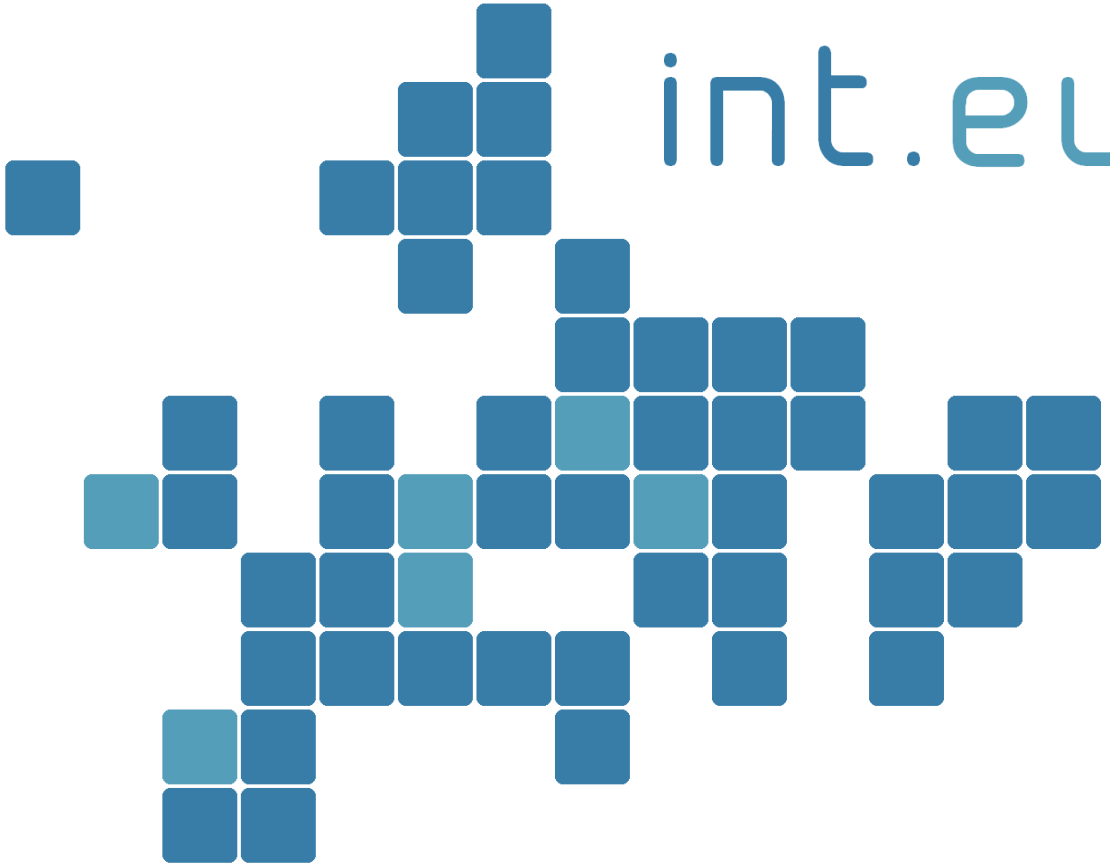


int.eu.grid

http://www.interactive-grid.eu

Grid Zugriff für USCT

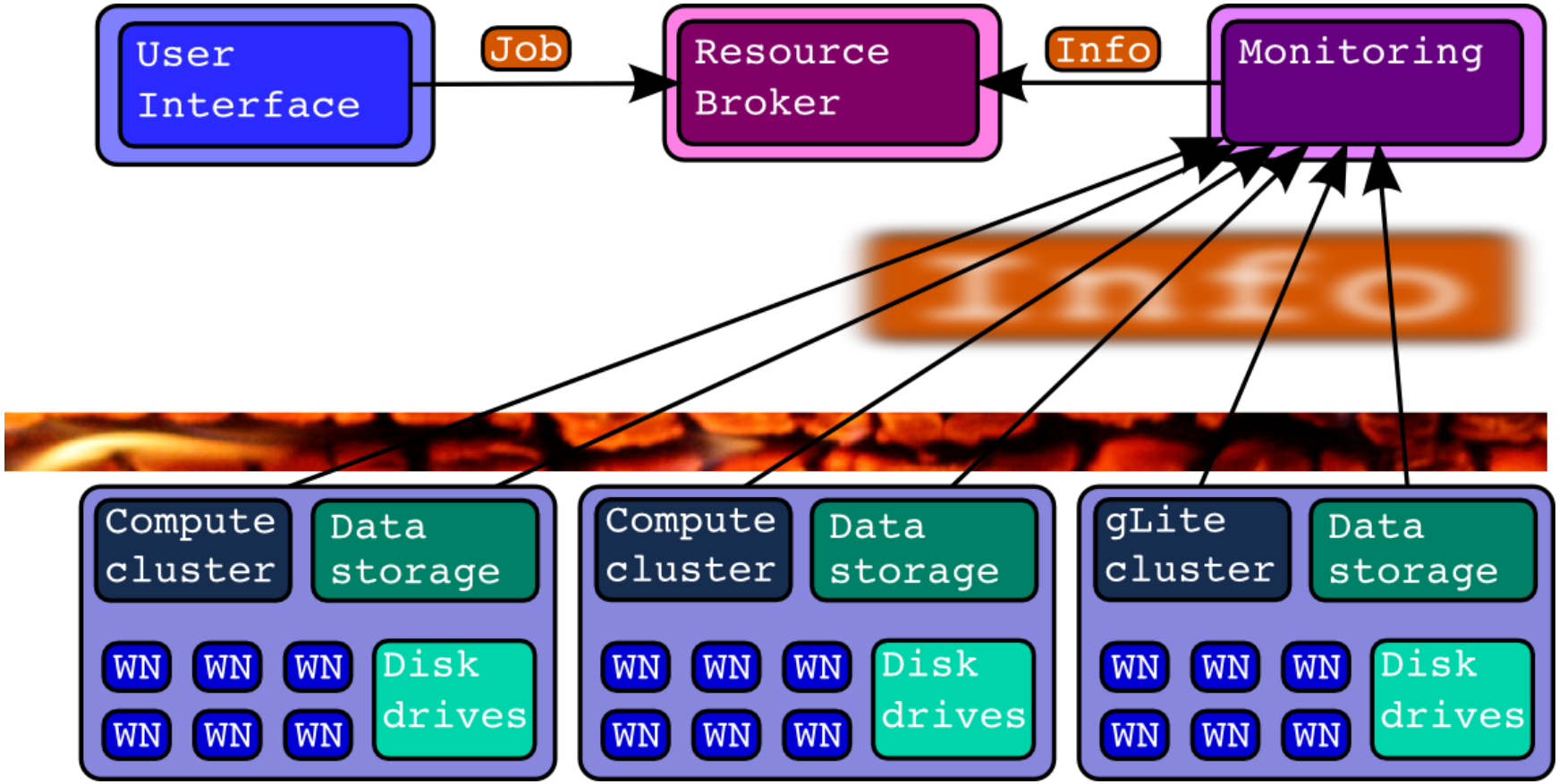
Marcus Hardt
IWR

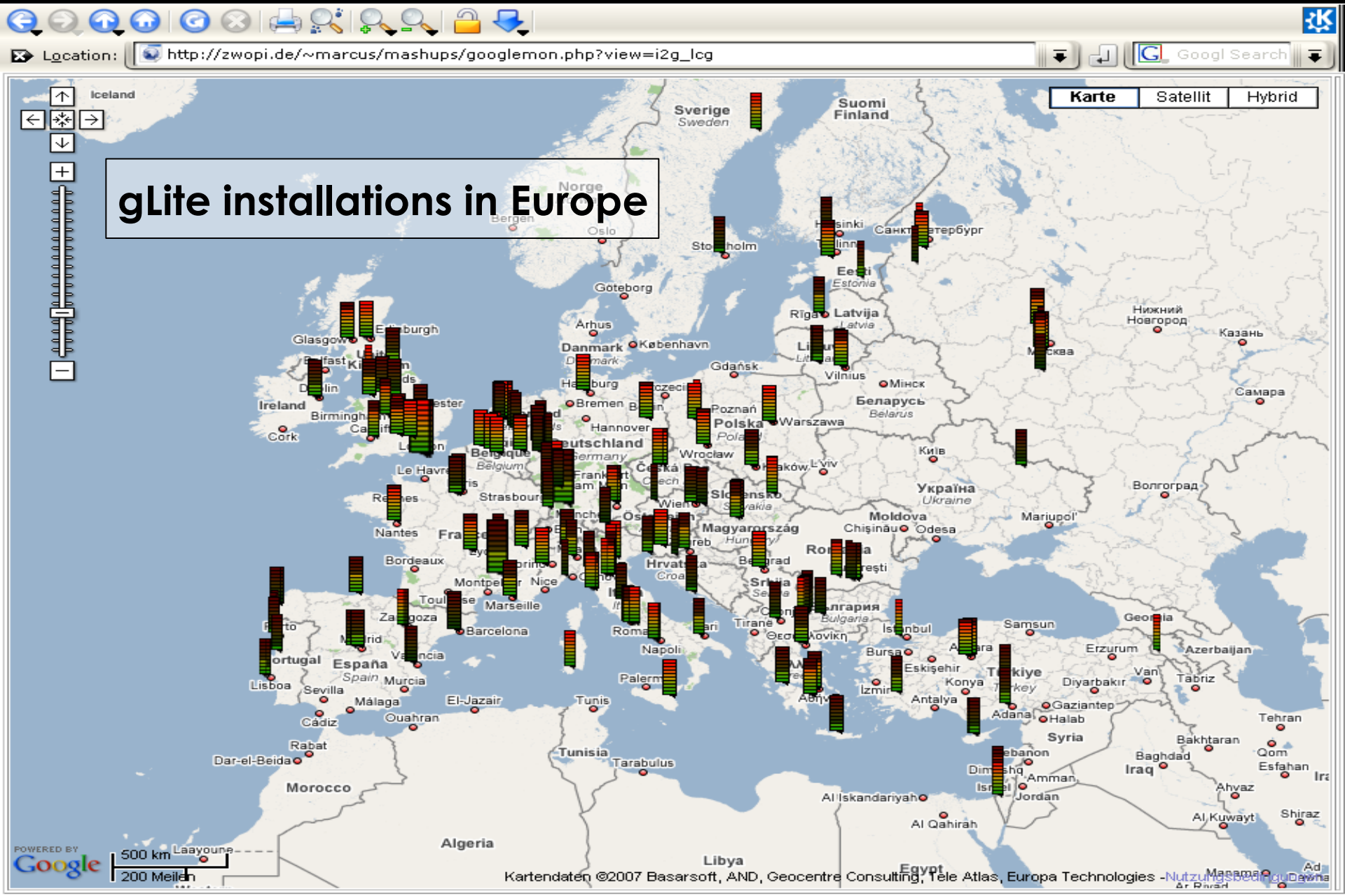
Grid middleware gLite

- 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)

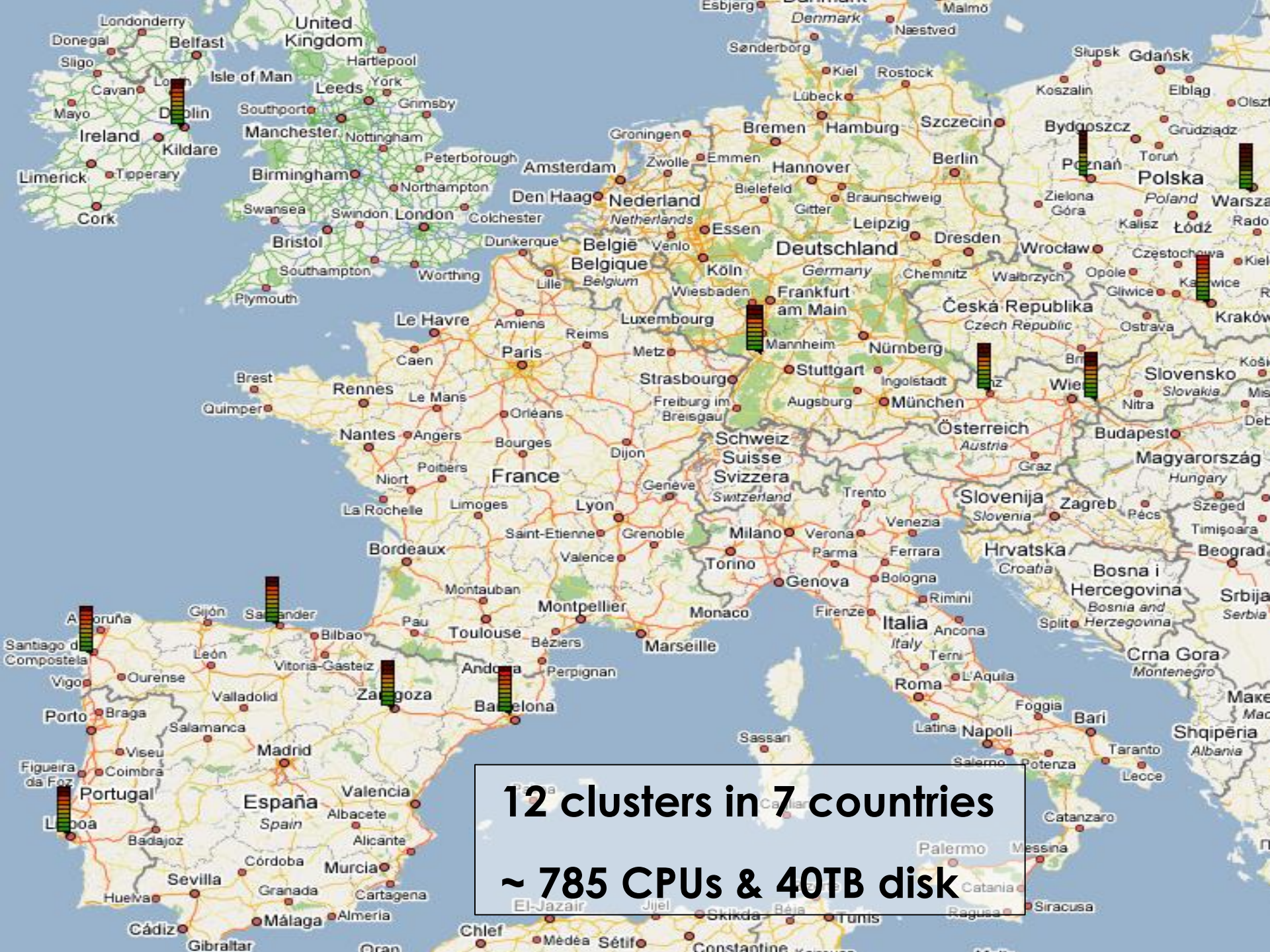
0-9 CPUs --- 10-99 CPUs --- 100-999 CPUs --- > 1000 CPUs

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**

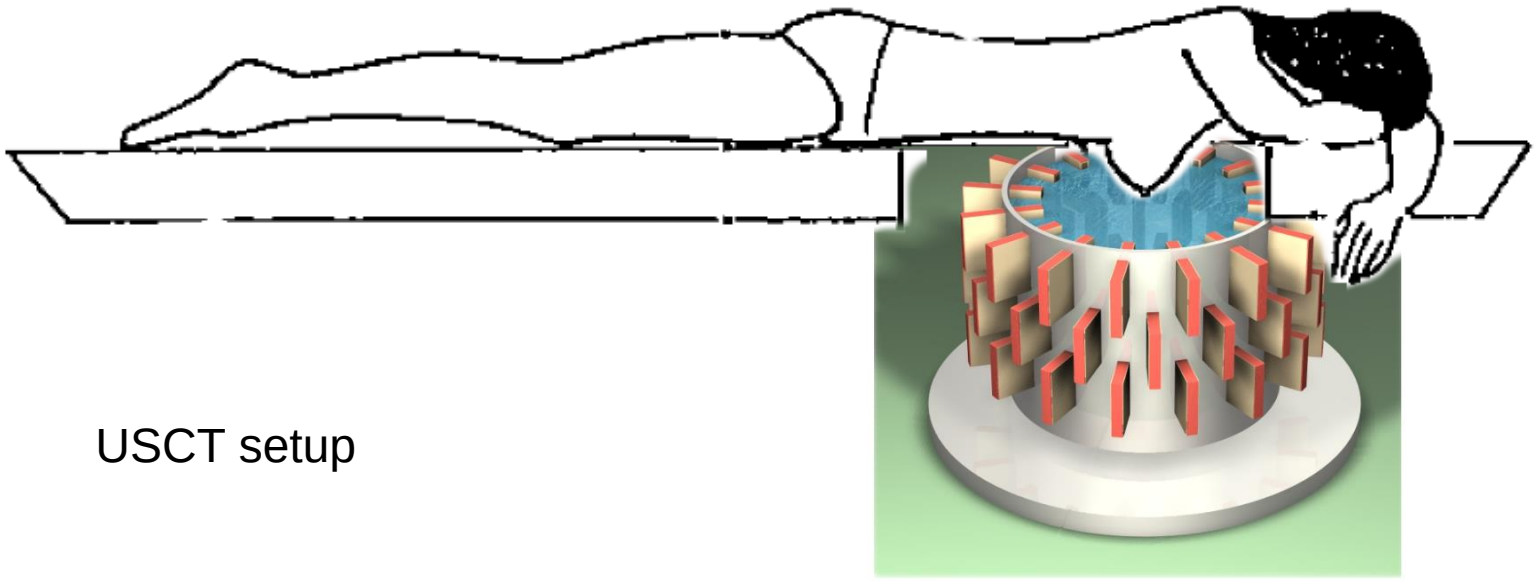
Interactive grid

- 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

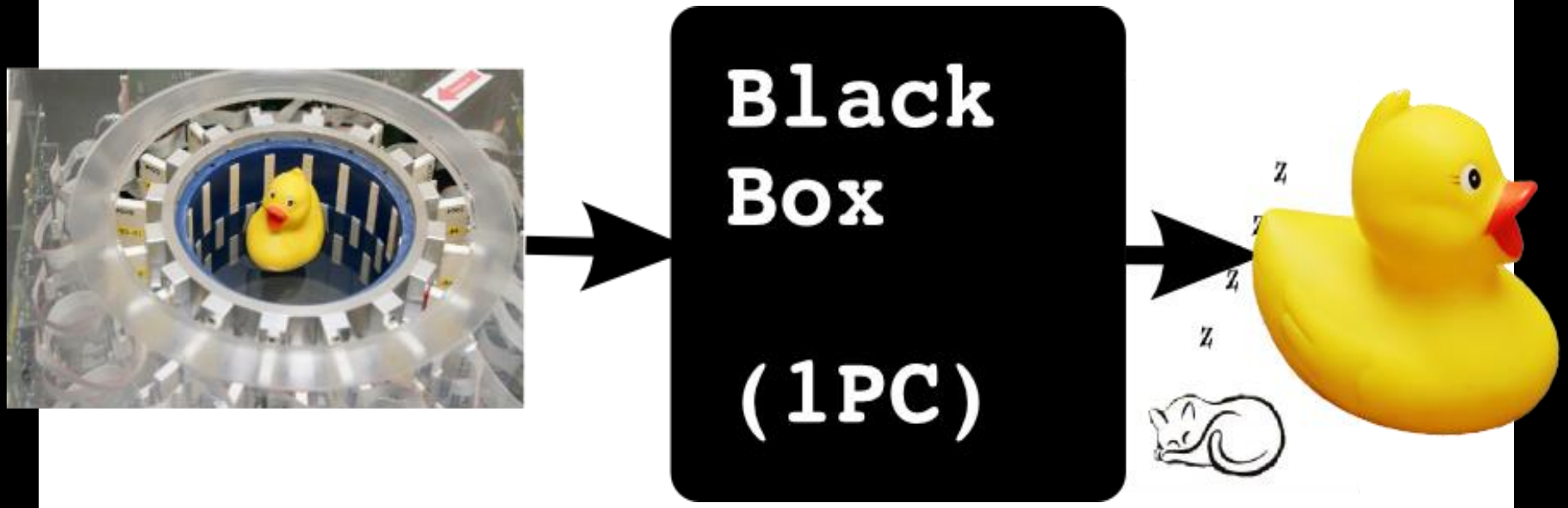


USCT setup

USCT reconstruction := “Black Box”

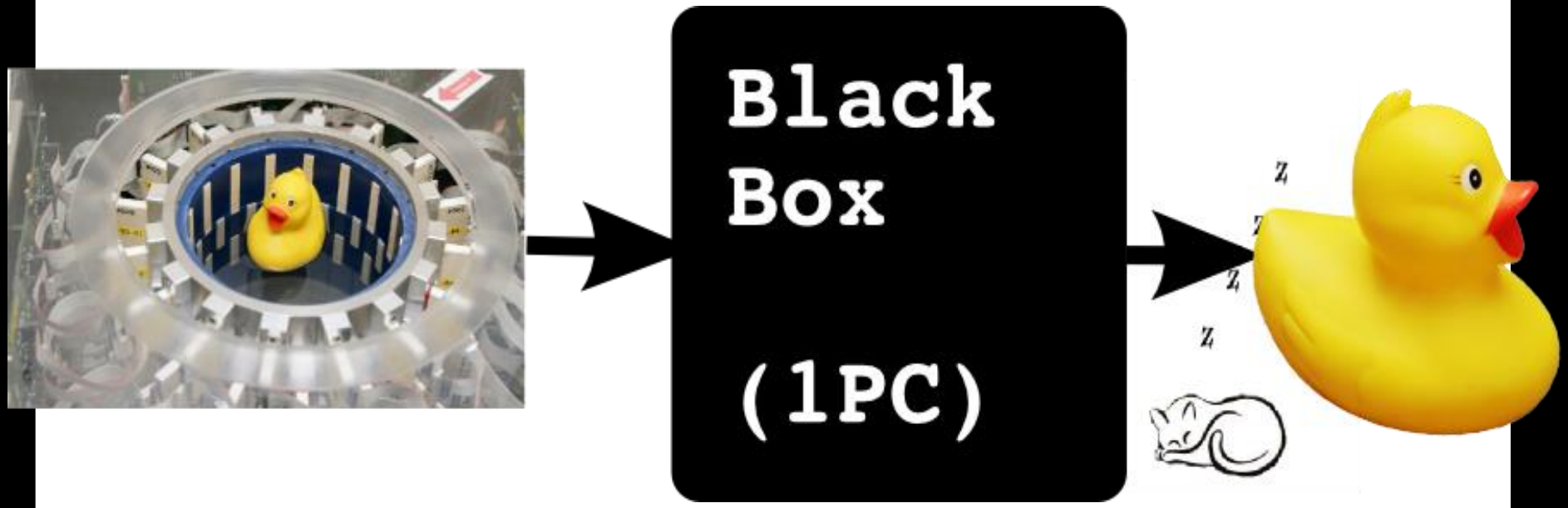
- Algorithm:
  - Based on ellipsoidal backprojection (SAFT)
  - Converts ultrasound signals to 3D volume graphics
    - Input: ~ 20 GB
    - Output: ~ 8 GB
  - Computing time:
    - 4096^2(2D)/35 MB ... 128^2 x100/20 GB ... 4096^2 x 3410/20 GB
      1hour ... 1.5 Months ... 150 Years
- Matlab
  - Problem solving environment
    - similar to Maple, Mathematica, Scilab ...
  - Strategic development platform
  - But: not possible to “submit matlab to the grid”

The USCT application at FZK



- Computation takes long (days, weeks, years)

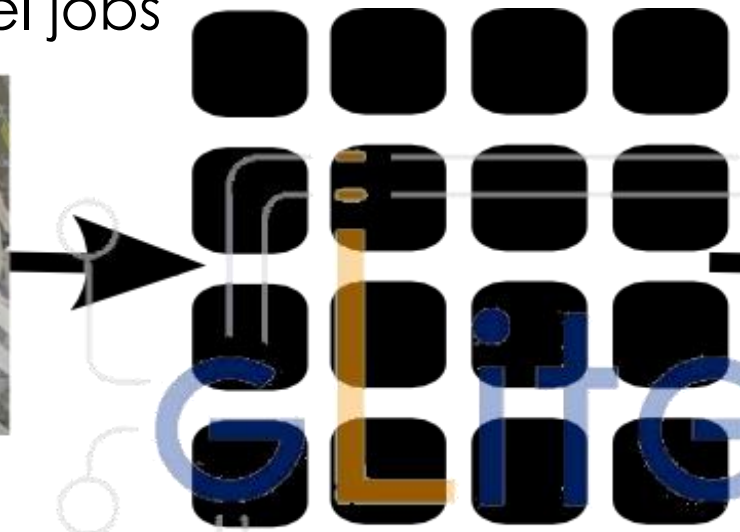
The USCT application at FZK

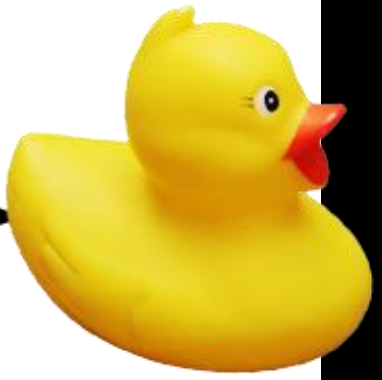


- Computation takes long (days, weeks, years)
- **Goal:**
  - **Seamless, interactive, grid access**
  - **from Matlab**

Using gLite

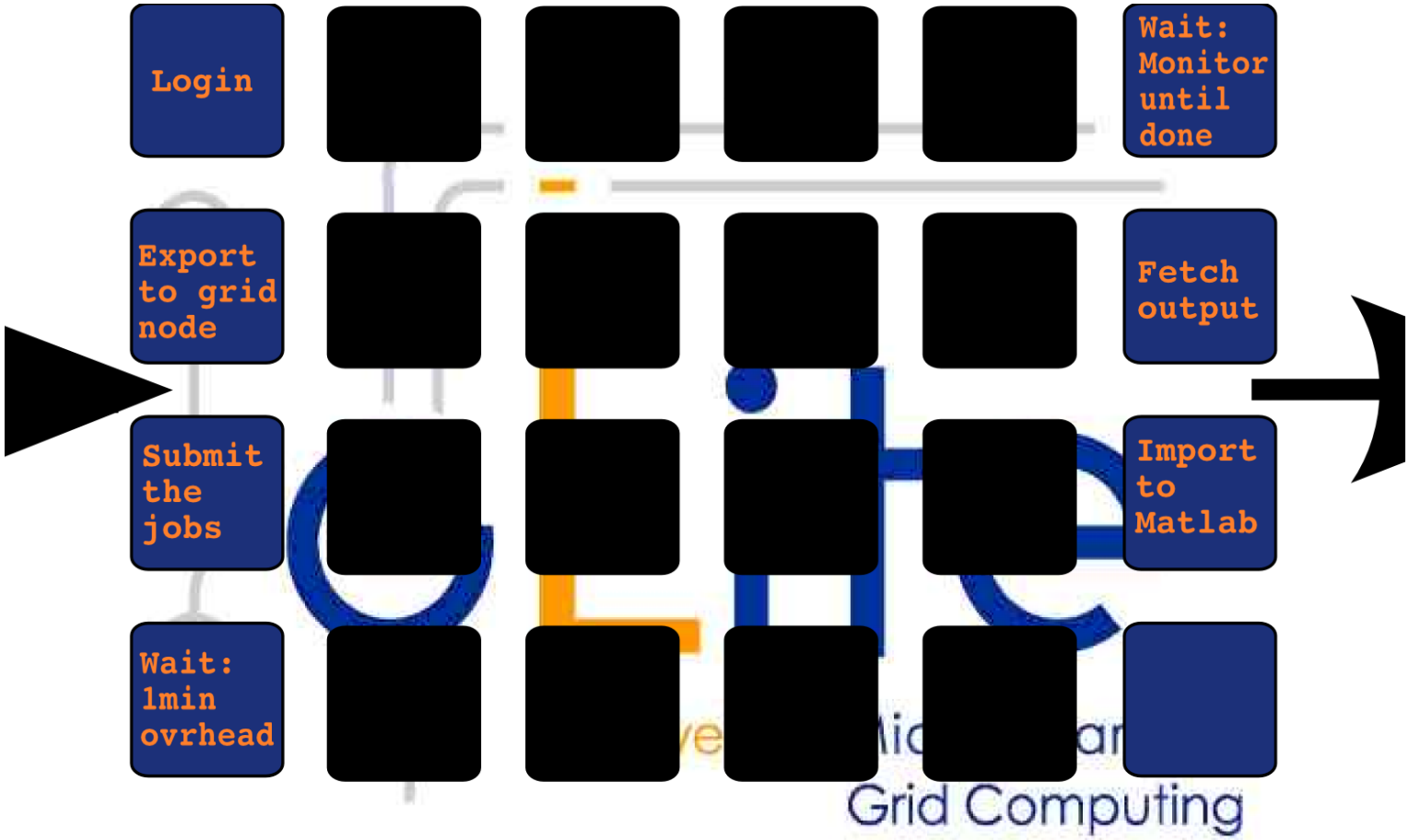
- Initial approach to parallel execution:
  - Partitioning of data
  - Many parallel jobs





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 movem
- From Matlab
  - Run Matlab-functions rer

- **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:

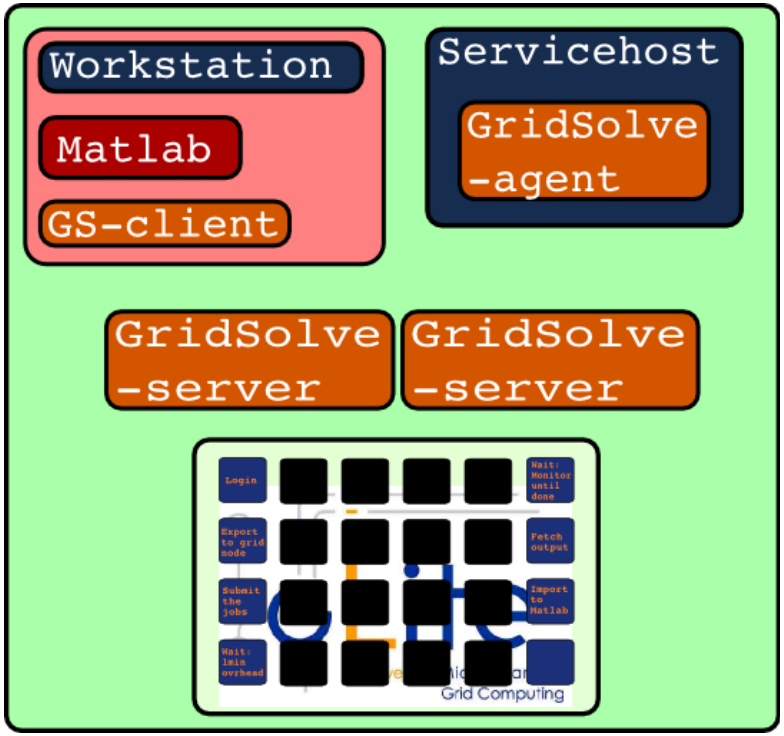
    y=problem(x) <=> y=gs_call('problem', x)

    - 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



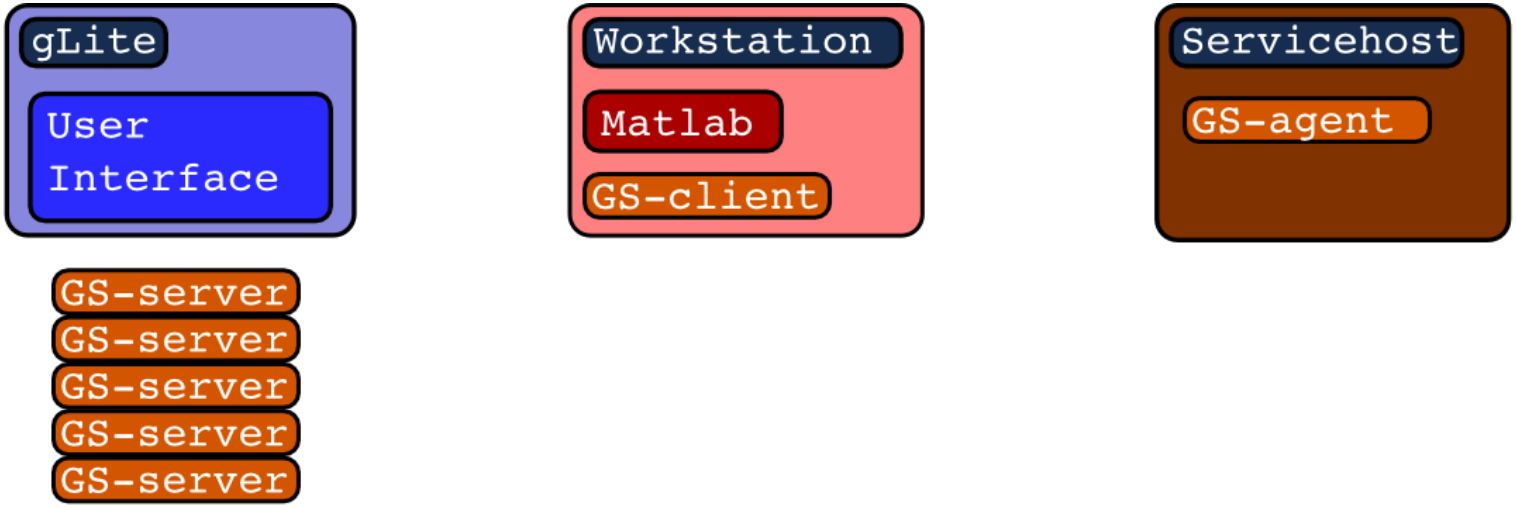
=> **G**rid **i**n **M**atlab using **G**ridsolv**e** & **R**PC

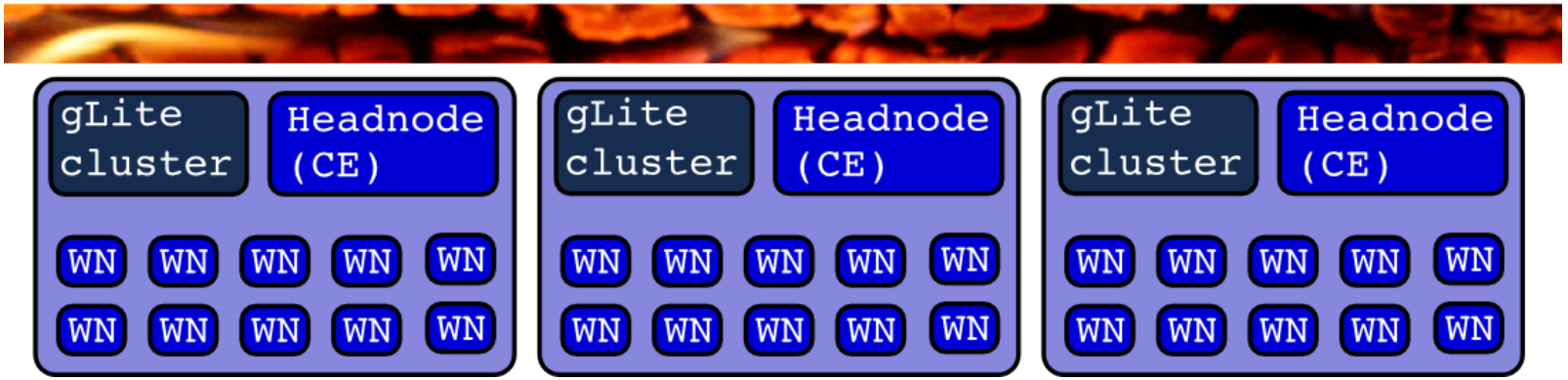
GIMGER

GridSolve (GS)/gLite integration

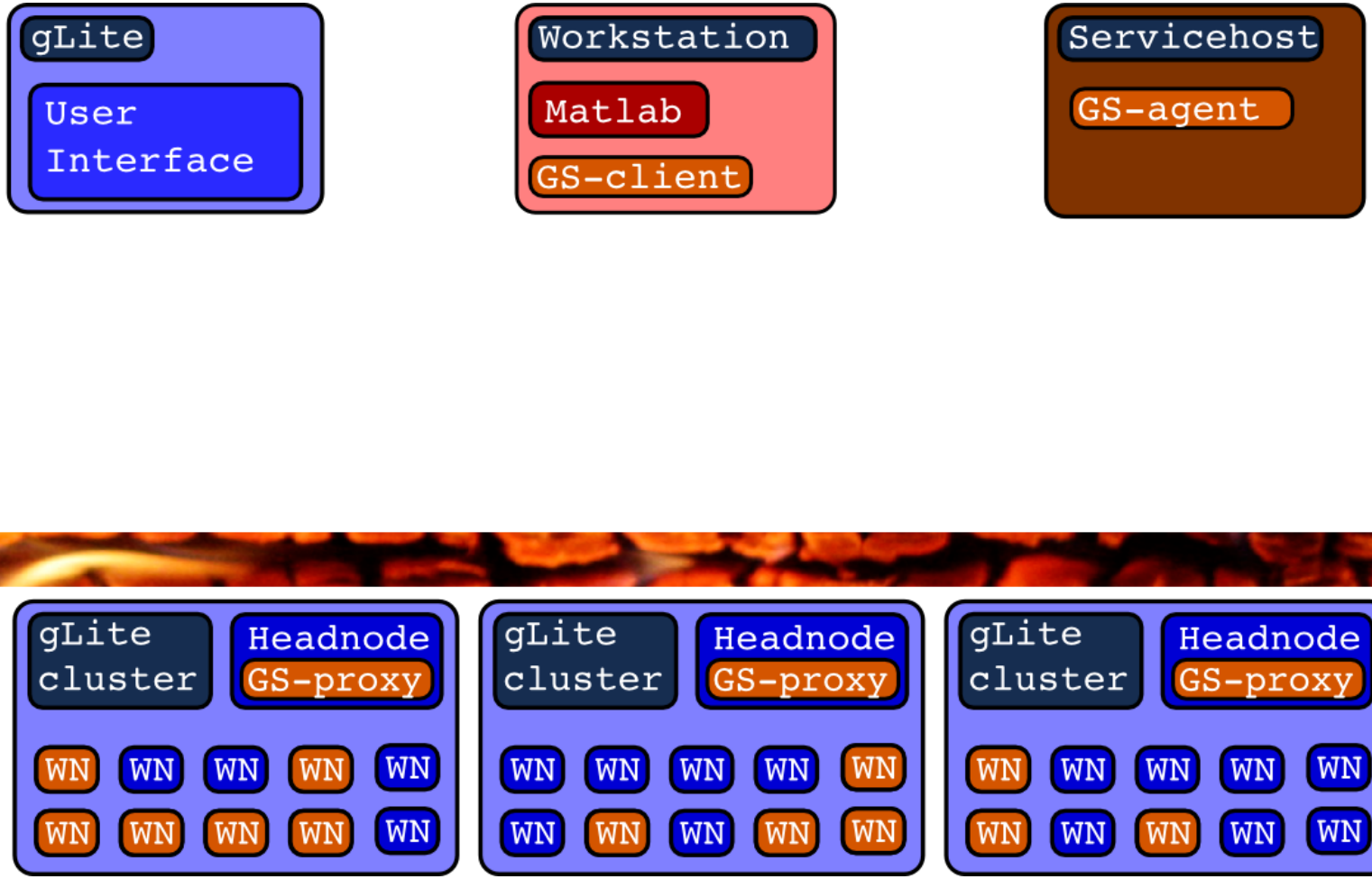
- Create GS-service hosts (GS-agent)
- Send 100s of GS-servers to gLite infrastructure
  - Setup build infrastructure
  - Package GridSolve
  - Create gLite jobs
  - Install GridSolve on WorkerNodes (WN)
- Ensure network connectivity
  - GS-client, GS-agent, GS-proxy, GS-server

GridSolve startup on gLite

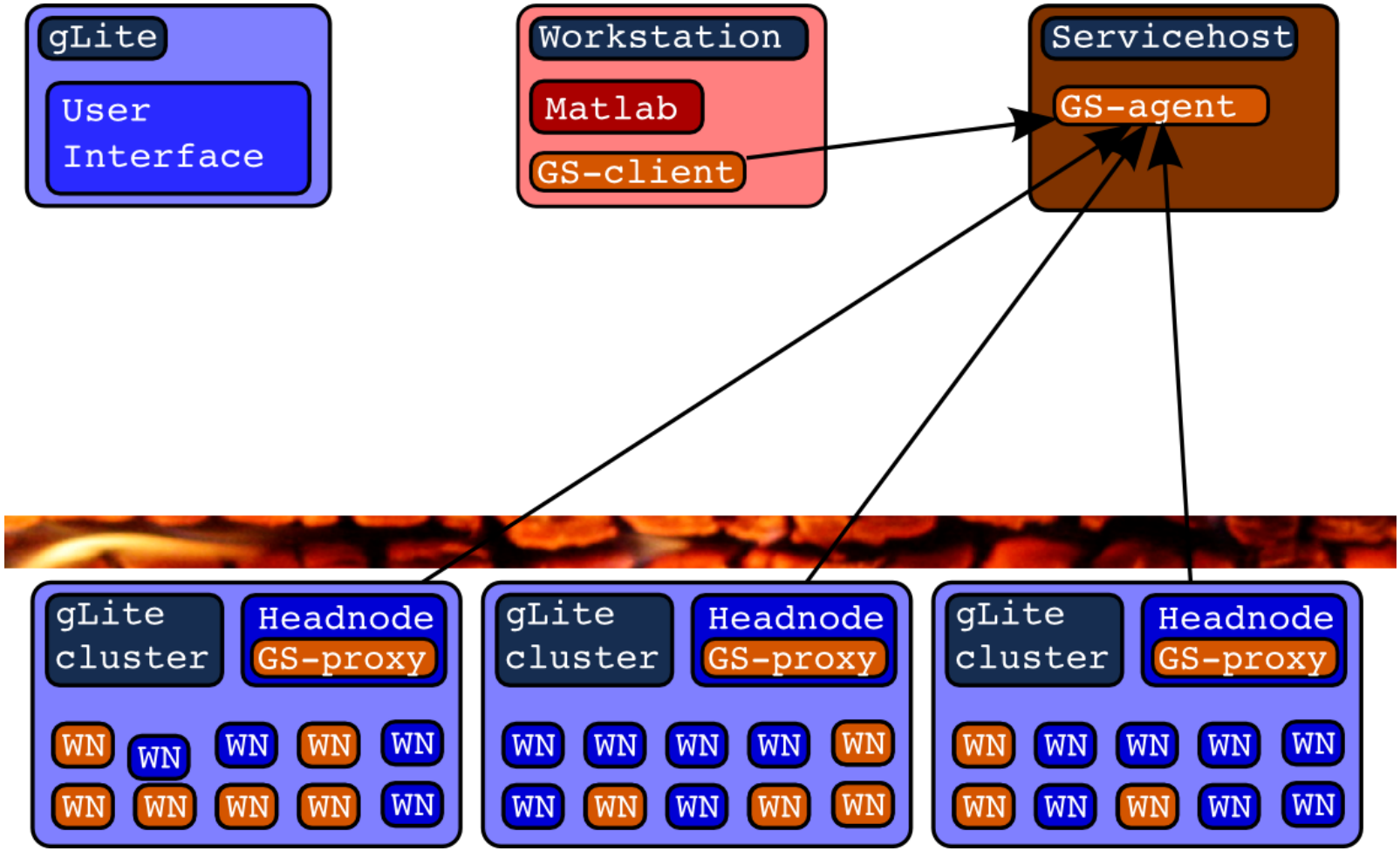




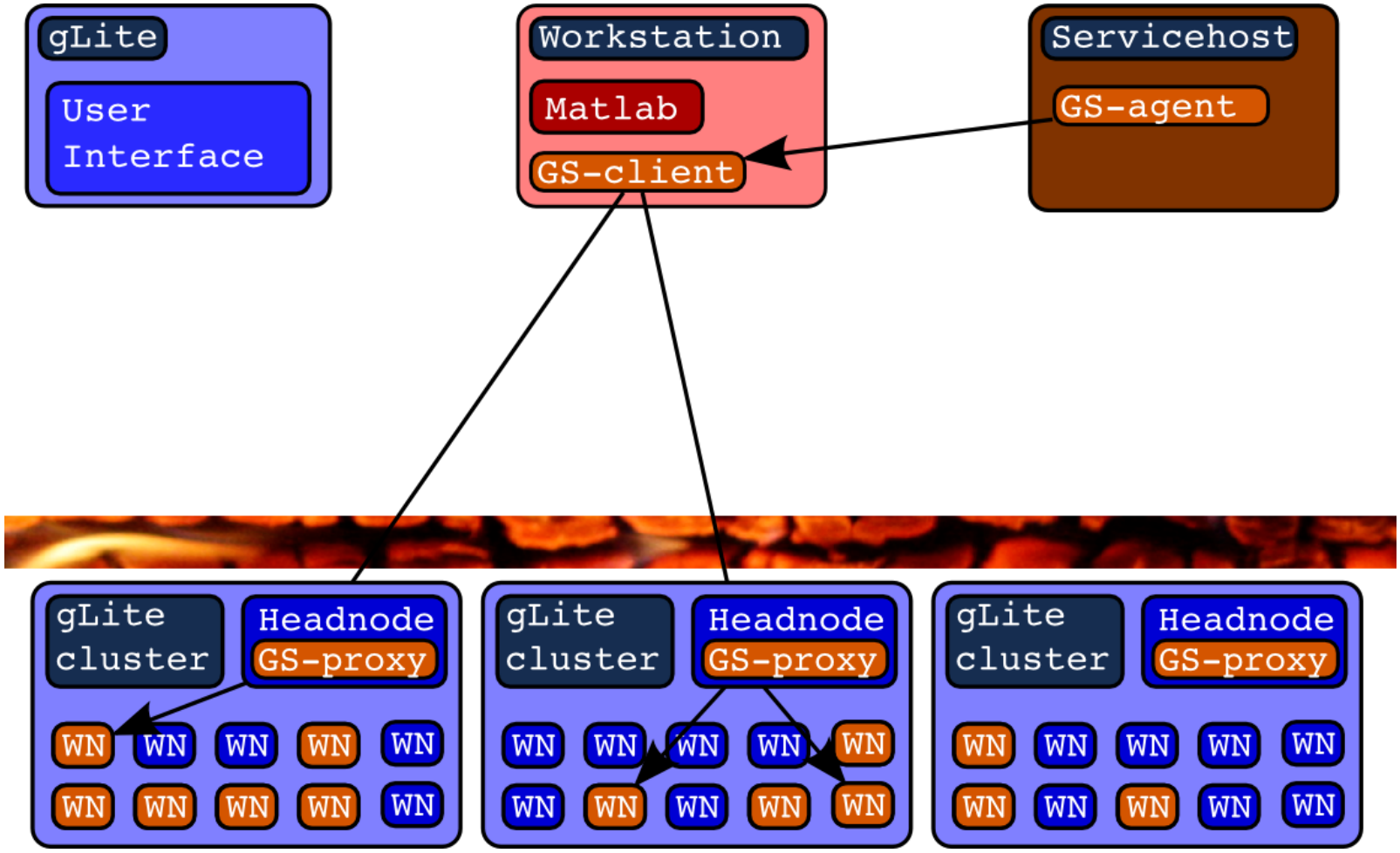
GridSolve ready for action



GridSolve ready in action

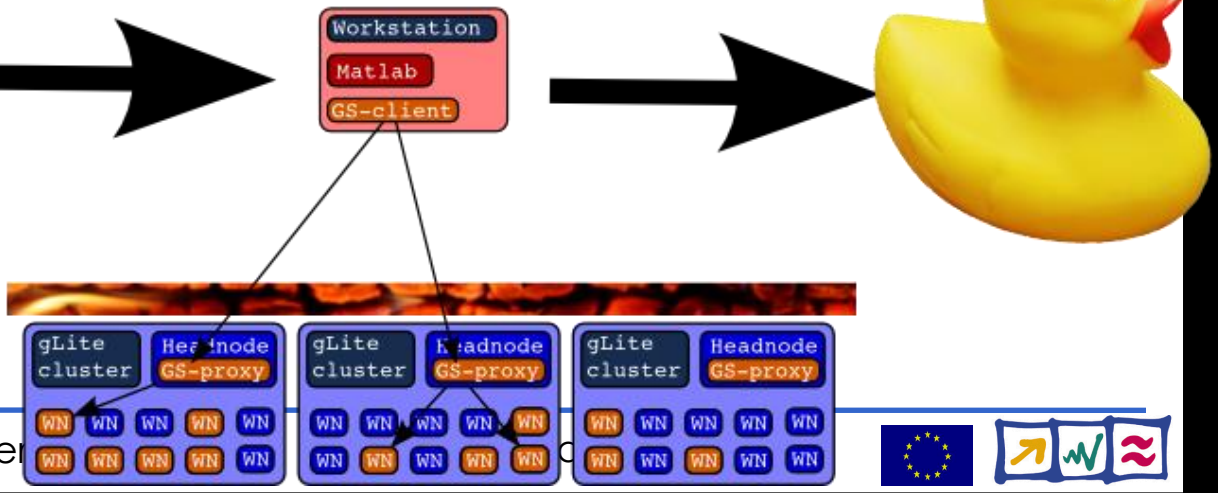


GridSolve ready in action



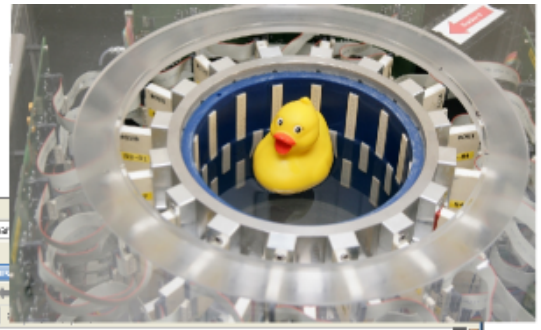
Putting things together

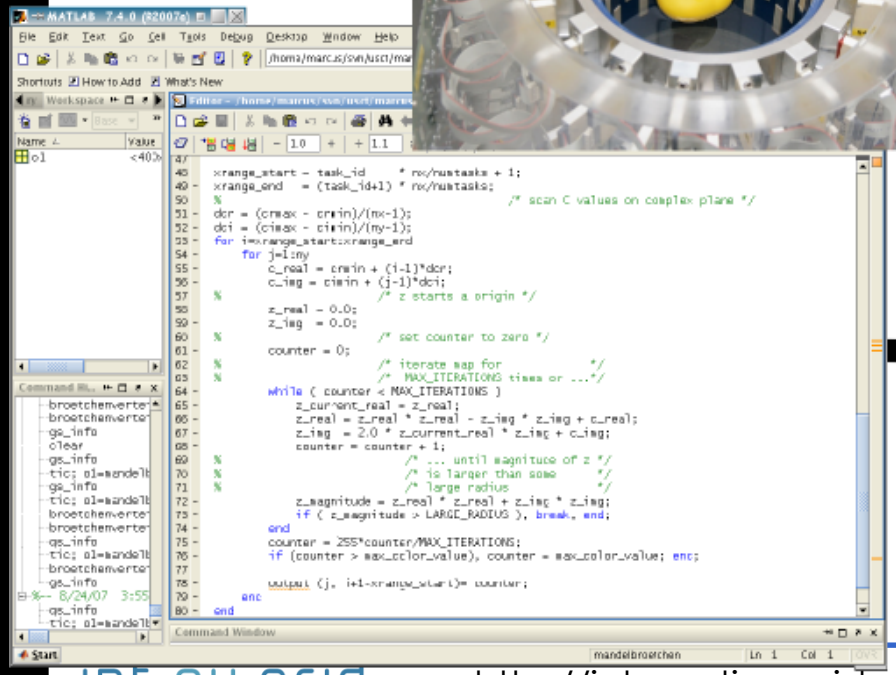
- GridSolve + gLite
  = GIMGER

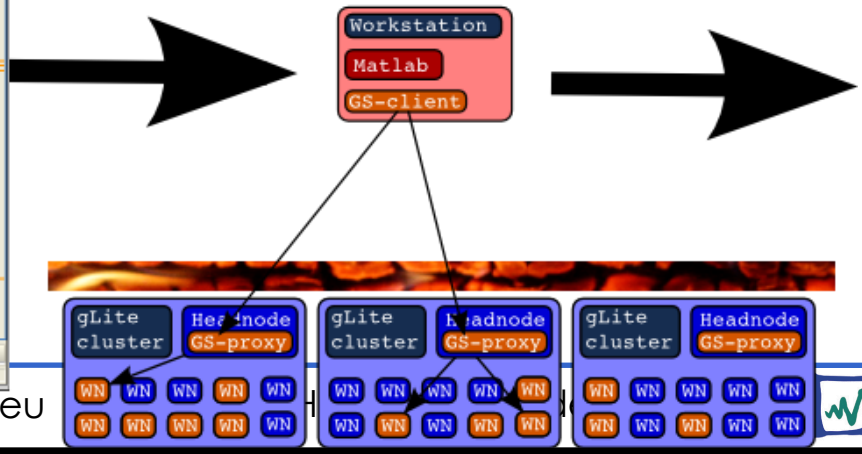


Demonstration

- Simulation: Mandelbrot fractal
- Using the same infrastructure





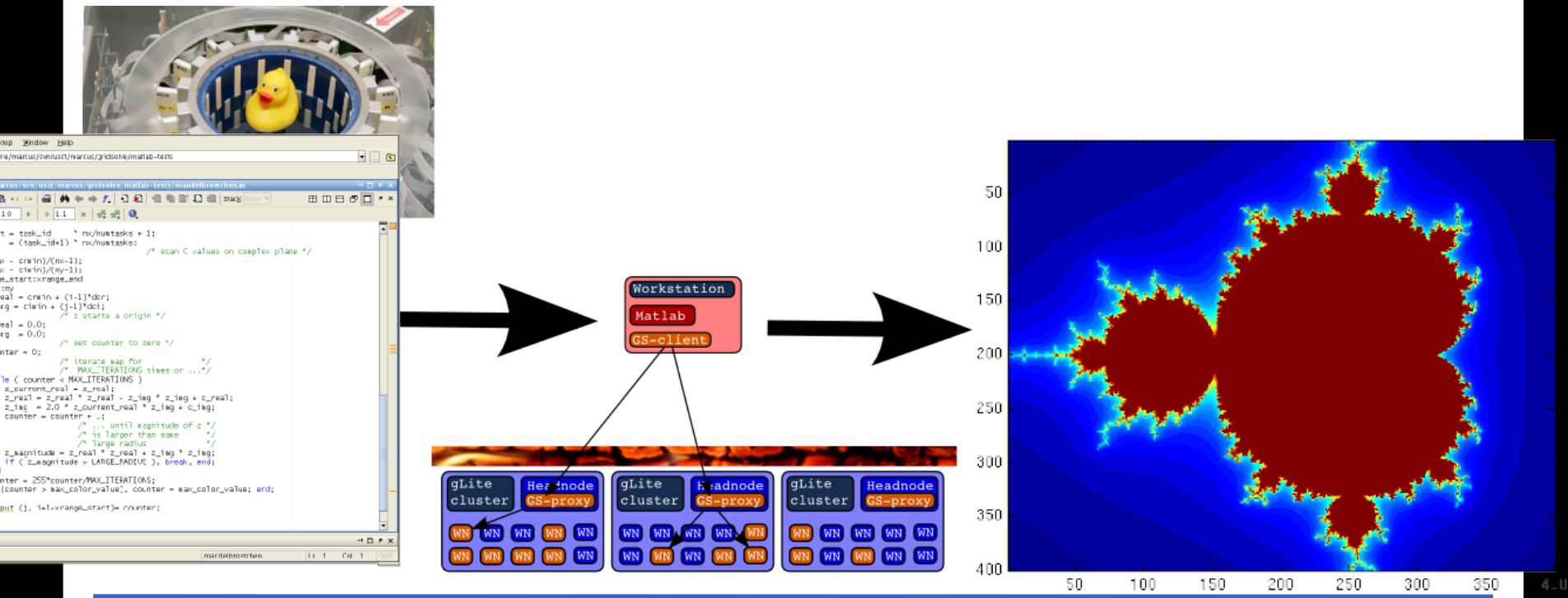


Life-Demo

- 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
```

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


Google
interactive grid
I'm Feeling Lucky