



Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften



Using virtualisation for tailor-made HPC solutions

Ferdinand JAMITZKY,

Deputy Leader Application Group at Leibniz Supercomputing Centre, Garching

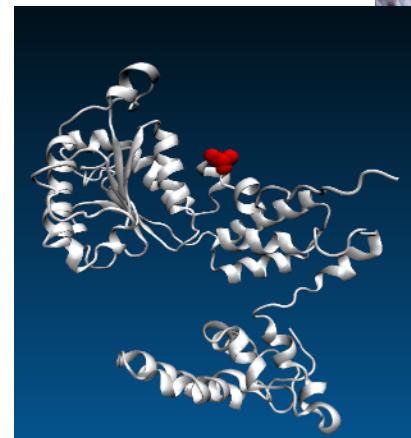
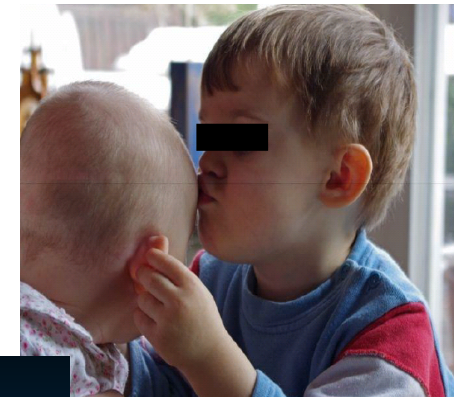
TERATEC 2016 Forum

Workshop 1 - Wednesday, June 29, 2016

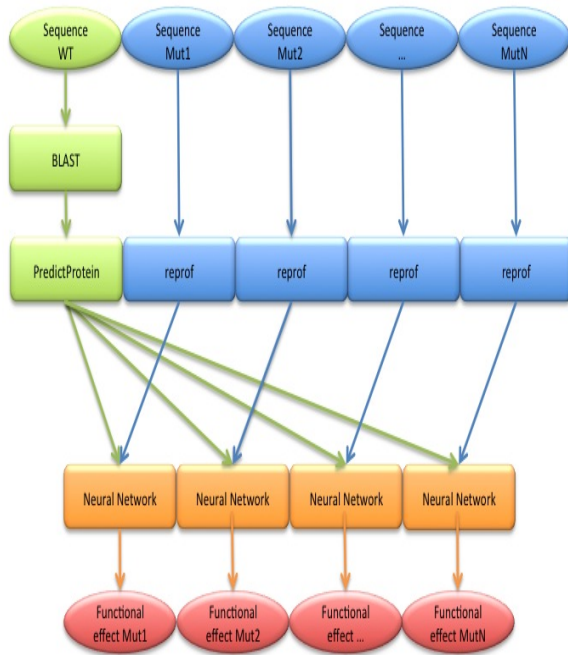
Take existing bioinformatics projects at LRZ and implement them using workflow tools

- Metagenomics in Childhood Asthma research
- Systems Biology: Predict effects of small changes in amino-sequence on protein function

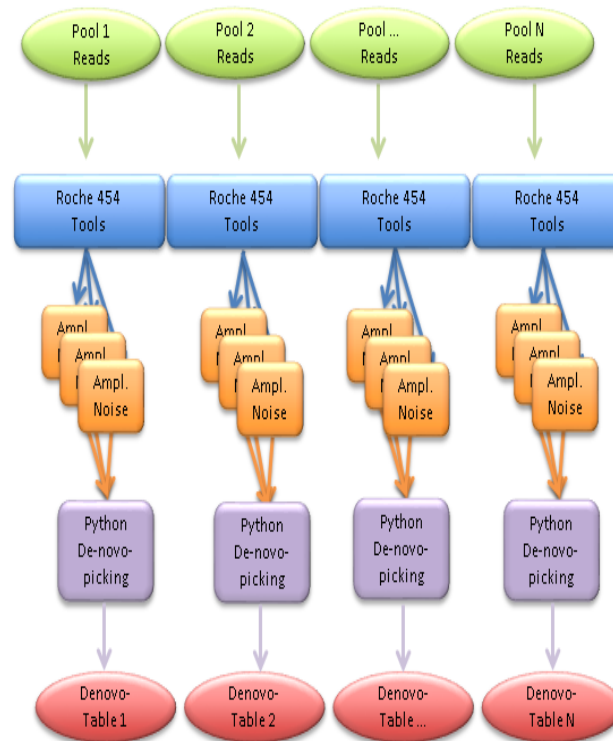
Co-Design: Infrastructure/Platform-Provider, Workflow-Developers, End-Users



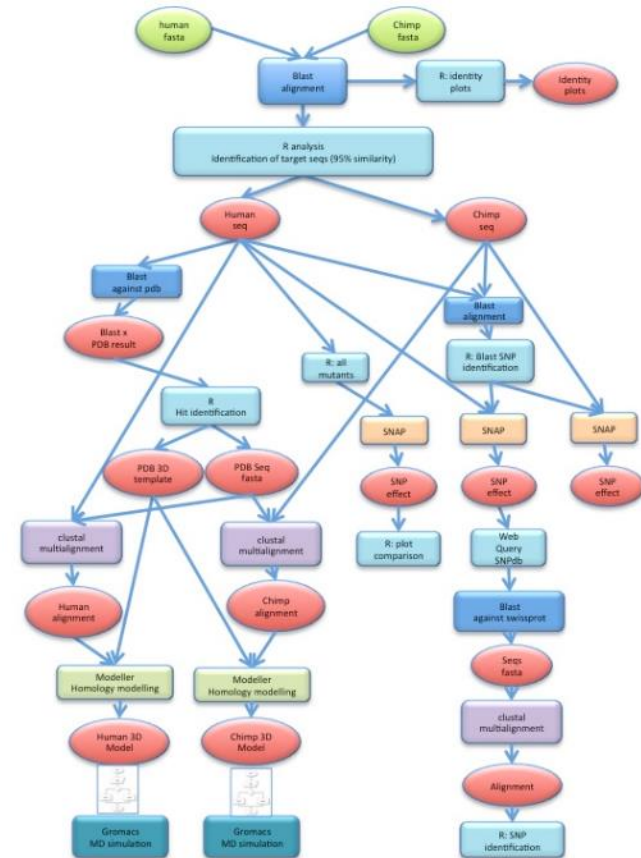
Proteomics



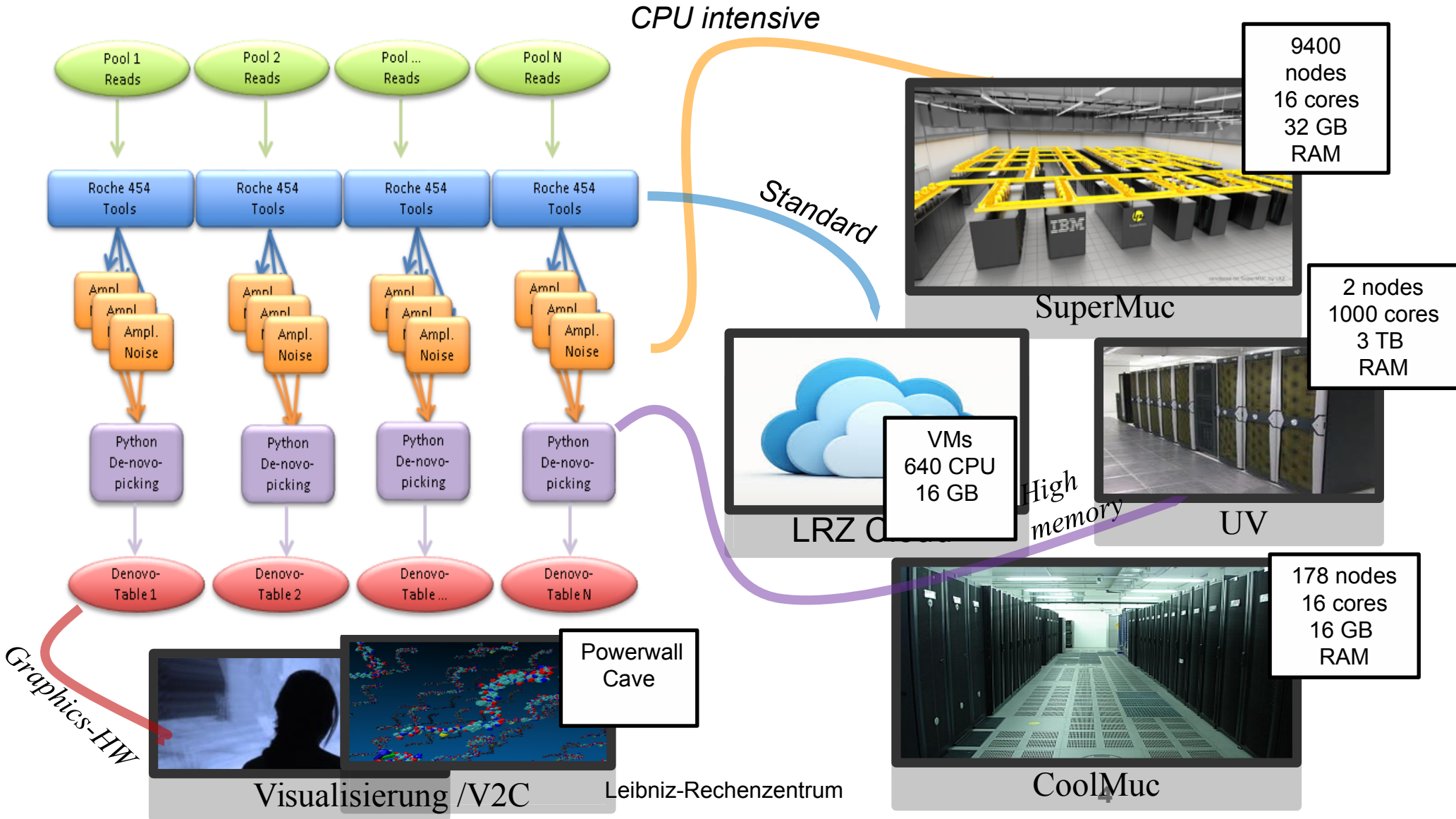
NGS



Systems Biology



Mapping on LRZ Hardware

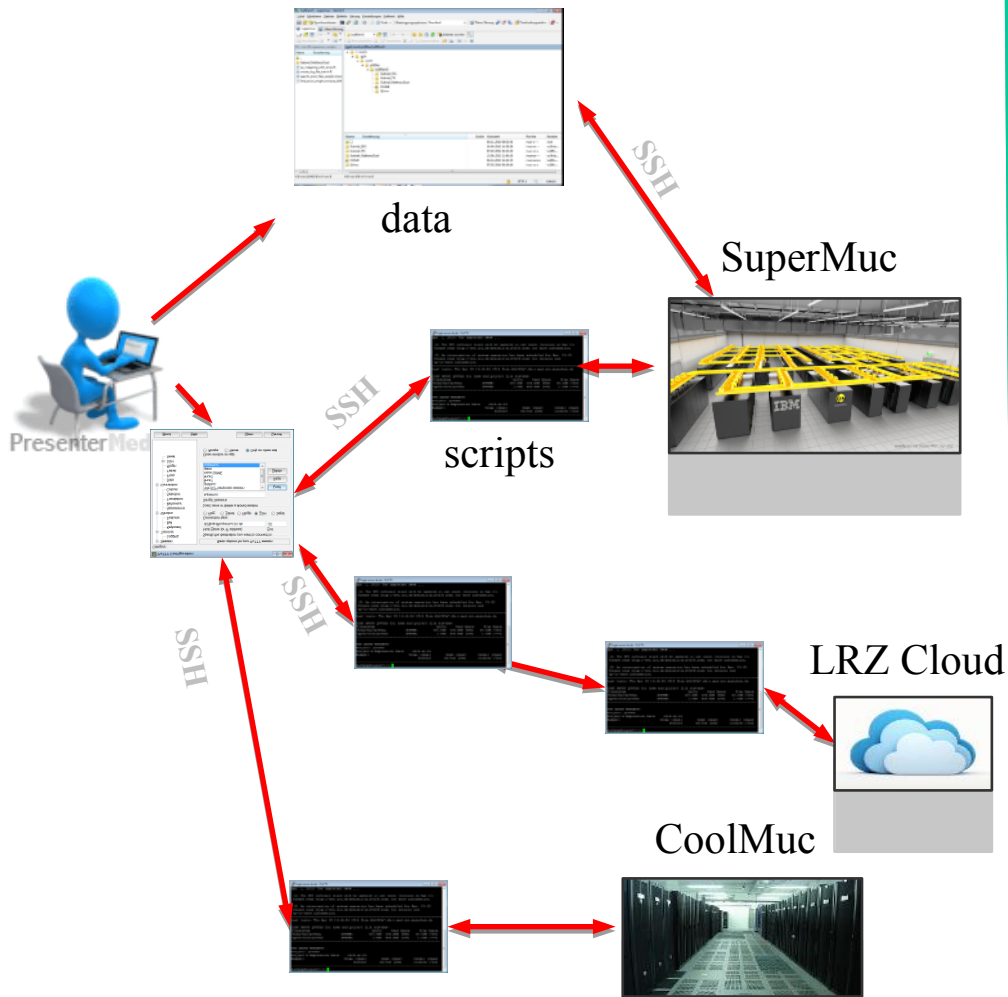




Overview: Before vs After

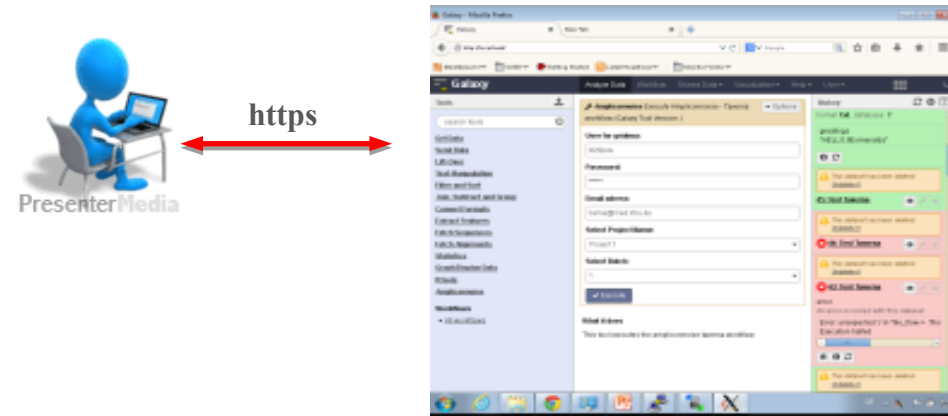
Before

LRZ – HPC Systems



After

LRZ – HPC Systems
Web-Interface

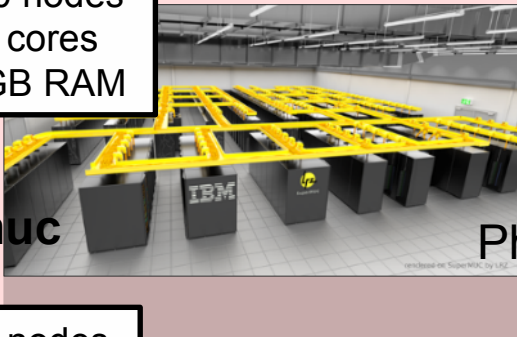




Obstacle: Separated HPC Systems



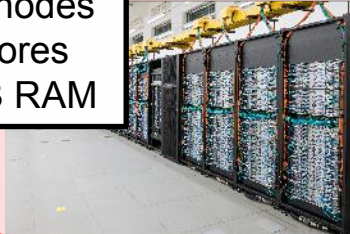
9400 nodes
16 cores
32 GB RAM



Supermuc

Phase 1

3072 nodes
28 cores
64 GB RAM



Phase 2



LRZ Linux Cluster



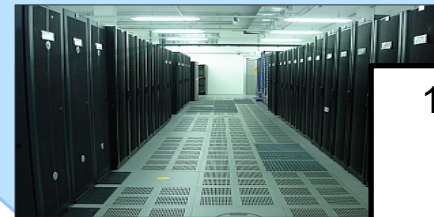
2 nodes
1000 cores
3 TB
RAM

UV



384 nodes
10752 cores
64 GB
RAM

CoolMUC2



CoolMUC

178 nodes
16 cores
16 GB
RAM



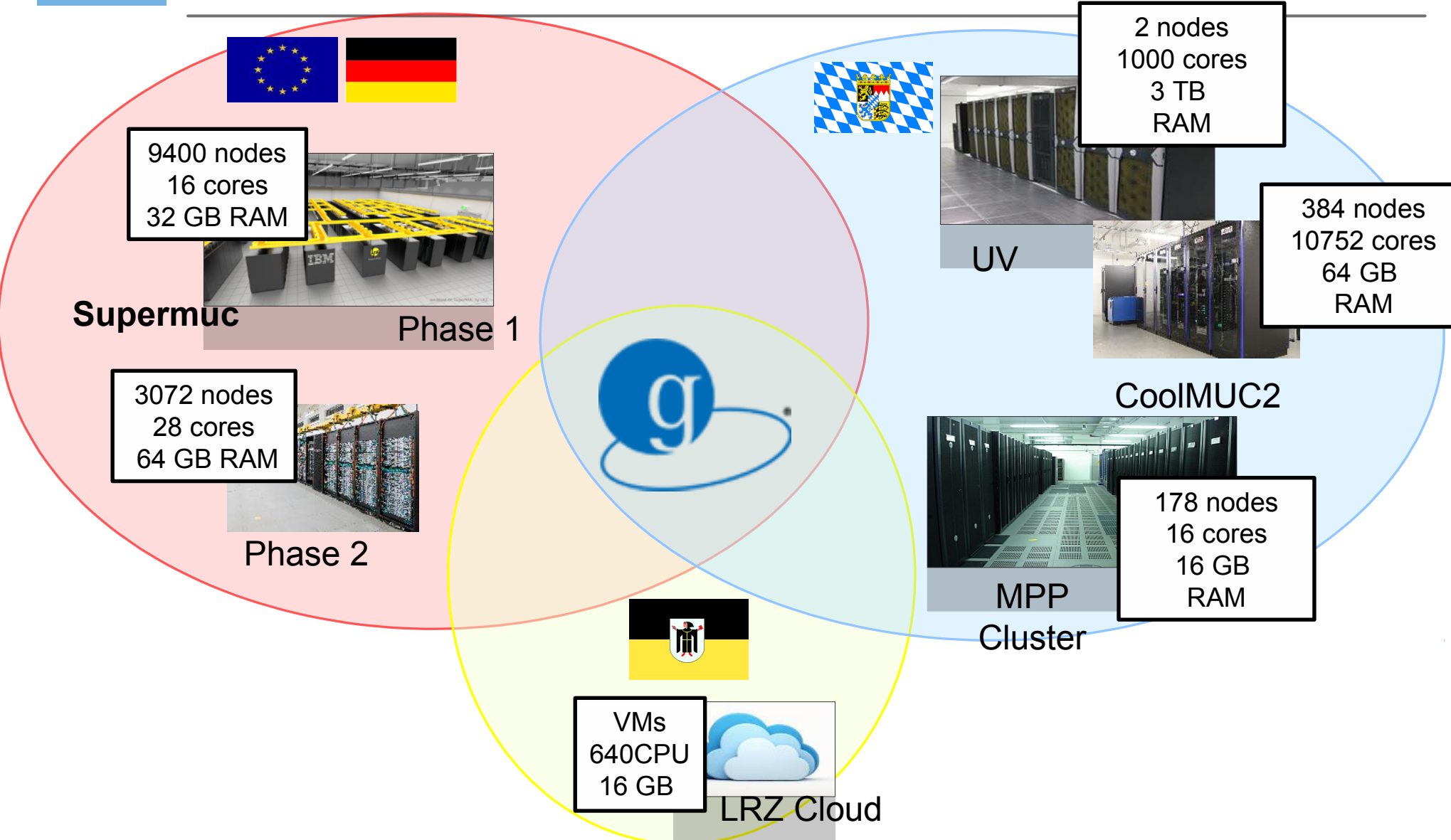
VMs
640CPU
16 GB



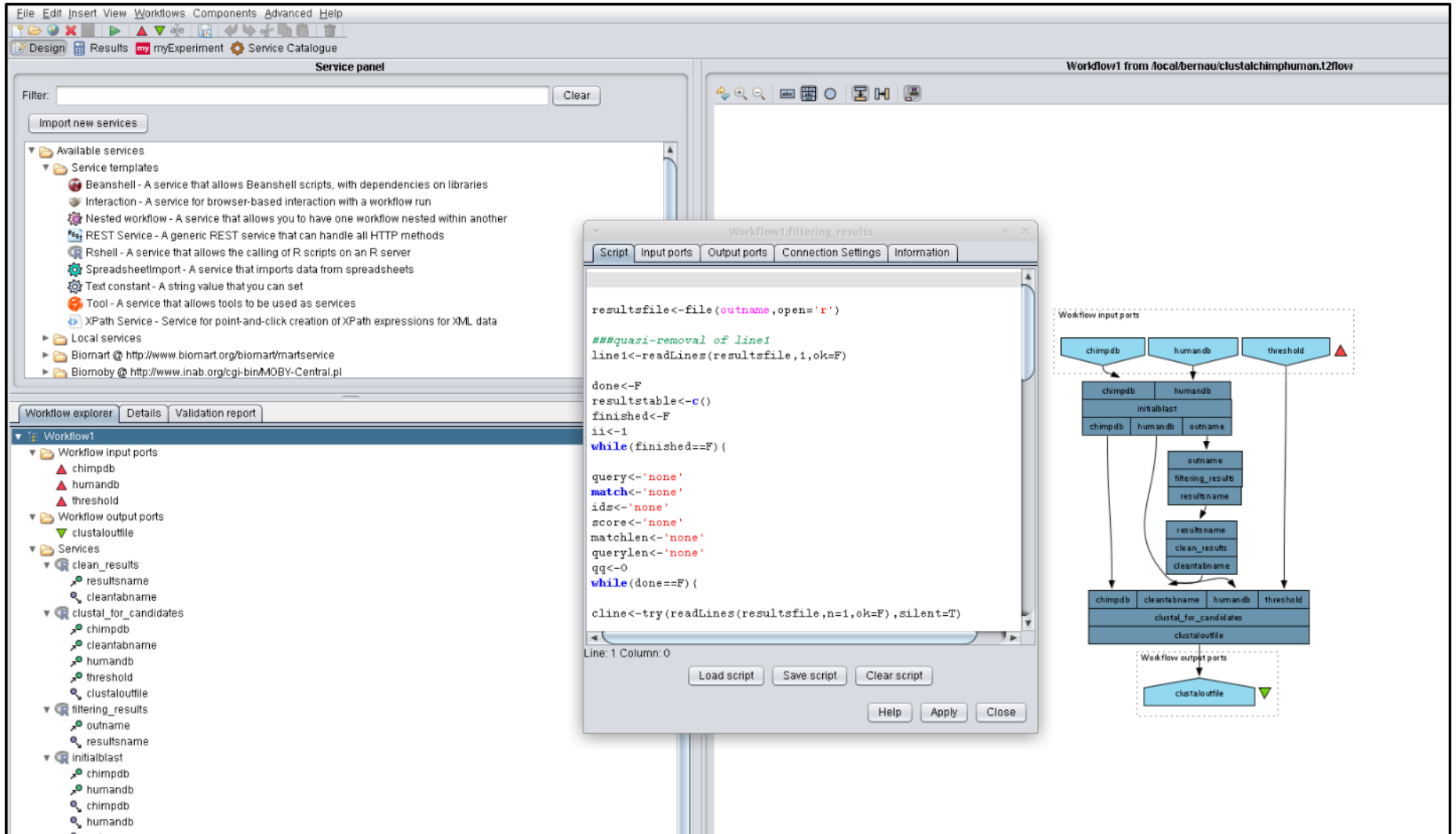
LRZ Cloud



Solution: Connect via Globus Grid Services



Workflows: Taverna Workbench using R



The screenshot displays the Taverna Workbench interface. On the left, the 'Service panel' lists various services like 'Beanshell', 'REST Service', and 'Rshell'. Below it, the 'Workflow explorer' shows the structure of 'Workflow1', including input ports (chimpdb, humandb, threshold), output ports (clustaloutfile), and services (clean_results, filtering_results, initialblast).

The central window, titled 'Workflow1:filtering_results', shows the R script used in the workflow:

```

resultsfile<-file(outname,open='r')

###quasi-removal of line1
line1<-readLines(resultsfile,1,ok=F)

done<-F
resultstable<-c()
finished<-F
ii<-1
while(finished==F){

query<-'none'
match<-'none'
ids<-'none'
score<-'none'
matchlen<-'none'
querylen<-'none'
gg<-0
while(done==F){

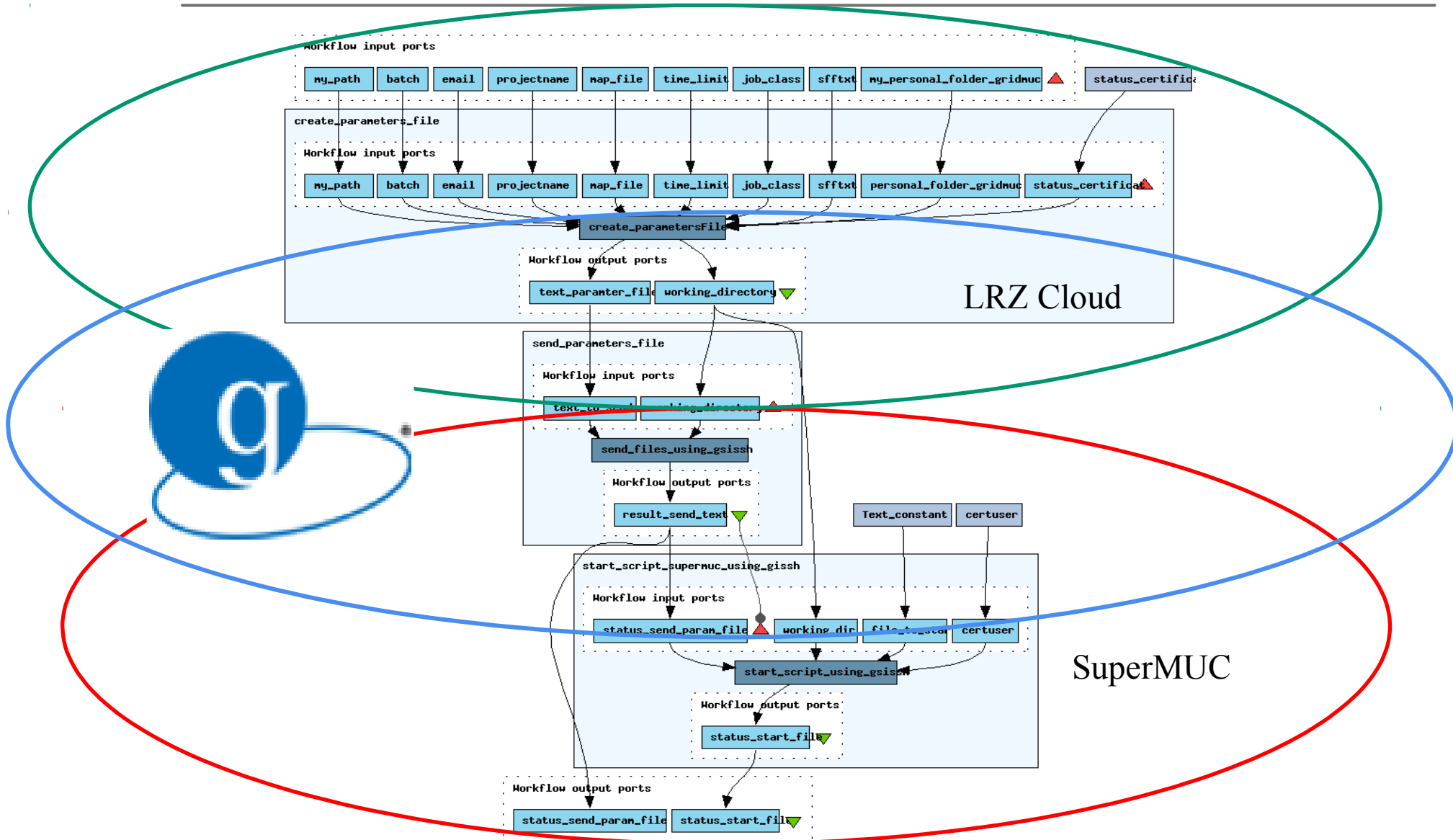
cline<-try(readLines(resultsfile,n=1,ok=F),silent=T)

```

On the right, a flow diagram illustrates the workflow's data flow. It starts with 'Workflow input ports' (chimpdb, humandb, threshold) feeding into an 'initialblast' service. The output of 'initialblast' goes to 'clean_results', which then feeds into 'filtering_results'. The output of 'filtering_results' goes to 'clean_results', which then feeds into 'clustal_for_candidates'. The output of 'clustal_for_candidates' goes to 'clustaloutfile', which is the 'Workflow output port'.



Example: Taverna workbench





User Web Interface: Galaxy



Galaxy - Mozilla Firefox

Galaxy New Tab

http://localhost/

Meistbesucht SUSE Getting Started Latest Headlines Mozilla Firefox

Galaxy Analyze Data Workflow Shared Data Visualization Help User Using 48.8

Tools

- Lift-Over
- Text Manipulation
- Filter and Sort
- Join, Subtract and Group
- Convert Formats
- Extract Features
- Fetch Sequences
- Fetch Alignments
- Statistics
- Graph/Display Data
- RTools
- Ampliconnoise**
- Workflows
 - All workflows

Ampliconnoise Execute Ampliconnoise Options

- Taverna workflow (Galaxy Tool Version .)

User for gridmuc: di29pav

Password: *****

Email address: name@med.lmu.de

Select Project Name: Project 1

Select Batch: 1

Execute

What it does

History

- format: txt, database: ?
- greetings "HELLO, \tEsmeralda"
- This dataset has been deleted. Undelete it
- 45: Test Taverna**
- This dataset has been deleted. Undelete it
- 44: Test Taverna**

Even more obstacles:

- Bioinformatic software is distributed as debian packages and cannot be easily installed on SuperMUC (SLES11)
- Solution: Linux Containers
 - docker: security problems, has to run as root
 - proot: linux containers in userspace
- re-mount your own $\$HOME/var$ to $/var$, re-mount your own root folder and fake root account

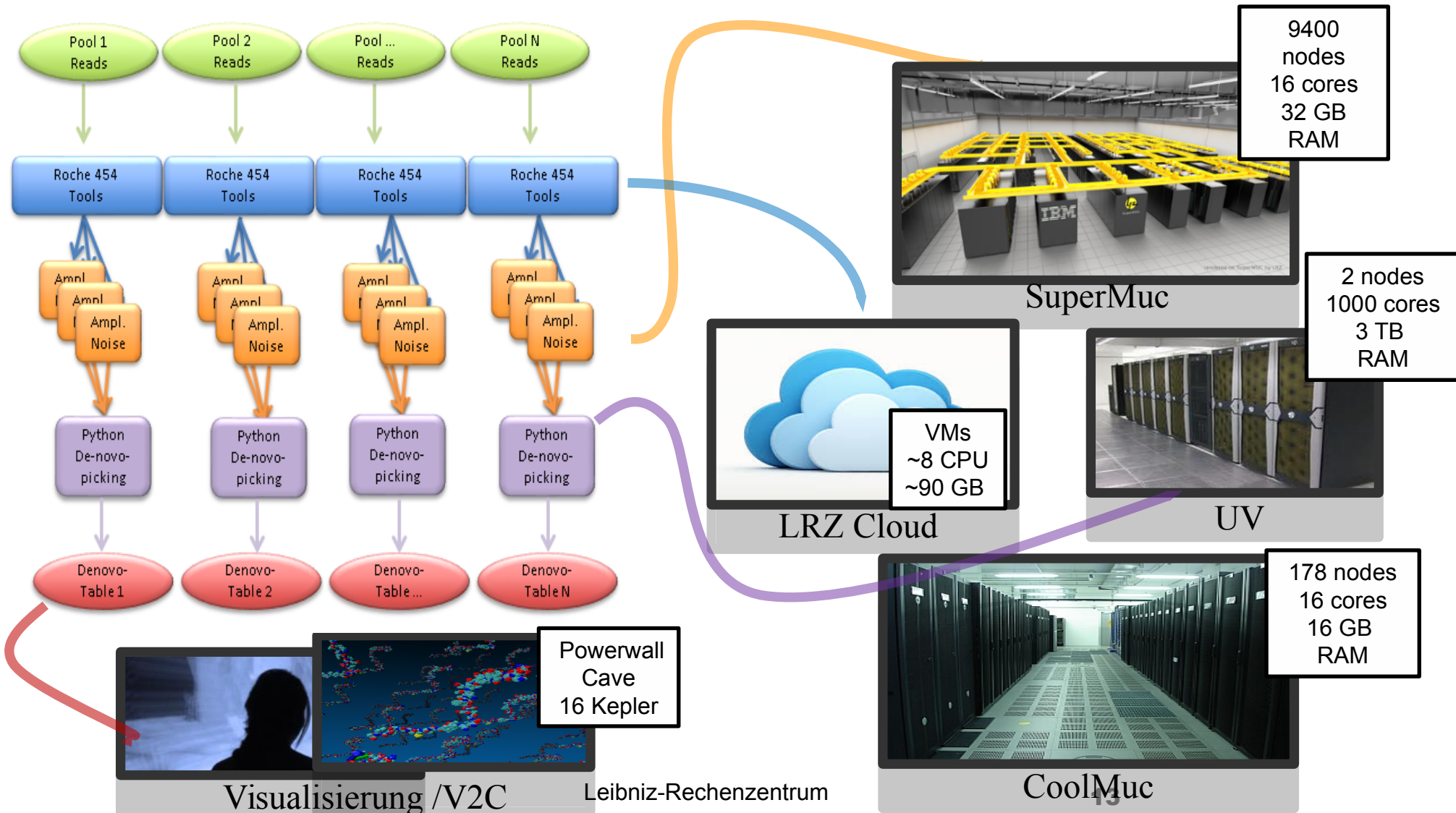
```
$ proot -b  $\$HOME/var$ :/var -r  $\$HOME/debian8$  -0
```

- You can trick software packages into believing they are root and they can write to system folders
- Everything is fake. Programs may break if they really need root access.
- You can fake e.g. a debian distribution running on suse. Debian can access /proc, /var, /dev and /etc like it was its own.
- Download at <http://proot.me>
- proot can not modify kernel and drivers!
- Who uses it?
 - Sony, STMicroelectronics, Ericsson, Cisco



Putting it all together

openNebula + taverna + globus + proot + virtualGL + galaxy + R/python





Summary and Conclusions: Virtualization

Taverna/R

- Taverna workbench is a flexible, powerful tool for designing workflows
- tools are designed as web services and can thus easily be moved to other resources
- Workflow developers in our test projects clearly favoured it over Galaxy
- Dislike: Taverna does not provide a working web interface (no multi-user interface)

Galaxy/python

- Nice web interface to make workflows available to end users
- Inconvenient for workflow developers (server needs to be restarted for changes to take effect)
- Dislike: runs everything under same account, no user file protection

Globus

- Available and working on all LRZ systems (almost an unique characteristic)
- Provides certificates and user mapping
- Provides a convenient way to automatically request certificates at login or workflow runtime via LRZMyProxy

Proot

- Run software in lightweight container
- Good security model
- Dislike: Some programs may not work correctly (system files, low level drivers)



Thanks for your attention!

Project Members (LRZ Application Labs):

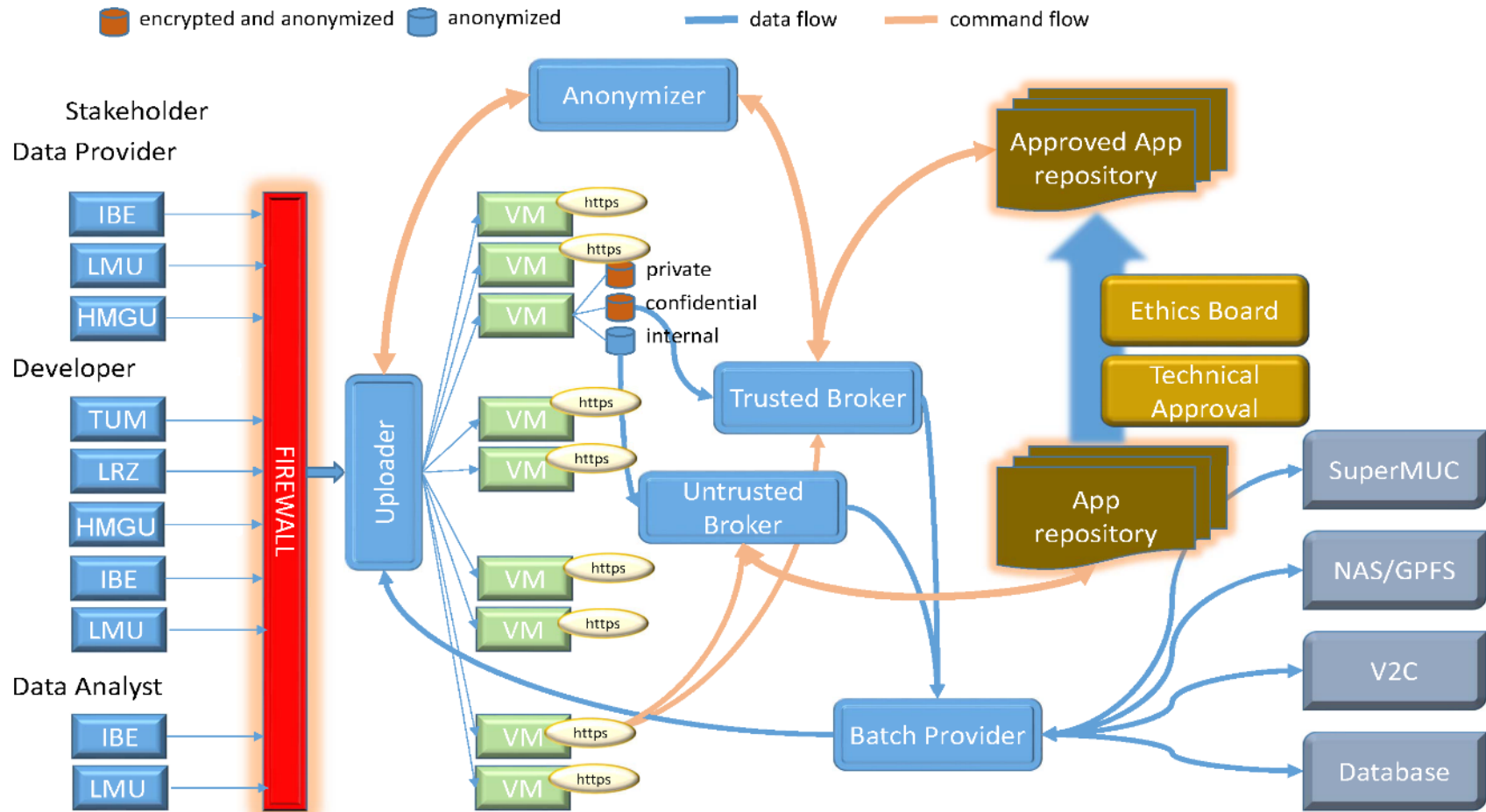
- Ferdinand Jamitzky
- Christoph Bernau
- Shaila Rössle-Blank
- Yu Wang
- Helmut Satzger
- Momme Allalen
- Gerald Mathias

Collaborators:

- LMU Munich (Esmeralda Vicedo, Markus Ege)
- TU Munich (Andrea Schafferhans, Timothy Karl, Burkhard Rost)
- Max-von-Pettenkofer-Institute (Debora Garzetti)

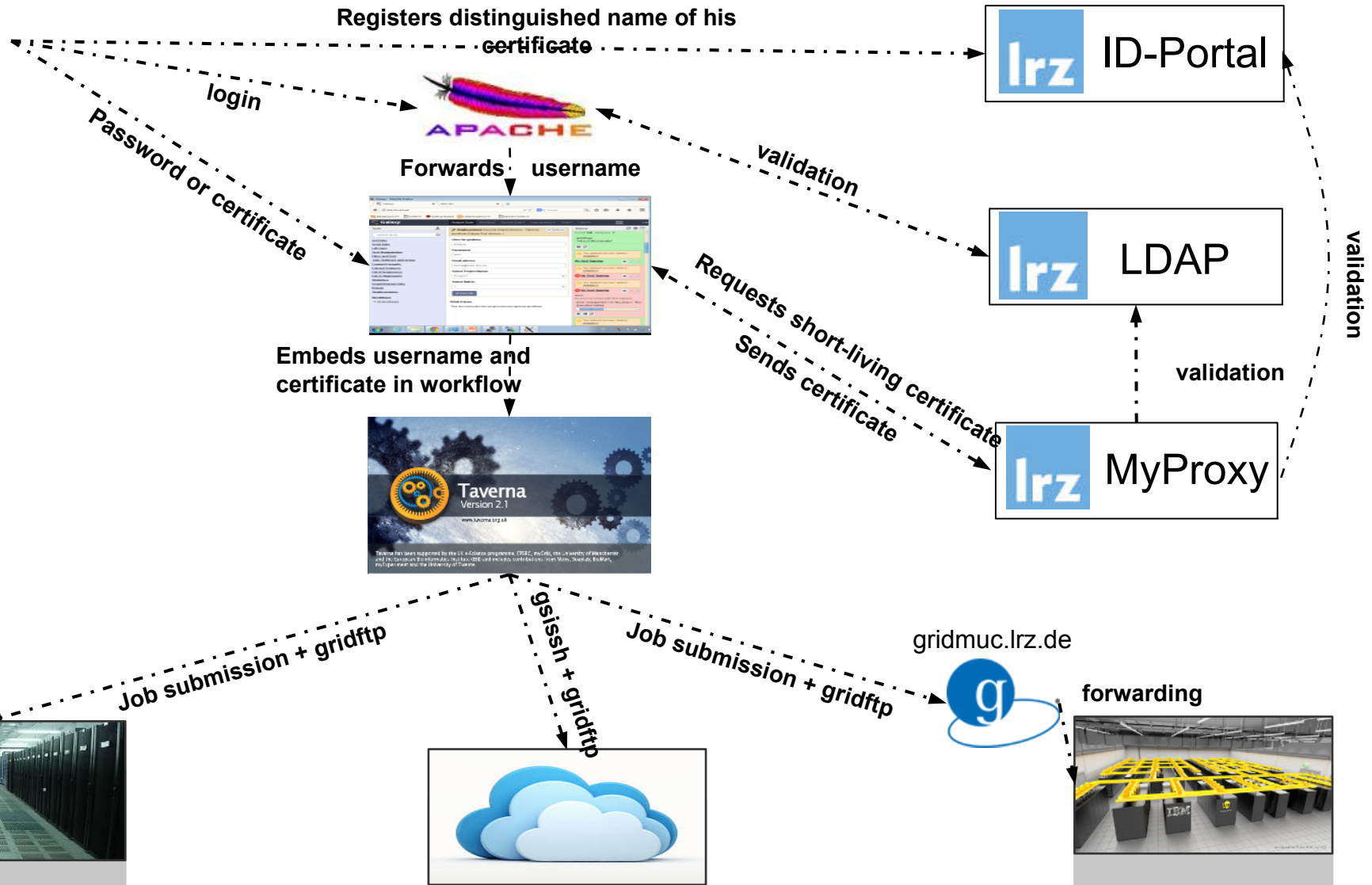
Outlook 2020: BG.DAF

Bavarian Genome Data and Analysis Facility





Platform overview: current status



Linux Containers: Why?

- Software package xxx needs to write to a system directory during installation (e.g. write to /var)
- You have a rpm and want to install it in another place (e.g. /lrz/sys)
- You want to install debian software packages on SUSE
- Your program requires root user to be run, but does not need root privileges. (e.g. apt-get)

How is it done?

ptrace is a system call found in several Unix and Unix-like operating systems. By using ptrace (the name is an abbreviation of "process trace") one process can control another, enabling the controller to inspect and manipulate the internal state of its target. ptrace is used by debuggers and other code-analysis tools, mostly as aids to software development.

Solution: Linux Container using proot

- re-mount your own \$HOME/var to /var:

```
$ proot -b $HOME/var:/var
```

- re-mount your own root folder

```
$ proot -r $HOME/debian8
```

- fake root account

```
$ proot -0
```

