int.eu.grid

http://www.interactive-grid.eu

Interactive grid access for Matlab

Marcus Hardt
SCC @ FZK

Grid Computing

- Idea: **Computer power <=> Electrical power**

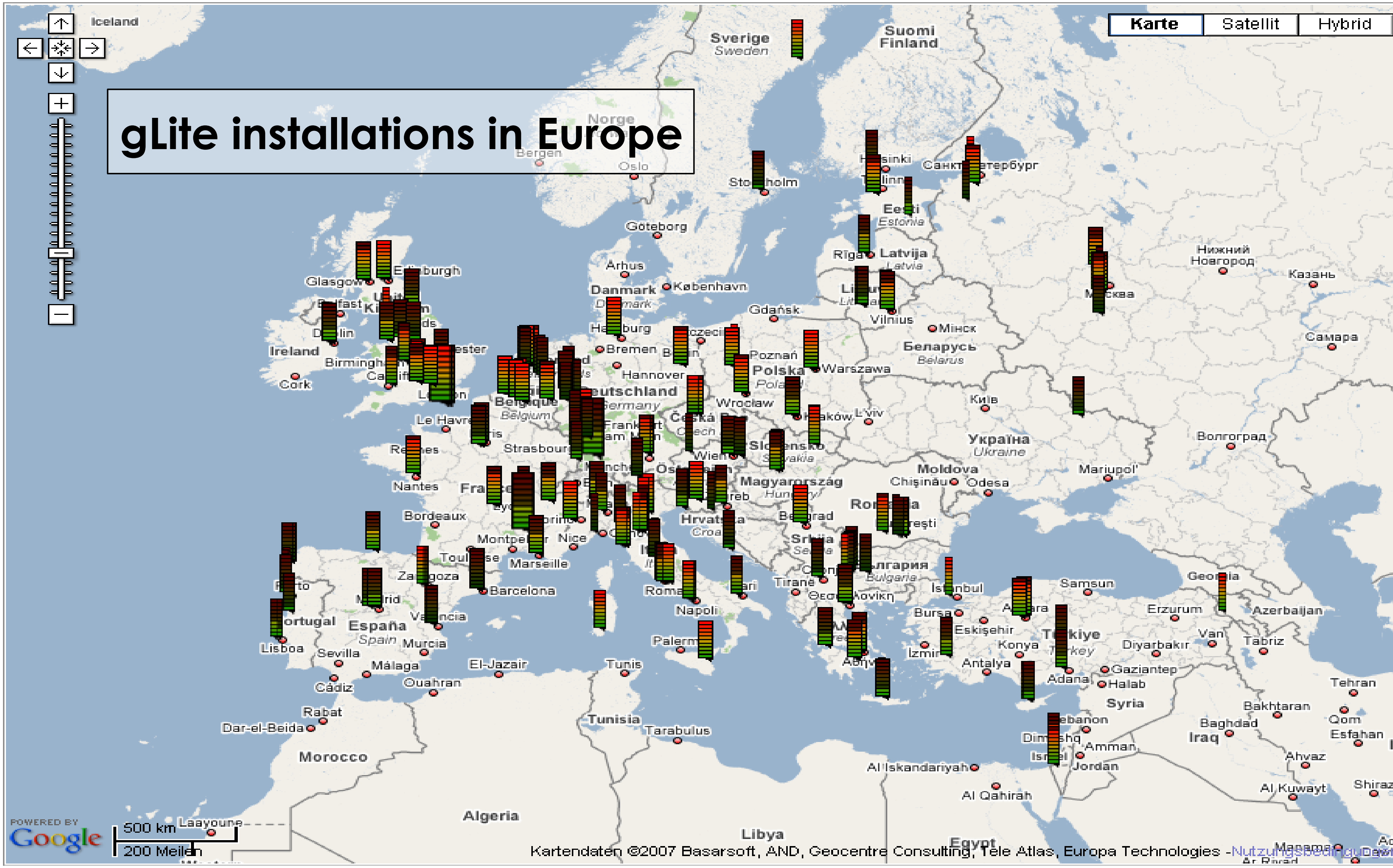
  From Electrical power grid => computational grid

  - Across organisational domains / countries
  - Transparent access to
    - Computing
    - Data
    - Network
  - Large scale installations

Grid middleware

- Middleware
  := Softwarelayer between
  application and operating system
- **gLite:** one grid middleware
  - Development driven by CERN
  - Tools for data+computing of new accelerator
  - 10 PB/year * 20 years, random access
  - **Job** based:
    - Job = Complete application + description
    - Send **job** to remote compute center
    - Get result back after job is finished

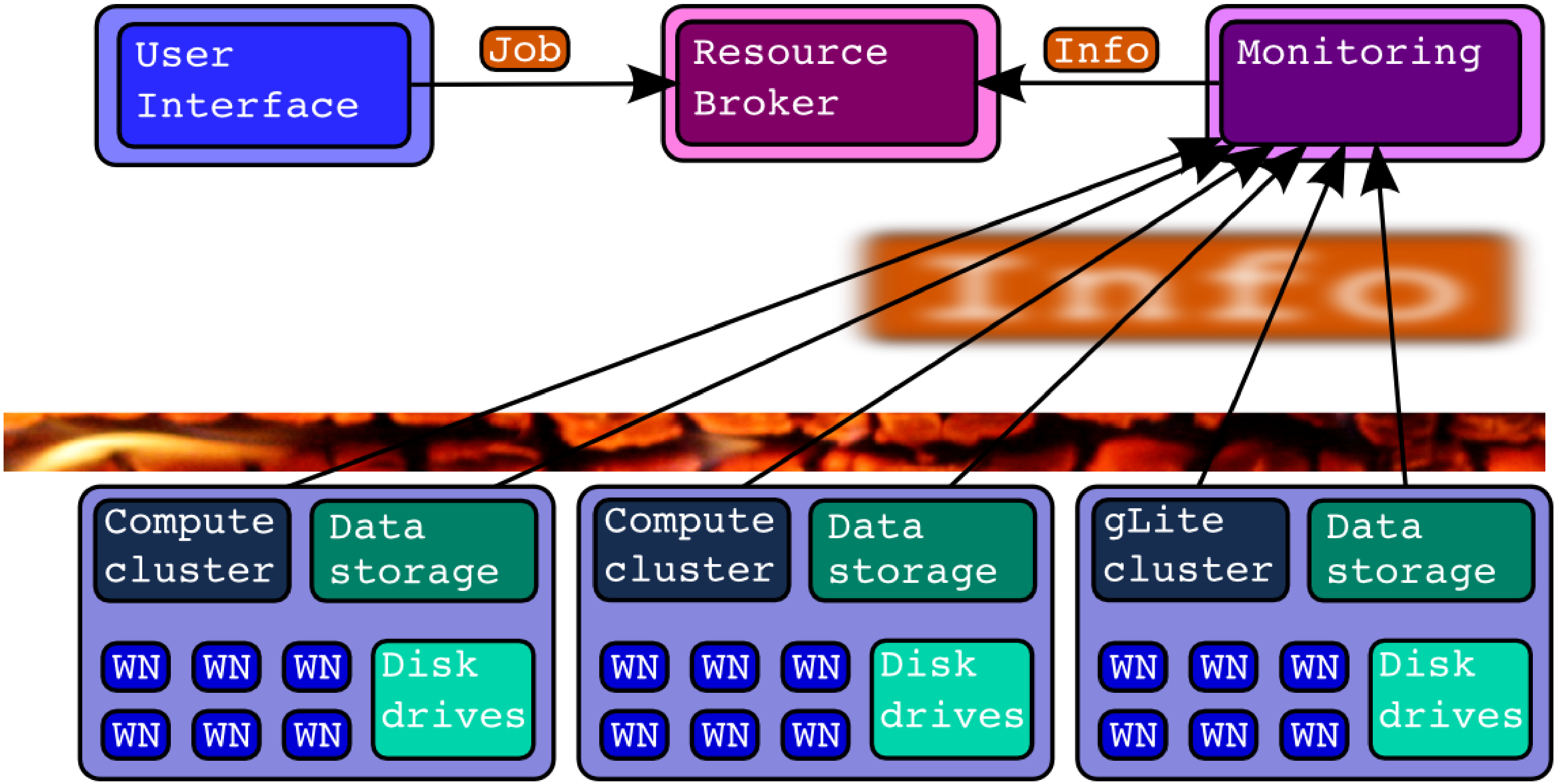
Location: http://zwopi.de/~marcus/mashups/googlemon.php?view=i2g_lcg



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

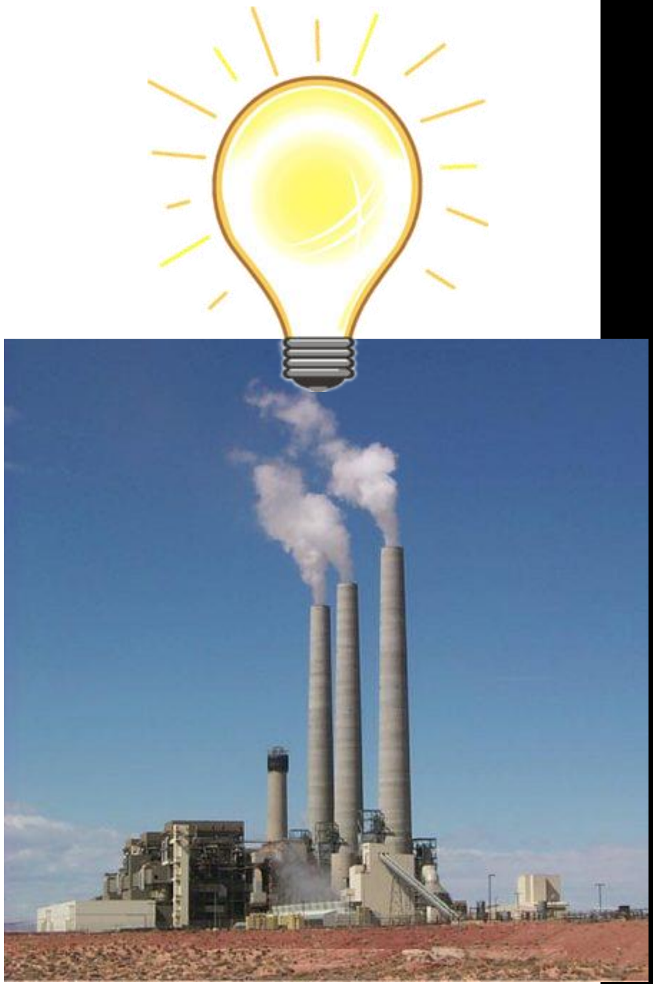
0-9 CPUs --- 10-99 CPUs --- 100-999 CPUs --- > 1000 CPUs

gLite architecture



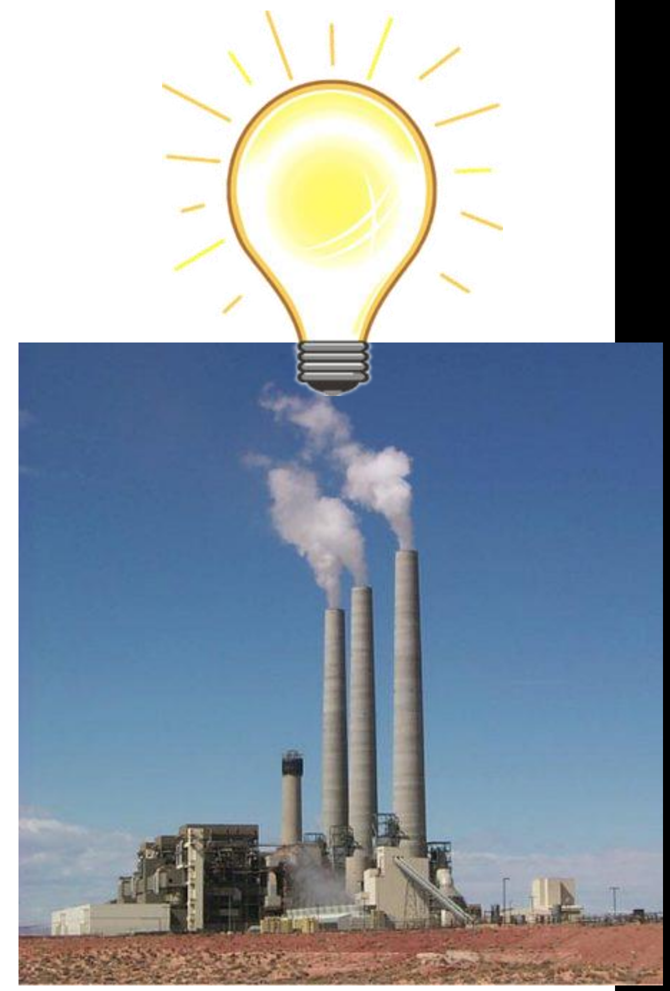
Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...



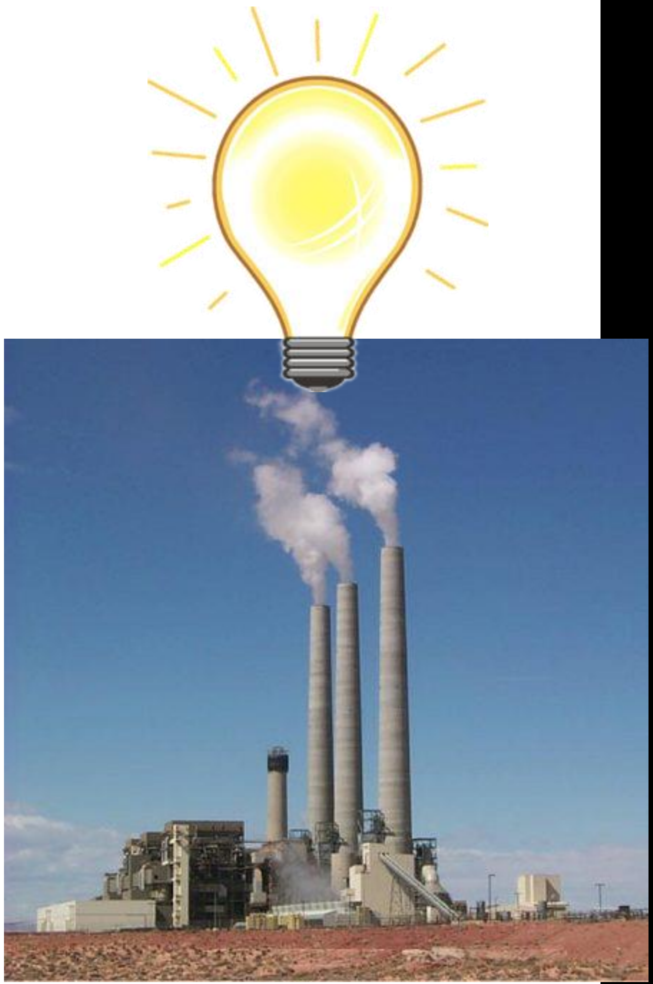
Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker



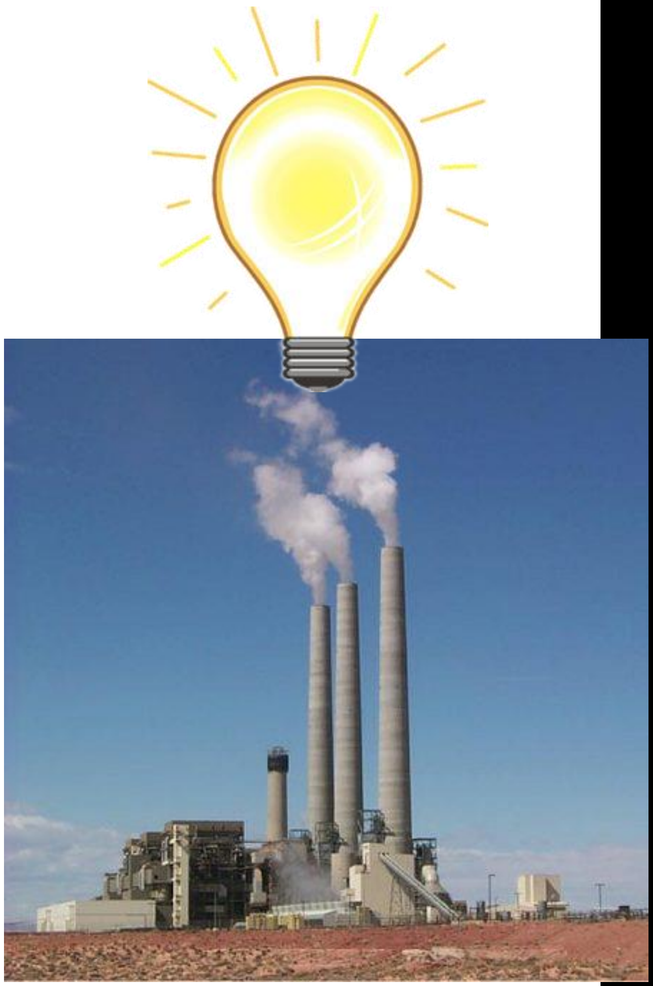
Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you



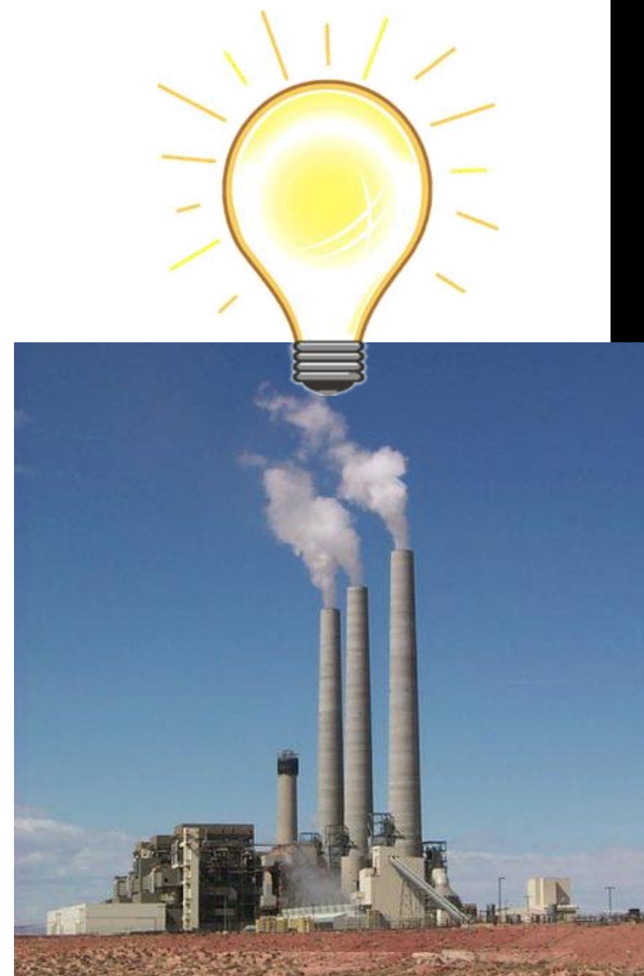
Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you
  - => Send lightbulb to powerplant



Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you
  - => Send lightbulb to powerplant
  - => Wait for electricity

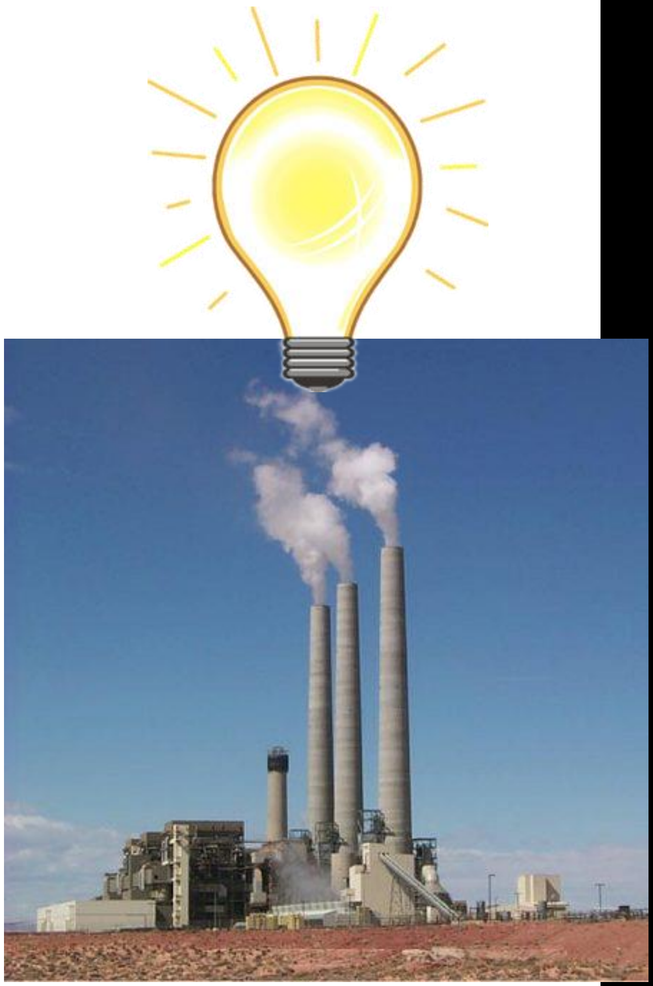


Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you
  - => Send lightbulb to powerplant
  - => Wait for electricity
  - **=> Lightbulb glows**

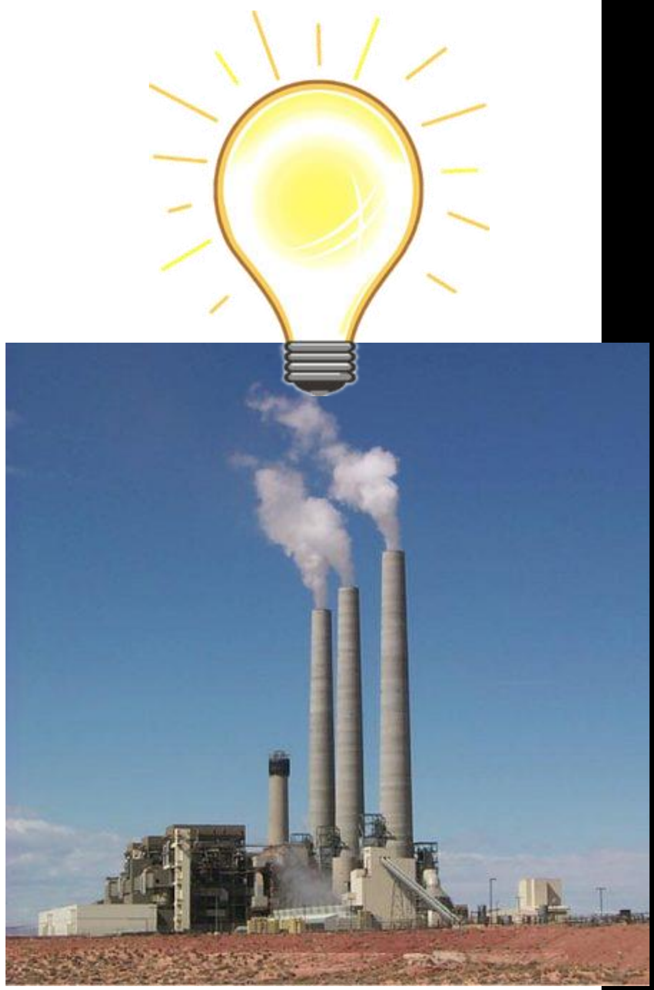
Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you
  - => Send lightbulb to powerplant
  - => Wait for electricity
  - **=> Lightbulb glows**
- Results come back



Using a lightbulb in the job based grid world

- Describe the lightbulb
  - Voltage, Watts, Amount
  - Lighting_time, ...
- Submit request for electricity to broker
  - => Powerplant chosen for you
  - => Send lightbulb to powerplant
  - => Wait for electricity
  - **=> Lightbulb glows**
- Results come back
  - About 20% of the bulbs broken



Is interactivity a solution?

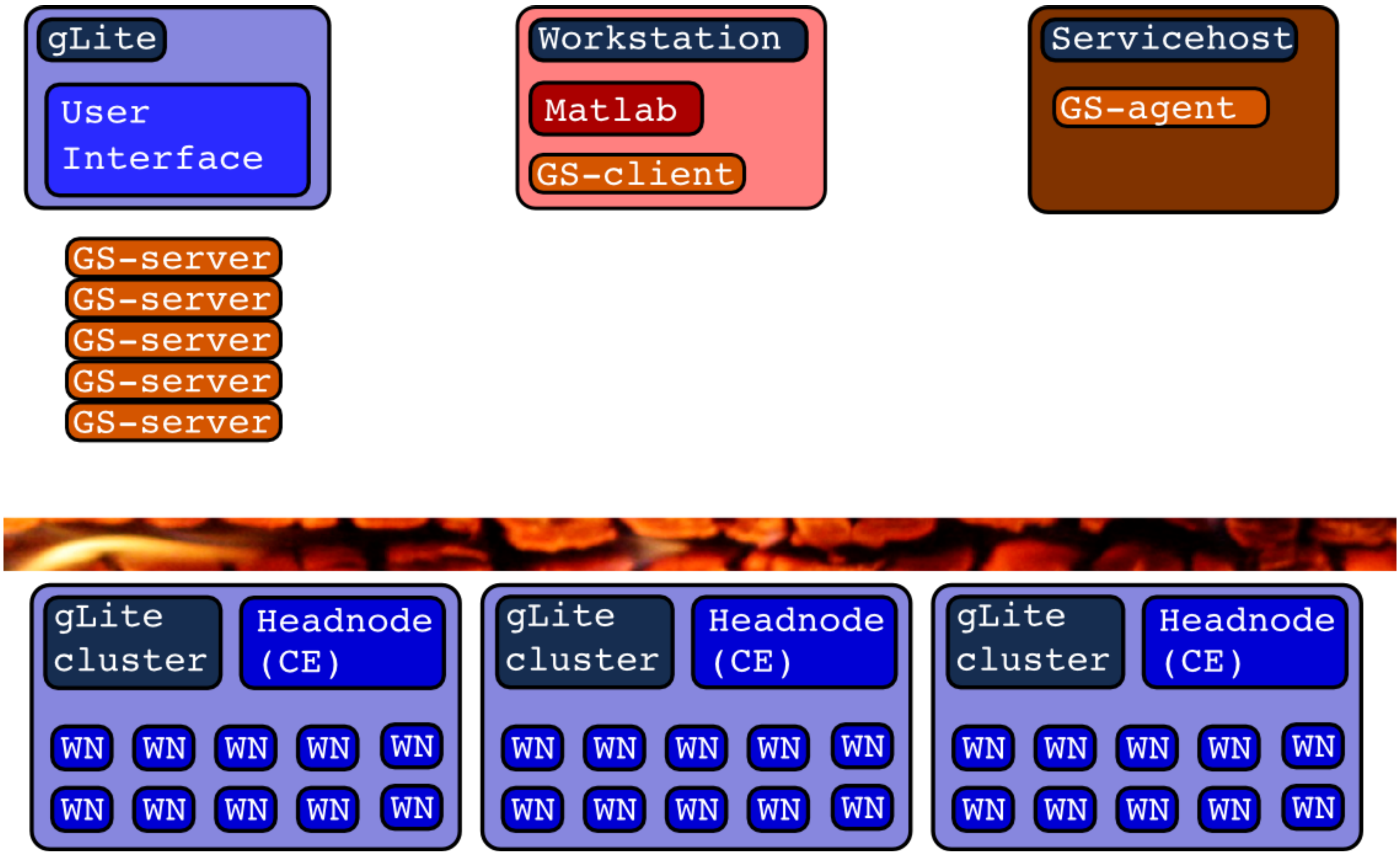
Is interactivity a solution?

Yes!
We submit a cable-job !

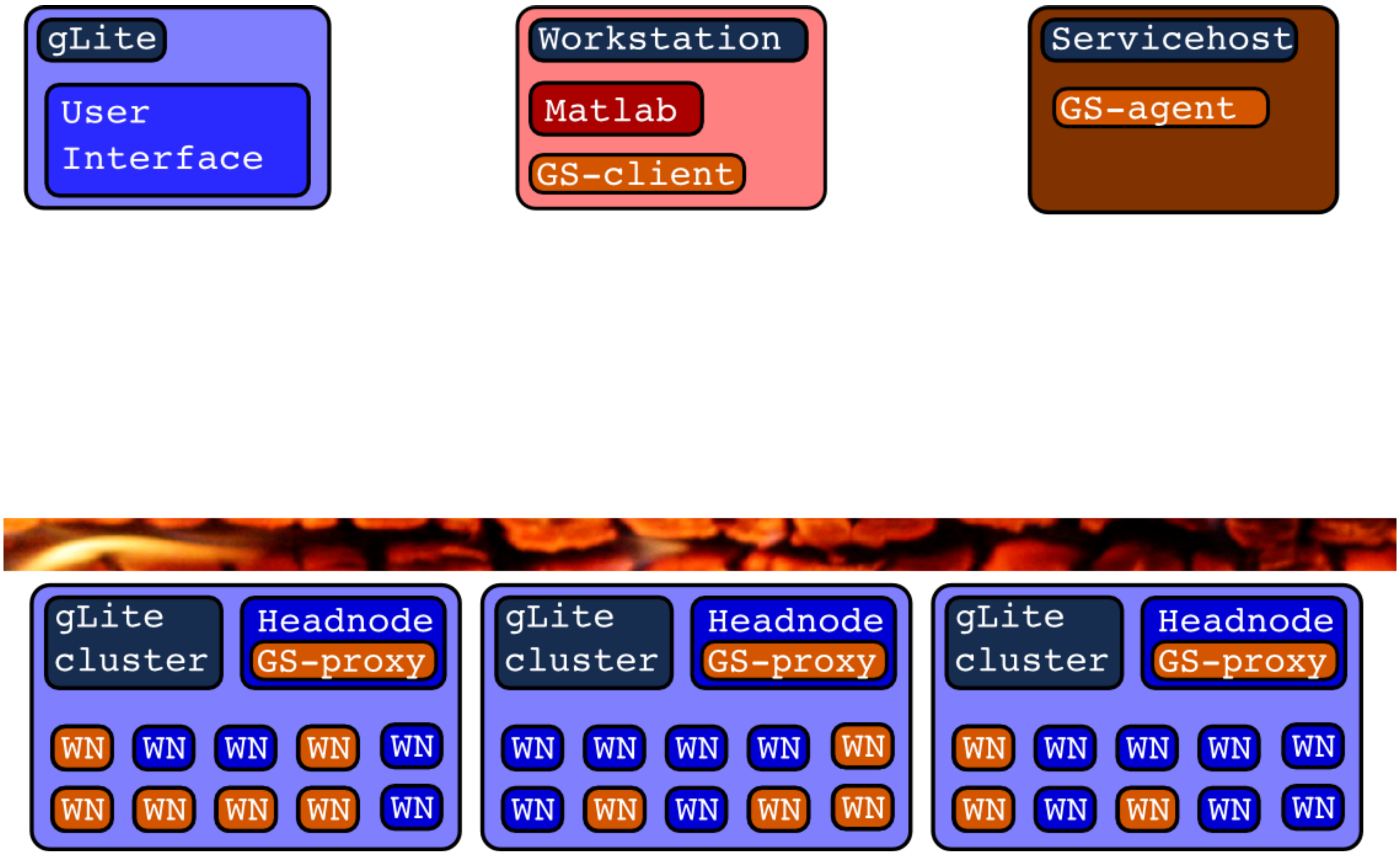
The interactive channel

A “cable” connects user with resource

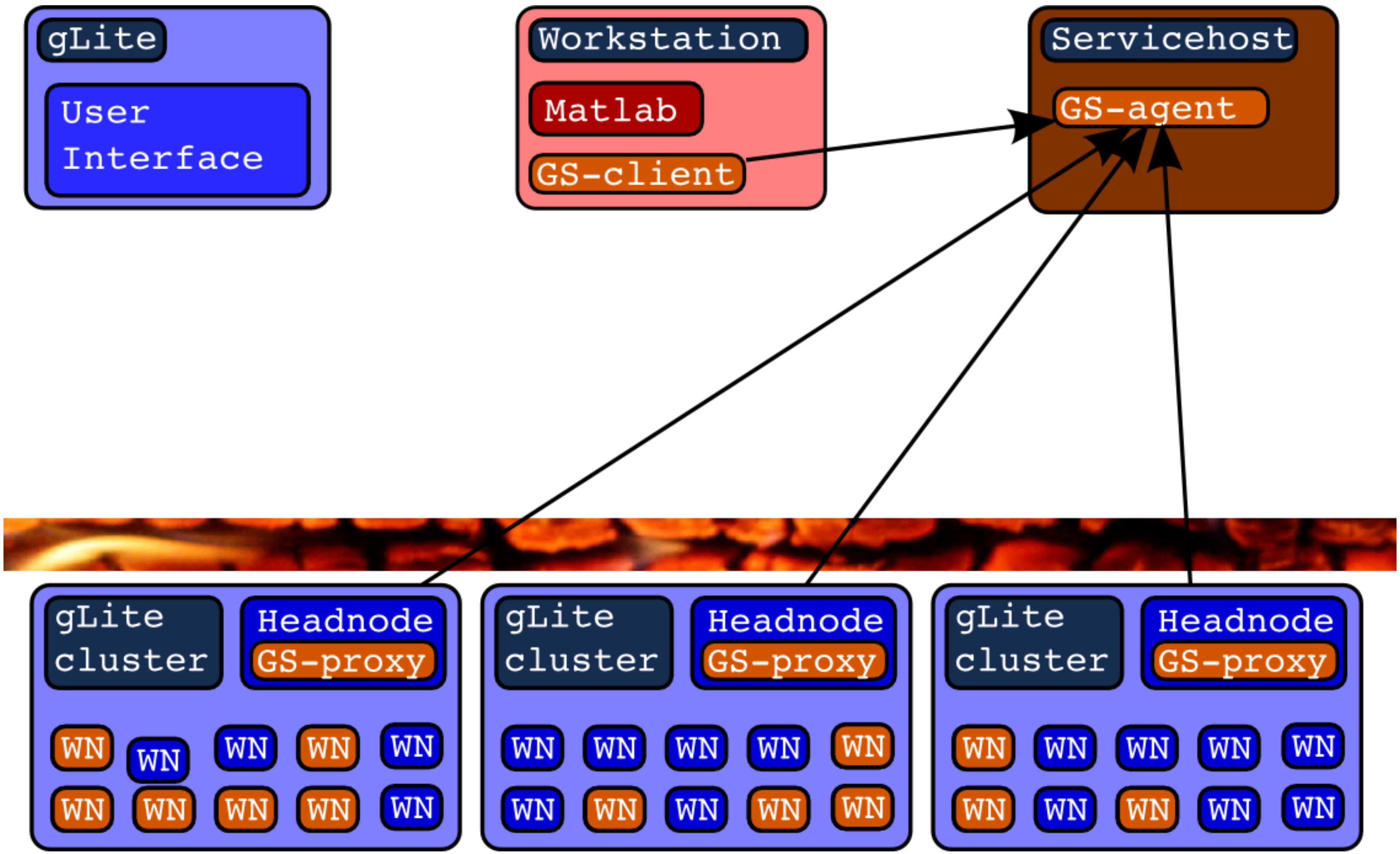
Our cable: GridSolve



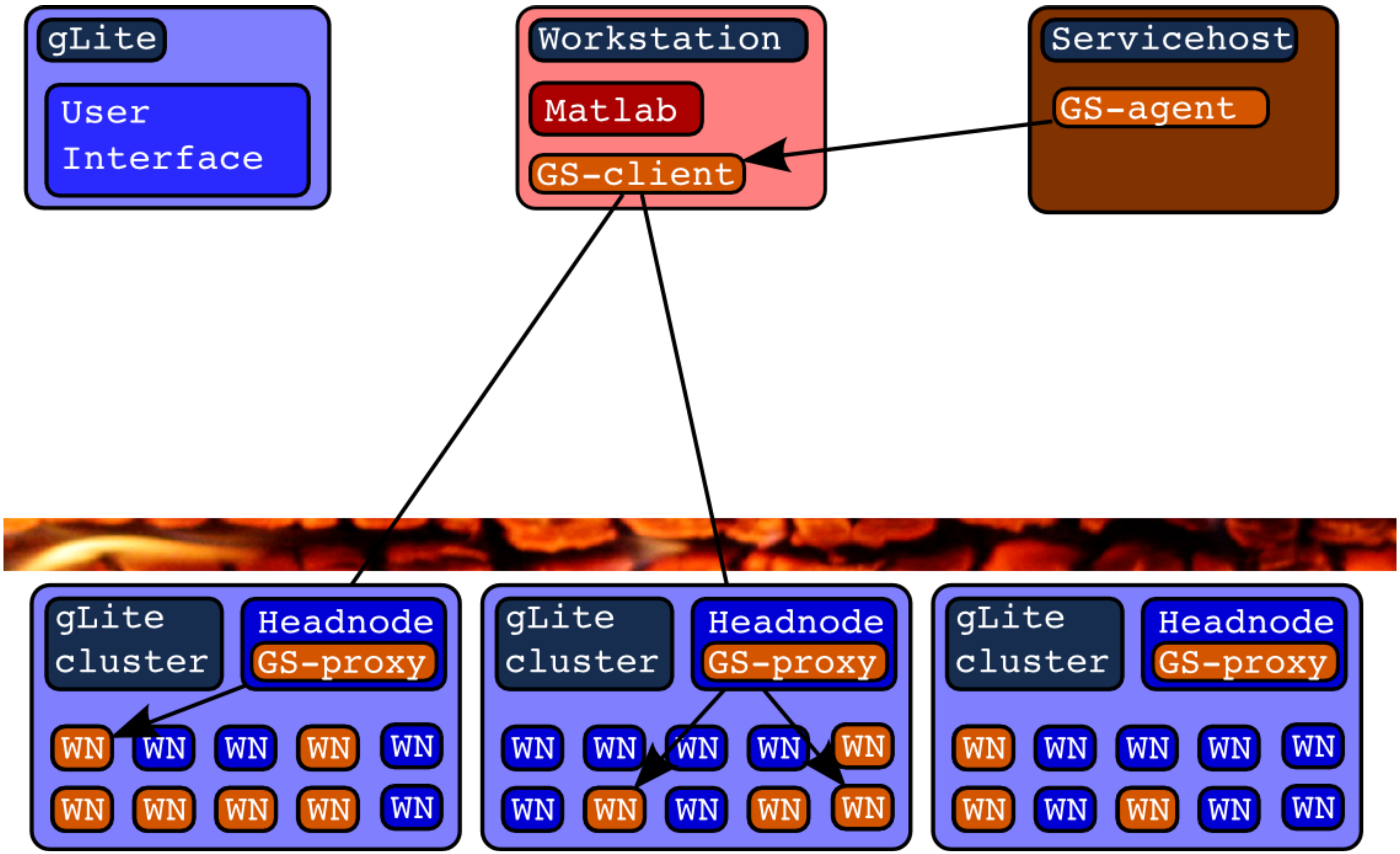
GridSolve submitted to the WNs



GridSolve network connectivity



GridSolve workflow



GridSolve interface

- 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

Life-Demo

- Two concurrent runs of "**backpropagation**" algorithm
- **Local part:**
  - Matlab computes "backpropagation" on my laptop
- **Grid part:**
  - Matlab computes "backpropagation" using the grid
  - We use
    - **GridSolve** interfaces (used in backpropagation_**parallel**)
    - Migrating Desktop (**MD**): grid Integration environment
    - Grid resource allocation (via jobs from MD)

Life-Demo

- We can...
  ... use grid resources from Matlab
  ... compute **more** pixels
  ... in **shorter** time
  ... **develop** algorithms **faster**

- Download a movie of the life demonstration:
  - **http://marcus.hardt-it.de/grid4matlab**

Summary

- Current status
  - Grid useable within Applications (like Matlab)
    - Interactively
    - Without much grid specific knowledge
- Work in progress
  - Improve minor itches with GridSolve
  - Simplify grid allocation
- Future work
  - Software deployment
  - Data management
  - Inter process communication (MPI)


Google
interactive grid
I'm Feeling Lucky

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
```

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

Example:
Large Excel Table
- **Excel must run locally**
- **Computation in the grid**

USCT Algorithm

- Characteristicts:
  - Input: 20 GB (full set)
  - Computing time depends
    - on output size / resolution
    - amount of input data

35MB	20GB	20GB	Data
4096^2	128^2x100	4096^2x3410	Voxels
1 Hour	1.5 Months	150 Years	Time

- Matlab
  - Strategic development platform (95% sourcecode)

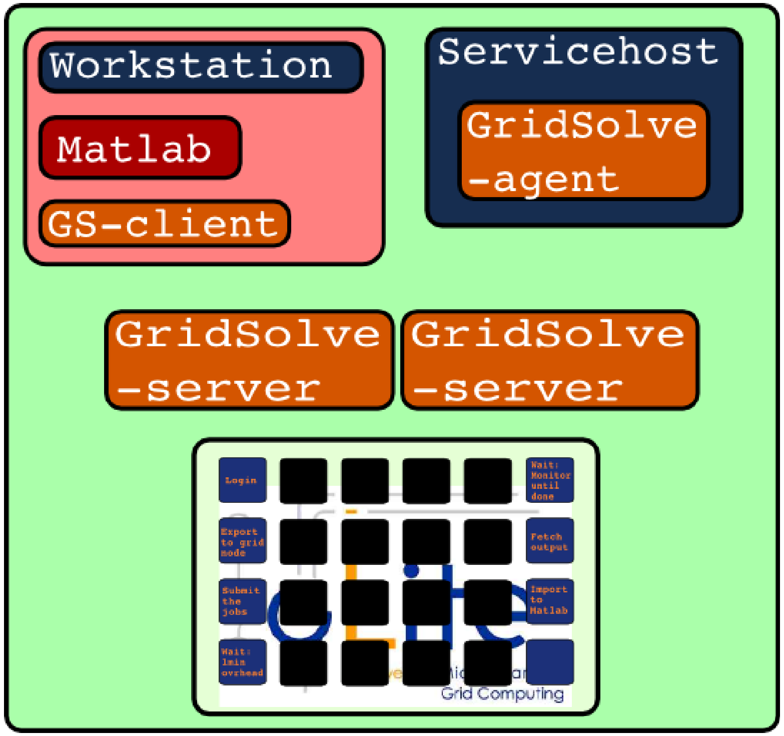
- **Goals for grid access:**
  - **Seamless**
  - **Interactive**
  - **from Matlab**

GridSolve (GS)/gLite integration

- Send GS-servers to gLite clusters
  - Package GridSolve + My software
  - Send packages into gLite jobs
  - Install packages on WorkerNodes (WN)
- Create GS-service hosts (GS-agent)
- Ensure network connectivity
  - GS-client, GS-agent, GS-proxy, GS-server
- All this happens behind the scenes!

How to do it?

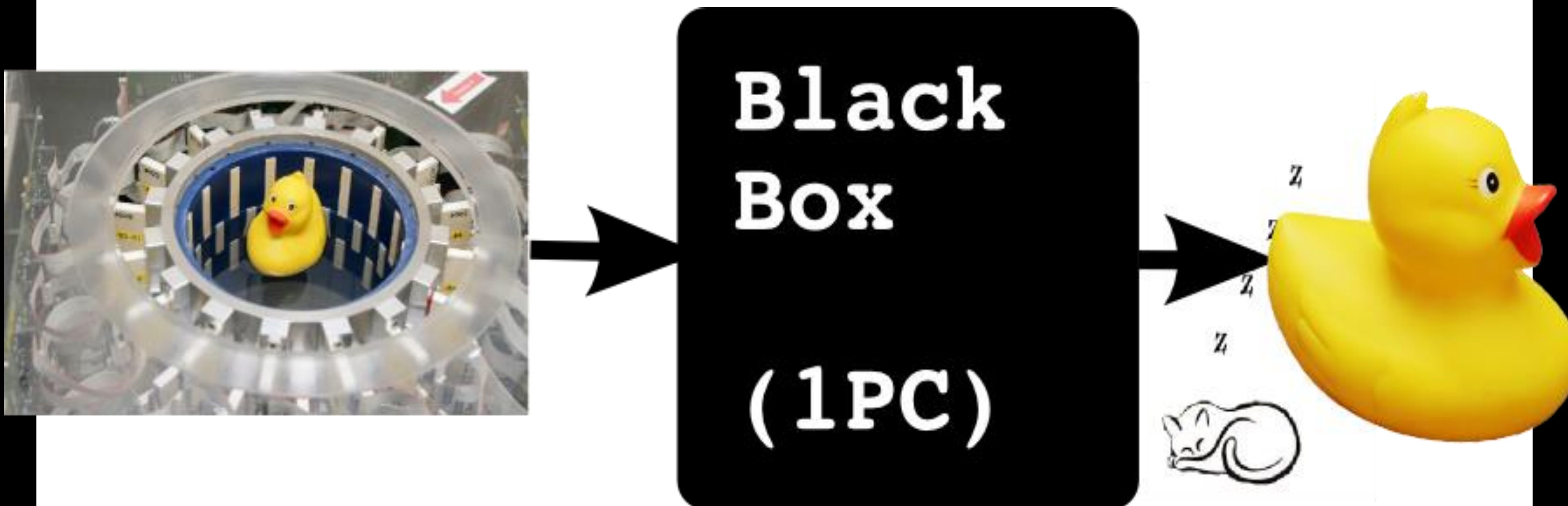
- 1. Make Matlab run on gLite
- 2. Integrate GridSolve with gLite



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

GIMGER

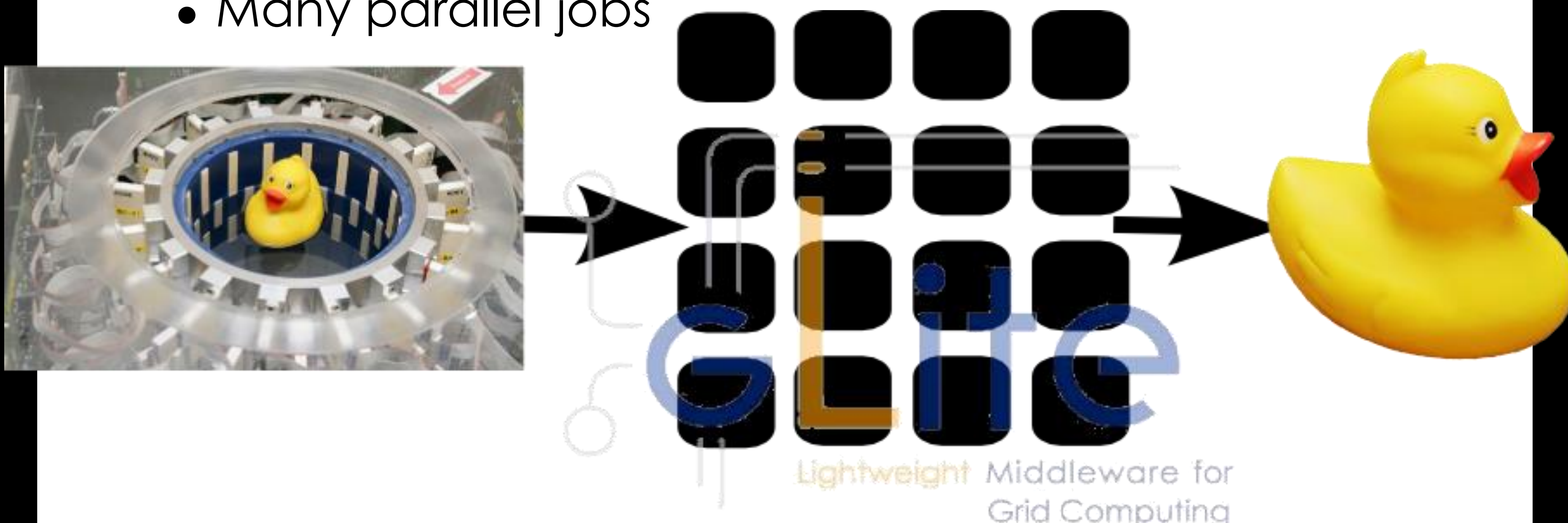
USCT reconstruction := "Black Box"



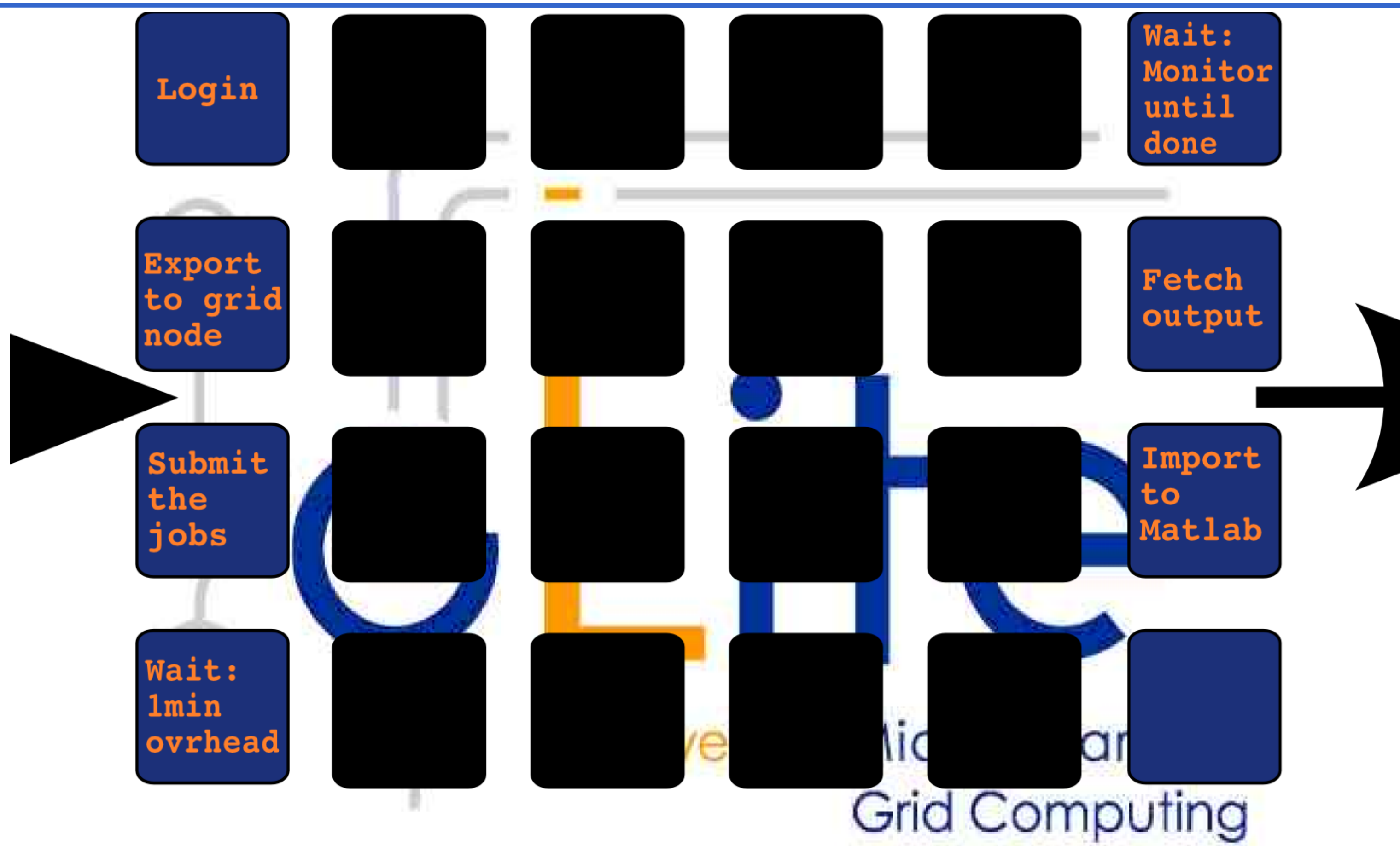
- Computation takes long (days, weeks, years)

Using gLite

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



Using gLite in practise



- A lot of work is left to the user

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

Marcus.Hardt@iwr.fzk.de