Harnessing the power of Jupyter{Hub,Lab} to make Jean Zay HPC resources more accessible

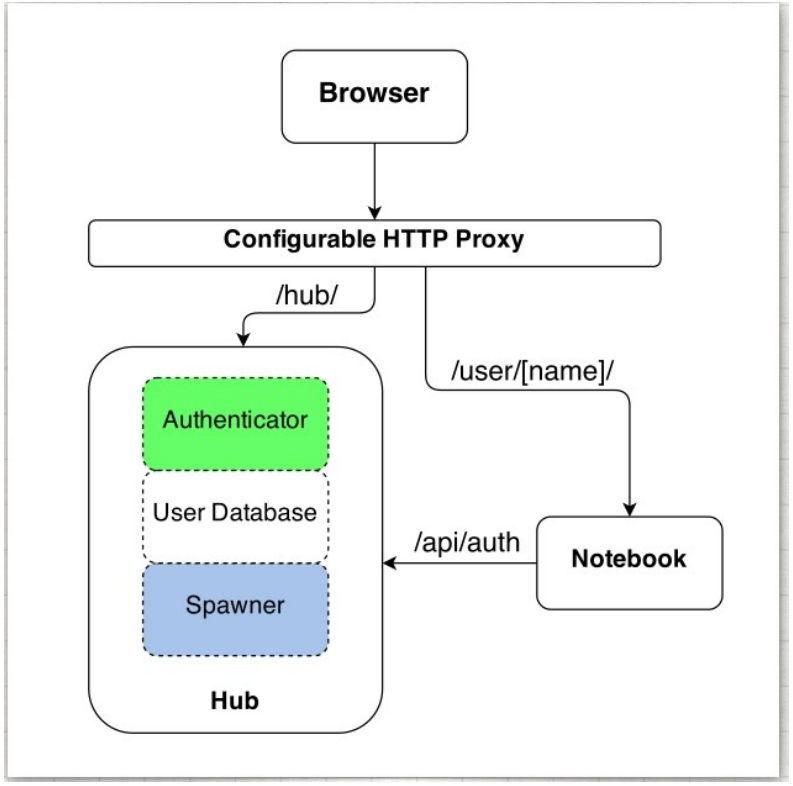
Mahendra PAIPURI
IDRIS-CNRS, Orsay

jupyter

cnrs

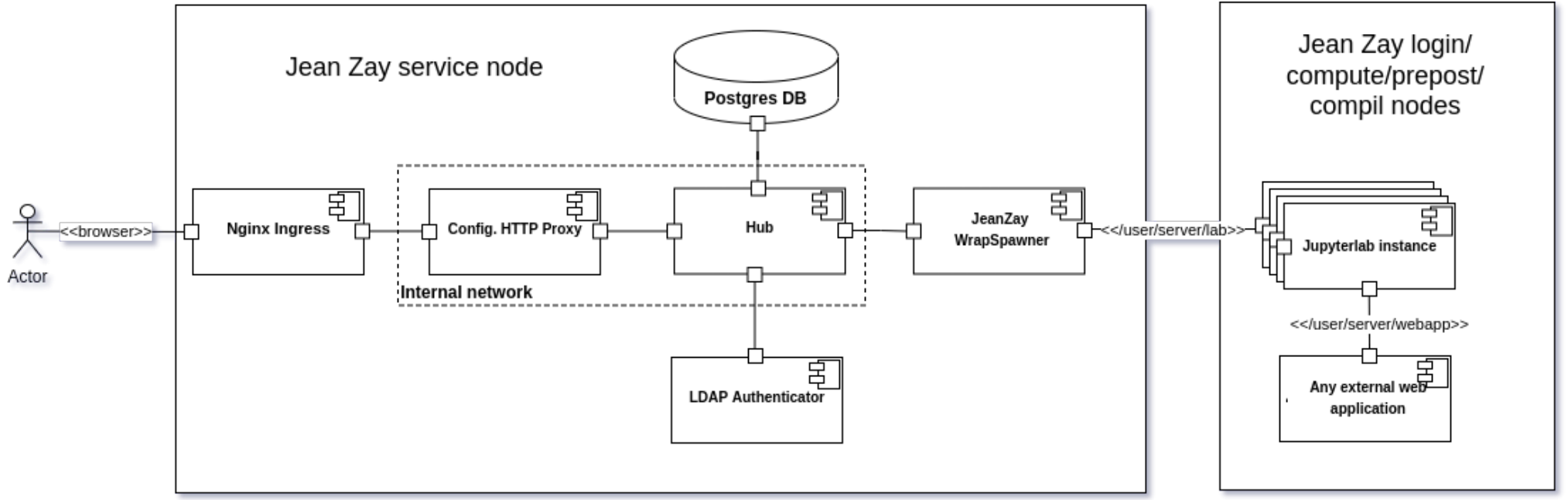
INSTITUT DU DÉVELOPPEMENT ET DES RESSOURCES EN INFORMATIQUE SCIENTIFIQUE

JupyterHub Architecture



- Modular and extensible
- Proxy: CHP and Traefik
- Authenticators: LDAP, OAuth, SAML, Kerberos,…
- Spawners: KubeSpawner, BatchSpawner, DockerSpawner, SystemdSpawner,…
- Can be used to spawn any **arbitrary** web servers **not just JupyterLab and Notebook**

JupyterHub Architecture on JZ



Authenticator

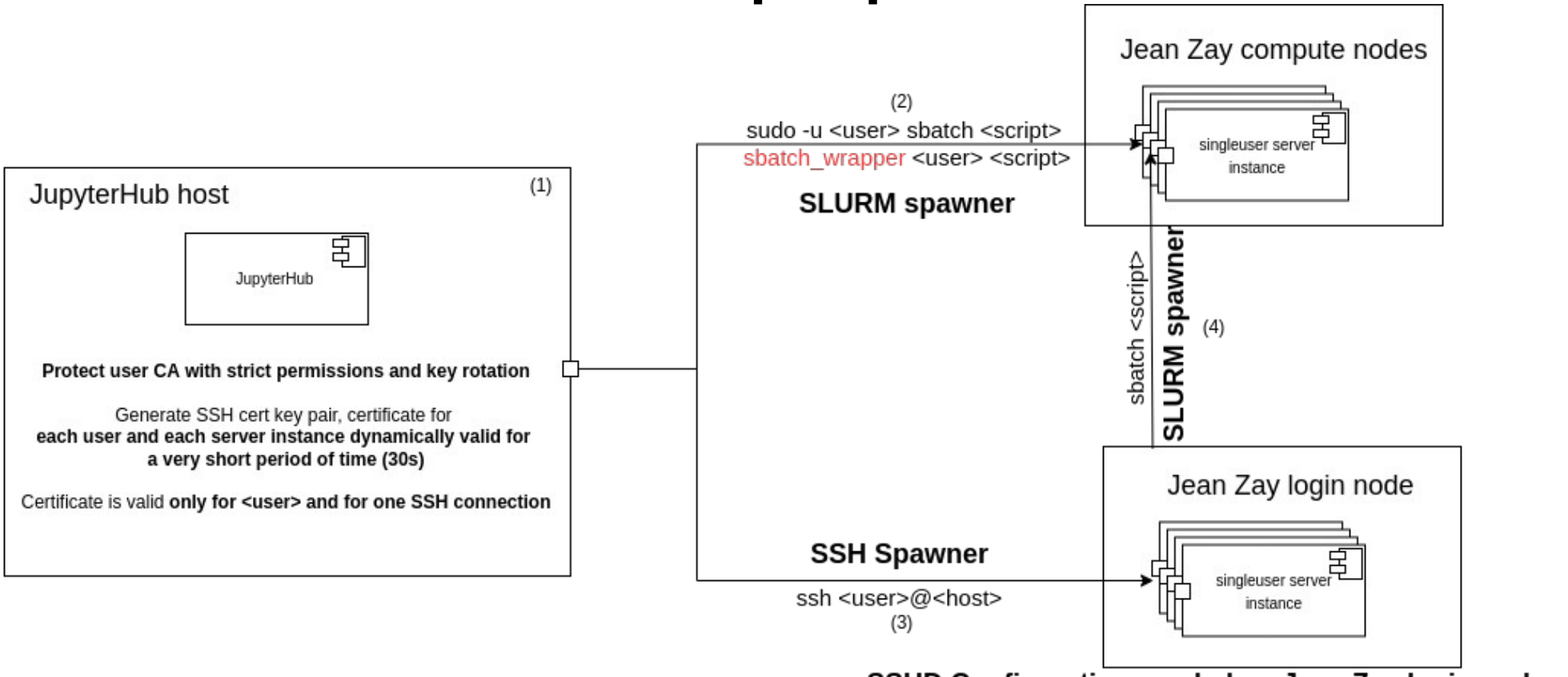
- Custom LDAP authenticator tailored for JZ
- Authenticate and authorize users of JZ
- LDAP bind – Authentication
- Verify Client IP address – Authorization
- User specific data like HOME, WORK and SCRATCH directories, active projects are passed to Spawner
- Eventually move to a SSO solution and use OAuthenticator

Spawner

- Custom WrapSpawner that creates either a SSHSpawner or SlurmSpawner at runtime
- SSHSpawner → Login node,

  SlurmSpawner → Slurm nodes
- Privilege escalation to spawn on behalf of users
  - Using sudo -u <user> <cmd>
  - Using a wrapper with cap_setuid (Linux capabilities)
  - One shot SSH certificates with short validity
- Current deployment supports all three methods

JZ WrapSpawner

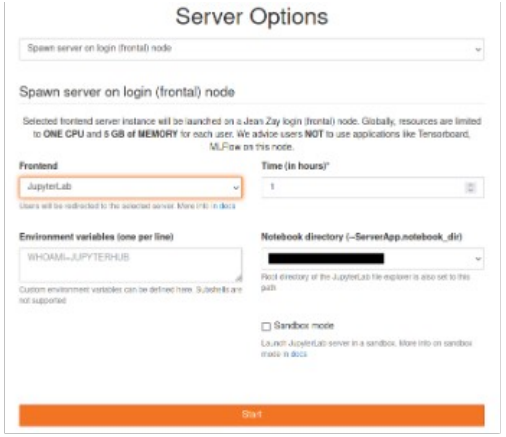


SSHD Configuration needed on Jean Zay login node:

HostCertificate /etc/ssh/ssh_host_rsa_key-cert.pub

TrustedUserCAKeys /etc/ssh/user_ca.pub

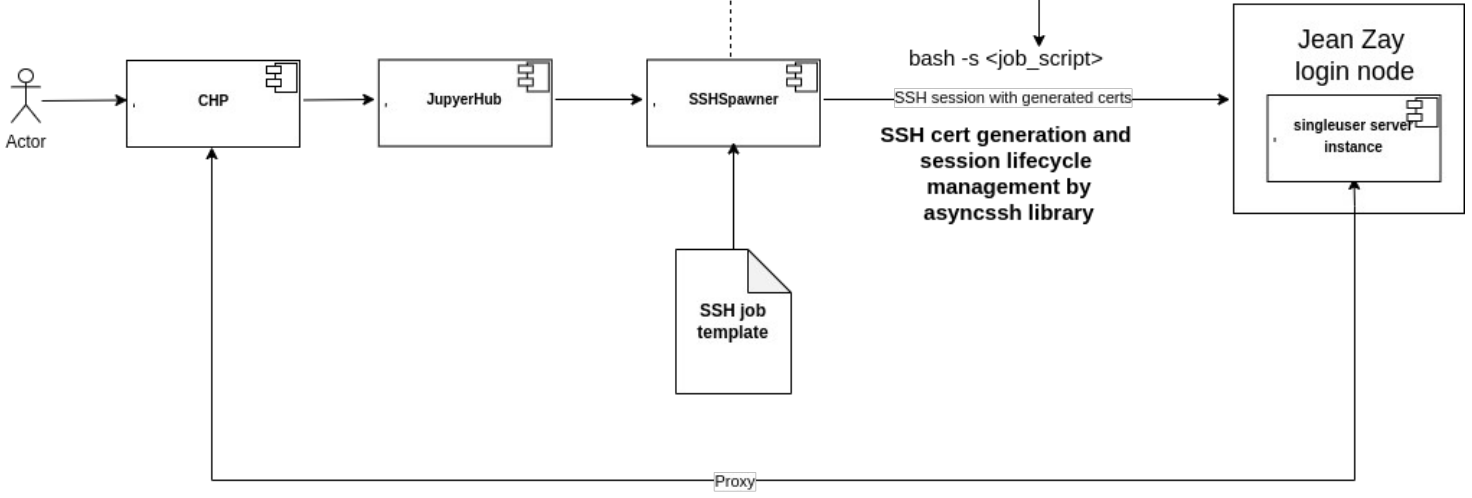
SSHSpawner Workflow



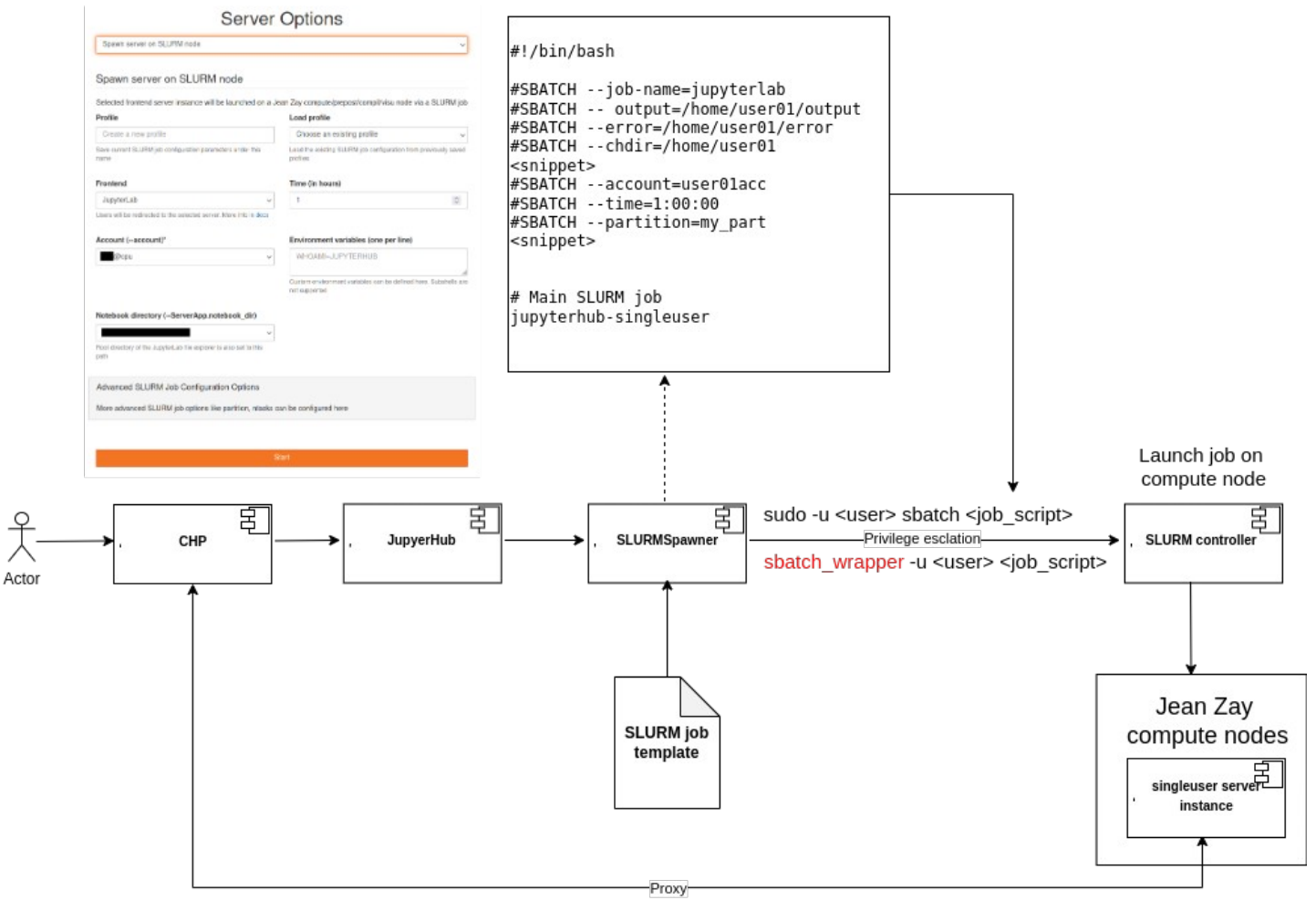

```
#!/bin/bash

<prepare environment>

# Main job
timeout <walltime> jupyterhub-
singleuser &
```



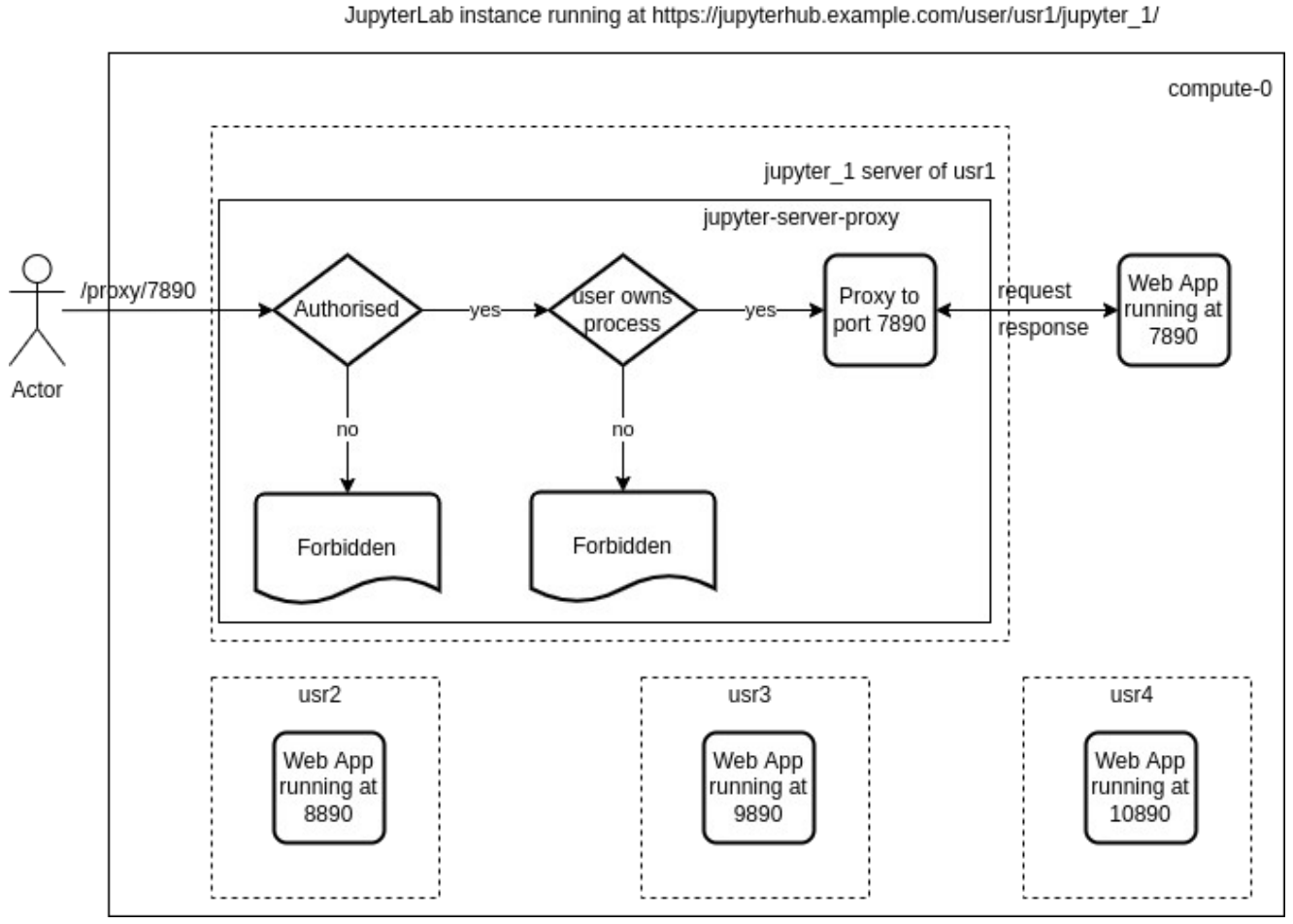
SlurmSpawner Workflow



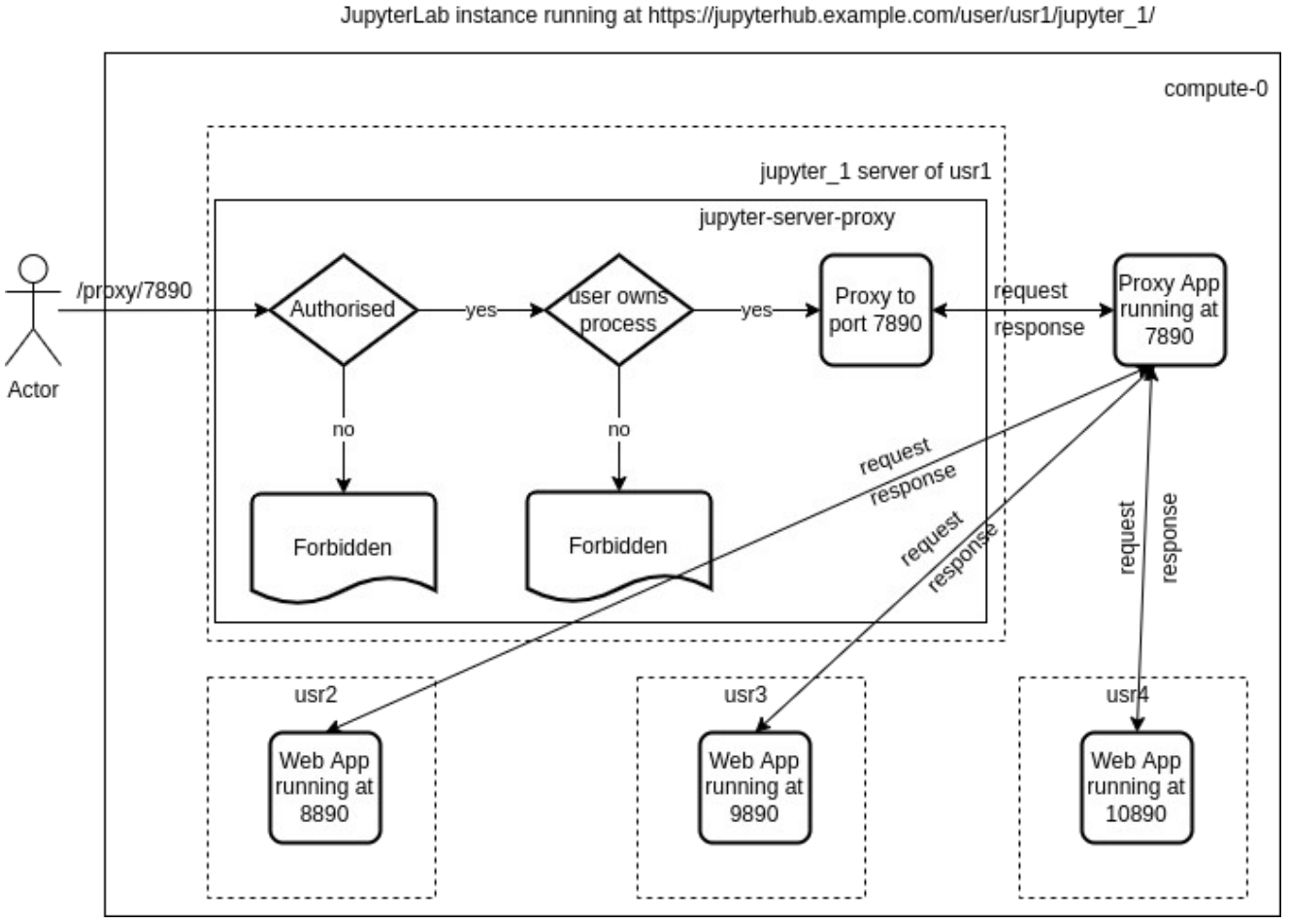
Spawn arbitrary web apps

- Objective is to spawn arbitrary web apps (such as RStudio, Tensorboard, *etc.*) alongside JupyterLab/Notebook and provide authenticated web access to them.
- Use Jupyter server, which is backbone server for both JupyterLab and Notebook, to proxy the requests to the arbitrary web app.
- jupyter-server-proxy accomplishes it by proxying both HTTP and WS traffic. Supports UNIX sockets as well.
- Caveat is jupyter-server-proxy can proxy *absolutely any web app* that is running on *any TCP port.*
- A fork is maintained for Jean Zay JupyterHub deployment that adds important functionalities like life cycle management of web apps, security checks, *etc.*

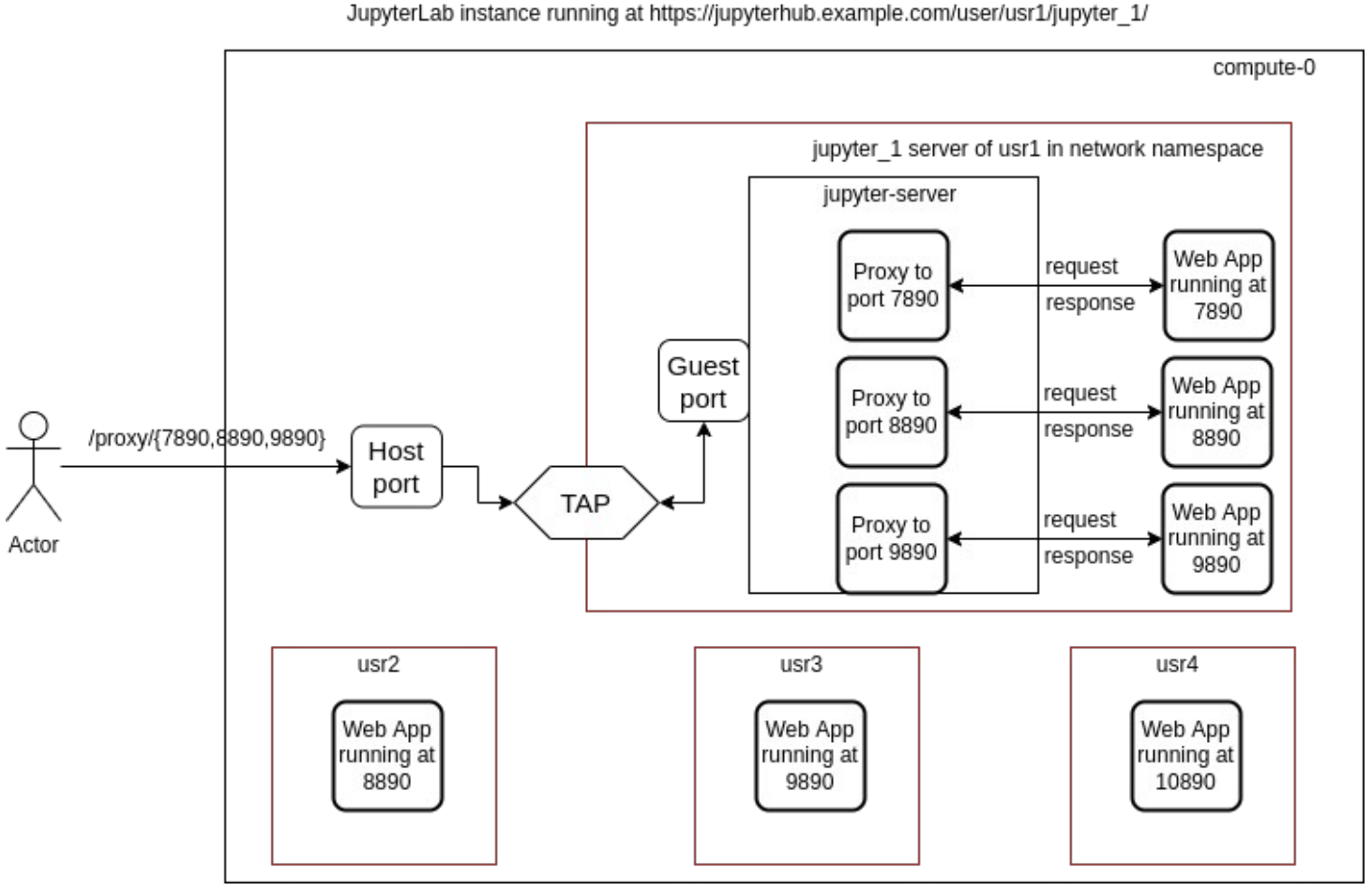
Spawn arbitrary web apps



Spawn arbitrary web apps



Spawn arbitrary web apps



bubblewrap and slirp4netns are used to create network ns and TAP device

JupyterLab extensions

- nb-jeanzay-conda-kernels discover kernels in Jean Zay environment modules automatically and make them available to users *via* JupyterLab.
- Real time CPU and GPU energy usage and CO_2 emissions.
  eCO2mix from RTE is used to estimate CO_2 emissions.
- Users can interact with environment modules from JupyterLab.
- Several web apps like VSCode, Tensorboard, MLFlow, noVNC Desktop, Cylc UI, NerfStudio are supported.
- Dask and Ray dashboards are supported.
- Launcher is customized for Jean Zay.

Deployment details

- Deployment *via* Ansible playbook.
- A playbook is being maintained and tested for Ubuntu 22, Debian 11, CentOS 8 and Rocky 8 in CI.
- Jupyter{Hub,Lab} stack is installed within conda environment on a network file system.
- PostgreSQL DB is used.
- Run JupyterHub and CHP separately and use systemd for supervision.
- Hardened nginx systemd unit to partially “containerize” the process.
- Monitoring stack based on Prometheus, Grafana Loki and Grafana.
- Migrate to Traefik proxy as there is a known memory leak in CHP.
- Idle server culler service to terminate inactive servers.

Demo time