

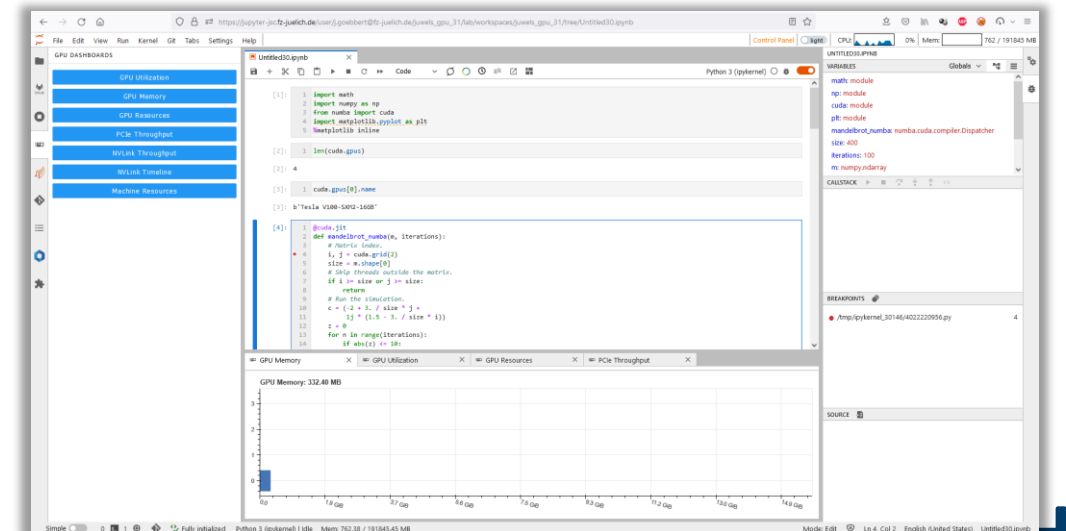
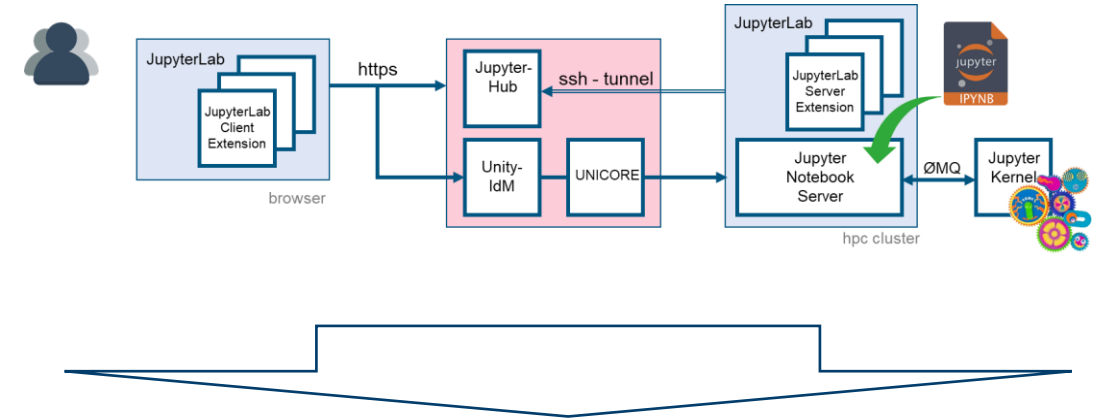


INTERACTIVE HPC WITH JUPYTERLAB

EDIH (European-Data-Innovation-Hub) -Workshop, Part 3

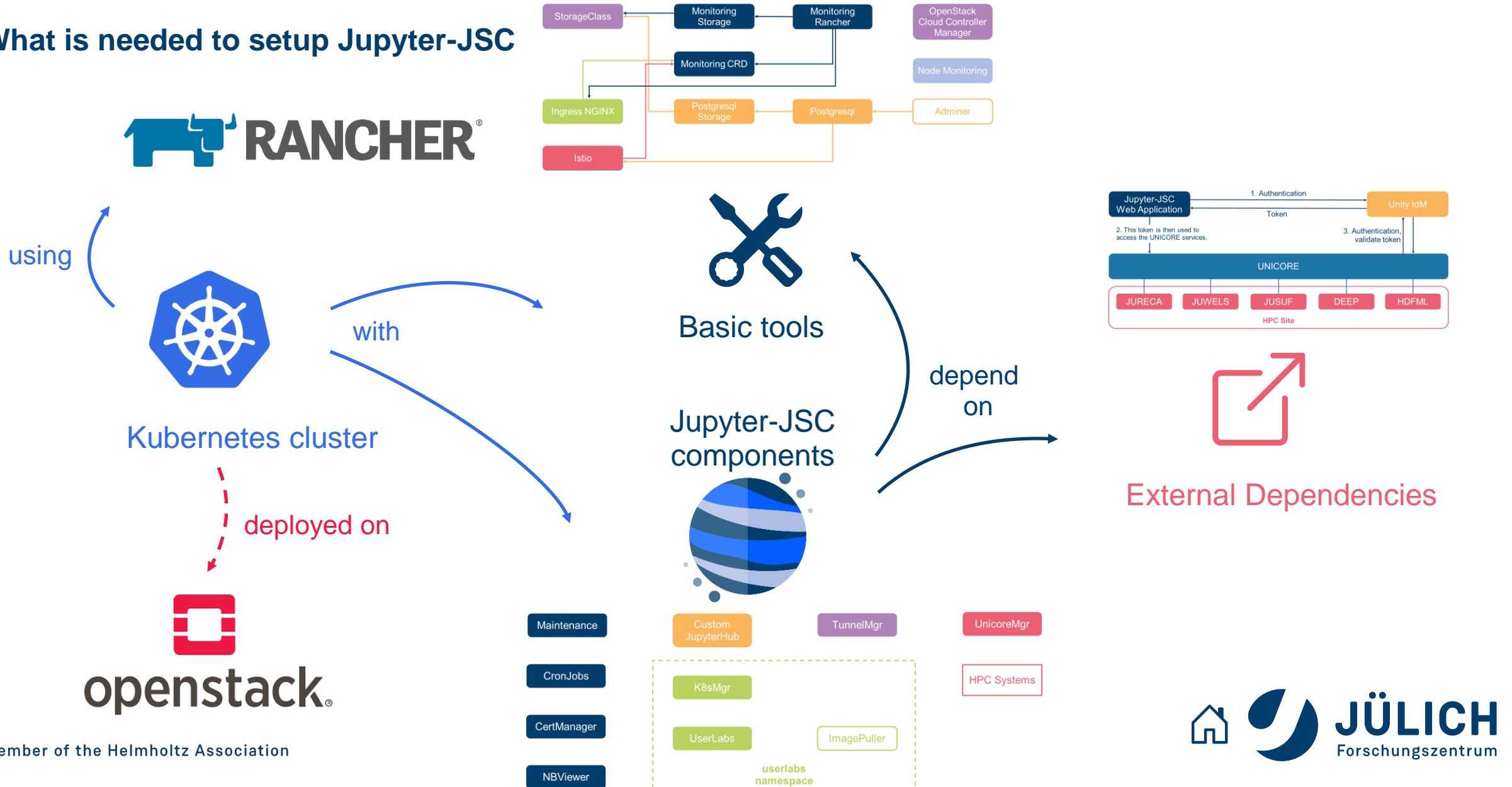
2023-10-23 | JENS HENRIK GÖBBERT (J.GOEBBERT@FZ-JUELICH.DE)

Start your JupyterLab



OVERVIEW

What is needed to setup Jupyter-JSC



CUSTOM JUPYTERHUB COMPONENTS

Jupyter-JSC JupyterHub

Authenticator



Authorizes users to use the Hub and single user notebook servers. Saves HPC access related information in the user's auth state. Keeps the Hub's configuration up to date.



HTML Templates

Custom templates provide complete control over JupyterHub's appearance.



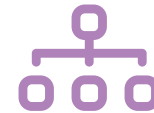
Endpoints

- Enables 2FA
- Spawn progress updates

Configuration



Contains information about DRF services, Unity, UNICORE, HPC systems, allowed JupyterLab configurations per user group (VO), etc. Synchronized via git outside of JupyterHub.



Spawner

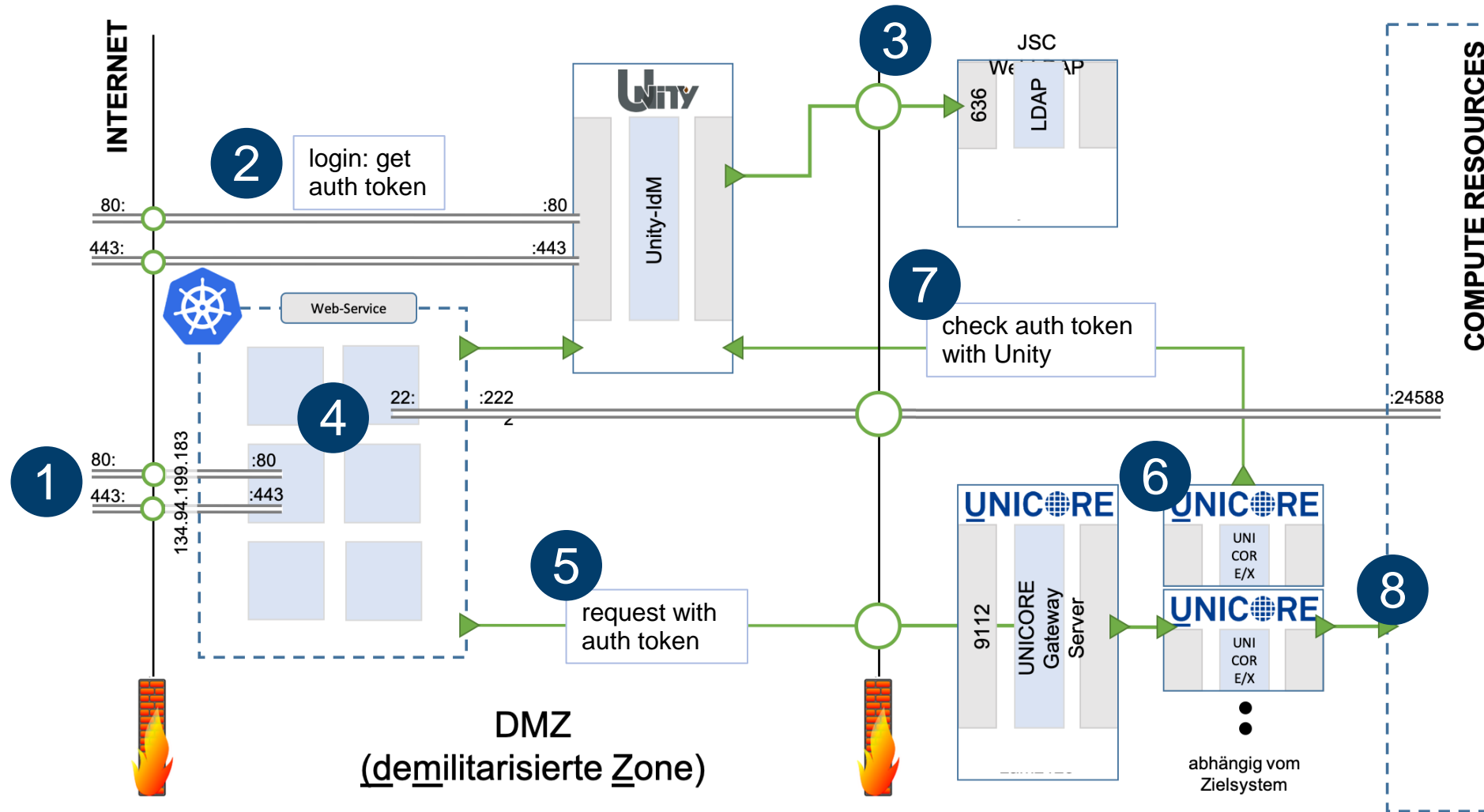
Starts each single-user notebook server. Needs to be able to start, poll and stop the servers.



Patches

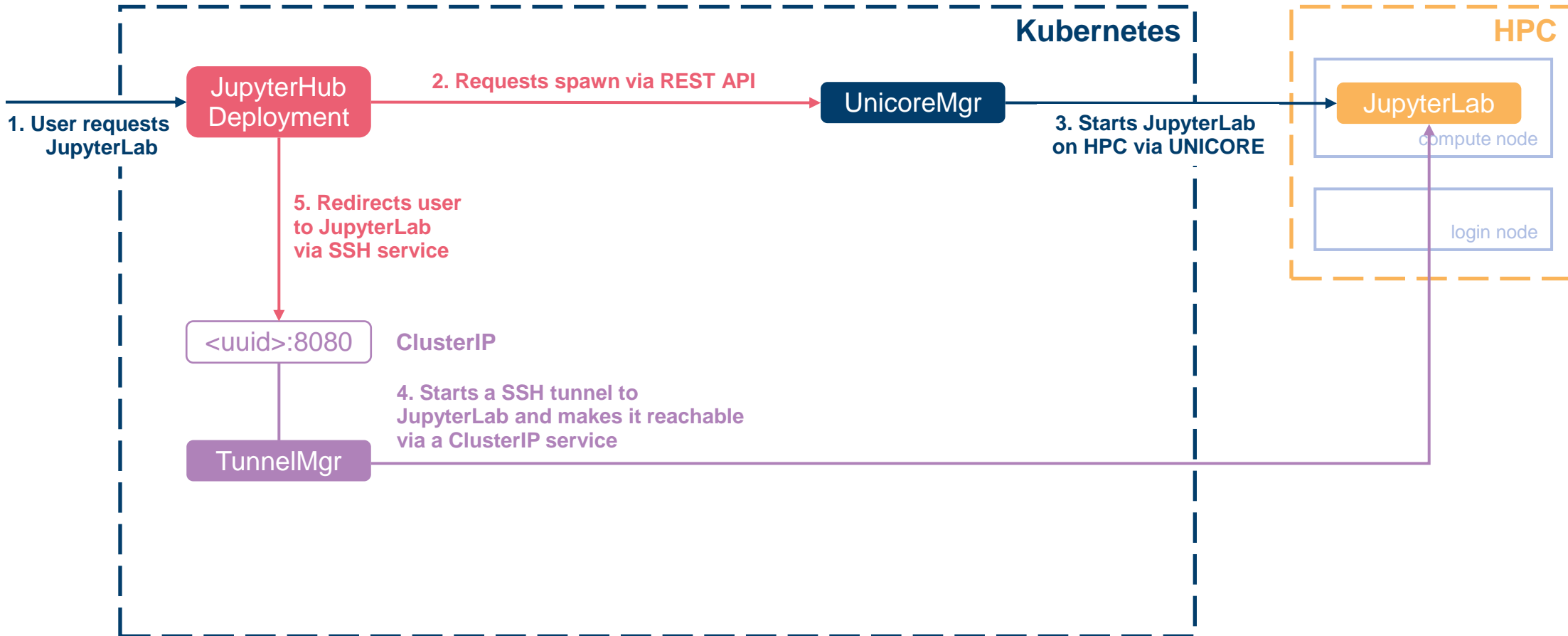
Adapt “standard” JupyterHub components to handle customizations.

JUPYTER-JSC ARCHITECTURE – SECURITY



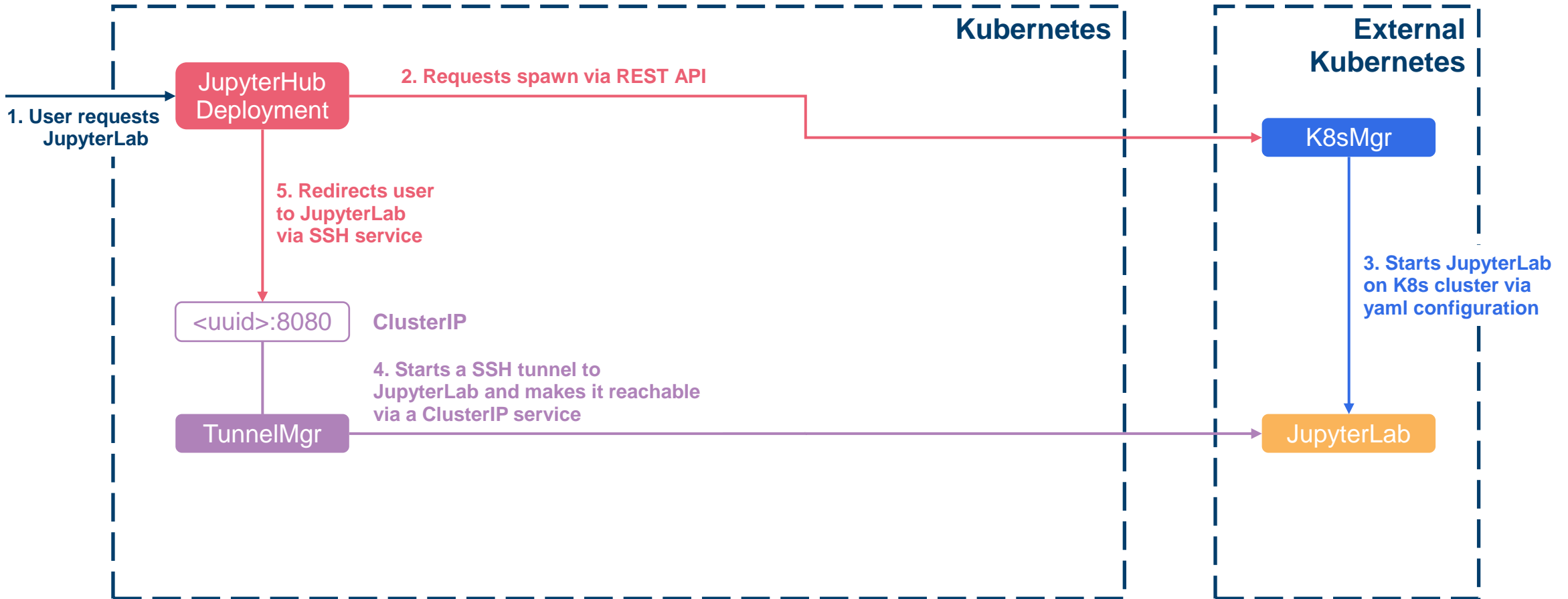
STARTING A JUPYTERLAB – HIGH LEVEL VIEW HPC

Jupyter-JSC



STARTING A JUPYTERLAB – HIGH LEVEL VIEW K8S

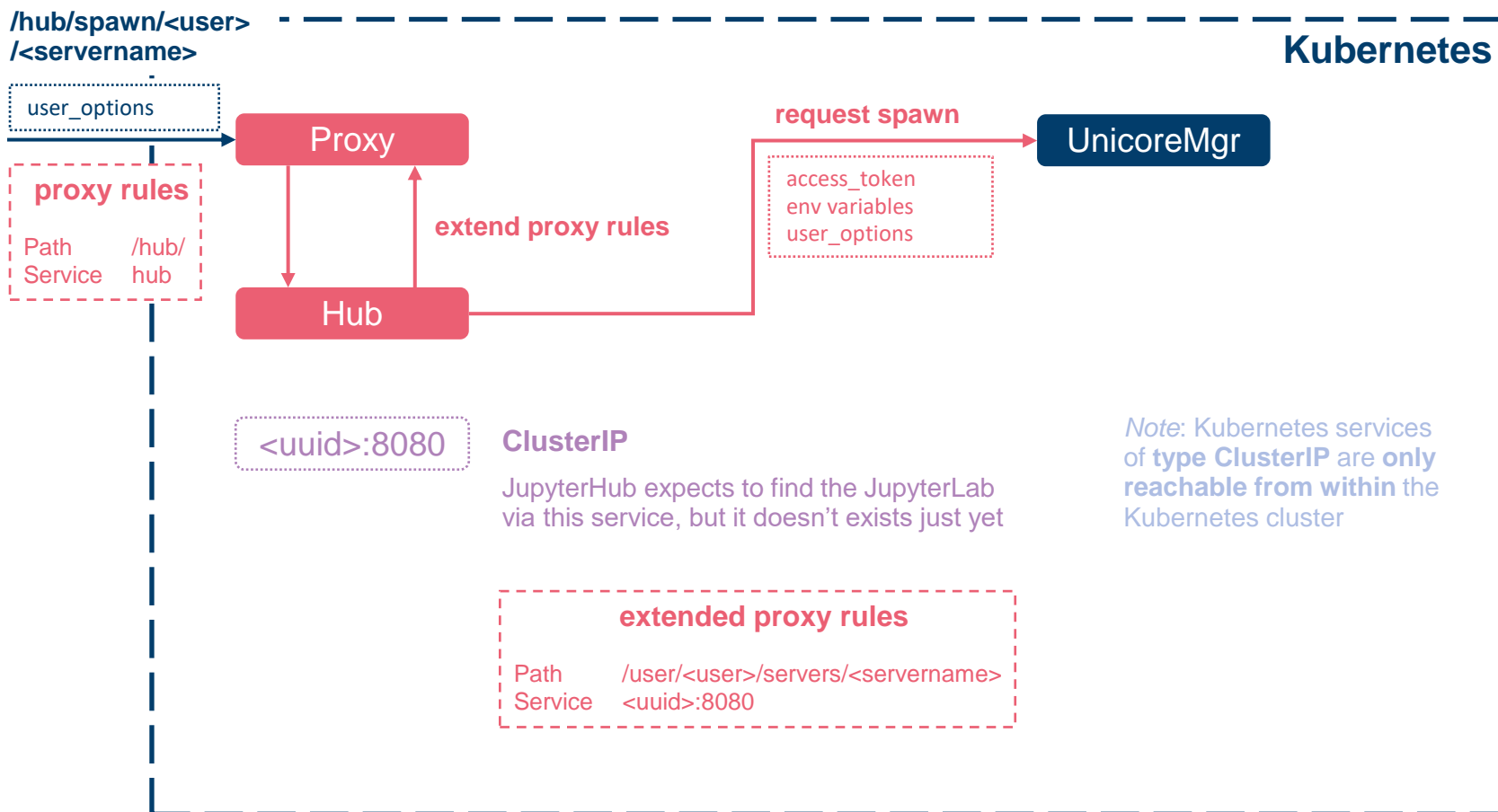
Jupyter-JSC



STARTING A JUPYTERLAB (HPC)

Jupyter-JSC - /hub/spawn/<user>/<servername>

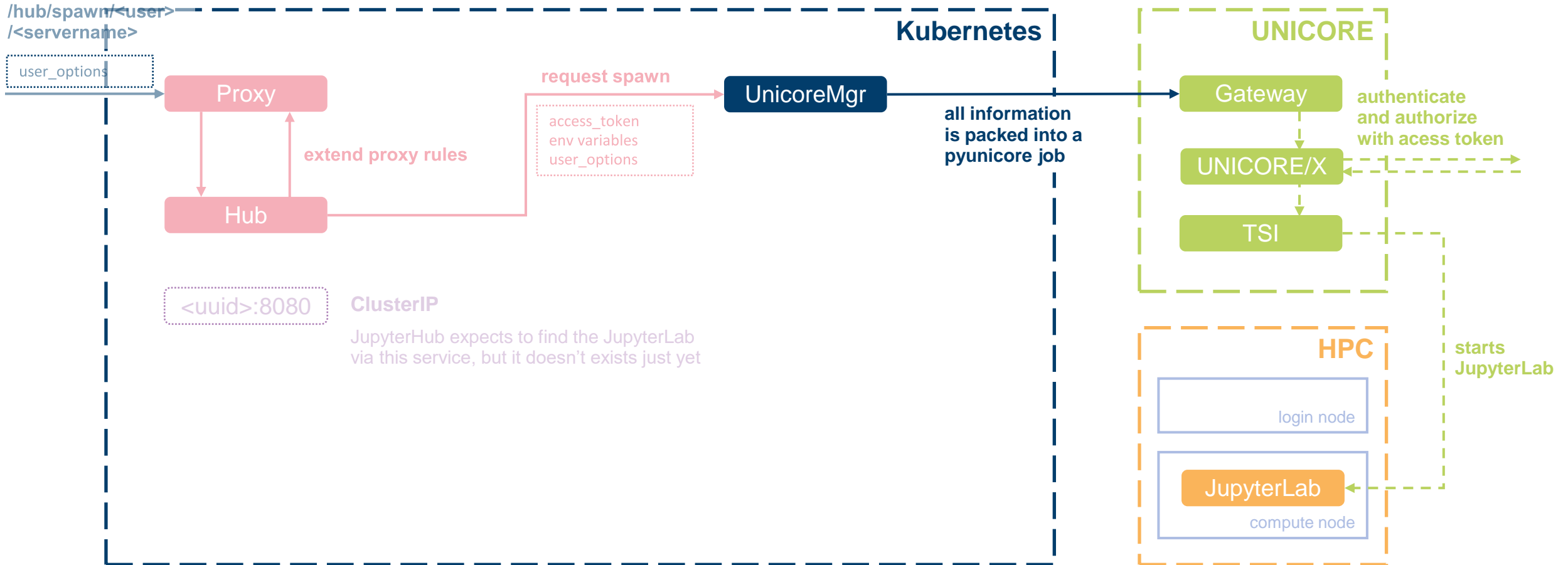
—— POST request
- - - - other communication



STARTING A JUPYTERLAB (HPC)

Jupyter-JSC - /hub/spawn/<user>/<servername>

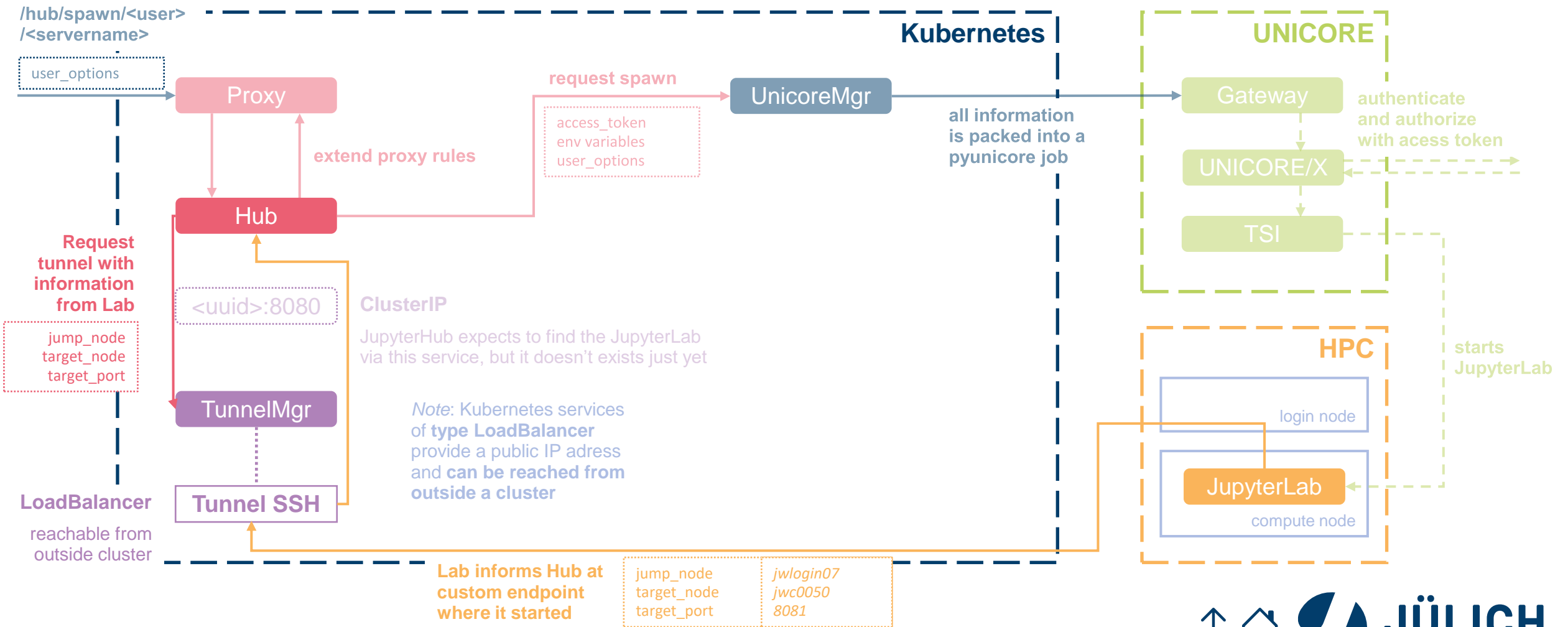
—— POST request
- - - other communication



STARTING A JUPYTERLAB (HPC)

Jupyter-JSC - /hub/spawn/<user>/<servername>

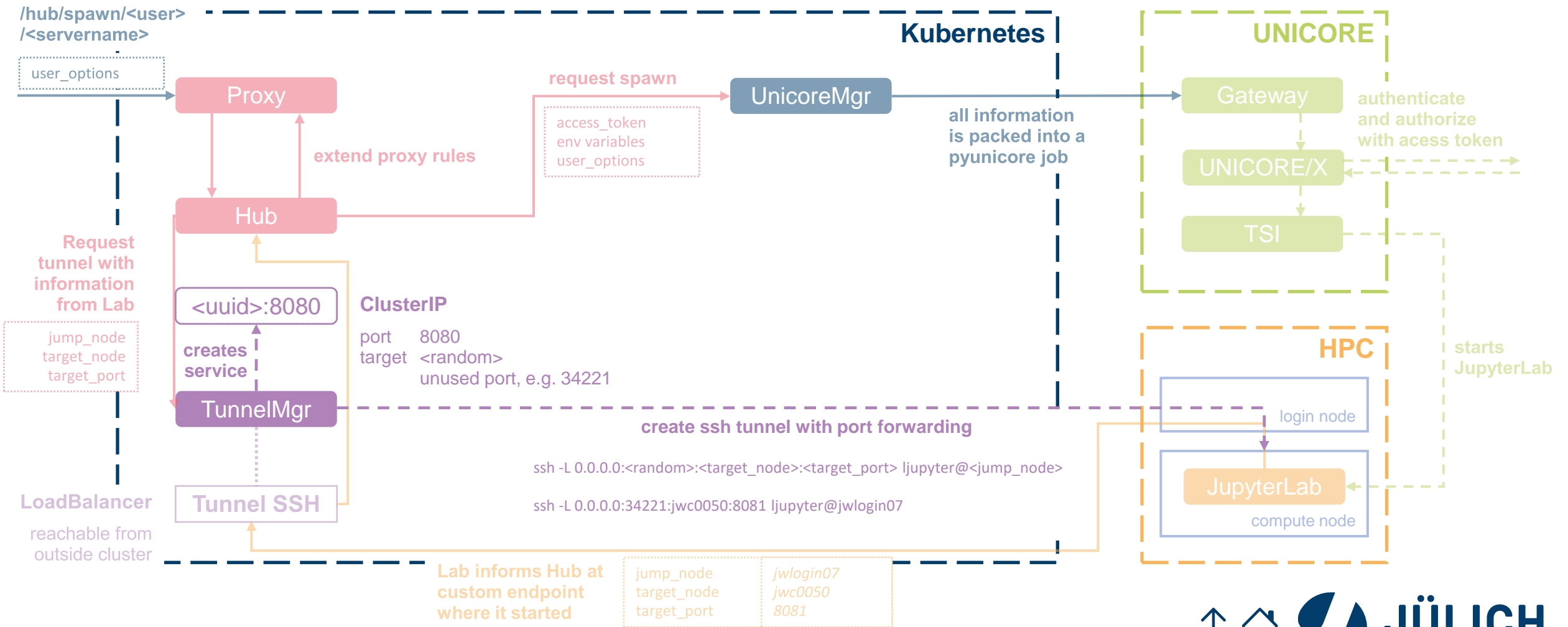
—— POST request
- - - other communication



STARTING A JUPYTERLAB (HPC)

Jupyter-JSC - /hub/spawn/<user>/<servername>

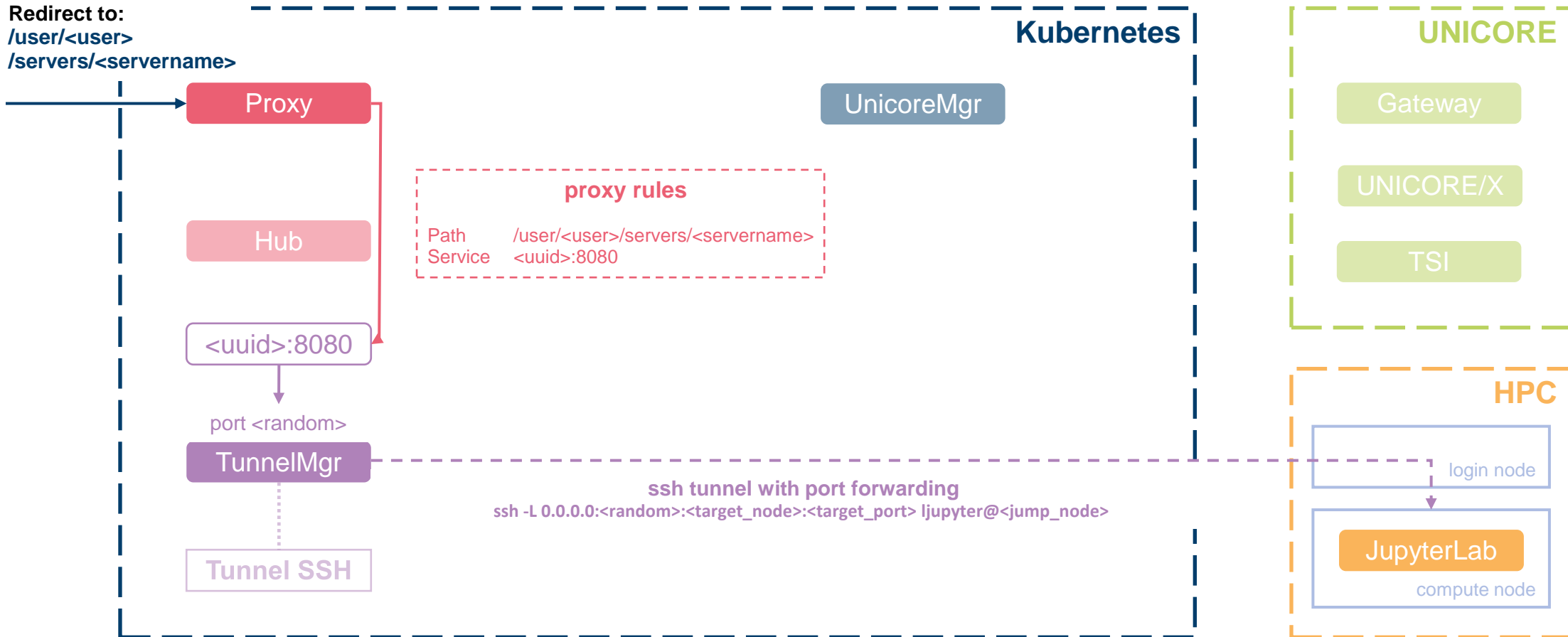
—— POST request
- - - other communication



STARTING A JUPYTERLAB (HPC)

Jupyter-JSC - /user/<user>/servers/<servername>

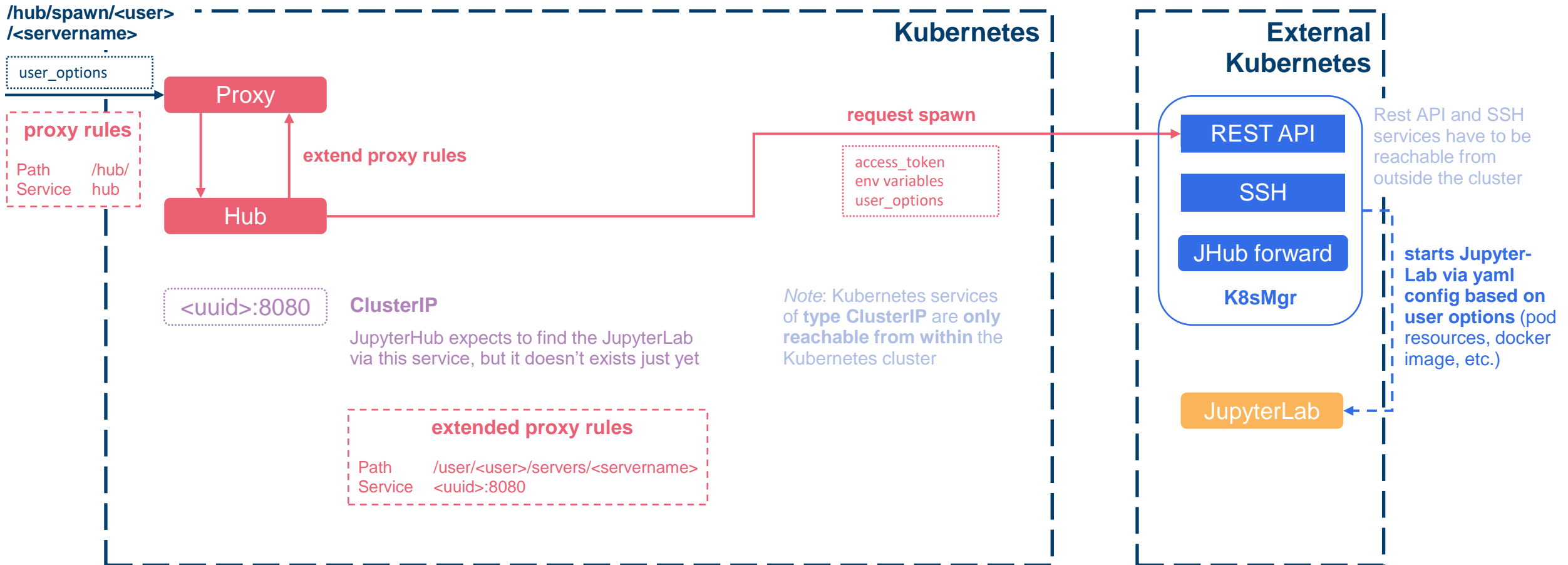
—— POST request
---- other communication



STARTING A JUPYTERLAB (KUBERNETES)

Jupyter-JSC - /hub/**spawn**/**<user>**/**<servername>**

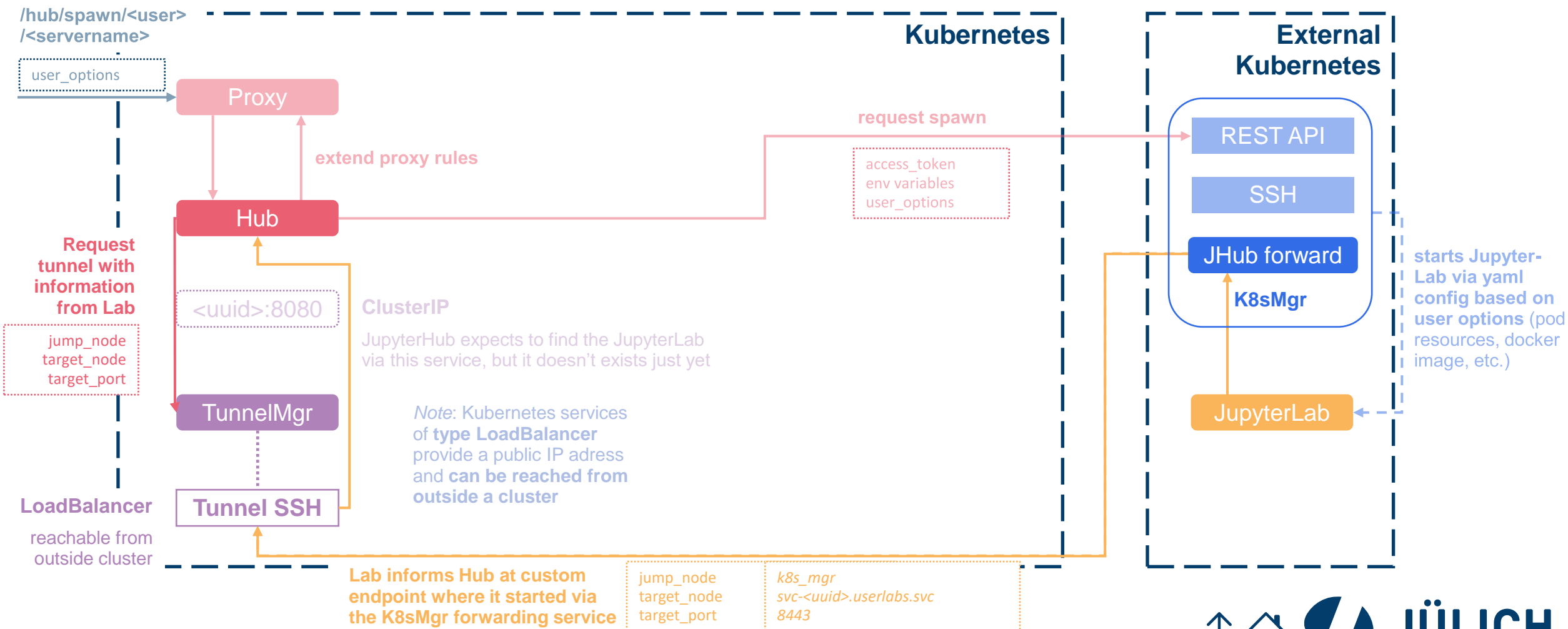
—— POST request
---- other communication



STARTING A JUPYTERLAB (KUBERNETES)

Jupyter-JSC - /hub/spawn/<user>/<servername>

—— POST request
---- other communication



— POST request
- - - other communication

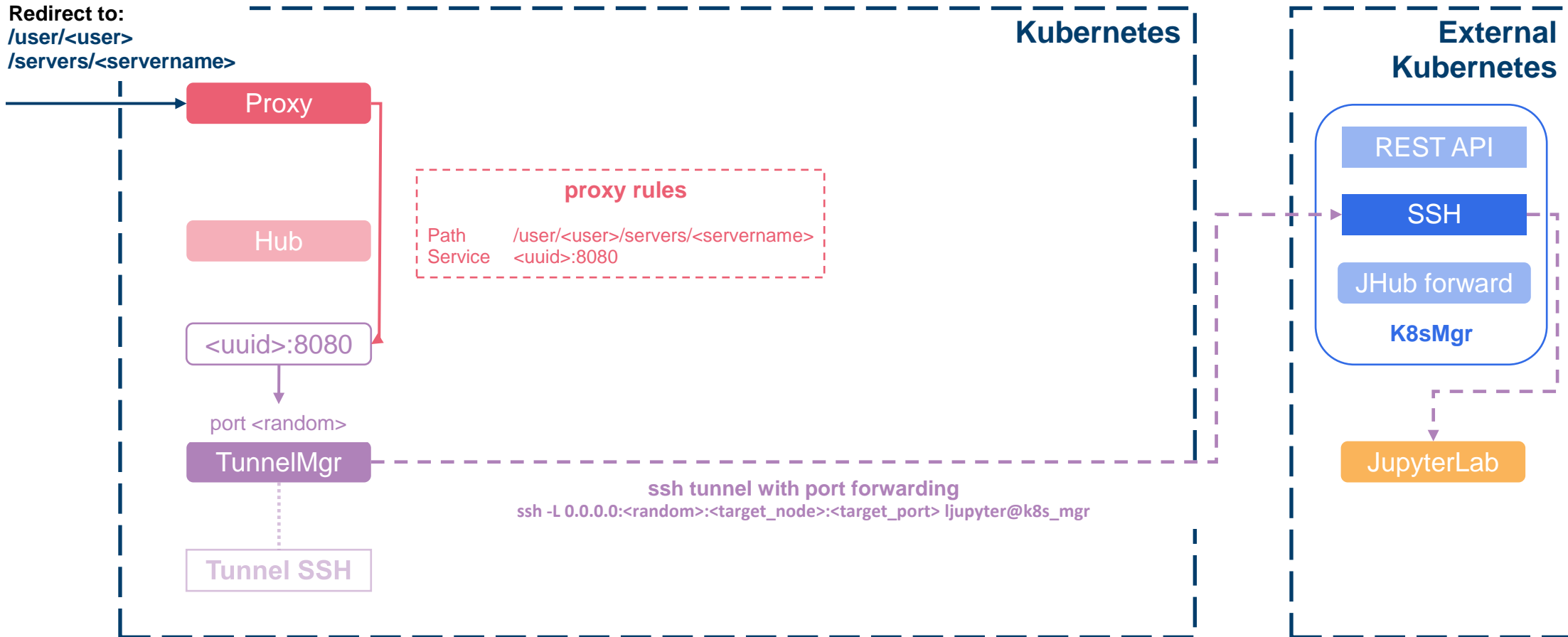
```
/hub/spawn/<user>  
/<servername>
```



STARTING A JUPYTERLAB (KUBERNETES)

Jupyter-JSC - /**user**/**<user>**/servers/**<servername>**

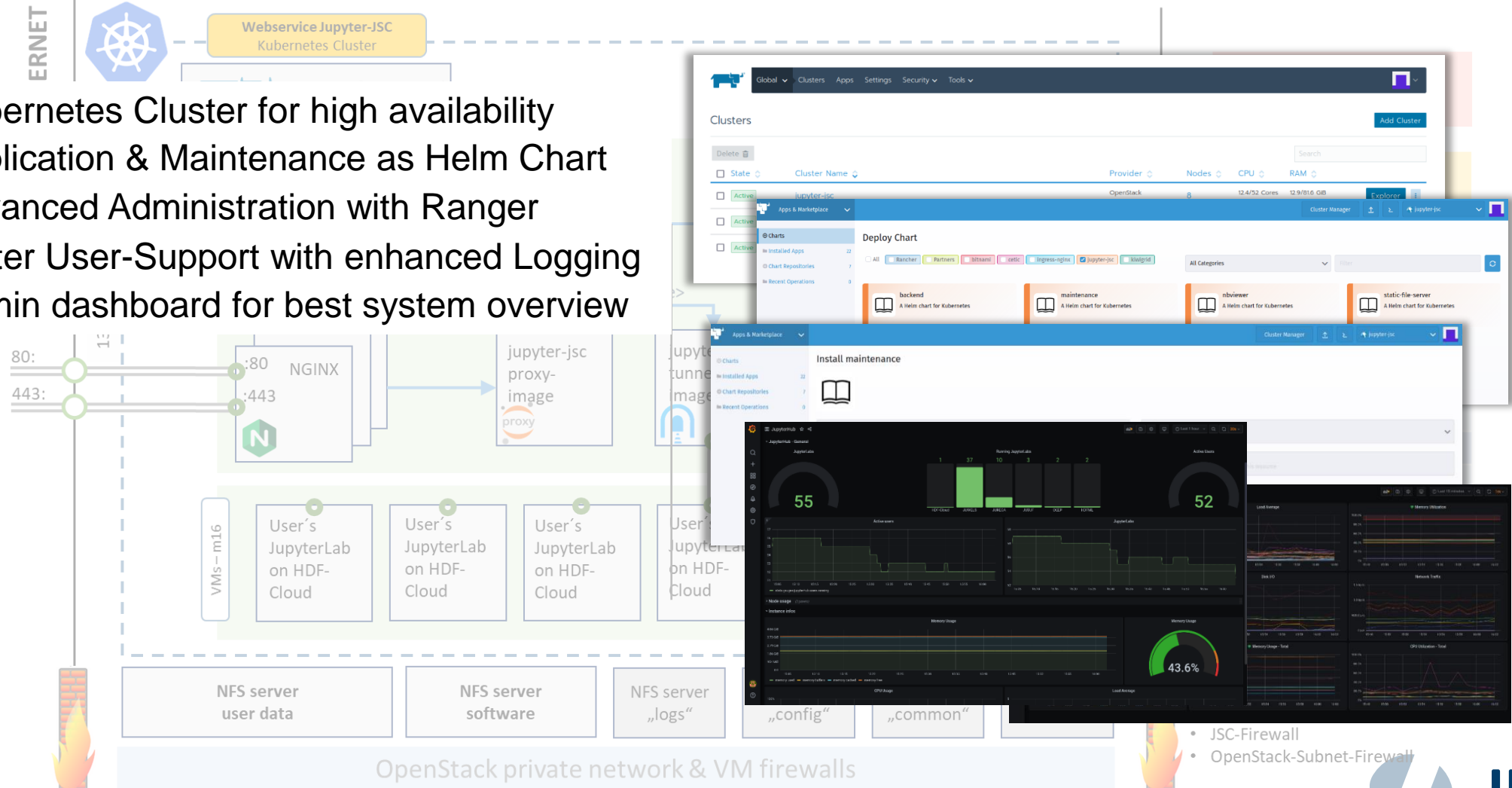
—— POST request
- - - other communication



JUPYTER HUB WEBSERVICE

On the example of <https://jupyter-jsc.fz-juelich.de>

- Kubernetes Cluster for high availability
- Application & Maintenance as Helm Chart
- Advanced Administration with Ranger
- Better User-Support with enhanced Logging
- Admin dashboard for best system overview



JUPYTERLAB EVERYWHERE

Community, Project and System Entrypoints

The flexible software design allows one backend for multiple frontends. Do not reinvent the wheel!

1. Specific URL
 - <https://juniq.fz-juelich.de> (in production)
 - <https://coec-project.eu> (in production)
 - <https://jupyter.eurocc-gcs.de> (coming soon)
 - <https://fenix-ri.eu> (brainstorming)
 - ...
2. Specific branding
 - HTML/CSS design (easy updates through git)
 - JupyterLab Settings
3. Specific configurations
 - Systems, Partitions, Projects, etc.
 - E.g. Entering through <https://juniq.fz-juelich.de> will give only access to JUNIQ projects/resources
4. Accessing/integration of other resources
 - Dedicated resources
 - Sharing/trusting IdMs
 - External resources (Kubernetes manager)

