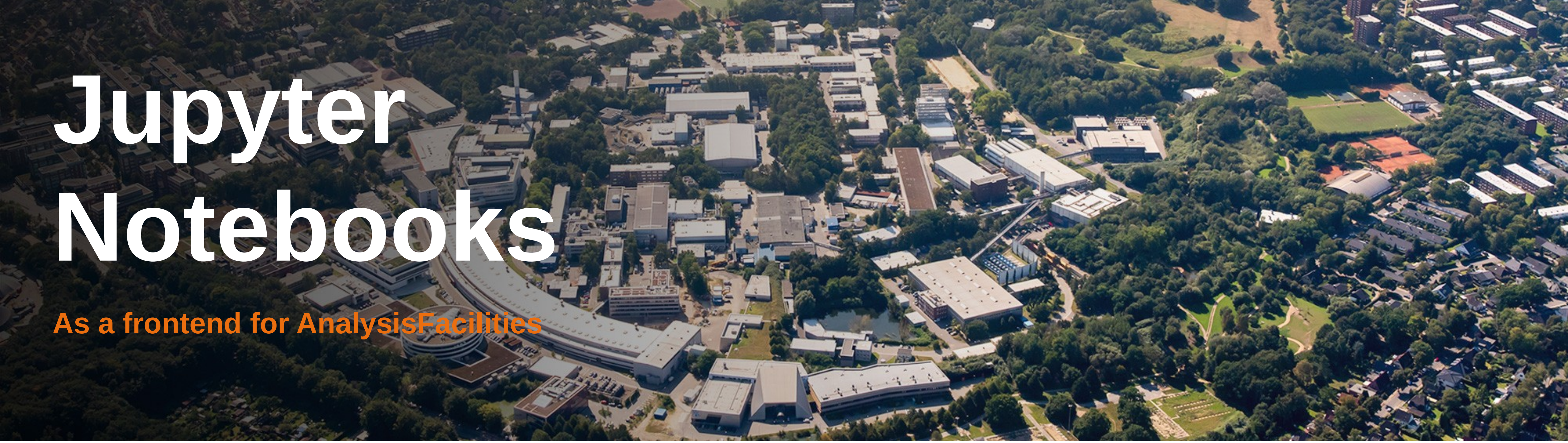
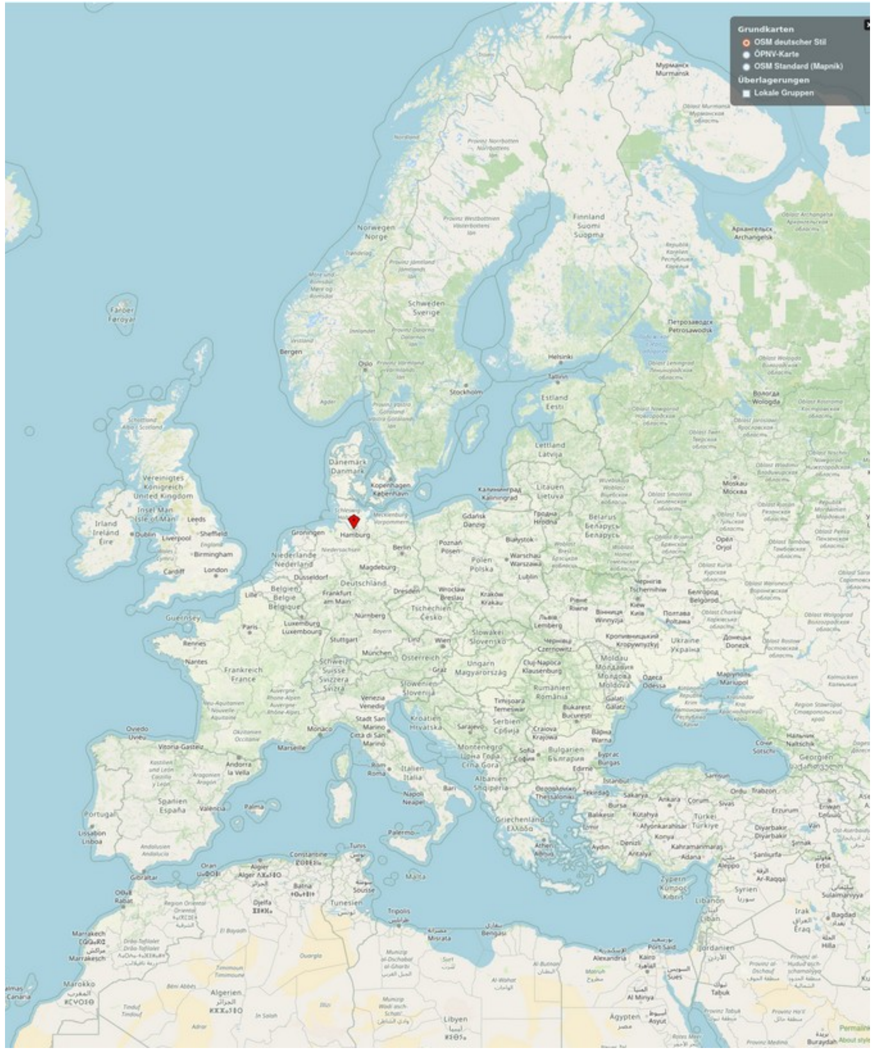


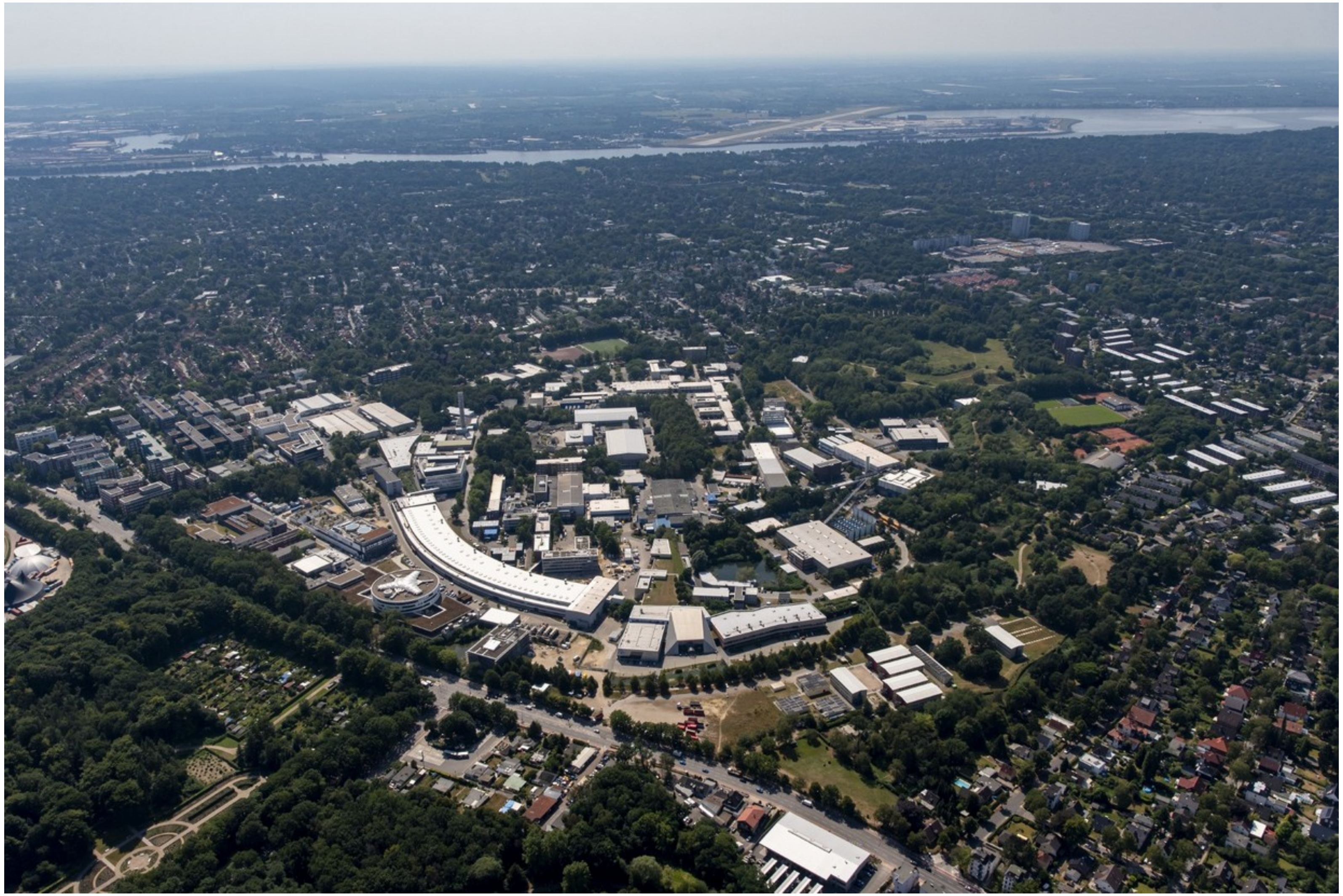
Jupyter Notebooks

As a frontend for AnalysisFacilities

Beyer, Christoph
Madison, 12-07-2024

Welcome to DESY





HELMHOLTZ

DESY at a glance

A research center of the Helmholtz Association

Deutsches **E**lektronen-**Sy**nchrotron
Founded in 1959 as a German Particle Physics Laboratory

More than 2.800 employees, including 1.100 researchers (31 % women)
More than 3.000 visiting researchers from over 40 countries annually

232 million € Basic budget 2020
90% from the Federal Government (BMBF) and 10% from the City of Hamburg/State of Brandenburg

Two sites

DESY at a glance

A research center of the Helmholtz Association

Two sites

DESY at a glance

A research center of the Helmholtz Association

Two sites

Campus Hamburg

DESY at a glance

A research center of the Helmholtz Association

Two sites

Campus Hamburg

Campus Zeuthen (Brandenburg)

DESYs research areas

The cornerstones of DESY's mission

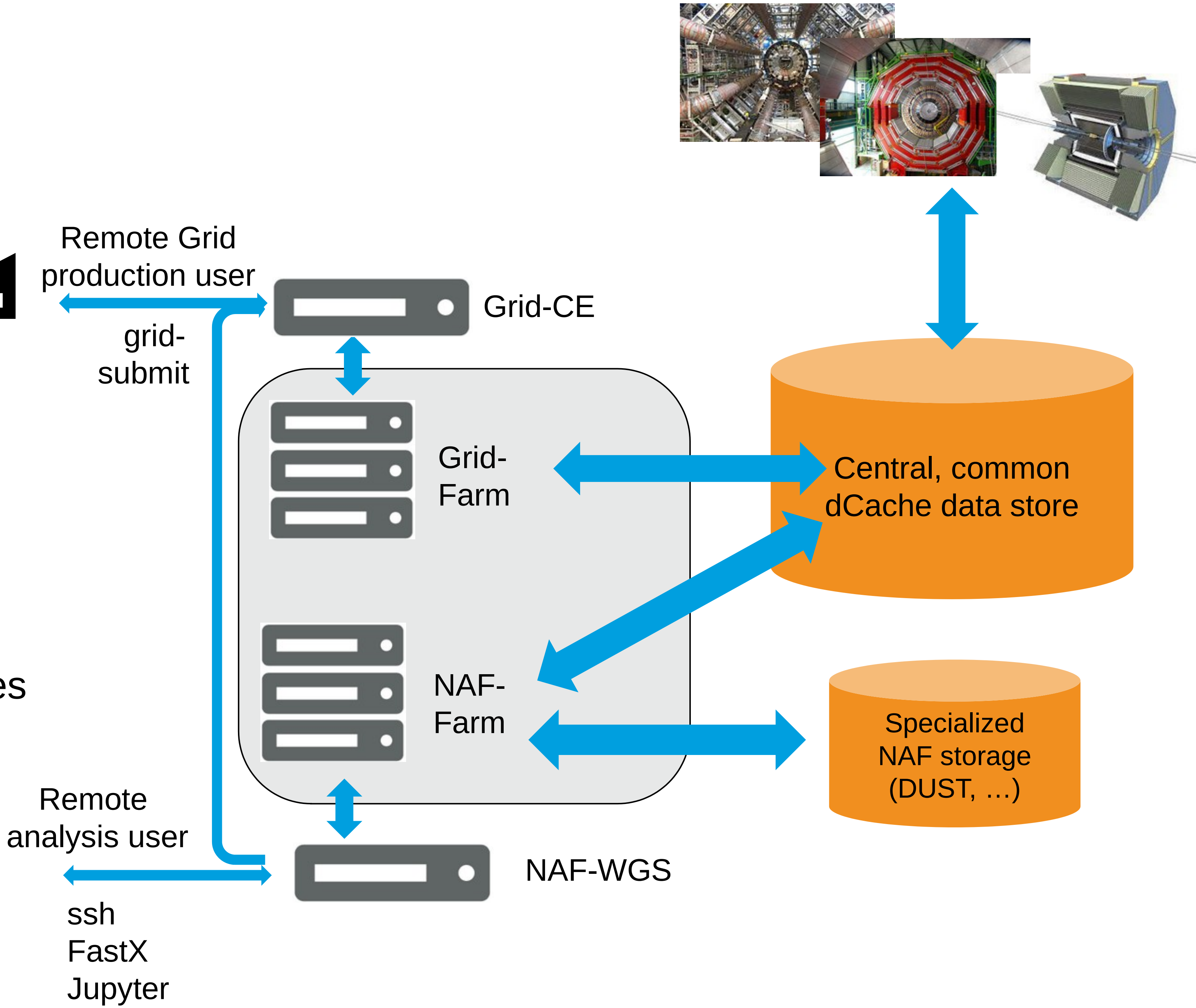
GRID & NAF

Two HTC AFs

Grid: Serves worldwide HEP community through Grid protocols

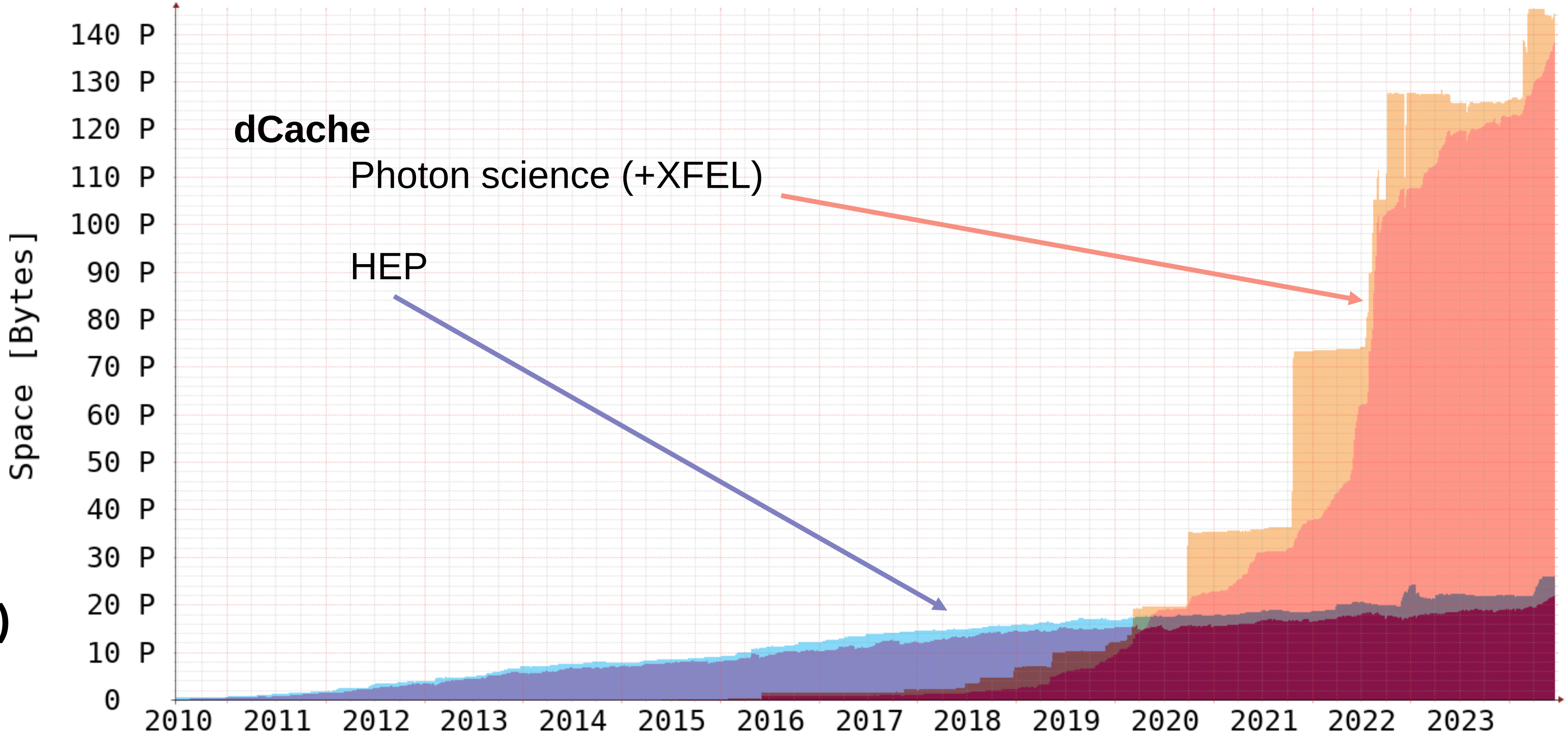
NAF: Serves national HEP community through interactive protocols: Fast turn-around

Access protocol is just one/few boxes large compute behind, as well as storage infrastructure and access is (mostly) identical


Remote Grid production user
grid-submit
Grid-CE
Grid-Farm
NAF-Farm
Central, common dCache data store
Specialized NAF storage (DUST, …)
Remote analysis user
NAF-WGS
ssh
FastX
Jupyter

Everything is pretty much data driven

- **HPC + Grid + NAF**
- **~180 PB data on disk**
- **dCache + GPFS + BeeGFS**
- **~60.000 CPU cores, ~380 GPUs**
- **HTCondor, SLURM**
- **~2.700 server (compute, storage, management)**
- **~ >0.8 Megawatt (out of 1.3 MW)**



Jupyter Lab & Jupyter Notebooks

Jupyter Lab

- is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner.

Jupyter Notebooks

- are a community standard for communicating and performing interactive computing. They are a document that blends computations, outputs, explanatory text, mathematics, images, and rich media representations of objects.
- JupyterLab is one interface used to create and interact with Jupyter Notebooks.

Jupyter Notebooks as a frontend to a HTC AF

Why do user possibly want that ?

- Jupyter Notebooks are no ‘batch-tools’ as such and can run pretty much everywhere

  (local laptop, cloud, …) running them on an AF is interesting from some perspectives

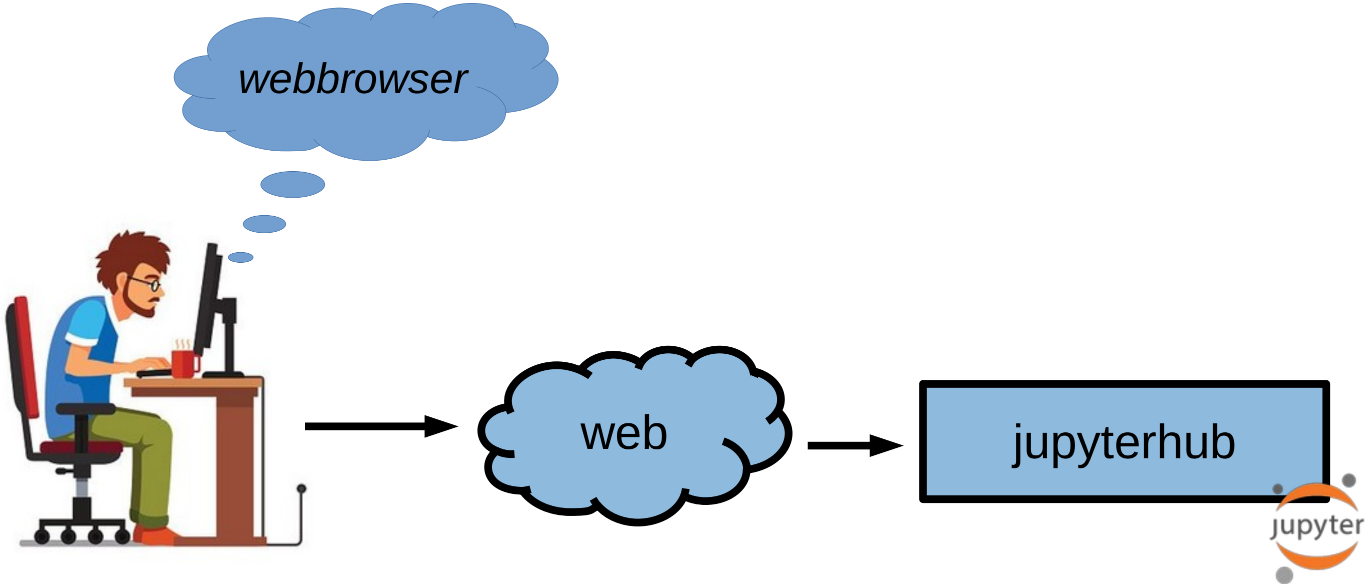
  - The JHUB running the webinterface does bridge the AF access into the WWW in a controlled and save manner (admin interest)
  - It might have access to bigger/different resources than locally available (e.g. cores, RAM, GPU)
  - There is a predefined software stack well looked after by the VO admins readily available (e.g in CVMFS)
  - The environment (mounts, Paths, software versions) is steady and defined by the VO admins
  - Authentication and authorization is managed by the JHUB including all tickets, tokens etc. needed
  - Storage access is managed and fast (well – hopefully, most of the times, faster than ‘at home’ in any case)
  - Very popular with traveling scientists who e.g. have 3 days of beamtime at PETRAIII
  - Very popular with younger people in general – sighhh

Jupyter Notebooks in NAF

Setup standard using batchspawner

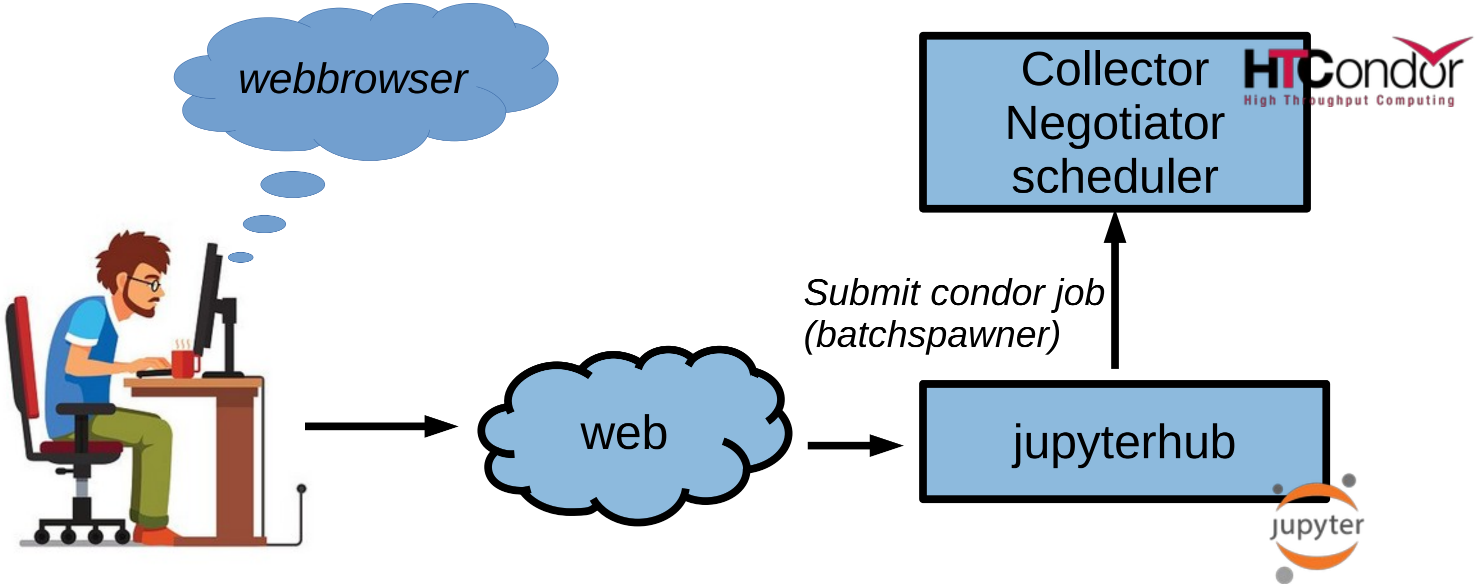
Jupyter Notebooks in NAF

Setup standard using batchspawner



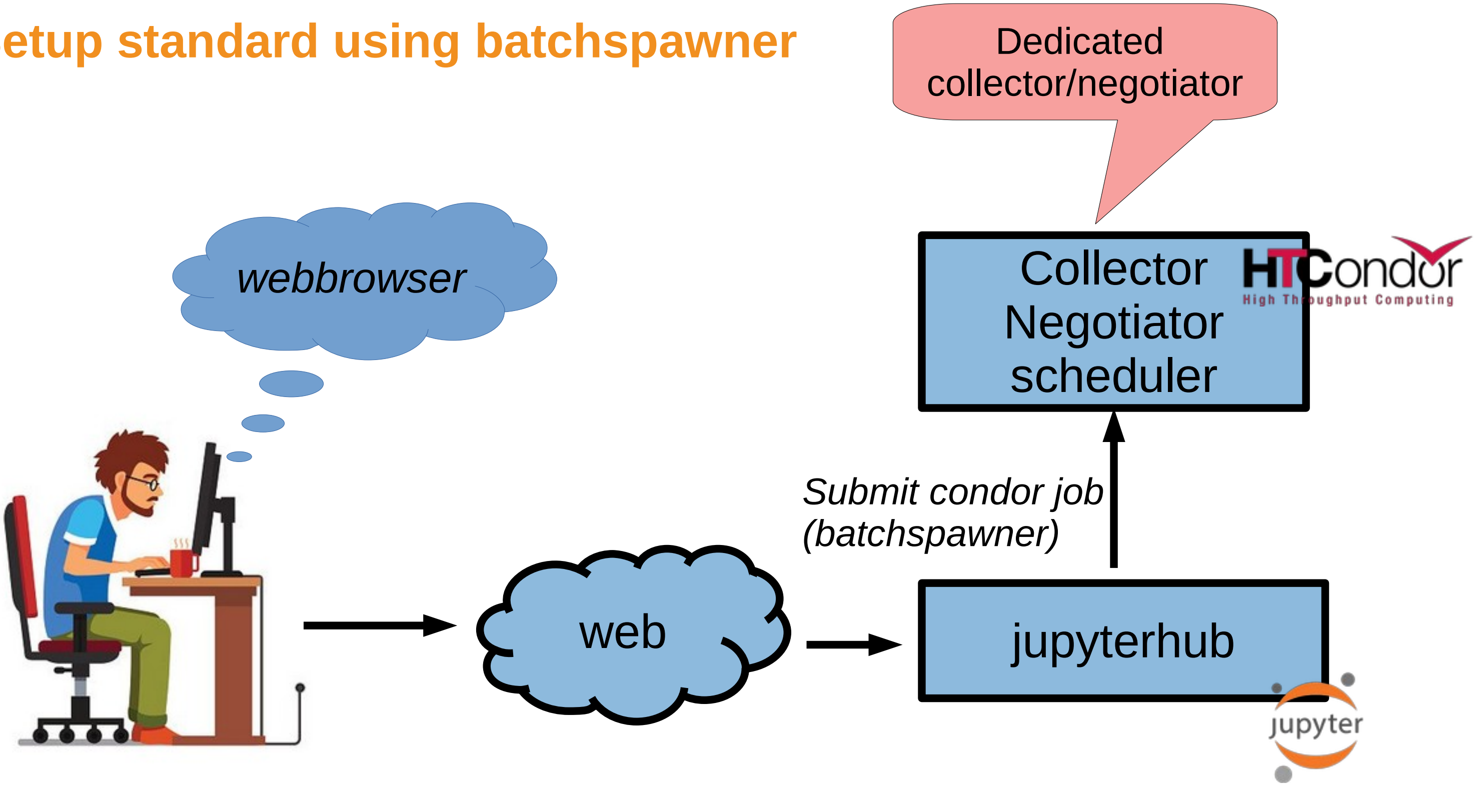
Jupyter Notebooks in NAF

Setup standard using batchspawner



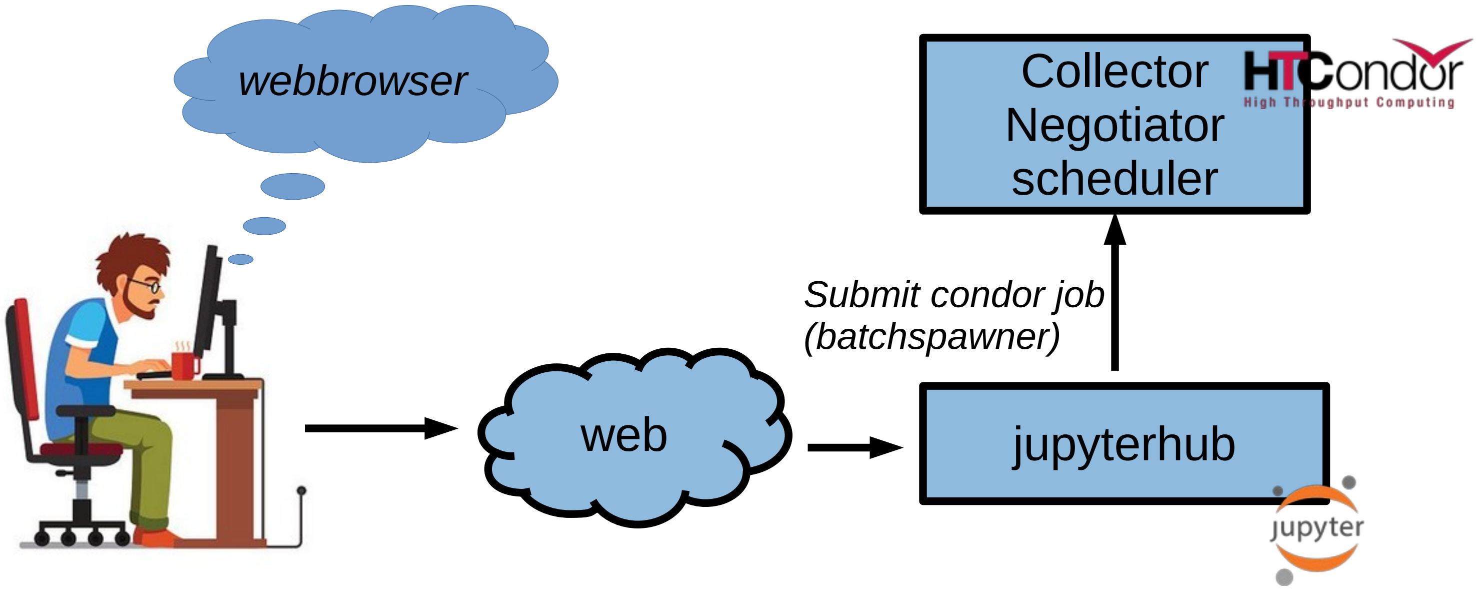
Jupyter Notebooks in NAF

Setup standard using batchspawner



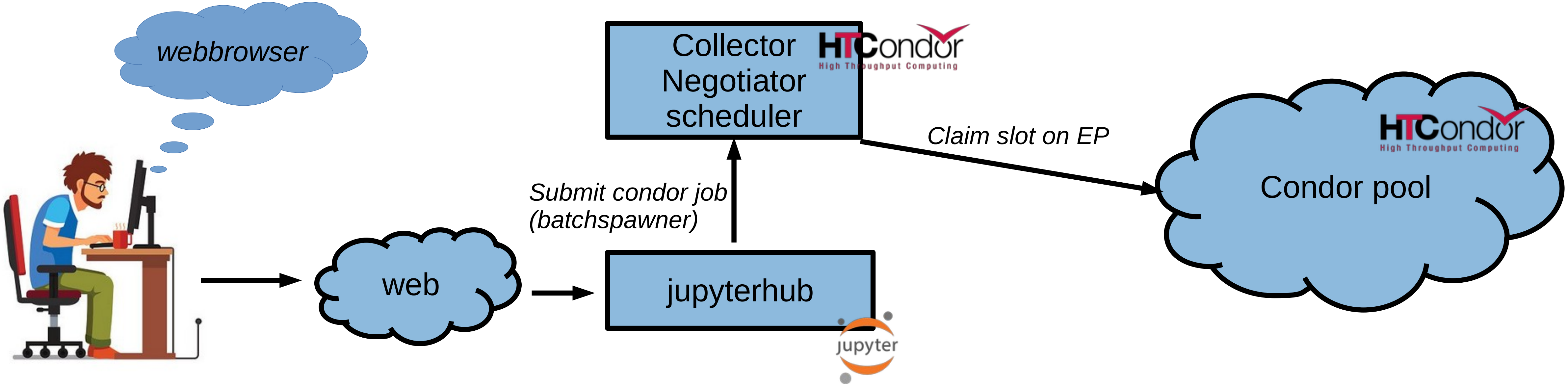
Jupyter Notebooks in NAF

Setup standard using batchspawner



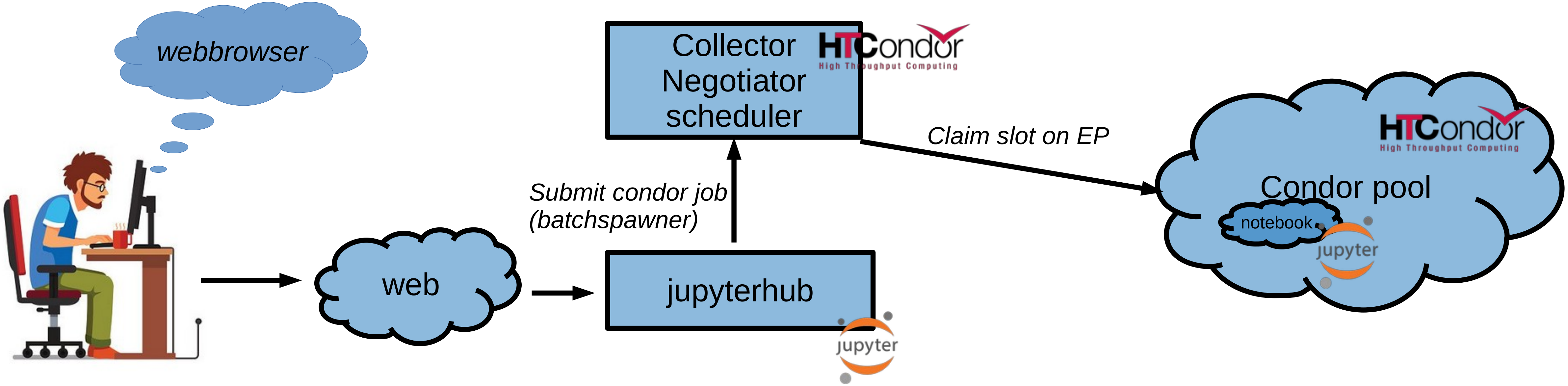
Jupyter Notebooks in NAF

Setup standard using batchspawner



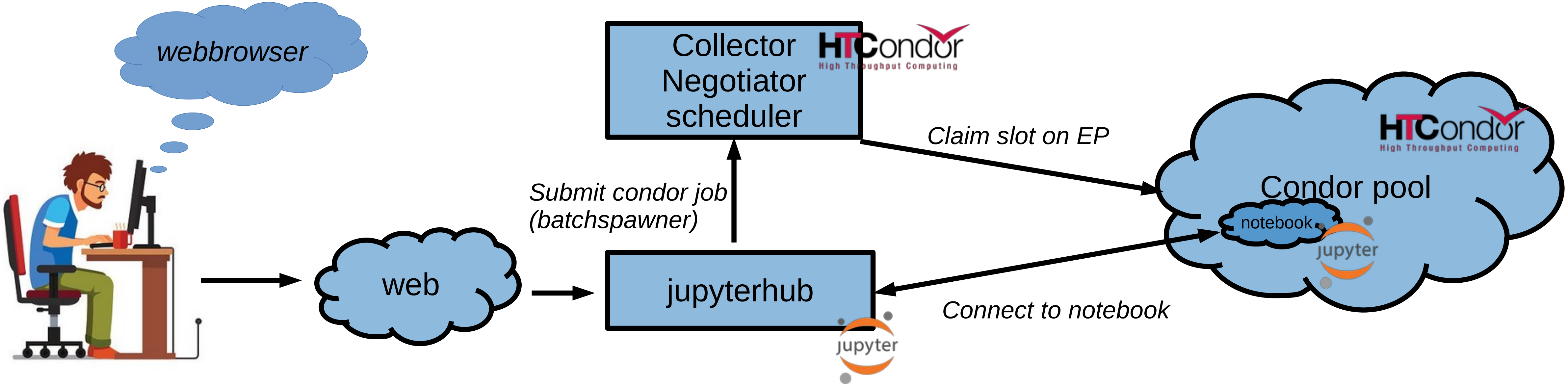
Jupyter Notebooks in NAF

Setup standard using batchspawner



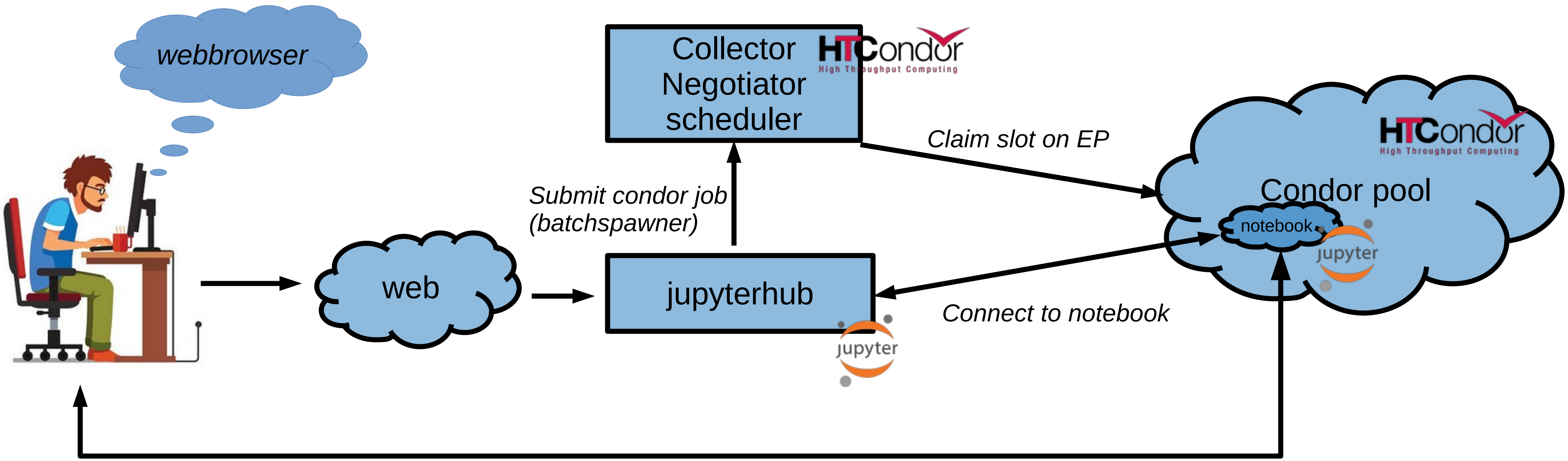
Jupyter Notebooks in NAF

Setup standard using batchspawner



Jupyter Notebooks in NAF

Setup standard using batchspawner



webbrowser

web

Collector
Negotiator
scheduler

jupyterhub

Submit condor job
(batchspawner)

Claim slot on EP

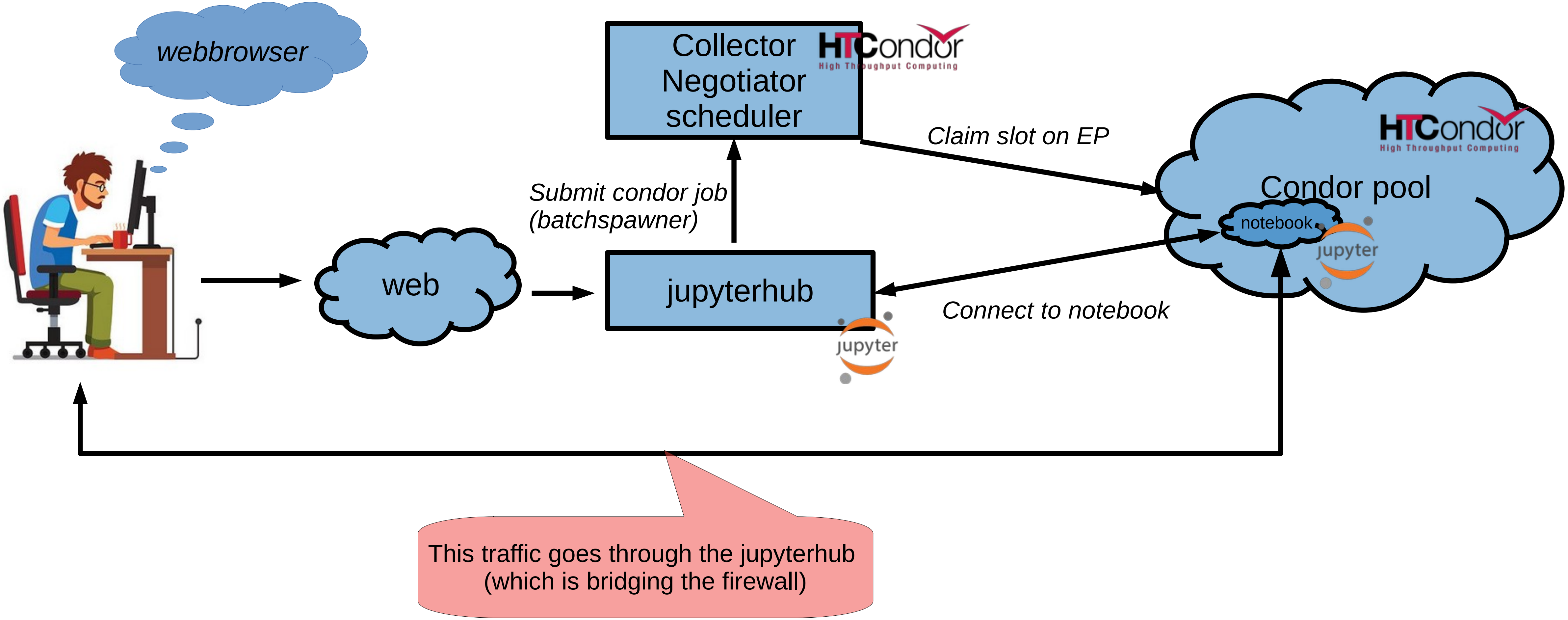
Condor pool

notebook

Connect to notebook

Jupyter Notebooks in NAF

Setup standard using batchspawner

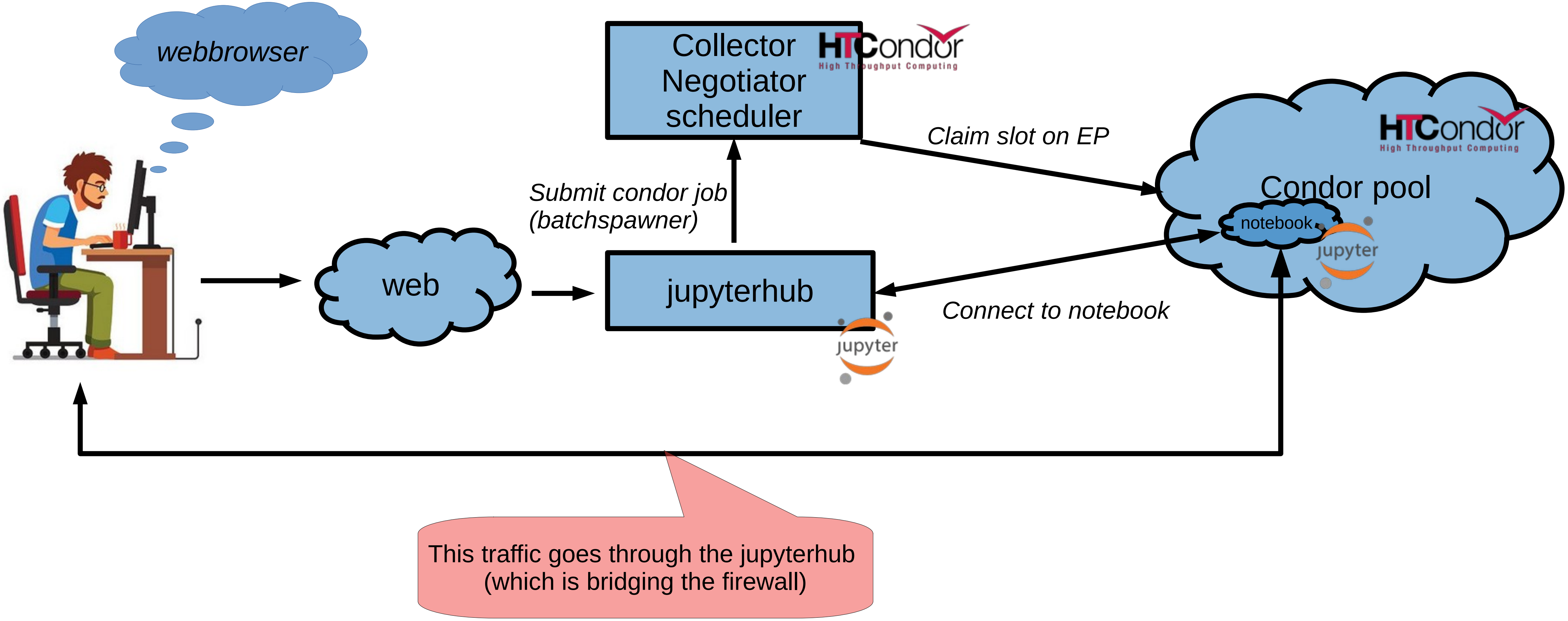


This traffic goes through the jupyterhub
(which is bridging the firewall)

Jupyter Notebooks in NAF

Setup standard using batchspawner

See Olivers talk in a minute how to do it properly ;)



JUPYTER notebooks -

Just another condor_submit-wrapper -right ?

How we envisioned – and implemented it

- Introduced 2018 @ DESY when IT infrastructure collaboration with BELLE2 was intensified
- Jupyterhub bridging the NAF into the WAN
- Small reserved slots for notebooks sufficient on the NAF workers
  - 1 core 1,5 GB memory
  - Soft policy, notebook stopped if mem-usage > 4,5 GB
- Fast start of notebooks due to separate negotiator/collector (<10 secs)
- Users use htmap and python bindings to outsource workload into the pool
- ‘Older’ VO’s like ATLAS and CMS will adapt to jupyter notebooks and it will become a default mean of access to the NAF
- BELLE will heavily rely on notebooks as they are widely accepted in their community
- Debugging issues beyond the notbook start itself will be time consuming and python knowledge will be necessary

JUPYTER notebooks – insights after 2 years of usage

A mixed bag

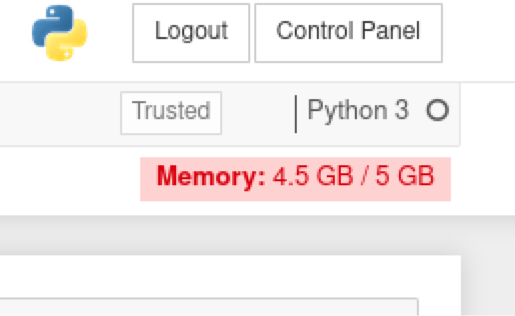
Reality strikes again

- Jupyterhub bridging the NAF into the WAN ✓
- Small reserved slots for notebooks sufficient on the NAF workers ✗
  - 1 core 1,5 GB memory
  - Soft policy, notebook stopped if mem-usage > 4,5 GB
- Fast start of notebooks due to separate negotiator/collector (<10 secs) ✓
- Users use htmap and python bindings to outsource workload into the pool ✗
- ‘Established’ VO’s like ATLAS and CMS will adapt to jupyter notebooks and it will become a default mean of access to the NAF ✓
- BELLE will heavily rely on notebooks as they are widely accepted in their community ✓
- Debugging issues beyond the notebook start itself will be time consuming and python knowledge will be necessary ✓

JUPYTER notebooks

Summary & outlook

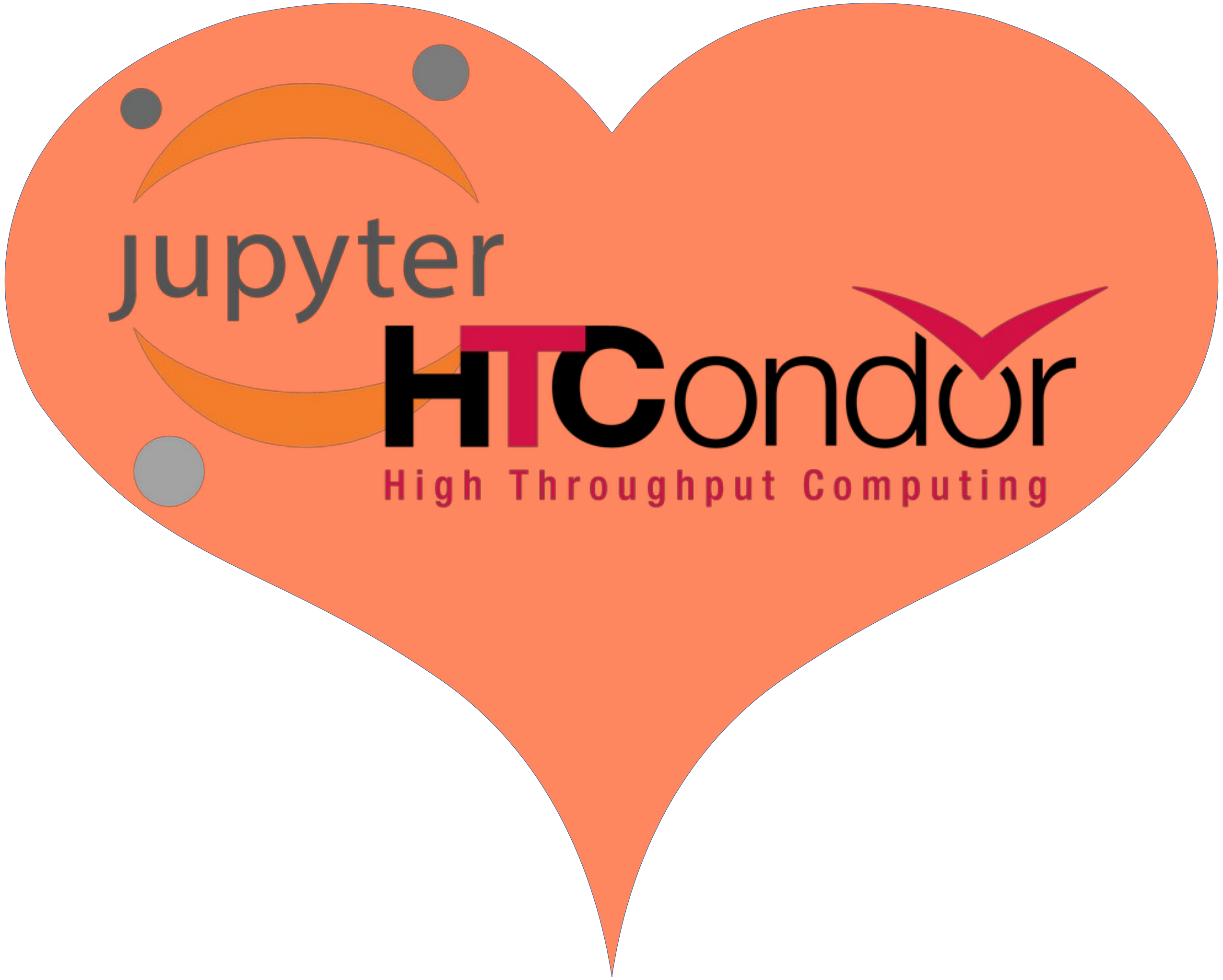
- User want to scroll through bigger amounts of data and in general are not prepared to outsource any workloads
  - Lately first user interest in submitting from inside notebooks though
- Bigger notebooks in terms of memory only partly a solution (similar to fixing a memory leak) ;)
- Will provide 3 classes of notebooks ~~in the future~~ – active since wednesday
  - 12/20/48 GB Memory 1/2/2 cores 12/6/3 h runtime
  - Dropdown chooser in jhub (see live demo later on)
  - Bigger notebooks run on 2 dedicated servers, regular ones in the smaller reserved slots
  - Will see how this works, if nothing else it will raise the awareness that there are limits
- Jupyter-resource-usage (pip install) provides a nice memory reminder in the upper right corner of the notebook
- DASK & SPARC as memory scheduler in conjunction with Jupyter notebooks are tested and work but need some tweaking and setup knowledge by the user – not an out-of-the-box solution at the moment



While have your attention

Some things would be nice to have

- Both HTC and Jupyter notebooks run as if they were industry standards – maybe that is because they are!
- Administering the both of them playing together is much less of a hassle than anticipated in the first place
- Any kind of comunication backchannel from HTC to the notebook would be necessary to deploy more sophisitcated setups
  - Pop-ups in case of limited runtime and memory usage nearing the limit
  - Small stats about the performance of the node we are on
  - In general: make the notebook more aware that it is running on a remote ressource with some possible boundaries



Last slide :)

Subheading, optional

Thank you for your attention !

Questions ?