INTERACTIVE HPC WITH JUPYTERLAB

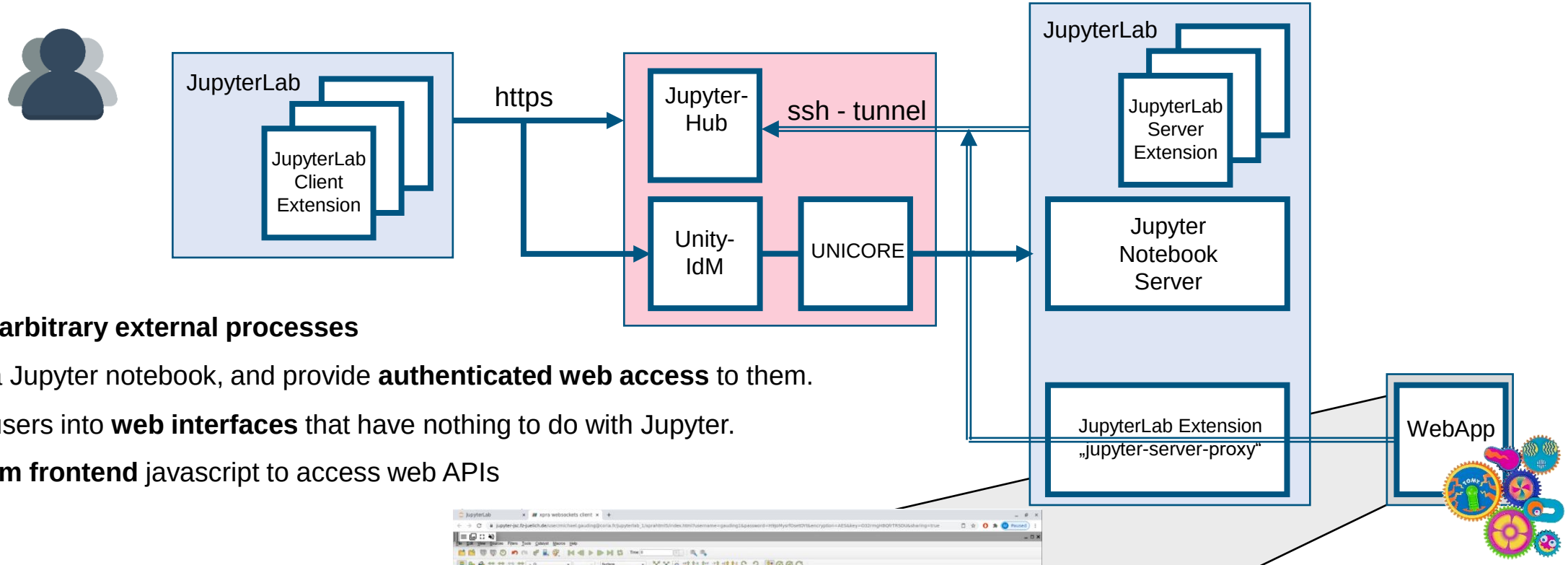
Training Course – Jupyter Server Proxy

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

JUPYTER SERVER PROXY

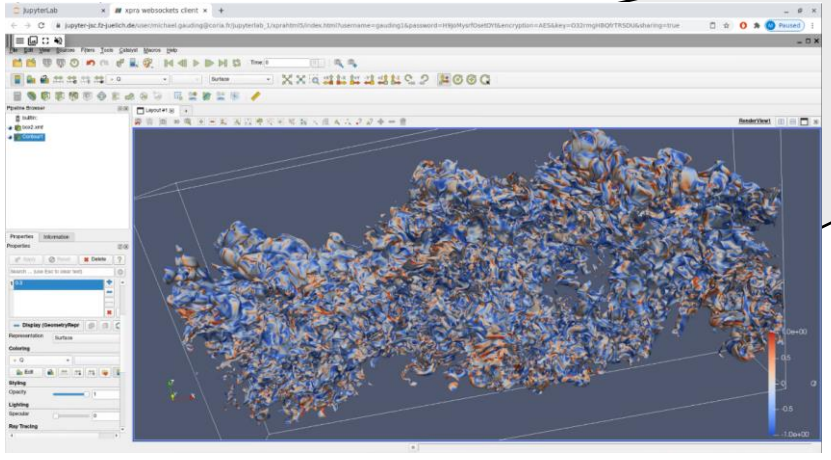
JUPYTERLAB – WEBSERVICE PROXY

Extension: jupyter-server-proxy



Allows to run **arbitrary external processes**

- alongside a Jupyter notebook, and provide **authenticated web access** to them.
- launching users into **web interfaces** that have nothing to do with Jupyter.
- **access from frontend** javascript to access web APIs

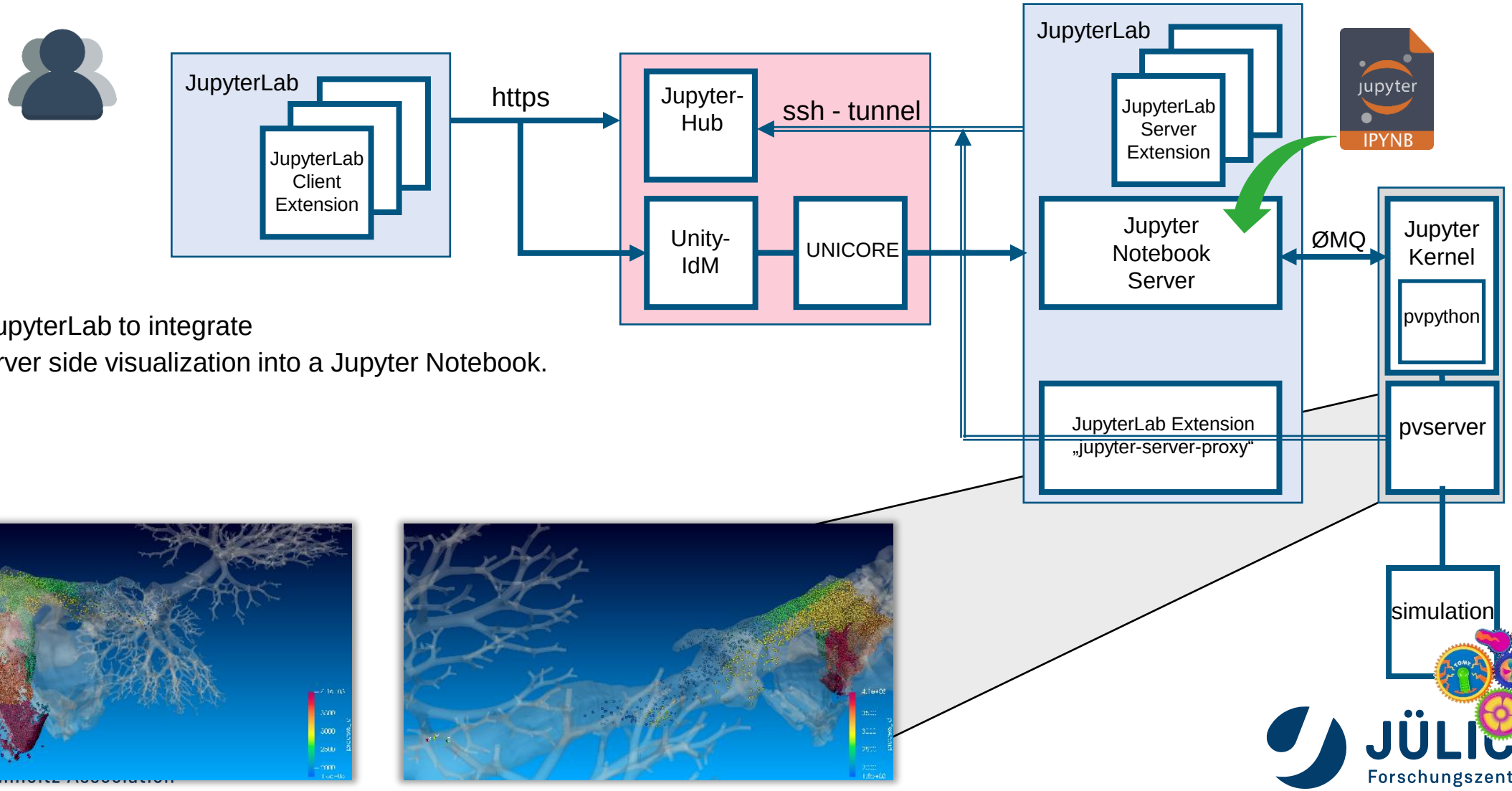


Turbulent mixing with variable density,
subset of 1939x600x3584 grid points, Michael Gauding, CORIA

https://github.com/jupyterhub/jupyter-server-proxy

JUPYTERLAB – WEBSERVICE PROXY

Extension: jupyter-server-proxy



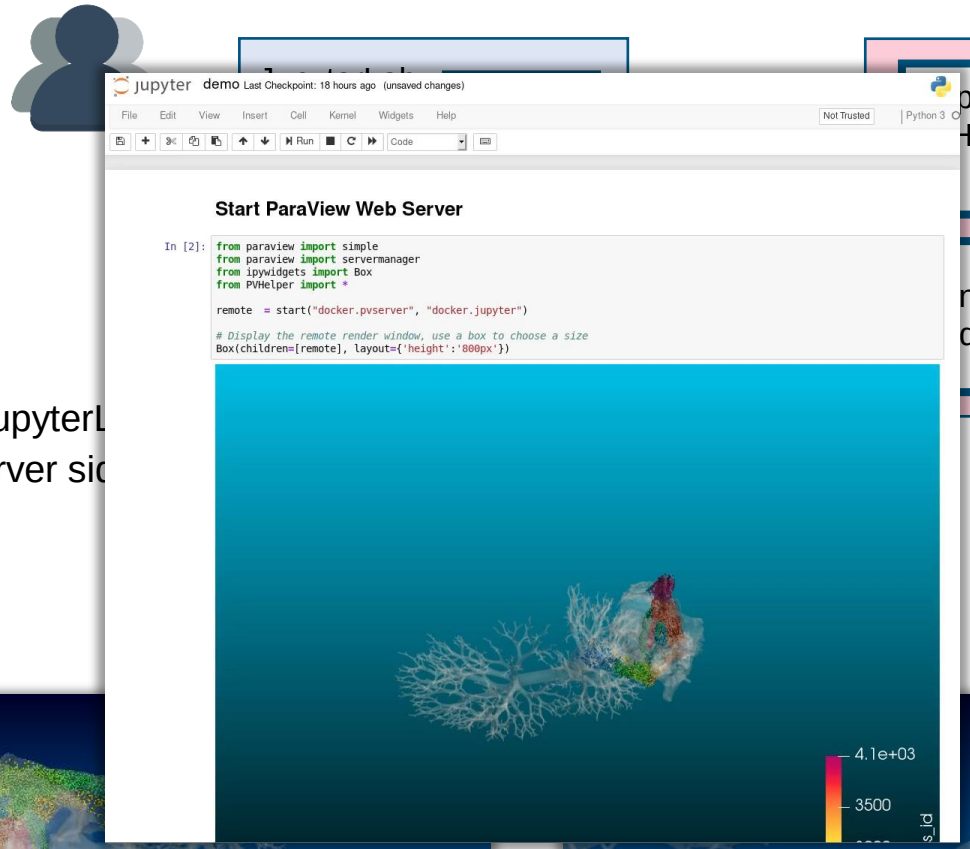
How to use JupyterLab to integrate
interactive server side visualization into a Jupyter Notebook.

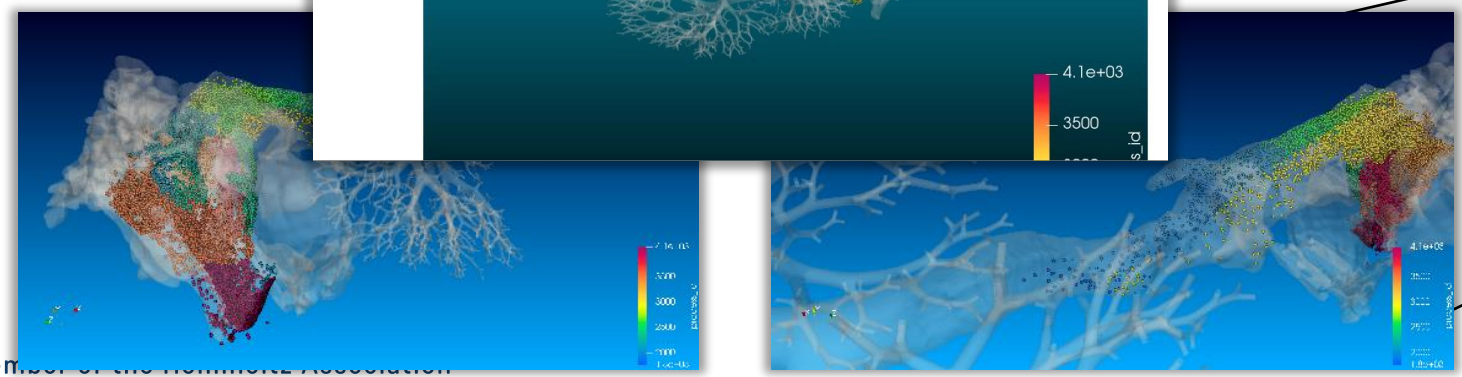
JÜLICH Forschungszentrum

JUPYTERLAB – WEBSERVICE PROXY

Extension: jupyter-server-proxy

How to use JupyterL
interactive server sic





JÜLICH Forschungszentrum

PORT TUNNELING – WEBSERVICE PROXY

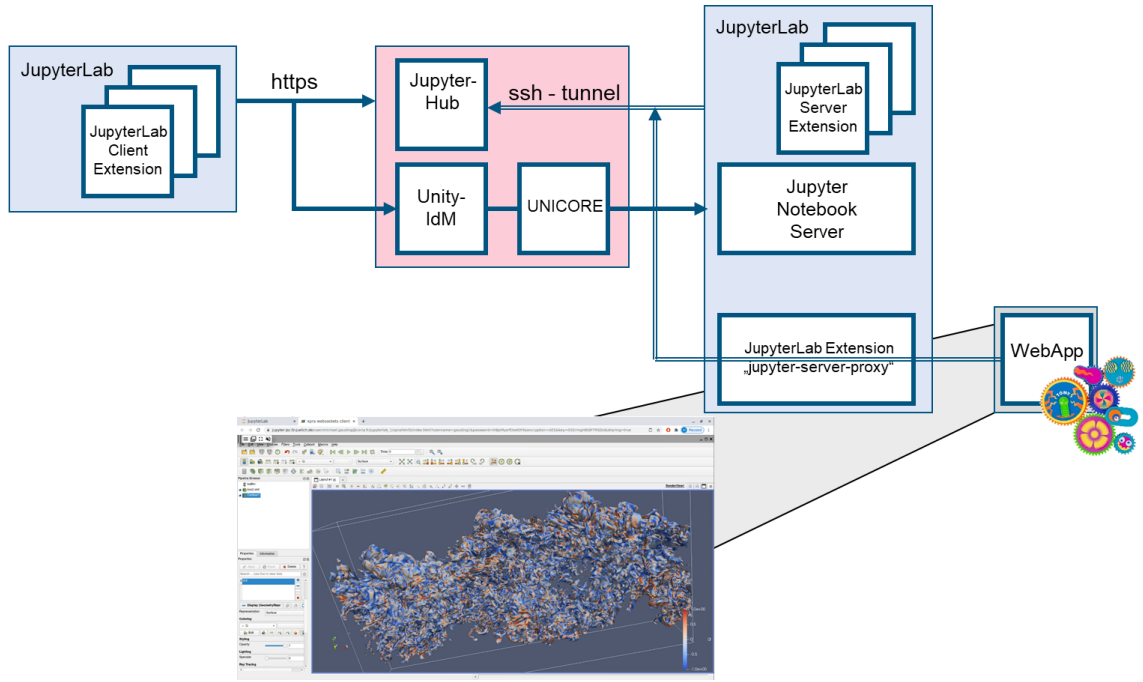
Extension: jupyter-server-proxy

Accessing Arbitrary Ports or Hosts from the Browser

If you have a web-server running on the server
listening on <port>, you can access it through the notebook at
<notebook-base>/proxy/<port>
The URL will be rewritten to remove the above prefix.

You can disable URL rewriting by using
<notebook-base>/proxy/absolute/<port>
so your server will receive the full URL in the request.

This works for all ports listening on the local machine.



Example:
https://jupyter-jsc.fz-juelich.de/user/j.goebbert@fz-juelich.de/juwels_login/**proxy/<port>**
https://jupyter-jsc.fz-juelich.de/user/j.goebbert@fz-juelich.de/juwels_login/**proxy/<host>:<port>**

Upcoming: Support proxying to a server process via a Unix socket (#337)

https://jupyter-server-proxy.readthedocs.io/en/latest/arbitrary-ports-hosts.html

PORT TUNNELING – WEBSERVICE PROXY

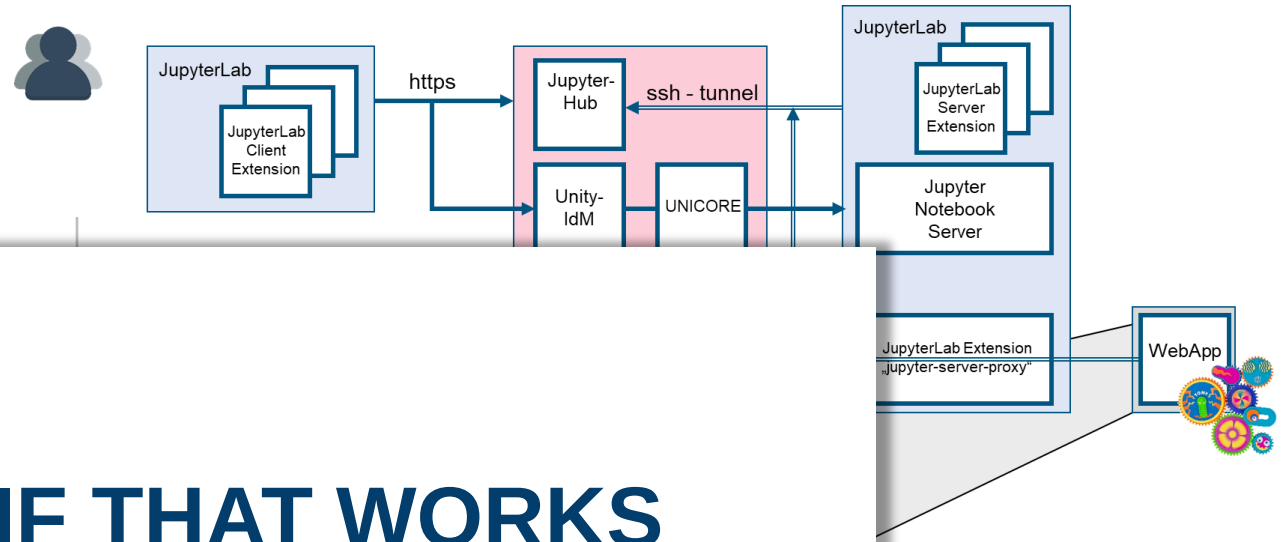
Extension: jupyter-server-proxy

Accessing Arbitrary Ports or Hosts from the Browser

If you have a web-server running on the server
listening on <port>
<notebook-base>
The URL will be re

You can disable U
<notebook-base>
so your server will

This works for all p



LET'S TEST IF THAT WORKS

Example:

https://jupyter-jsc.fz-juelich.de/user/j.goebbert@fz-juelich.de/juwels_login/**proxy/<port>**
https://jupyter-jsc.fz-juelich.de/user/j.goebbert@fz-juelich.de/juwels_login/**proxy/<host>:<port>**

Upcoming: Support proxying to a server process via a Unix socket (#337)

https://jupyter-server-proxy.readthedocs.io/en/latest/arbitrary-ports-hosts.html

JUPYTER SERVER PROXY
ON THE EXAMPLE OF
REMOTE DESKTOP BASED ON XPRA

JUPYTERLAB – REMOTE DESKTOP

Run your X11-Applications in the browser

Jupyter-JSC gives you easy access to a remote desktop

1. https://jupyter-jsc.fz-juelich.de
2. Click on “Xpra”

Xpra - X Persistent Remote Applications
is a tool which runs X clients on a remote host and directs their display to the local machine.

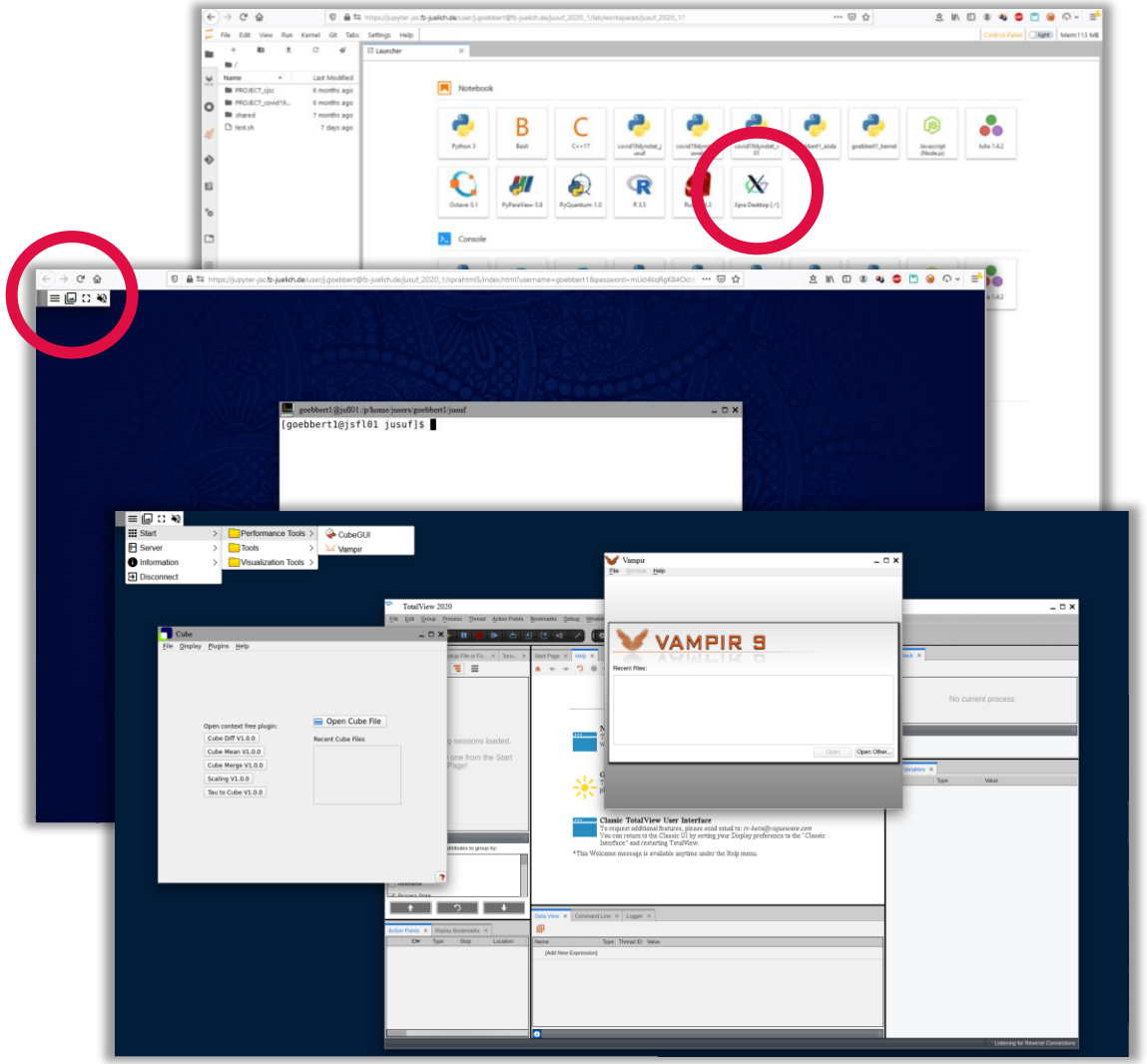
- Runs in a browser
- allows dis-/reconnection without disrupting the forwarded application
- https://xpra.org

The remote desktop will run on the same node as your JupyterLab does (this includes compute nodes).

It gets killed, when you stop your JupyterLab session.

Hint:

- `CTRL + C -> CTRL + Insert`
- `CTRL + V -> SHIFT + Insert`



JUPYTERLAB – REMOTE DESKTOP

Run your X11-Applications in the browser

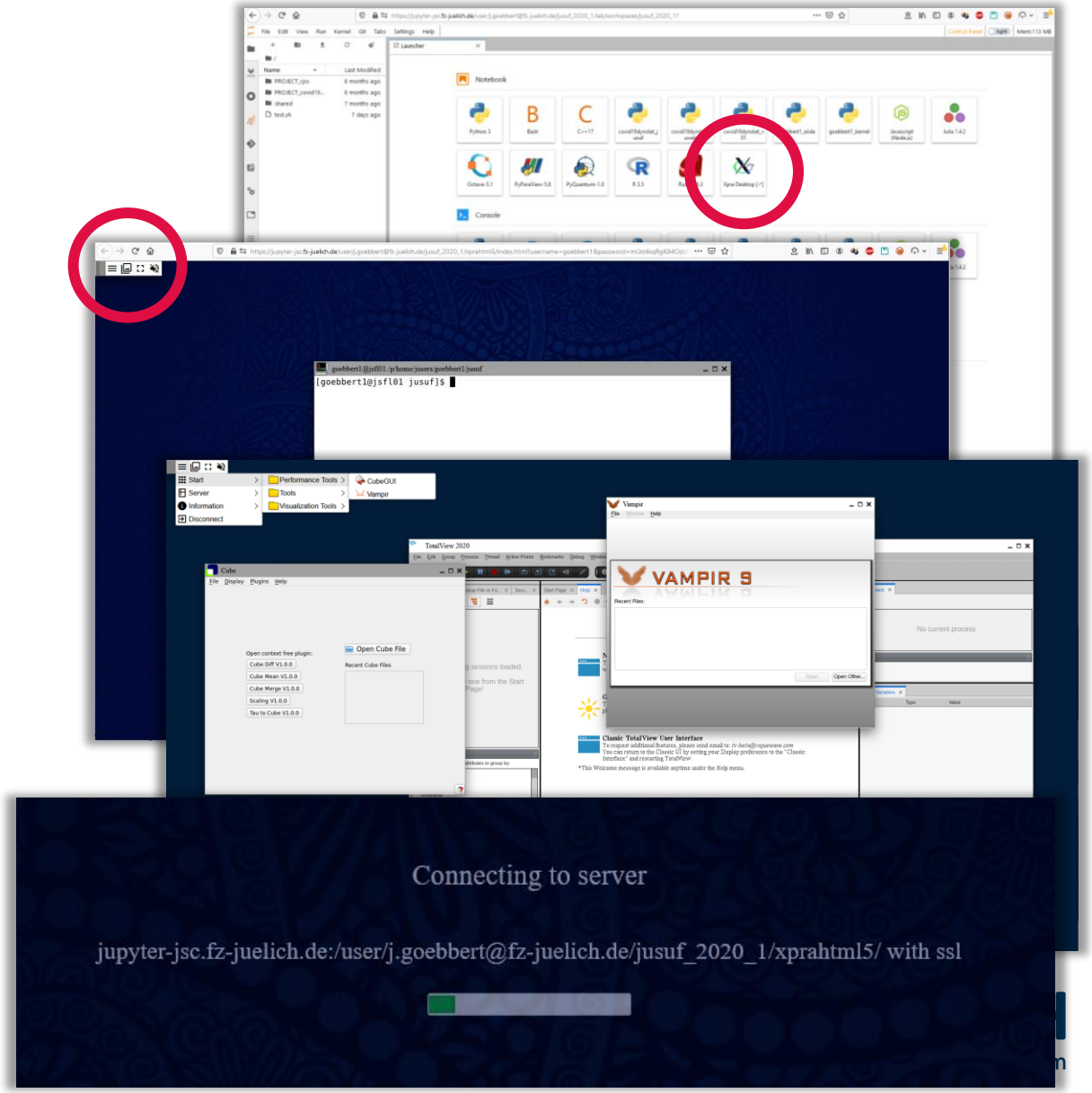
Jupyter-JSC gives you easy access to a remote desktop

1. https://jupyter-jsc.fz-juelich.de
2. Click on “Xpra”

Xpra - X Persistent Remote Applications
is a tool which runs X clients on a remote host and directs their display to the local machine.

- Runs in a browser
- allows dis-/reconnection without disrupting the forwarded application
- https://xpra.org

If the connection got lost at some point,
just hit the “reload” button of your browser.

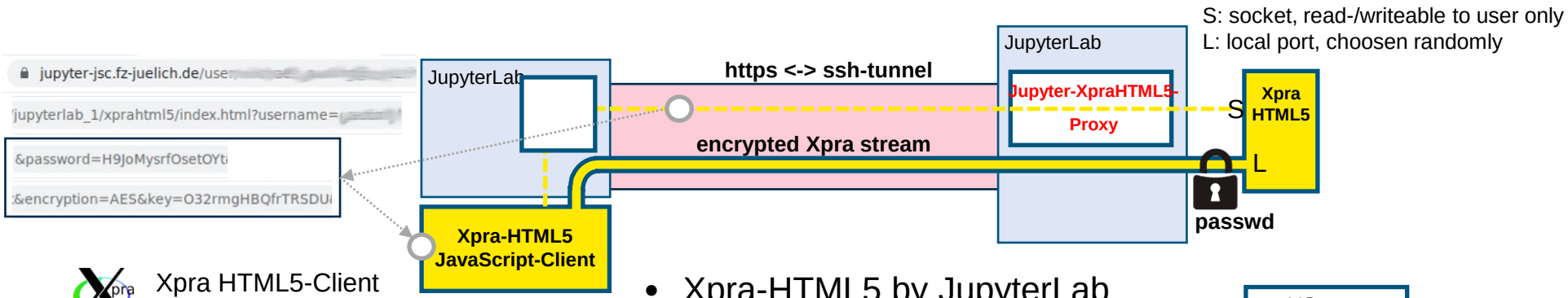


Hint:

- `CTRL + C -> CTRL + Insert`
- `CTRL + V -> SHIFT + Insert`

JUPYTERLAB – REMOTE DESKTOP

Run your X11-Applications in the browser



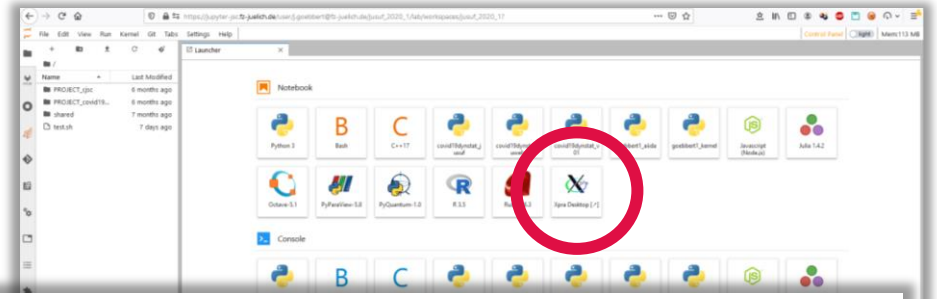
- Xpra-HTML5 by JupyterLab
  - Access through **JupyterLab-URL** by configurable https proxy
  - Auto-generated **one-time password & encryption key** is communicated through https-proxy

XServer (dummy driver)

https://github.com/FZJ-JSC/jupyter-xprahtml5-proxy

JUPYTERLAB – REMOTE DESKTOP

jupyter-xprahtml-proxy – behind the scenes



1. **Register jupyter-server-proxy plugin**
   Python entry_point in setup.py

```
entry_points = {
    'jupyter_serverproxy_servers': [
        'xprahtml5 = jupyter_xprahtml5_proxy:setup_xprahtml5',
    ]
},
```

2. **Launcher entry gets created**
   based on the returned values of setup_xprahtml5()

```
return {
    'environment': {  # as '--socket-dir' does not work as expected, we set this
        'XDG_RUNTIME_DIR': socket_path,
    },
    'command': cmd,
    'mappath': _xprahtml5_mappath,
    'absolute_url': False,
    'timeout': 90,
    'new_browser_tab': True,
    'launcher_entry': {
        'enabled': True,
        'icon_path': os.path.join(HERE, 'icons/xpra-logo.svg'),
        'title': 'Xpra Desktop',
        'path_info': path_info,
    },
}
```

```
cmd = [
    get_xpra_executable('xpra'),
    'start',
    '--html=on',
    '--bind-tcp=0.0.0.0:{port}',
    # '--socket-dir="' + socket_path + '/"',  # fixme: socket_dir not recognized
    # '--server-idle-timeout=86400',  # stop server after 24h with no client connection
    # '--exit-with-client=yes',  # stop Xpra when the browser disconnects
    '--start=xterm -fa "DejaVu Sans Mono" -fs 14',
    # '--start-child=xterm', '--exit-with-children',
    '--tcp-auth=file:filename=' + fpath_passwd,
    '--tcp-encryption=AES',
    '--tcp-encryption-keyfile=' + fpath_aeskey,
    '--clipboard-direction=both',
    '--keyboard-sync=no',  # prevent keys from repeating unexpectedly on high latency
    '--no-mdns',  # do not advertise the xpra session on the local network
    '--no-bell',
    '--no-speaker',
    '--no-printing',
    '--no-microphone',
    '--no-notifications',
    '--no-systemd-run',  # do not delegated start-cmd to the system wide proxy server instance
    # '--dpi=96',  # only needed if Xserver does not support dynamic dpi change
    '--sharing',  # this allows to open the desktop in multiple browsers at the same time
    '--no-daemon',  # mandatory
```

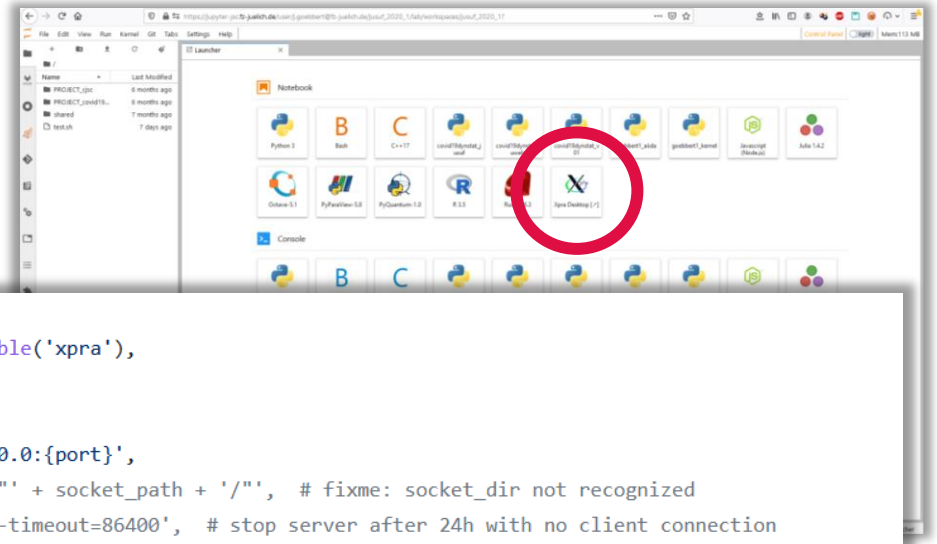
https://github.com/FZJ-JSC/jupyter-xprahtml5-proxy

JUPYTERLAB – REMOTE DESKTOP

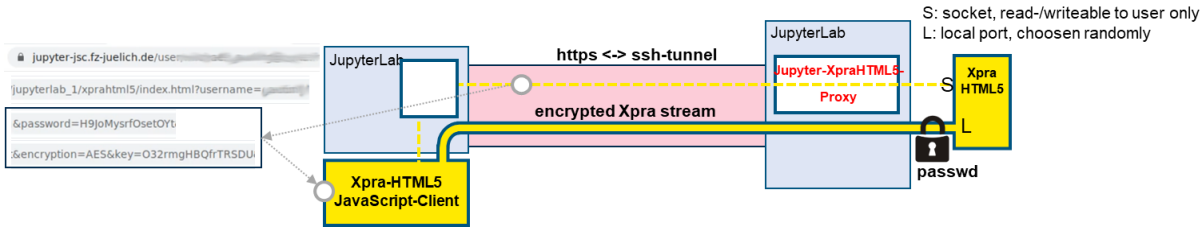
jupyter-xprahtml-proxy – behind the scenes

1. **--tcp-auth=file:filename**

2. **--tcp-encryption=AES**
   --tcp-encryption-keyfile



```
cmd = [
    get_xpra_executable('xpra'),
    'start',
    '--html=on',
    '--bind-tcp=0.0.0.0:{port}',
    # '--socket-dir="' + socket_path + '/"',  # fixme: socket_dir not recognized
    # '--server-idle-timeout=86400',  # stop server after 24h with no client connection
    # '--exit-with-client=yes',  # stop Xpra when the browser disconnects
    '--start=xterm -fa "DejaVu Sans Mono" -fs 14',
    # '--start-child=xterm', '--exit-with-children',
    '--tcp-auth=file:filename=' + fpath_passwd,
    '--tcp-encryption=AES',
    '--tcp-encryption-keyfile=' + fpath_aeskey,
    '--clipboard-direction=both',
    '--keyboard-sync=no',  # prevent keys from repeating unexpectedly on high latency
    '--no-mdns',  # do not advertise the xpra session on the local network
    '--no-bell',
    '--no-speaker',
    '--no-printing',
    '--no-microphone',
    '--no-notifications',
    '--no-systemd-run',  # do not delegated start-cmd to the system wide proxy server instance
    # '--dpi=96',  # only needed if Xserver does not support dynamic dpi change
    '--sharing',  # this allows to open the desktop in multiple browsers at the same time
    '--no-daemon',  # mandatory
```



https://github.com/FZJ-JSC/jupyter-xprahtml5-proxy

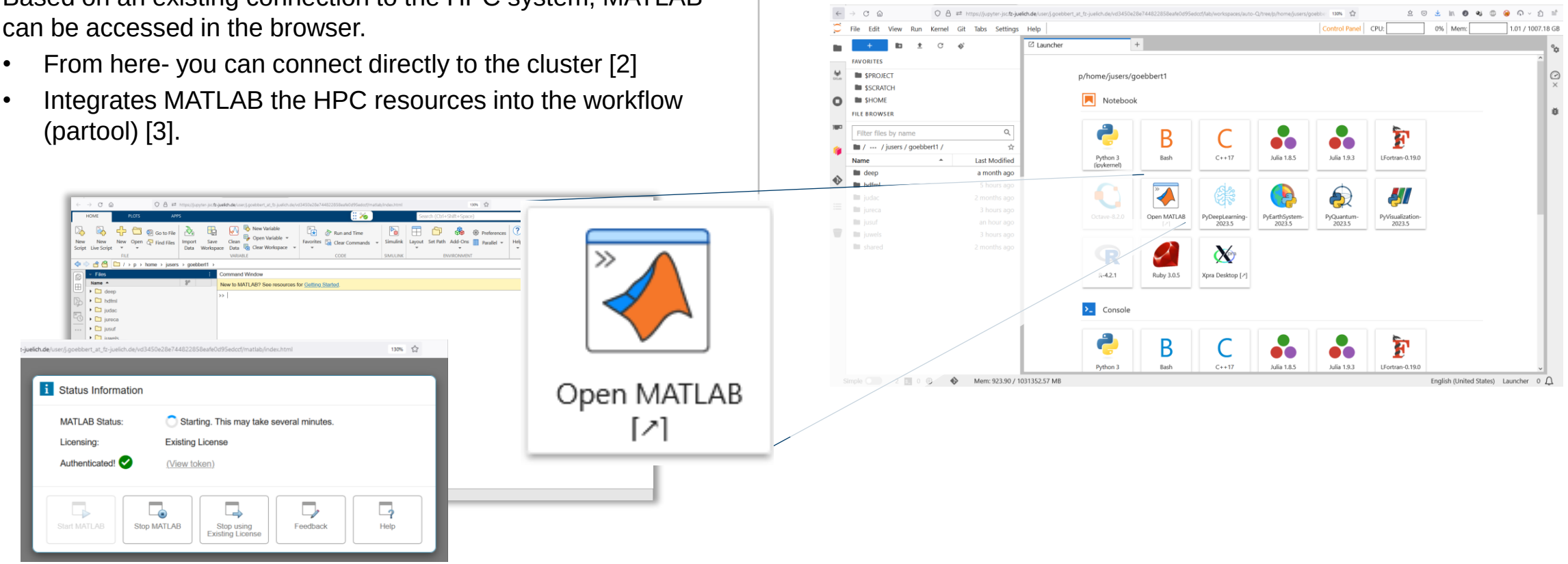
JUPYTERLAB – MATLAB

Web-based GUI for MATLAB

MATLAB – Web-based GUI

Based on an existing connection to the HPC system, MATLAB can be accessed in the browser.

- From here- you can connect directly to the cluster [2]
- Integrates MATLAB the HPC resources into the workflow (partool) [3].



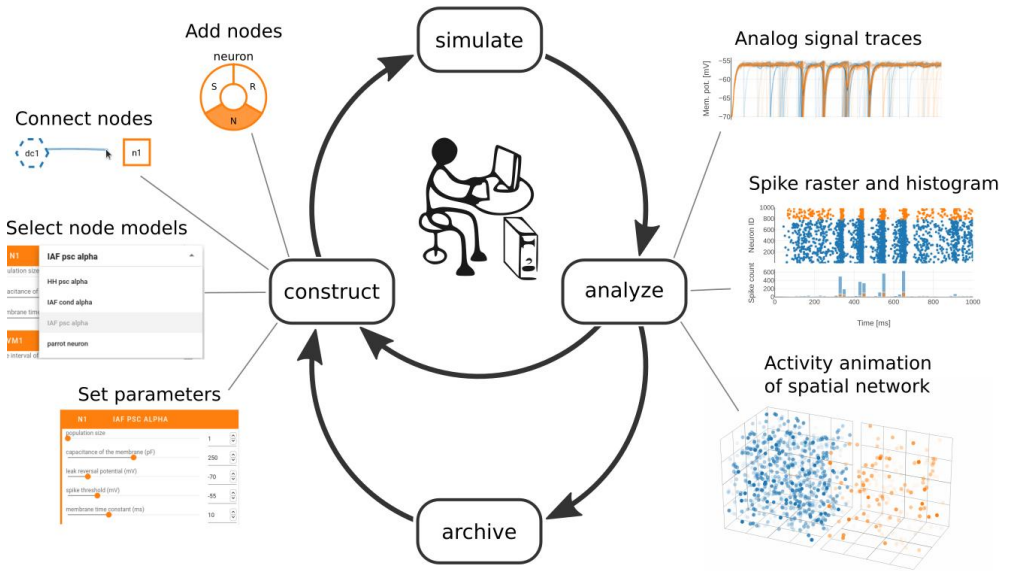
[1] https://www.fz-juelich.de/en/ias/jsc/services/user-support/software-tools/matlab
[2] https://de.mathworks.com/help/parallel-computing/remoteclusteraccess.html
[3] https://de.mathworks.com/products/parallel-computing.html

JUPYTERLAB – NEST DESKTOP

Web-based GUI for Neuroscientists using NEST

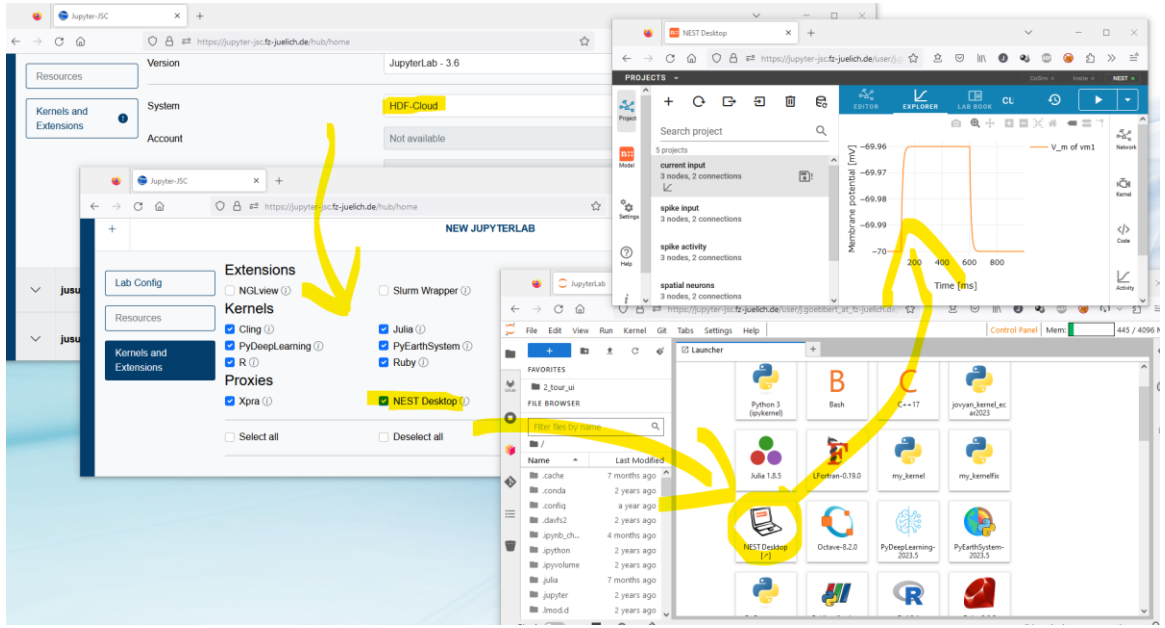
NEST-Desktop

NEST Desktop is a web-based GUI application for NEST Simulator, an advanced simulation tool for the computational neuroscience.



Jupyter-JSC gives you easy access to a NEST-Desktop

With Jupyter-JSC using Jupyter-Server-Proxy authenticated & authorized users get secure access to the WebUI of NEST-Desktop running NEST-simulations on HPC.



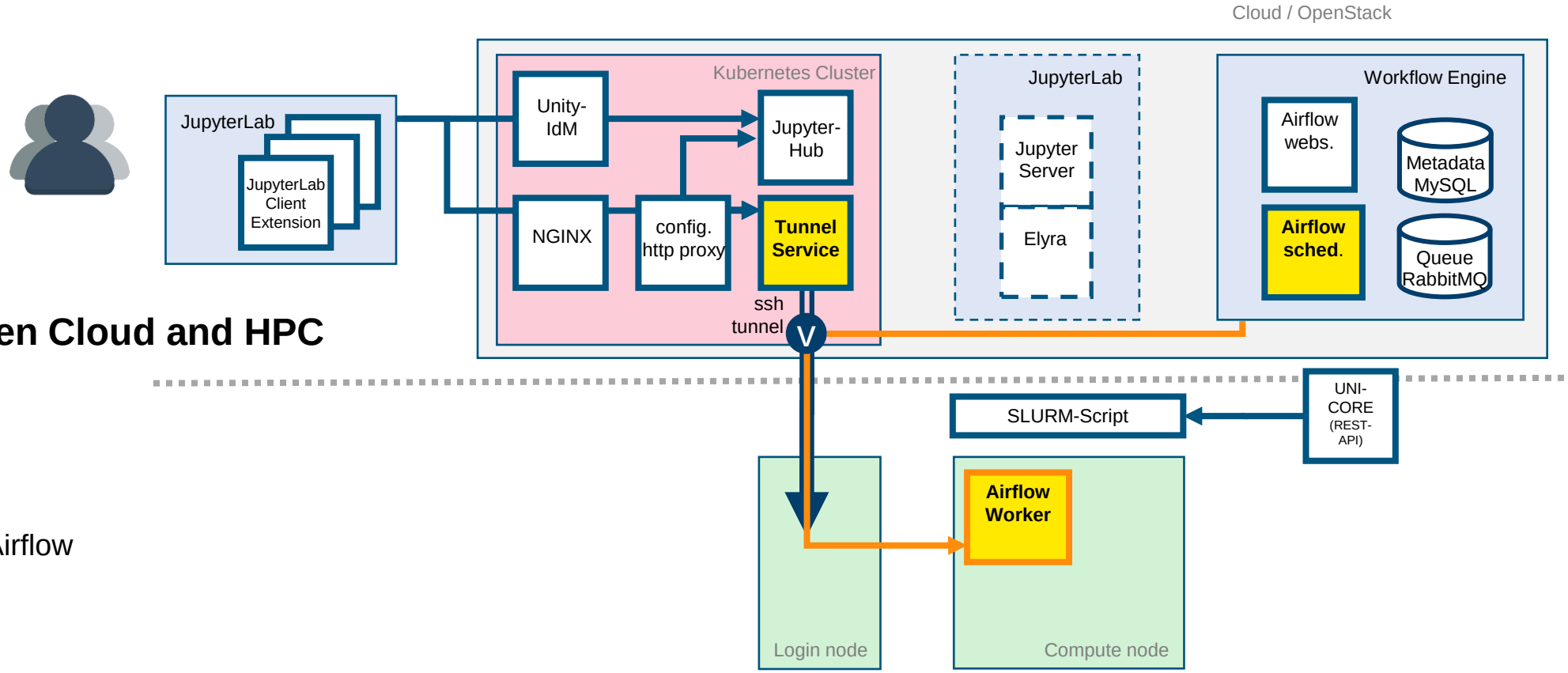
Plugin for Jupyter-Server-Proxy: jupyter-xprahtml5-proxy
https://github.com/jhgoebbert/jupyter-nestdesktop-proxy

[1] https://nest-desktop.readthedocs.io
[2] https://www.nest-simulator.org

JUPYTER-SERVER-PROXY HAS LIMITATIONS

TUNNELING BETWEEN CLOUD AND HPC

Example: workflow manager Apache Airflow



Tunneling between Cloud and HPC

Cloud:

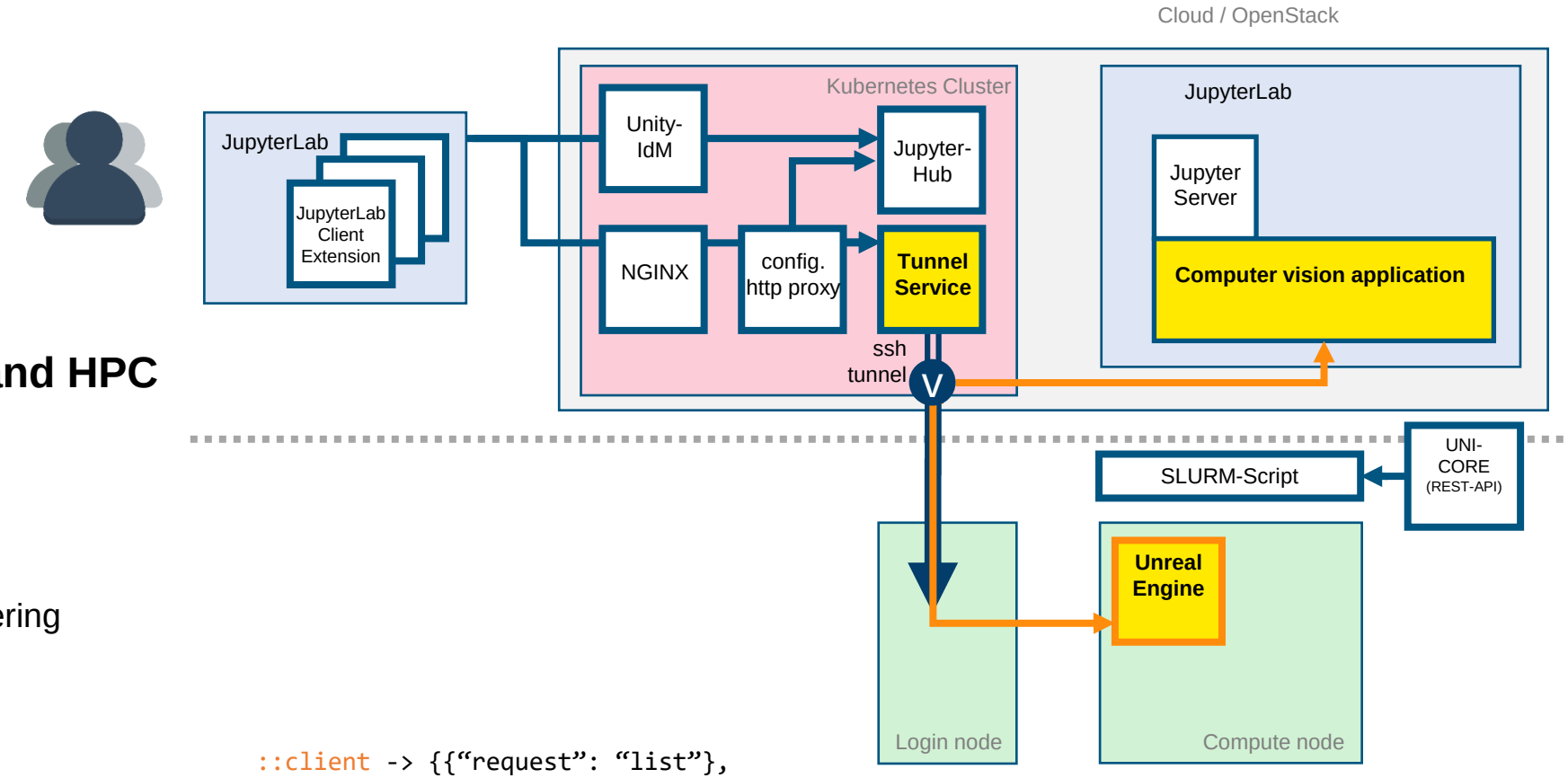
- Apache Airflow

HPC:

- Worker of Apache Airflow

TUNNELING BETWEEN CLOUD AND HPC

Example: Coupled cloud and HPC



Tunneling between Cloud and HPC

Cloud:
- Computer vision application

HPC:
- Unreal Engine for cinematic rendering

WebRTC bridge:
- Video stream
- Data stream, e.g. labels, camera position, etc. ➔

https://github.com/dhelmrich/WebRTCBridge

```
::client -> {{“request”: “list”},
{“kind”:”actor”}}

::server -> {{“answer”:”list”},
{“data”:[“player0”:Pawn,
“terrain”:LandscapeActor,
“grass”:FoilageActor, …]}
```


process
interface
integrierbar
Workflows
visualization
agile
development
analyse
presentation
lifecycle
infographic
activity
collaboration
kollaborativ
flexibel
meeting
teilen
akzeptiert
Webdienste
innovativ
open-source
sicher
data
bekannt
discovery
unterstützt
iterative
big
Jupyter
community
agility
share
multiuser
workflow
erweiterbar
plan
cycle
science
parallel
einfach
interaction
management
intuitiv
cognitive
developement
Industrieakzeptanz
funktionsreich
Spezialisierungen
design
Offline-Einsatz
analytics
domainübergreifend
methodology
knowledge
information