# INTERACTIVE HPC WITH JUPYTERLAB

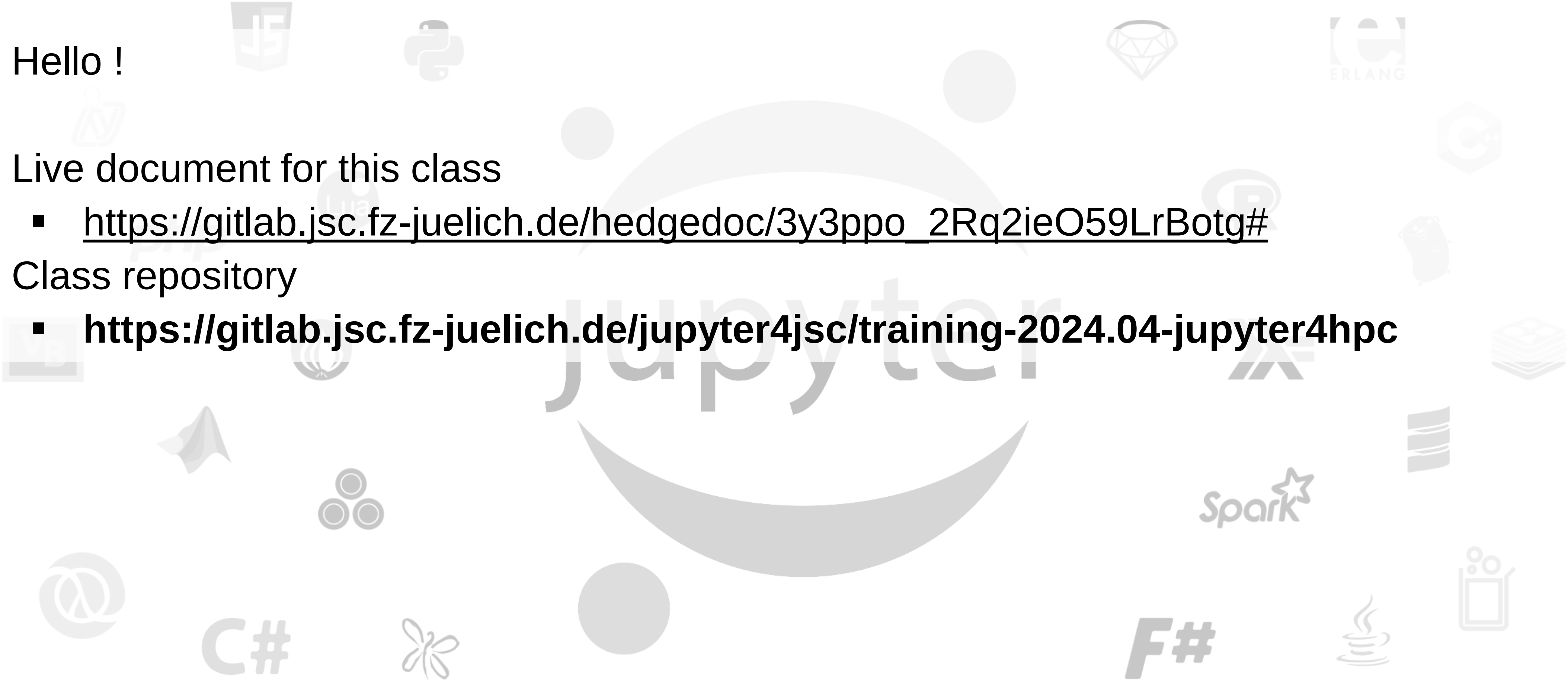
## Training Course – Welcome Day 1

2024-04-22..23 I JENS HENRIK GÖBBERT (J.GOEBBERT@FZ-JUELICH.DE)
HERWIG ZILKEN (H.ZILKEN@FZ-JUELICH.DE)

# WELCOME

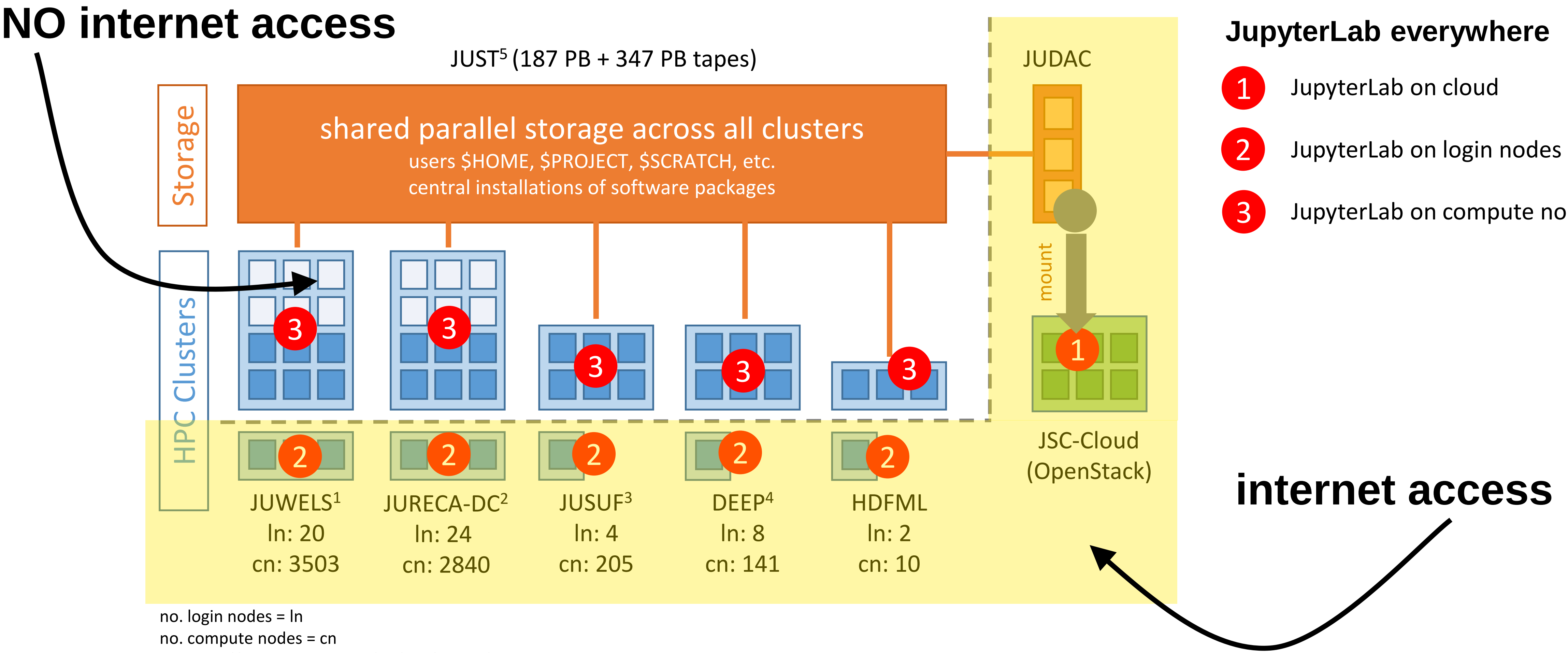
- Hello !

- Live document for this class
  - https://gitlab.jsc.fz-juelich.de/hedgedoc/3y3ppo_2Rq2ieO59LrBotg#
- Class repository
  - **https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc**



https://jupyter.org

# JUPYTERLAB EVERYWHERE



**JupyterLab everywhere**

1 JupyterLab on cloud

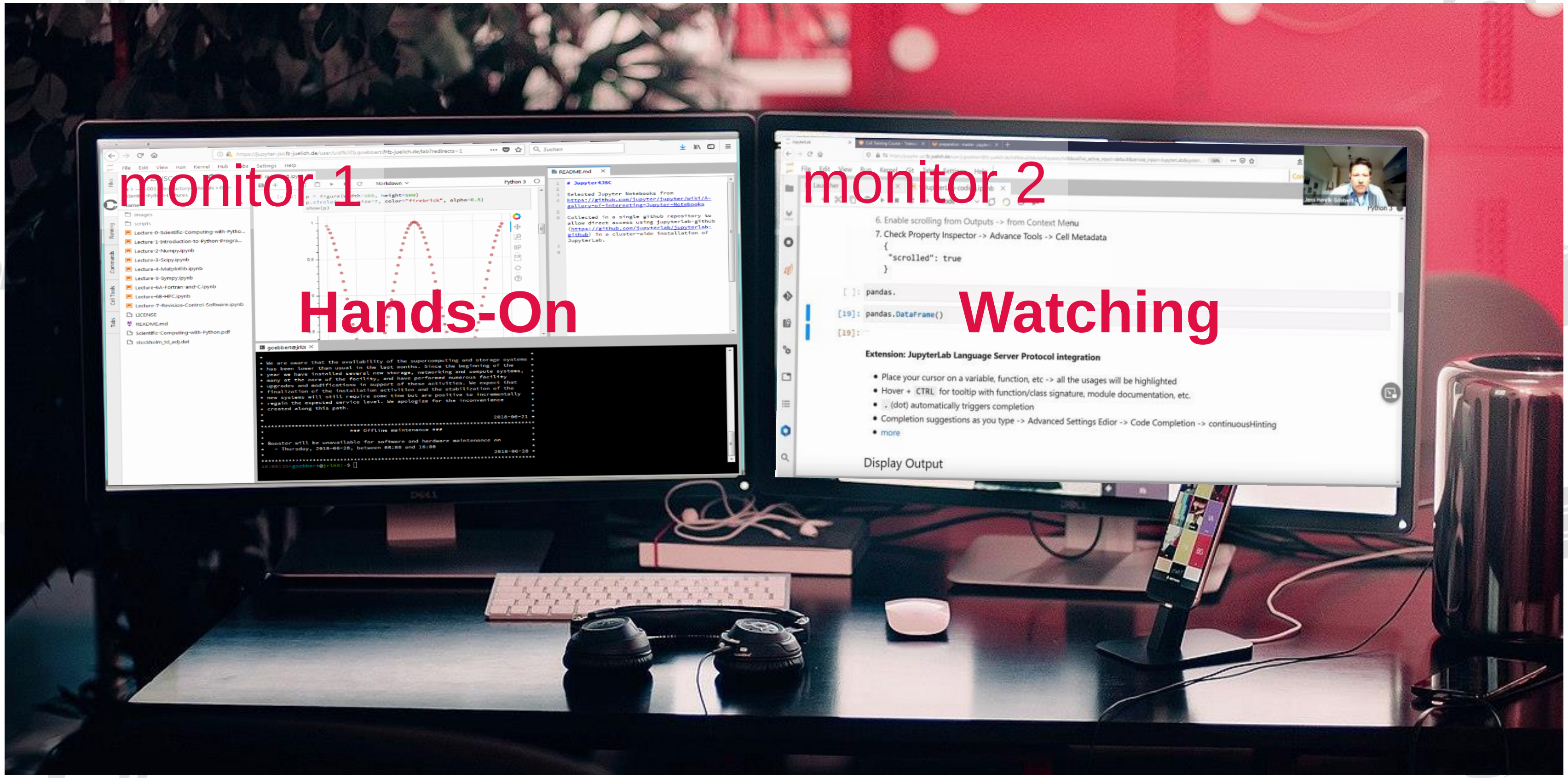
2 JupyterLab on login nodes

3 JupyterLab on compute nodes

no. login nodes = ln
no. compute nodes = cn

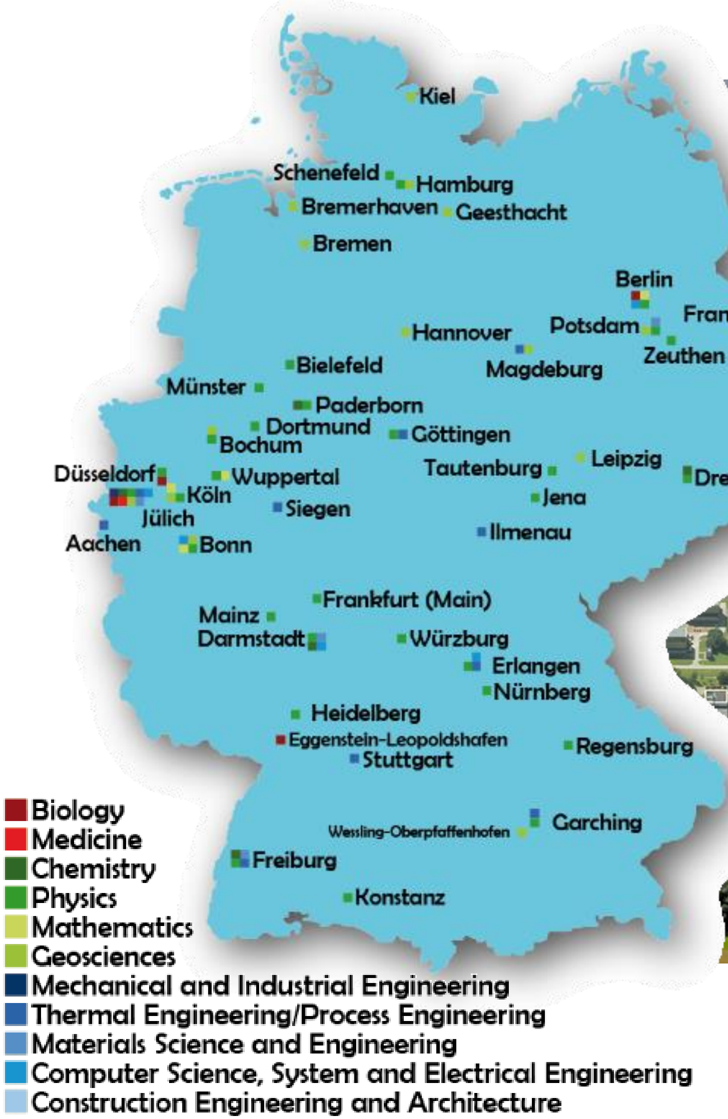
[1] https://apps.fz-juelich.de/jsc/hps/juwels/configuration.html
[2] https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html
[3] https://apps.fz-juelich.de/jsc/hps/jusuf/configuration.html
[4] https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system
[5] https://apps.fz-juelich.de/jsc/hps/just/configuration.html

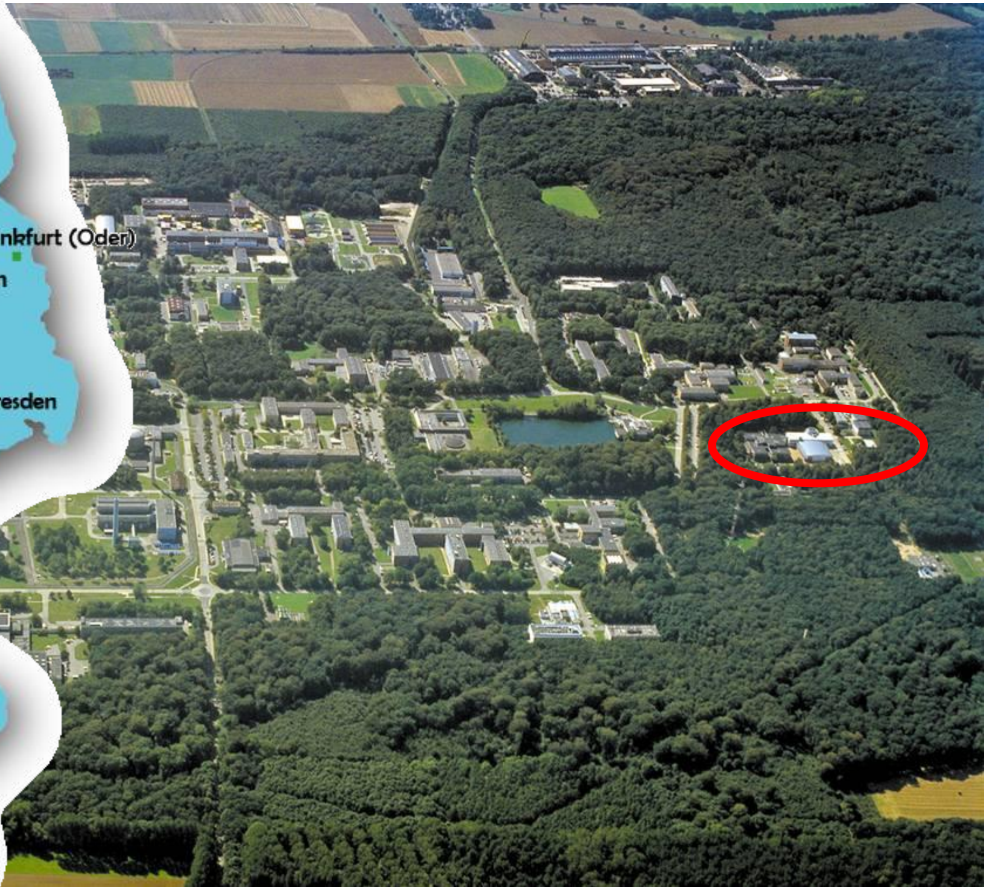
# RECOMMENDED EQUIPMENT

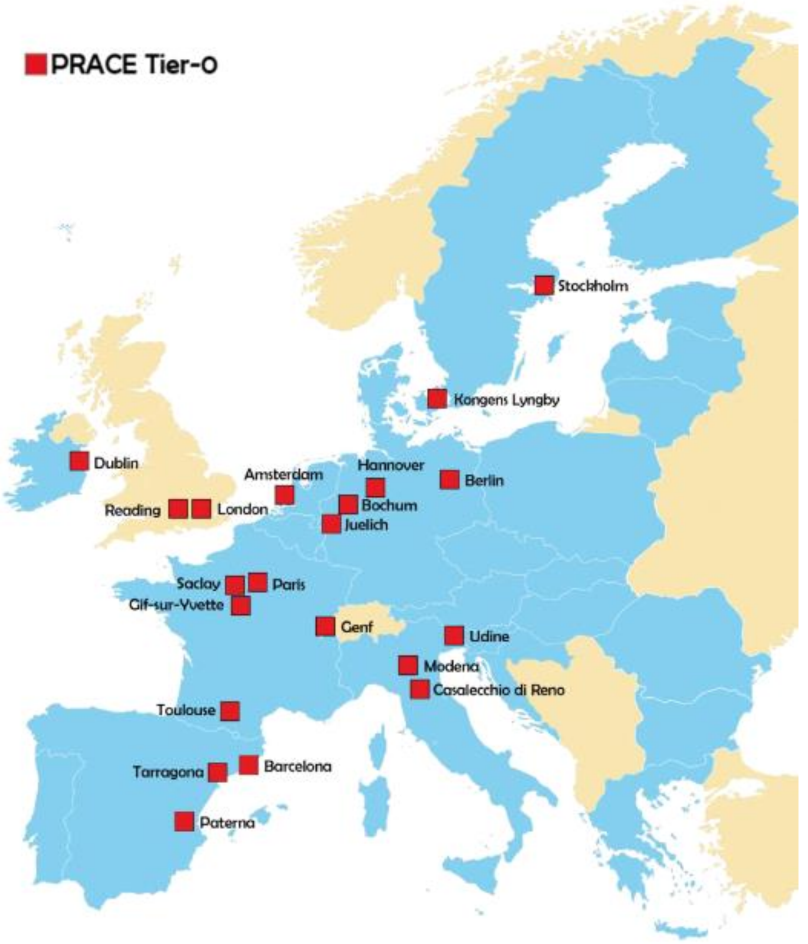


# JÜLICH SUPERCOMPUTING CENTRE

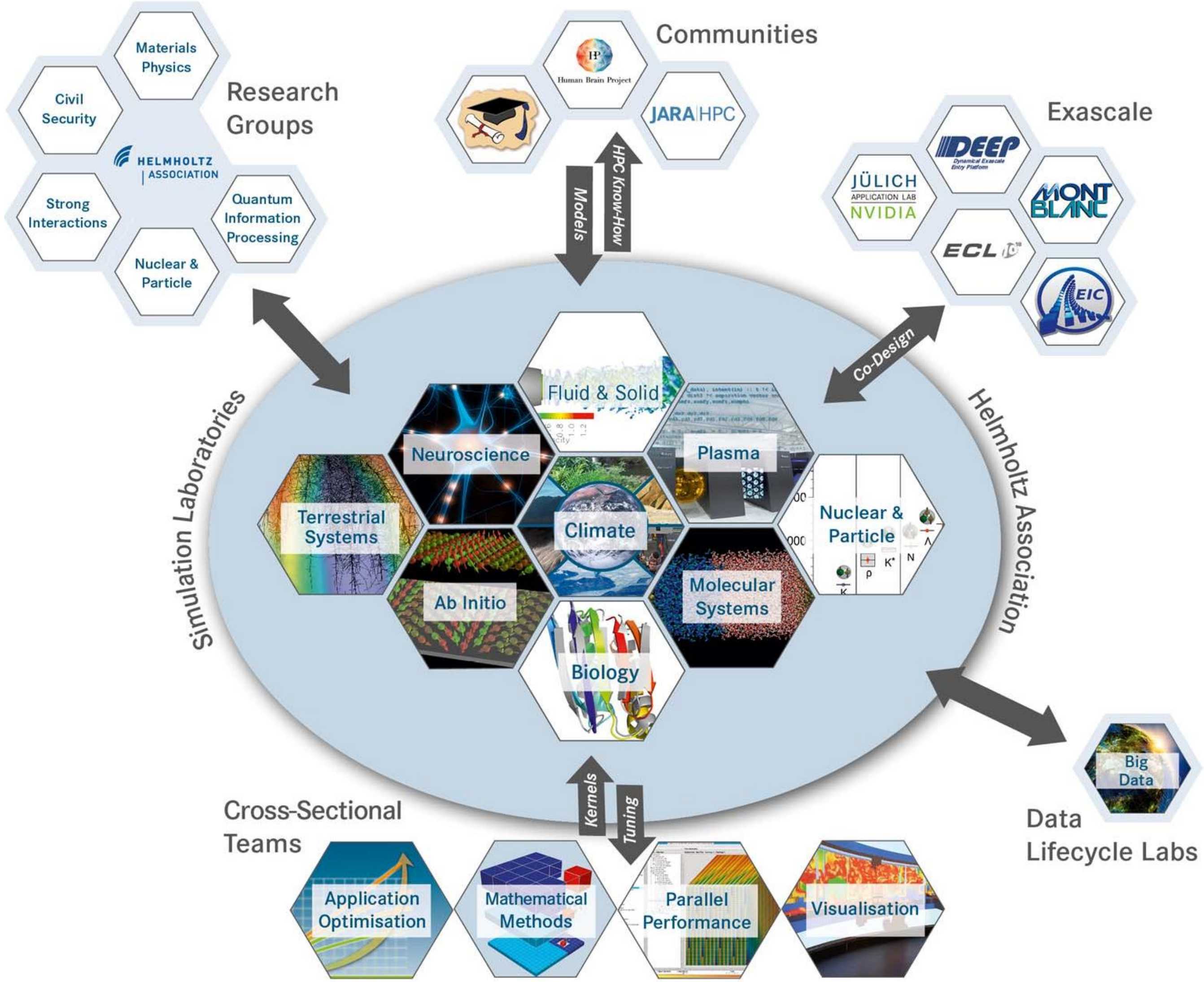
# JÜLICH SUPERCOMPUTING CENTRE (JSC)
# TIER-0/1 HPC RESOURCES OF THE HIGHEST PERF. CLASS



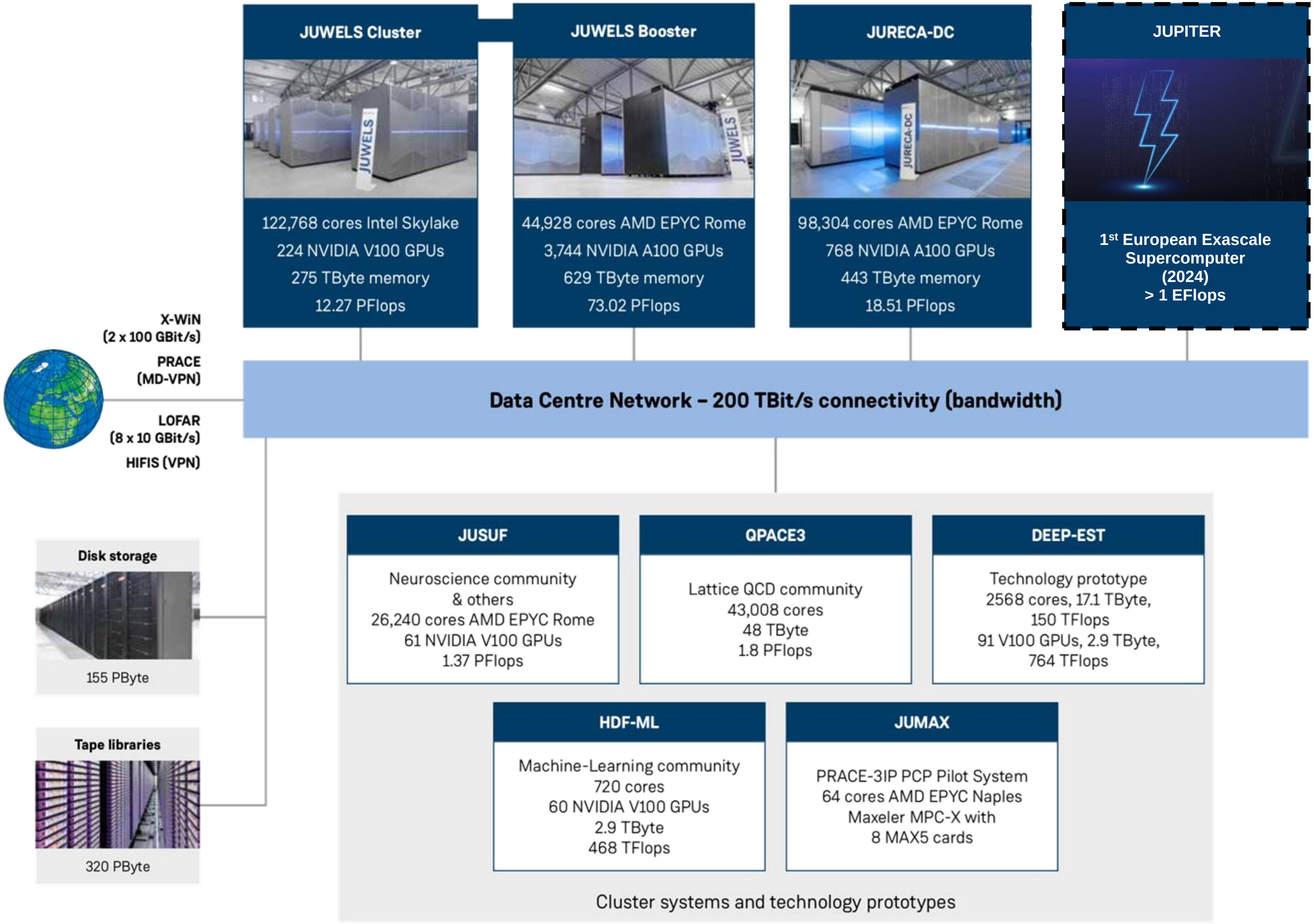




# DOMAIN SPECIFIC SUPPORT



JÜLICH Forschungszentrum



**JUWELS Cluster**

122,768 cores Intel Skylake
224 NVIDIA V100 GPUs
275 TByte memory
12.27 PFlops

**JUWELS Booster**

44,928 cores AMD EPYC Rome
3,744 NVIDIA A100 GPUs
629 TByte memory
73.02 PFlops

**JURECA-DC**

98,304 cores AMD EPYC Rome
768 NVIDIA A100 GPUs
443 TByte memory
18.51 PFlops

**JUPITER**

1st European Exascale
Supercomputer
(2024)
> 1 EFlops

X-WiN
(2 x 100 GBit/s)
PRACE
(MD-VPN)
LOFAR
(8 x 10 GBit/s)
HIFIS (VPN)

**Data Centre Network – 200 TBit/s connectivity (bandwidth)**

**Disk storage**
155 PByte

**Tape libraries**
320 PByte

**JUSUF**

Neuroscience community
& others
26,240 cores AMD EPYC Rome
61 NVIDIA V100 GPUs
1.37 PFlops

**QPACE3**

Lattice QCD community
43,008 cores
48 TByte
1.8 PFlops

**DEEP-EST**

Technology prototype
2568 cores, 17.1 TByte,
150 TFlops
91 V100 GPUs, 2.9 TByte,
764 TFlops

**HDF-ML**

Machine-Learning community
720 cores
60 NVIDIA V100 GPUs
2.9 TByte
468 TFlops

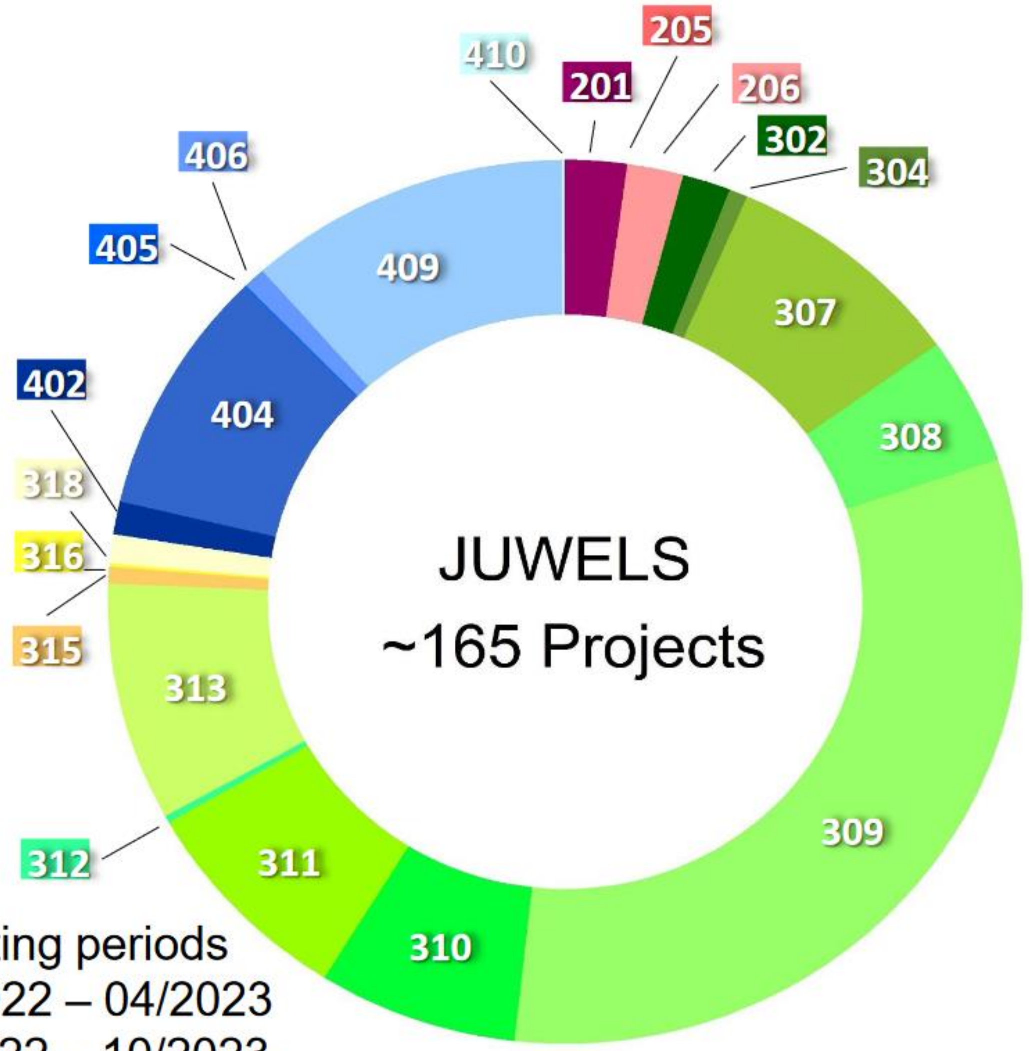
**JUMAX**

PRACE-3IP PCP Pilot System
64 cores AMD EPYC Naples
Maxeler MPC-X with
8 MAX5 cards

Cluster systems and technology prototypes

JÜLICH
Forschungszentrum

# RESEARCH FIELDS ON JUWELS (CLUSTER + BOOSTER)

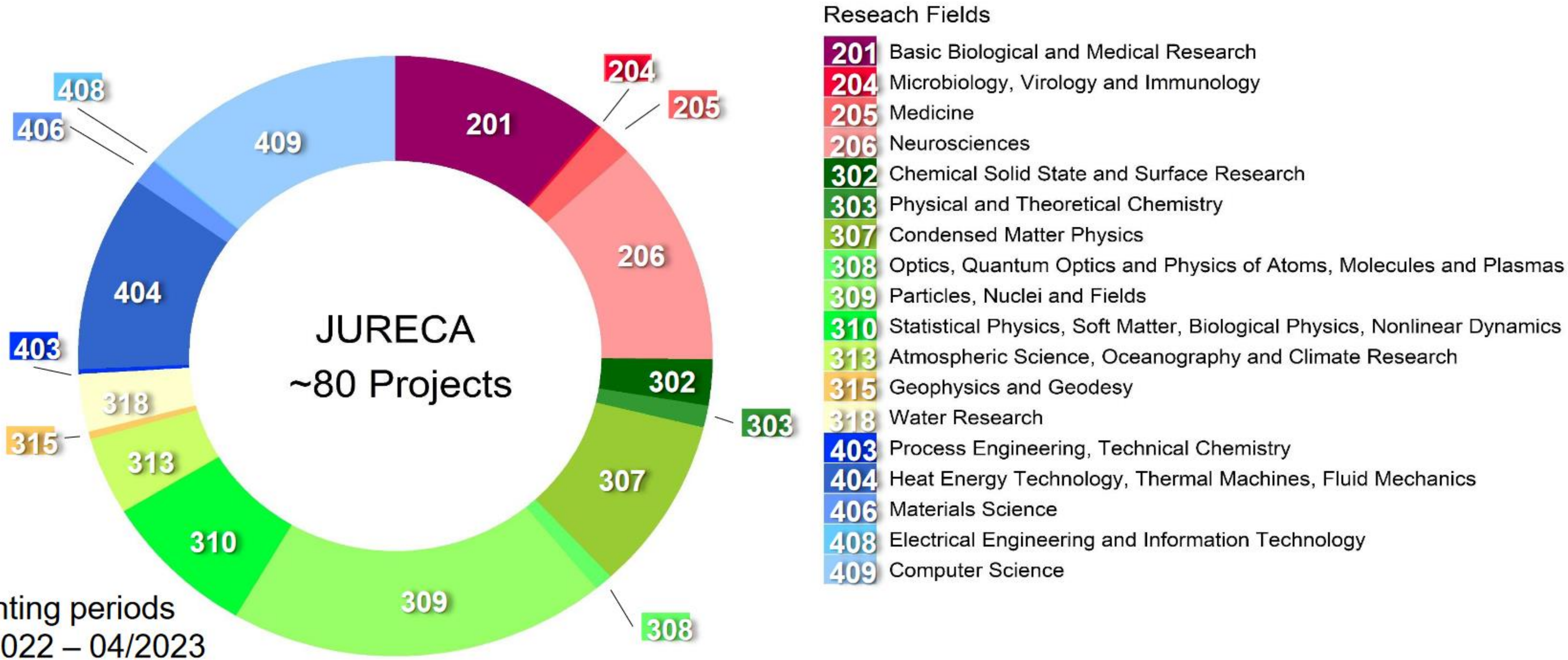


Reseach Fields

| | |
|---|---|
| 201 | Basic Biological and Medical Research |
| 205 | Medicine |
| 206 | Neurosciences |
| 302 | Chemical Solid State and Surface Research |
| 304 | Analytical Chemistry, Method Development (Chemistry) |
| 307 | Condensed Matter Physics |
| 308 | Optics, Quantum Optics and Physics of Atoms, Molecules and Plasmas |
| 309 | Particles, Nuclei and Fields |
| 310 | Statistical Physics, Soft Matter, Biological Physics, Nonlinear Dynamics |
| 311 | Astrophysics and Astronomy |
| 312 | Mathematics |
| 313 | Atmospheric Science, Oceanography and Climate Research |
| 315 | Geophysics and Geodesy |
| 316 | Geochemistry, Mineralogy and Crystallography |
| 318 | Water Research |
| 402 | Mechanics and Constructive Mechanical Engineering |
| 404 | Heat Energy Technology, Thermal Machines, Fluid Mechanics |
| 405 | Materials Engineering |
| 406 | Materials Science |
| 409 | Computer Science |
| 410 | Construction Engineering and Architecture |

Granting periods
05/2022 – 04/2023
11/2022 – 10/2023

# RESEARCH FIELDS ON JURECA (CLUSTER + BOOSTER)



Granting periods
05/2022 – 04/2023
11/2022 – 10/2023

## JUSUF (Jülich Support for Fenix)

- Serves the ICEI project (**I**nteractive **C**omputing **E**-**I**nfrastructure for the Human Brain Project) as part of the EU **Fenix e-infrastructure**.
- Contains 2 partitions, **HPC and Cloud**, which sizes are reconfigurable according to demand.
- Air-cooled, less dense than other systems
- Operation started in May 2020
- **Project partners: Atos, NVIDIA, ParTec**

## System architecture

HPC partition:

- 124 compute nodes + 49 GPU nodes (incl. V100 NVIDIA GPUs)
  - **2× 64-core** AMD Epyc 7742 Rome CPUs
  - **2x 128 GB** DDR4 @ 3.2 GHz
  - 1x HDR100 InfiniBand adapter (100Gbps)
  - 1x 40 GbE adapter (for storage)
  - **1TB NVMe** local scratch

Cloud partition:

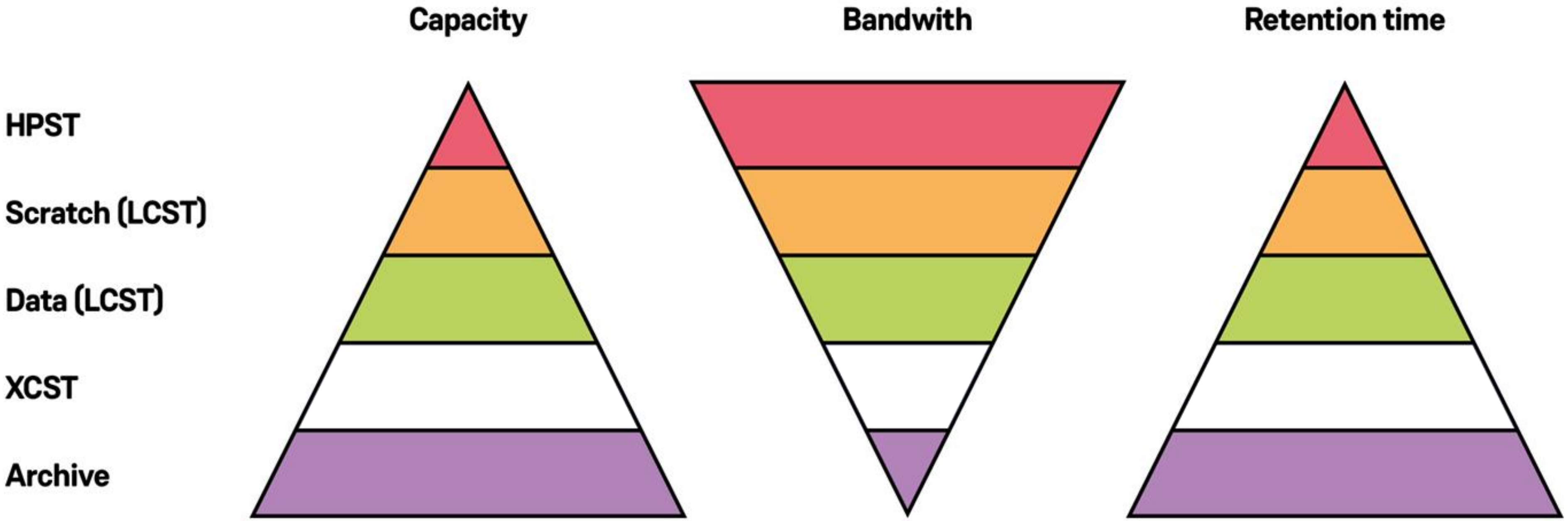
- 4 compute nodes + 12 GPU nodes

(HPC/Cloud partitions are reconfigurable according to demand)

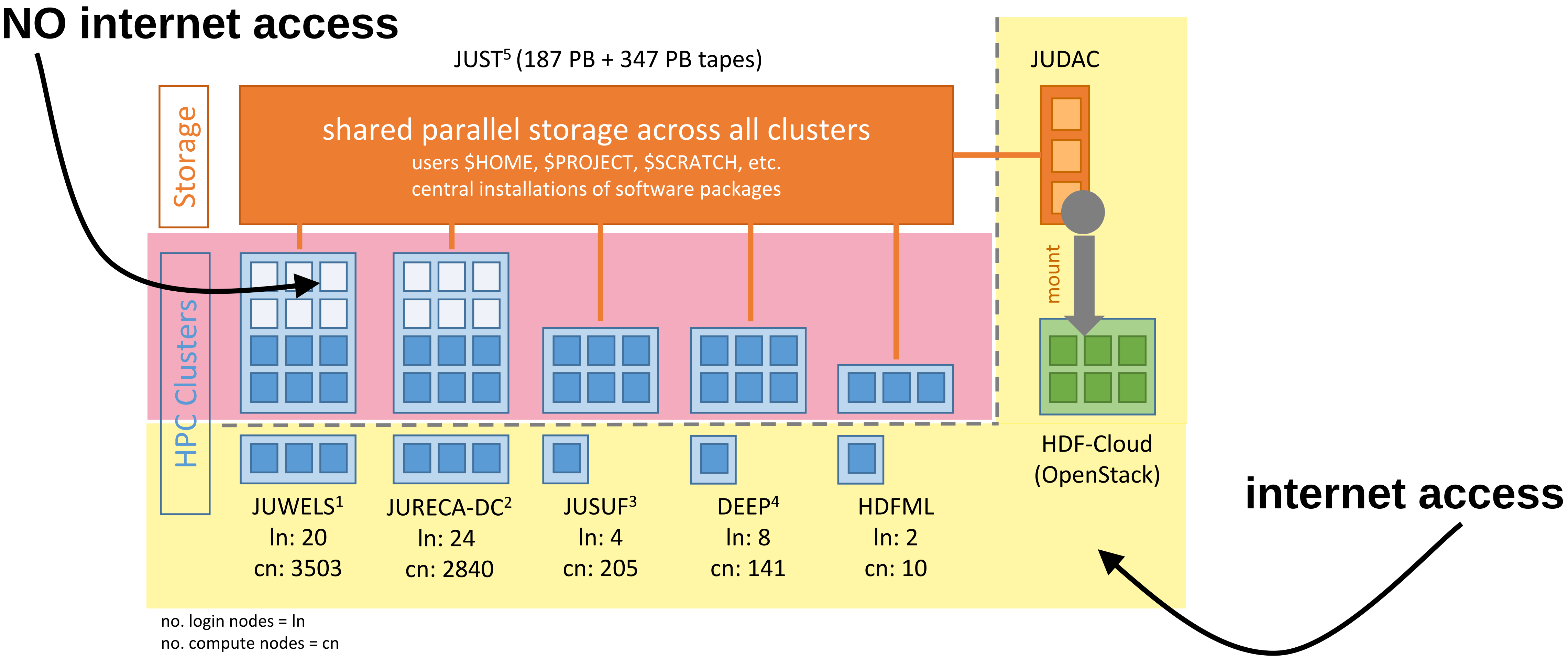
https://apps.fz-juelich.de/jsc/hps/jusuf/ https://fenix-ri.eu/

## JUST (Juelich Storage cluster)

- One central storage infrastructure for HPC
- Total gross capacity
  - NVMe disks: ~2 PB
  - Spinning disks: ~187 PB
  - Tape: ~347 PB
- Software:
  - IBM Spectrum Scale
  - IBM Spectrum Protect
  - DDN Infinite Memory Engine
- **Project partners: DDN, IBM, Lenovo, ProCom**



# SUMMARY – COMPUTE RESOURCES @ JSC



NO internet access

JUST[5] (187 PB + 347 PB tapes)

Storage

shared parallel storage across all clusters

users $HOME, $PROJECT, $SCRATCH, etc.

central installations of software packages

JUDAC

mount

HPC Clusters

| JUWELS[1] | JURECA-DC[2] | JUSUF[3] | DEEP[4] | HDFML |
|---|---|---|---|---|
| ln: 20 | ln: 24 | ln: 4 | ln: 8 | ln: 2 |
| cn: 3503 | cn: 2840 | cn: 205 | cn: 141 | cn: 10 |

HDF-Cloud
(OpenStack)

internet access

no. login nodes = ln

no. compute nodes = cn

[1] https://apps.fz-juelich.de/jsc/hps/juwels/configuration.html

[2] https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html

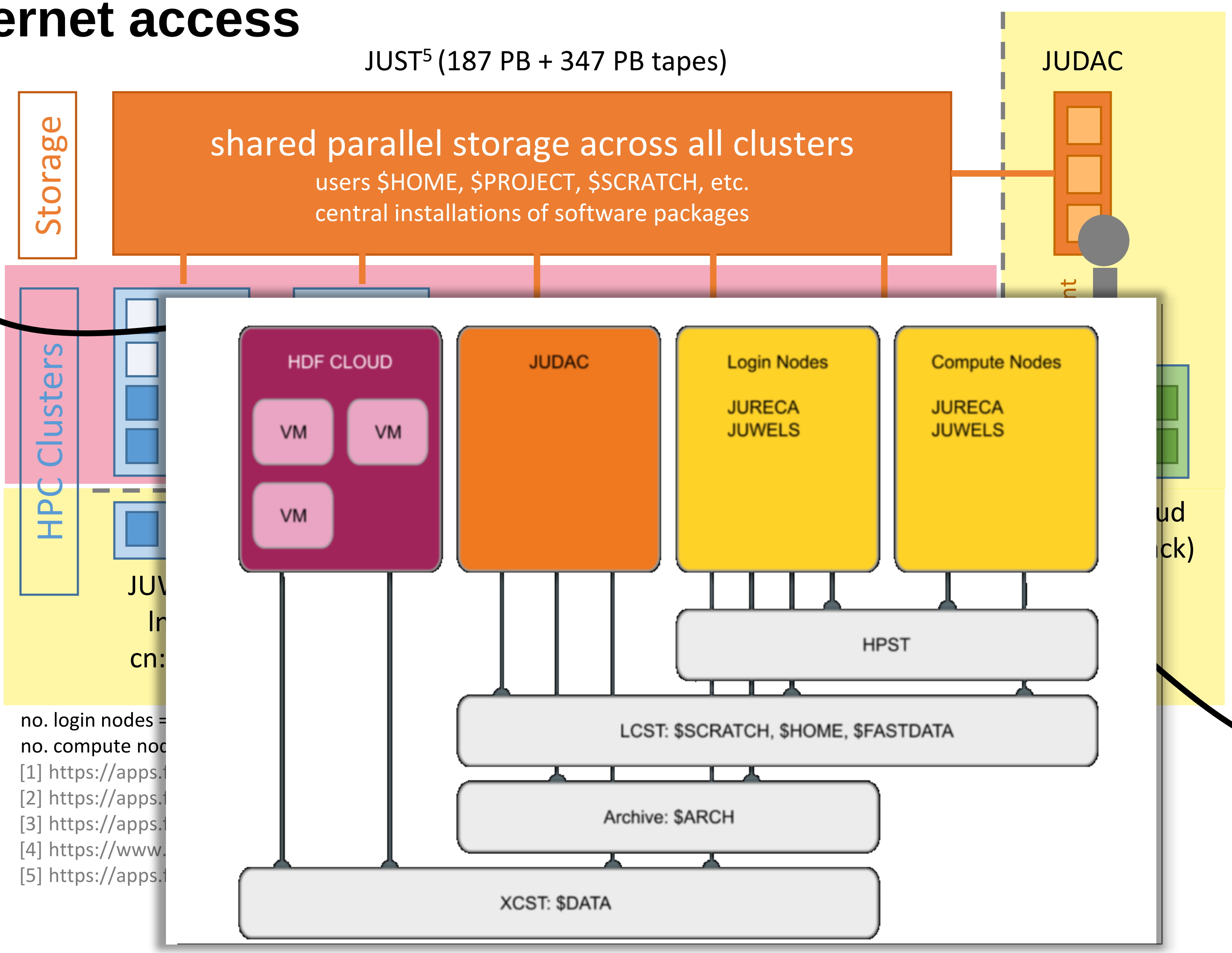
[3] https://apps.fz-juelich.de/jsc/hps/jusuf/configuration.html

[4] https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system

[5] https://apps.fz-juelich.de/jsc/hps/just/configuration.html

# SUMMARY – COMPUTE RESOURCES @ JSC

NO internet access



JUST5 (187 PB + 347 PB tapes)

JUDAC

Storage

shared parallel storage across all clusters

users $HOME, $PROJECT, $SCRATCH, etc.

central installations of software packages

HPC Clusters

JUV

In

cn:

nt

ud

ck)

no. login nodes =

no. compute noc

[1] https://apps.f

[2] https://apps.f

[3] https://apps.f

[4] https://www.

[5] https://apps.f

HDF CLOUD

VM

VM

VM

JUDAC

Login Nodes

JURECA
JUWELS

Compute Nodes

JURECA
JUWELS

HPST

LCST: $SCRATCH, $HOME, $FASTDATA

Archive: $ARCH

XCST: $DATA

internet access

# ACCESS
# TO COMPUTE RESOURCES

# PRE-ACCESS TODOS

**1) Register & Login**

✓ https://judoor.fz-juelich.de

**2) Join** the project „training2412 “

✓ Wait to get joined by the project PI

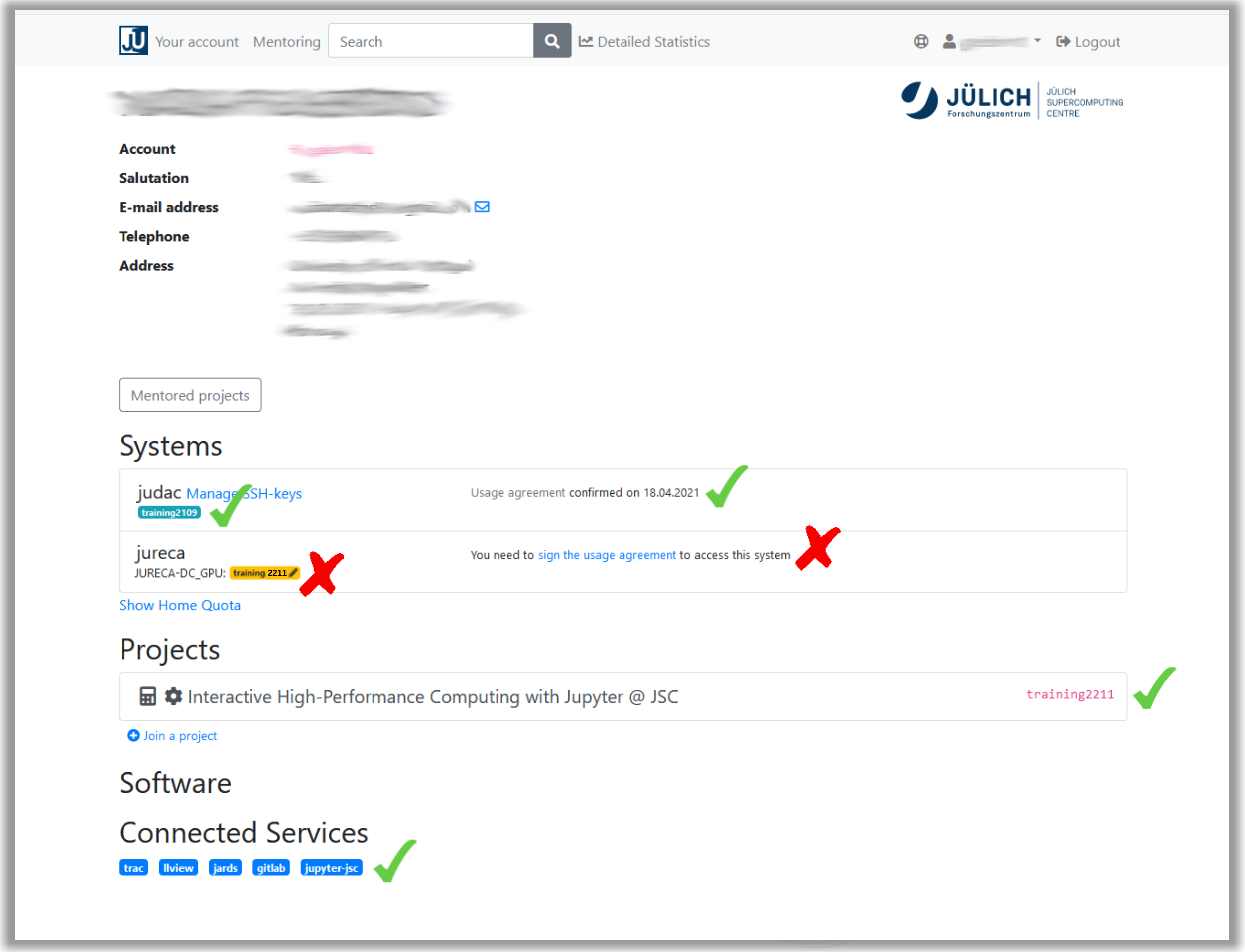
**3) Sign usage agreement**

✓ Wait for creation of HPC accounts

✓ Update of the SLURM DB

**4) Check Connected Services:**

✓ jupyter-jsc



JU Your account Mentoring Search Detailed Statistics [redacted] Logout

[redacted]

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

| | |
|---|---|
| **Account** | [redacted] |
| **Salutation** | [redacted] |
| **E-mail address** | [redacted] |
| **Telephone** | [redacted] |
| **Address** | [redacted] |

Mentored projects

## Systems

judac Manage SSH-keys training2109 ✓ — Usage agreement confirmed on 18.04.2021 ✓

jureca JURECA-DC_GPU: training 2211 ✗ — You need to sign the usage agreement to access this system ✗

Show Home Quota

## Projects

Interactive High-Performance Computing with Jupyter @ JSC — training2211 ✓

Join a project

## Software

## Connected Services

trac llview jards gitlab jupyter-jsc ✓

For more details, please visit
https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc/-/blob/main/README.md

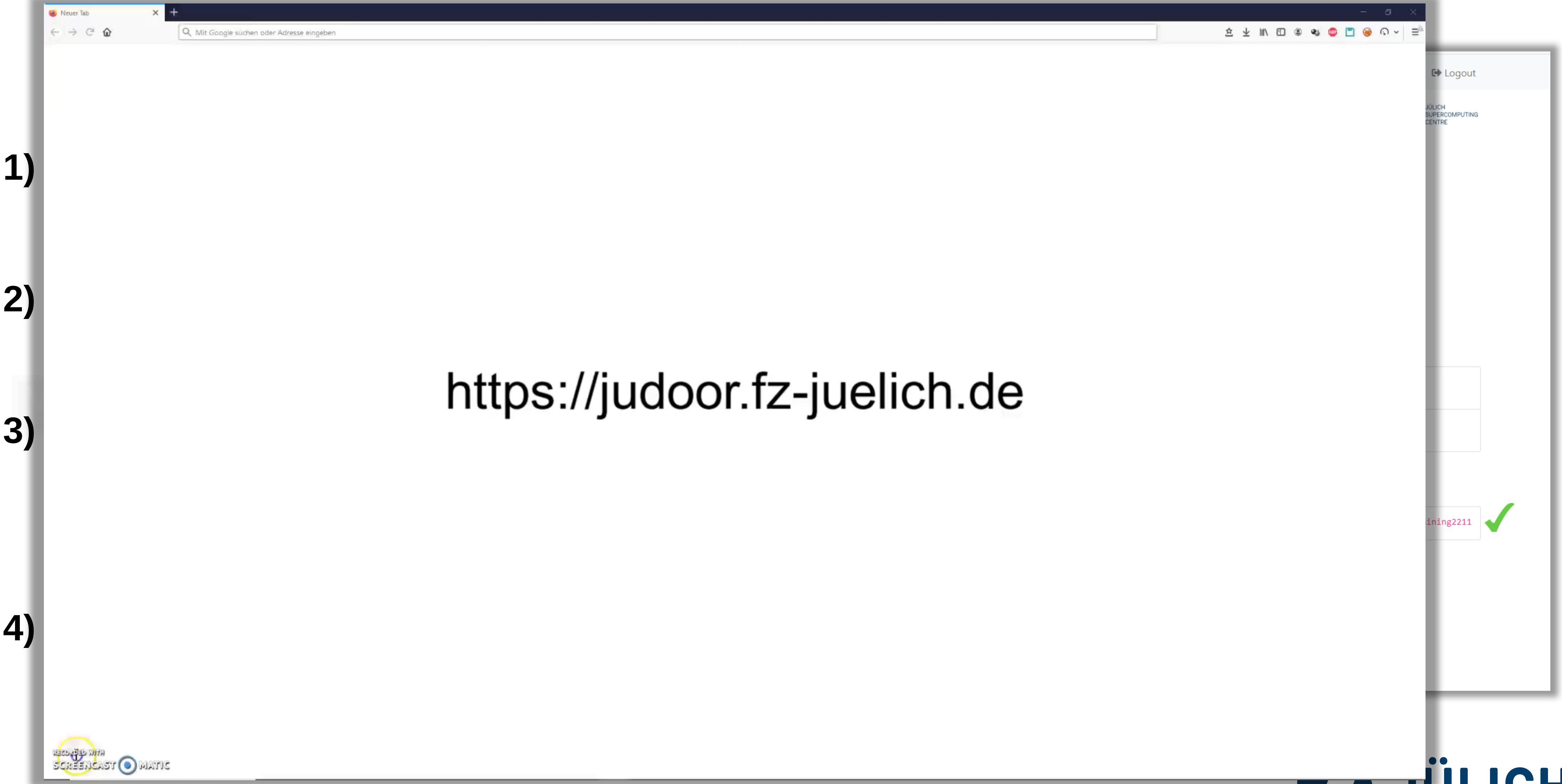
# PRE-ACCESS TODOS

1)

2)

3)

4)



Neuer Tab
Mit Google suchen oder Adresse eingeben
Logout
JÜLICH SUPERCOMPUTING CENTRE
ining2211
✓
RECORDED WITH SCREENCAST-O-MATIC

https://judoor.fz-juelich.de

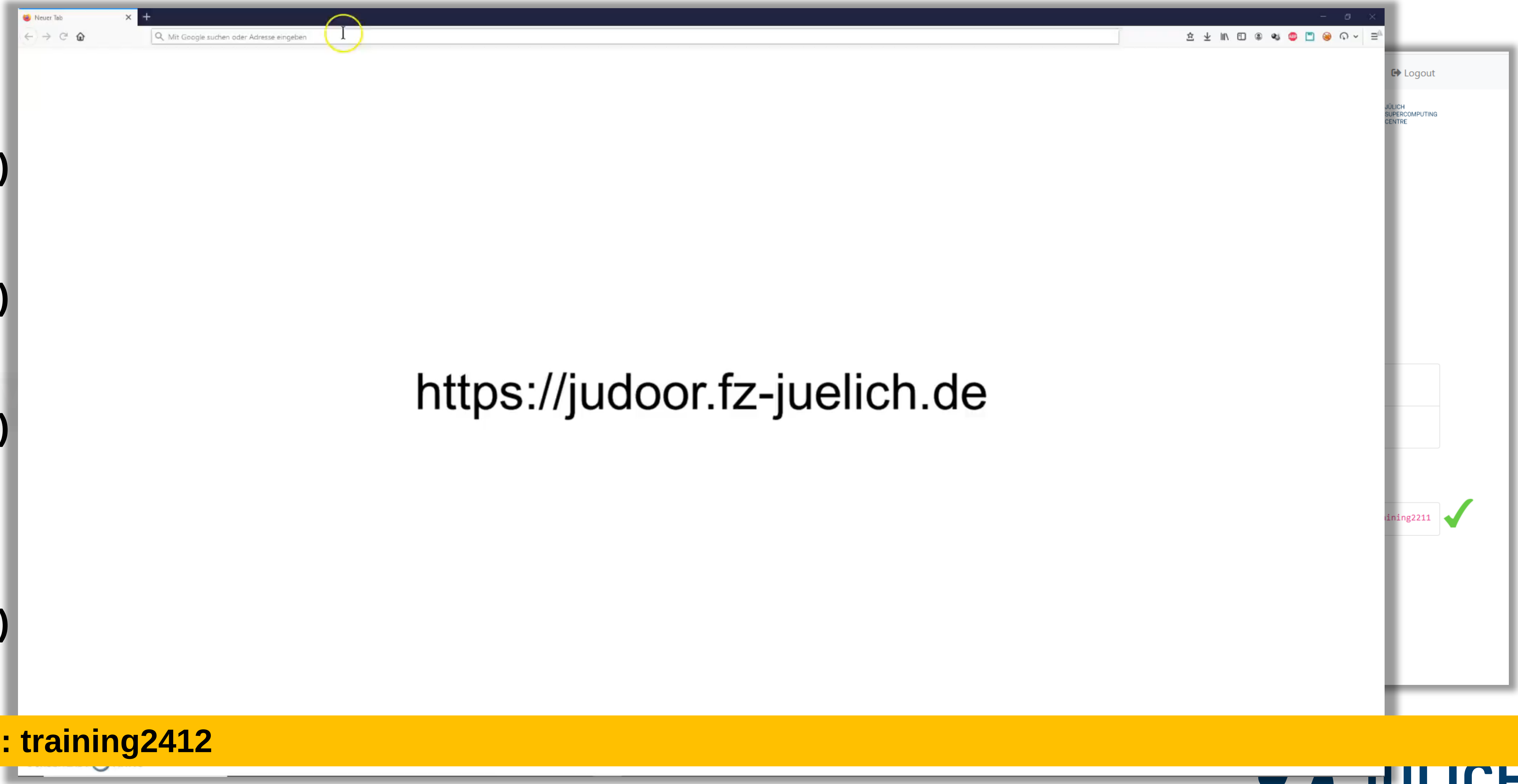
# PRE-ACCESS TODOS

1)

2)

3)

4)


Neuer Tab
Mit Google suchen oder Adresse eingeben
Logout
JÜLICH SUPERCOMPUTING CENTRE
ining2211

https://judoor.fz-juelich.de

**Project id: training2412**

# PRE-ACCESS TODOS

**1) Register & Login**

- ✓ https://judoor.fz-juelich.de

**2) Join** the project „training2412“

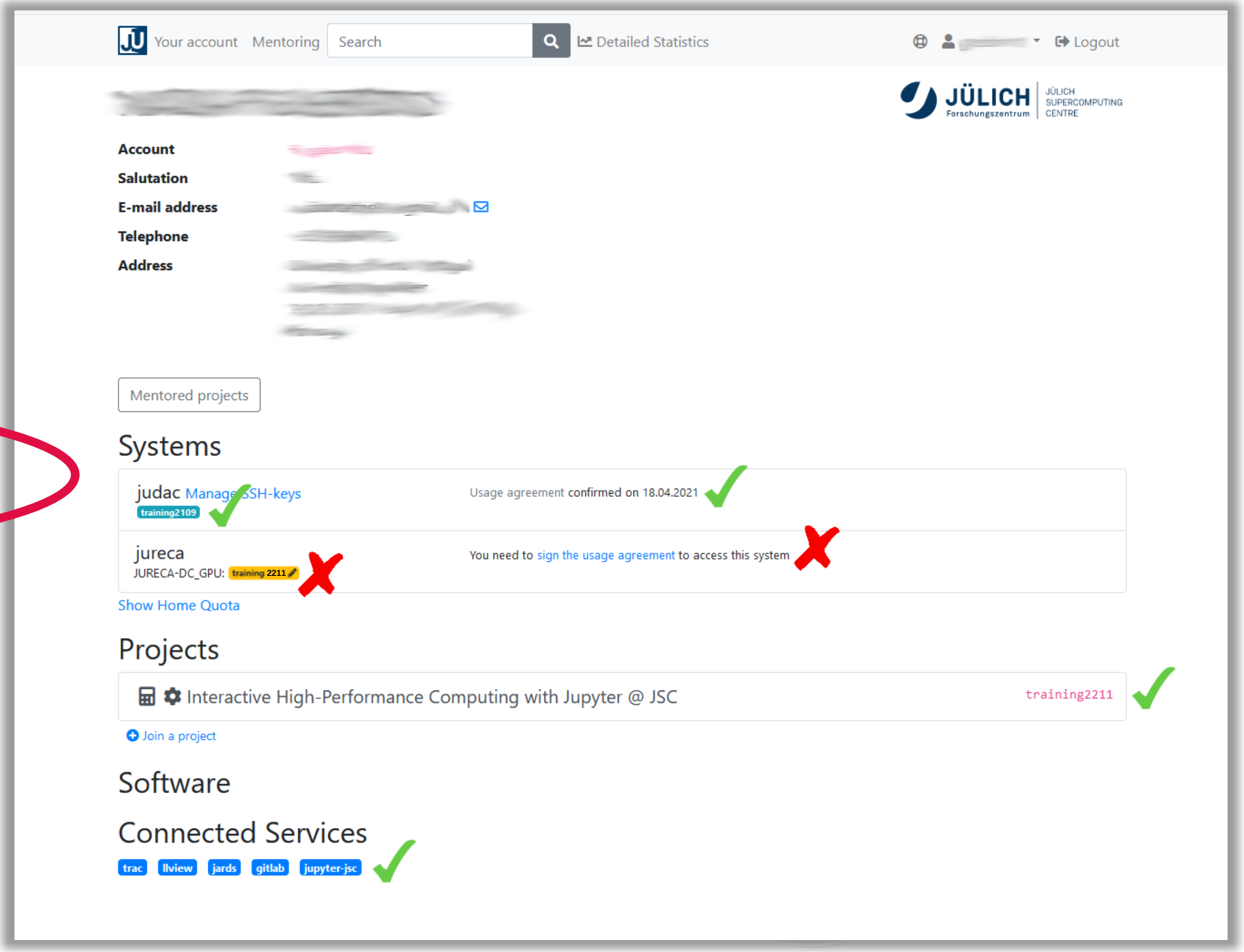
- ✓ Wait to get joined by the project PI

**3) Sign usage agreement**

- ✓ Wait for creation of HPC accounts
- ✓ Update of the SLURM DB

**4) Check Connected Services:**

- ✓ jupyter-jsc



Your account   Mentoring   Search   Detailed Statistics   [redacted]   Logout

[redacted]

**Account** [redacted]
**Salutation** [redacted]
**E-mail address** [redacted]
**Telephone** [redacted]
**Address** [redacted]

Mentored projects

Systems

judac Manage SSH-keys
training2109
Usage agreement confirmed on 18.04.2021 ✓

jureca
JURECA-DC_GPU: training 2211
You need to sign the usage agreement to access this system ✗

Show Home Quota

Projects

Interactive High-Performance Computing with Jupyter @ JSC   training2211 ✓

Join a project

Software

Connected Services

trac   llview   jards   gitlab   jupyter-jsc ✓

For more details, please visit
https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc/-/blob/main/README.md

# PRE-ACCESS TODOS

**1) Register & Login**

- ✓ https://judoor.fz-juelich.de

**2) Join** the project „training2412“

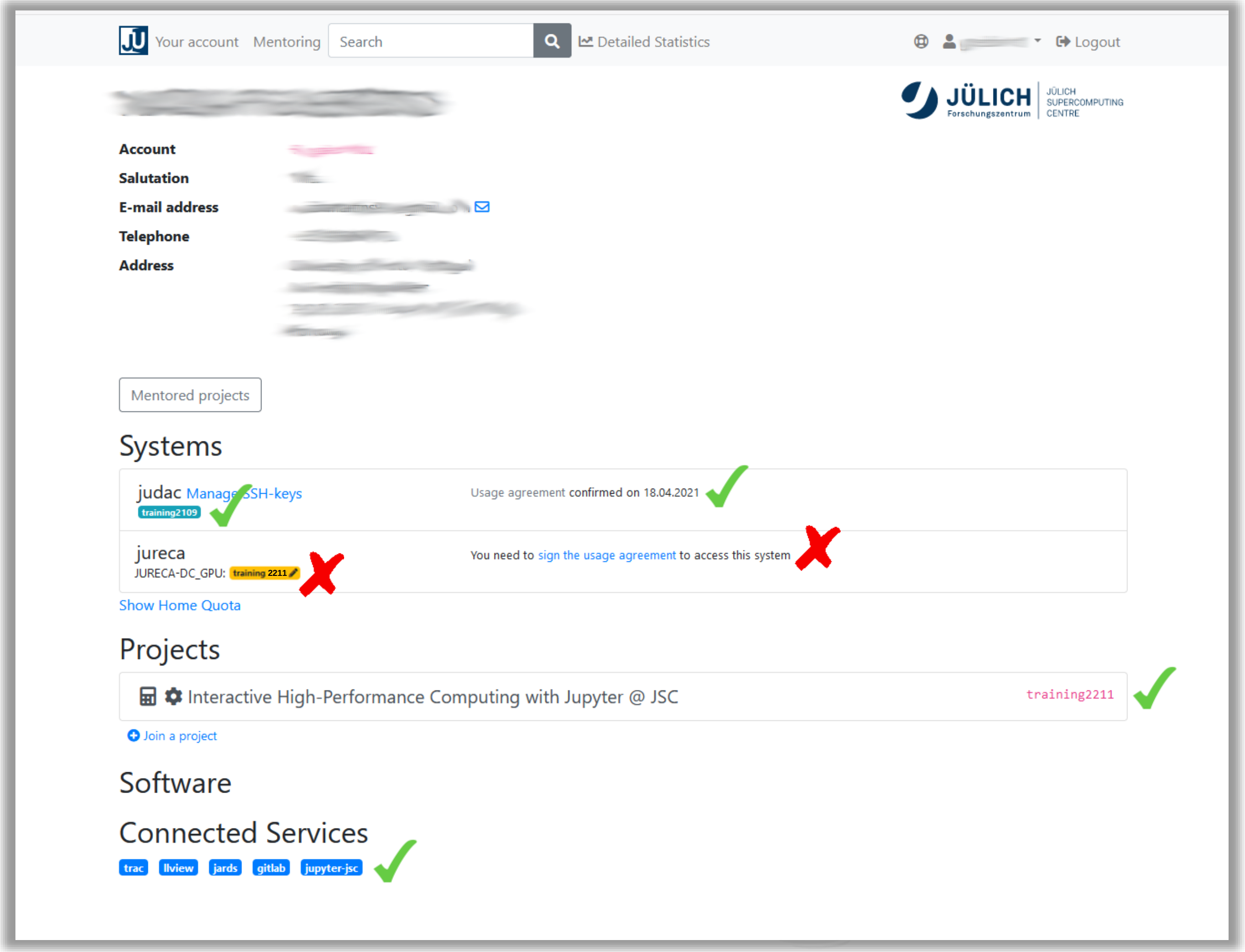
- ✓ Wait to get joined by the project PI

**3) Sign usage agreement**

- ✓ Wait for creation of HPC accounts
- ✓ Update of the SLURM DB

**4) Check Connected Services:**

- ✓ jupyter-jsc



[redacted]

**Account** [redacted]

**Salutation** [redacted]

**E-mail address** [redacted]

**Telephone** [redacted]

**Address** [redacted]

Systems

judac Manage SSH-keys training2109 ✓ — Usage agreement confirmed on 18.04.2021 ✓

jureca JURECA-DC_GPU: training 2211 ✗ — You need to sign the usage agreement to access this system ✗

Show Home Quota

Projects

Interactive High-Performance Computing with Jupyter @ JSC training2211 ✓

Join a project

Software

Connected Services

trac llview jards gitlab jupyter-jsc ✓

For more details, please visit
https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc/-/blob/main/README.md

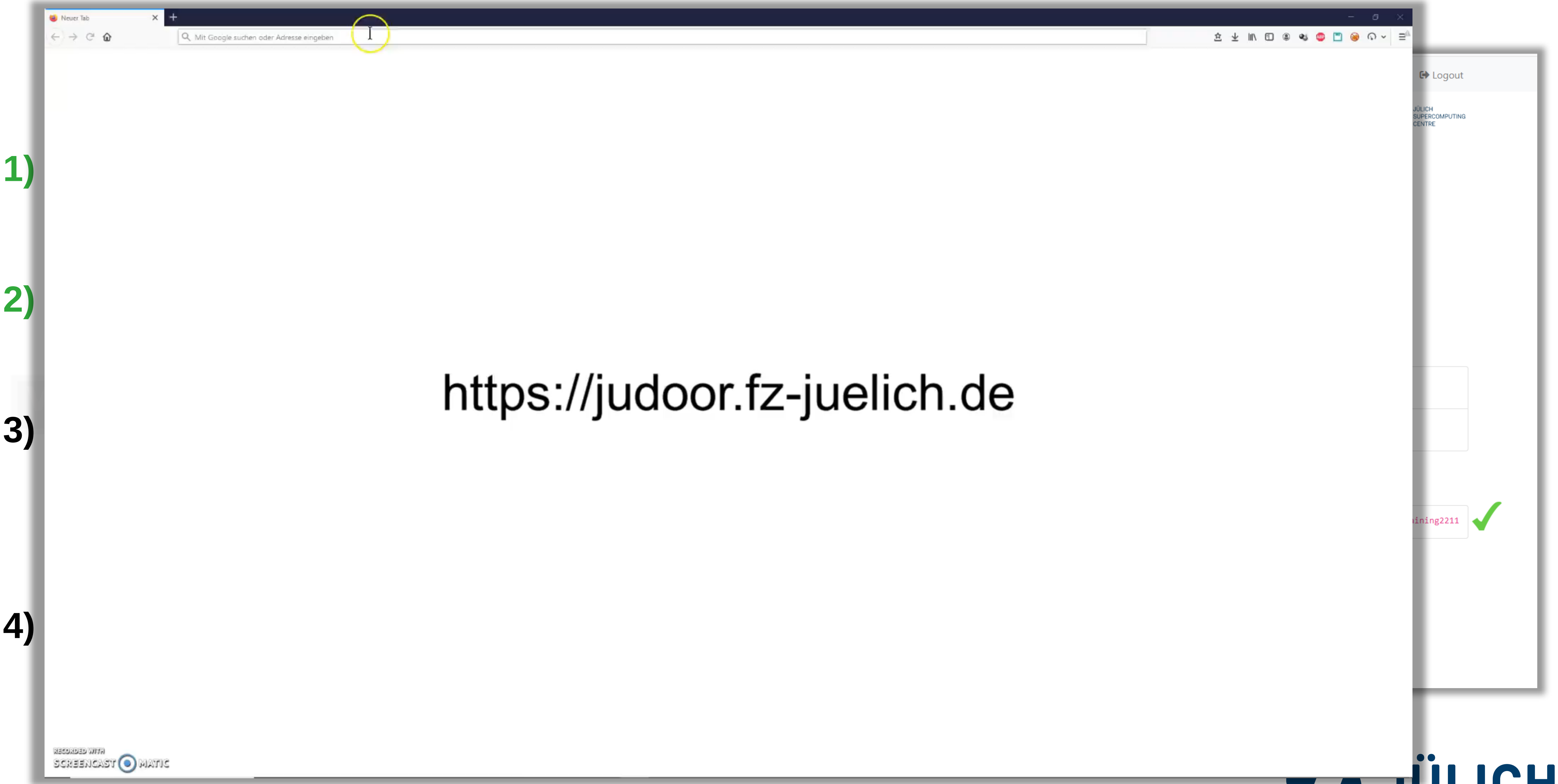
# PRE-ACCESS TODOS

1)

2)

3)

4)



https://judoor.fz-juelich.de

Neuer Tab
Mit Google suchen oder Adresse eingeben
Logout
JÜLICH SUPERCOMPUTING CENTRE
ining2211
✓
RECORDED WITH SCREENCAST-O-MATIC

# PRE-ACCESS TODOS

1) **Register & Login**

- ✓ https://judoor.fz-juelich.de

2) **Join** the project „training2412“

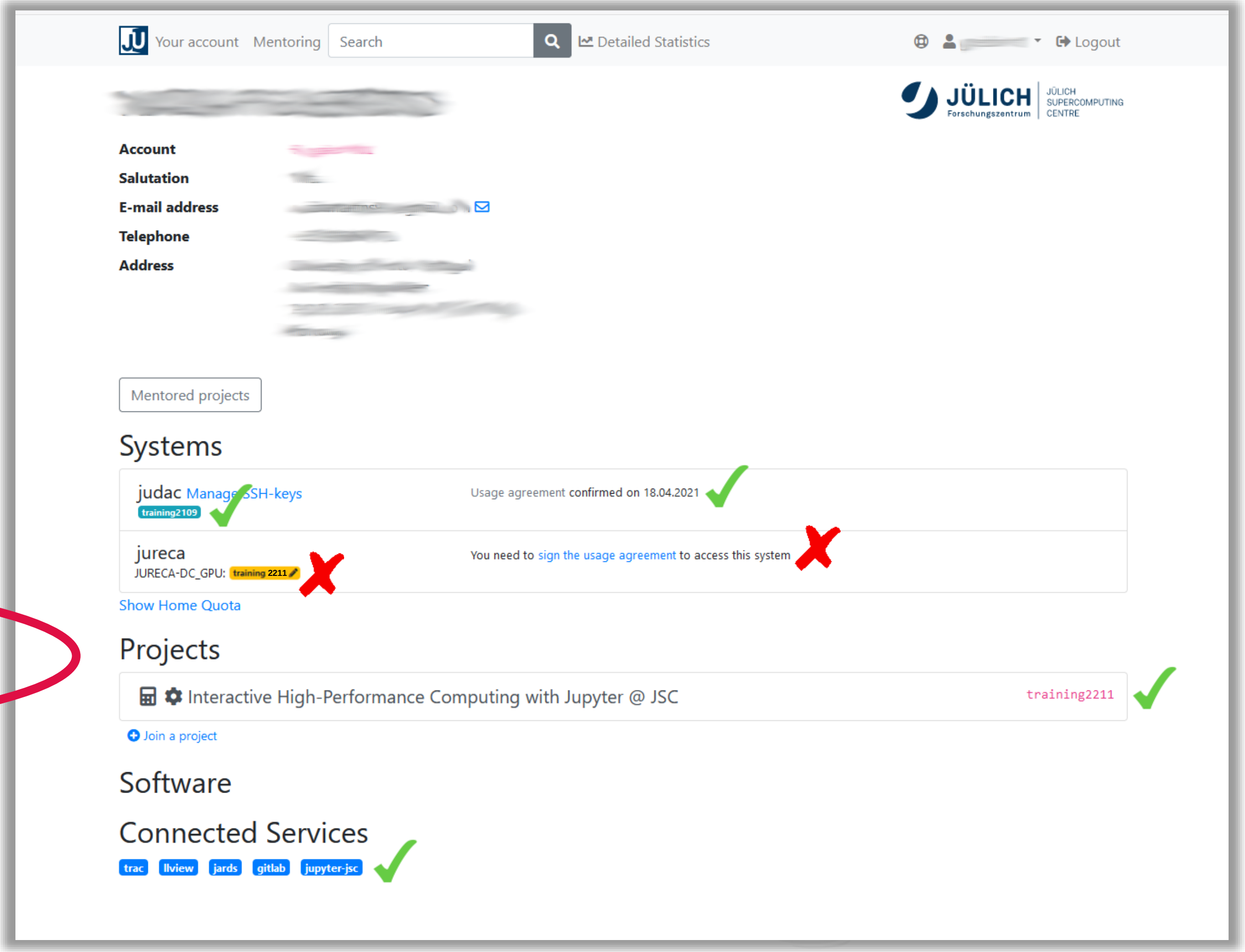
- ✓ Wait to get joined by the project PI

3) **Sign usage agreement**

- ✓ Wait for creation of HPC accounts
- ✓ Update of the SLURM DB

4) **Check Connected Services:**

- ✓ jupyter-jsc



[redacted]

Account [redacted]

Salutation [redacted]

E-mail address [redacted]

Telephone [redacted]

Address [redacted]

Mentored projects

Systems

judac Manage SSH-keys training2109 — Usage agreement confirmed on 18.04.2021 ✓

jureca JURECA-DC_GPU: training 2211 — You need to sign the usage agreement to access this system ✗

Show Home Quota

Projects

Interactive High-Performance Computing with Jupyter @ JSC training2211 ✓

Join a project

Software

Connected Services ✓

trac llview jards gitlab jupyter-jsc

For more details, please visit
https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc/-/blob/main/README.md

# MOTIVATION

# MOTIVATION

## your thinking, your reasoning, your insides, your ideas

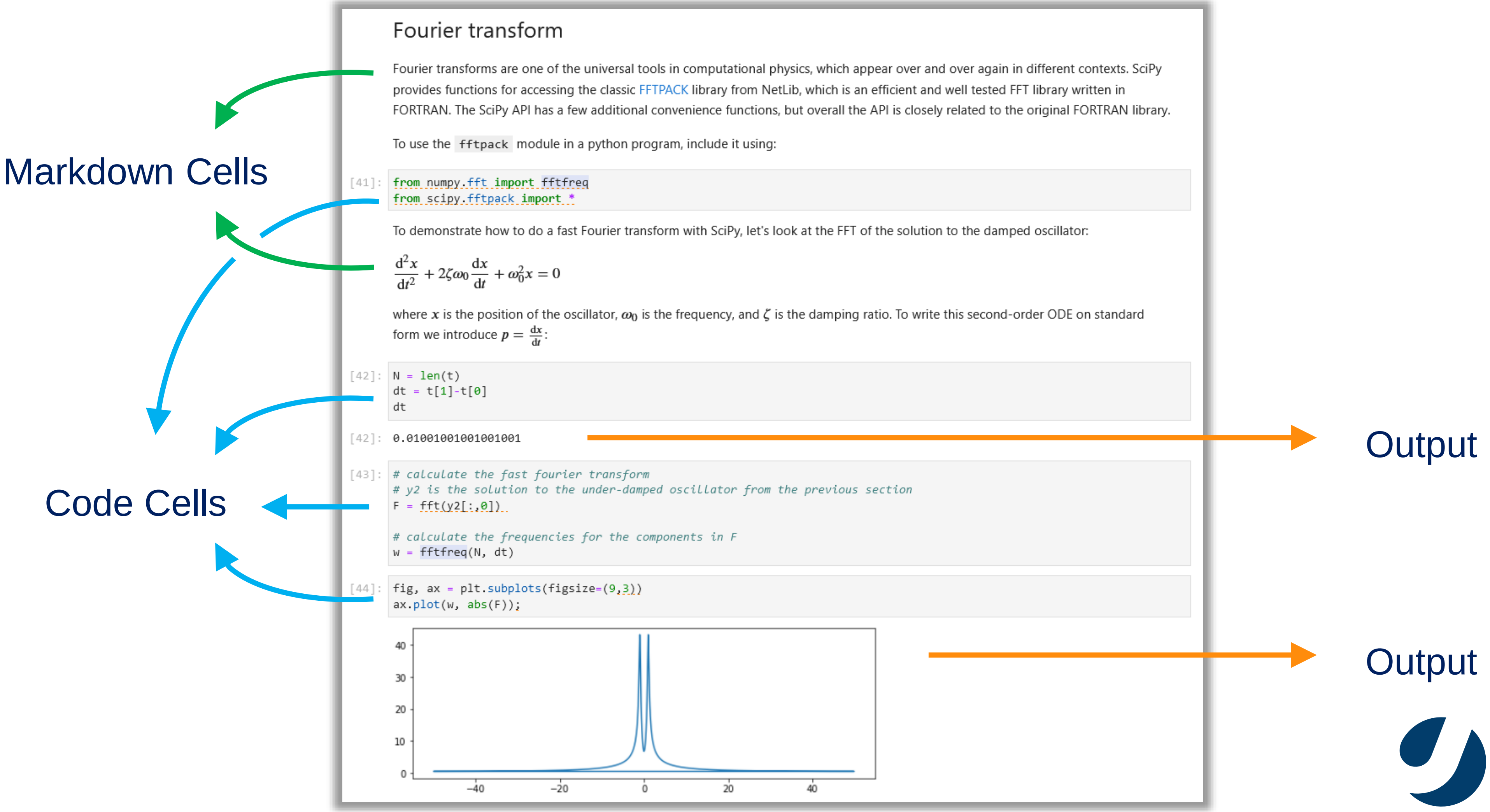
“It is all about using and building a machinery **interface between** computational researchers and data, supercomputers, laptops, cloud **and** your thinking, your reasoning, your insides, your ideas about a problem.”

Fernando Perez, Berkely Institute for Data Science
Founder of Project Jupyter

https://jupyter.org

# JUPYTER NOTEBOOK

## creating reproducible computational narratives



Markdown Cells

Code Cells

Output

Output

### Fourier transform

Fourier transforms are one of the universal tools in computational physics, which appear over and over again in different contexts. SciPy provides functions for accessing the classic FFTPACK library from NetLib, which is an efficient and well tested FFT library written in FORTRAN. The SciPy API has a few additional convenience functions, but overall the API is closely related to the original FORTRAN library.

To use the `fftpack` module in a python program, include it using:

```
[41]: from numpy.fft import fftfreq
      from scipy.fftpack import *
```

To demonstrate how to do a fast Fourier transform with SciPy, let's look at the FFT of the solution to the damped oscillator:

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\zeta\omega_0\frac{\mathrm{d}x}{\mathrm{d}t} + \omega_0^2 x = 0$$

where $x$ is the position of the oscillator, $\omega_0$ is the frequency, and $\zeta$ is the damping ratio. To write this second-order ODE on standard form we introduce $p = \frac{\mathrm{d}x}{\mathrm{d}t}$:

```
[42]: N = len(t)
      dt = t[1]-t[0]
      dt
```

```
[42]: 0.01001001001001001
```

```
[43]: # calculate the fast fourier transform
      # y2 is the solution to the under-damped oscillator from the previous section
      F = fft(y2[:,0])

      # calculate the frequencies for the components in F
      w = fftfreq(N, dt)
```

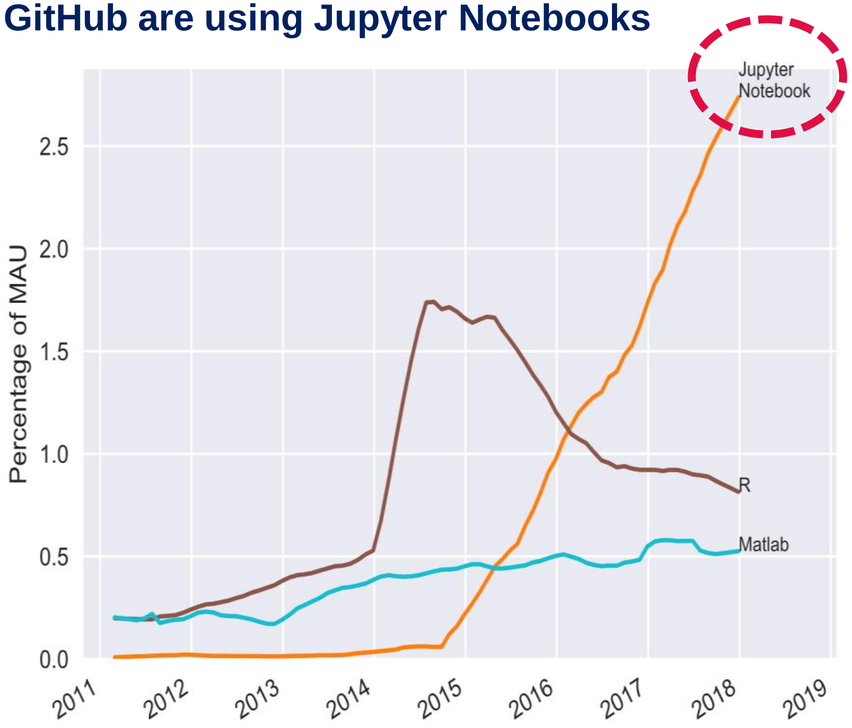
```
[44]: fig, ax = plt.subplots(figsize=(9,3))
      ax.plot(w, abs(F));
```

# MOTIVATION

## Rise of Jupyter´s popularity

- In 2007, Fernando Pérez and Brian Granger announced
  „**Ipython**: a system for interactive scientific computing“ [1]
- In 2014, Fernando Pérez announced
  a spin-off project from IPython called **Project Jupyter**.
  - IPython continued to exist as a Python shell and a kernel for Jupyter, while the Jupyter notebook moved under the Jupyter name.
- In 2015, GitHub and the Jupyter Project announced
  native rendering of Jupyter notebooks file format (.ipynb files) on the **GitHub**
- In 2017, the **first JupyterCon** was organized by O‘Reilly in New York City.
  Fernando Pérez opened the conference with an inspiring talk. [2]
- In 2018, **JupyterLab** was announced
  as the next-generation web-based interface for Project Jupyter.
- In 2019, JupyterLab 1.0 …
  In 2020, JupyterLab 2.0 …
  In 2021, JupyterLab 3.0 …
  In 2023, JupyterLab 4.0 …

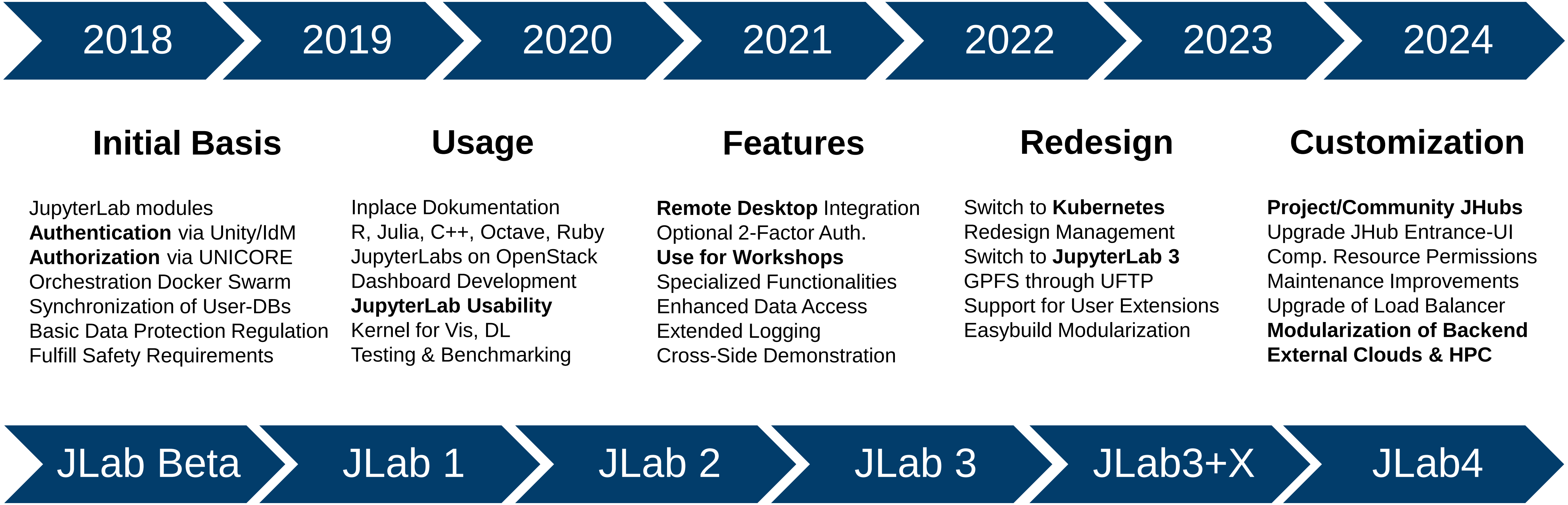
**Counting how many Monthly Active Users (MAU) on GitHub are using Jupyter Notebooks**



https://www.benfrederickson.com/ranking-programming-languages-by-github-users/
https://github.com/benfred/github-analysis

[1] Pérez F, Granger BE (2007) Ipython: a system for interactive scientific computing. Comput Sci Eng 9(3):21–29
[2] Pérez F, Project Jupyter: From interactive Python to open science -> https://www.youtube.com/watch?v=xuNj5paMuow

# HISTORY OF JUPYTERLAB AT JSC



2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024

### Initial Basis

JupyterLab modules
**Authentication** via Unity/IdM
**Authorization** via UNICORE
Orchestration Docker Swarm
Synchronization of User-DBs
Basic Data Protection Regulation
Fulfill Safety Requirements

### Usage

Inplace Dokumentation
R, Julia, C++, Octave, Ruby
JupyterLabs on OpenStack
Dashboard Development
**JupyterLab Usability**
Kernel for Vis, DL
Testing & Benchmarking

### Features

**Remote Desktop** Integration
Optional 2-Factor Auth.
**Use for Workshops**
Specialized Functionalities
Enhanced Data Access
Extended Logging
Cross-Side Demonstration

### Redesign

Switch to **Kubernetes**
Redesign Management
Switch to **JupyterLab 3**
GPFS through UFTP
Support for User Extensions
Easybuild Modularization

### Customization

**Project/Community JHubs**
Upgrade JHub Entrance-UI
Comp. Resource Permissions
Maintenance Improvements
Upgrade of Load Balancer
**Modularization of Backend**
**External Clouds & HPC**

JLab Beta | JLab 1 | JLab 2 | JLab 3 | JLab3+X | JLab4

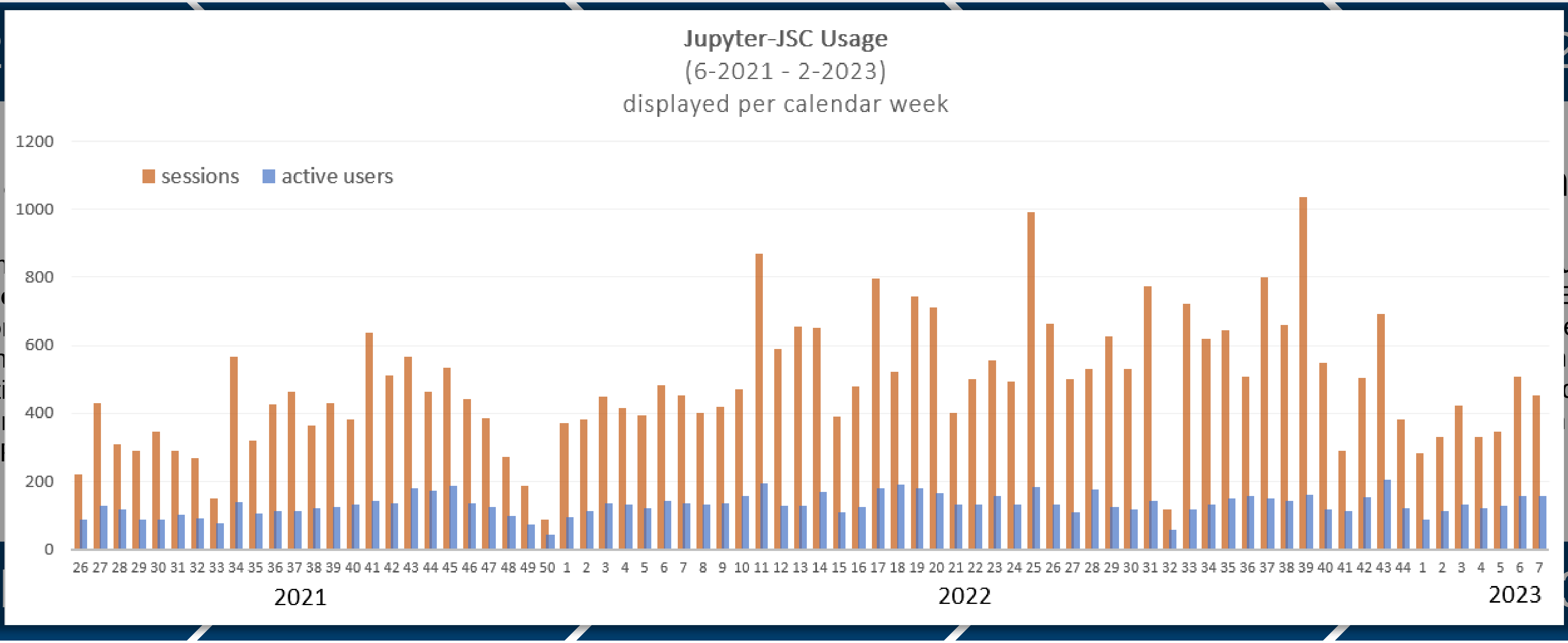
# HISTORY OF JUPYTERLAB AT JSC

2

**Initi**

JupyterLab m
**Authenticati**
**Authorizatio**
Orchestration
Synchronizati
Basic Data P
Fulfill Safety I

JLa


Jupyter-JSC Usage
(6-2021 - 2-2023)
displayed per calendar week

22

**ization**

**unity JHubs**
Entrance-UI
e Permissions
provements
d Balancer
**of Backend**

3+X

# TERMINOLOGY

# TERMINOLOGY

## What is JupyterLab

**JupyterLab**

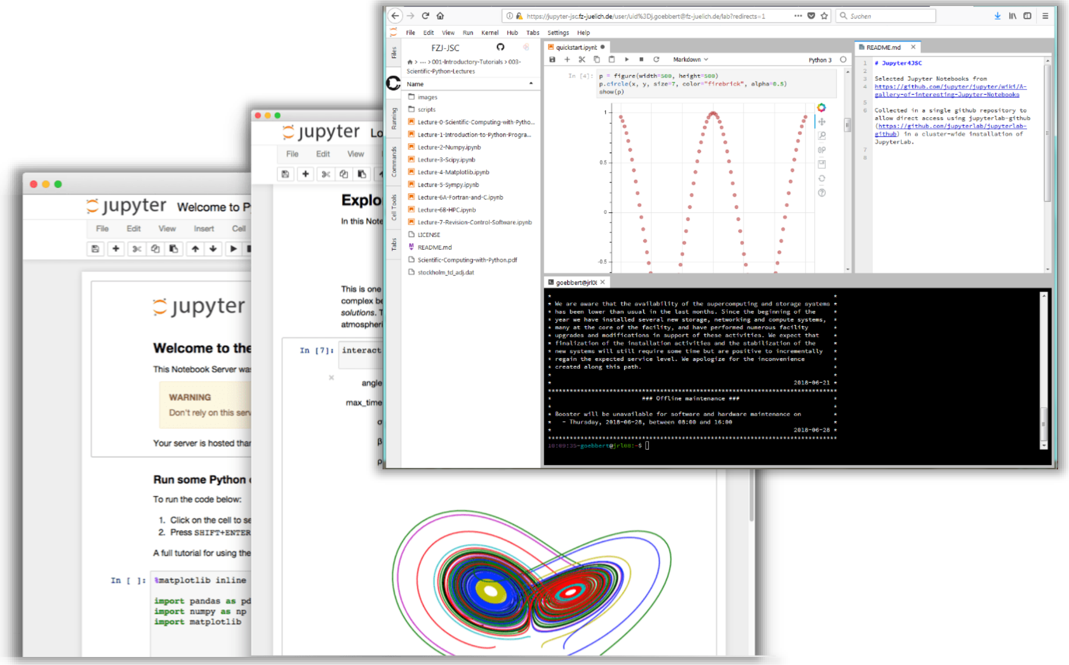
- **Interactive** working environment in the web browser
- For the creation of **reproducible** computer-aided narratives
- Very **popular** with researchers from all fields
- Jupyter = Julia + Python + R

Multi-purpose working environment

- Language agnostic
- Supports execution environments (“*kernels*”)
  - For dozens of languages: Python, R, Julia, C++, ...
- Extensible software design („*extensions*“)
  - many server/client plug-ins available
  - Eg. in-browser-terminal and file-browsing

Document-Centered Computing (“*notebooks*”)

- Combines code execution,
  rich text, math, plots and rich media.
- All-in-one document called Jupyter Notebook



jupyter Welcome to P
File Edit View Insert Cell
jupyter
jupyter Lo
File Edit View
Explo
https://jupyter-jsc.fz-juelich.de/user/uid%3Dj.goebbert@fz-juelich.de/lab?redirects=1
Suchen
File Edit View Run Kernel Hub Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
FZJ-JSC
… > 001-Introductory-Tutorials > 003-Scientific-Python-Lectures
Name
images
scripts
Lecture-0-Scientific-Computing-with-Pytho...
Lecture-1-Introduction-to-Python-Progra...
Lecture-2-Numpy.ipynb
Lecture-3-Scipy.ipynb
Lecture-4-Matplotlib.ipynb
Lecture-5-Sympy.ipynb
Lecture-6A-Fortran-and-C.ipynb
Lecture-6B-HPC.ipynb
Lecture-7-Revision-Control-Software.ipynb
LICENSE
README.md
Scientific-Computing-with-Python.pdf
stockholm_td_adj.dat
quickstart.ipynb
Markdown
Python 3
1
0.5
0
-0.5
README.md
goebbert@jrl0
angle
max_time
σ
β
ρ

Welcome to the

This Notebook Server wa

WARNING

Don't rely on this serv

Your server is hosted tha

Run some Python

To run the code below:

1. Click on the cell to se
2. Press SHIFT+ENTER

A full tutorial for using the

```
In [ ]: %matplotlib inline

        import pandas as pd
        import numpy as np
        import matplotlib
```

In this Note

This is one
complex be
*solutions*. T
atmospheri

```
In [7]: interact
```

```
In [4]: p = figure(width=500, height=500)
        p.circle(x, y, size=7, color="firebrick", alpha=0.5)
        show(p)
```

```
# Jupyter4JSC

Selected Jupyter Notebooks from
https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks

Collected in a single github repository to
allow direct access using jupyterlab-github
(https://github.com/jupyterlab/jupyterlab-github) in a cluster-wide installation of
JupyterLab.
```

```
*                                                                      *
* We are aware that the availability of the supercomputing and storage systems *
* has been lower than usual in the last months. Since the beginning of the    *
* year we have installed several new storage, networking and compute systems, *
* many at the core of the facility, and have performed numerous facility      *
* upgrades and modifications in support of these activities. We expect that   *
* finalization of the installation activities and the stabilization of the   *
* new systems will still require some time but are positive to incrementally  *
* regain the expected service level. We apologize for the inconvenience       *
* created along this path.                                                    *
*                                                                              *
*                                                                   2018-06-21 *
********************************************************************************
*                         ### Offline maintenance ###                          *
*                                                                              *
* Booster will be unavailable for software and hardware maintenance on        *
*   - Thursday, 2018-06-28, between 08:00 and 16:00                            *
*                                                                   2018-06-28 *
********************************************************************************
10:09:35-goebbert@jrl08:~$ 
```

https://jupyterlab.readthedocs.io

# TERMINOLOGY

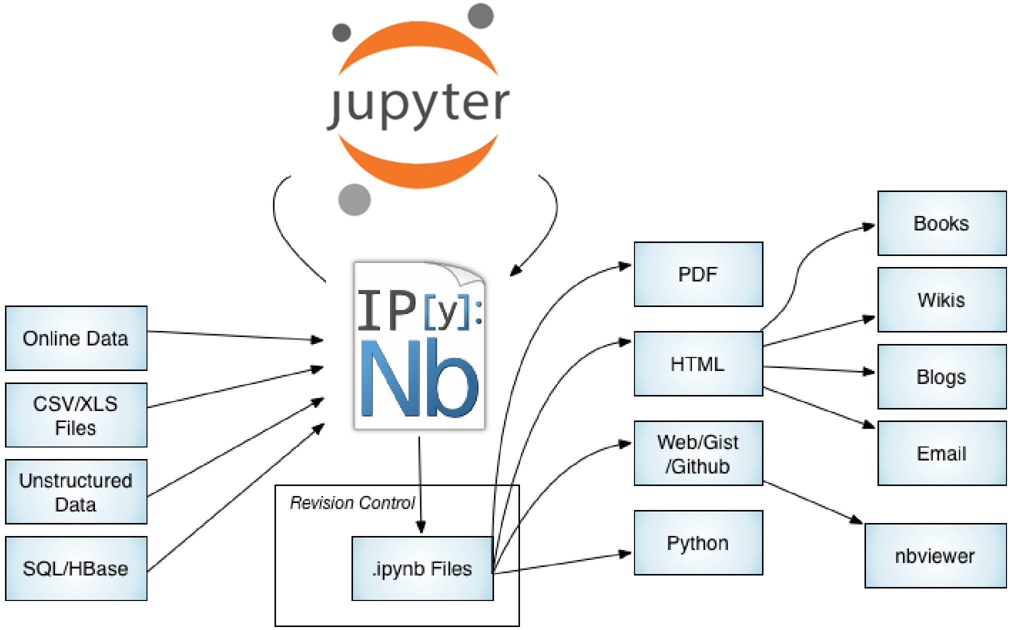
## What is a Jupyter Notebook?

**Jupyter Notebook**
A notebook document (file extension .ipynb)
is a document that can be rendered in a web browser

- It is a file, which stores your work in JSON format
- Based on a set of open standards for interactive computing

- Allows development of custom applications with embedded interactive computing.
- Can be extended by third parties

- Directly convertible to PDF, HTML, LateX ...
- Supported by many applications
such as GitHub, GitLab, etc..



https://jupyter-notebook.readthedocs.io/
https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks

# TERMINOLOGY

## What is a Jupyter Kernel?

**Jupyter Kernel**
A “kernel” refers to the separate process
which executes code cells within a Jupyter notebook.

Jupyter Kernel

- **run code** in different programming languages and environments.
- can be **connected to** a notebook (one at a time).
- **communicates** via ZeroMQ with the JupyterLab.

- Multiple **preinstalled** Jupyter Kernels can be found on our clusters
  - Python, R, Julia, Bash, C++, Ruby, JavaScript
  - Specialized kernels for visualization, quantum-computing

- You can easily **create your own kernel** which for example runs your specialized virtual Python environment.

https://jupyter-notebook.readthedocs.io/
https://github.com/jupyter/jupyter/wiki/Jupyter-kernels
https://zeromq.org

# TERMINOLOGY

## What is a JupyterLab Extension?

**JupyterLab Extension**

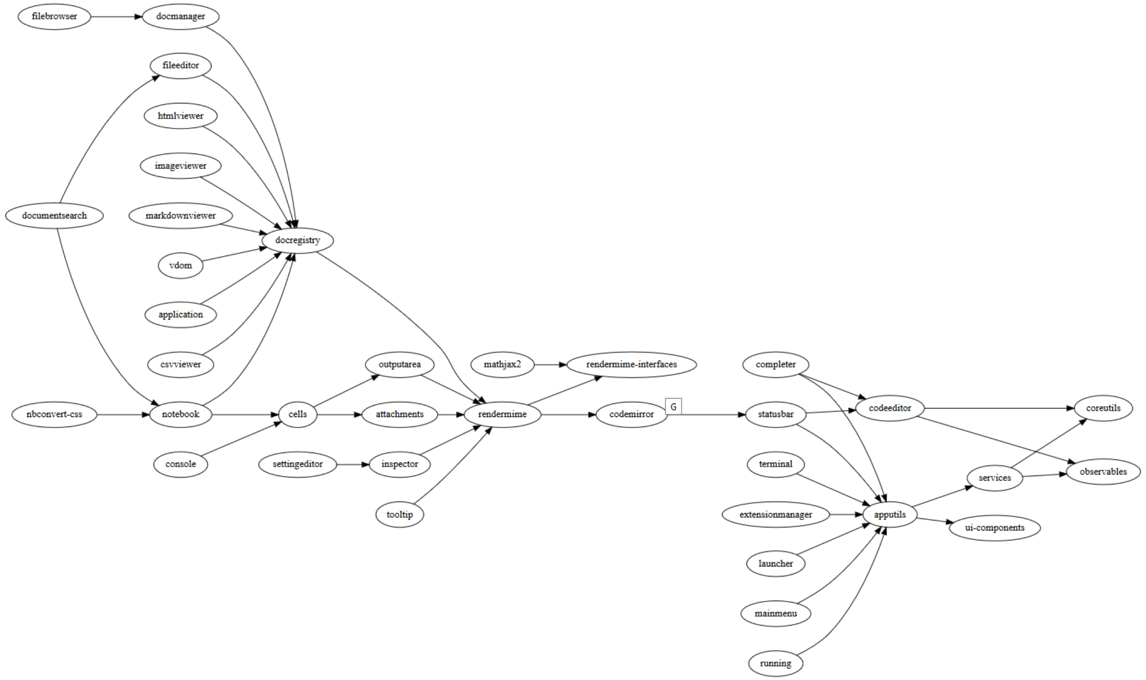
JupyterLab extensions can customize or enhance any part of JupyterLab.

JupyterLab Extensions

- provide new file viewers, editors, themes
- provide renderers for rich outputs in notebooks
- add items to the menu or command palette
- add keyboard shortcuts
- add settings in the settings system.

- Extensions can even provide an API for other extensions to use and can depend on other extensions.

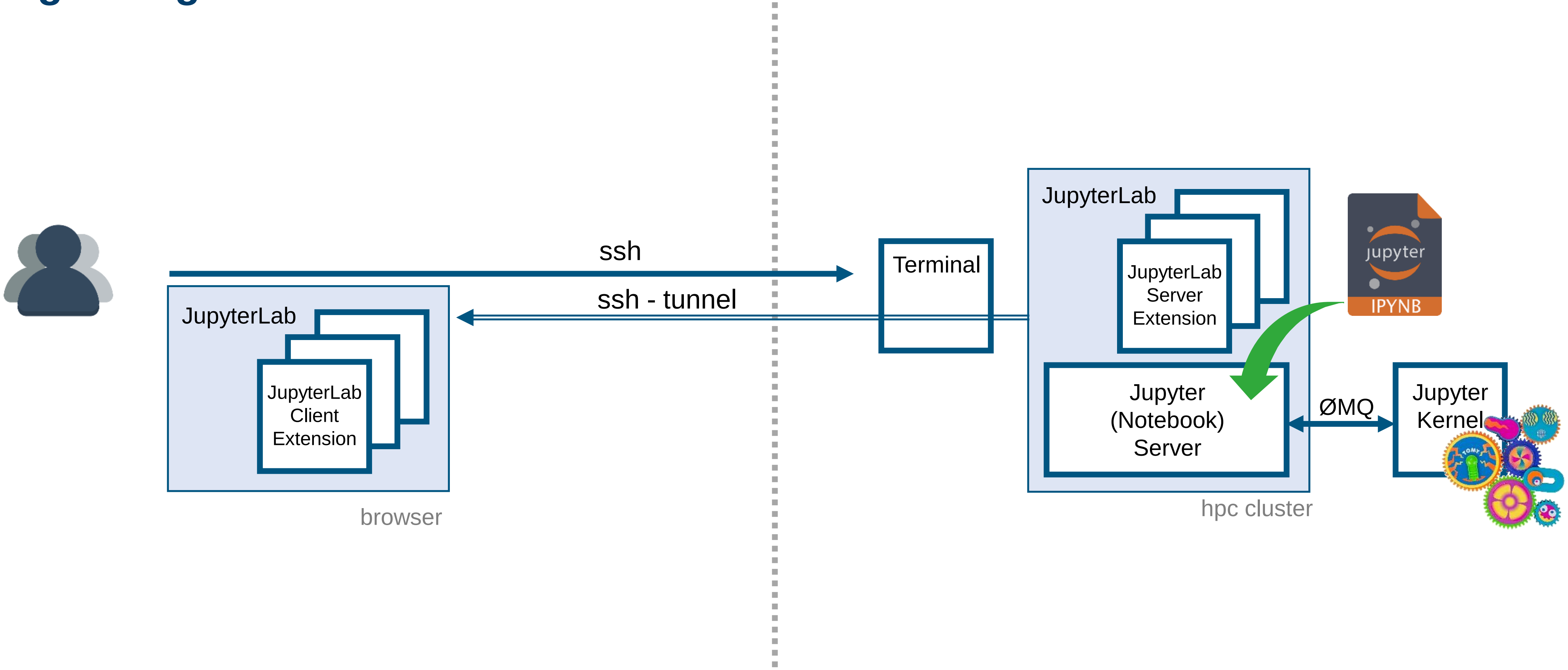
The whole JupyterLab itself is simply a **collection of extensions** that are no more powerful or privileged than any custom extension.



https://jupyterlab.readthedocs.io/en/stable/user/extensions.html
https://github.com/topics/jupyterlab-extension

# TERMINOLOGY

## Bringing all together



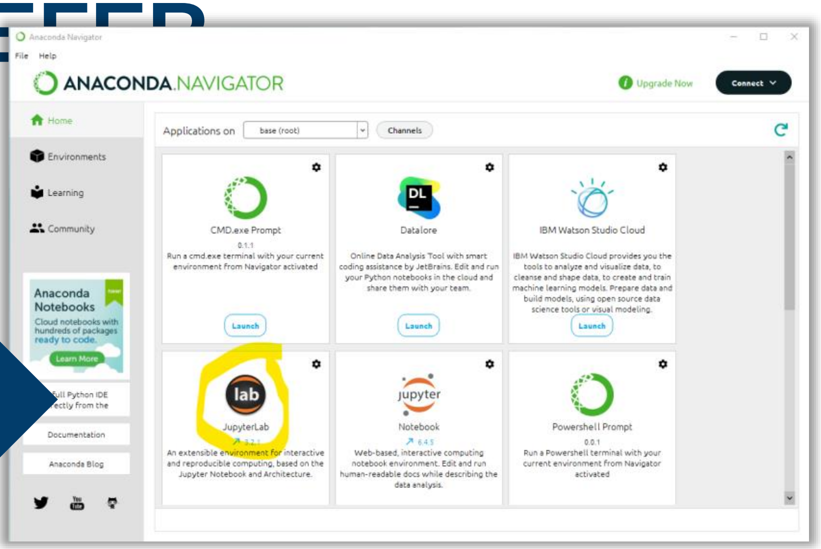
# INSTALLATION

# JUPYTERLAB - WHEREVER YOU PREFER

## Local, Remote, Browser-only

### Local installation:

- **JupyterLab** installed using conda, mamba, pip, pipenv or docker.
  ➔ https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html



Cloud notebooks with hundreds of packages ready to code.

Full Python IDE
rectly from the

Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

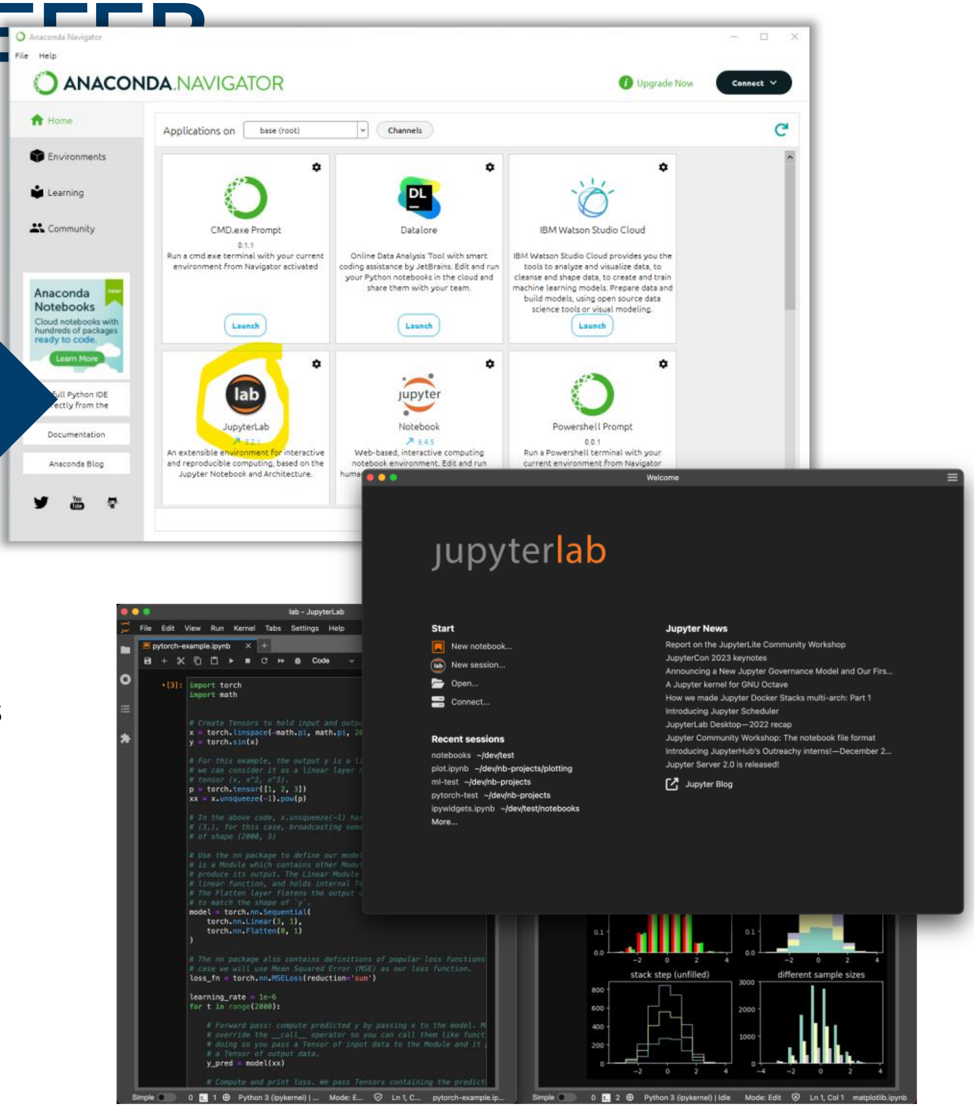
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Run a Powershell terminal with your current environment from Navigator activated

# JUPYTERLAB - WHEREVER YOU PREFER

## Local, Remote, Browser-only



Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run huma

Run a Powershell terminal with your current environment from Navigator

```
[3]: import torch
     import math

     # Create Tensors to hold input and outpu
     x = torch.linspace(-math.pi, math.pi, 20
     y = torch.sin(x)

     # For this example, the output y is a li
     # we can consider it as a linear layer n
     # tensor (x, x^2, x^3).
     p = torch.tensor([1, 2, 3])
     xx = x.unsqueeze(-1).pow(p)

     # In the above code, x.unsqueeze(-1) has
     # (3,), for this case, broadcasting sema
     # of shape (2000, 3)

     # Use the nn package to define our model
     # is a Module which contains other Modul
     # produce its output. The Linear Module
     # linear function, and holds internal Te
     # The Flatten layer flatens the output o
     # to match the shape of `y`.
     model = torch.nn.Sequential(
         torch.nn.Linear(3, 1),
         torch.nn.Flatten(0, 1)
     )

     # The nn package also contains definitions of popular loss functions
     # case we will use Mean Squared Error (MSE) as our loss function.
     loss_fn = torch.nn.MSELoss(reduction='sum')

     learning_rate = 1e-6
     for t in range(2000):

         # Forward pass: compute predicted y by passing x to the model. M
         # override the __call__ operator so you can call them like funct
         # doing so you pass a Tensor of input data to the Module and it
         # a Tensor of output data.
         y_pred = model(xx)

         # Compute and print loss. We pass Tensors containing the predict
```

Recent sessions:

```
notebooks         ~/dev/test
plot.ipynb        ~/dev/nb-projects/plotting
ml-test           ~/dev/nb-projects
pytorch-test      ~/dev/nb-projects
ipywidgets.ipynb  ~/dev/test/notebooks
More...
```

Jupyter News:

- Report on the JupyterLite Community Workshop
- JupyterCon 2023 keynotes
- Announcing a New Jupyter Governance Model and Our Firs...
- A Jupyter kernel for GNU Octave
- How we made Jupyter Docker Stacks multi-arch: Part 1
- Introducing Jupyter Scheduler
- JupyterLab Desktop—2022 recap
- Jupyter Community Workshop: The notebook file format
- Introducing JupyterHub's Outreachy interns!—December 2...
- Jupyter Server 2.0 is released!

### Local installation:

- **JupyterLab** installed using conda, mamba, pip, pipenv or docker.
  ➔ https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html
- **JupyterLab** installed as normal desktop application **= JupyterLab Desktop**
  ➔ https://github.com/jupyterlab/jupyterlab-desktop/releases

**JupyterLab Desktop** is the cross-platform desktop application for JupyterLab. It is probably the quickest and easiest way to get started with Jupyter notebooks on your personal computer, with the flexibility for advanced use cases.

(Windows, macOS, Debian/Ubuntu, RedHat/Fedora)

# JUPYTERLAB - WHEREVER YOU PREFER

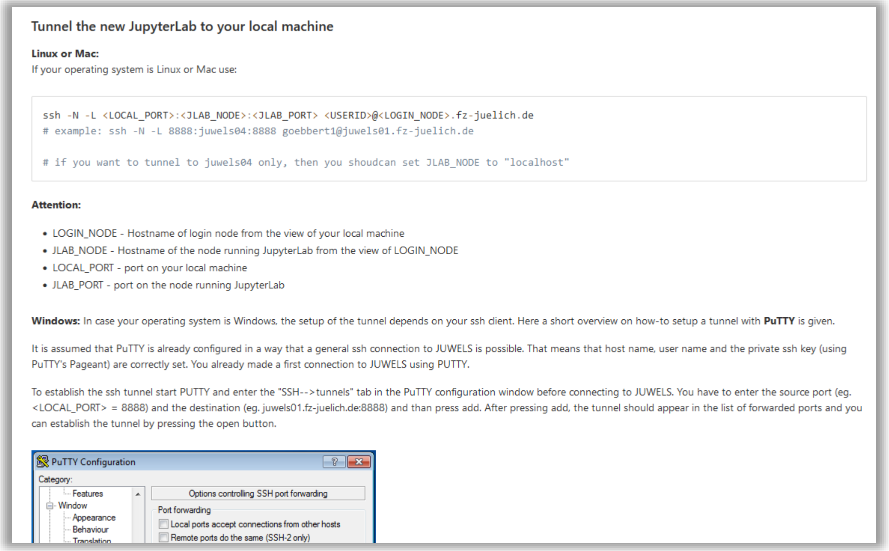
## Local, Remote, Browser-only

### Local installation:

- **JupyterLab** installed using conda, mamba, pip, pipenv or docker.
  ➔ https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html
- **JupyterLab** installed as normal desktop application **= JupyterLab Desktop**
  ➔ https://github.com/jupyterlab/jupyterlab-desktop/releases

### Remote (cluster) installation:

- **JupyterLab** installed on a remote server and accessed through the browser
  - in $HOME (e.g. using pip or miniconda)
  - system-wide (e.g. with Easybuild, Spark) by the admins.



Tunnel the new JupyterLab to your local machine

**Linux or Mac:**
If your operating system is Linux or Mac use:

```
ssh -N -L <LOCAL_PORT>:<JLAB_NODE>:<JLAB_PORT> <USERID>@<LOGIN_NODE>.fz-juelich.de
# example: ssh -N -L 8888:juwels04:8888 goebbert1@juwels01.fz-juelich.de

# if you want to tunnel to juwels04 only, then you shoudcan set JLAB_NODE to "localhost"
```

**Attention:**

- LOGIN_NODE - Hostname of login node from the view of your local machine
- JLAB_NODE - Hostname of the node running JupyterLab from the view of LOGIN_NODE
- LOCAL_PORT - port on your local machine
- JLAB_PORT - port on the node running JupyterLab

**Windows:** In case your operating system is Windows, the setup of the tunnel depends on your ssh client. Here a short overview on how-to setup a tunnel with **PuTTY** is given.

It is assumed that PuTTY is already configured in a way that a general ssh connection to JUWELS is possible. That means that host name, user name and the private ssh key (using PuTTY's Pageant) are correctly set. You already made a first connection to JUWELS using PUTTY.

To establish the ssh tunnel start PUTTY and enter the "SSH-->tunnels" tab in the PuTTY configuration window before connecting to JUWELS. You have to enter the source port (eg. <LOCAL_PORT> = 8888) and the destination (eg. juwels01.fz-juelich.de:8888) and than press add. After pressing add, the tunnel should appear in the list of forwarded ports and you can establish the tunnel by pressing the open button.

□ Local ports accept connections from other hosts
□ Remote ports do the same (SSH-2 only)

# JUPYTERLAB - WHEREVER YOU PREFER

## Local, Remote, Browser-only

### Local installation:

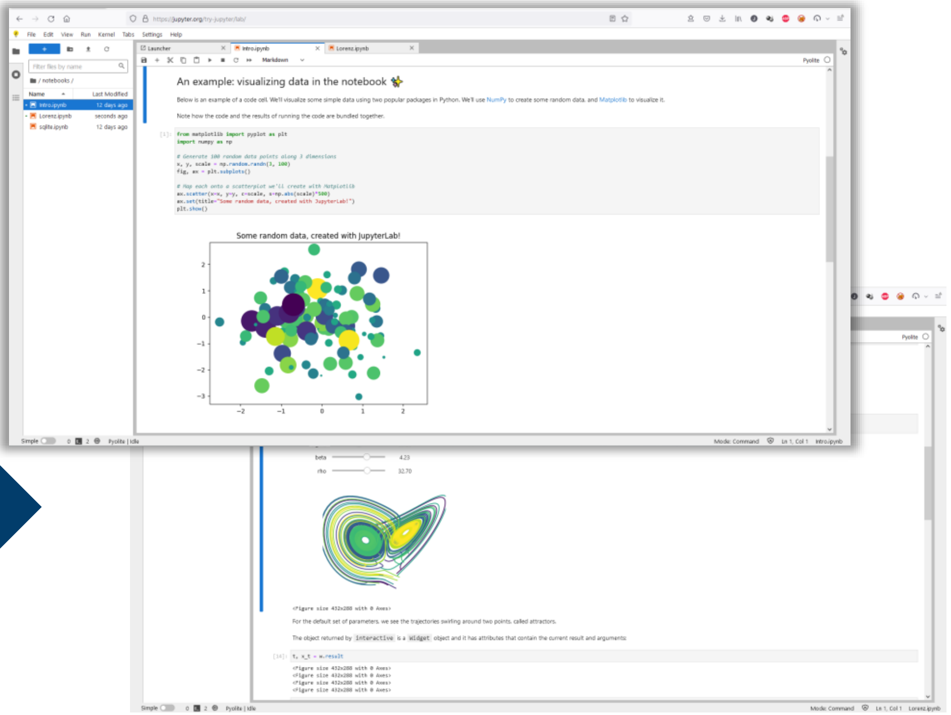
- **JupyterLab** installed using conda, mamba, pip, pipenv or docker.
  ➔ https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html
- **JupyterLab** installed as normal desktop application **= JupyterLab Desktop**
  ➔ https://github.com/jupyterlab/jupyterlab-desktop/releases

### Remote (cluster) installation:

- **JupyterLab** installed on a remote server and accessed through the browser
  - in $HOME (e.g. using pip or miniconda)
  - system-wide (e.g. with Easybuild, Spark) by the admins.

### Browser-only installation (limited feature set):

- **JupyterLab** local with server + client in your browser **= JupyterLite**
  Includes a browser-ready Python environment named Pyodide.
  ➔ https://jupyter.org/try-jupyter/lab


https://jupyter.org/try-jupyter/lab/
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ notebooks /
Name
Last Modified
Intro.ipynb 12 days ago
Lorenz.ipynb seconds ago
sqlite.ipynb 12 days ago
Launcher
Intro.ipynb
Lorenz.ipynb
Markdown
Pyolite
Some random data, created with JupyterLab!
2
1
0
−1
−2
−3
−2
−1
0
1
2
beta 4.23
rho 32.70
Simple 0 2 Pyolite | Idle
Mode: Command Ln 1, Col 1 Intro.ipynb
Mode: Command Ln 1, Col 1 Lorenz.ipynb

An example: visualizing data in the notebook ✨

Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python. We'll use NumPy to create some random data, and Matplotlib to visualize it.

Note how the code and the results of running the code are bundled together.

```
[1]: from matplotlib import pyplot as plt
     import numpy as np

     # Generate 100 random data points along 3 dimensions
     x, y, scale = np.random.randn(3, 100)
     fig, ax = plt.subplots()

     # Map each onto a scatterplot we'll create with Matplotlib
     ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
     ax.set(title="Some random data, created with JupyterLab!")
     plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

For the default set of parameters, we see the trajectories swirling around two points, called attractors.

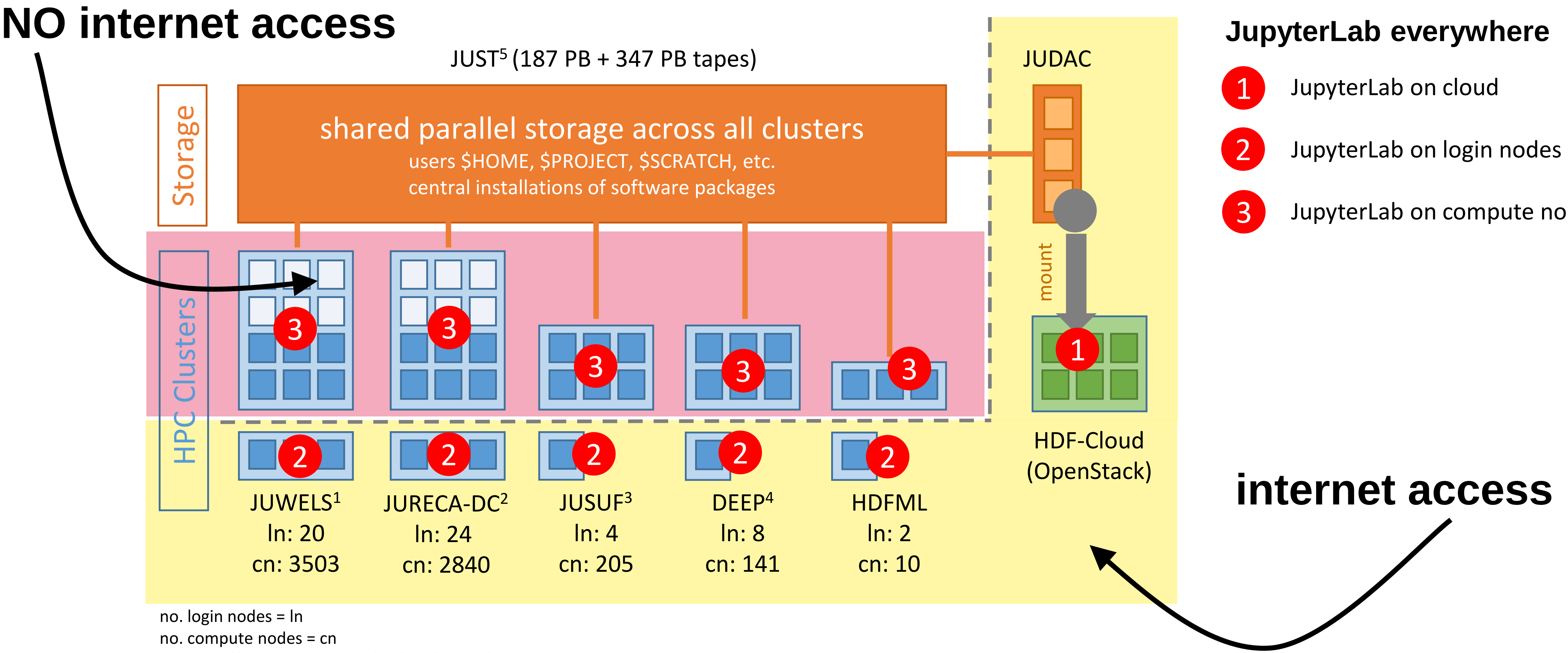
The object returned by `interactive` is a `Widget` object and it has attributes that contain the current result and arguments:

```
[14]: t, x_t = w.result
```

```
<Figure size 432x288 with 0 Axes>
<Figure size 432x288 with 0 Axes>
<Figure size 432x288 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

# START & LOGIN

# JUPYTERLAB EVERYWHERE



**JupyterLab everywhere**

1 JupyterLab on cloud

2 JupyterLab on login nodes

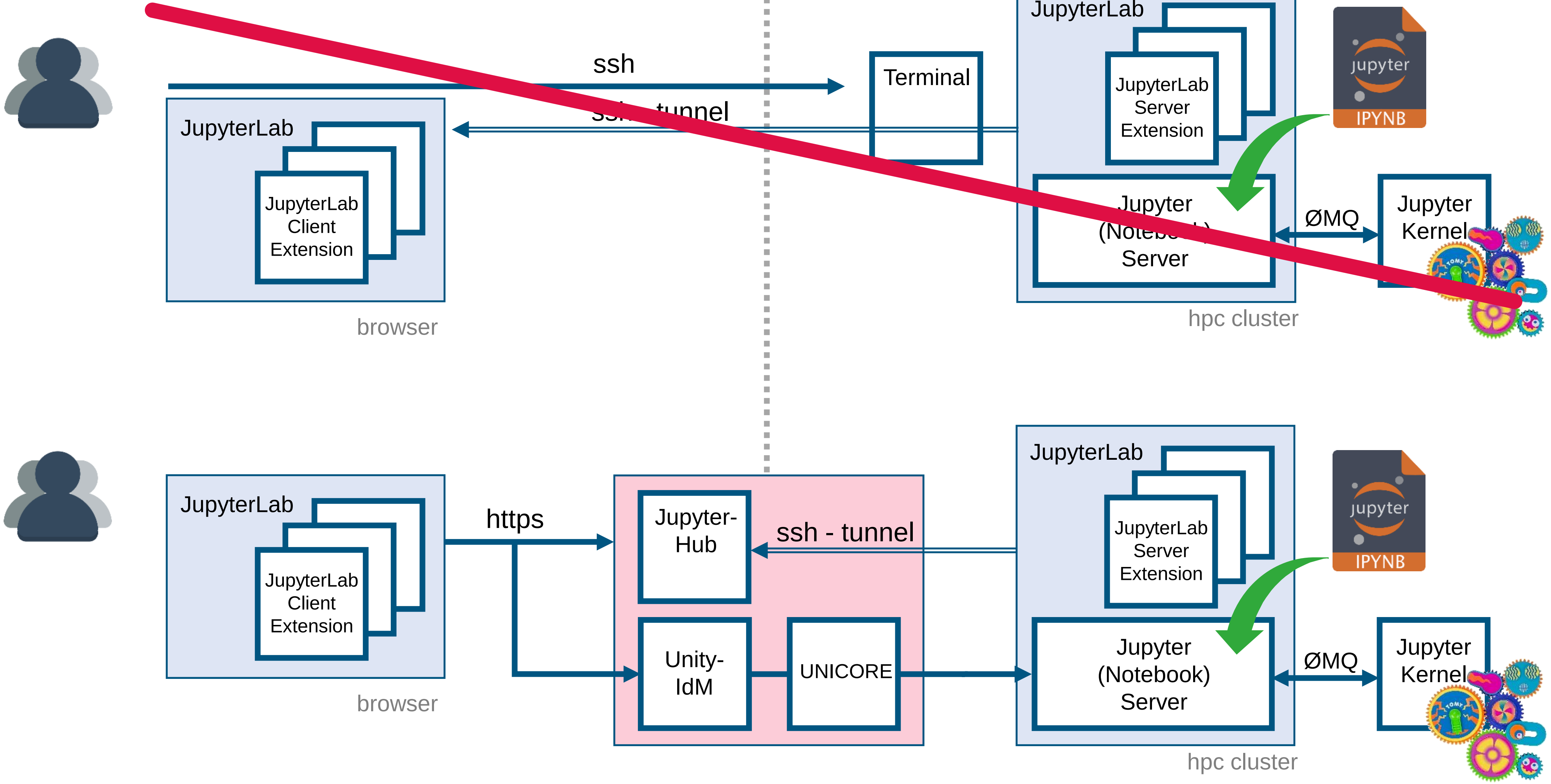
3 JupyterLab on compute nodes

no. login nodes = ln
no. compute nodes = cn

[1] https://apps.fz-juelich.de/jsc/hps/juwels/configuration.html
[2] https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html
[3] https://apps.fz-juelich.de/jsc/hps/jusuf/configuration.html
[4] https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system
[5] https://apps.fz-juelich.de/jsc/hps/just/configuration.html

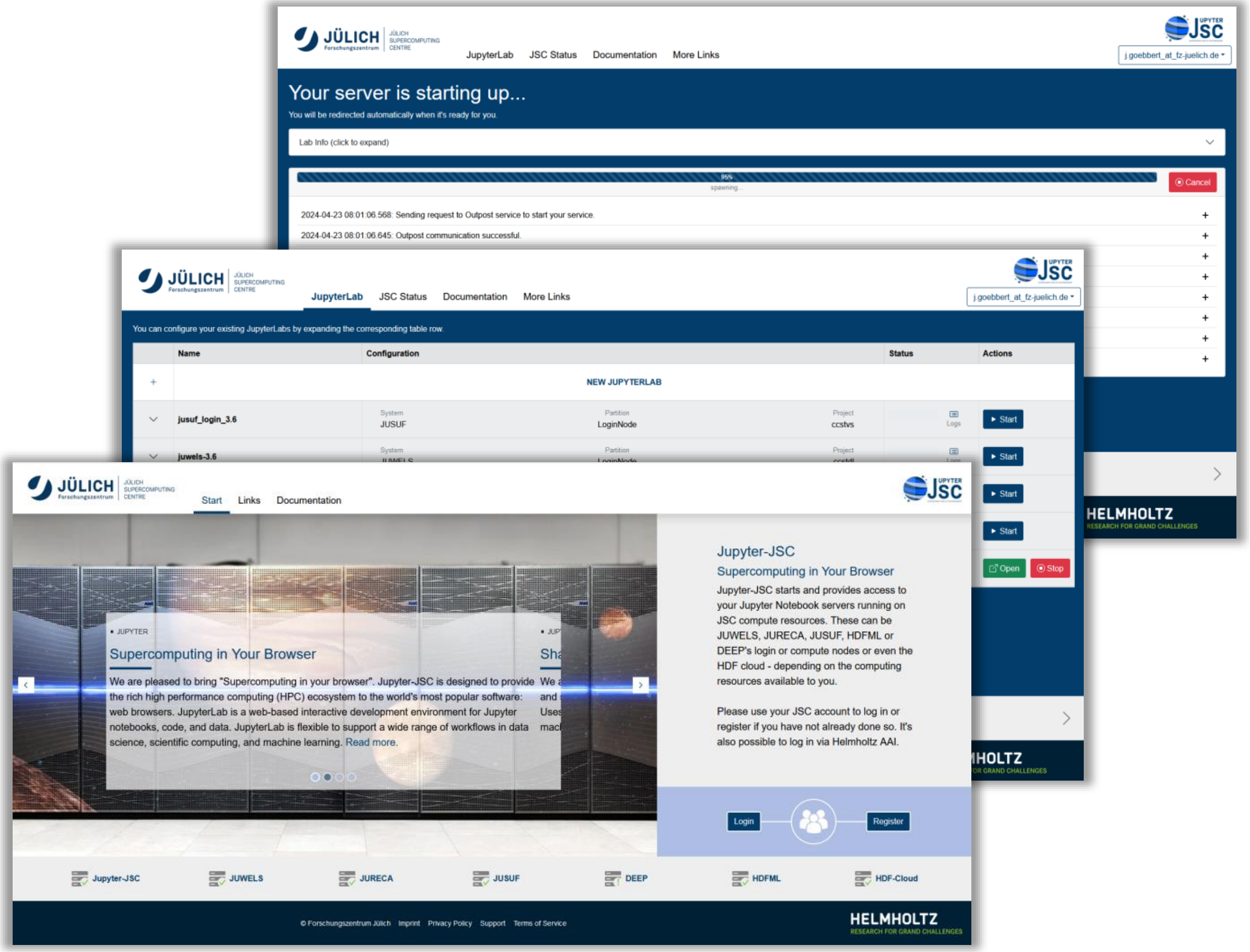
# JUPYTER-JSC WEBSERVICE

## Start your JupyterLab (the easy way)



# JUPYTER-JSC WEBSERVICE

## Start your JupyterLab



JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE
JupyterLab
JSC Status
Documentation
More Links
j.goebbert_at_fz-juelich.de
Lab Info (click to expand)
95%
spawning...
Cancel
Start
Links
Logs
Open
Stop
Login
Register
Jupyter-JSC
JUWELS
JURECA
JUSUF
DEEP
HDFML
HDF-Cloud
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

Your server is starting up...

You will be redirected automatically when it's ready for you.

```
2024-04-23 08:01:06.568: Sending request to Outpost service to start your service.
2024-04-23 08:01:06.645: Outpost communication successful.
```

You can configure your existing JupyterLabs by expanding the corresponding table row.

| | Name | Configuration | | | Status | Actions |
|---|---|---|---|---|---|---|
| + | NEW JUPYTERLAB | | | | | |
| ⌄ | jusuf_login_3.6 | System<br>JUSUF | Partition<br>LoginNode | Project<br>ccstvs | Logs | ▸ Start |
| ⌄ | juwels-3.6 | System<br>JUWELS | Partition<br>LoginNode | Project<br>ccstdl | Logs | ▸ Start |

Jupyter-JSC

Supercomputing in Your Browser

Jupyter-JSC starts and provides access to your Jupyter Notebook servers running on JSC compute resources. These can be JUWELS, JURECA, JUSUF, HDFML or DEEP's login or compute nodes or even the HDF cloud - depending on the computing resources available to you.

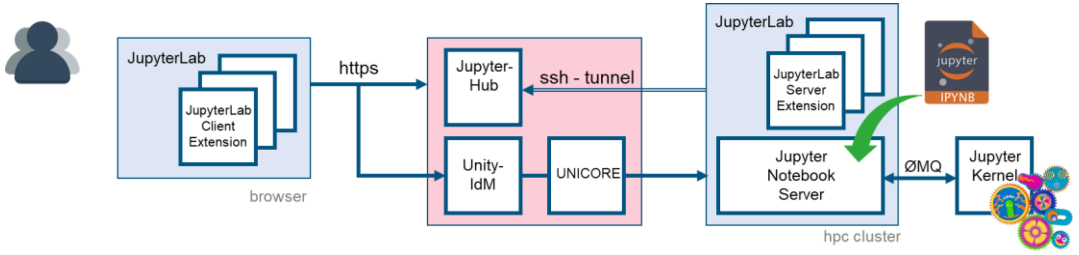
Please use your JSC account to log in or register if you have not already done so. It's also possible to log in via Helmholtz AAI.

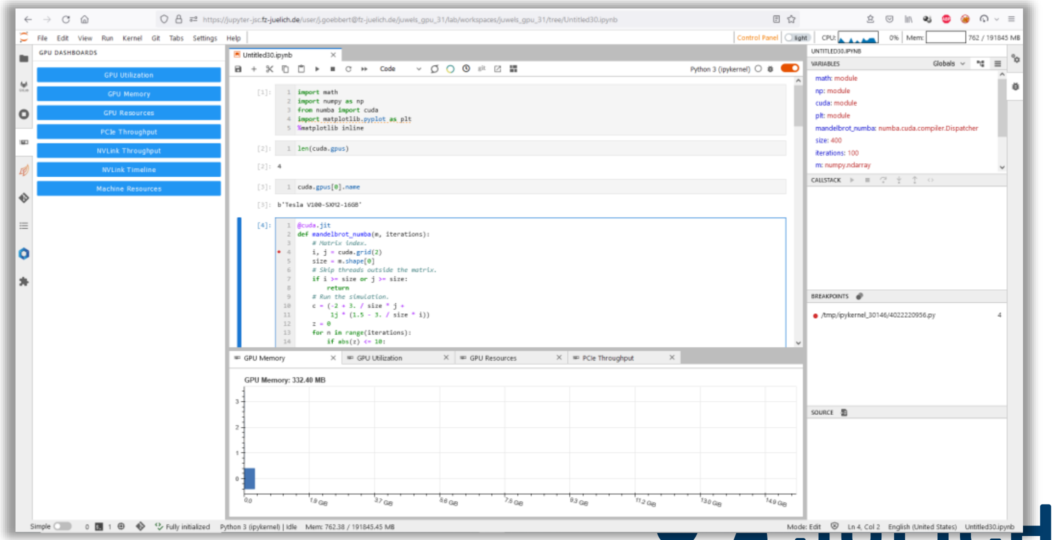
▪ JUPYTER

Supercomputing in Your Browser

We are pleased to bring "Supercomputing in your browser". Jupyter-JSC is designed to provide the rich high performance computing (HPC) ecosystem to the world's most popular software: web browsers. JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible to support a wide range of workflows in data science, scientific computing, and machine learning. Read more.

© Forschungszentrum Jülich Imprint Privacy Policy Support Terms of Service





```
[1]: import math
     import numpy as np
     from numba import cuda
     import matplotlib.pyplot as plt
     %matplotlib inline

[2]: len(cuda.gpus)
[2]: 4

[3]: cuda.gpus[0].name
[3]: b'Tesla V100-SXM2-16GB'

[4]: @cuda.jit
     def mandelbrot_numba(m, iterations):
         # Matrix index.
         i, j = cuda.grid(2)
         size = m.shape[0]
         # Skip threads outside the matrix.
         if i >= size or j >= size:
             return
         # Run the simulation.
         c = (-2 + 3. / size * j +
              1j * (1.5 - 3. / size * i))
         z = 0
         for n in range(iterations):
             if abs(z) <= 10:
```

```
math: module
np: module
cuda: module
plt: module
mandelbrot_numba: numba.cuda.compiler.Dispatcher
size: 400
iterations: 100
m: numpy.ndarray
```

# PRE-ACCESS TODOS

**1) Register & Login**

- ✓ https://judoor.fz-juelich.de

**2) Join** the project „training2412“

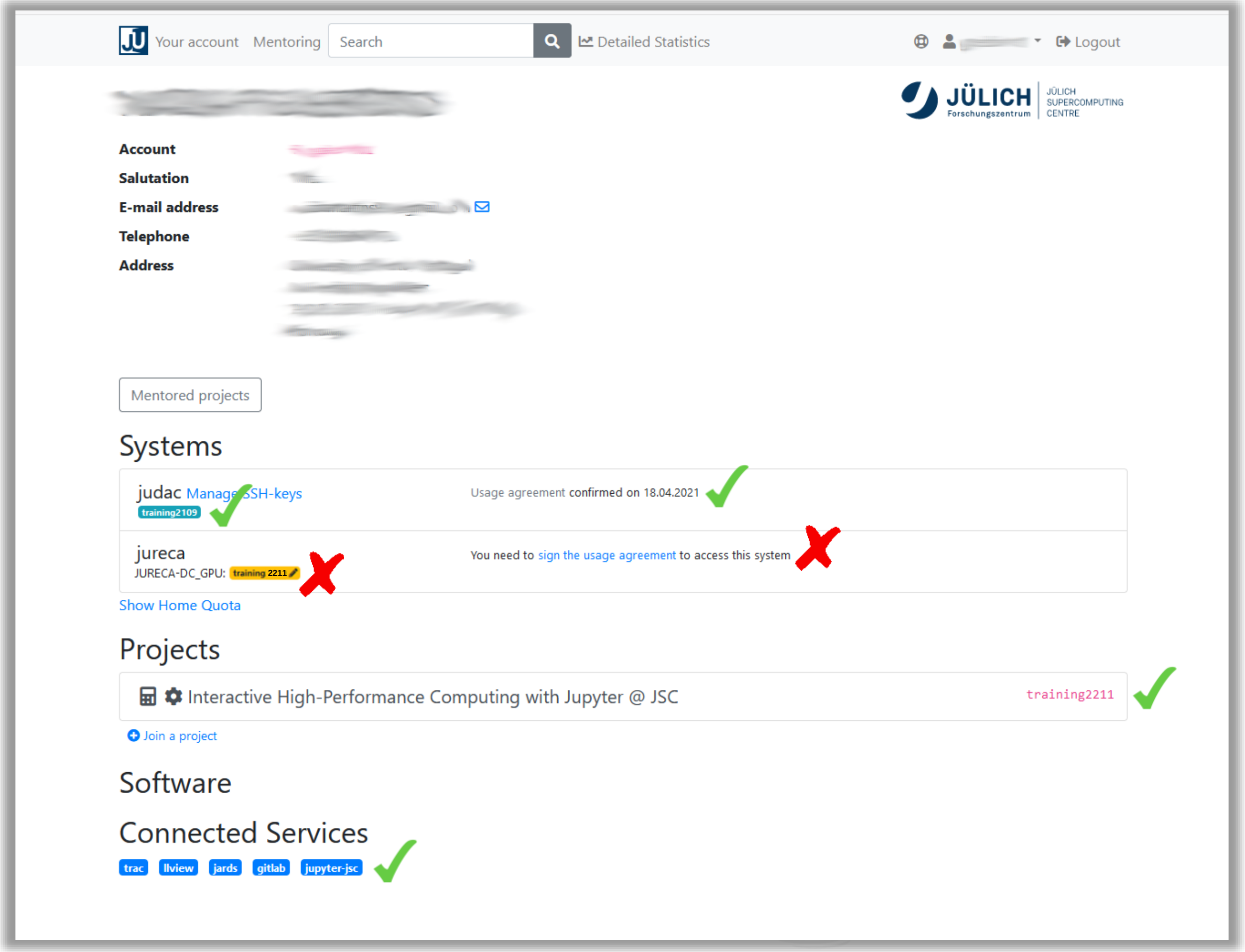
- ✓ Wait to get joined by the project PI

**3) Sign usage agreement**

- ✓ Wait for creation of HPC accounts
- ✓ Update of the SLURM DB

**4) Check Connected Services:**

- ✓ jupyter-jsc



[redacted]

Account [redacted]
Salutation [redacted]
E-mail address [redacted]
Telephone [redacted]
Address [redacted]

## Systems

judac Manage SSH-keys training2109 ✓ — Usage agreement confirmed on 18.04.2021 ✓

jureca JURECA-DC_GPU: training 2211 ✗ — You need to sign the usage agreement to access this system ✗

Show Home Quota

## Projects

Interactive High-Performance Computing with Jupyter @ JSC — training2211 ✓

Join a project

## Software

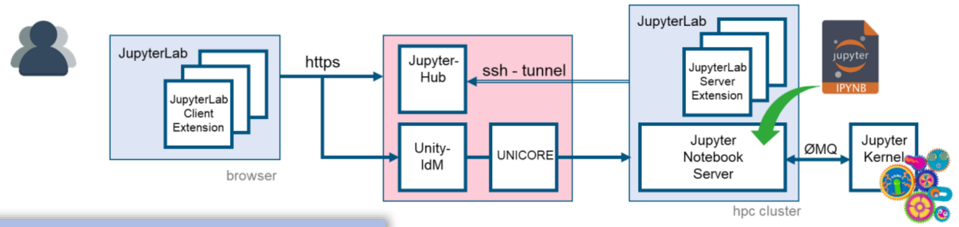
## Connected Services

trac llview jards gitlab jupyter-jsc ✓

For more details, please visit
https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2024.04-jupyter4hpc/-/blob/main/README.md

# JUPYTER-JSC WEBSERVICE

## First time login

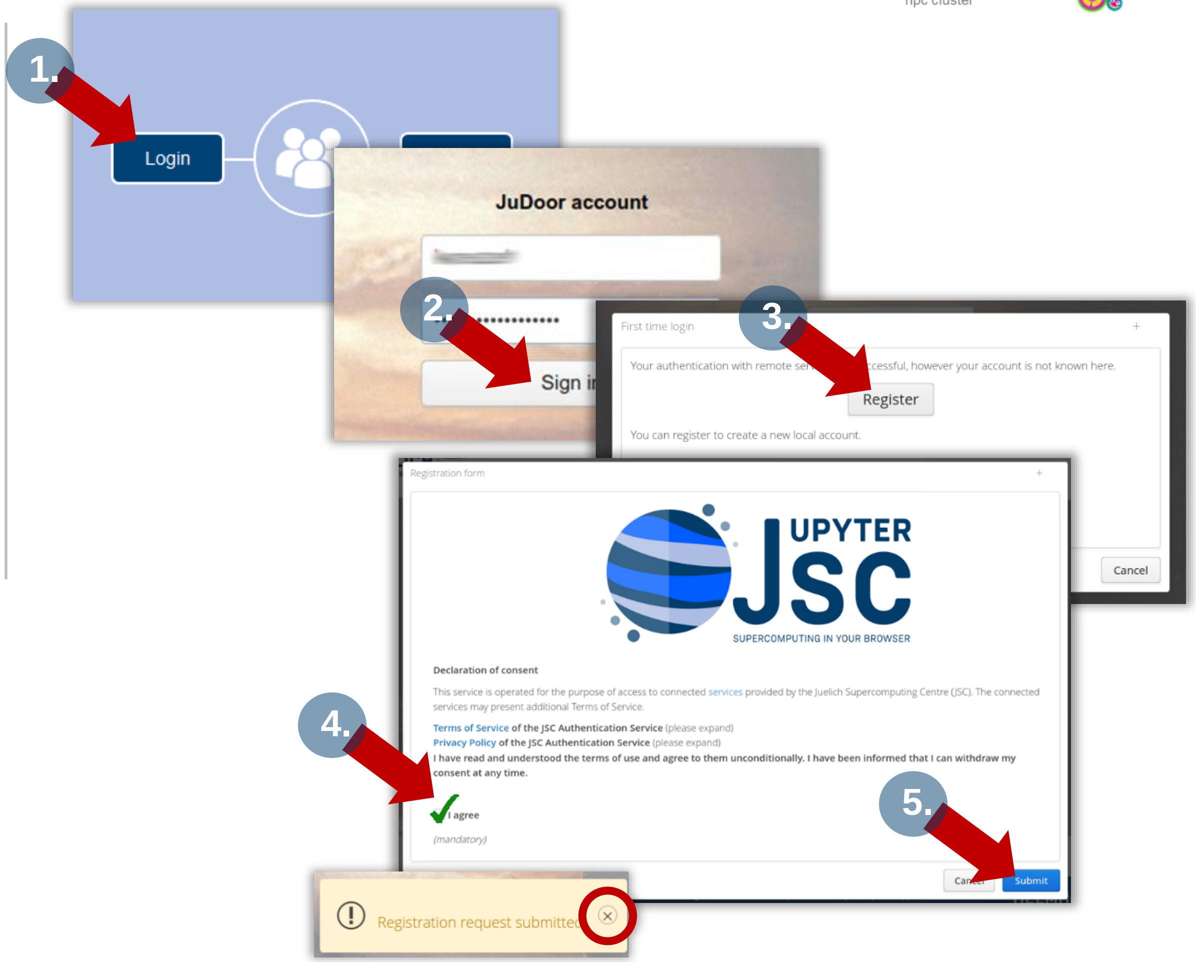


## => https://jupyter-jsc.fz-juelich.de

**Jupyter-JSC first time login**

- Requirements:
  - Registered at **judoor.fz-juelich.de**
    - **(check “Connected Services” = jupyter-jsc)**
  - Project membership + signed systems usage agreement
  - **Waited ~10 minutes**

1. Login at https://jupyter-jsc.fz-juelich.de
2. Sign in with your JSC account
3. Register to Jupyter-JSC
4. Accept usage agreement
5. Submit the registration
6. Wait for email and confirm your email address



JuDoor account

[redacted]

••••••••••••••

Sign in

First time login

Your authentication with remote se ccessful, however your account is not known here.

Register

You can register to create a new local account.

Cancel

Registration form

UPYTER JSC

SUPERCOMPUTING IN YOUR BROWSER

Declaration of consent

This service is operated for the purpose of access to connected services provided by the Juelich Supercomputing Centre (JSC). The connected services may present additional Terms of Service.

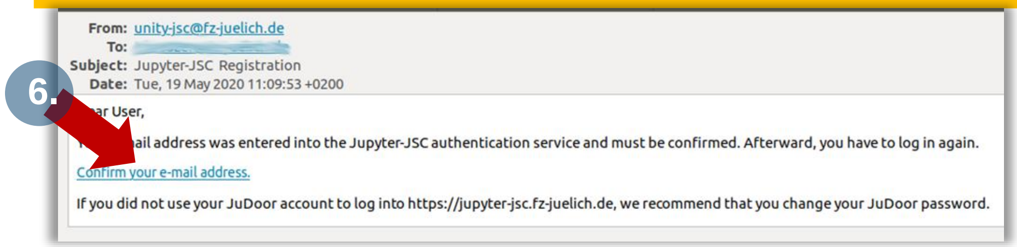
Terms of Service of the JSC Authentication Service (please expand)
Privacy Policy of the JSC Authentication Service (please expand)
I have read and understood the terms of use and agree to them unconditionally. I have been informed that I can withdraw my consent at any time.

✓ I agree

*(mandatory)*

Cancel Submit

Registration request submitted



**From:** unity-jsc@fz-juelich.de
**To:** [redacted]
**Subject:** Jupyter-JSC Registration
**Date:** Tue, 19 May 2020 11:09:53 +0200

ar User,

ail address was entered into the Jupyter-JSC authentication service and must be confirmed. Afterward, you have to log in again.

Confirm your e-mail address.

If you did not use your JuDoor account to log into https://jupyter-jsc.fz-juelich.de, we recommend that you change your JuDoor password.

JÜLICH Forschungszentrum

# JUPYTER-JSC WEBSERVICE

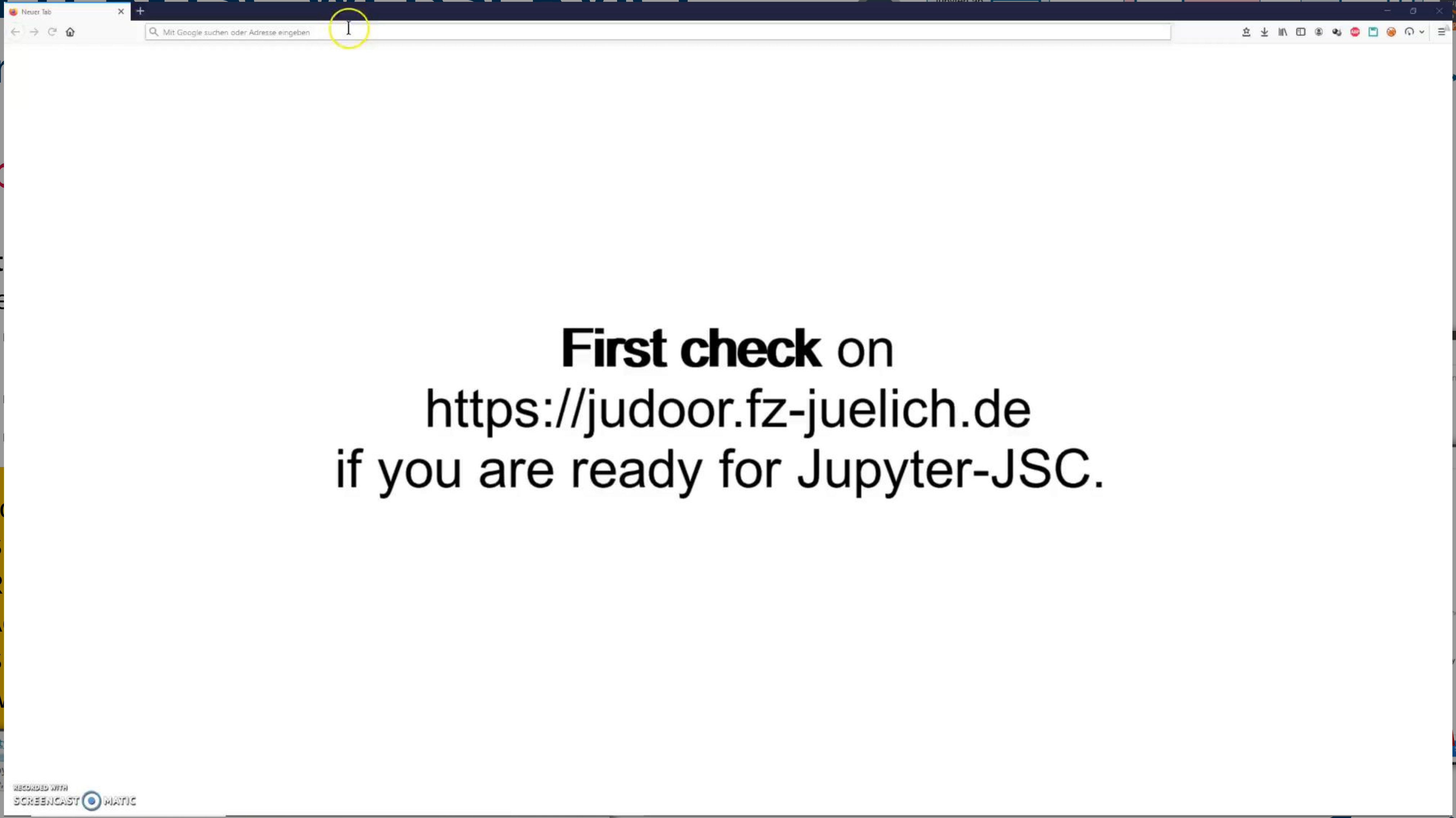
## First ti

=> htt

**Jupyt**

- Re

1. L
2. S
3. R
4. A
5. S
6. W



Neuer Tab
Mit Google suchen oder Adresse eingeben
RECORDED WITH SCREENCAST MATIC

**First check** on
https://judoor.fz-juelich.de
if you are ready for Jupyter-JSC.

From: unit
To: [redacted]
Subject: Jupy
Date: Tue,

6.

ar User,

ail address was entered into the Jupyter-JSC authentication service and must be confirmed. Afterward, you have to log in again.

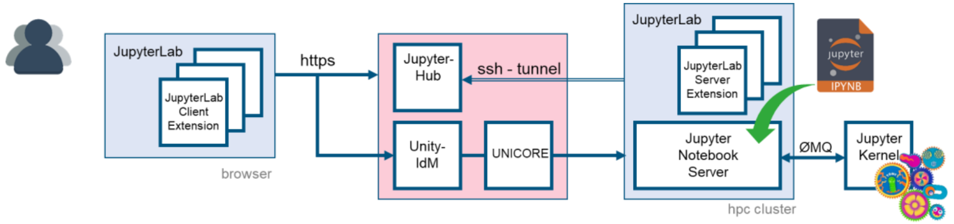
Confirm your e-mail address.

If you did not use your JuDoor account to log into https://jupyter-jsc.fz-juelich.de, we recommend that you change your JuDoor password.

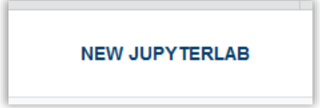
JÜLICH
Forschungszentrum

# JUPYTER-JSC WEBSERVICE

## Control Panel

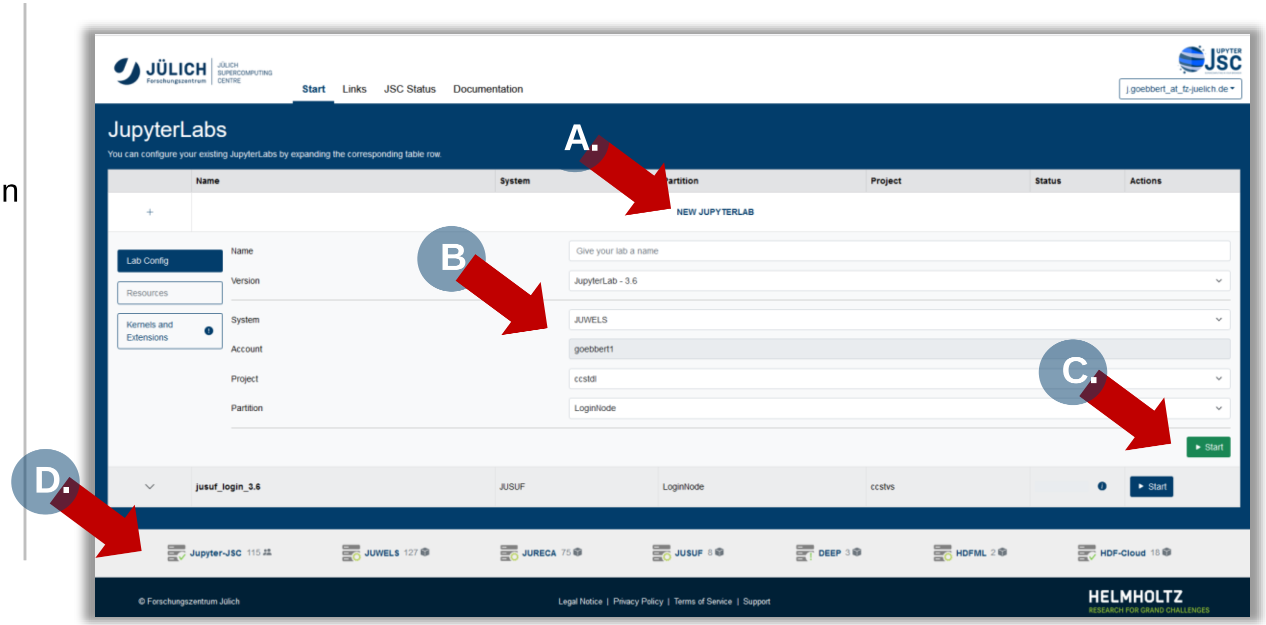


### A. New JupyterLab


NEW JUPYTERLAB

### B. Configuration Dialog

- Lab Config: set Name, Version, System, Account, Project, Partition
- Resources: if running on a compute node
- Kernels and Extensions: Optional addons


JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Start  Links  JSC Status  Documentation

JUPYTER JSC

j.goebbert_at_fz-juelich.de

JupyterLabs

You can configure your existing JupyterLabs by expanding the corresponding table row.

| | Name | System | Partition | Project | Status | Actions |
|---|---|---|---|---|---|---|
| + | | | NEW JUPYTERLAB | | | |
| ⌄ | jusuf_login_3.6 | JUSUF | LoginNode | ccstvs | | ▶ Start |

Lab Config / Resources / Kernels and Extensions

Name: Give your lab a name

Version: JupyterLab - 3.6

System: JUWELS

Account: goebbert1

Project: ccstdl

Partition: LoginNode

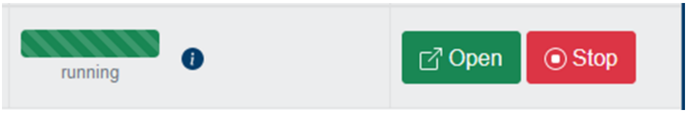
▶ Start

Jupyter-JSC 115  JUWELS 127  JURECA 75  JUSUF 8  DEEP 3  HDFML 2  HDF-Cloud 18

© Forschungszentrum Jülich   Legal Notice | Privacy Policy | Terms of Service | Support   HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

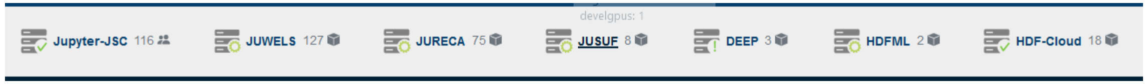
### C. Actions

- Start/Open/Stop a JupyterLab
- Change/Delete **configuration**


running

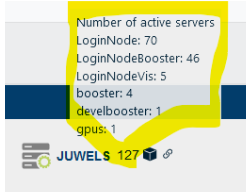
Open  Stop

### D. Statusbar


develgpus: 1

Jupyter-JSC 116  JUWELS 127  JURECA 75  JUSUF 8  DEEP 3  HDFML 2  HDF-Cloud 18

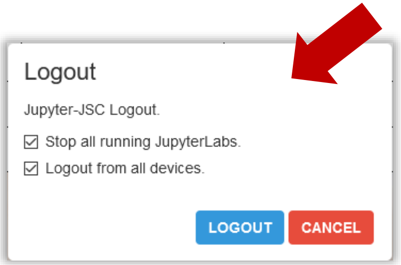
- Shows, (hover to get more details)
  - Number of active users in the last 24h
  - Number of running JupyterLabs
- Click to see system status page


Number of active servers
LoginNode: 70
LoginNodeBooster: 46
LoginNodeVis: 5
booster: 4
develbooster: 1
gpus: 1

JUWELS 127

### E. Logout

- **Logout will ask what you want to do with the running JupyterLabs – be careful what you answer!**


Logout

Jupyter-JSC Logout.

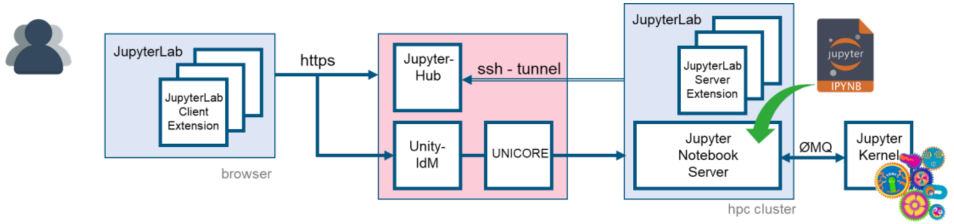
☑ Stop all running JupyterLabs.

☑ Logout from all devices.

LOGOUT  CANCEL

# JUPYTER-JSC WEBSERVICE

## JupyterLab Configuration



**Jupyter-JSC – Configuration**

Available options **depend on**

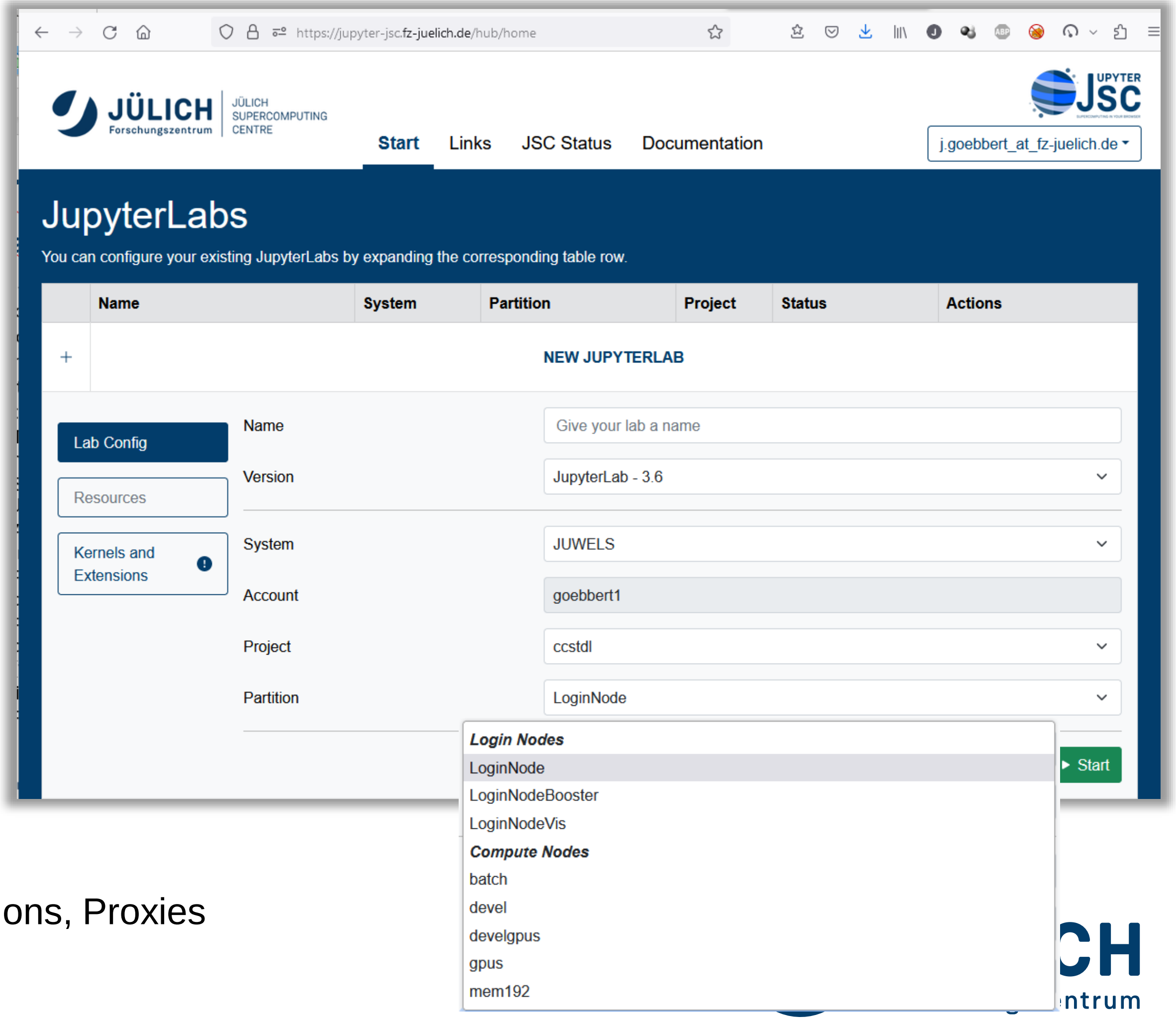
- user account settings visible in **judoor.fz-juelich.de**
- system specific usage agreement on JuDoor is signed (!!!)
- currently available systems in all of your projects

**Basic options**

- Version:
  multiple versions of JupyterLab are installed
- System:
  JUWELS, JURECA, JUSUF, DEEP, HDFML, HDF-Cloud
- Account:
  In general users only have a single account
- Project:
  project which have access to the selected system
- Partition:
  partition which are accessible by the project
  (this includes the decision for LoginNode and ComputeNode)

**Extra options**

- Partition == compute      Resources
- Kernel and Extensions      non-default JupyterKernel, Extensions, Proxies


https://jupyter-jsc.fz-juelich.de/hub/home
JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE
JUPYTER JSC
Start
Links
JSC Status
Documentation
j.goebbert_at_fz-juelich.de

JupyterLabs

You can configure your existing JupyterLabs by expanding the corresponding table row.

| | Name | System | Partition | Project | Status | Actions |
|---|---|---|---|---|---|---|
| + | NEW JUPYTERLAB | | | | | |

Name: Give your lab a name

Version: JupyterLab - 3.6

System: JUWELS

Account: goebbert1

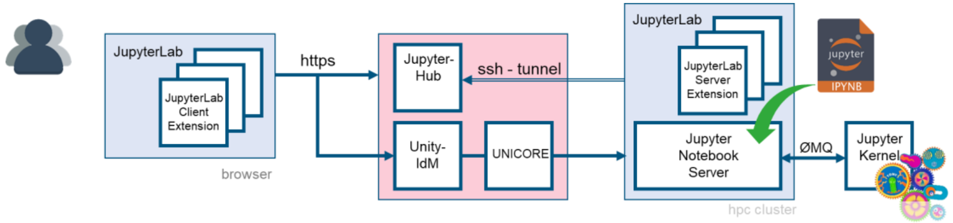
Project: ccstdl

Partition: LoginNode

***Login Nodes***
LoginNode
LoginNodeBooster
LoginNodeVis
***Compute Nodes***
batch
devel
develgpus
gpus
mem192

# JUPYTER-JSC WEBSERVICE

## JupyterLab Configuration

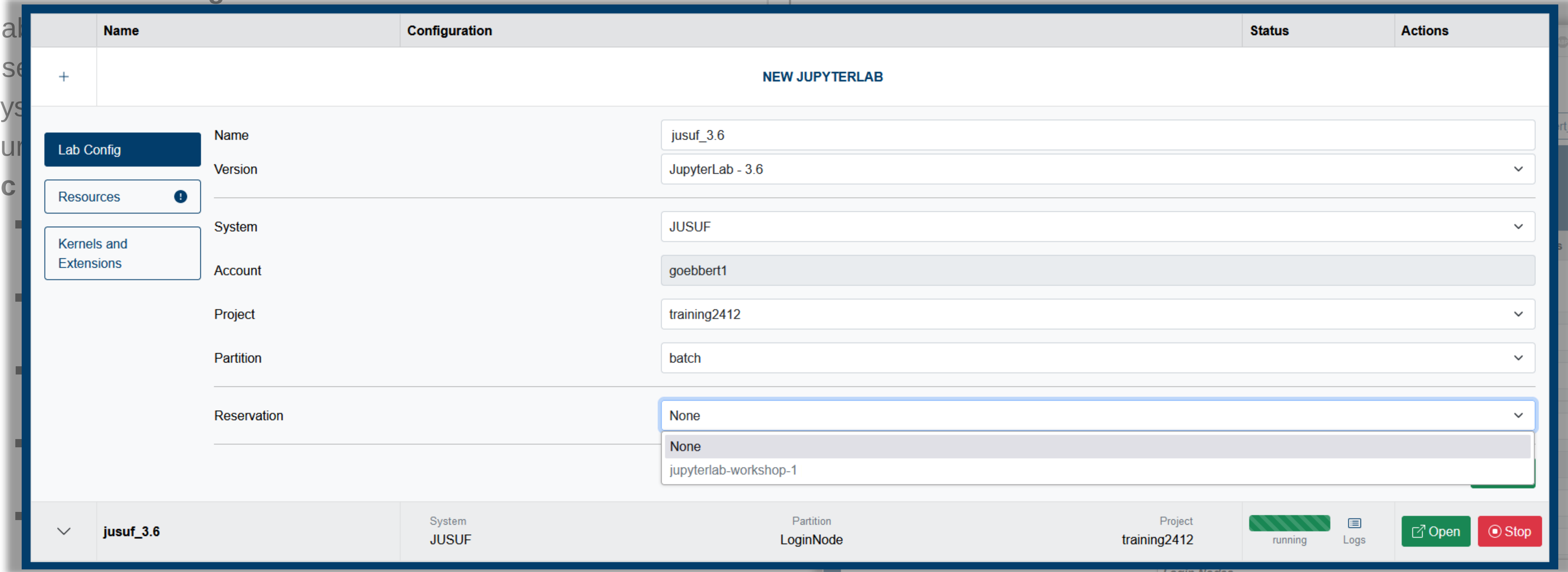


### Jupyter-JSC – Configuration

Availa

- use
- sys
- cur

Basic



| | Name | Configuration | Status | Actions |
|---|---|---|---|---|
| + | NEW JUPYTERLAB | | | |

Name: jusuf_3.6

Version: JupyterLab - 3.6

System: JUSUF

Account: goebbert1

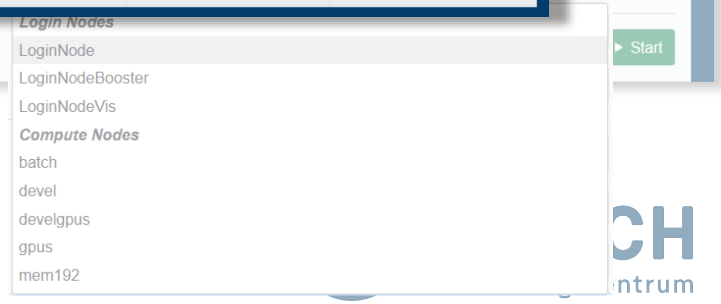
Project: training2412

Partition: batch

Reservation: None

- None
- jupyterlab-workshop-1

| | | System | Partition | Project | Status | Actions |
|---|---|---|---|---|---|---|
| ⌄ | jusuf_3.6 | JUSUF | LoginNode | training2412 | running<br>Logs | Open<br>Stop |



*Login Nodes*
- LoginNode
- LoginNodeBooster
- LoginNodeVis

*Compute Nodes*
- batch
- devel
- develgpus
- gpus
- mem192

(this includes the decision for LoginNode and ComputeNode)

Extra options

- Partition == compute  Resources
- Kernel and Extensions  non-default JupyterKernel, Extensions, Proxies

# CONCLUSION

## Why Jupyter is so popular among Data Scientists

JupyterLab …

… is a **web-based platform for interactive computing and data analysis** that is well-suited to the needs of research software engineers.

… provides researchers with a **comprehensive environment** for working with code, text, multimedia, and data, making it an ideal tool for a wide range of research tasks.

… is designed to be **flexible and customizable**, and can be modified to suit the specific needs and workflows of individual researchers.

… supports the creation of **reproducible research** through its support for Jupyter notebooks.

… supports **collaboration and sharing** of research work through its support for sharing notebooks, dashboards, and other elements of a research project.

… provides a wide range of **extensions and plugins** that can be used to integrate other tools and services into the environment.

… is an **open-source project**, which means that researchers have access to the source code and can contribute to its development.

JÜLICH
Forschungszentrum