



int.eu.grid

<http://www.interactive-grid.eu>



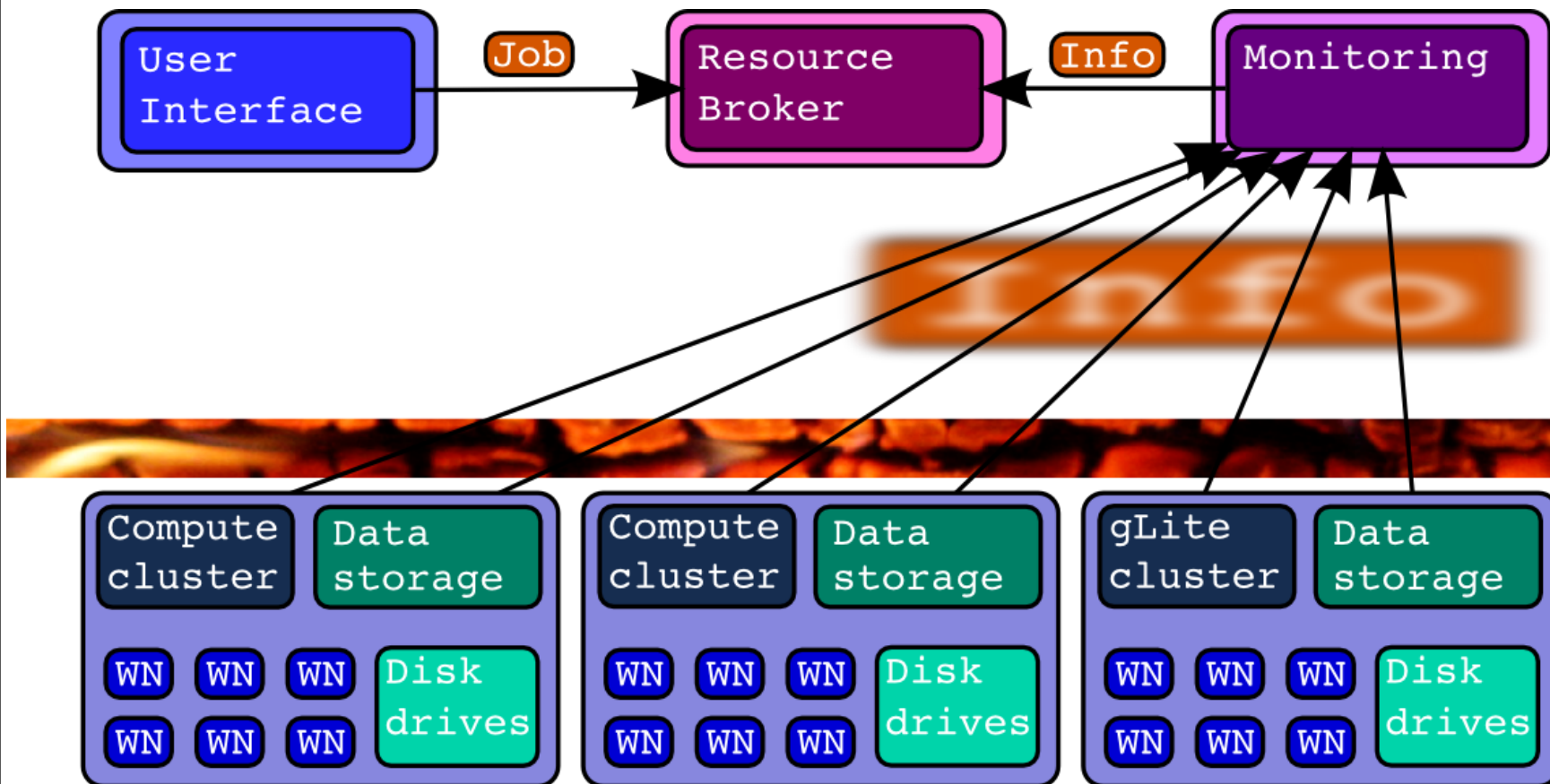
Grid Zugriff für USCT

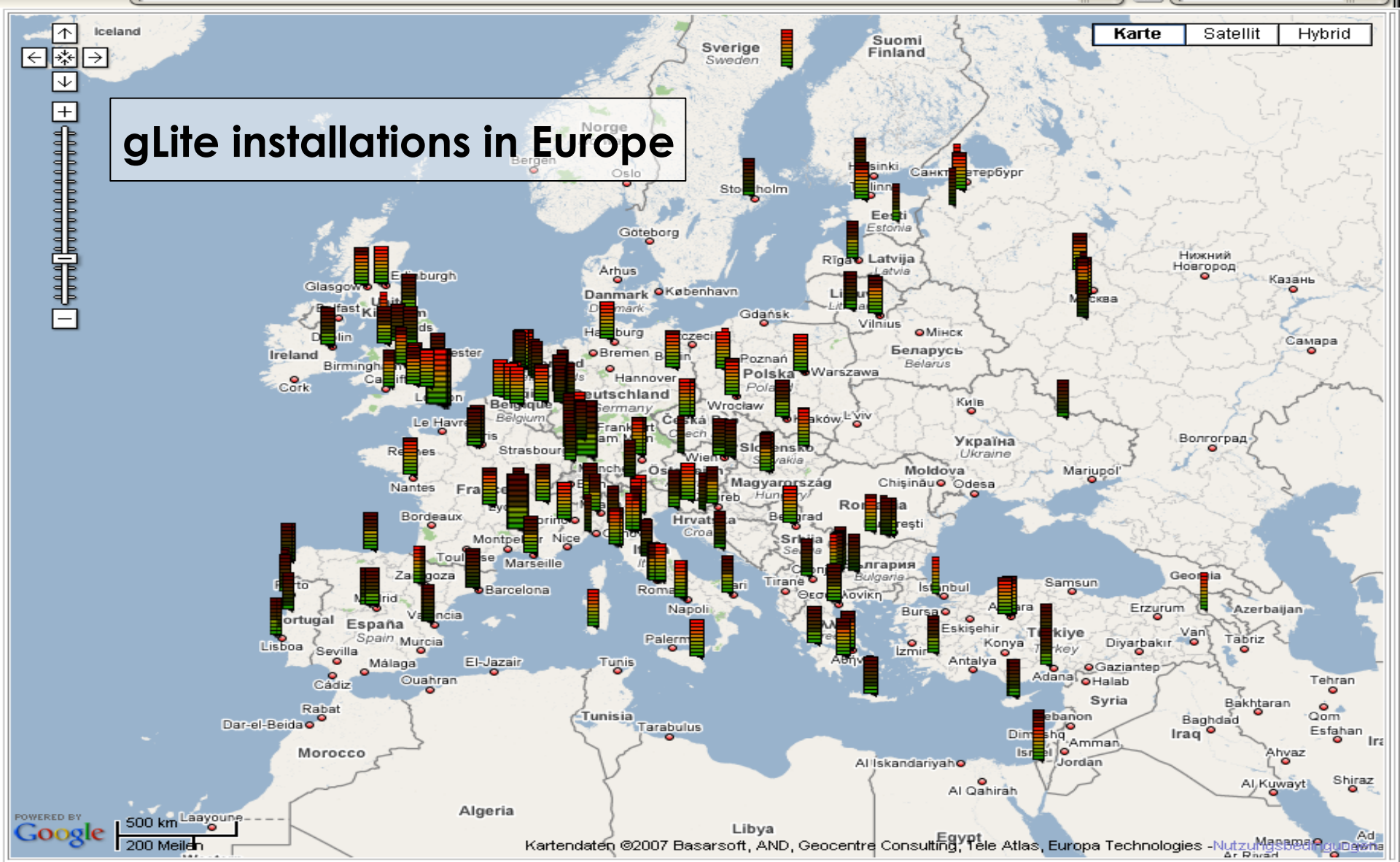
Marcus Hardt
IWR

Grid middleware gLite

- Middleware
 - := Layer between application and operating system
- gLite Development driven by CERN
 - Tools for data+computing of new accelerator
 - 10TB/year * 20years, random access
- Paradigm: **Send job to where the data is**
- Job: Self contained application
- Building blocks integrate functionality:
 - Monitoring
 - Resource Brokerage
 - Computer cluster
 - Data storage

Grid middleware gLite





- Sites: 243 (in 49 countries)
- CPUS: 42798 (176 per site)
- RAM: 19TB
- RAM/CPU: 468MB
- DISK [Tot / Avail]: [8042TB / 5408TB] ([33892GB / 22792GB] per site)

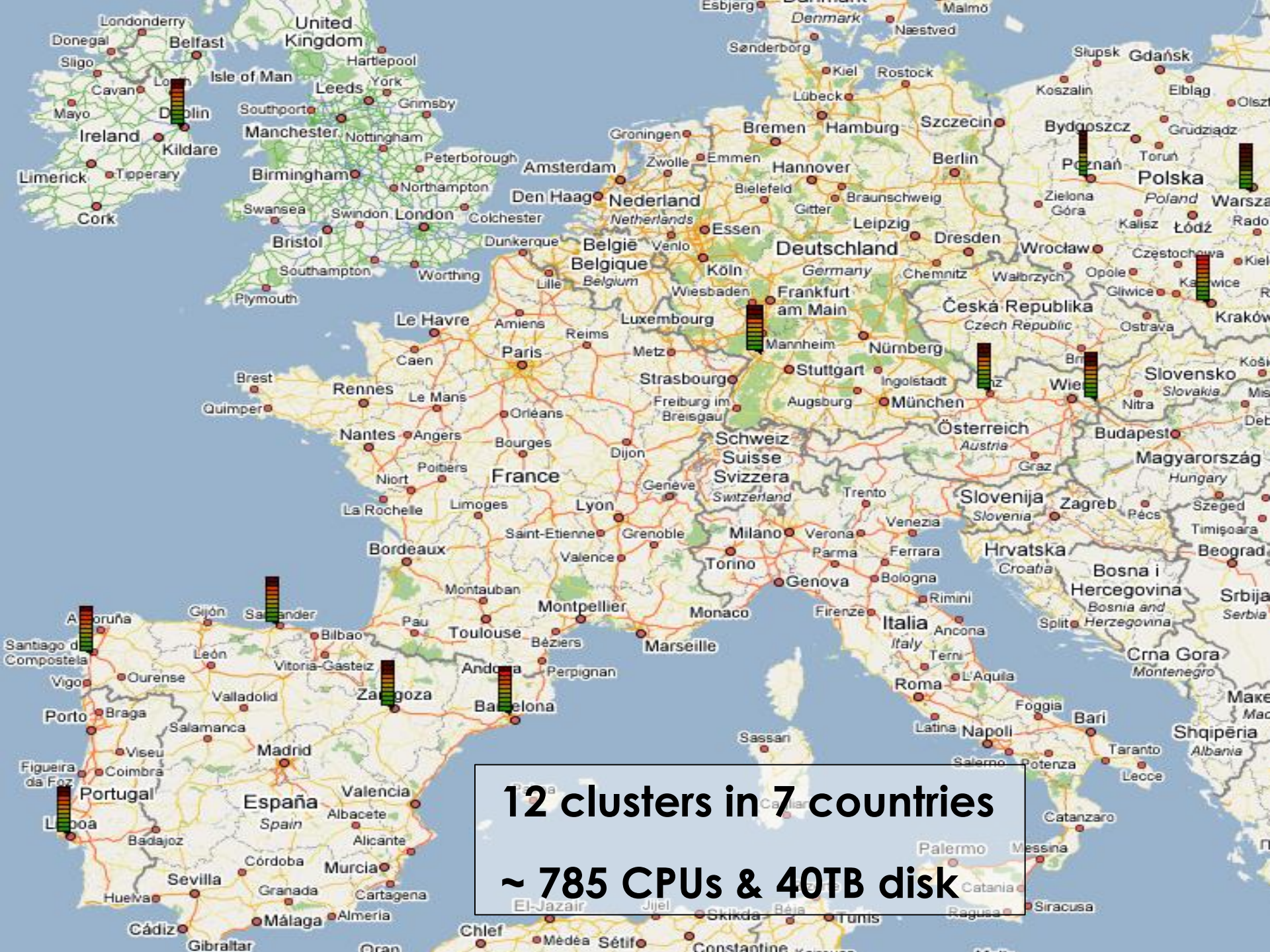


Many grid projects

- 72 grid-related EU-projects
- Why so many small grids?
 - Own the infrastructure
 - Know their owners
 - Influence on development
- FZK involvement:
 - EGEE
 - DGRID (Rainer Stotzka et.al.)
 - **Interactive European Grid Project**
 - = **int.eu.grid**
 - = **i2g**

Interactive grid

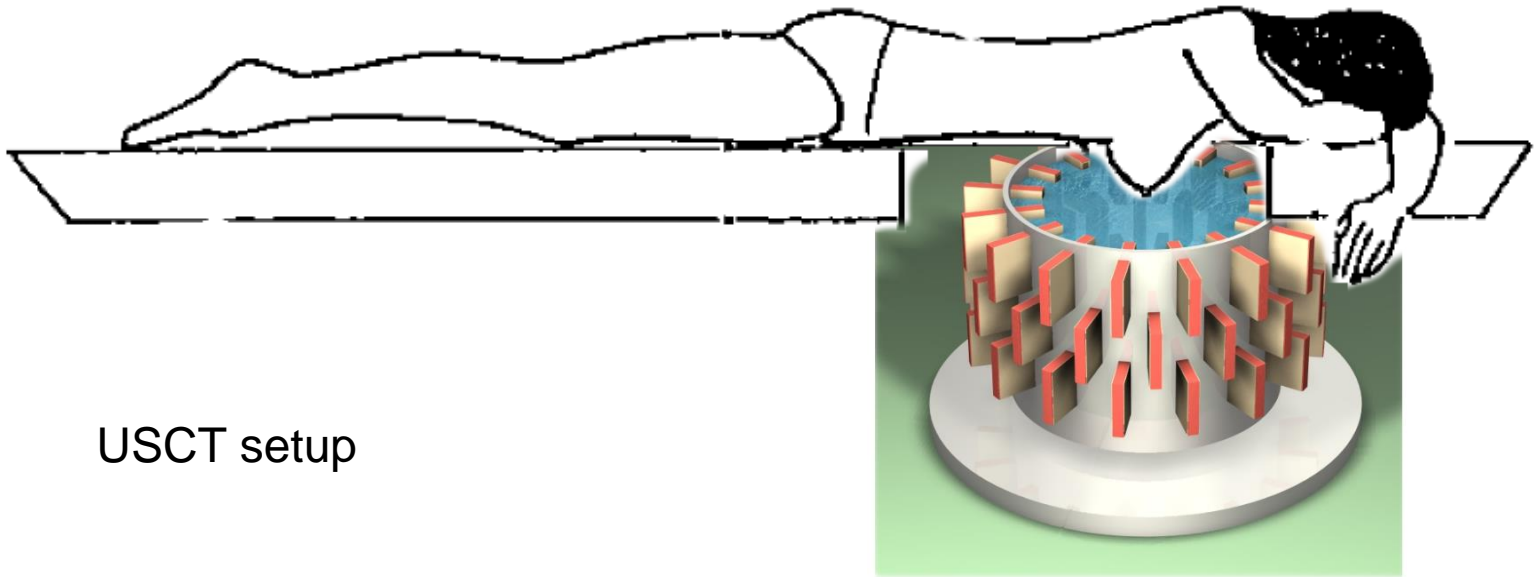
- 2 Year Project (May'06 - April'08)
- ~20 people
- Mission
 - 100% gLite compatible extensions
 - Bring grid to new user communities
 - Improve useability
- Application areas
 - Fusion
 - Medicine (USCT)
 - Environment
 - Astrophysics



12 clusters in 7 countries
~ 785 CPUs & 40TB disk

The application

- The application: Ultrasound CT (USCT)
 - New method for medical imaging
 - Application: Breast cancer diagnosis



USCT setup

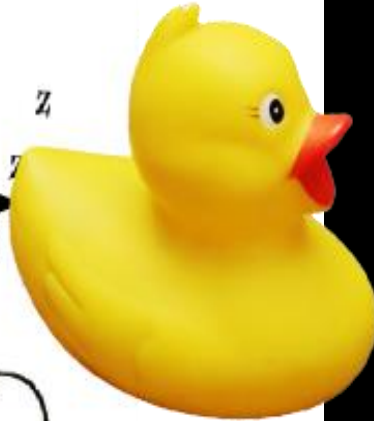
USCT reconstruction := “Black Box”

- Algorithm:
 - Based on ellipsoidal backprojection (SAFT)
 - Converts ultrasound signals to 3D volume graphics
 - Input: ~ 20 GB
 - Output: ~ 8 GB
 - Computing time:
 - $4096^2(2D)/35$ MB ... $128^2 \times 100/20$ GB ... $4096^2 \times 3410/20$ GB
 - 1 hour ... 1.5 Months... 150 Years
- Matlab
 - Problem solving environment
 - similar to Maple, Mathematica, Scilab ...
 - Strategic development platform
 - But: not possible to “submit matlab to the grid”

The USCT application at FZK

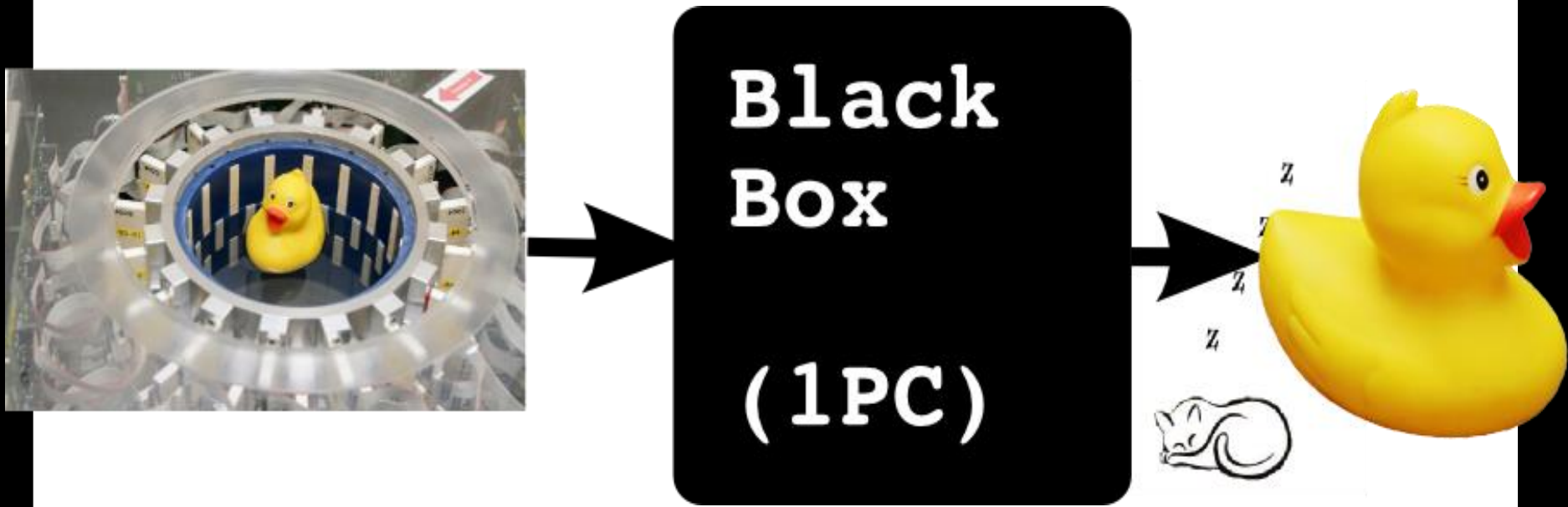


**Black
Box
(1PC)**



- Computation takes long (days, weeks, years)

The USCT application at FZK

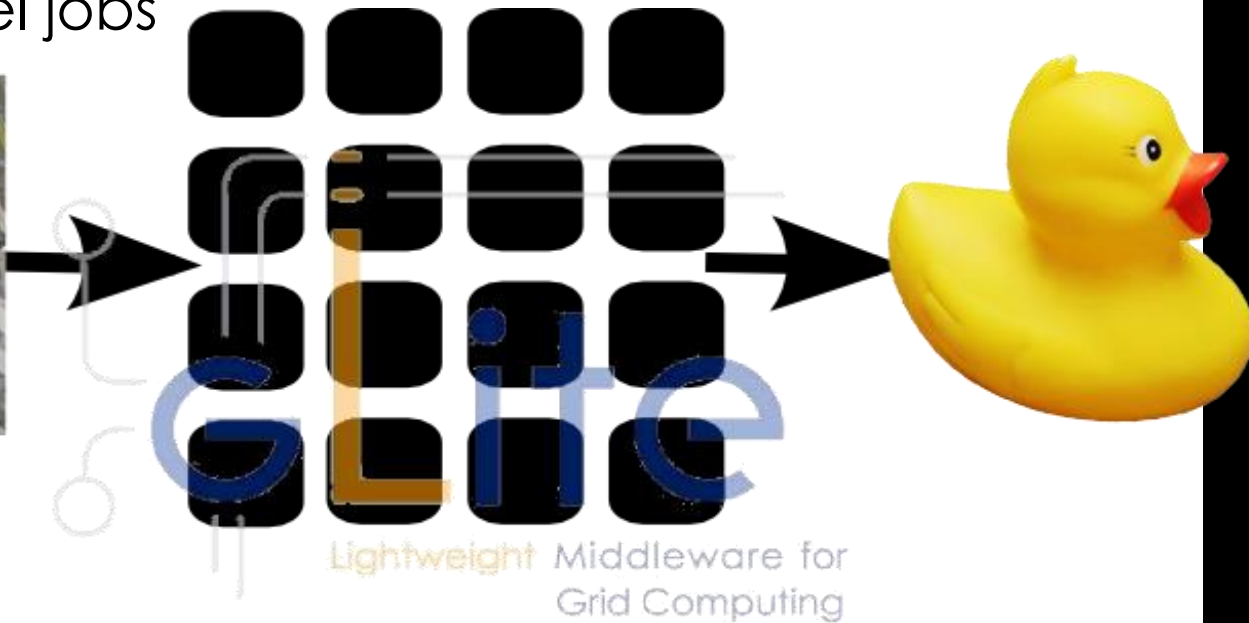
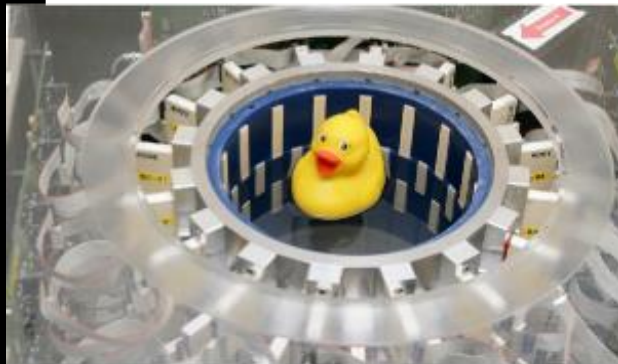


- Computation takes long (days, weeks, years)

- **Goal:**
 - Seamless, interactive, grid access
 - from Matlab

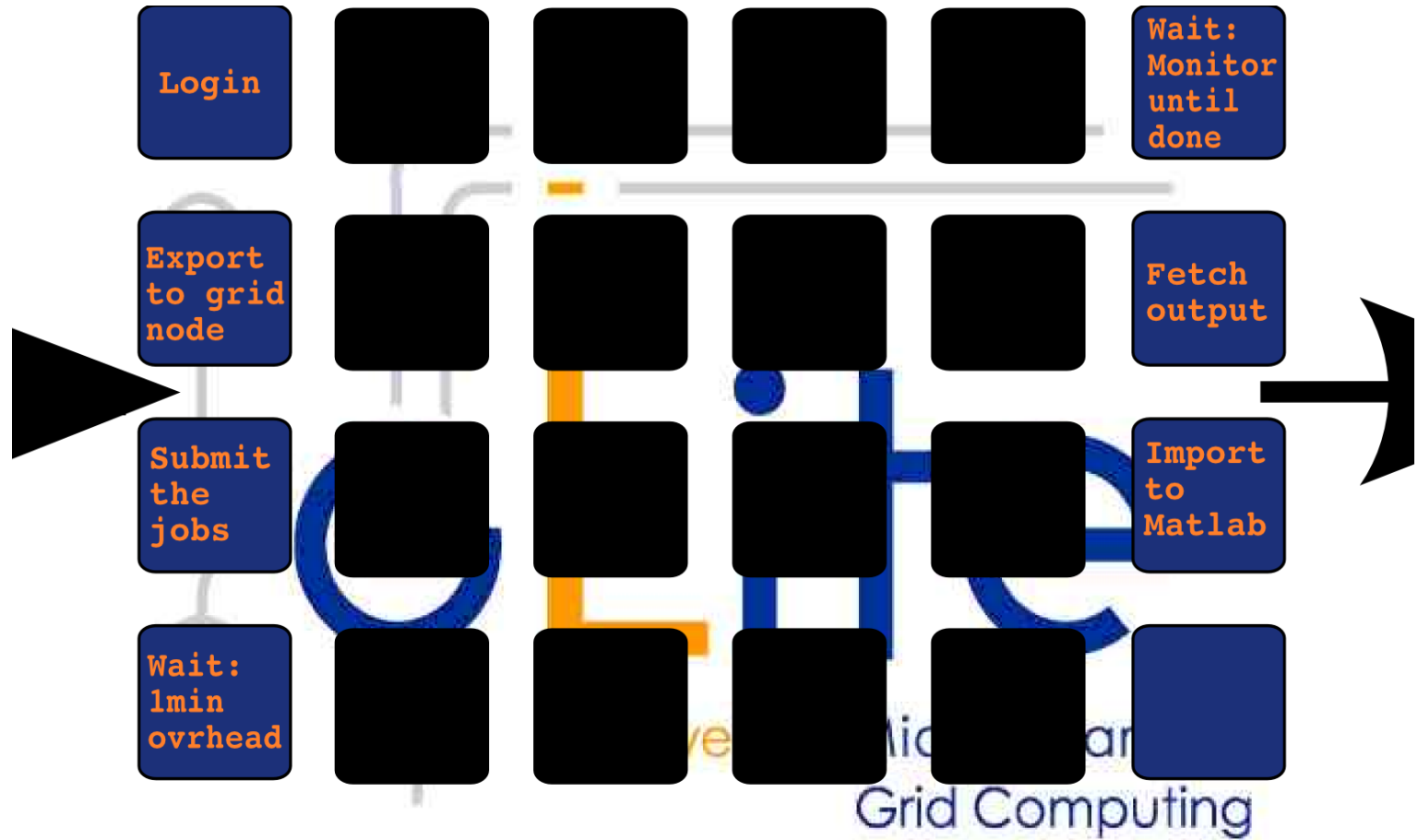
Using gLite

- Initial approach to parallel execution:
 - Partitioning of data
 - Many parallel jobs



Using gLite

- Lets take a closer look



Using gLite

- **Goal:**

- Seamless
- Interactive
- Grid access
- From matlab



What's missing?

- **Goal:**

- **Seamless**
- **Interactive**
- **Grid access**
- **From matlab**



- Seamless
 - Don't compile standalone application
- Interactive
 - No overhead (< 10 s)
 - No manual data movement
- From Matlab
 - Run Matlab-functions remotely

What's missing?

- **Goal:**

- Seamless
- Interactive
- Grid access
- From matlab



- Seamless

- Don't compile standalone application

- Interactive

- No overhead (< 10 s)
- No manual data movement

- From Matlab

- Run Matlab-functions remotely

- **Beispiel:**

Grosse Excel-Tabelle

- Excel soll lokal laufen
- Berechnungen im Grid

Improving grid access with GridSolve



- GridSolve



- Client interface for Java, C, Fortran, **Matlab**, Octave

- Easy to use:

`y=problem(x) <=> y=gs_call('problem', x)`

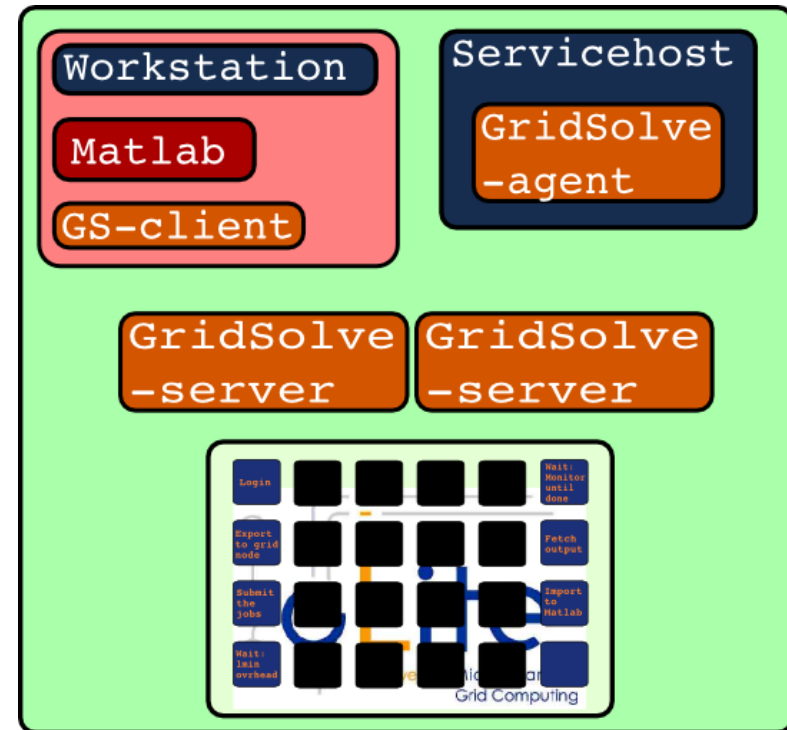
- Transport input parameters to remote side
- Execute “problem”
- Transport result back
- Implements an client/agent/server solution
- Developed at ICL, University Tennessee, Knoxville

=> Reduce complexity of the grid to one function call



How to do it?

- 1. Make Matlab run on gLite
- 2. Integrate GridSolve with gLite

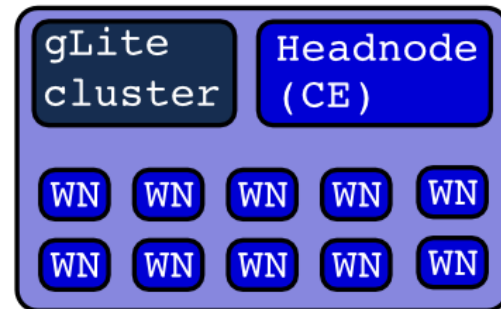
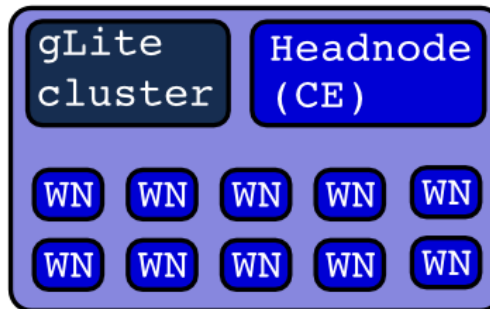
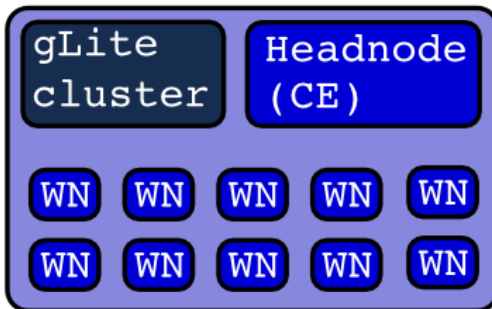
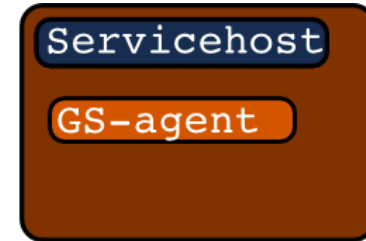
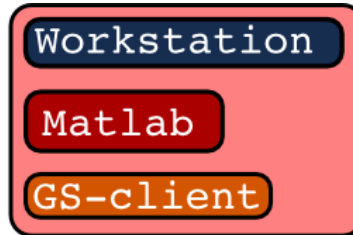
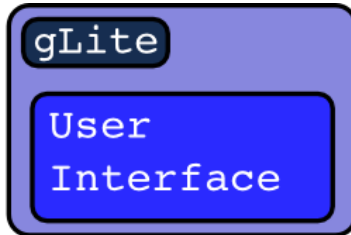


=> **Grid in Matlab using Gridsolve & RPC**
GIMGER

GridSolve (GS)/gLite integration

- Create GS-service hosts (GS-agent)
- Send 100s of GS-servers to gLite infrastructure
 - Setup build infrastructure
 - Package GridSolve
 - Create gLite jobs
 - Install GridSolve on WorkerNodes (WN)
- Ensure network connectivity
 - GS-client, GS-agent, GS-proxy, GS-server

GridSolve startup on gLite



GridSolve ready for action

gLite
User Interface

Workstation
Matlab
GS-client

Servicehost
GS-agent



gLite cluster Headnode
GS-proxy

WN WN WN WN WN
WN WN WN WN WN

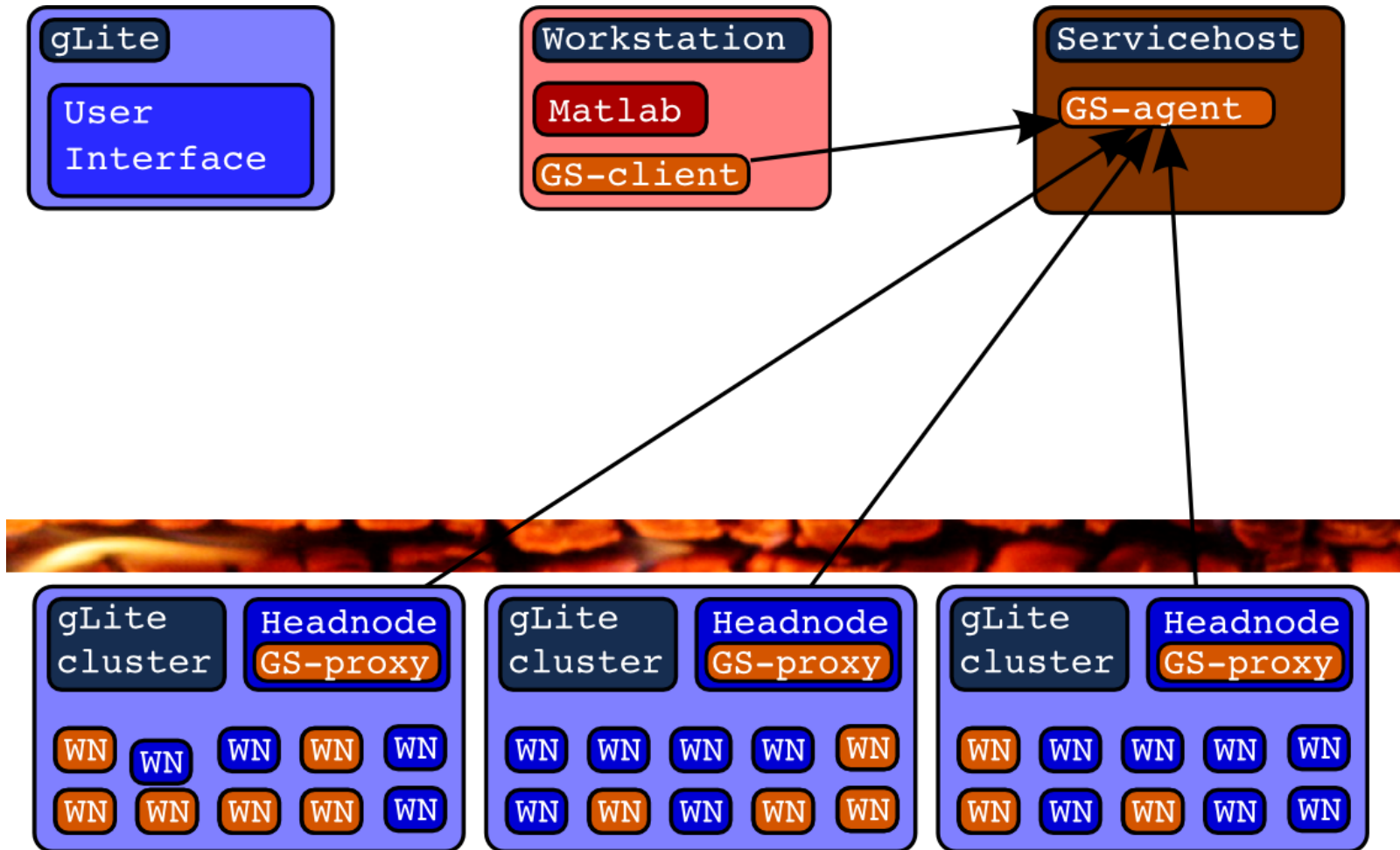
gLite cluster Headnode
GS-proxy

WN WN WN WN WN
WN WN WN WN WN

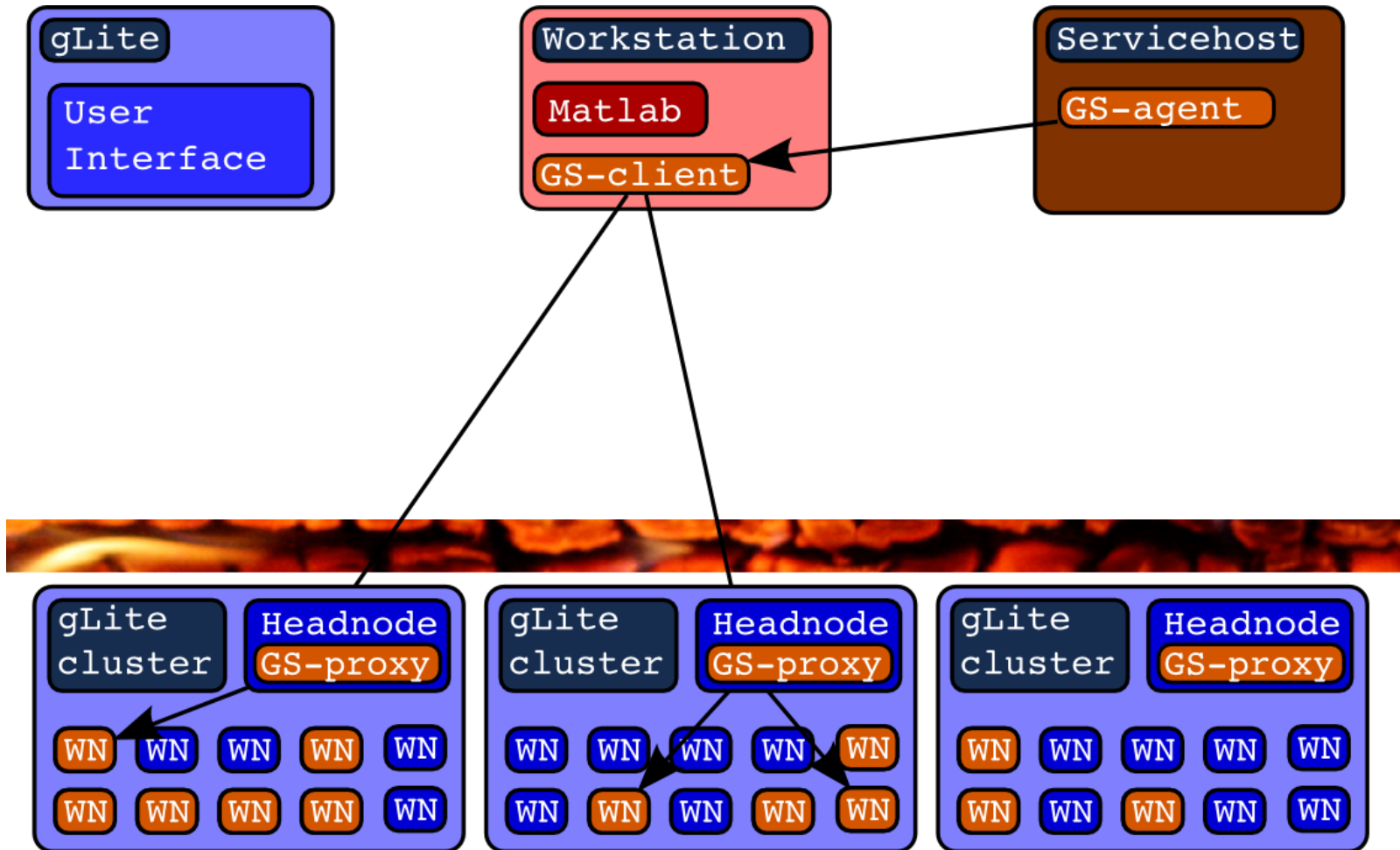
gLite cluster Headnode
GS-proxy

WN WN WN WN WN
WN WN WN WN WN

GridSolve ready in action

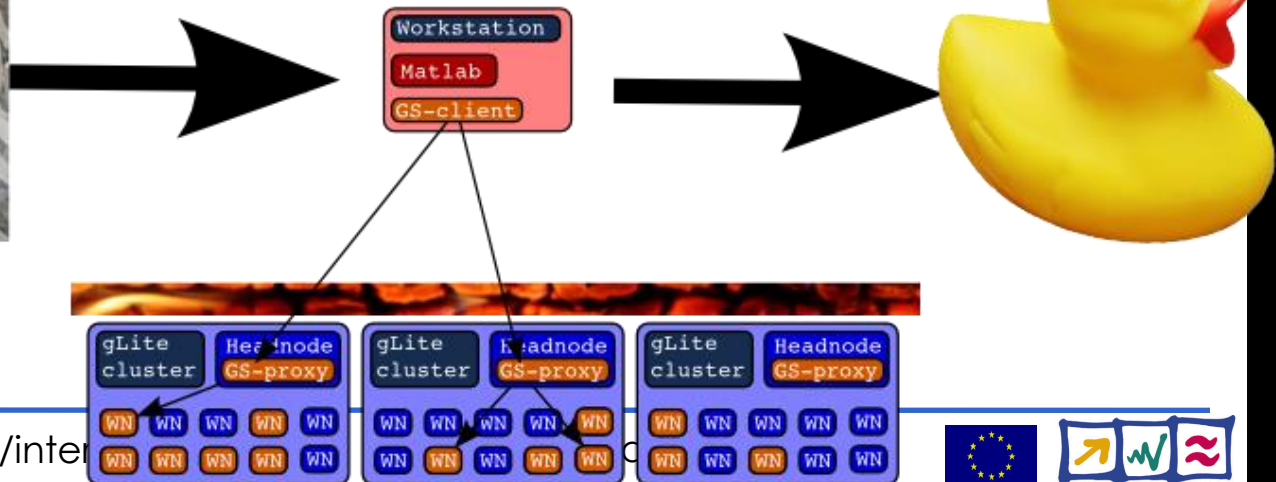


GridSolve ready in action



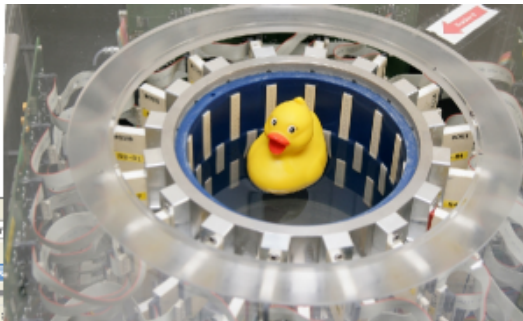
Putting things together

- GridSolve + gLite = GIMGER



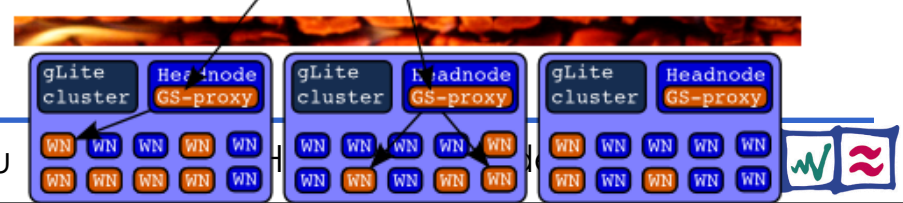
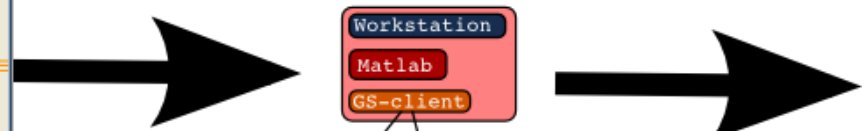
Demonstration

- Simulation: Mandelbrot fractal
- Using the same infrastructure



```

MATLAB 7.4.0 (R2007a)
File Edit Text Go Set Tools Debug Desktop Window Help
/home/marc.sziv/ncst/m...
Workspace: /home/marc.sziv/ncst/m...
Name Value
c1 <40>
47 xrange_start = task_id * rx/nustasks + 1;
48 xrange_end = (task_id+1) * rx/nustasks;
49 %
50 % /* scan C values on complex plane */
51 dcr = (cymax - crmin)/(mx-1);
52 dci = (ciymax - cimin)/(my-1);
53 for i=xrange_start:xrange_end
54     for j=1:ny
55         c_real = crmin + (i-1)*dcr;
56         c_imag = cimin + (j-1)*dci;
57 %
58 % z starts a origin */
59 z_real = 0.0;
60 z_imag = 0.0;
61 %
62 % /* set counter to zero */
63 counter = 0;
64 %
65 % /* iterate map for
66 % MAX_ITERATIONS times or ...*/
67 while ( counter < MAX_ITERATIONS )
68     z_current_real = z_real;
69     z_real = z_real * z_real - z_imag * z_imag + c_real;
70     z_imag = 2.0 * z_current_real * z_imag + c_imag;
71     counter = counter + 1;
72 %
73 % ... until magnitude of z
74 % is larger than some
75 % large radius
76 z_magnitude = z_real * z_real + z_imag * z_imag;
77 if ( z_magnitude > LARGE_RADIUS ), break, end;
78 end
79 counter = 255*counter/MAX_ITERATIONS;
80 if (counter > max_color_value), counter = max_color_value; end;
81 output (j, i-1-xrange_start) = counter;
82 end
83 end
    
```



Life-Demo

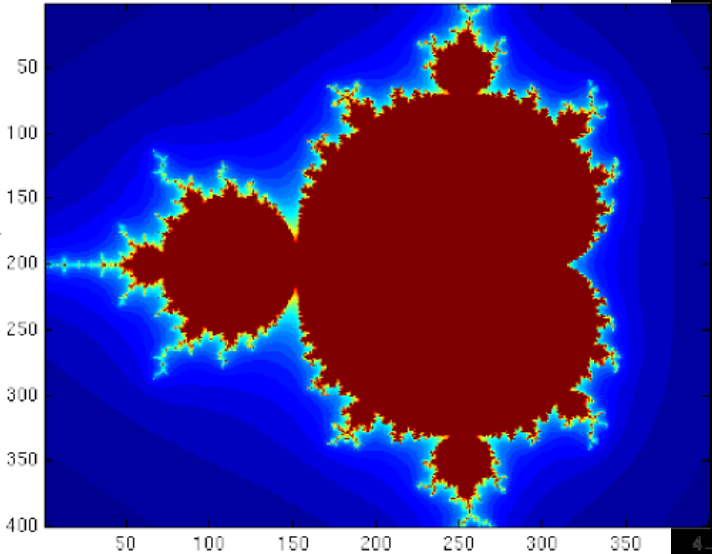
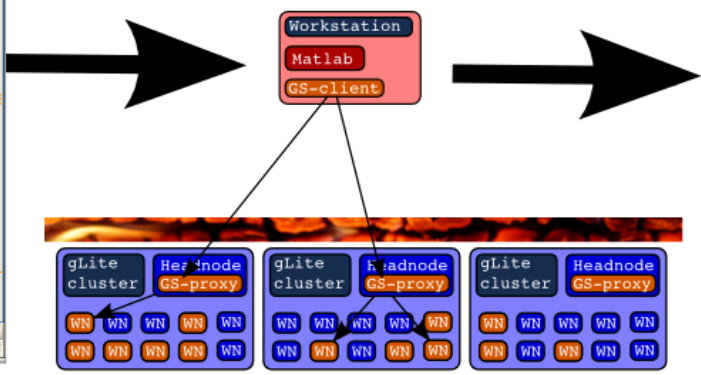
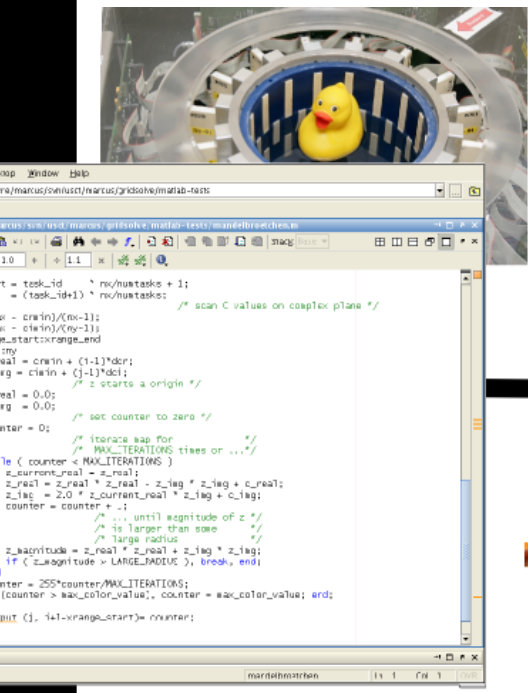


- Movie of the life demonstration:
 - <http://marcus.hardt-it.de/grid4matlab>
- **Life demo** on int.eu.grid



Result

- Simulation works
- Reasonable speedup (4x on 8 machines)



Source code



```
function f=broetchenverteiler_p (N, RESO, MAX_ITERATIONS)
for i=1:N;
    session_id(i)=gs_call_async('maendele', i-1, N, RESO, M
end
while (num_finished < N)
    for i=1:N;
        status(i)=gs_probe(session_id(i));
        if (status(i) == 0 )
            result=gs_wait(session_id(i));
        end
    end
end
end
```



Summary

- **Goal:**
 - **Seamless**
 - **Interactive**
 - **Grid access**
 - **From matlab**



- We can
 - Use the grid from matlab...
 - ... for hand-tuned functions
 - Run simple simulations in our infrastructure
- We want to...
 - Use real code
 - Cope with the data (20 GB in, 8 GB out)
 - Use gLite data handling methods
 - Identify Bottlenecks
 - Automatically send Matlab functions to the grid

