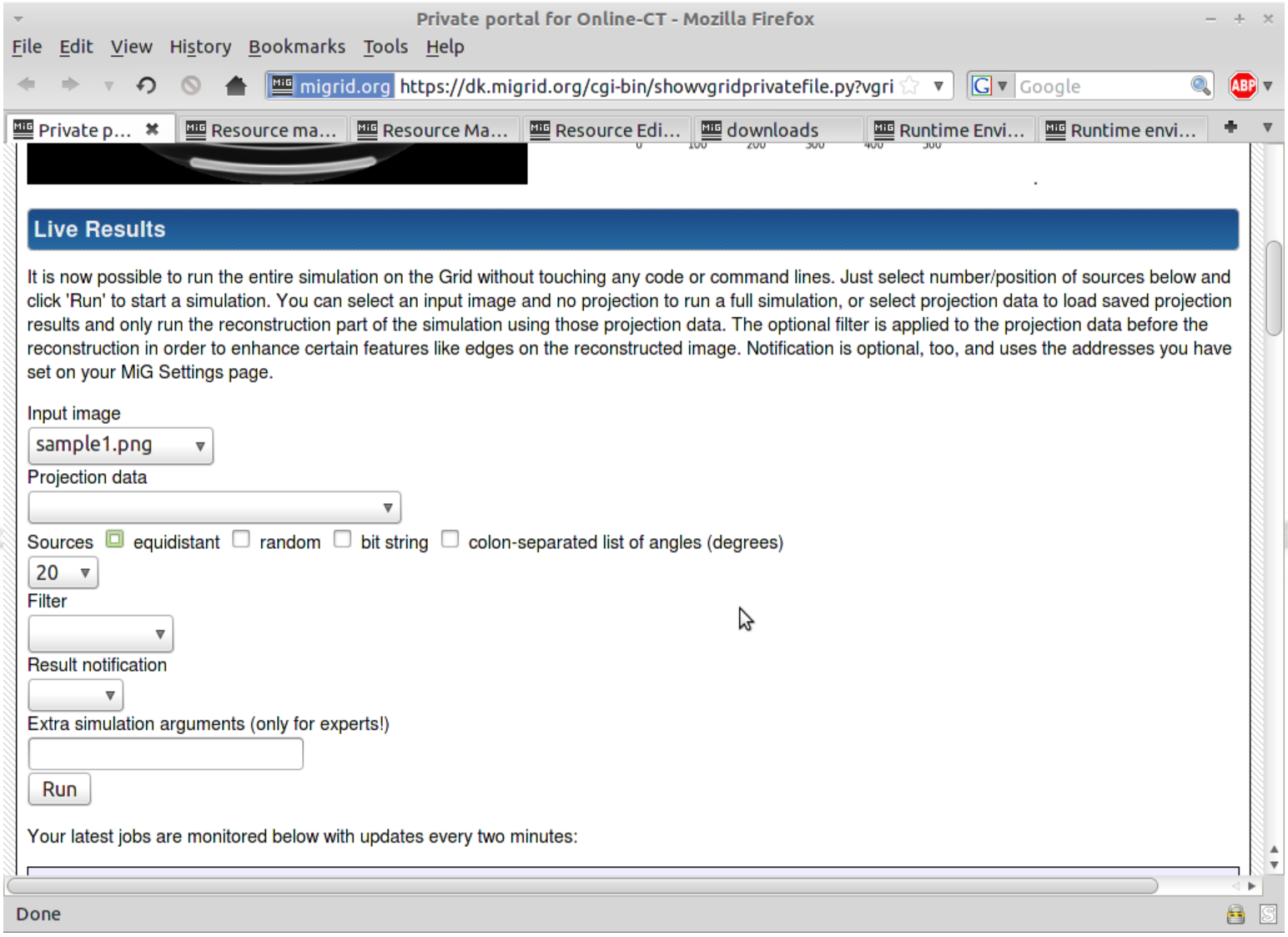
We generally simulate CT scanning as forward and back projection of a meat intersection like the ones below.



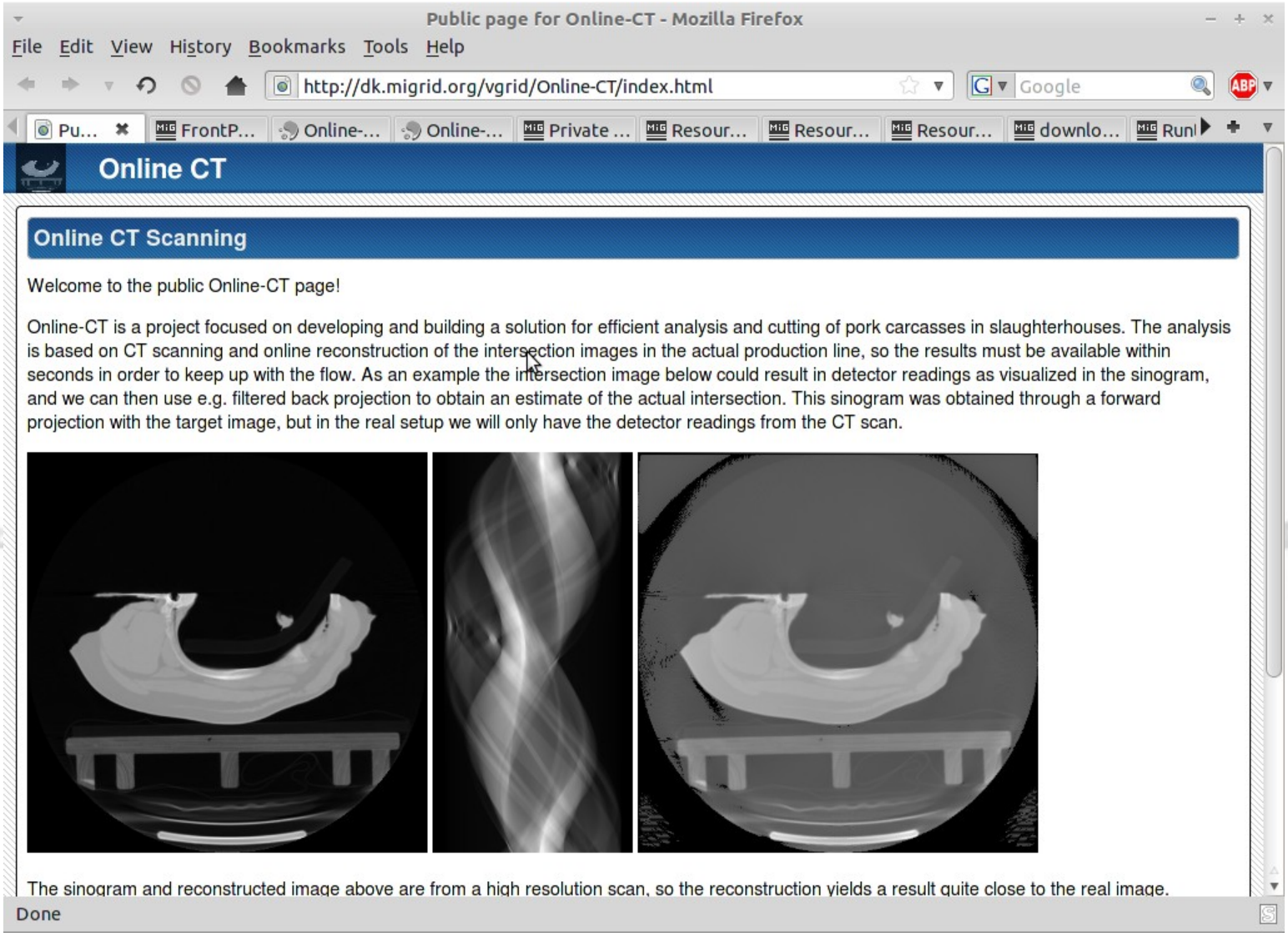


Done

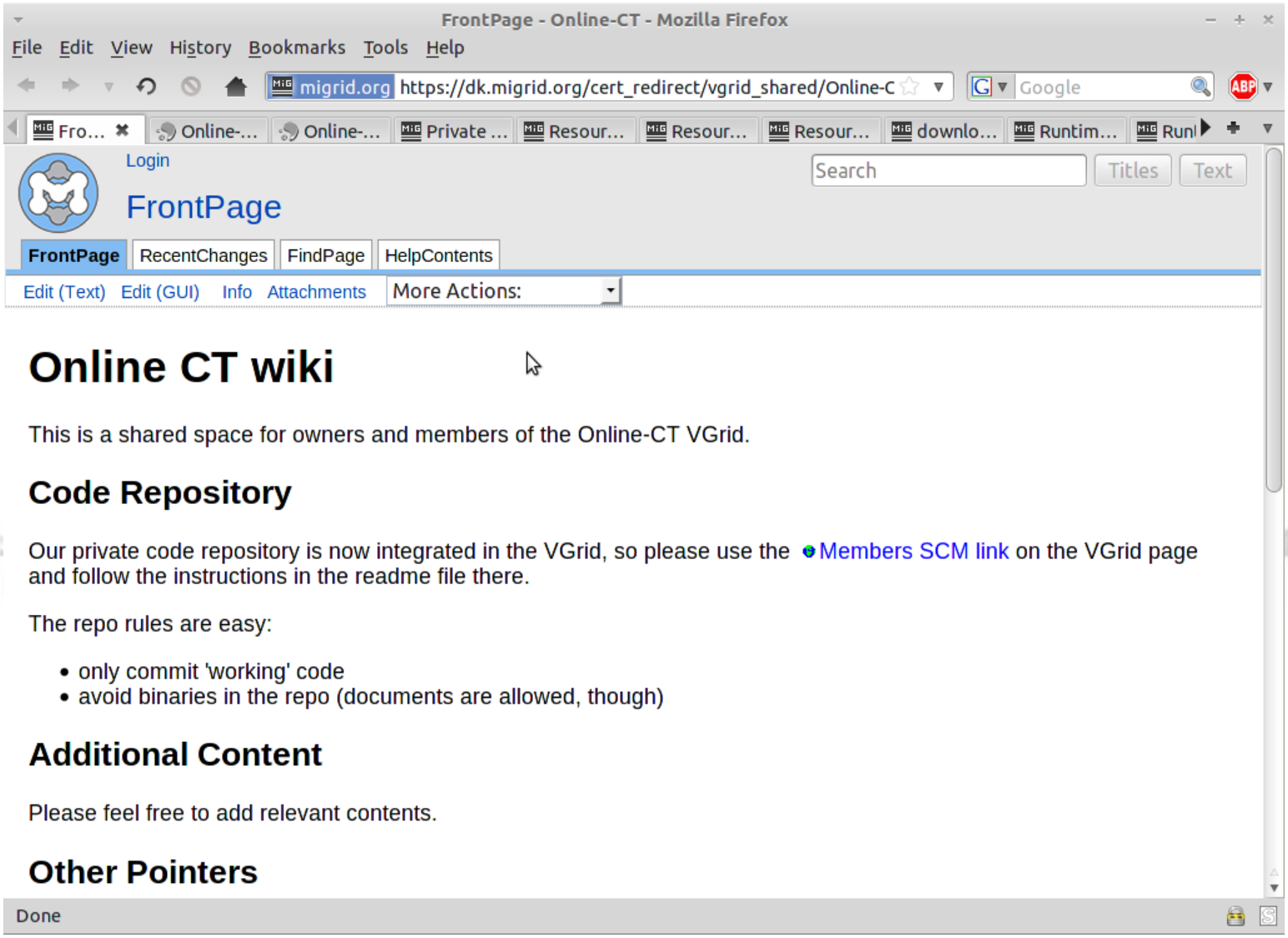
VGrid private homepage II



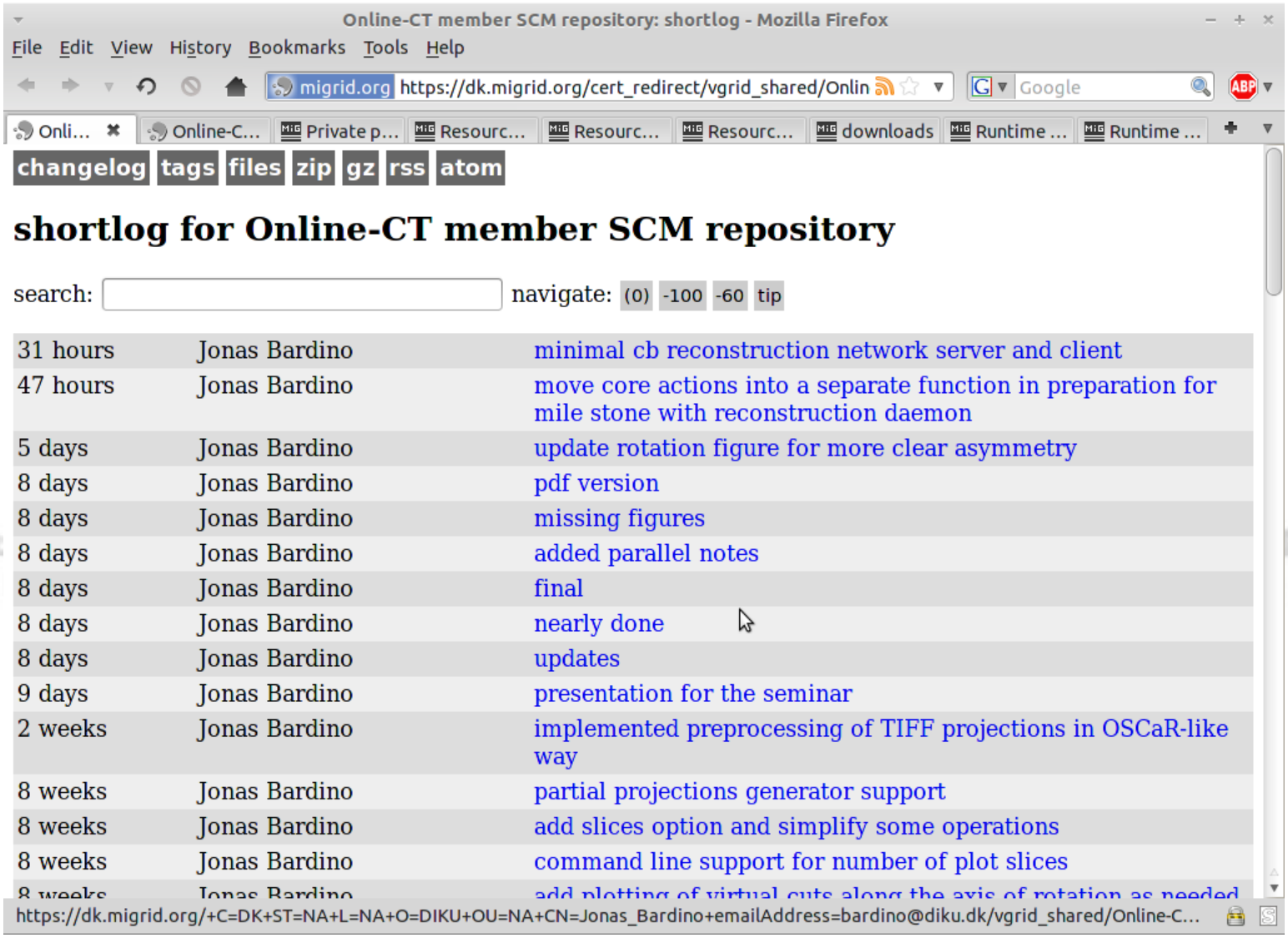
VGrid public homepage



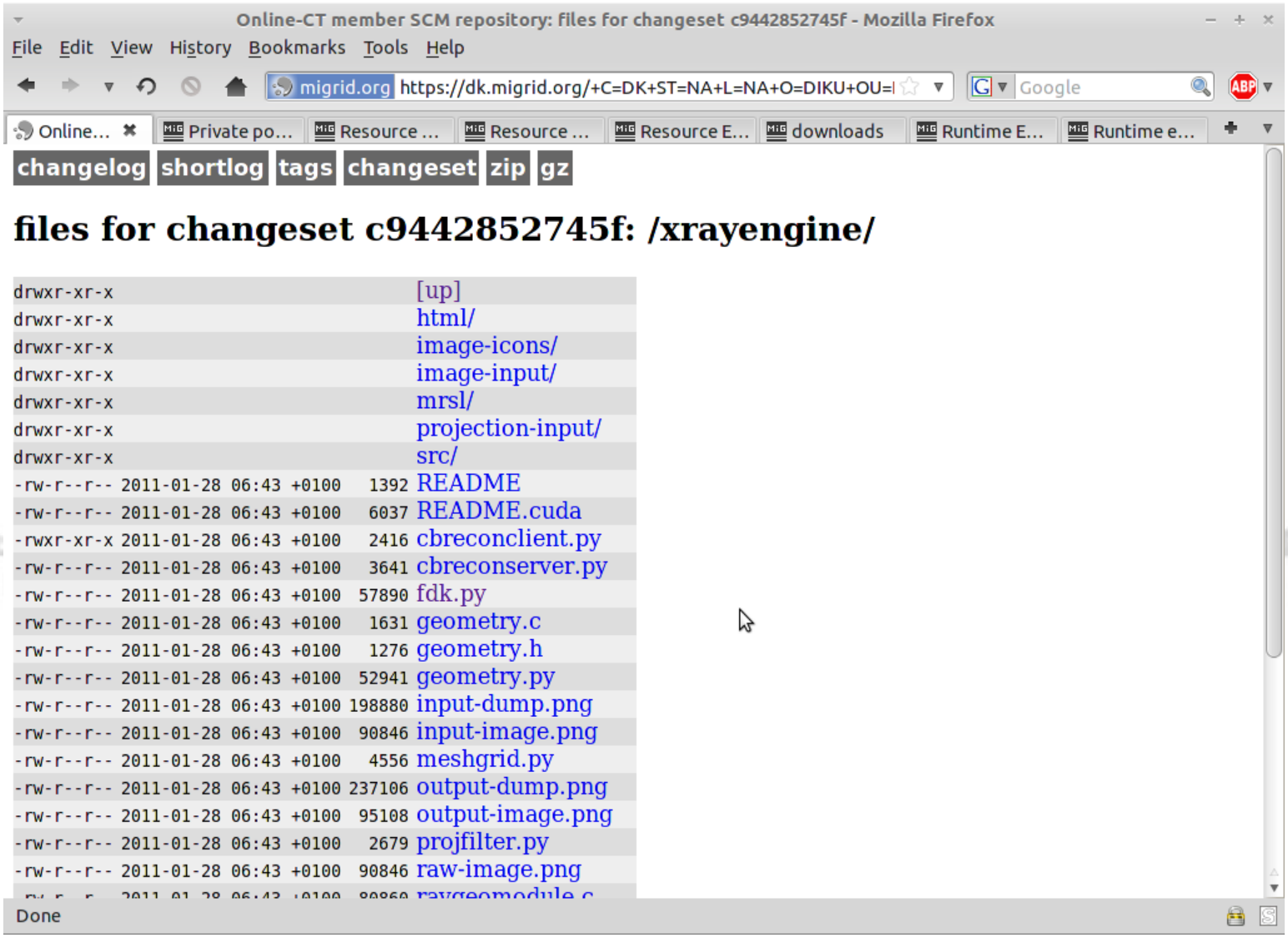
VGrid private wiki



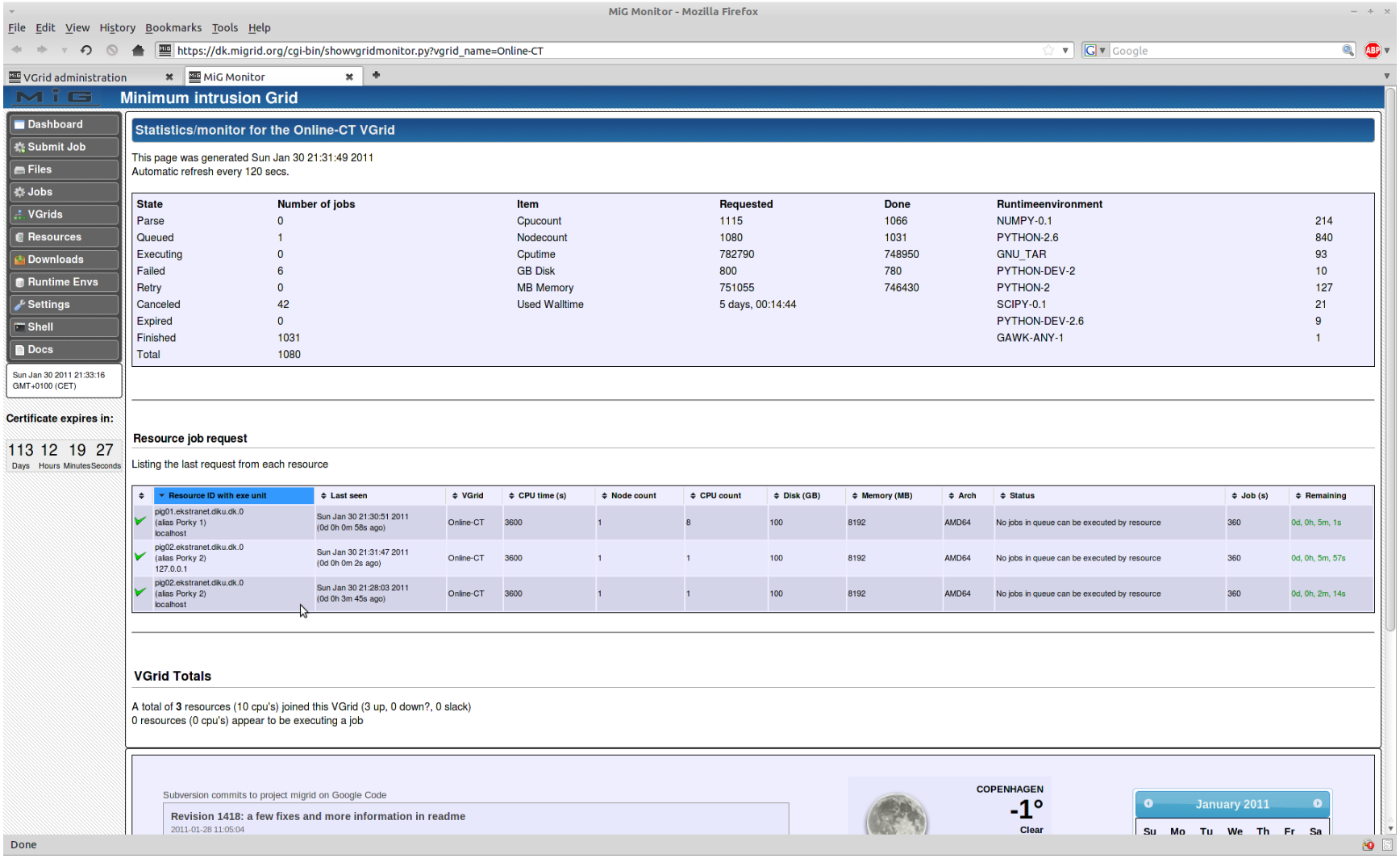
VGrid private SCM I



VGrid private SCM II



VGrid Monitor



MiG from the users POV

- Browser
  - x.509 certificate and HTTPS
- Manage files in grid home directory
- Submit jobs, view jobstatus etc.
- MiGscripts. Wrappers around “curl”.
- File handling:
  - migput, migget, migmkdir, migls, migrm, migcat, ...
- Job handling:
  - migsubmit, migstatus, migcancel

Using the scripts

Generate and download scripts from MiG
Download or write job specification (mRSL):
`http://dk.migrid.org/public/examples/io.mRSL`
This example creates and uploads 'input.txt' which is sent to output.txt
Run the job:

```
echo “test job” > input.txt
migput.sh input.txt input.txt
migsubmit.sh io.mRSL
```

returns a JOB_ID that is used for further treatment of the job
the job creates a file (outputfile)
`migstatus.sh JOB_ID`
to get the status of the job
`migcat.sh output.txt`
to show the job output when job is done
Further examples in tutorial at http://www.migrid.org

Resource models

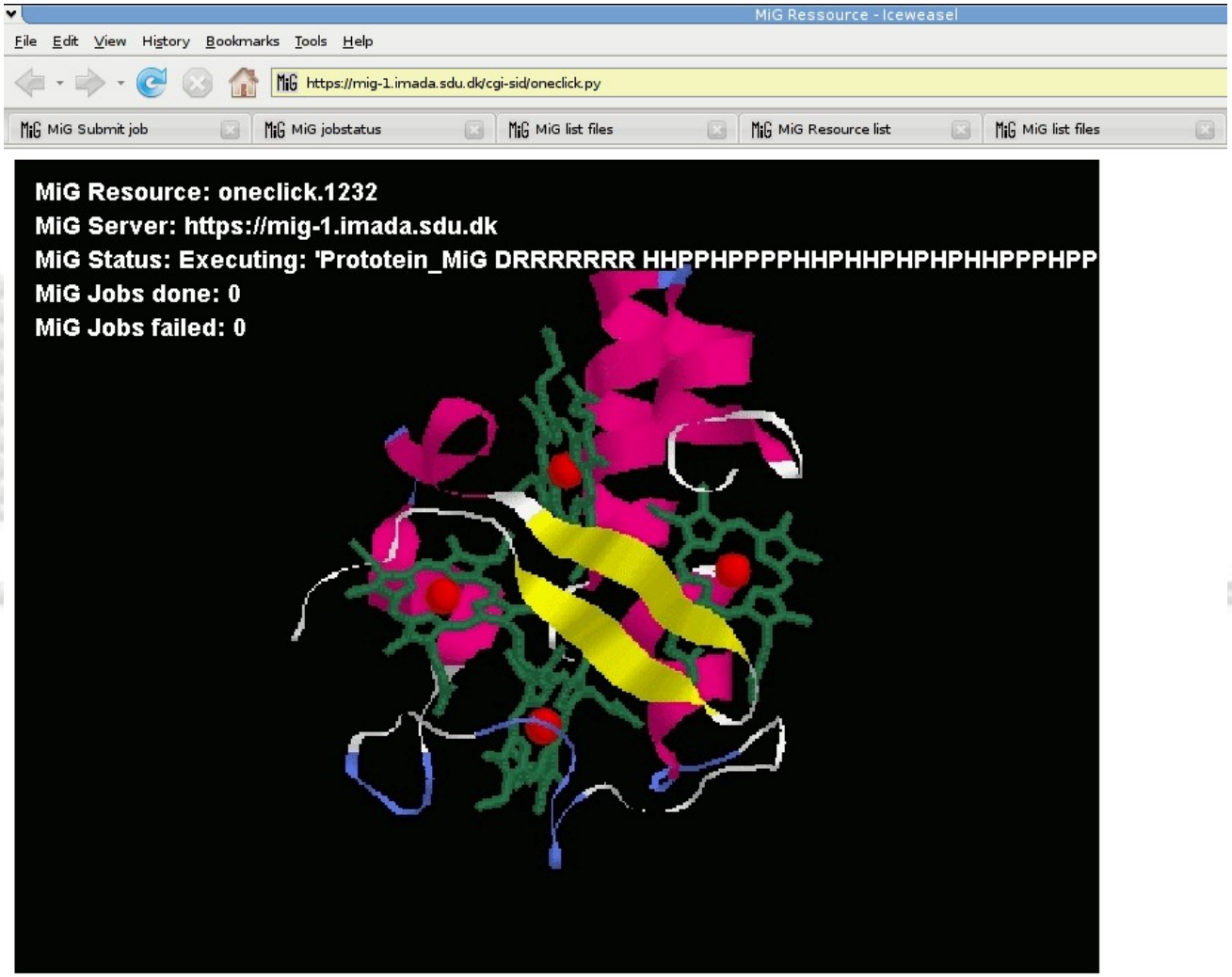
- For now there are 3 types of resources
- One-Click
  - The simplest possible way to contribute
- MiG-SSS
  - A very powerful screensaver
- Conventional resources
  - Like all other Grid systems, only easier

MINIMUM INTRUSION GRID

The One-Click resource

- The web-browser as a Grid resource
  - Uses java applet plugin
  - No installation required
  - No administrational work required
  - All devices capable of running a JVM can be addressed
- Security
  - Secure communication through https
  - Sandboxed through the java applet security model
  - No local file access
  - No local process execution

One Click



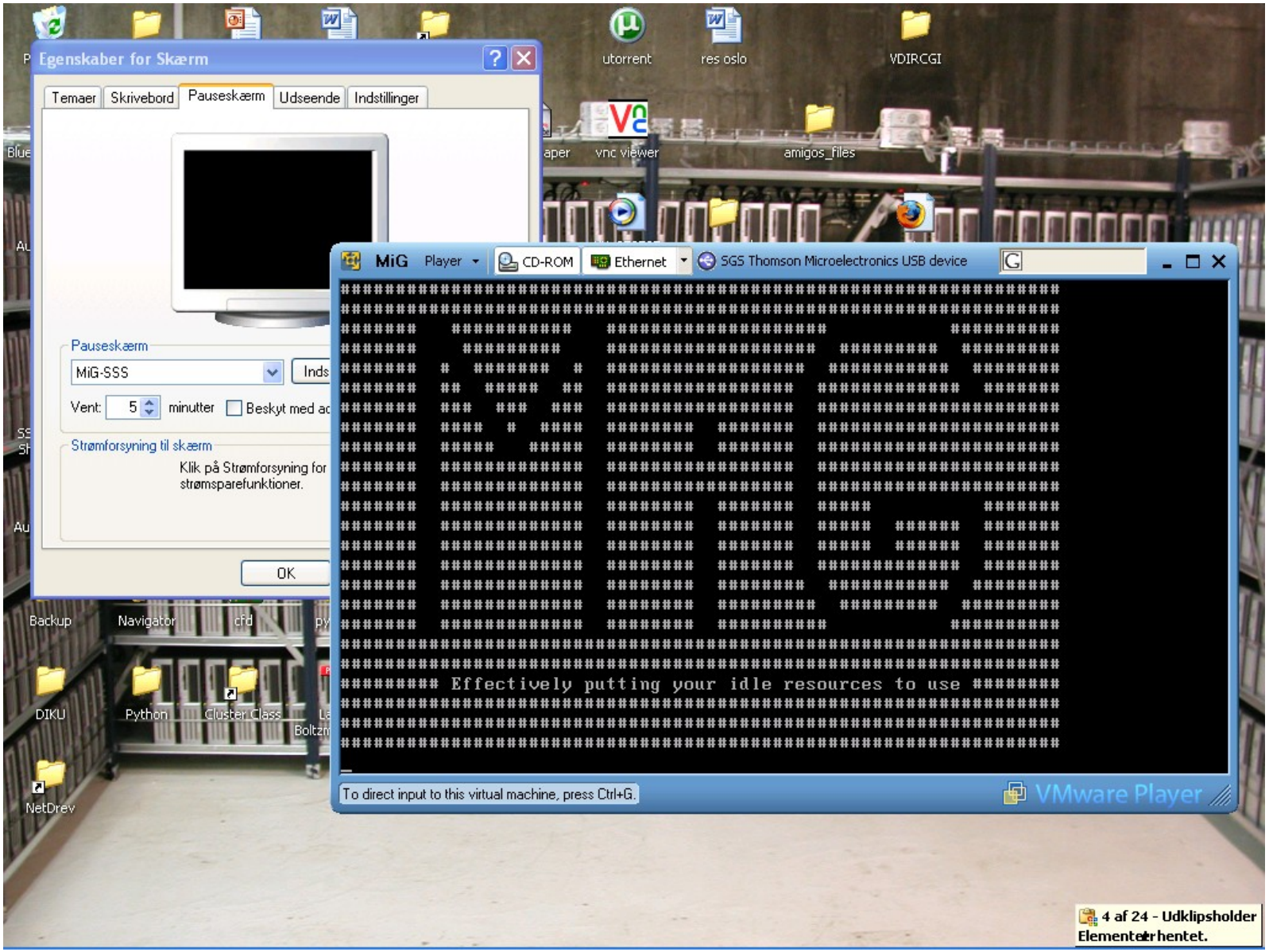
http://dk.migrid.org/public/examples/Prototeins.zip

MiG-SSS

- The goal is to combine Screen Saver Science and Public Resource Computing:
  - Harvest idle compute cycles from computers in screen saver mode
  - Put them on tap for grid jobs!
- Implemented as a virtual machine that is controlled by the screen-saver component
- Got a migration feature
  - But this requires high bandwidth and is currently not suited for ADSL lines

MINIMUM INTRUSION GRID

MiG-SSS


Egenskaber for Skærm
Temaer
Skrivebord
Pauseskærm
Udseende
Indstillinger
Pauseskærm
MiG-SSS
Vent:
5
minutter
Strømforsyning til skærm
Klik på Strømforsyning for strømsparefunktioner.
OK
utorrent
res oslo
VDIRCGI
vnc viewer
amigos_files
MiG
Player
CD-ROM
Ethernet
SGS Thomson Microelectronics USB device
Effectively putting your idle resources to use
To direct input to this virtual machine, press Ctrl+G.
VMware Player
Backup
Navigator
cfd
DIKU
Python
Cluster Class
NetDrev
4 af 24 - Udklipsholder
Elementet er hentet.

PS3 on MiG

- The Cell CPU is a multi-core microprocessor (9 cores)
  - Basically a (1 master)/(8 node) cluster on a chip
  - PS3 version only enables 6 of the nodes (SPE's)
- 200+ GFlops (SP) @ 3.2 GHz
- 20+ Gflops (DP) @ 3.2 Ghz
- A lot of PS3's are available
- A live Linux CD puts PS3’s on MiG

MINIMUM INTRUSION GRID

Adding conventional resources

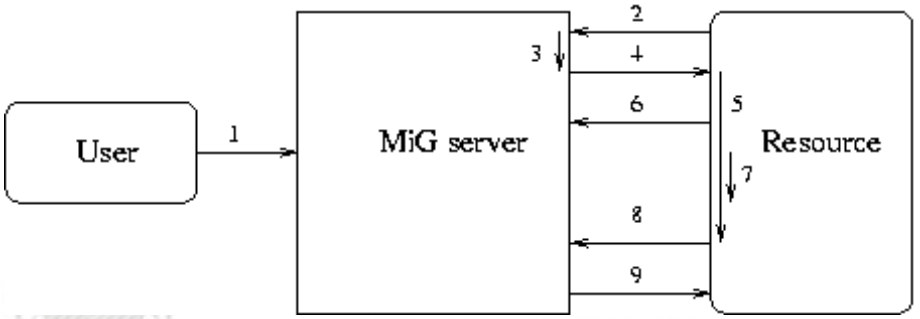
- Conventional resources include
  - Workstations
  - Clusters
  - Supercomputers
  - You may choose to execute with low priority on a resource
  - You may specify a execution pre-condition command to e.g. stop taking jobs when the CPU load is high
  - No Windows native resources

MINIMUM INTRUSION GRID

Requirements

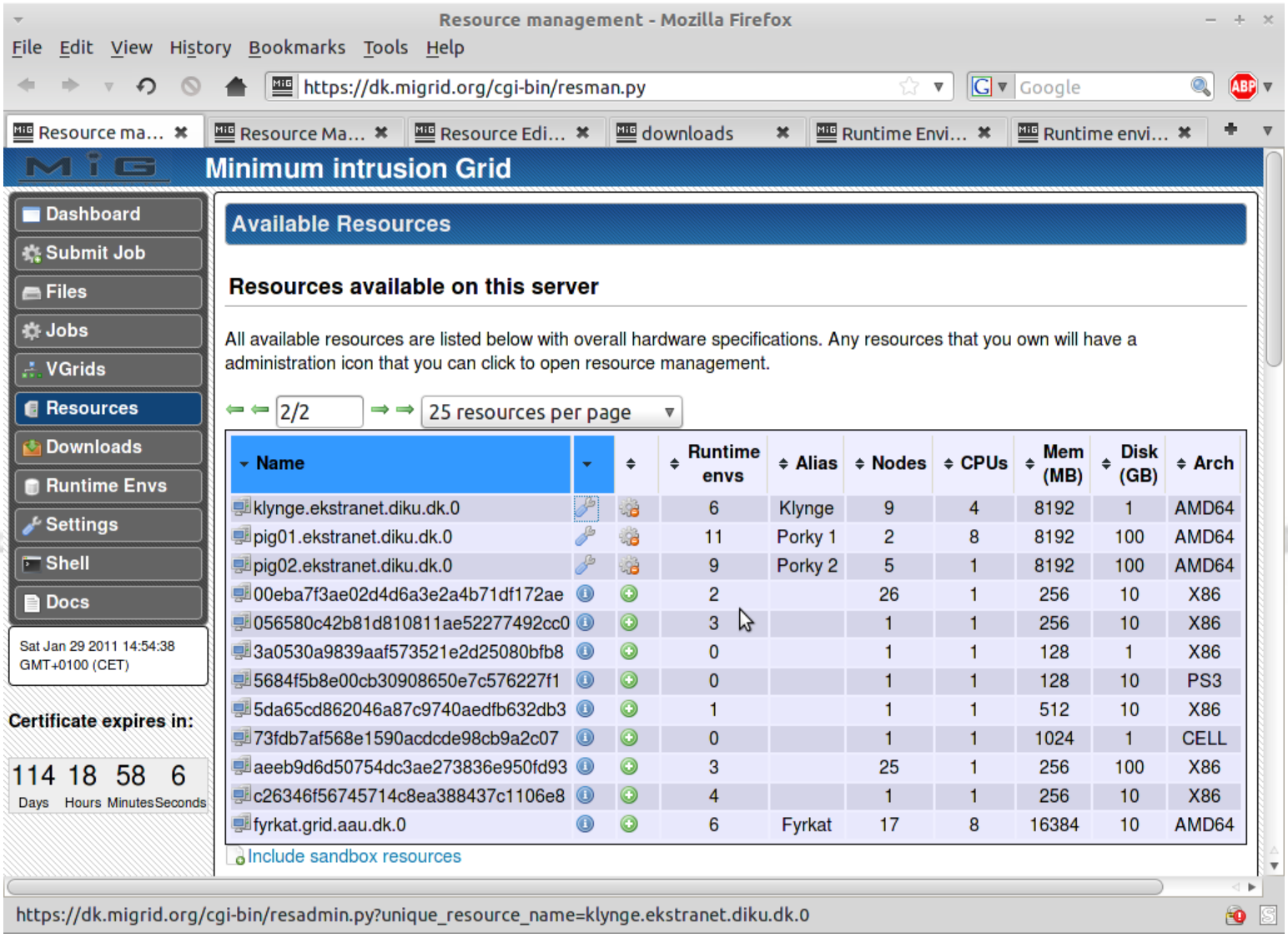
- You must know your resource ☺
- Requires ssh in
- Requires https out

Under the hood: job flow

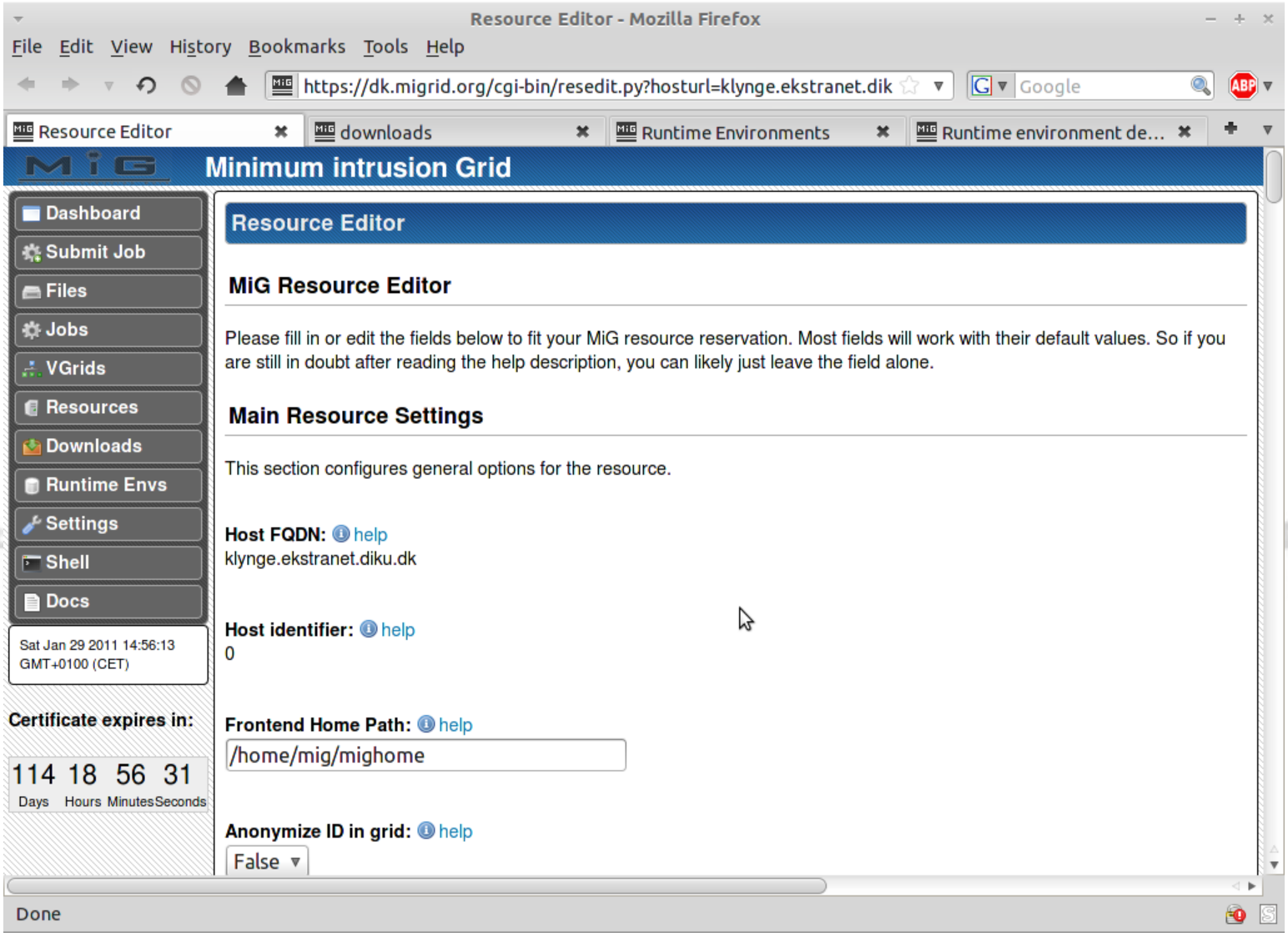


1. User communicates with the MiG server using HTTPS and certificates.
2. Resource requests a new job to execute (HTTPS)
3. The MiG server creates the job script.
4. MiG server sends the job to the resource using SCP.
5. The resource starts the job script.
6. Resource pulls the inputfiles from the MiG server (HTTPS)
7. The job is being executed
8. Resource sends outputfiles to the MiG server (HTTPS)
9. MiG server cleans up the resource using SSH (files and processes).

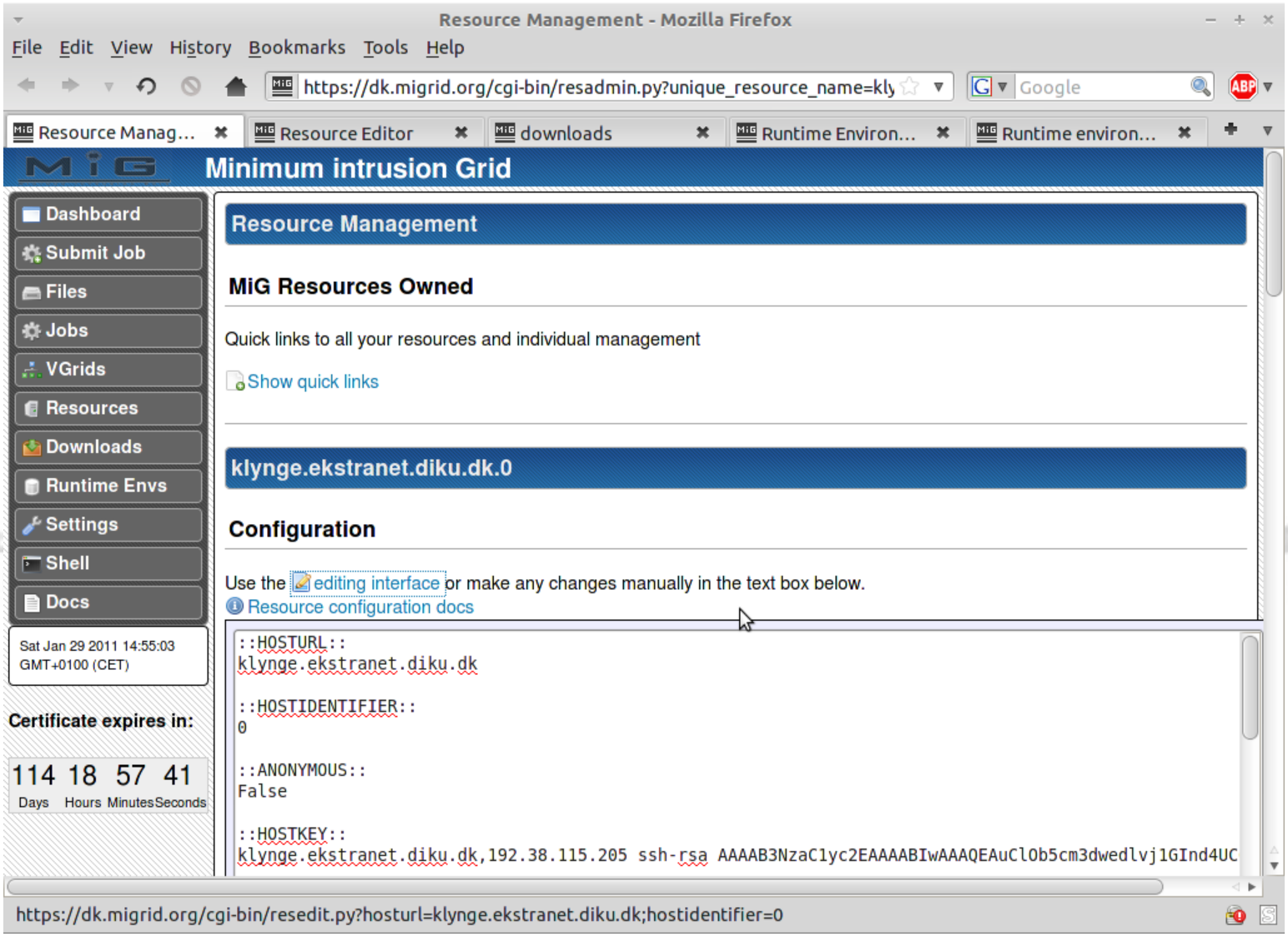
Viewing resources



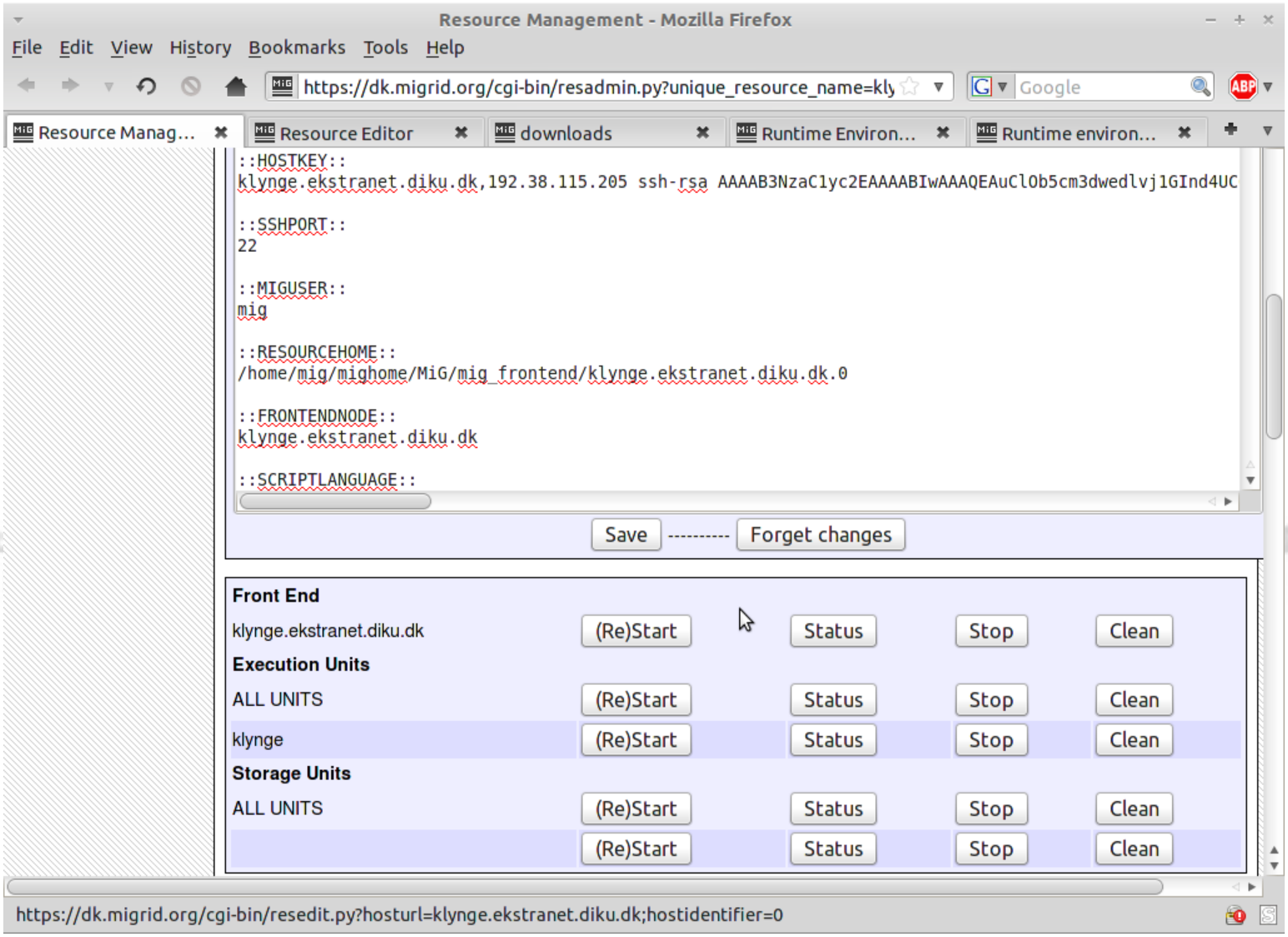
Adding/editing a resource



Managing your resources I



Managing your resources II



Screen Saver Science

PRC with deadlines

- Screen Saver Science is hard to merge with strong scheduling
  - Once a resource accepts a job – just how much time is it offering?
- We create an exponential averaged statistic prediction
  - 24x7 resolution

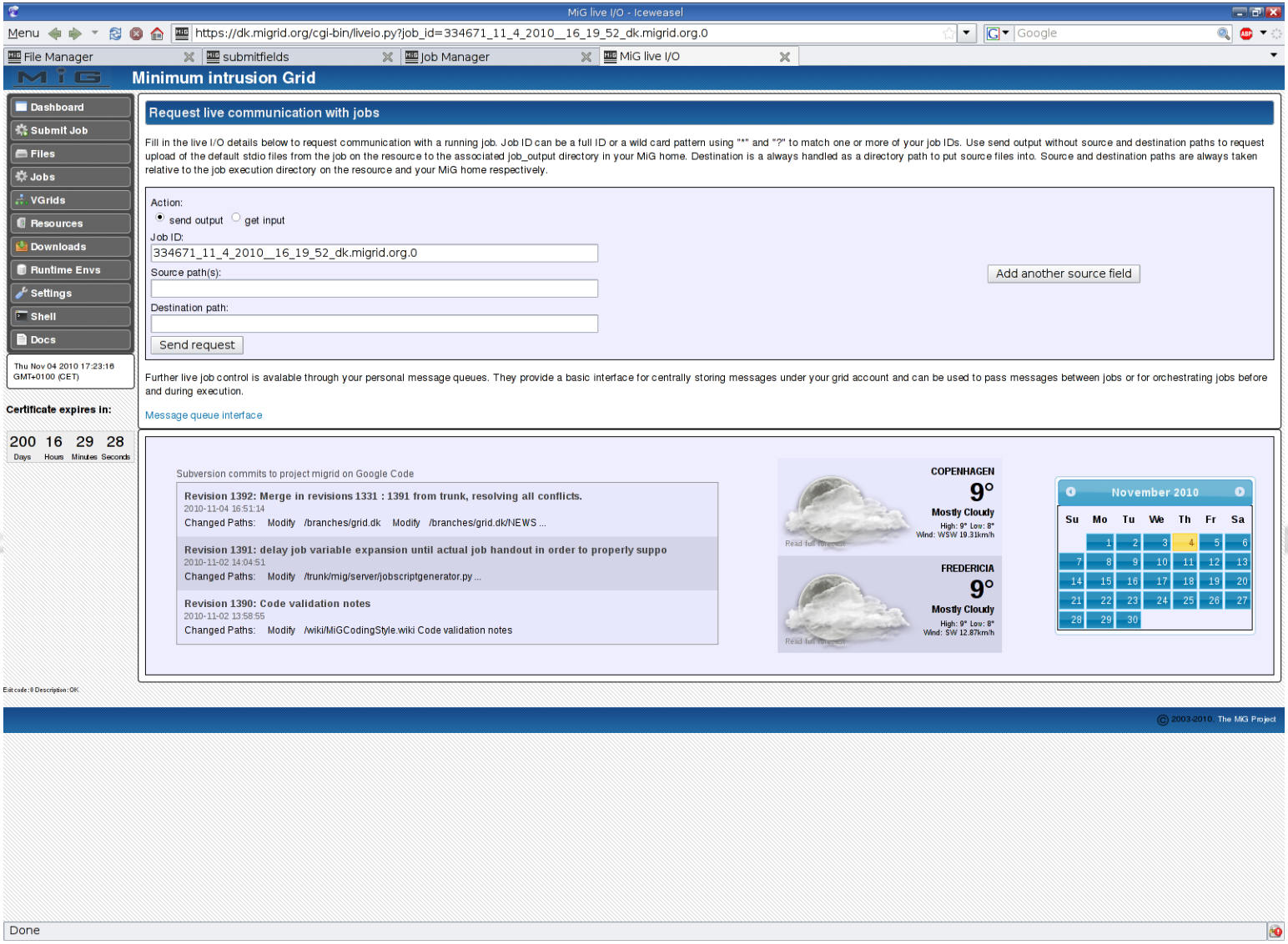
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
0	4,281289	0,609102	4,370668	6,52596	3,298105	5,535493	5,512131
1	0,140011	4,552117	2,280195	6,468878	6,98397	4,135618	6,896847
2	6,214814	2,297592	5,894947	2,217671	5,421552	5,990491	6,273165
3	0,909552	3,254109	5,488454	2,996888	7,434242	2,463183	2,790015
…	…	…	…	…	…	…	…
21	1,102019	5,191171	4,97769	3,163451	2,796037	2,292304	2,448104
22	5,448806	6,861695	6,110104	3,13554	0,411362	0,778049	0,470441
23	4,257551	2,140985	0,800296	5,465861	1,4688	6,577736	7,138229

Live job I/O

- Job input/output *during* execution
- Open job status and request live I/O
  - an update request is sent to the resource
  - it uploads/downloads the current specified files
  - From/to MiG home directory
  - Message queue support

MINIMUM INTRUSION GRID

Live job I/O example



Graphical Access

- Interactive jobs
- VNC display on server
  - anonymous display at MiG servers
- User: https+cert+java client
- Resource: X forwarding to server display
- CGI script
  - start VNC display
  - redirect user to display

MINIMUM INTRUSION GRID

Developer Accounts

- UNIX account on amigos18.diku.dk
- private apache server on high ports
  - only my-apache stop/start (not conf access)
- snapshot or svn checkout of MiG code
- run MiG daemon(s) from home dir
- simply use https://amigos18.diku.dk:PORT/
- allows most features

MINIMUM INTRUSION GRID

Transparent Remote File Access

- Huge input files incur a number of problems:
  - Download time vs. total execution time
  - Job execution on the resource is delayed
  - Storage requirements on resources
  - Often only small scattered fragments of input files are needed
- We seek to fix this using automatic on-demand download

Transparent remote file access

- Download time of input files is eliminated; the job is started immediately
- Only needed data is transferred
  - Or as little extra as possible
- Jobs previously impeded from grid submission due to huge input files are now ready for the grid
- The model outperforms a standard copy-model
- The library is entirely user-level
- Not in production!

Additional topics

- Automatic job variables
- Grid enabling apps with xmlrpc or json
- MiG Shell
- Sshfs mount MiG home
- Storage resources
- Custom page layout
- Widgets
- Android access (certificate proxy)

MINIMUM INTRUSION GRID

MiG Conclusions

- MiG is 'feature complete'
- easy to use: user / resource owner
- flexible resource models
- nearly full featured developer accounts
- outstanding issues
  - server distribution performance
  - execution model efficiency
  - cluster transparency
  - better interactive job support
  - more users/resources

How to join

And more information in general

www.migrid.org