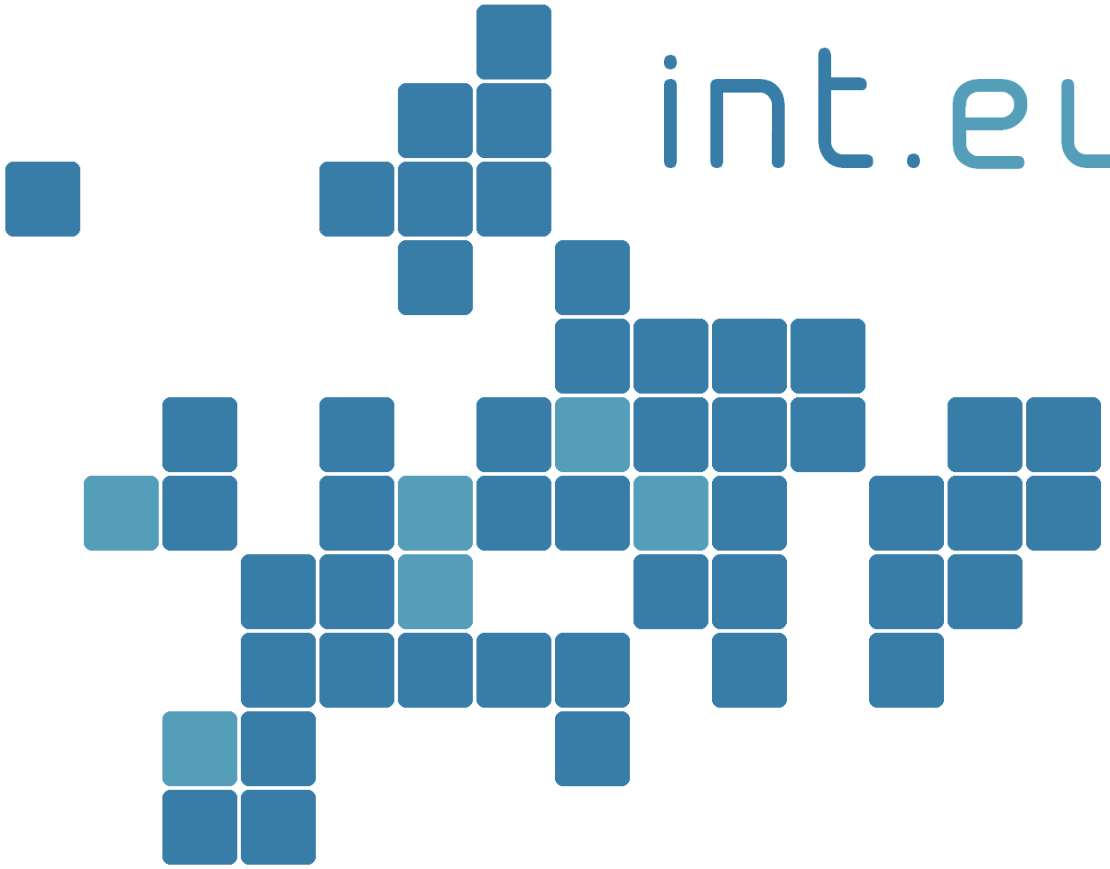
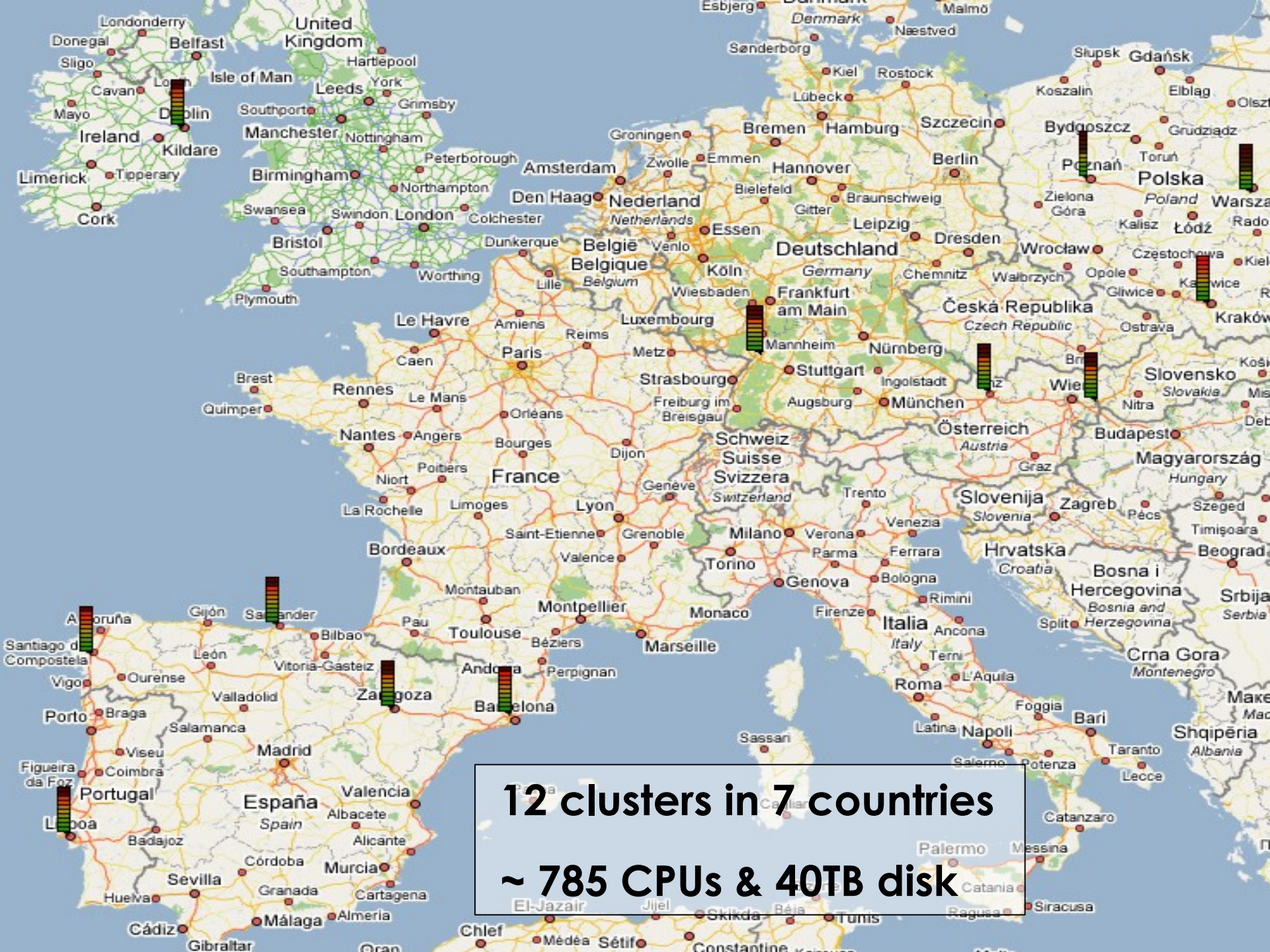


int.eu.grid

http://www.interactive-grid.eu

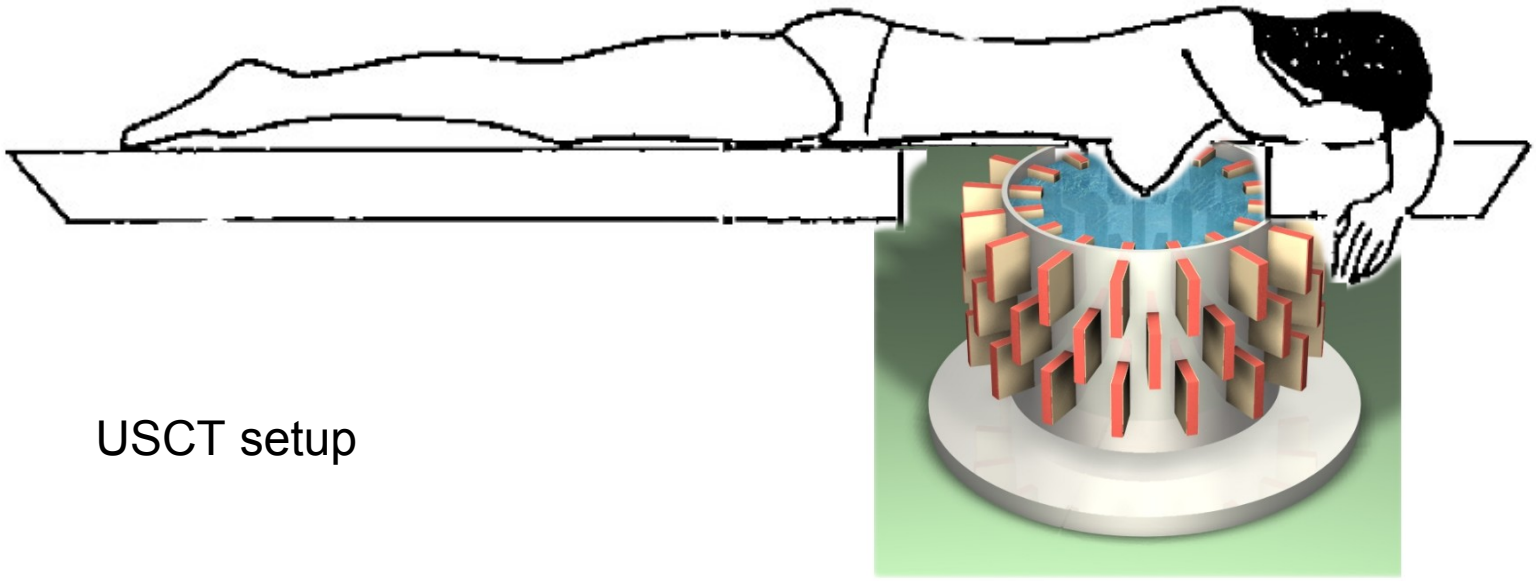
Interactive grid-access for Ultrasound-CT

Marcus Hardt
Forschungszentrum Karlsruhe


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



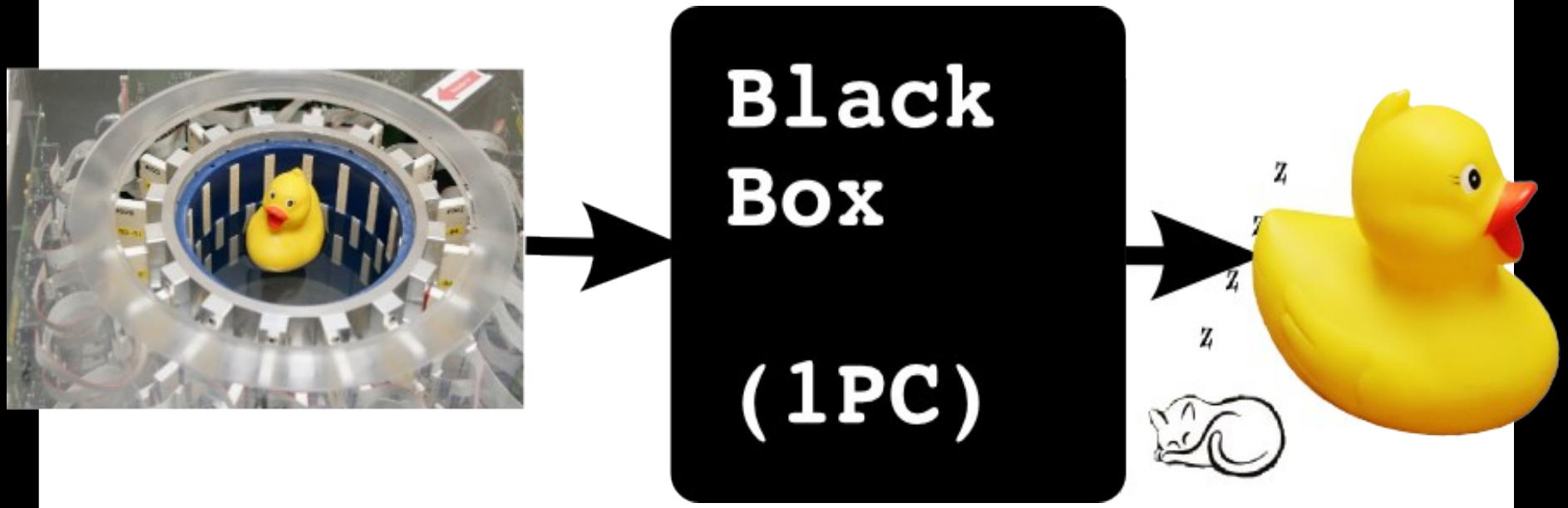
USCT setup

USCT Reconstruction := “Black Box”

- Algorithm:
  - Based on ellipsoidal backprojection (SAFT)
  - Converts ultrasound signals to 3D volume graphics
    - Input: ~ 20GB
    - Output: ~ 8GB
  - Computing time:
    - 4096^2(2D) ... 128^2x100 ... 4096^2 x 3410

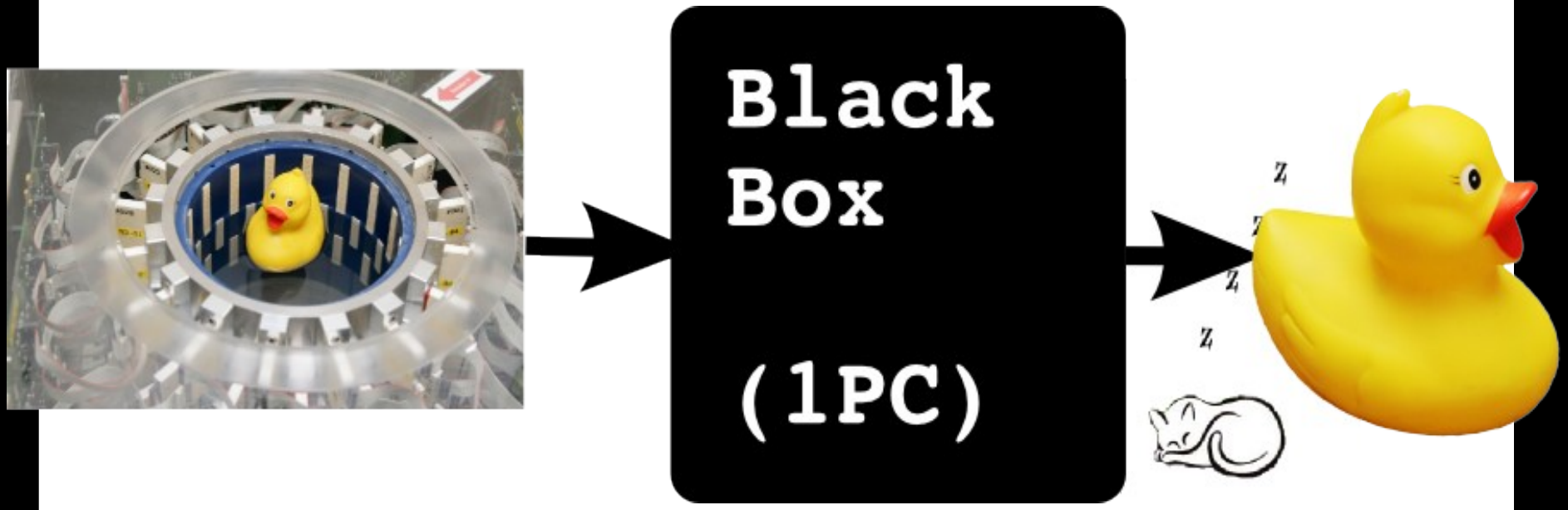
    <=> 1hour ... 1.5 Months ... 150 Years
- Matlab
  - Problem solving environment
    - similar to Maple, Mathematica, Scilab ...
  - Strategic development platform
  - Not easy to “submit matlab to the grid”

USCT Reconstruction



- Computation takes long (days, weeks, years)

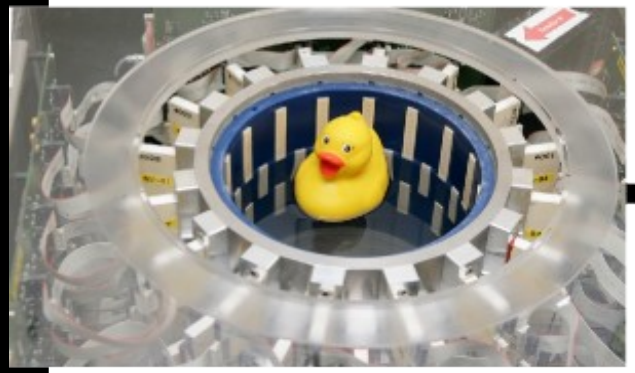
USCT Reconstruction

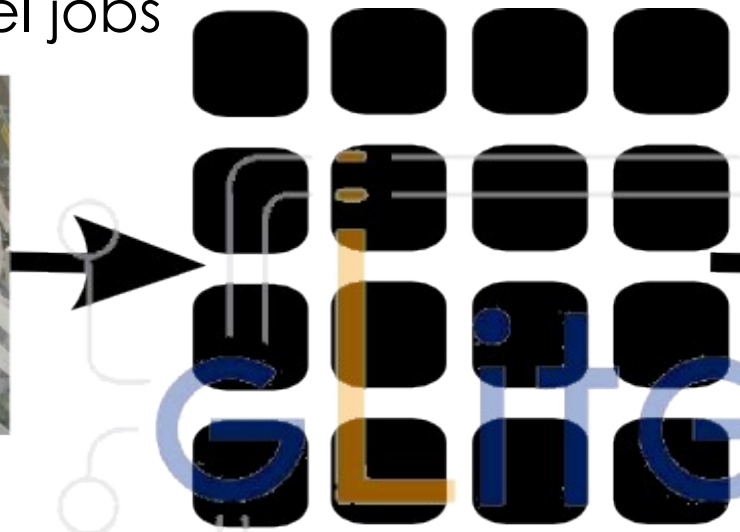


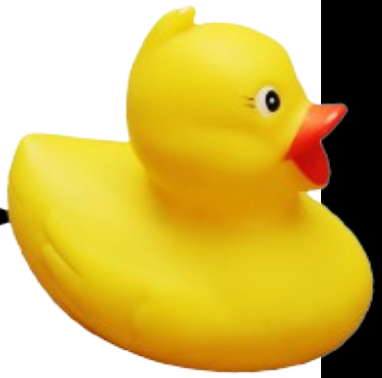
- Computation takes long (days, weeks, years)
- **Goal:**
  - **Seamless, interactive, grid access**
  - **from Matlab**

Using the grid

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

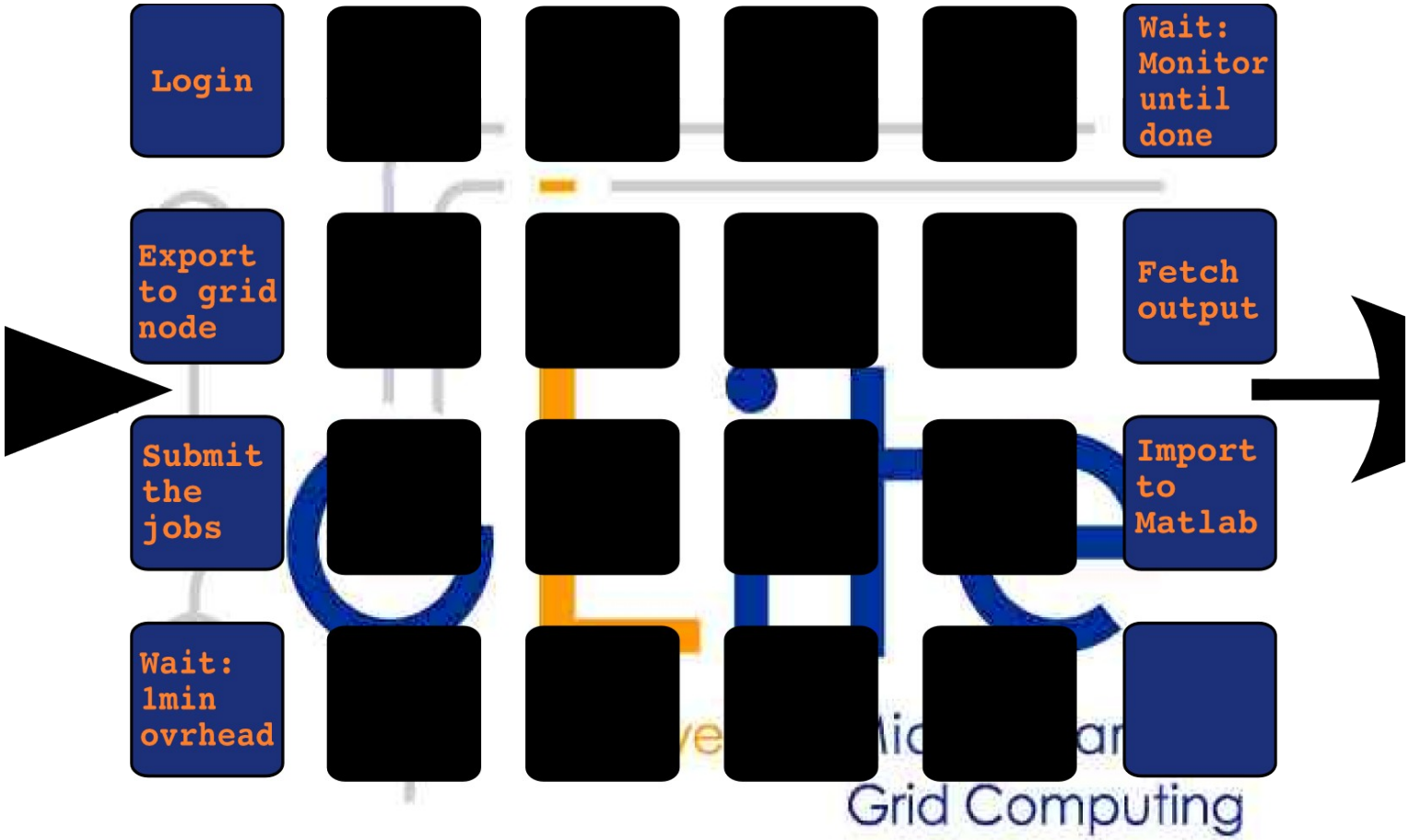






Using the grid

- Lets take a close look



Using the grid

- Goal:
  - Seamless ✘
  - interactive ✘
  - grid access ✔
  - from Matlab ✘

Using the grid

- **Goal:**
  - **Seamless** ✘
  - **interactive** ✘
  - **grid access** ✔
  - **from Matlab** ✘

Usability-test:

=> The users will run away

What's missing?

- **Goal:**
  - **Seamless** ✘
  - **interactive** ✘
  - **grid access** ✔
  - **from Matlab** ✘

- Seamless
  - User might not know if he uses the grid
- Interactive
  - No overhead (< 10 s)
  - No manual data movement
- From Matlab
  - Run Matlab-functions remotely

Improving Grid Access with RPC

- GridSolve
  - Developed at ICL, University Tennessee, Knoxville
  - Implements an RPC client/server solution
  - 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

=> Reduce complexity of the grid to one function call

How to do it?

- **Goal:**
  - **Seamless** ✘
  - **Interactive** ✘
  - **Grid access** ✔
  - **For scientists** ✘

- 1. Integrate GridSolve with gLite
  - **G**ridSolve **i**nte**g**ration with **gLi**t**e**
  - **GIGGLE**
- 2. Make Matlab run on gLite
  - **G**rid **i**n **M**atlab using **G**ridsolv**e** & **R**PC
  - **GIMGER** (speak: ginger)

GridSolve(GS)/gLite Integration

- Create GS-Service hosts (GS-agent + GS-proxy)
- Send 100s of GS-servers to gLite infrastructure
  - Build infrastructure
  - Package GridSolve + Matlab Runtim
  - Install GridSolve + Matlab on WNs
- Ensure network connectivity
  - GS-Server <=> GS-Proxy <=> GS-agent <=> GS-client

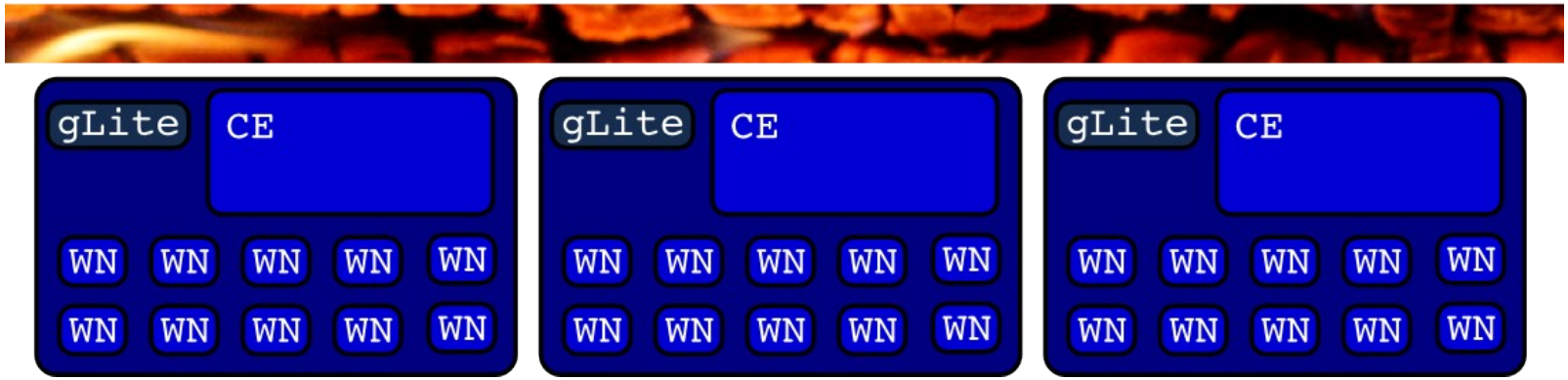
Matlab/gLite integration

- Matlab Compiler (toolbox)

=> Matlab Compiler Runtime (MCR)

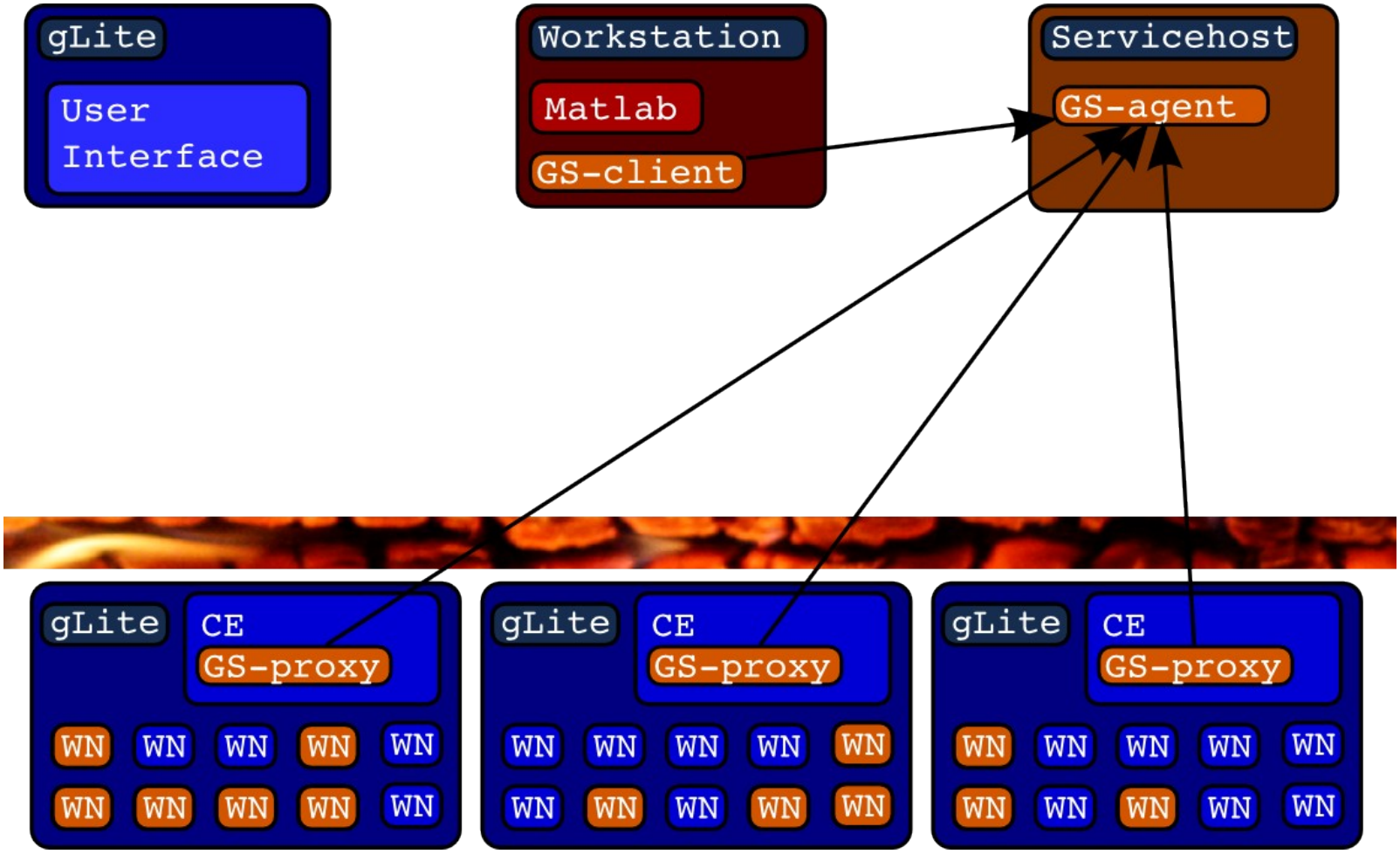
  - Install on the fly (as part of glite-job)
  - Fix linux glibc version incompatibility
    - Install new glibc on the fly
- Usability enhancement
  - Access GridSolve from Matlab
  - Point Matlab to service hosts
  - Support for RPC creation
    - Compilation/Linking/Deployment

GridSolve startup on gLite

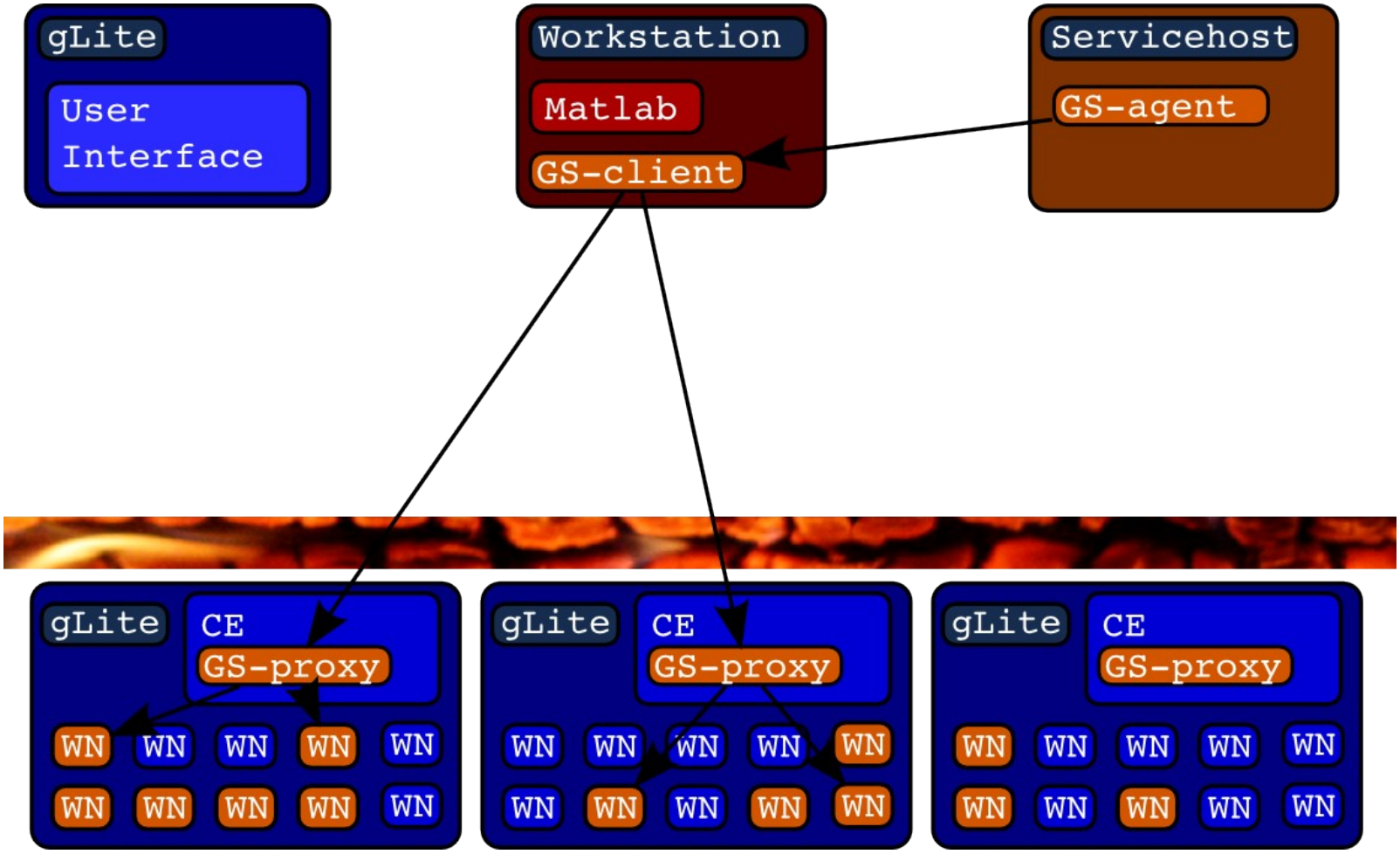


GridSolve ready for action

GridSolve ready in action



GridSolve ready in action

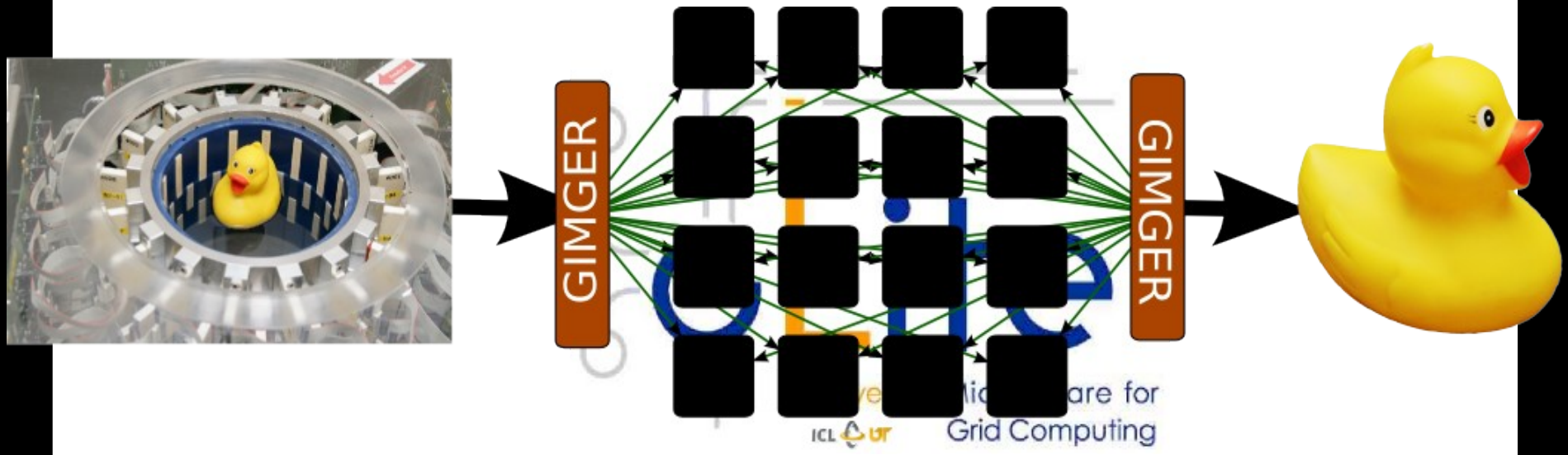


Putting things together

- RPC with GridSolve
- On top of int.eu.grid/gLite
- Using Matlab functionality
- GIMGER

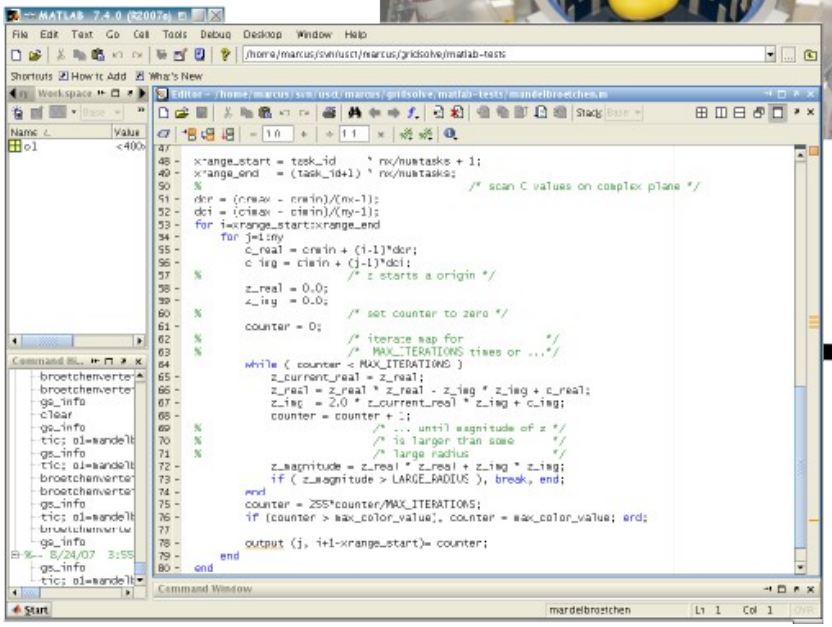
Putting things together

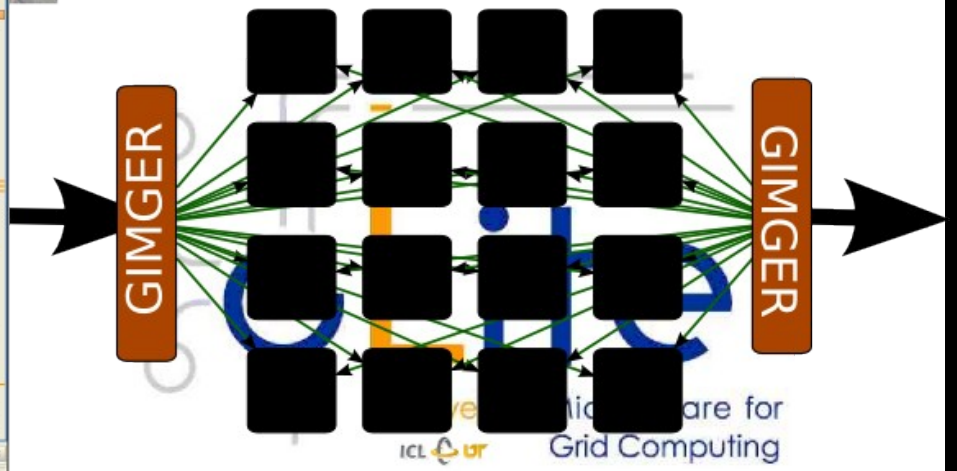
- RPC with GridSolve
- On top of int.eu.grid/gLite
- Using Matlab functionality
- GIMGER



Demonstration

- Simulation: Mandelbrot fractal
- Using the same infrastructure



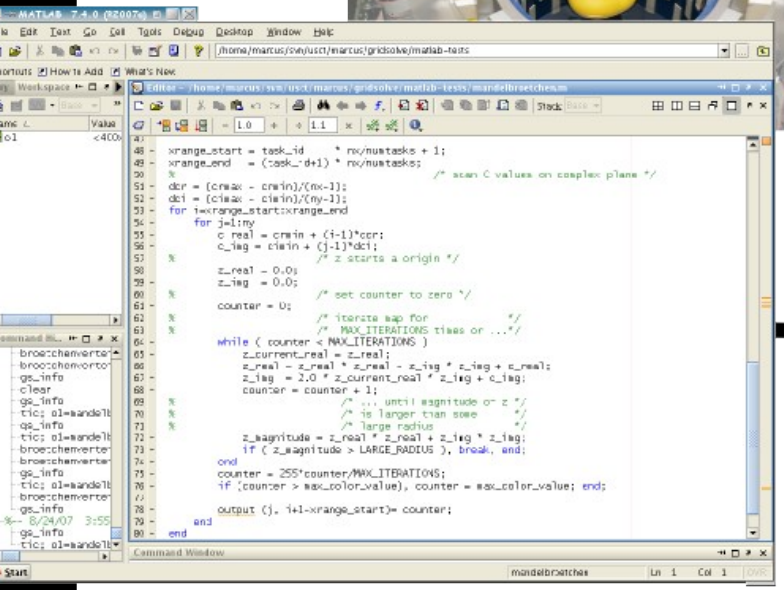


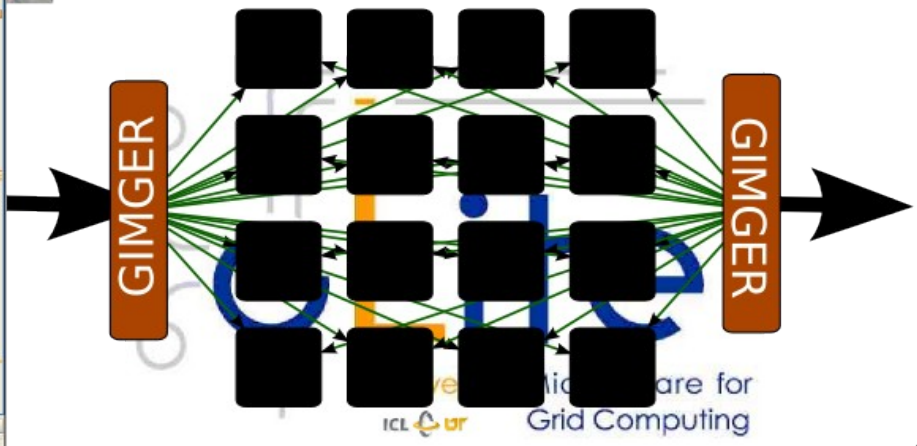
Life-Demo

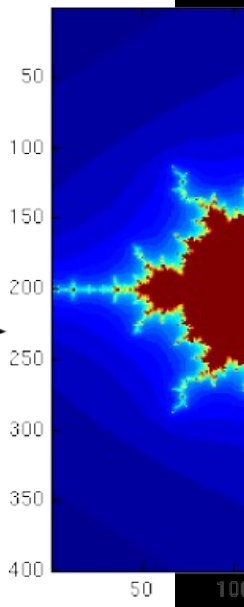
- Movie of the life demonstration:
  - **`http://marcus.hardt-it.de/grid4matlab`**
- **Real** life demo on int.eu.grid
  - Talk to me (any time during the breaks)

Result

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







Current status

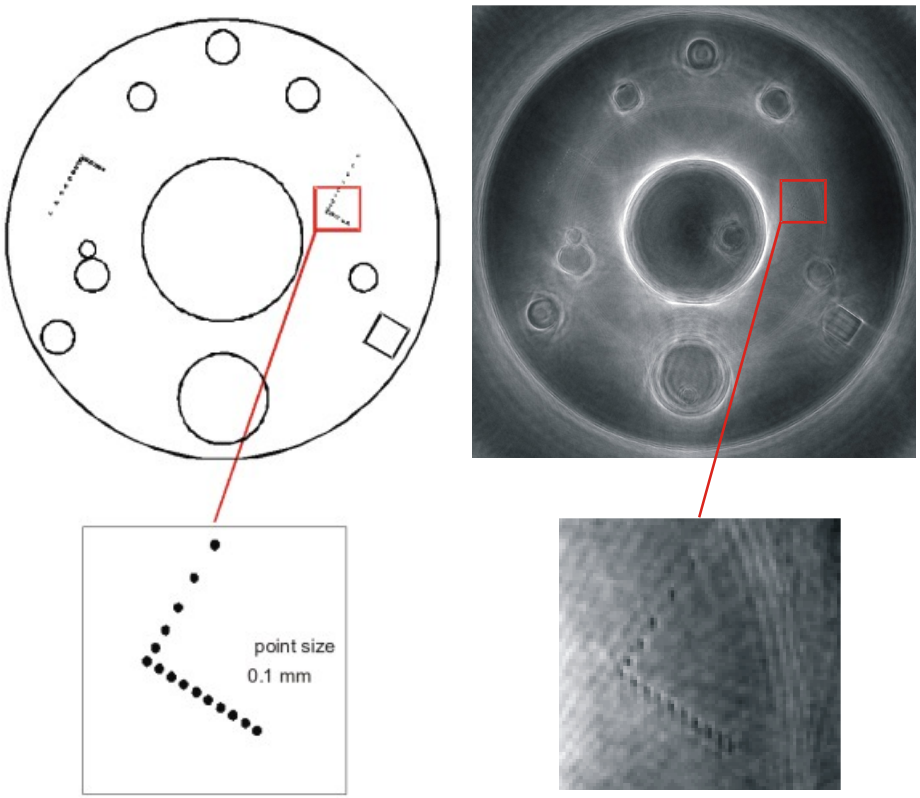
- We can
  - Convert Matlab functions to run on the grid
    - Involved hands-on work
    - Automatic: current work
  - Run simple simulations in our infrastructure
- We want to...
  - Use real code
  - Automatically send Matlab functions to the grid
    - Reduce hands-on work
  - Data Handling (GFAL)
- Interactive Grid
  - Submit from MD (better: from Matlab via RAS)
  - Interactive channel for debugging
  - MPI for collecting output data


Google
interactive grid
I'm Feeling Lucky

USCT Images

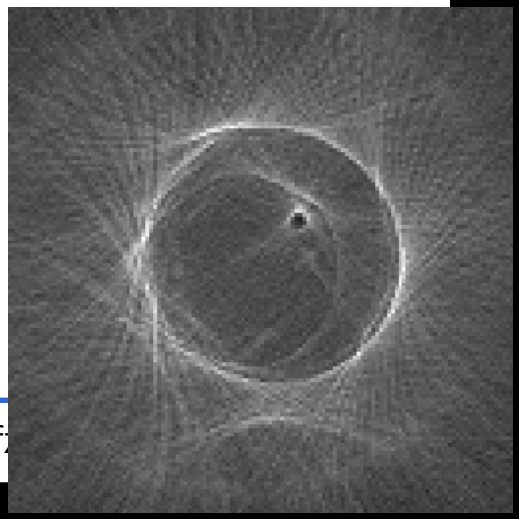
First results with old USCT:

- 0.1 mm Nylon threads visible



Current results with new hardware:

- EGG & Yolk visible
- 3D imaging



Improving grid access

- Idea: Remote Procedure Calls (RPC)
  - Submit daemon(s) as glite job(s)
  - Integrate client into Matlab
  - Connect to daemon(s) from client
    - Call remote procedures from client
    - Transfer input/output parameters
- Advantages:
  - "glite-submit-penalty" only for startup
  - Interactive answer via direct network connection
- Disadvantages:
  - Implement an RPC solution....