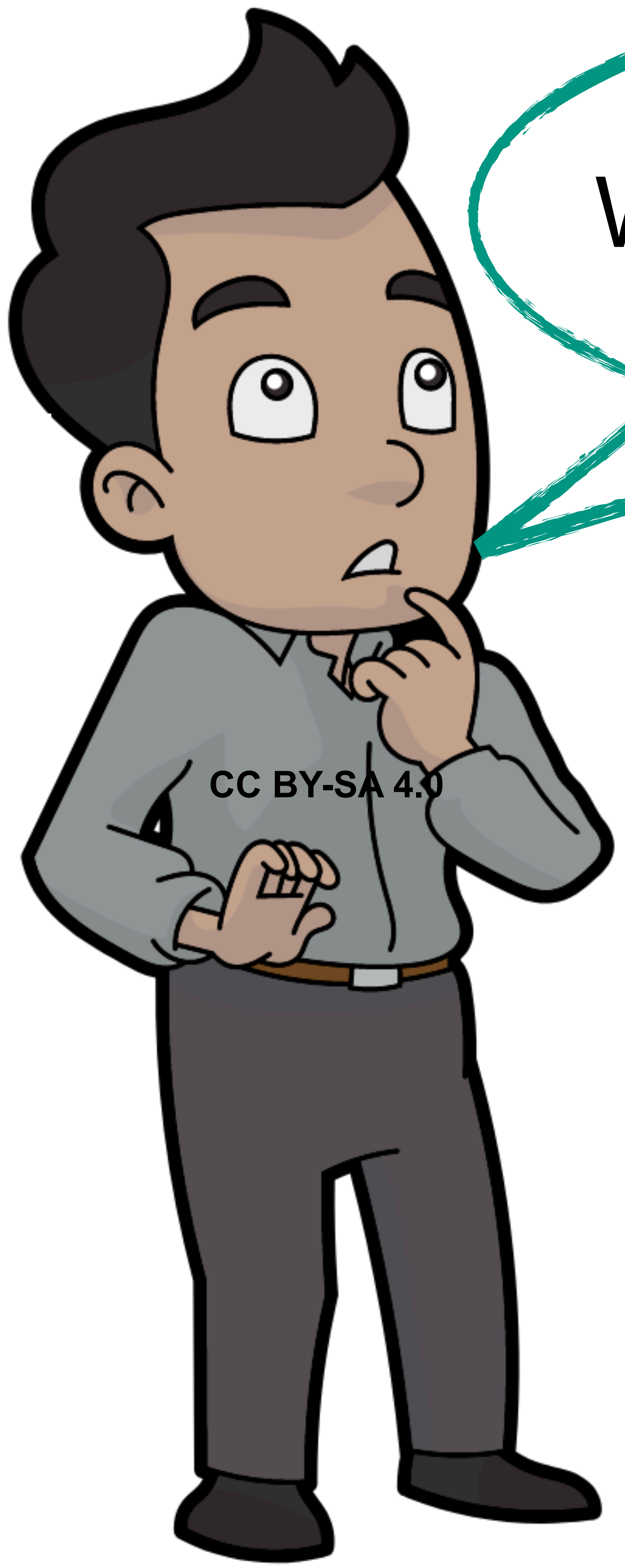
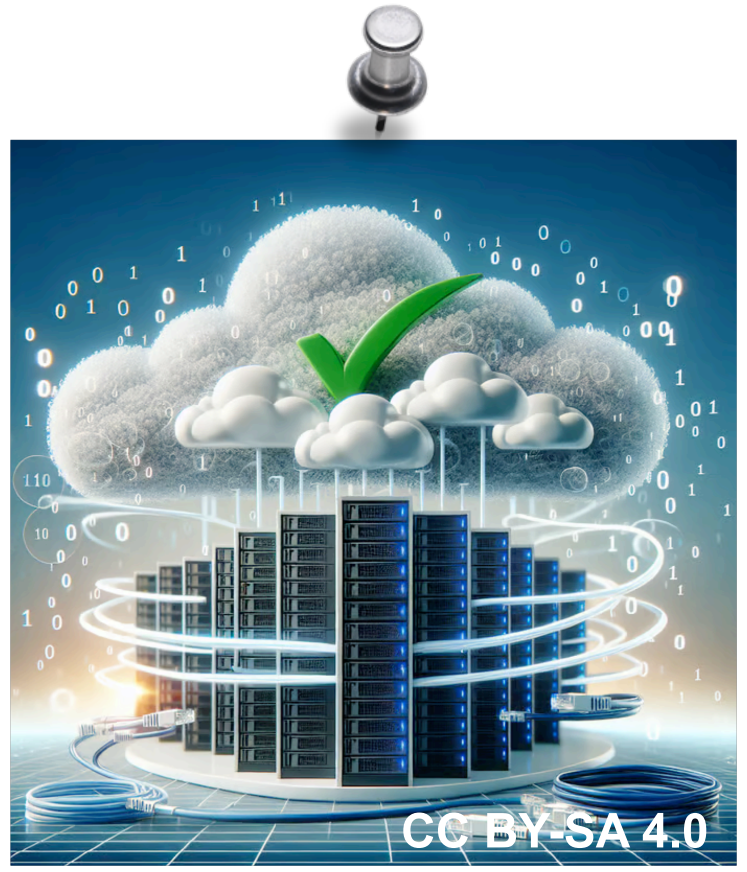
Dynamic resource integration with COBalD/TARDIS

HTCondor Workshop 2024
R. Florian v. Cube for the KIT computing group
Contents and slides provided by Manuel Giffels

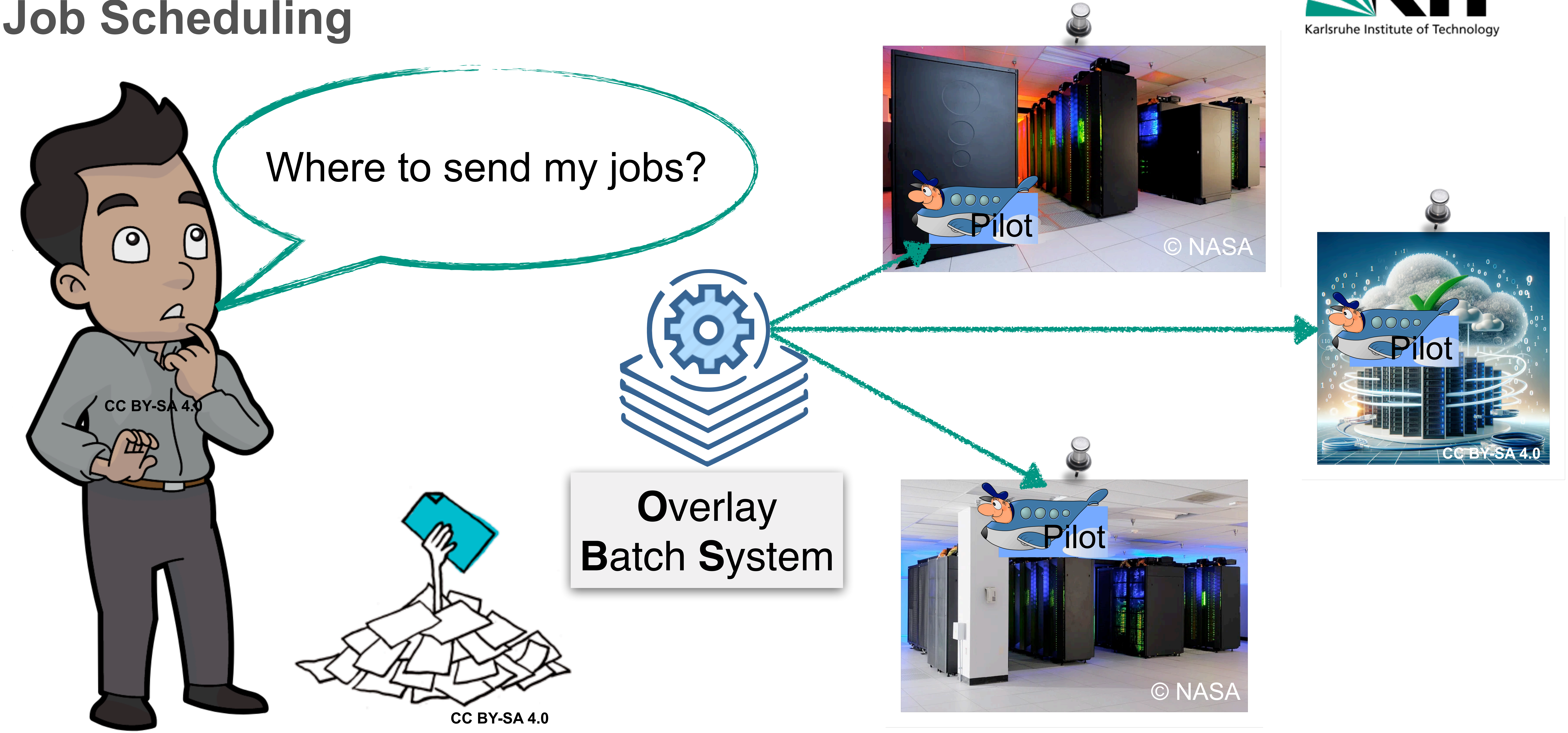
Setting the Scene
Job Scheduling

Where to send my jobs?

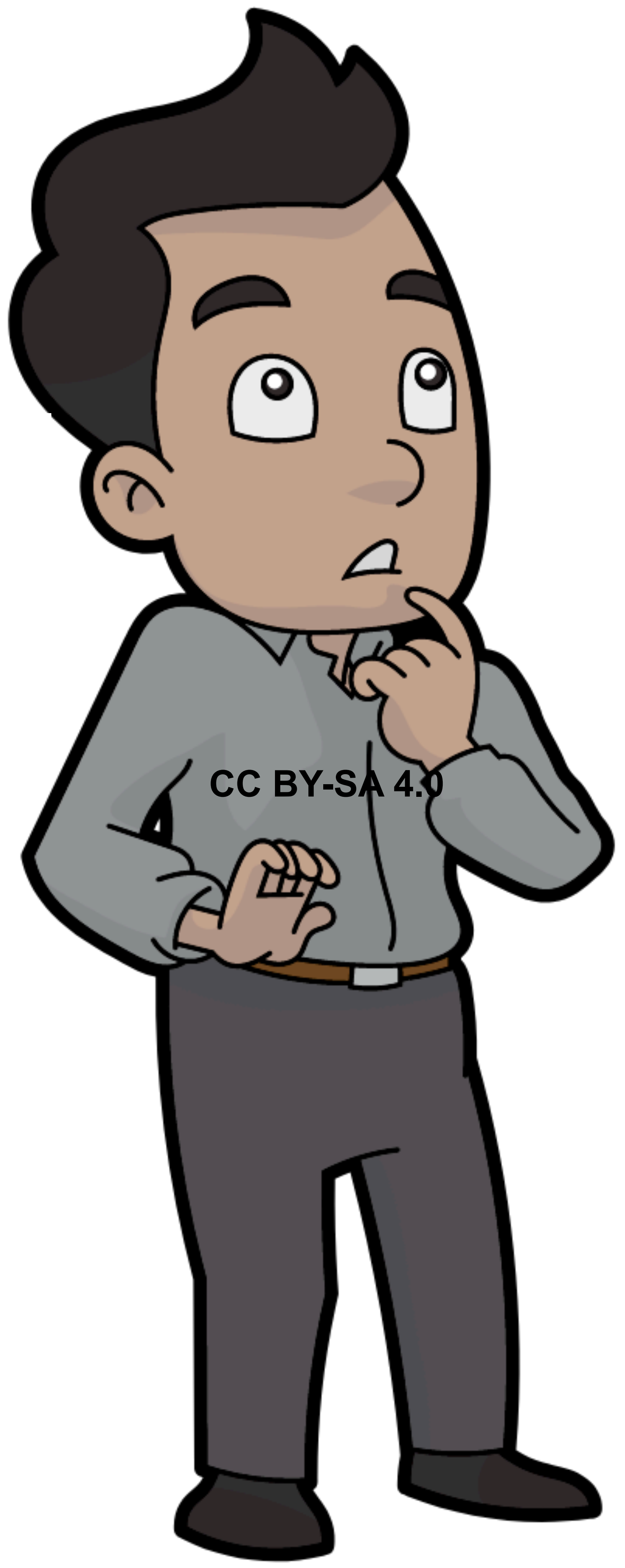


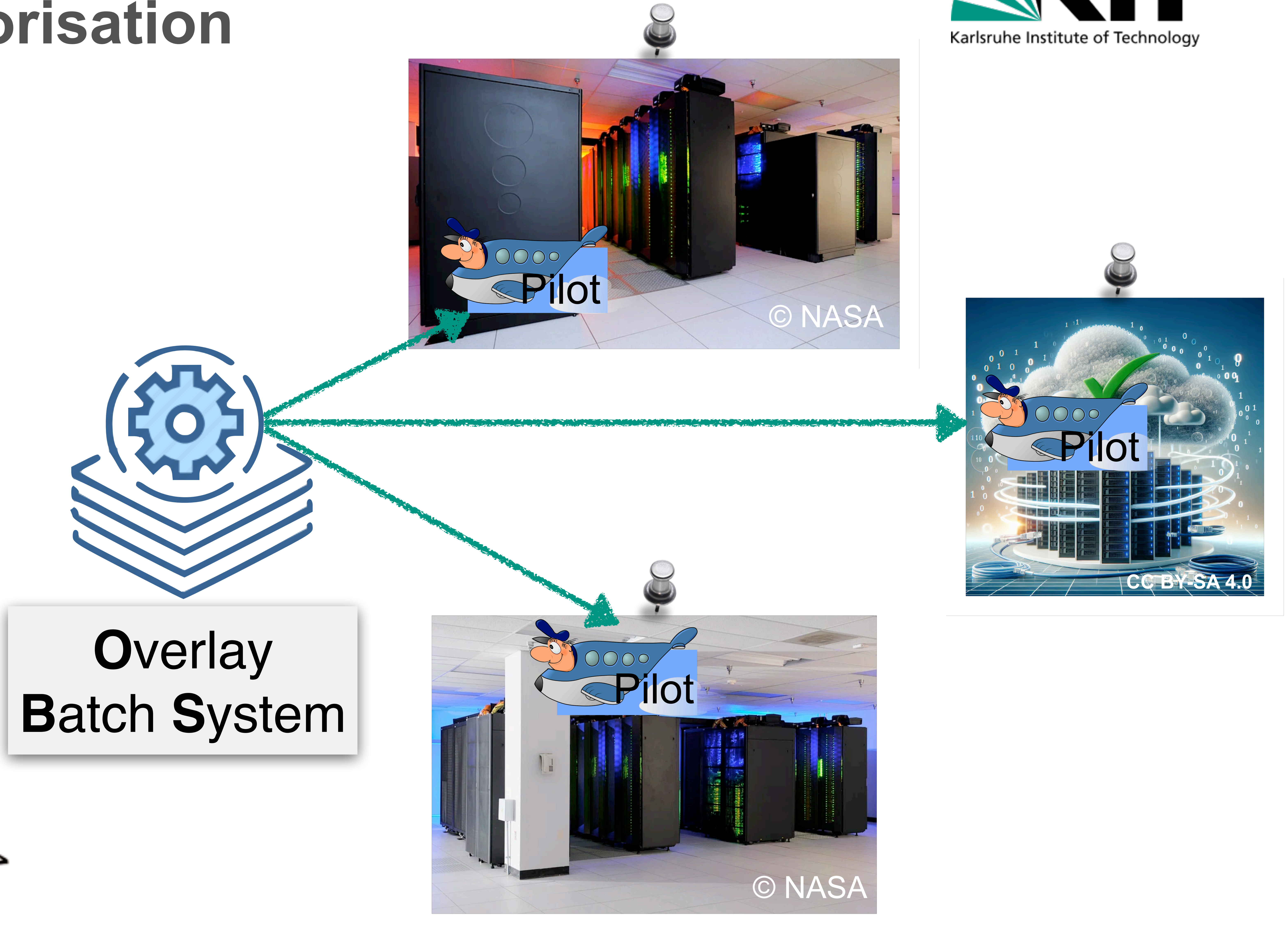


Setting the Scene
Job Scheduling

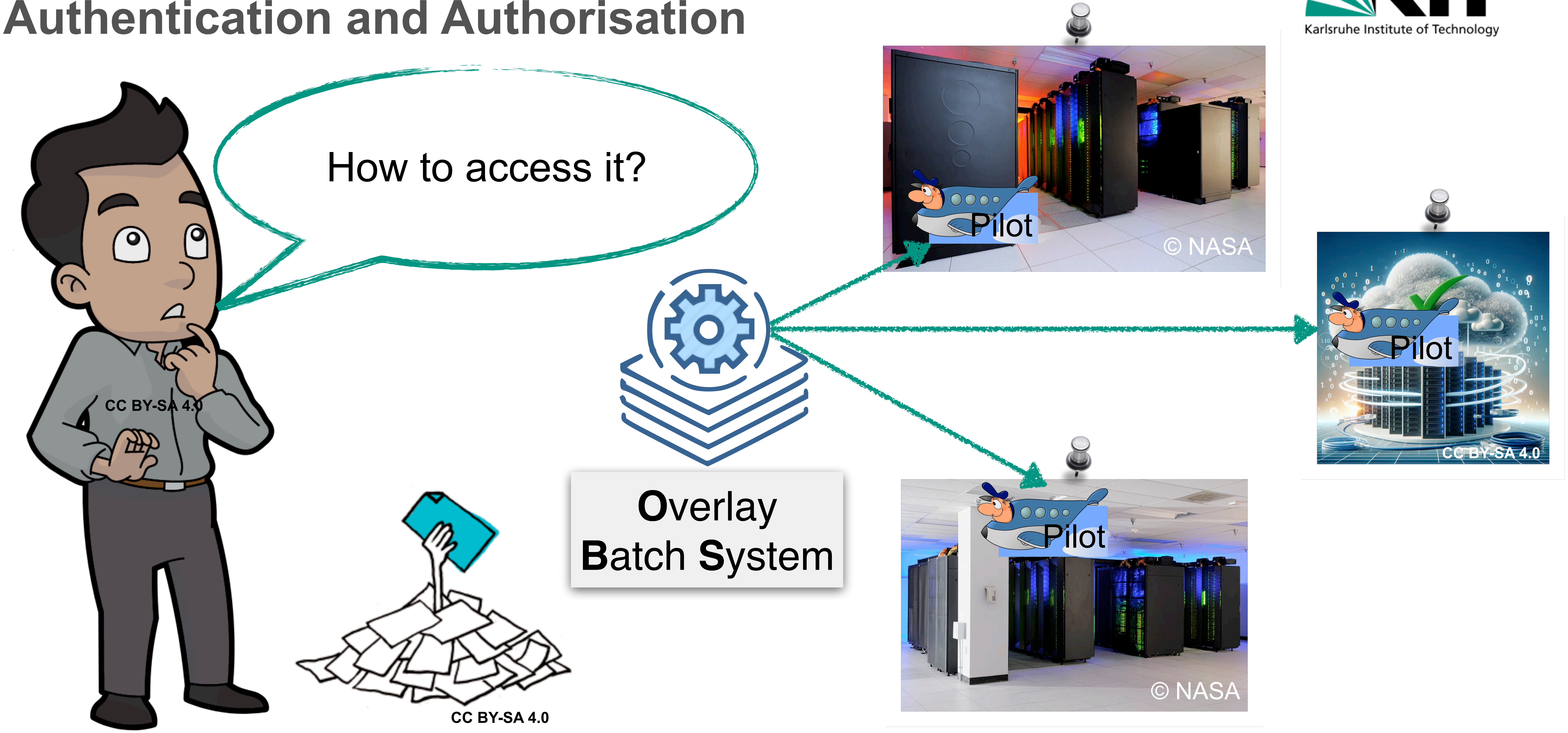


Setting the Scene
Authentication and Authorisation

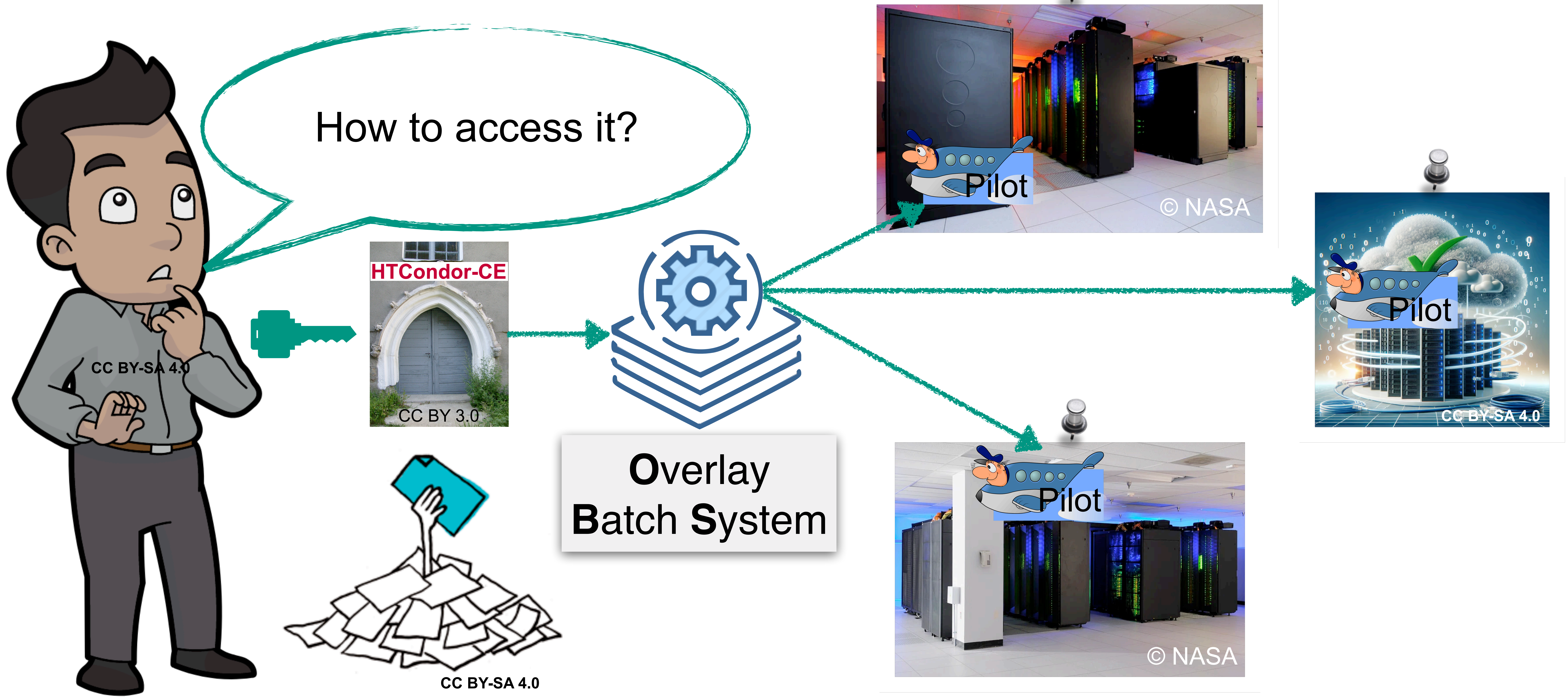




Setting the Scene
Authentication and Authorisation

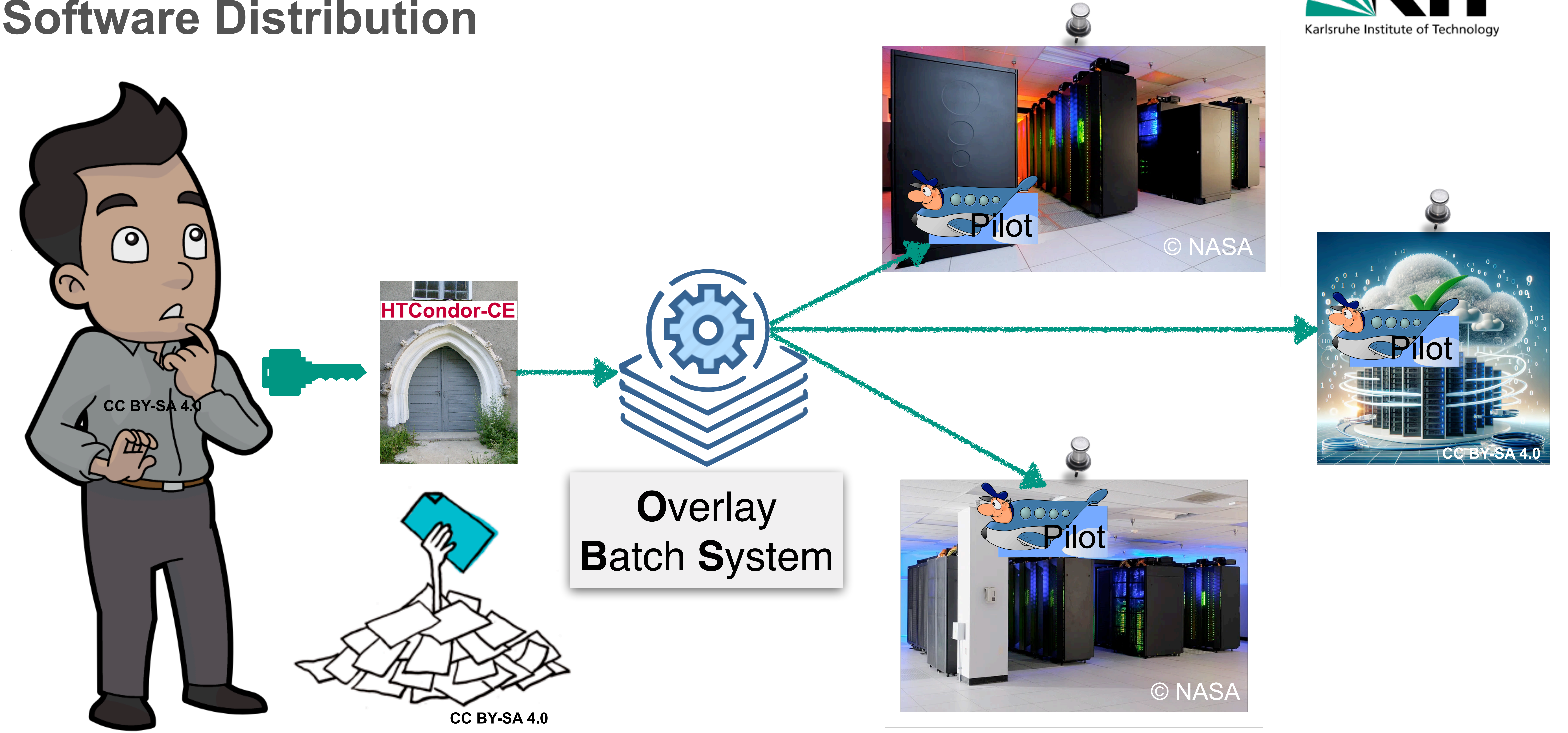


Setting the Scene
Authentication and Authorisation

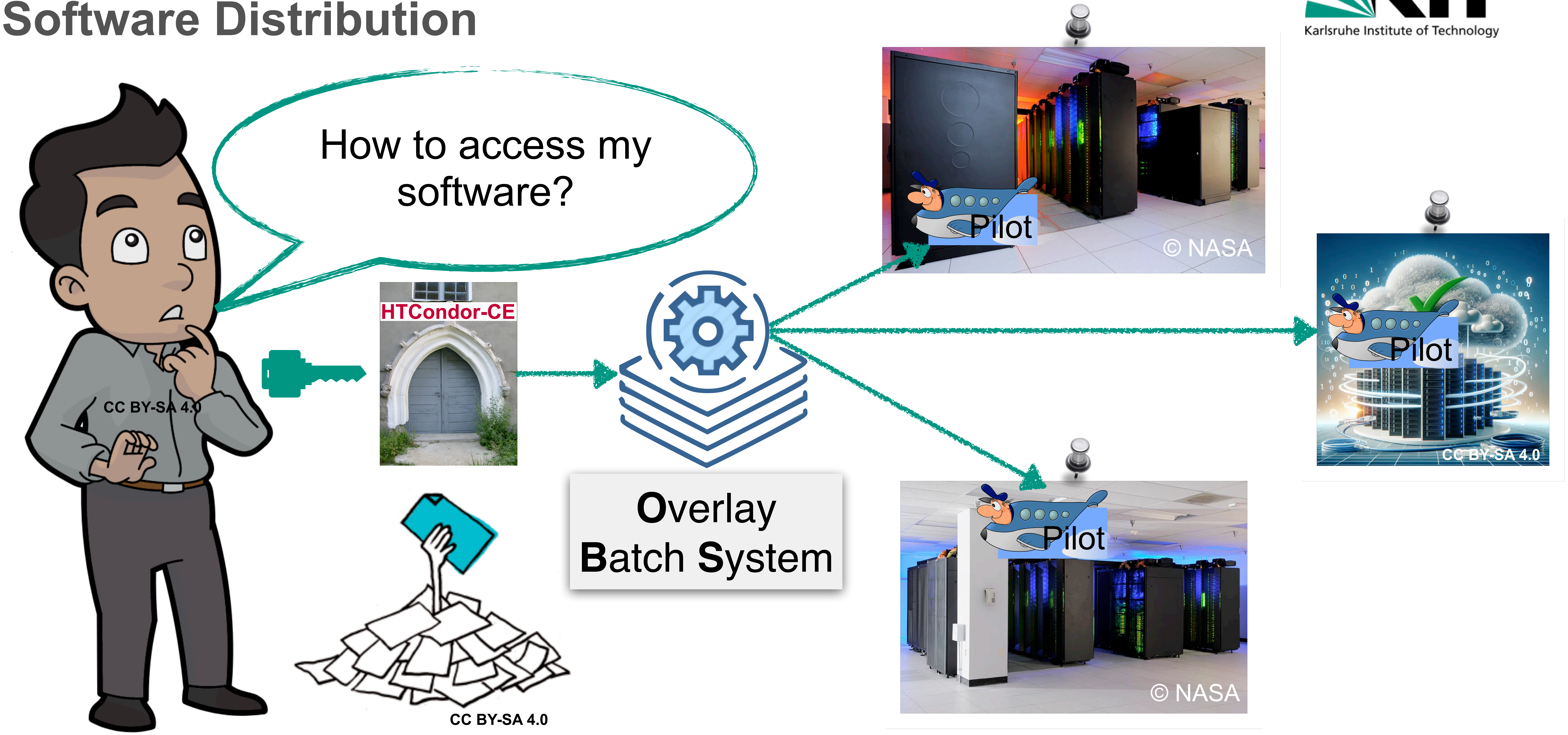


Setting the Scene
Software Distribution

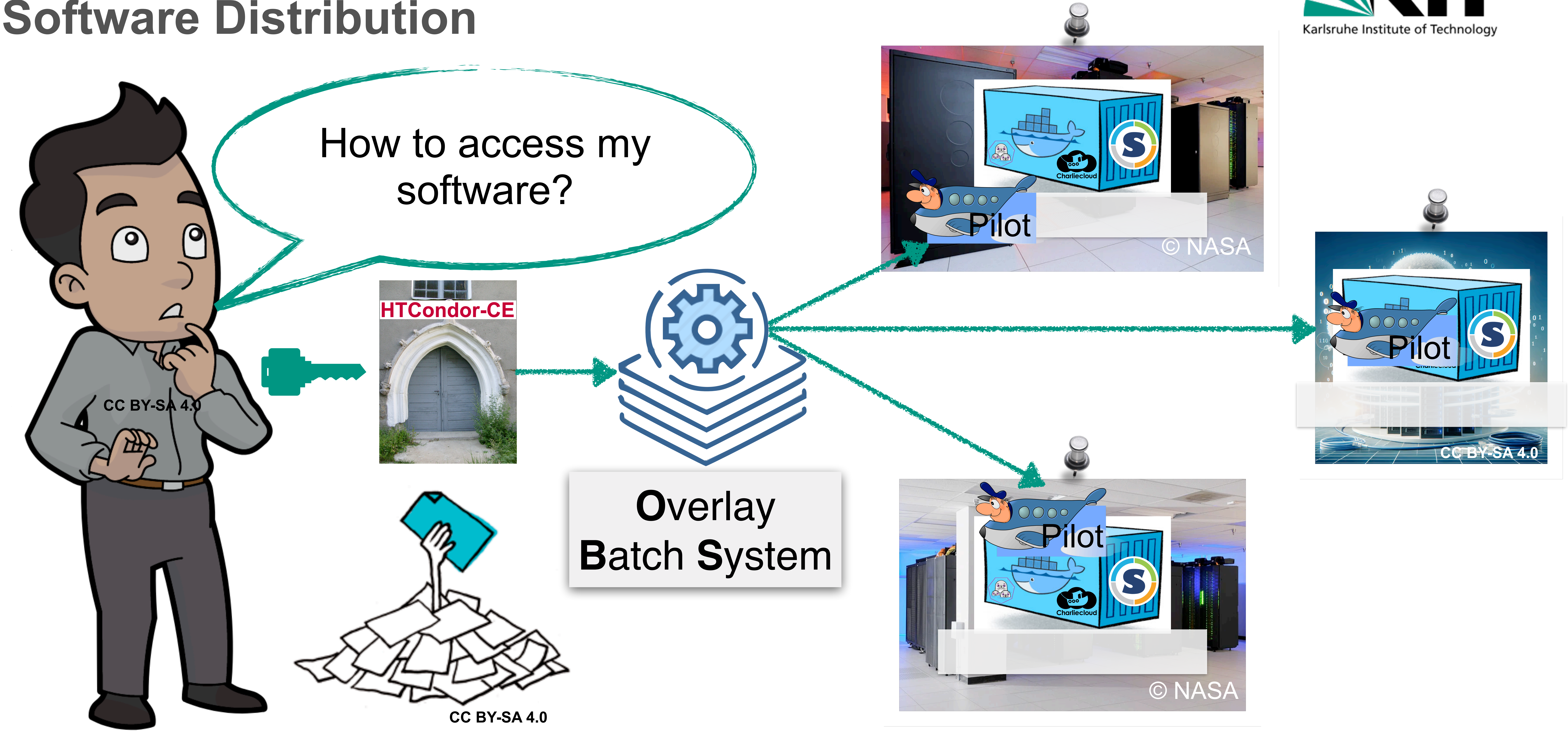
Karlsruhe Institute of Technology



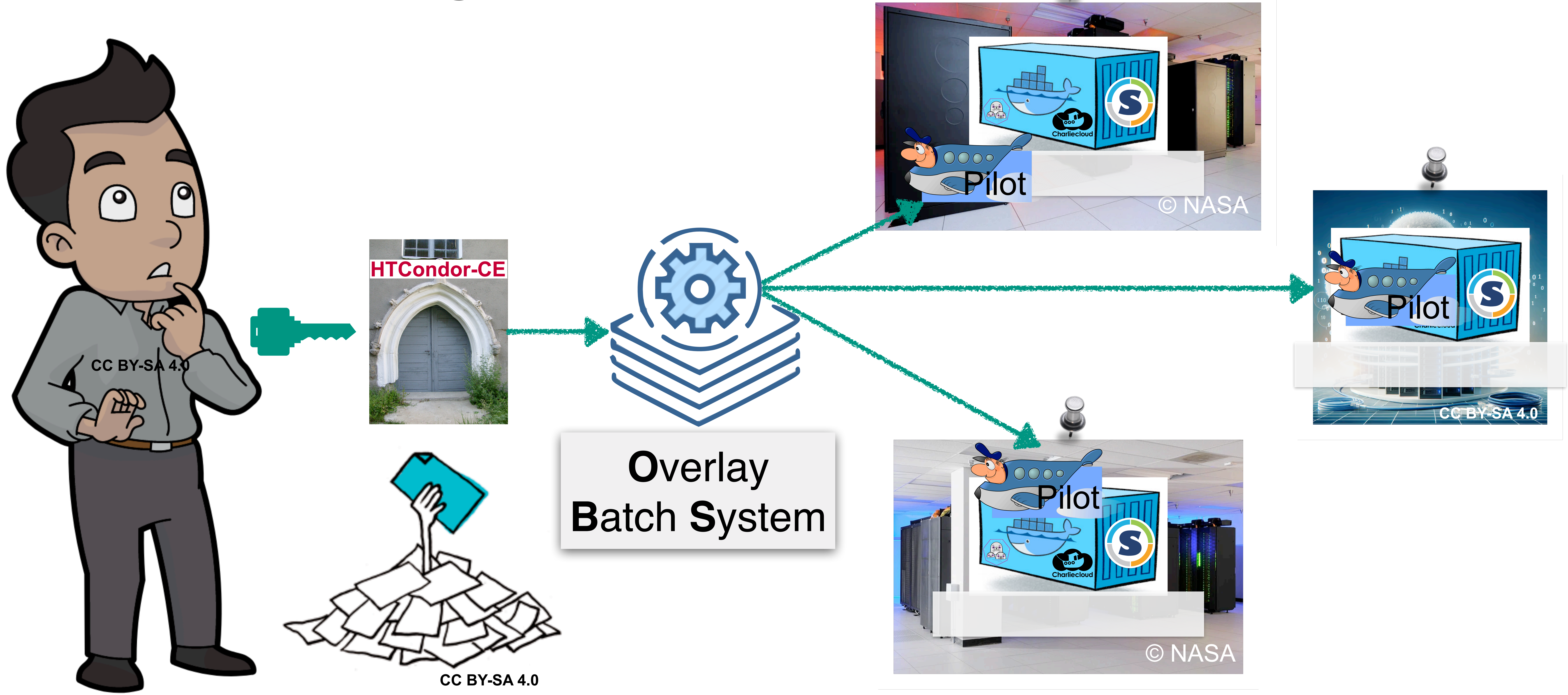
Setting the Scene
Software Distribution



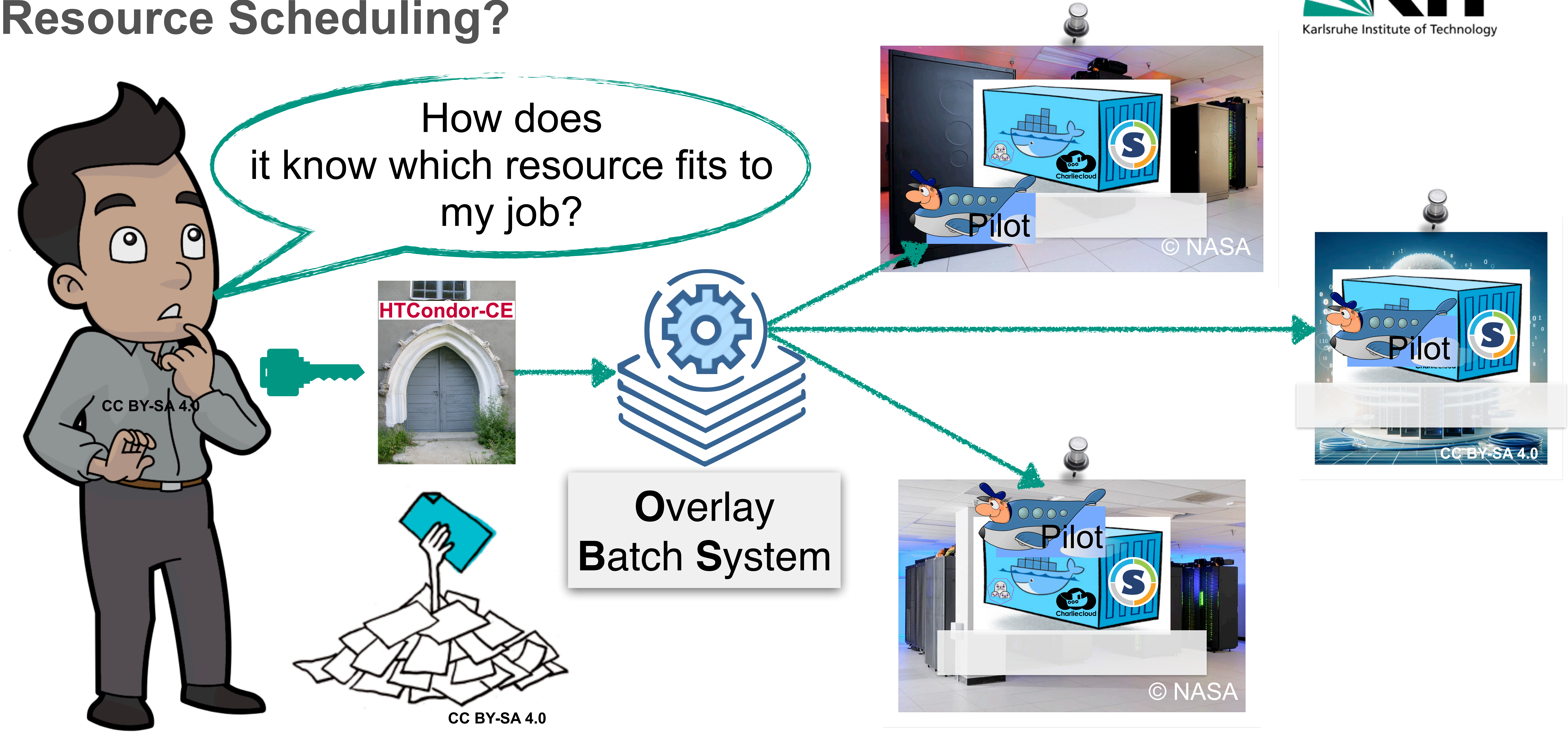
Setting the Scene
Software Distribution



Setting the Scene
Resource Scheduling?



Setting the Scene
Resource Scheduling?



The COBalD View of Resource Meta-Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard“ problem
- Usually based on predictions of the future resource availability and mixture of job classes (e.g. CPU intense, I/O intense, …)
- **<u>However:</u>** We usually care only about a simpler problem!

Based on a slide by Max Fischer

The COBalD View of Resource Meta-Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard
- Usually based on predictions of the future resource availa ity and mixture of job classes (e.g. CPU intense, I/O intense, …)
- <u>**However:**</u> We usually care only about a simpler problem!

Works perfectly fine in homogenous environments. (e.g. the Grid)

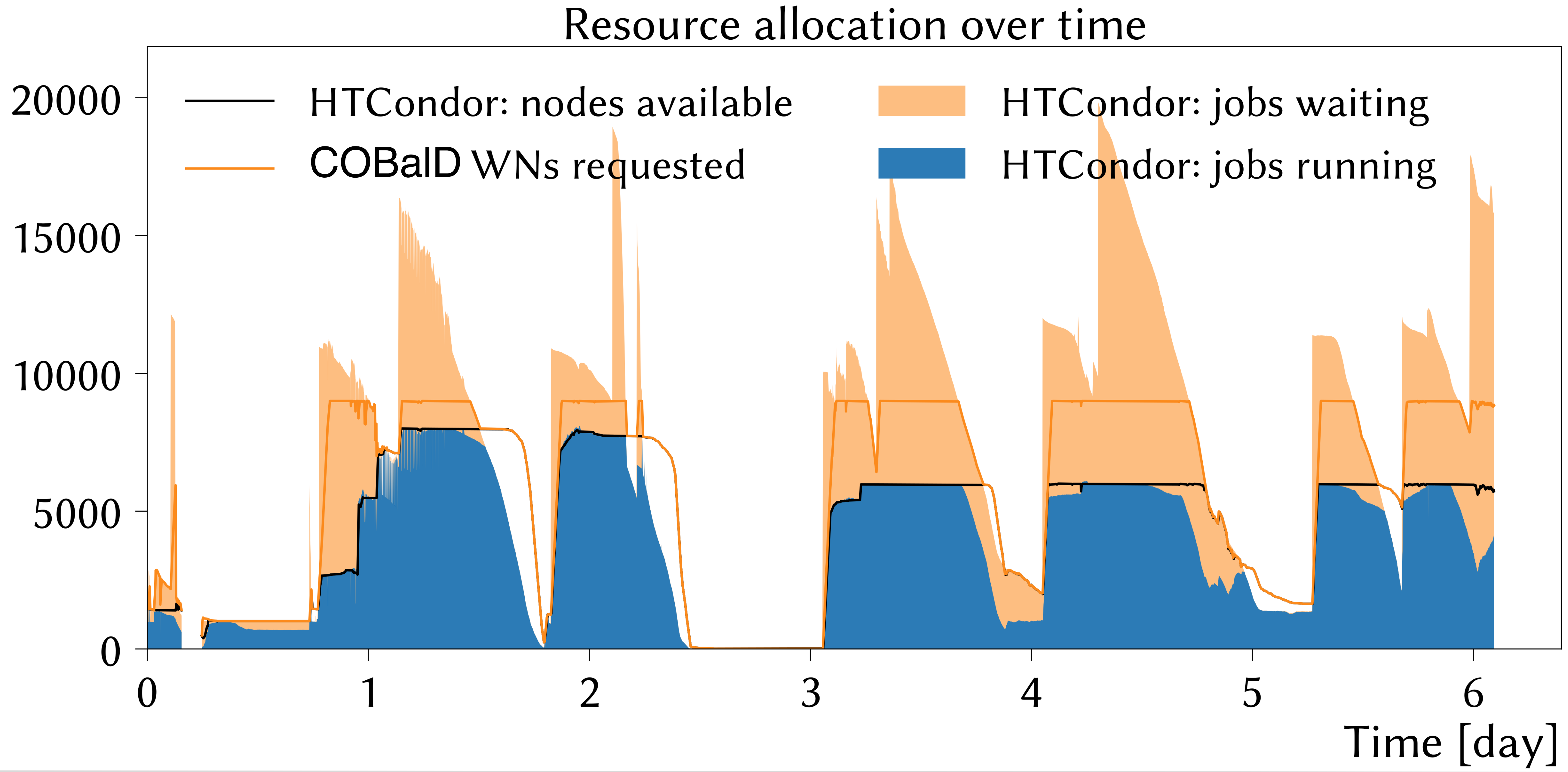
Based on a slide by Max Fischer

The COBalD View of Resource Meta-Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard
- Usually based on predictions of the future resource availa... and mixture of job classes (e.g. CPU intense, I/O intense, …)
- **However:** We usually care only about a simpler problem!

Fails in heterogeneous environments (e.g. HPCs & Clouds).



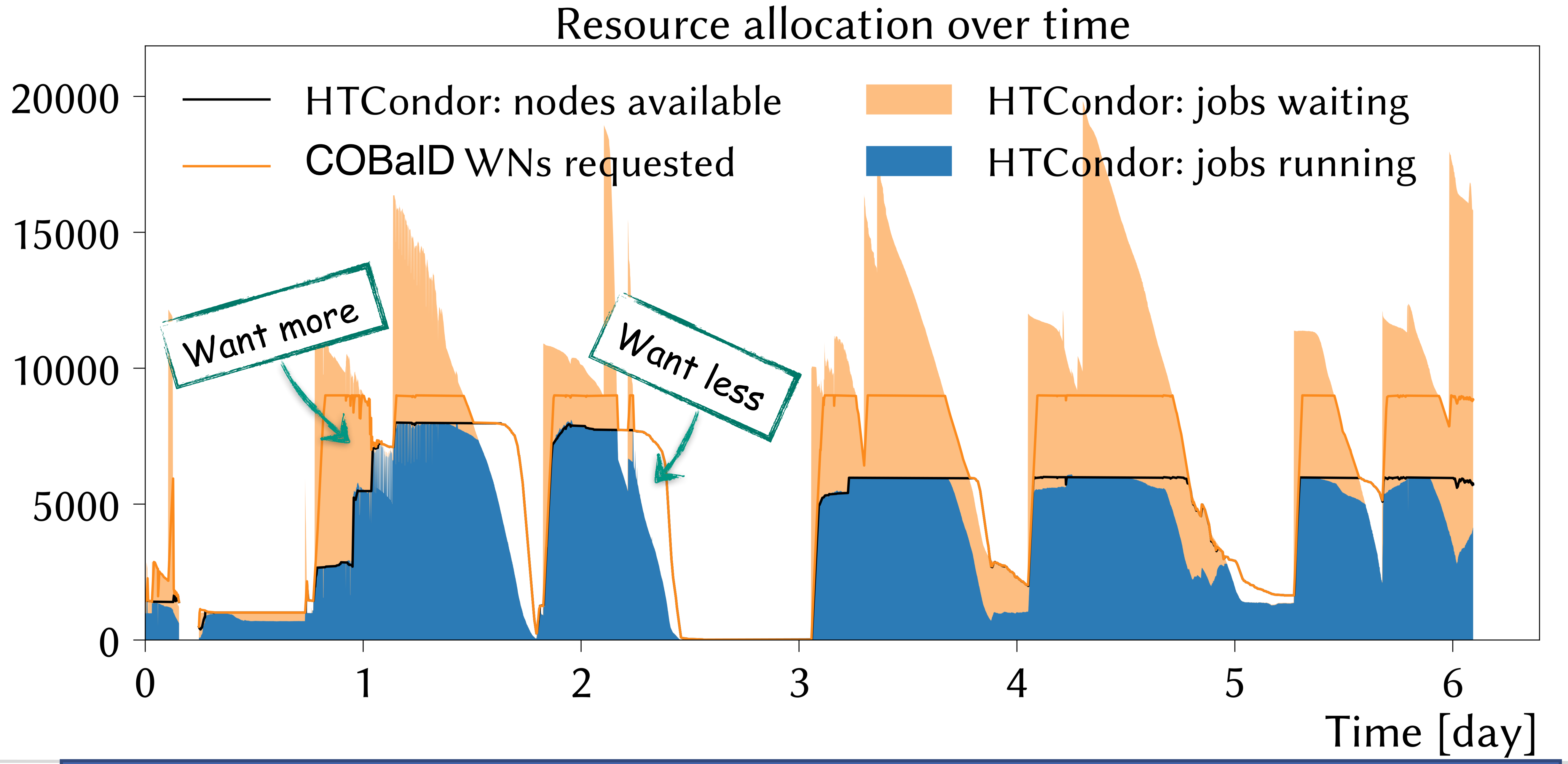
Based on a slide by Max Fischer

The COBalD View of Resource Meta-Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard
- Usually based on predictions of the future resource availa… and mixture of job classes (e.g. CPU intense, I/O intense, …)
- **<u>However:</u>** We usually care only about a simpler problem!

Fails in heterogeneous environments (e.g. HPCs & Clouds).



Based on a slide by Max Fischer

Feedback based Resource Meta-Scheduler

The COBalD View of Resource Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard“
- COBalD cares only about resources, not jobs
  - Observe how much and how well each resource is used
  - Increase well-utilized resources, reduce poor utilized resources

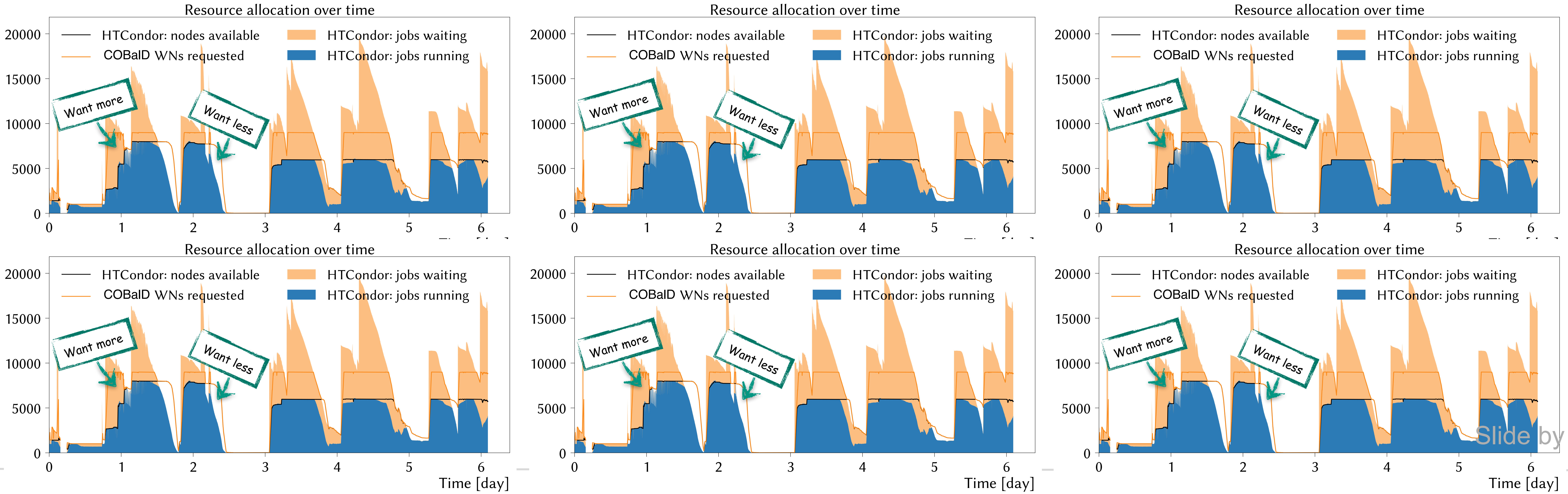
Slide by Max Fischer

The COBalD View of Resource Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard“
- COBalD cares only about resources, not jobs
  - Observe how much and how well each resource is used
  - Increase well-utilized resources, reduce poor utilized resources

COBalD



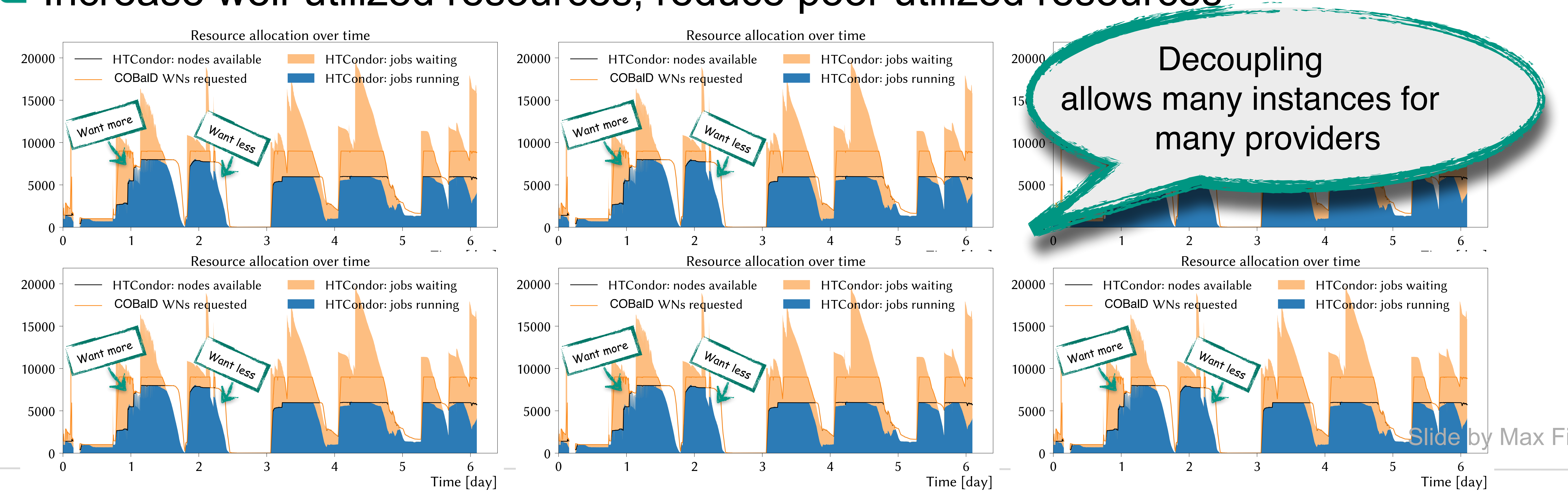
Slide by Max Fischer

The COBalD View of Resource Scheduling

[COBalD - the **O**pportunistic **Bal**ancing **D**aemon]

- Resource Meta-Scheduling for Job Scheduler is a „hard“
- COBalD cares only about resources, not jobs
  - Observe how much and how well each resource is used
  - Increase well-utilized resources, reduce poor utilized resources

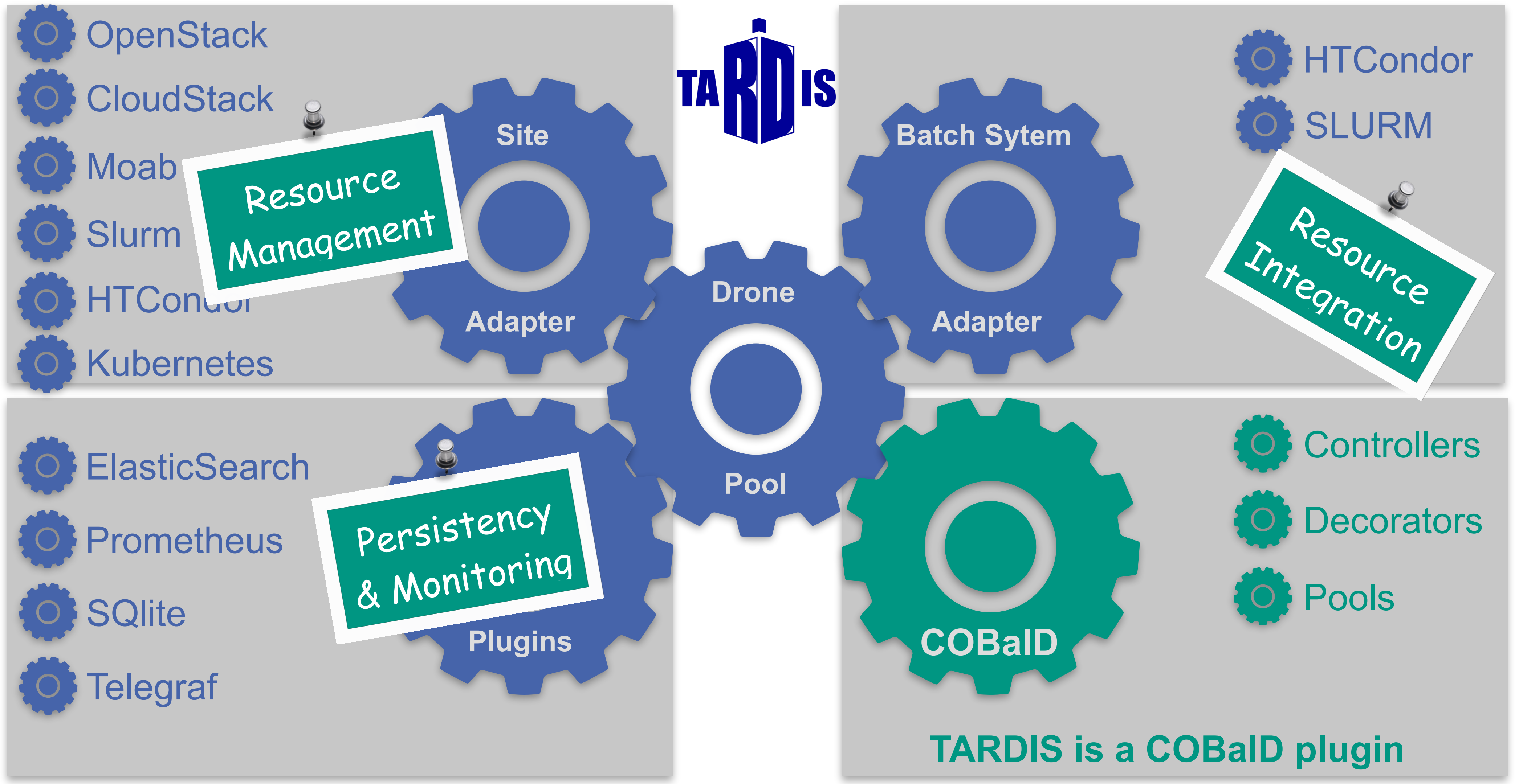
COBalD



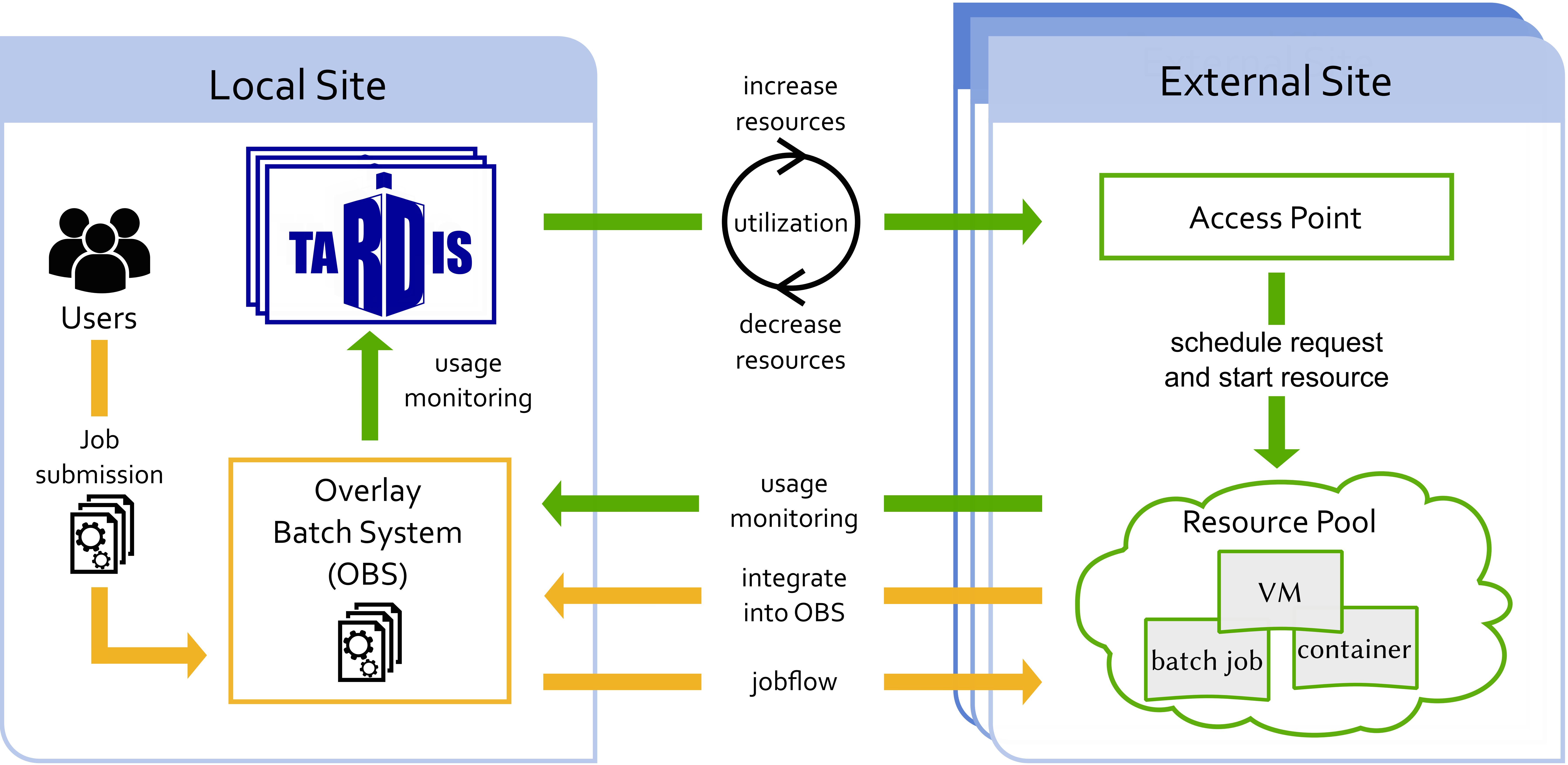
TARDIS - Out-of-the-Box Resource Adapters

[**T**ransparent **A**daptive **R**esource **D**ynamic **I**ntegration **S**ystem]

- Combine resource provider APIs with COBalD
  - Request, monitor, decommission individual resources (Manages resource life cycle)
  - Automatically matches resource demand via COBalD approach
  - Basically a „use-case agnostic autonomous Pilot factory“
- Support for common HPC batch systems, Cloud APIs, …
  - Behave like „regular users“ as much as possible
  - Customizable pilot for each centre’s peculiarities
  - HEP: Insert HTCondor+CVMFS as available



COBalD/TARDIS & Opportunistic Resources in Practice



The Entire COBalD/TARDIS Ecosystem

container-stacks

Container images to provide dedicated job environments

Available containers

Container	Environment provided
wlcg-wn	Provides a standard environment to run all jobs of VOs supported by WLCG
htcondor-wn	Provides a standard htcondor enabled workernode configurable using ansible

cobald Public

Cobald is an Opportunistic Balancing Deamon

Python 11 9

tardis Public

Transparent Adaptive Resource Dynamic Integration System

Python 15 17

H **HTCondor_configs**

Project ID: 3523 Leave project

236 Commits 3 Branches 0 Tags 278 KiB Project Storage

HTCondor configs for each site

condor-git-config 0.1.5

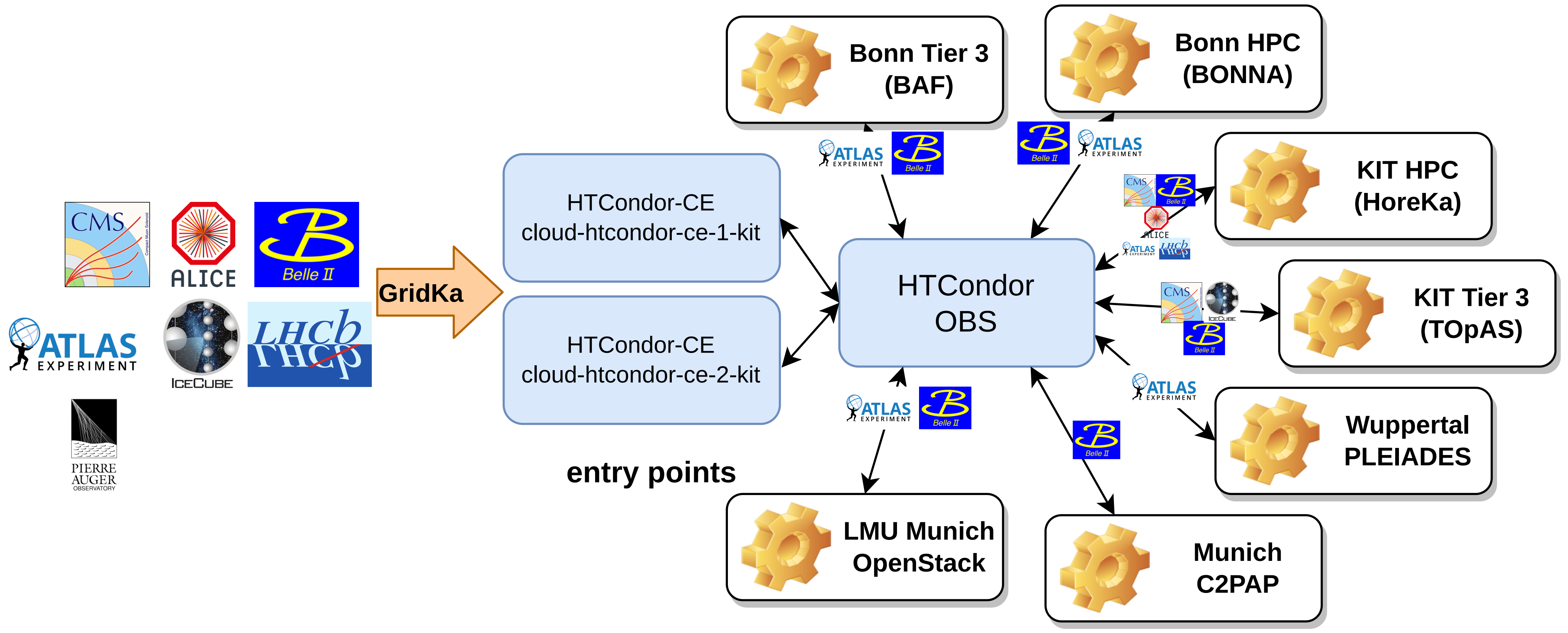
```
pip install condor-git-config
```

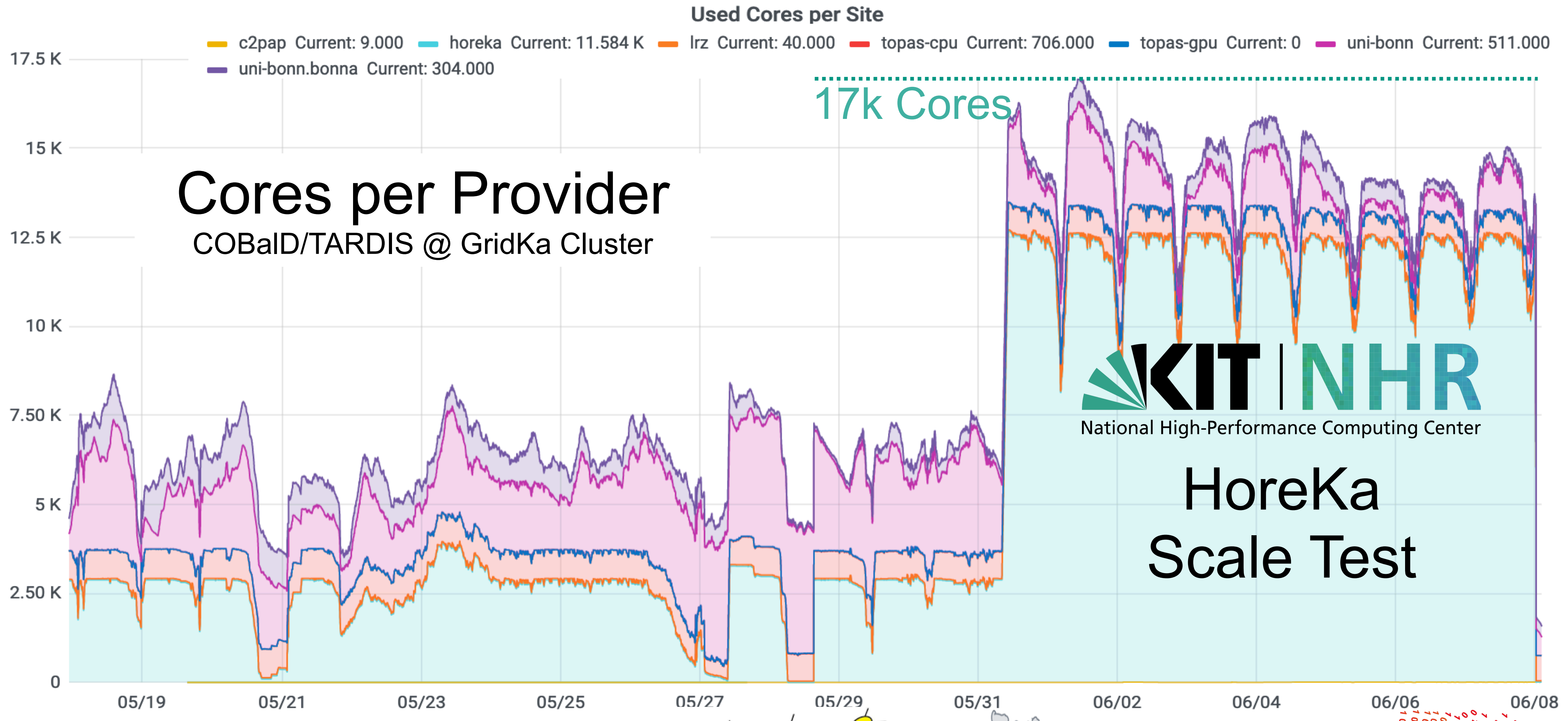
Use-cases so far …

Opportunistic Resources & WLCG in Practice

<u>Simplify provisioning and utilization of third-party compute resources for the GridKa communities:</u>

- Dynamic, transparent and on-demand integration via COBalD/TARDIS (in-house development)
- Provide community-overarching unified entry points to a variety of resources (HPCs, Clouds, …)
- Demonstrated production scale operation during scale test together with HoreKa (KIT HPC cluster)
- Production deployments across HEP institutes & HPC resources coordinated by KIT/GridKa
- Site specific accounting is now also possible with AUDITOR

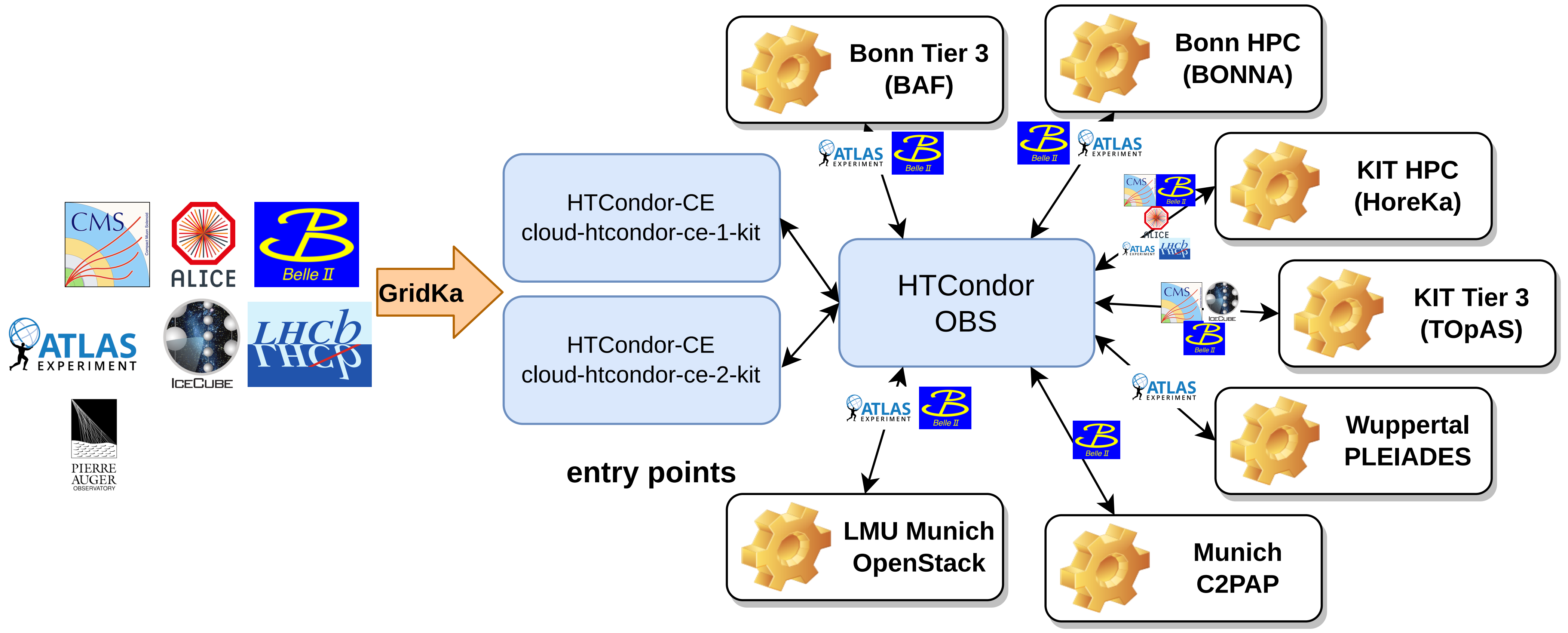


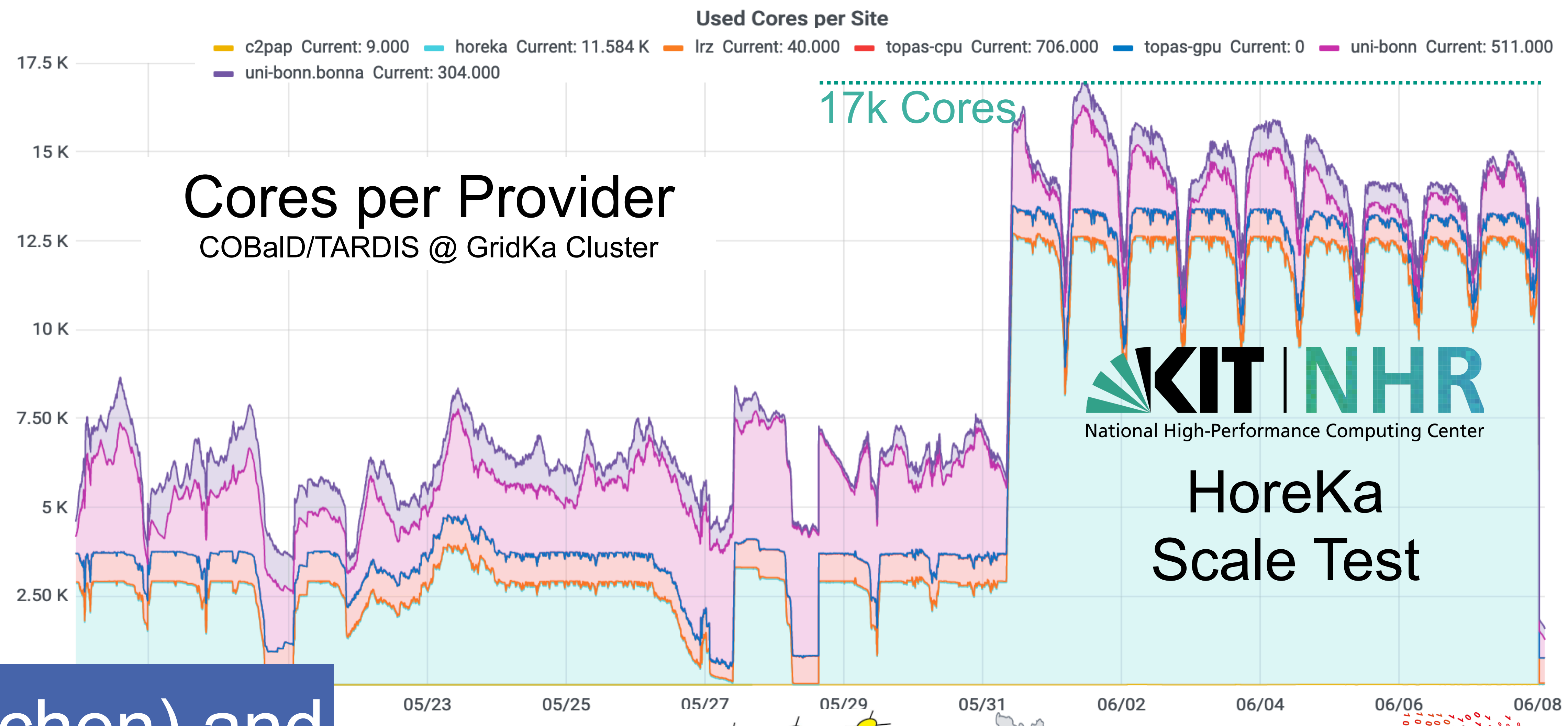


Opportunistic Resources & WLCG in Practice

<u>Simplify provisioning and utilization of third-party compute resources for the GridKa communities:</u>

- Dynamic, transparent and on-demand integration via COBalD/TARDIS (in-house development)
- Provide community-overarching unified entry points to a variety of resources (HPCs, Clouds, …)
- Demonstrated production scale operation during scale test together with HoreKa (KIT HPC cluster)
- Production deployments across HEP institutes & HPC resources coordinated by KIT/GridKa
- Site specific accounting is now also possible with AUDITOR





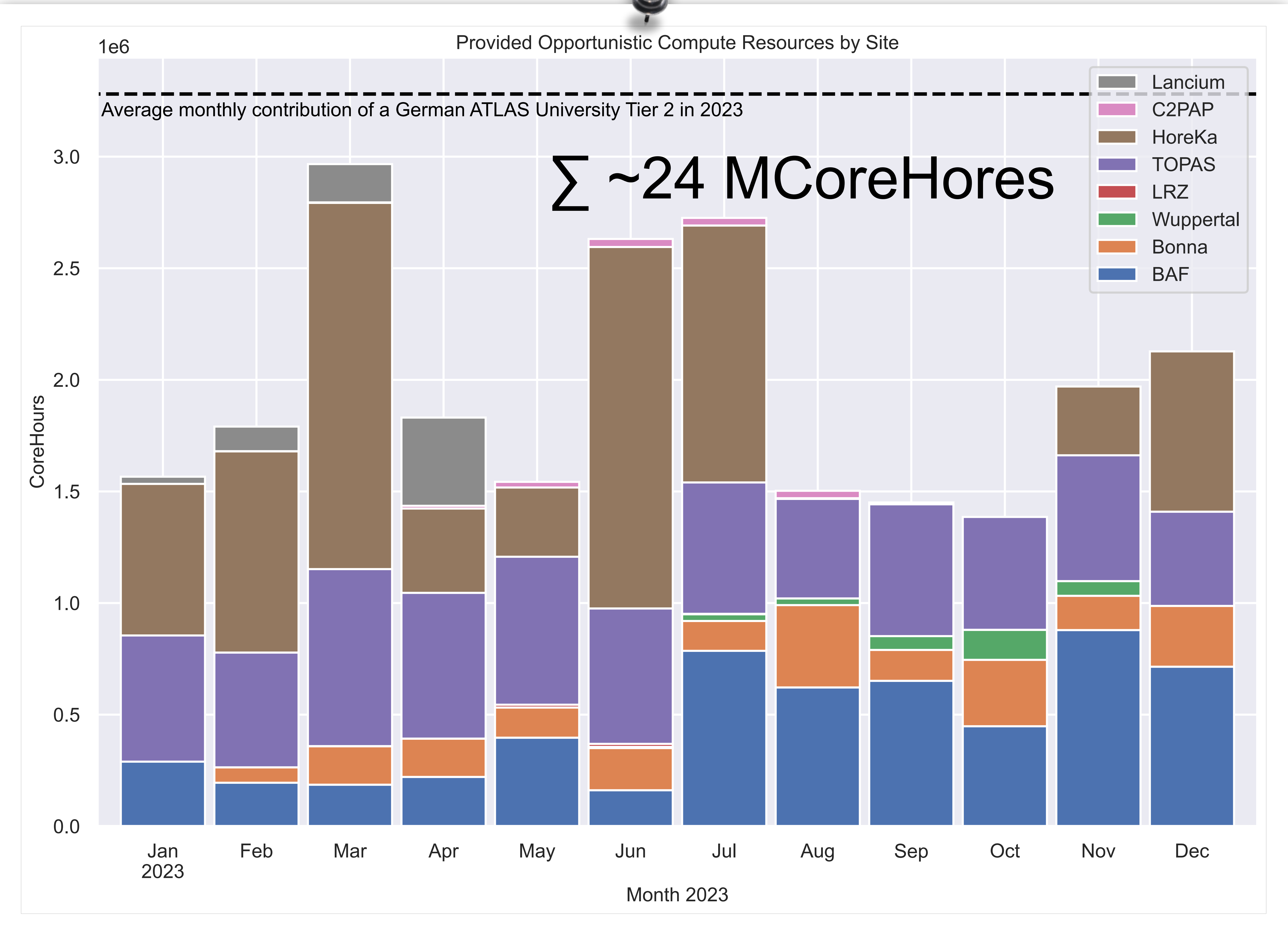
Similar setup deployed at CLAIX HPC (RWTH Aachen) and on-going deployment at Emmy (University of Göttingen)

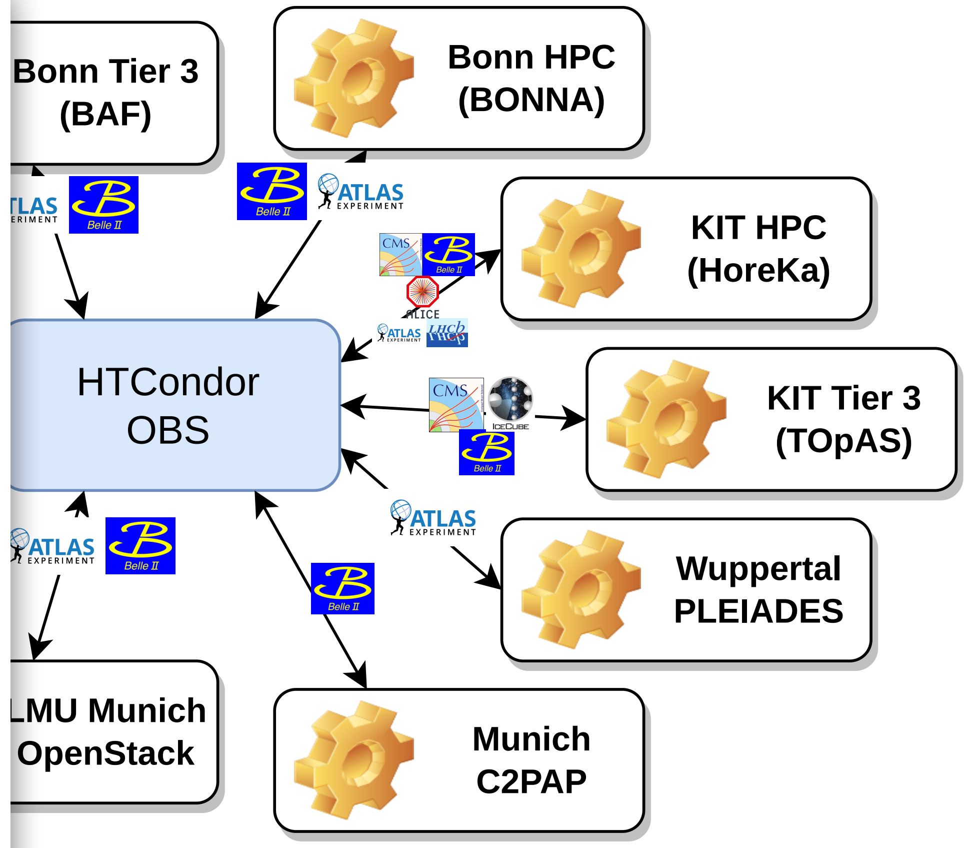
Opportunistic Resources & WLCG in Practice

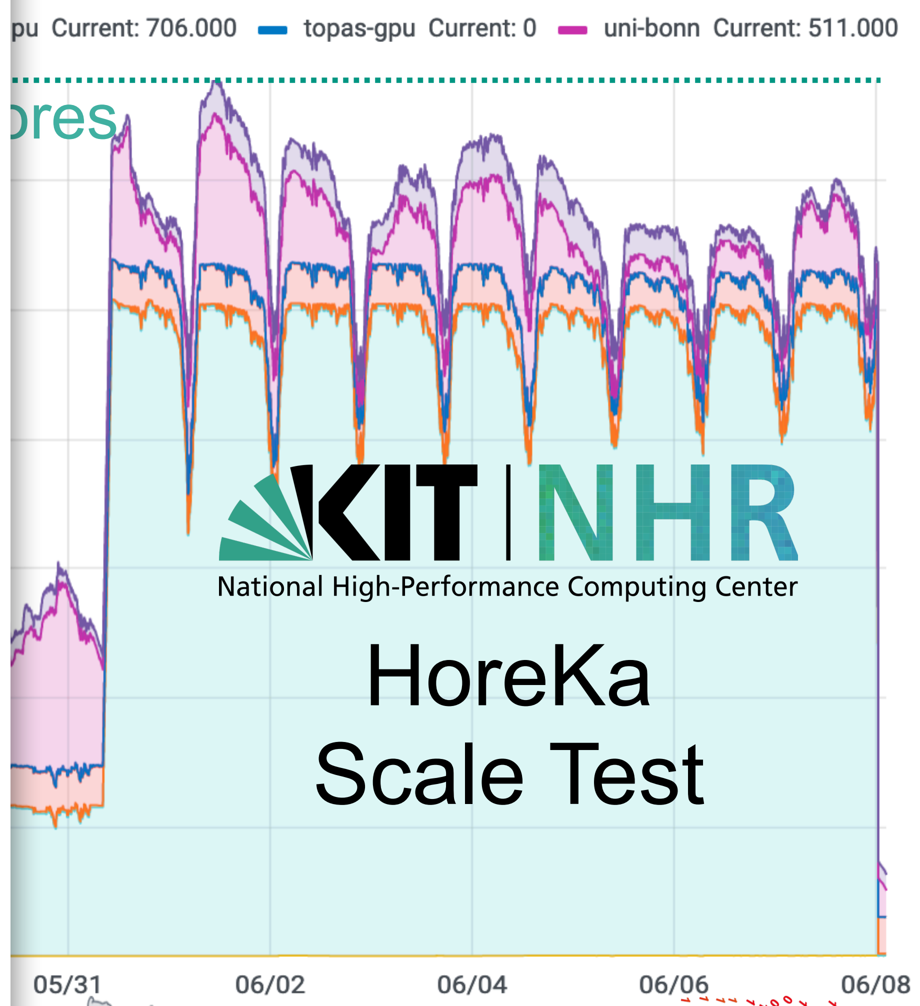
Karlsruhe Institute of Technology

Simplify provisioning
compute resources f

- Dynamic, transpar
  via COBalD/TARD
- Provide communit
  points to a variety
- Demonstrated pro
  scale test togethe
- Production deploy
  HPC resources co
- Site specific accou
  AUDITOR



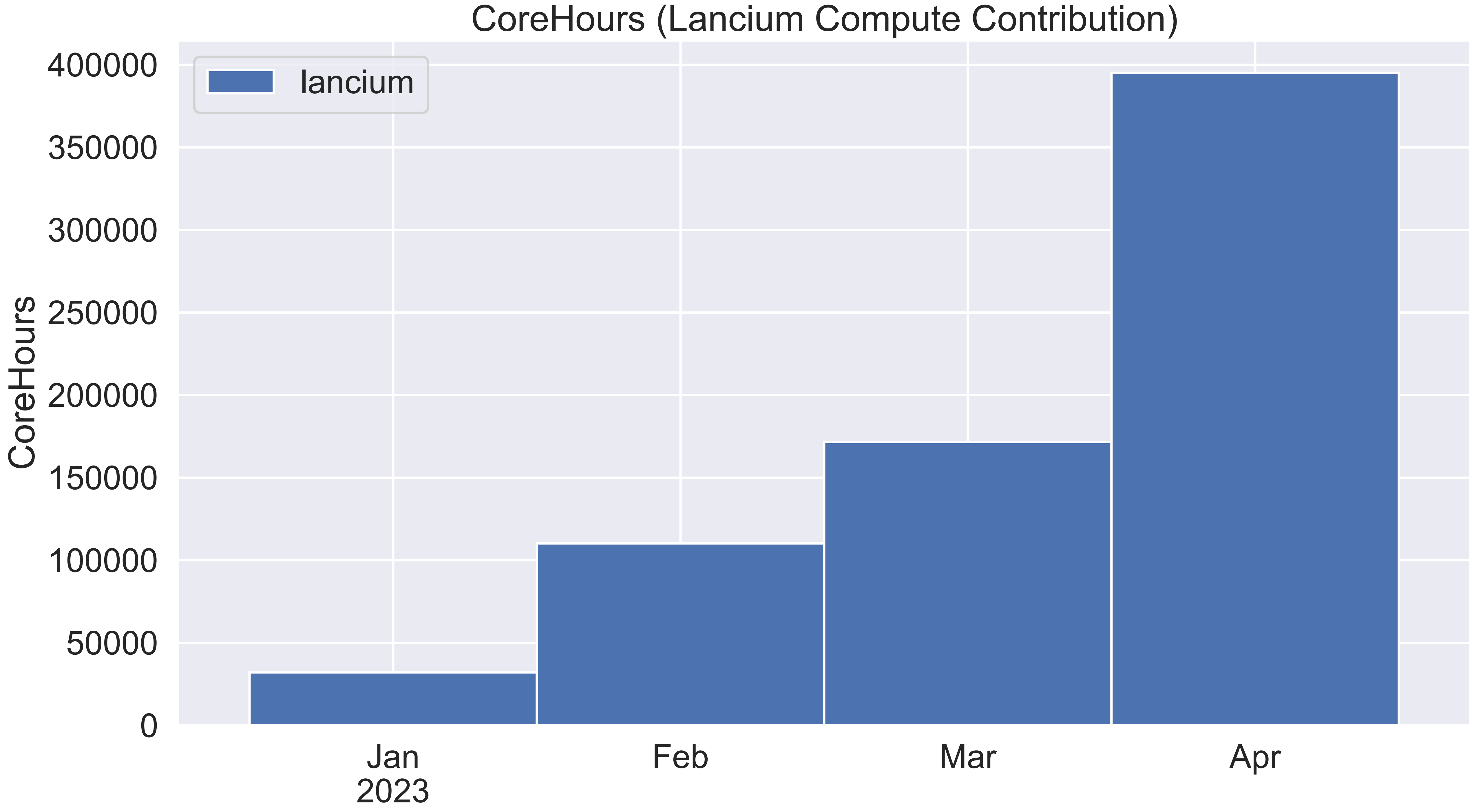


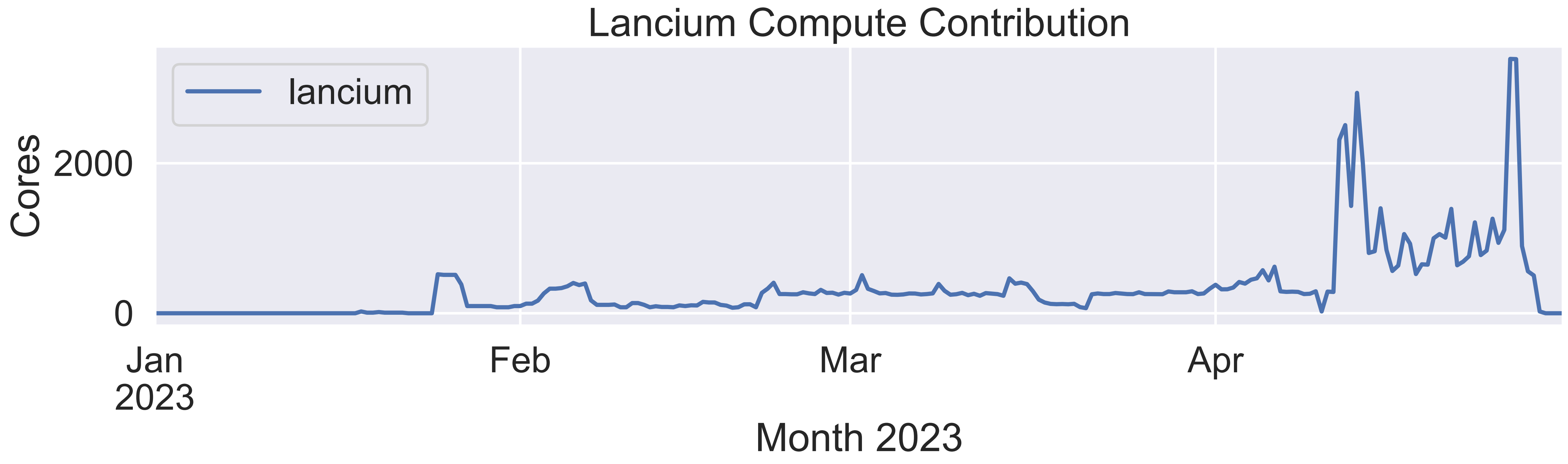


Similar setup deplo
on-going deployment at Emmy (University of Göttingen)

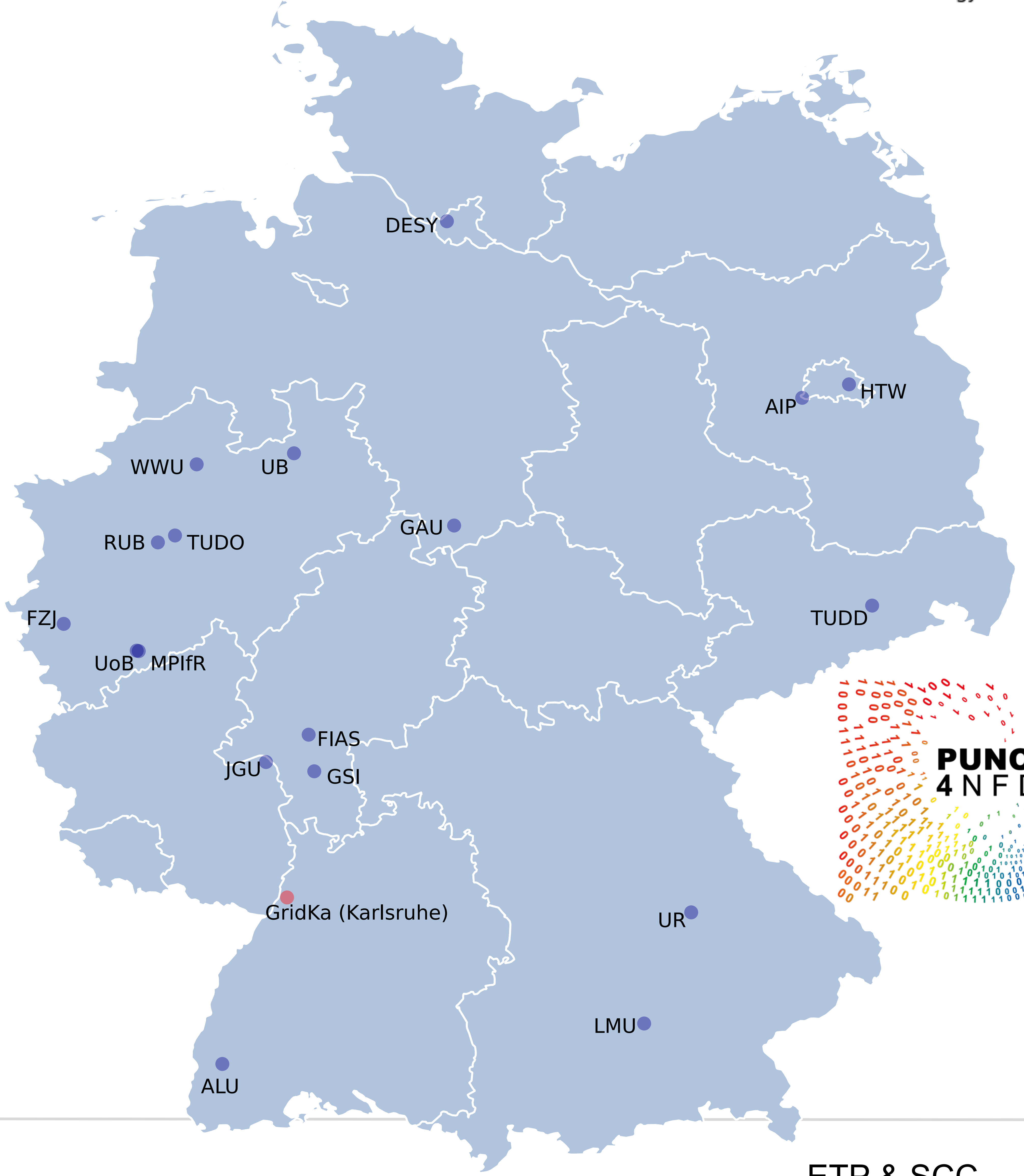
Enabling Access to Sustainable Compute Resources

- Lancium (US company) balancing the power grid by operating compute facilities close to renewables (wind & solar) - CO_2 neutral operation
- Dynamic, transparent and on-demand integration via COBalD/TARDIS
- Used for ATLAS/CMS MC generation (~700,000 CoreHours during PoC)
- Very smooth „Proof of Concept“ project, experiments did not even noticed that the jobs ran in the US
- Unfortunately, Lancium decided to get out of the PaaS business in April 2023



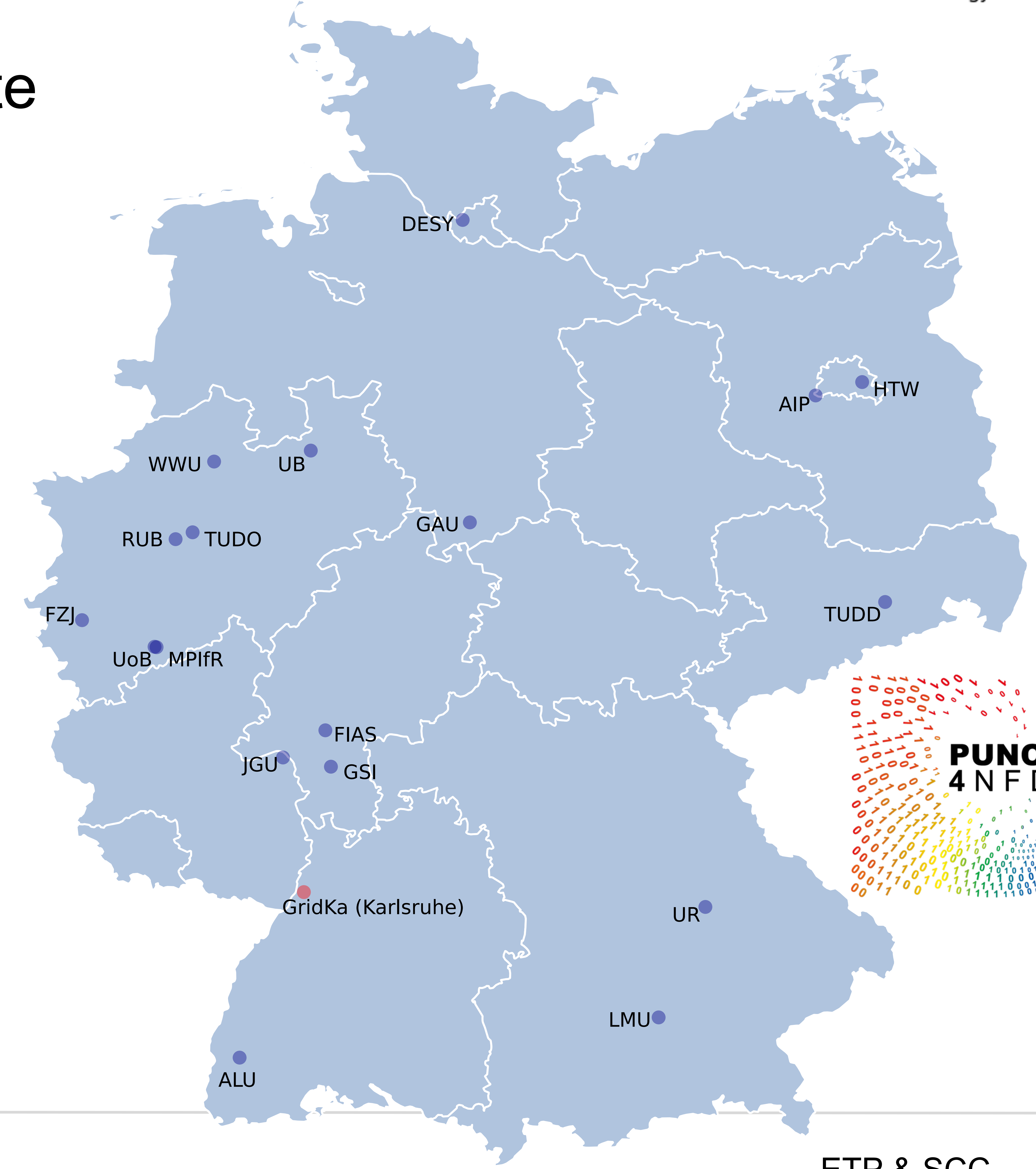


Towards the Compute4PUNCH Infrastructure



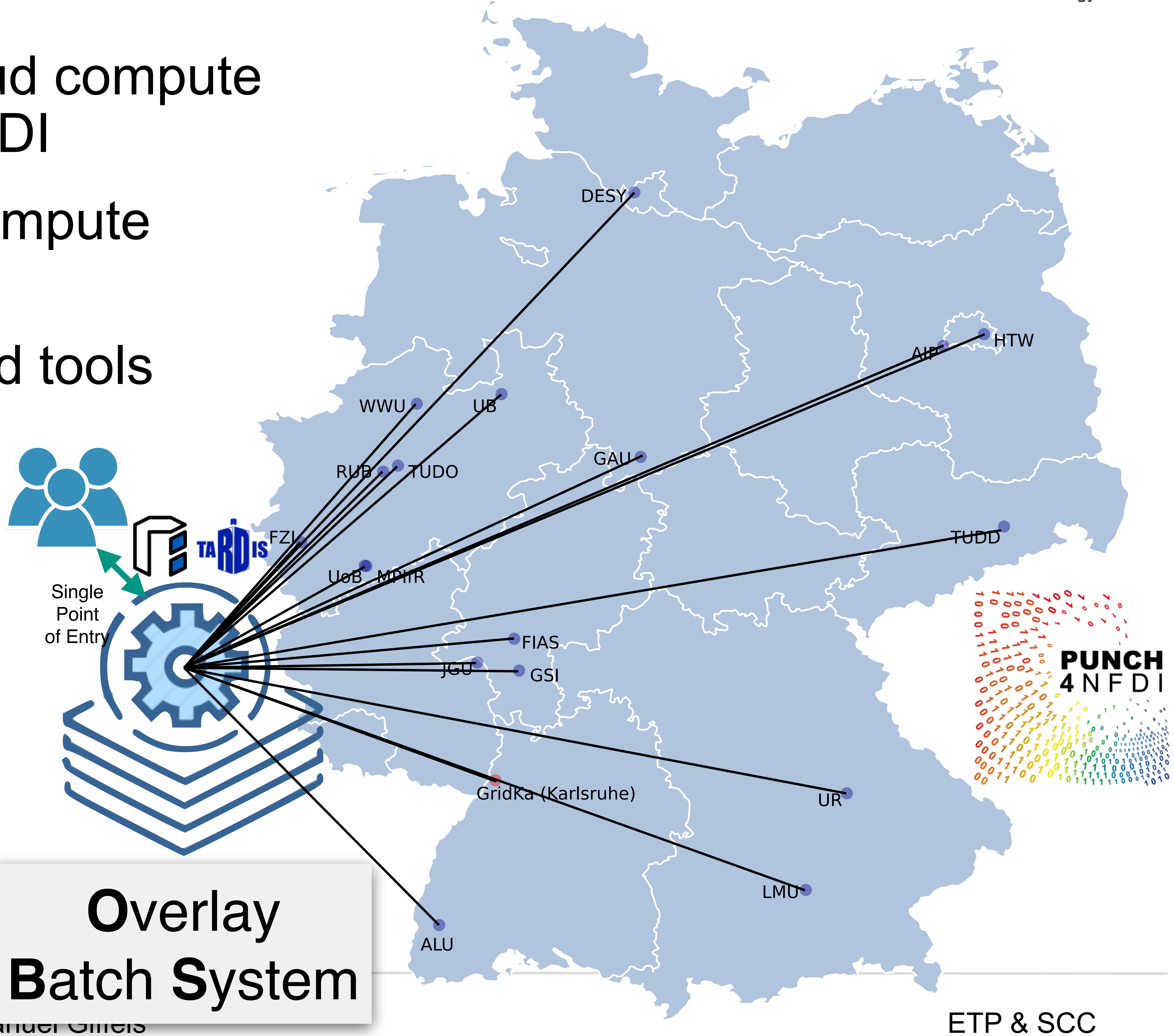
Towards the Compute4PUNCH Infrastructure

- Substantial amount of HTC, HPC, Cloud compute resources are provided to PUNCH4NFDI



Towards the Compute4PUNCH Infrastructure

- Substantial amount of HTC, HPC, Cloud compute resources are provided to PUNCH4NFDI
- Establish a federated heterogenous compute infrastructure for PUNCH4NFDI
- Benefit from experiences, concepts and tools available in HEP community

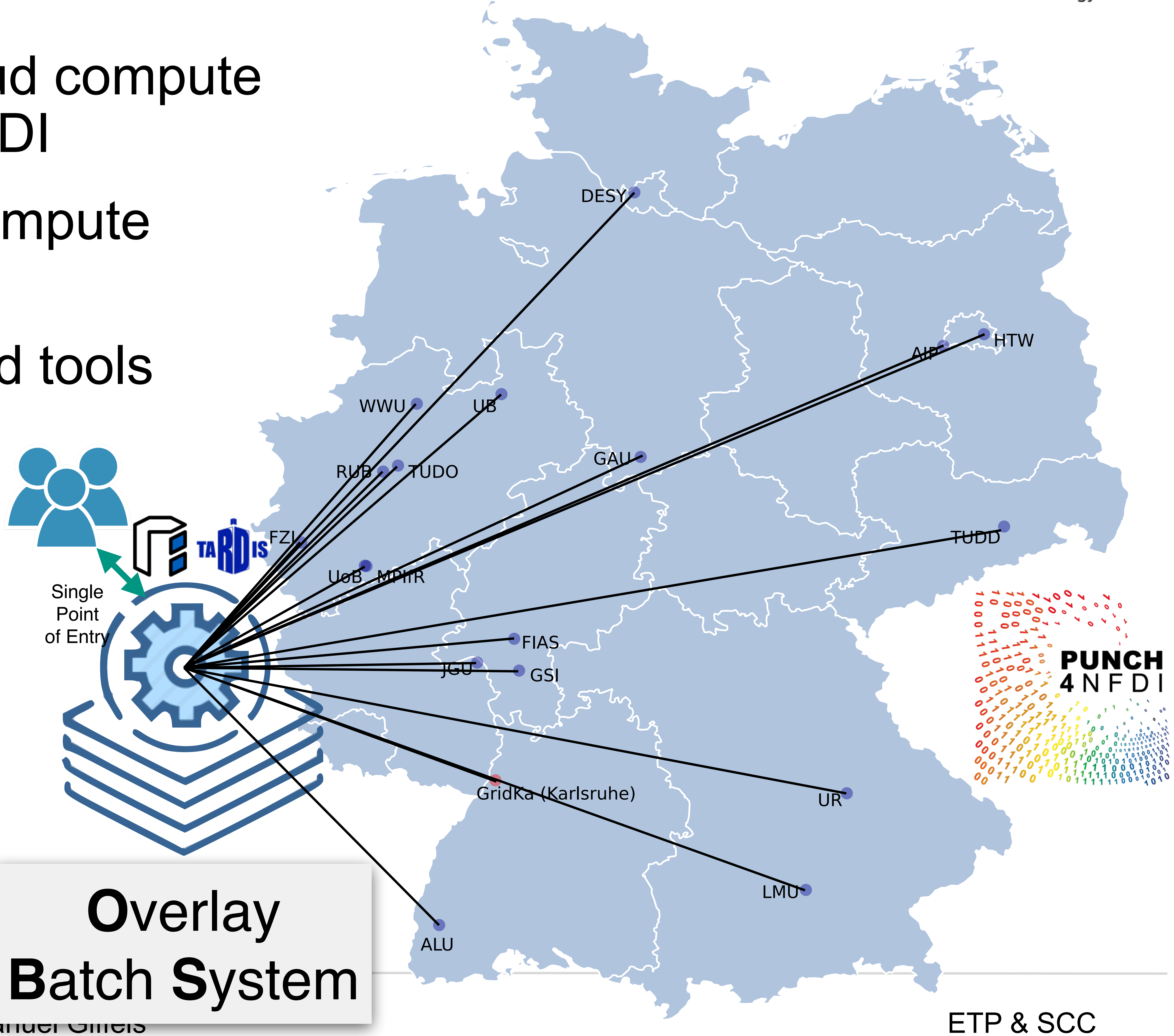


Towards the Compute4PUNCH Infrastructure

- Substantial amount of HTC, HPC, Cloud compute resources are provided to PUNCH4NFDI
- Establish a federated heterogenous compute infrastructure for PUNCH4NFDI
- Benefit from experiences, concepts and tools available in HEP community

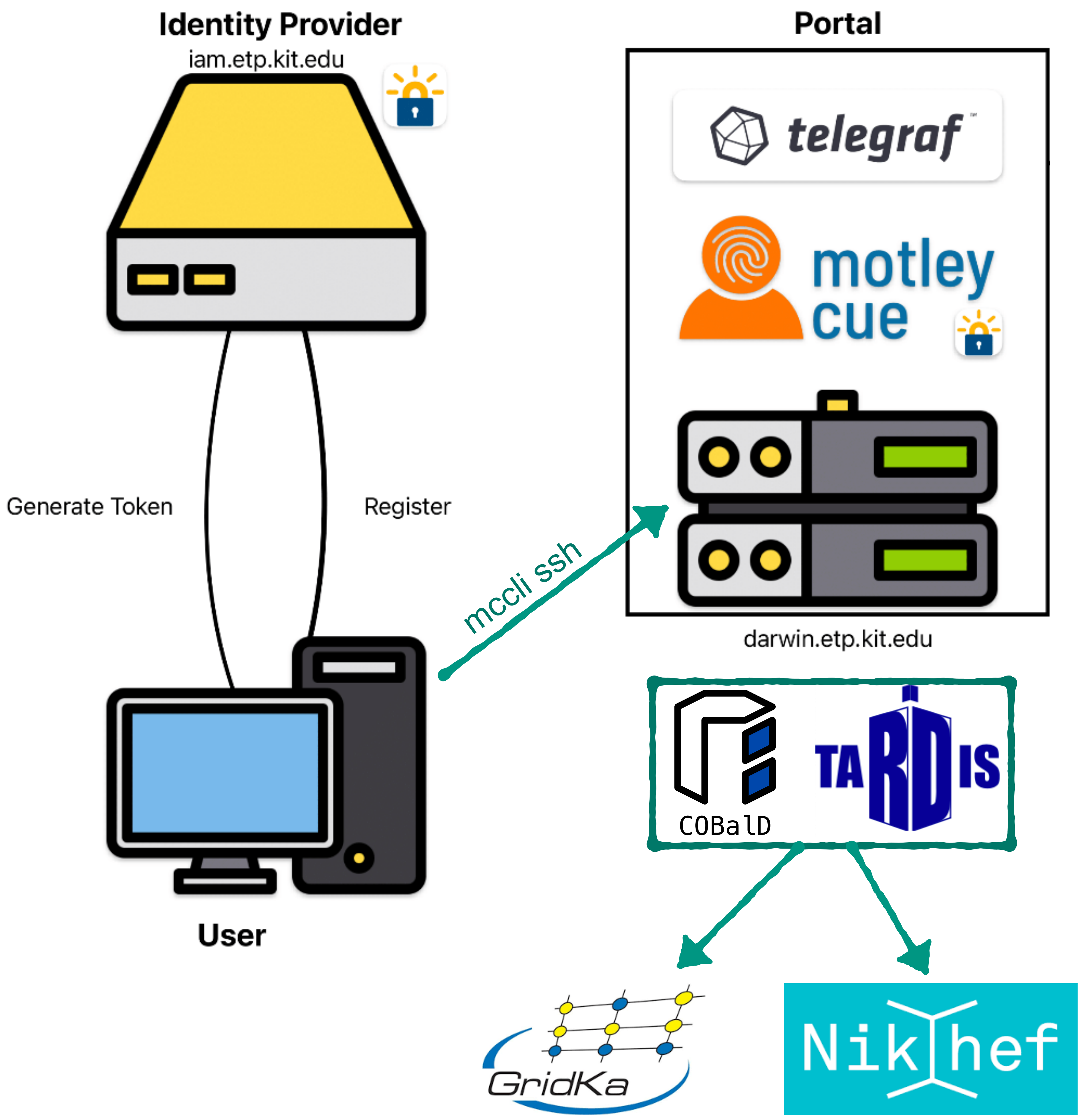
---

- Compute4PUNCH demonstrator is available
- Demonstration workflows of HEP (ATLAS/CMS), Astrophysics (LOFAR) and Lattice QCD have been successfully performed



Building a Computing Infrastructure for DARWIN

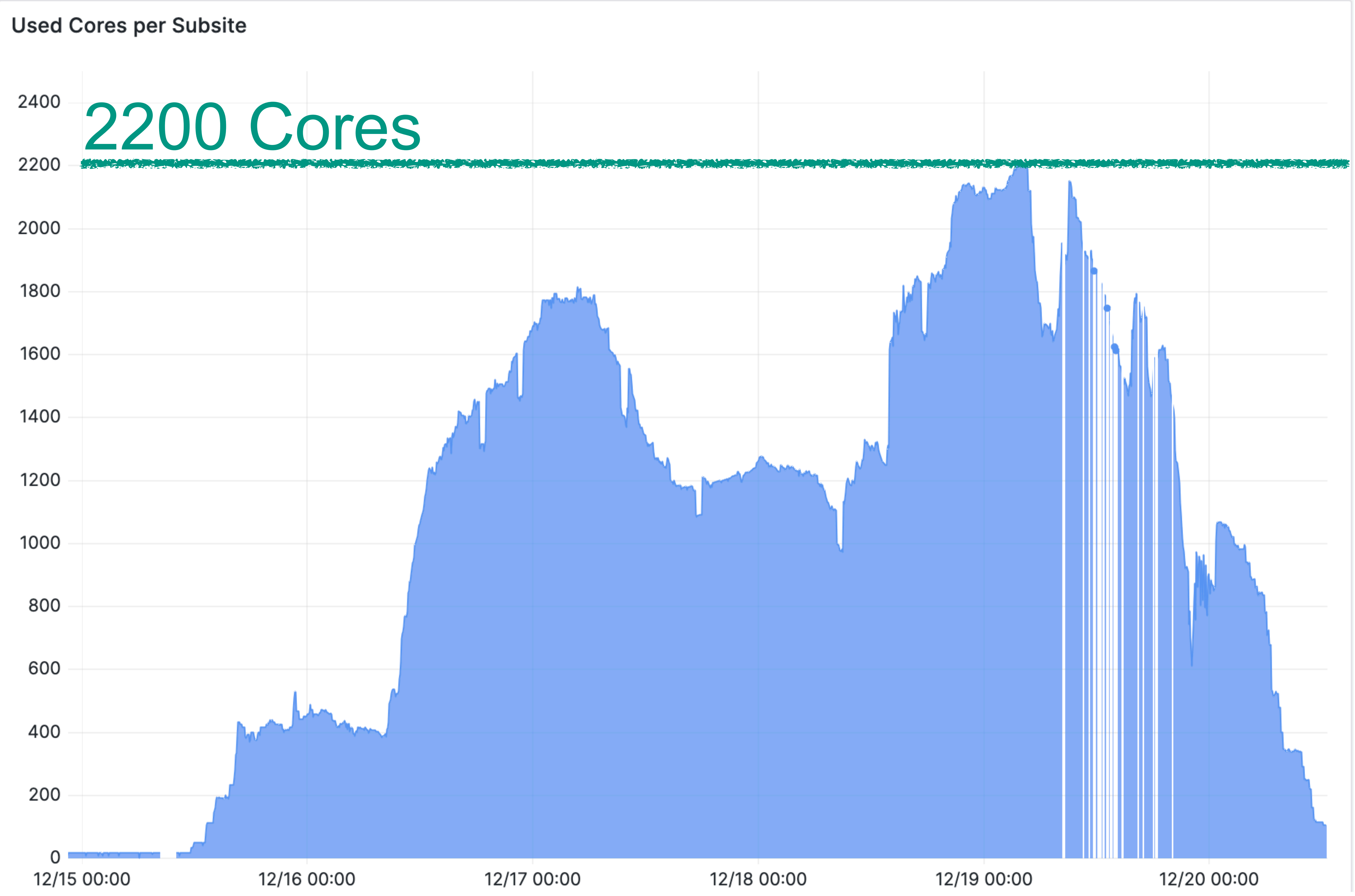
- Manage VO via Indigo Identity and access management service (IAM)
- Provide token based access via ssh to a login node (using Motley Cue)
- Provide dedicated JupyterHub for interactive data analysis
- Integrate external Grid resources using C/T as a pilot factory [GridKa & Nikhef (ongoing)]
- All resources a available via an HTCondor OBS on the login node



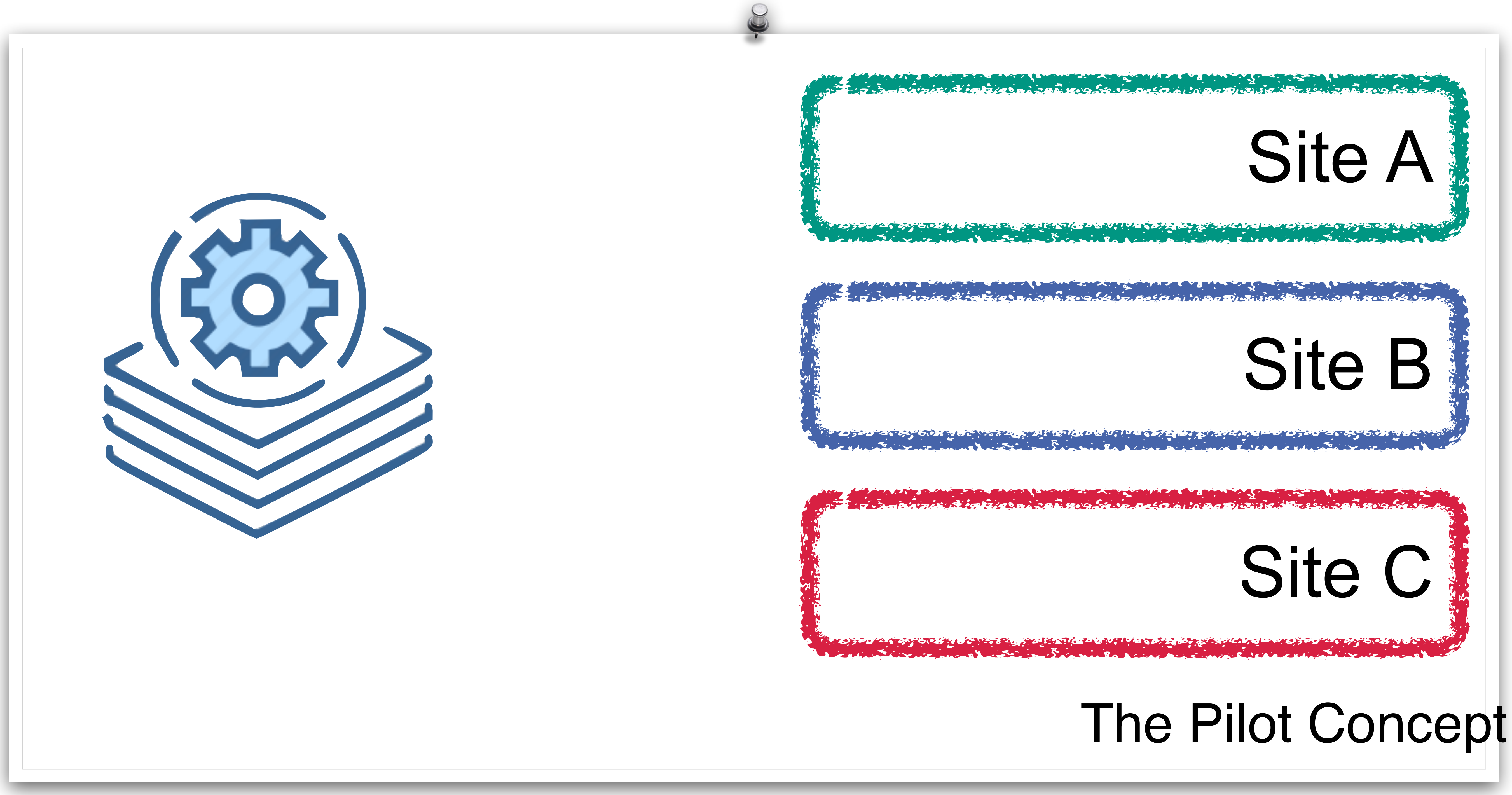
work by Sebastian Brommer (KIT)

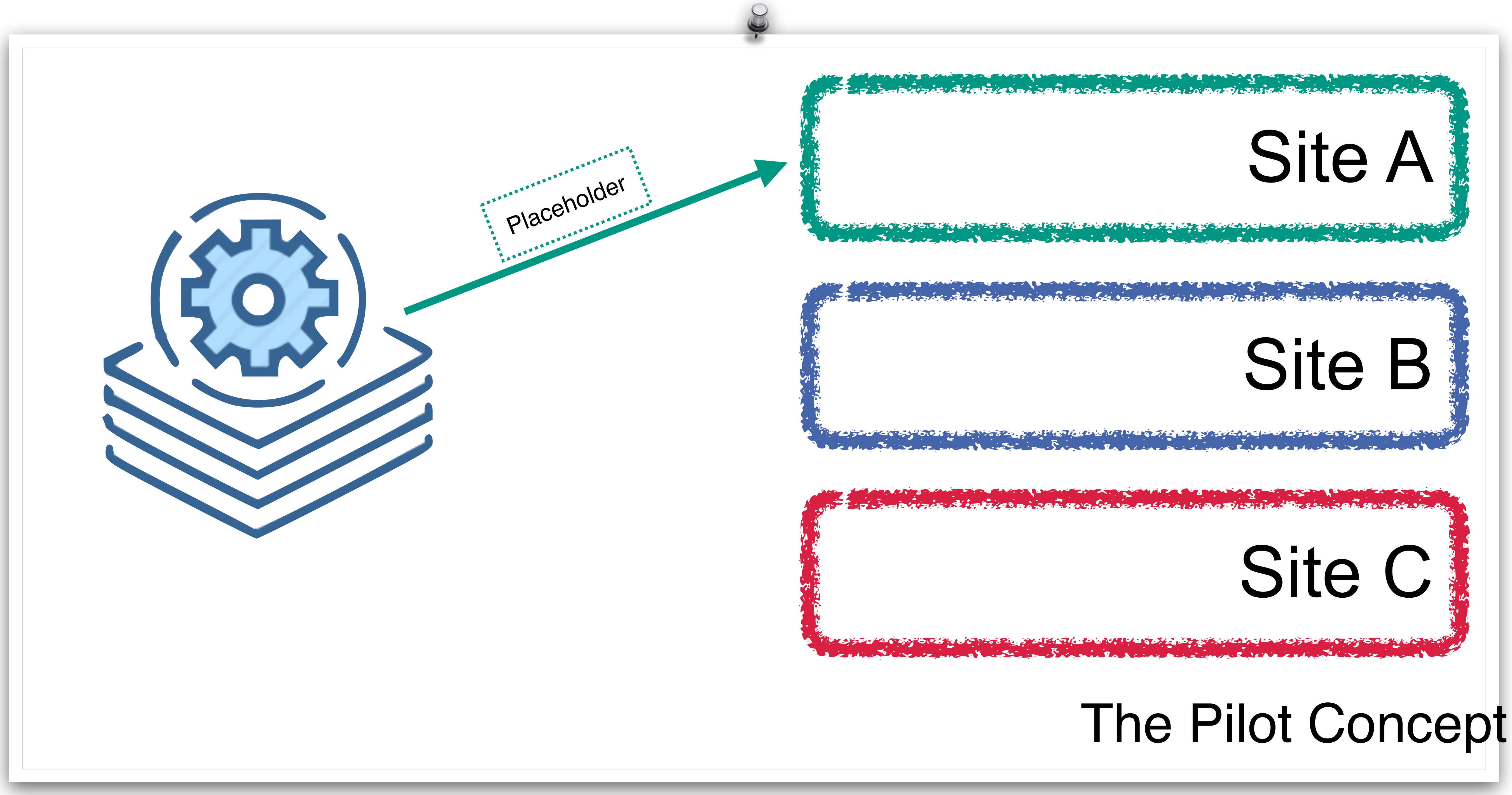
Remote Operation of the LMU Tier-2

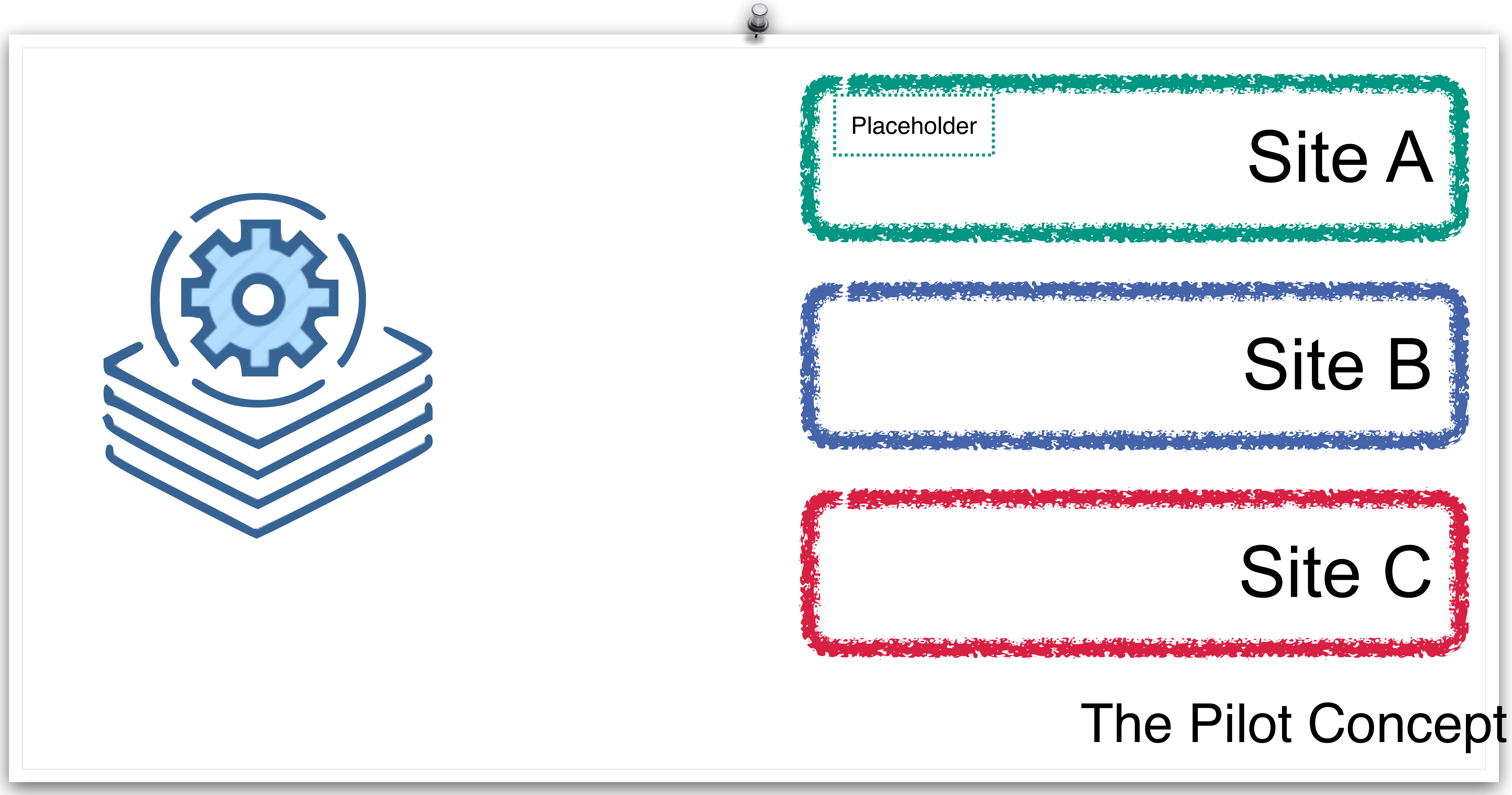
- In December 2023 Rod contacted us about a nice PoC idea (12.12.2023)
- There was a week long scheduled storage downtime at the LMU Tier-2
- So, how about integrating the LMU Tier-2 workers into the opportunistic compute cloud operated at GridKa?
- Rod was able to quickly set-up the C/T ecosystem at LMU supported by KIT
- During the downtime the LMU Tier-2 was fed with ATLAS jobs via GridKa (incl. remote data access)

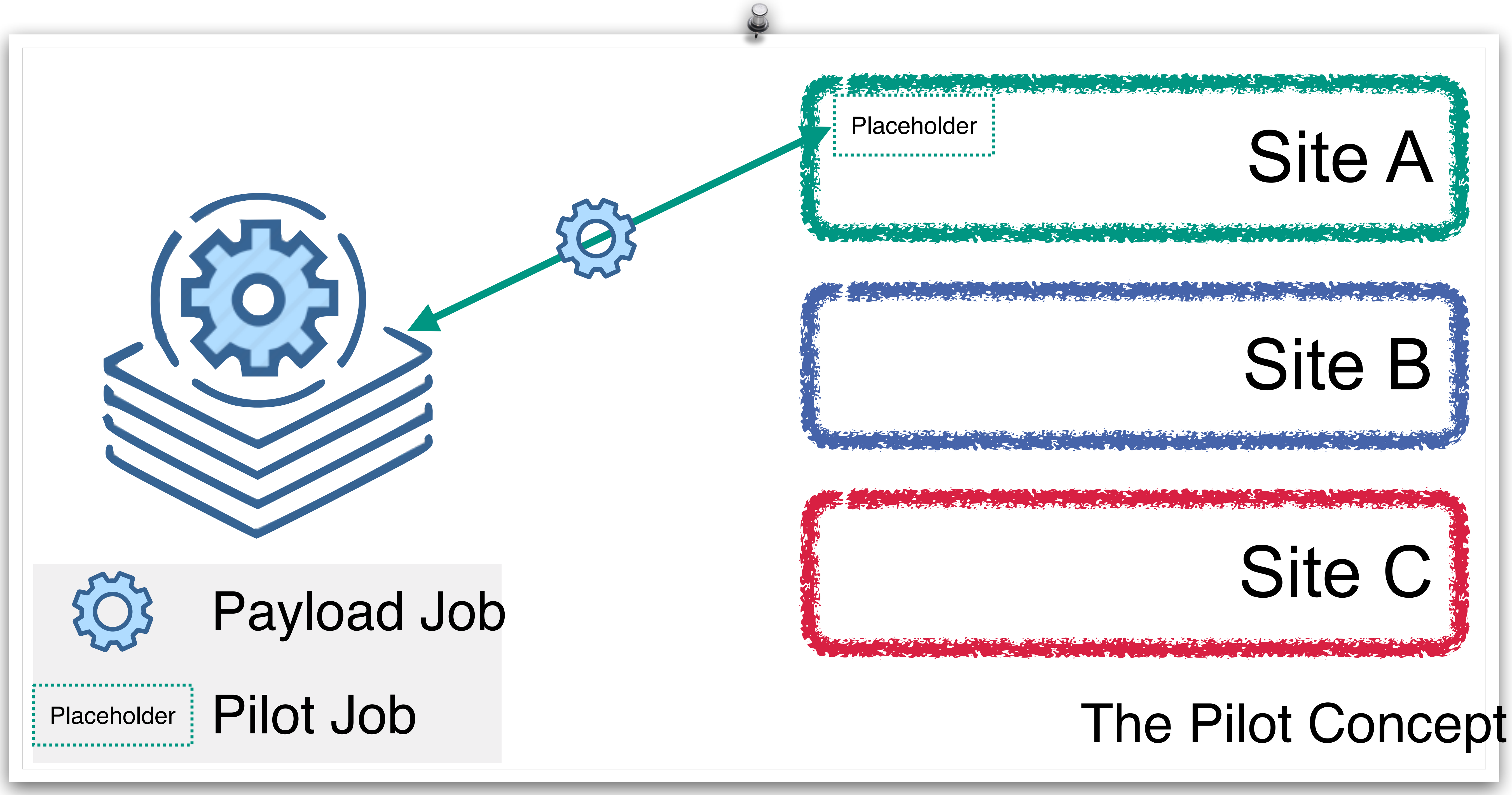


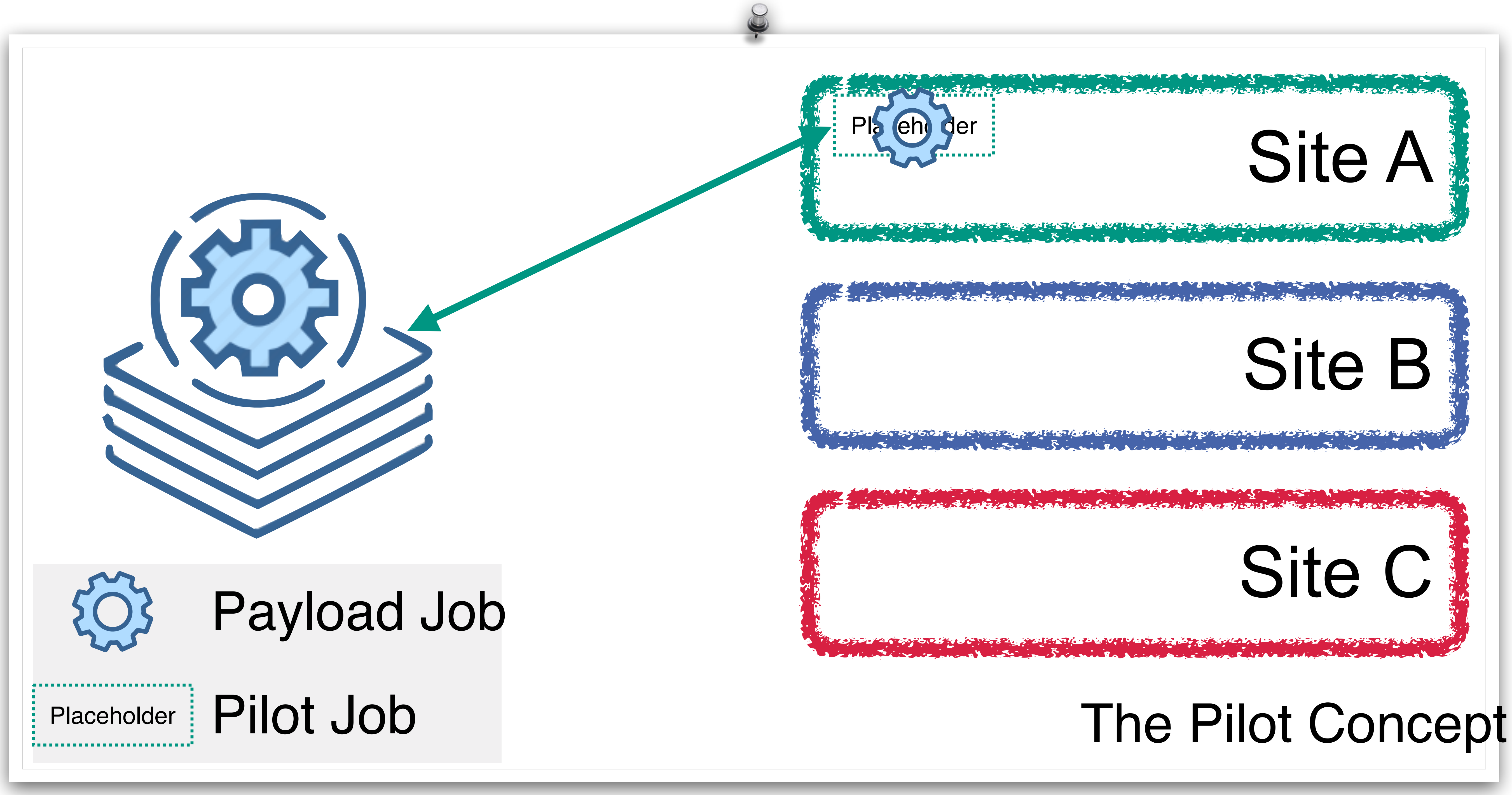
Backup

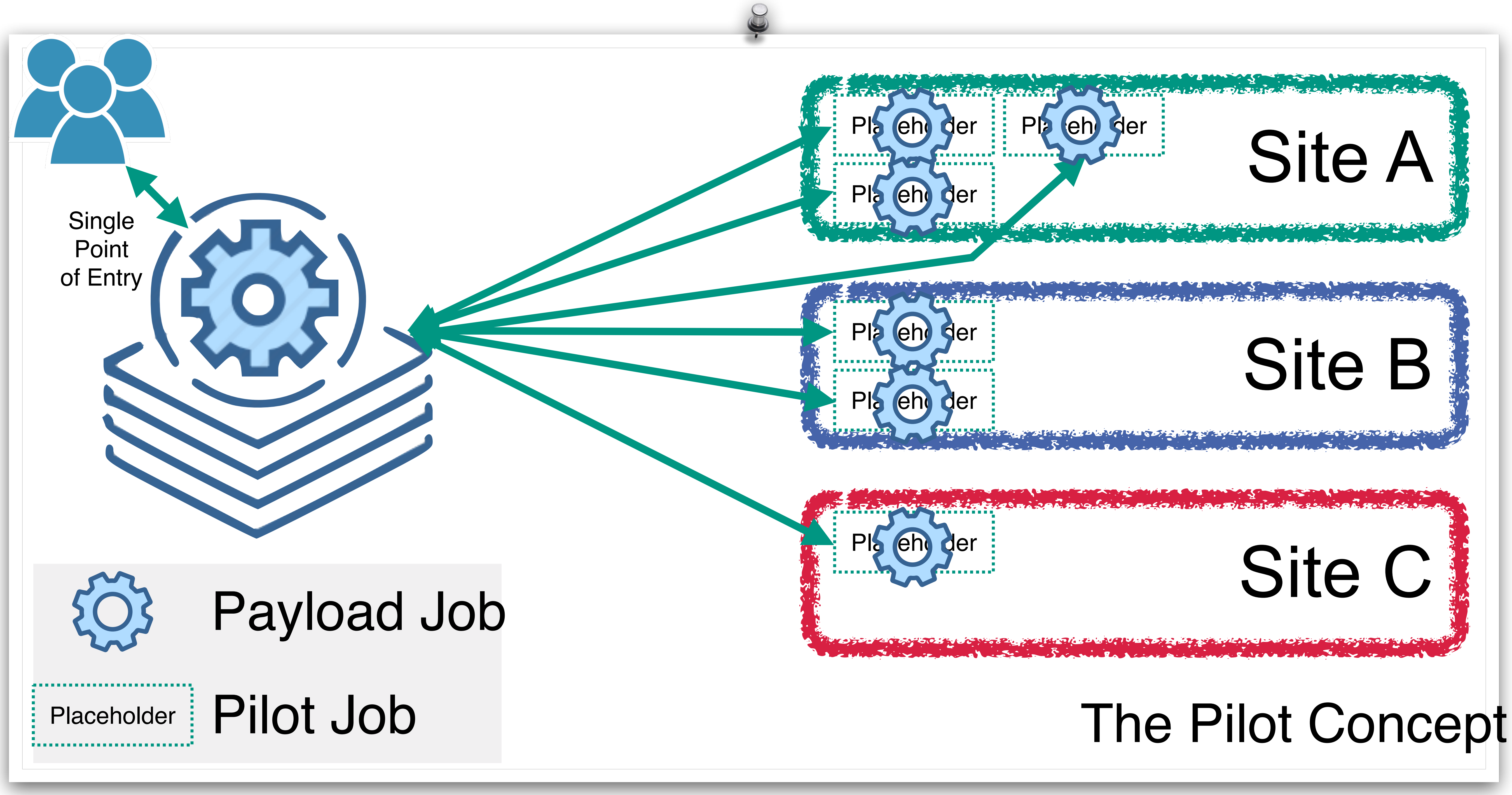

Site A
Site B
Site C
The Pilot Concept


Placeholder
Site A
Site B
Site C
The Pilot Concept


Placeholder
Site A
Site B
Site C
The Pilot Concept

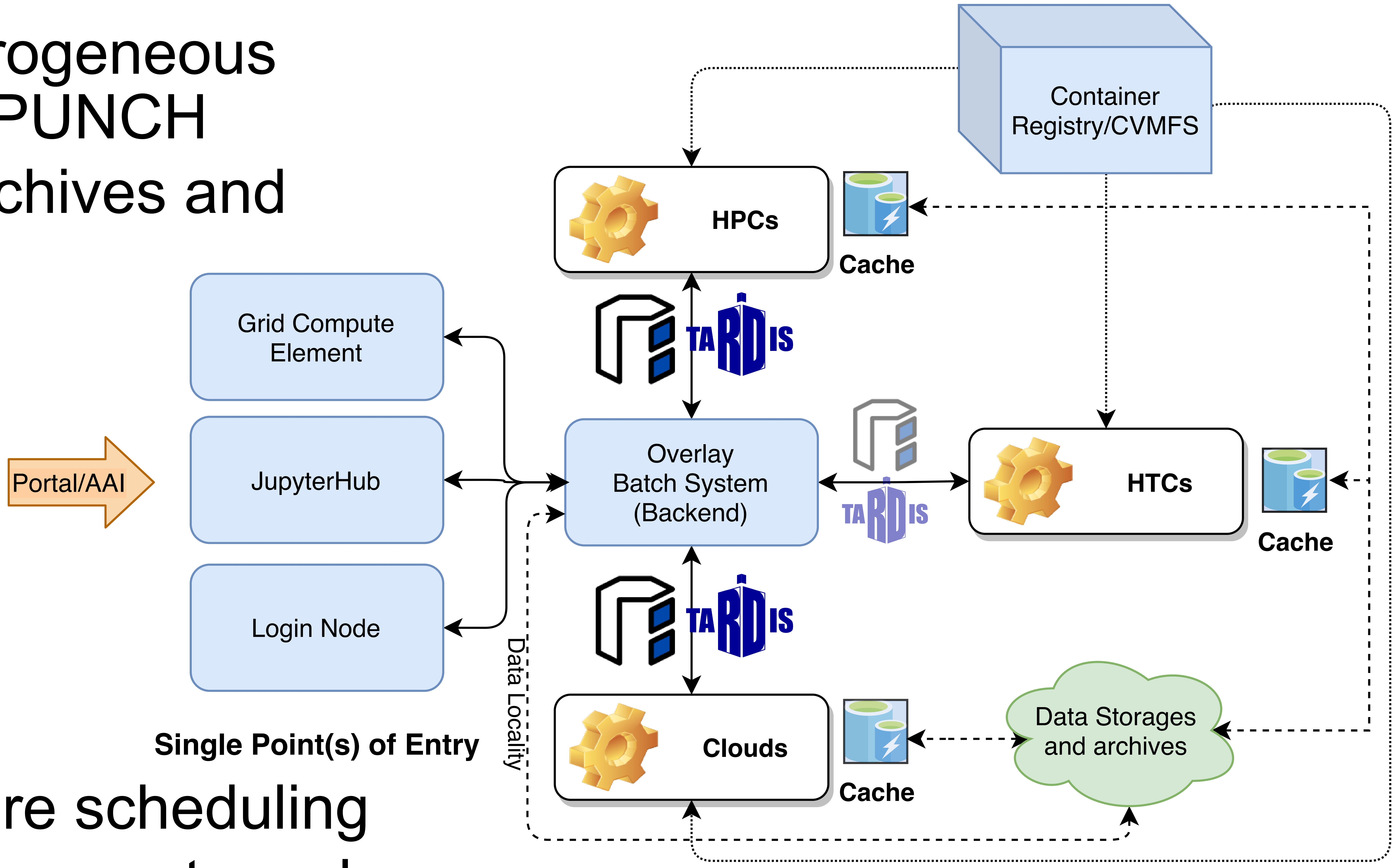

Placeholder
Site A
Site B
Site C
Payload Job
Placeholder
Pilot Job
The Pilot Concept


Placeholder
Site A
Site B
Site C
Payload Job
Placeholder
Pilot Job
The Pilot Concept


Single
Point
of Entry
Site A
Site B
Site C
Payload Job
Placeholder
Pilot Job
The Pilot Concept

Towards the Compute4PUNCH Infrastructure

- Establish a federated heterogeneous compute infrastructure for PUNCH
- Integrate data storages, archives and opportunistic caches



- Introduce data-locality aware scheduling
- Benefit from experiences, concepts and tools available in HEP community

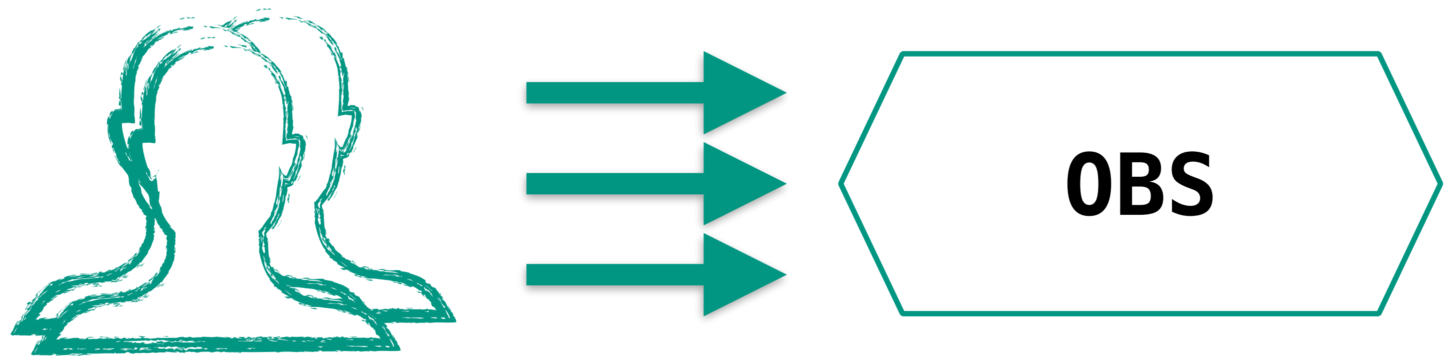
Resource Meta-Scheduler

Resource Meta-Scheduler

Classical Job to Resource to Job meta-scheduler:

Resource Meta-Scheduler

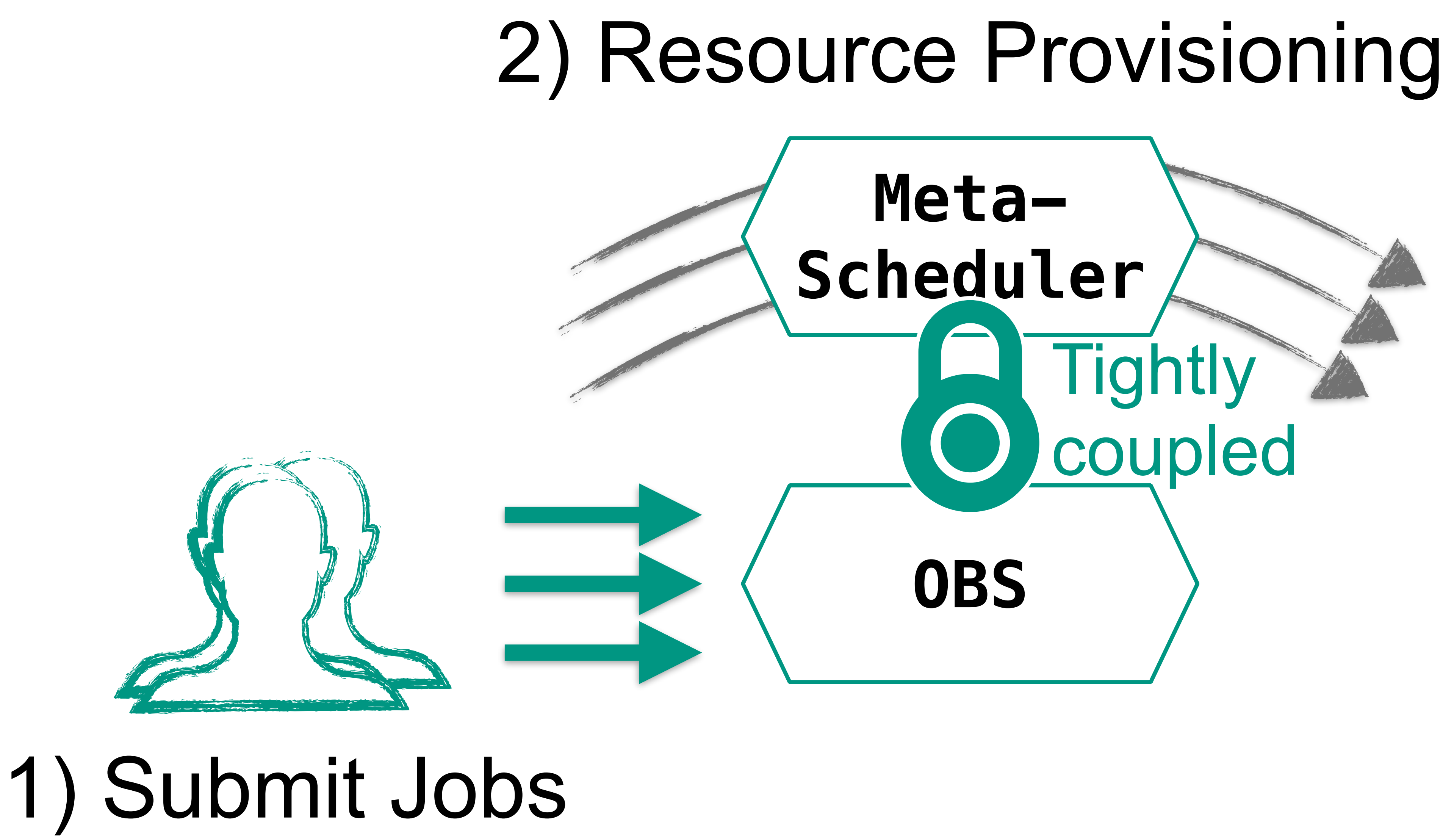
Classical Job to Resource to Job meta-scheduler:



1) Submit Jobs

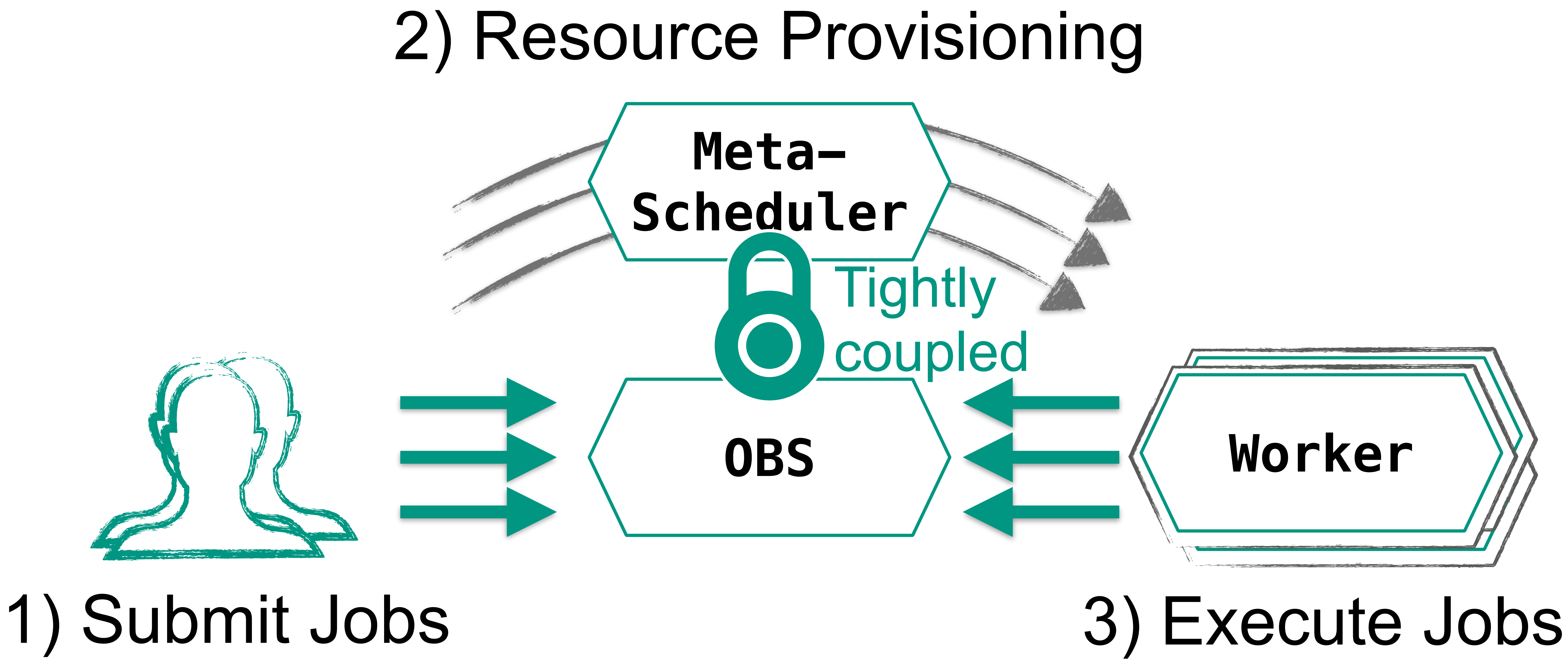
Resource Meta-Scheduler

Classical Job to Resource to Job meta-scheduler:



Resource Meta-Scheduler

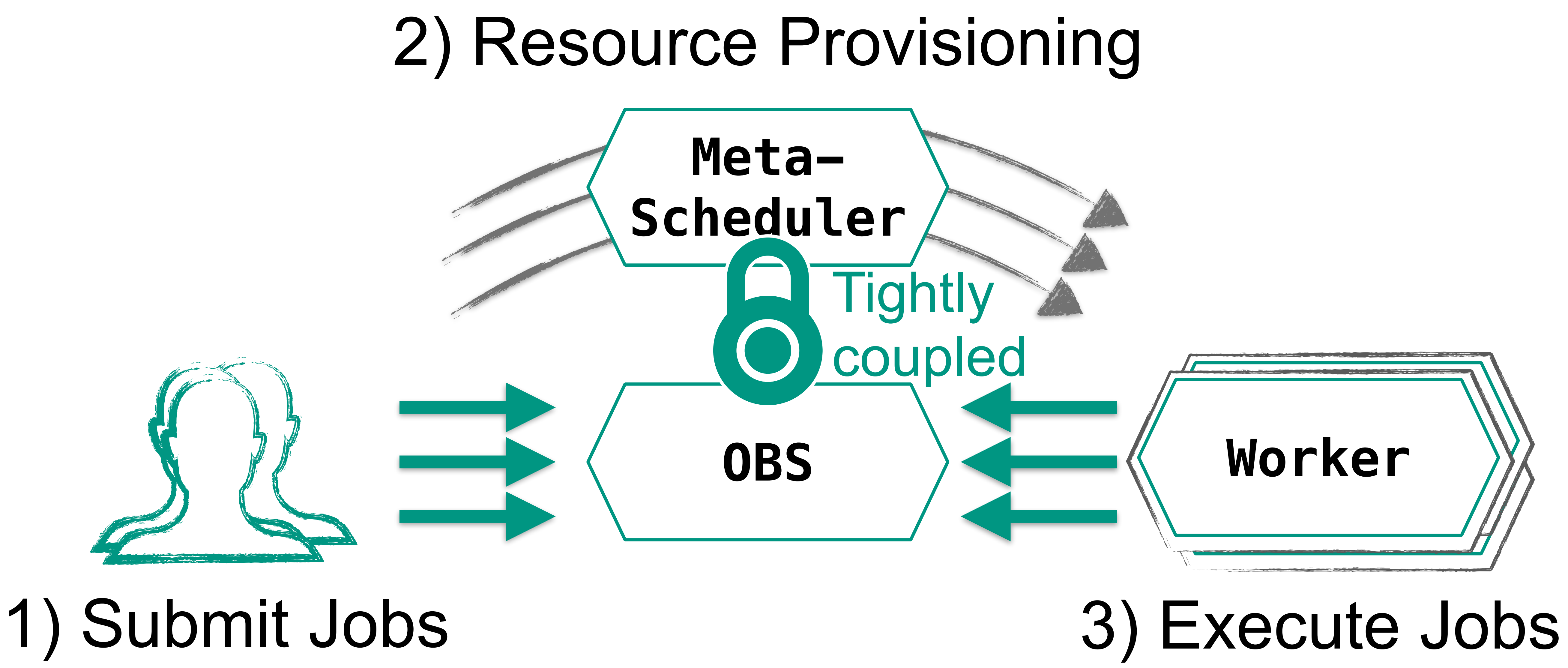
Classical Job to Resource to Job meta-scheduler:



Resource Meta-Scheduler

<u>Classical Job to Resource to Job meta-scheduler:</u>

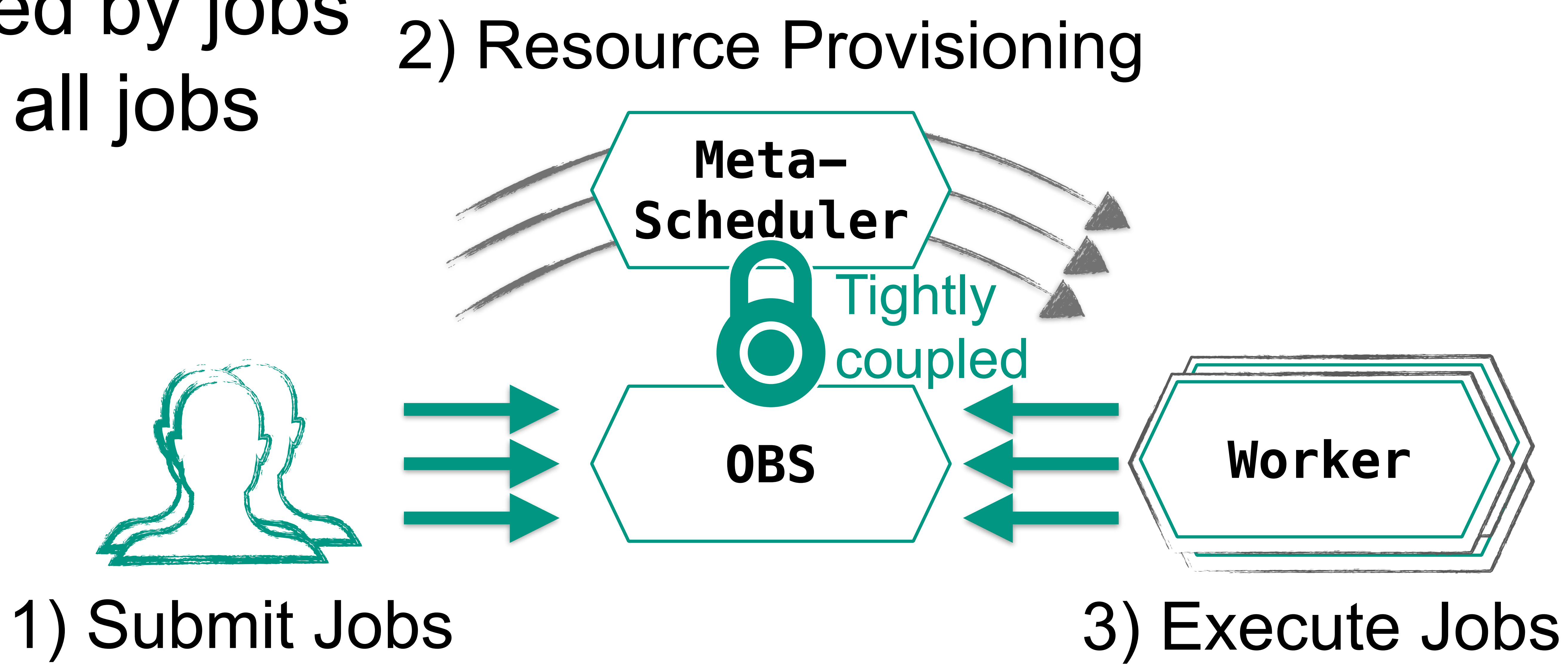
- Dynamic resource acquisition matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users



Resource Meta-Scheduler

Classical Job to Resource to Job meta-scheduler:

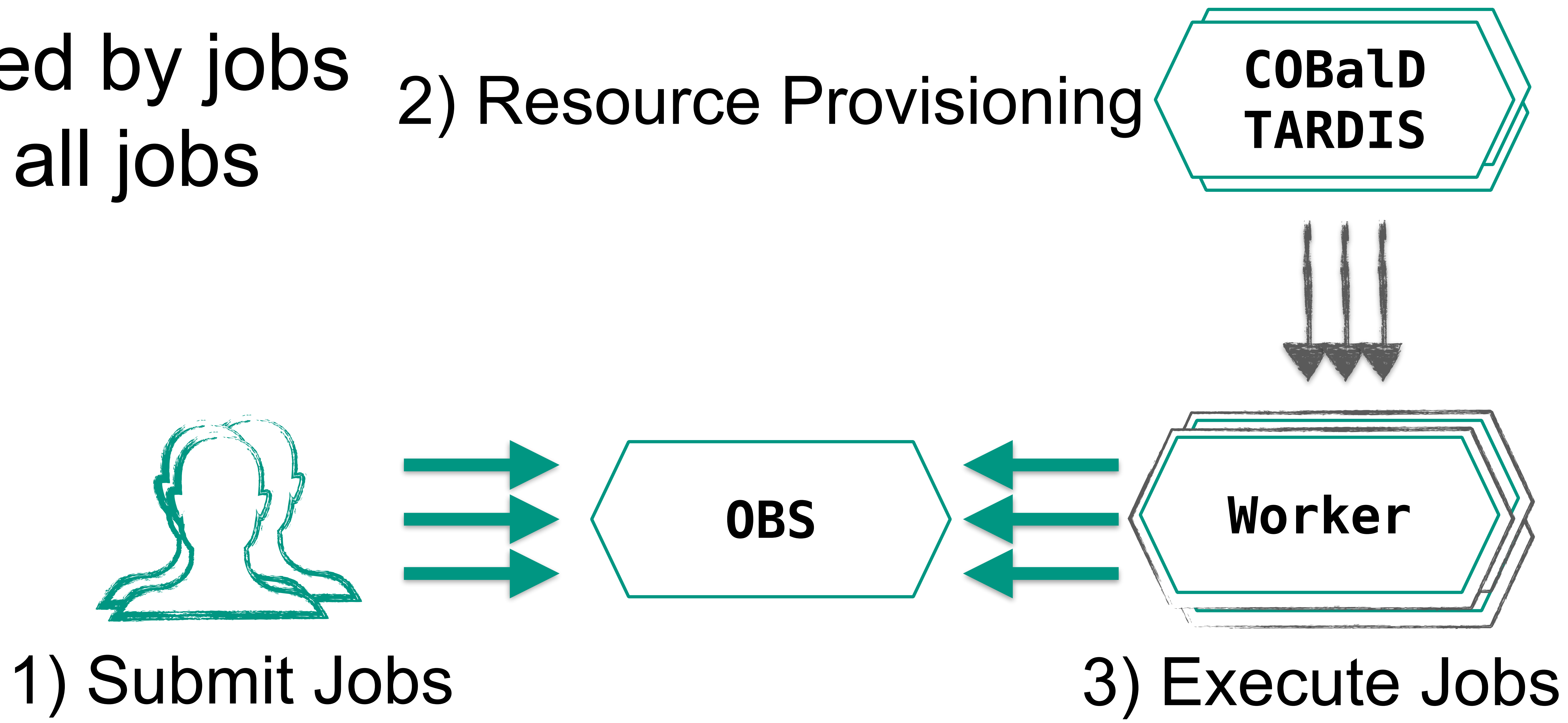
- Dynamic resource acquisition matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users
- Job scheduling in overlay batch system
  - Unreliable to predict resources used by jobs
  - Efficient to integrate resources for all jobs



Resource Meta-Scheduler

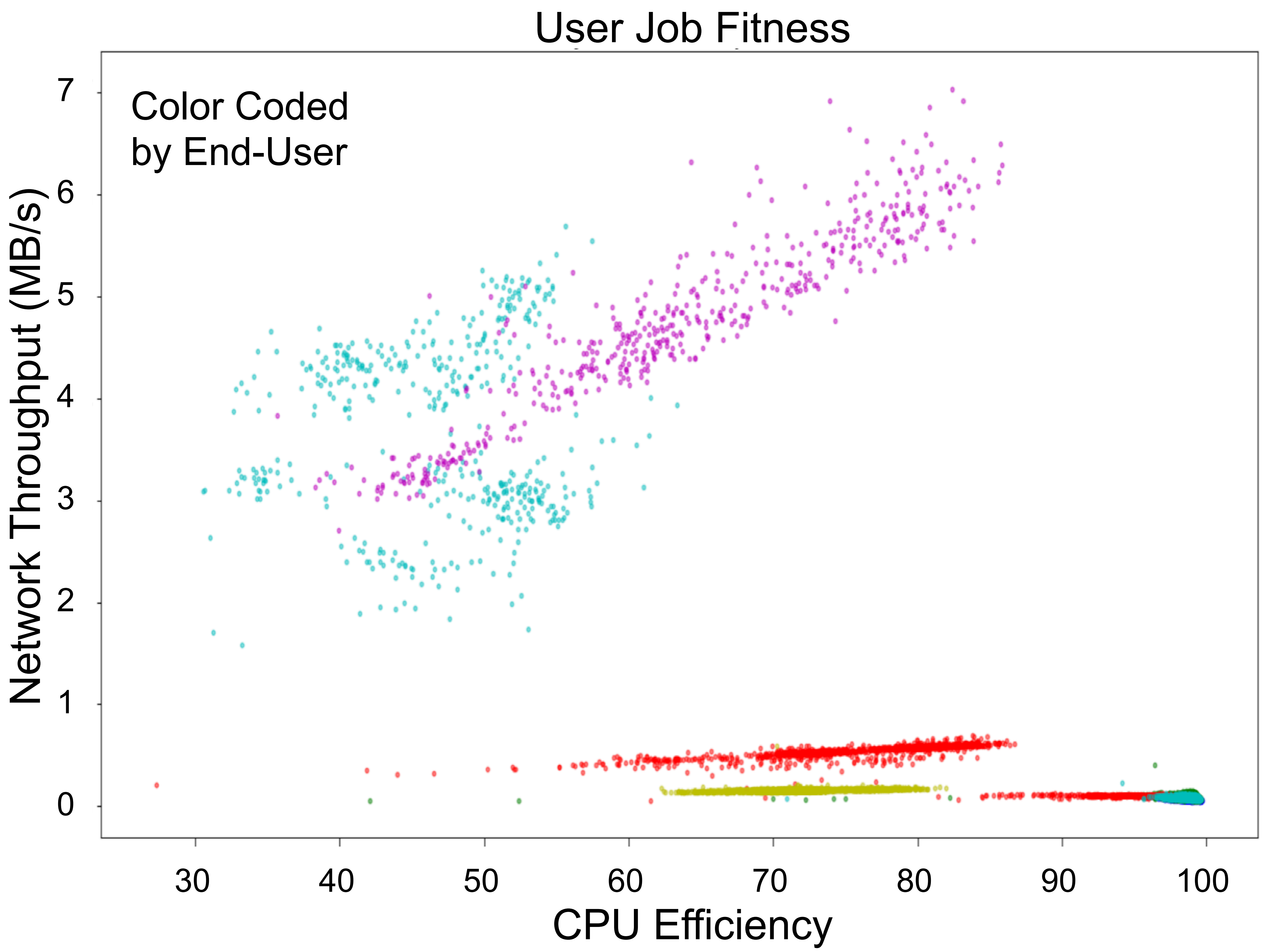
Classical Job to Resource to Job meta-scheduler:

- Dynamic resource acquisition matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users
- Job scheduling in overlay batch system
  - Unreliable to predict resources used by jobs
  - Efficient to integrate resources for all jobs



Implicit Resource Scheduling via Feedback

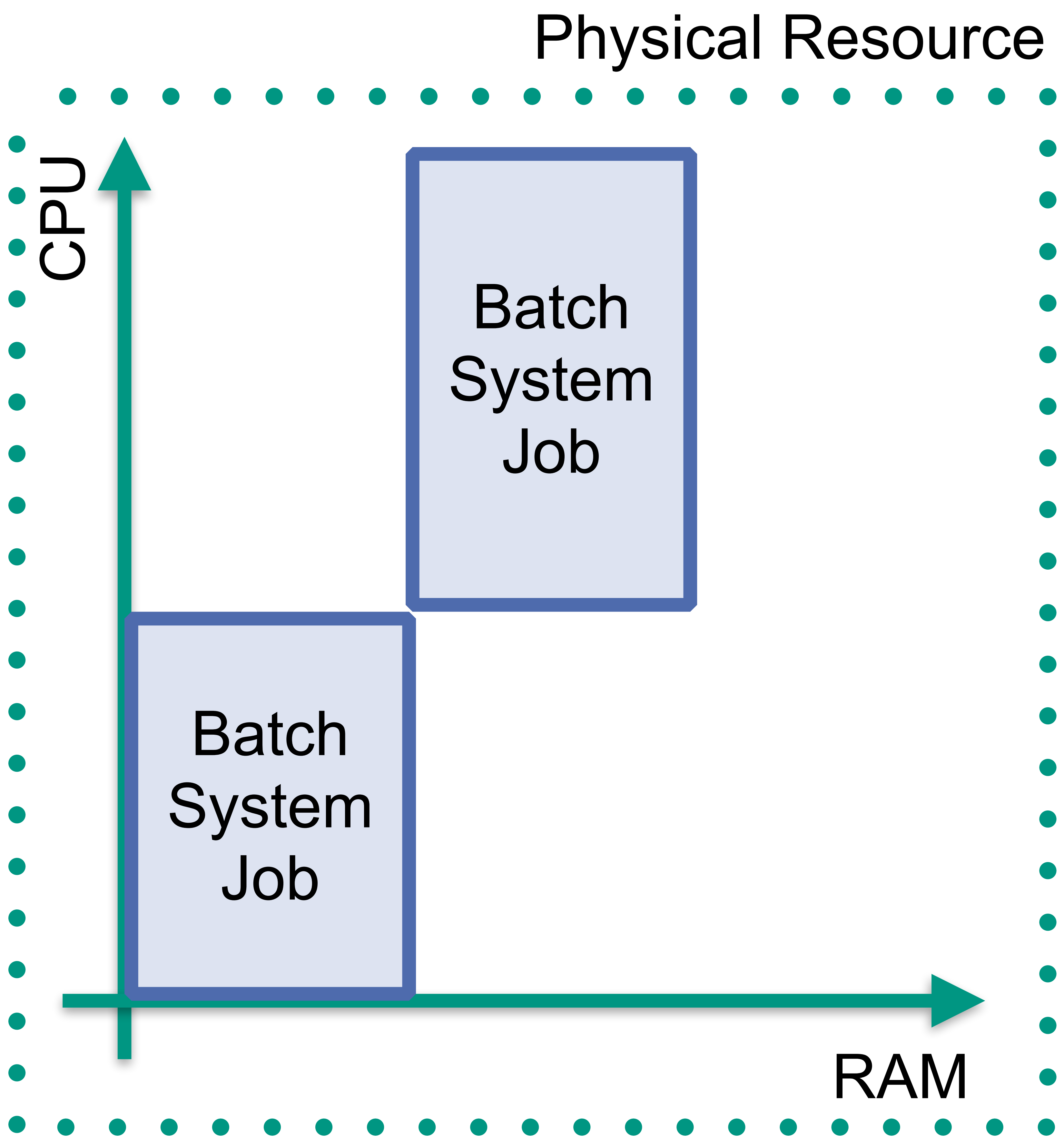
- Respect network availability and congestion for provisioning
  - Congested network is the bottleneck for opportunistic resources
  - Non-linear interference and noticeable measurement overhead



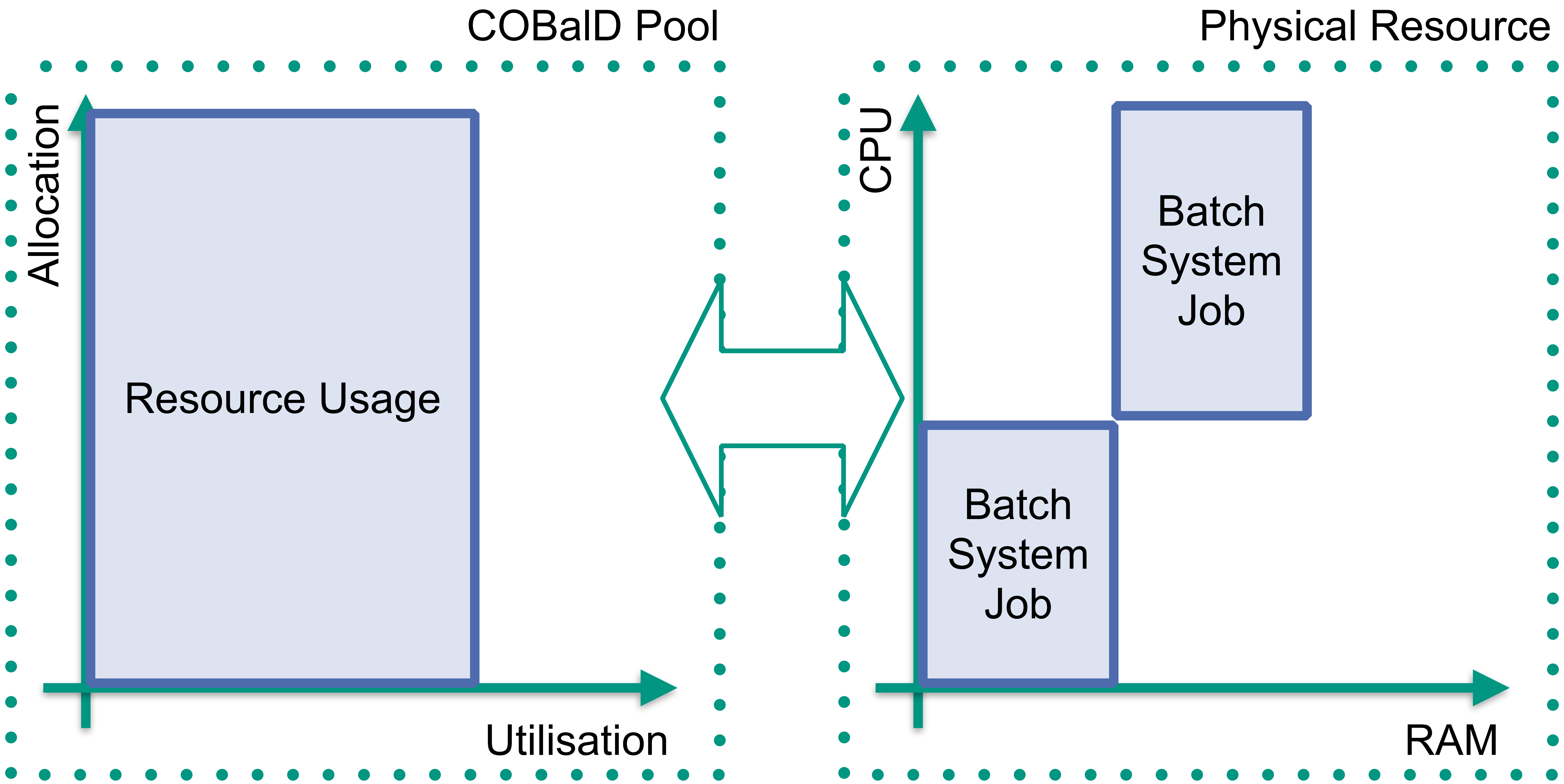
Implicit Resource Scheduling via Feedback

- Respect network availability and congestion for provisioning
  - Congested network is the bottleneck for opportunistic resources
  - Non-linear interference and noticeable measurement overhead
- Research: Implicitly schedule network capacity via side-effects
  - Cheap CPU efficiency query as boundary for network efficiency (and other resources)
    - CPU efficiency implies general fitness
  - Safeguard to push the maximum possible data analysis jobs to opportunistic resources

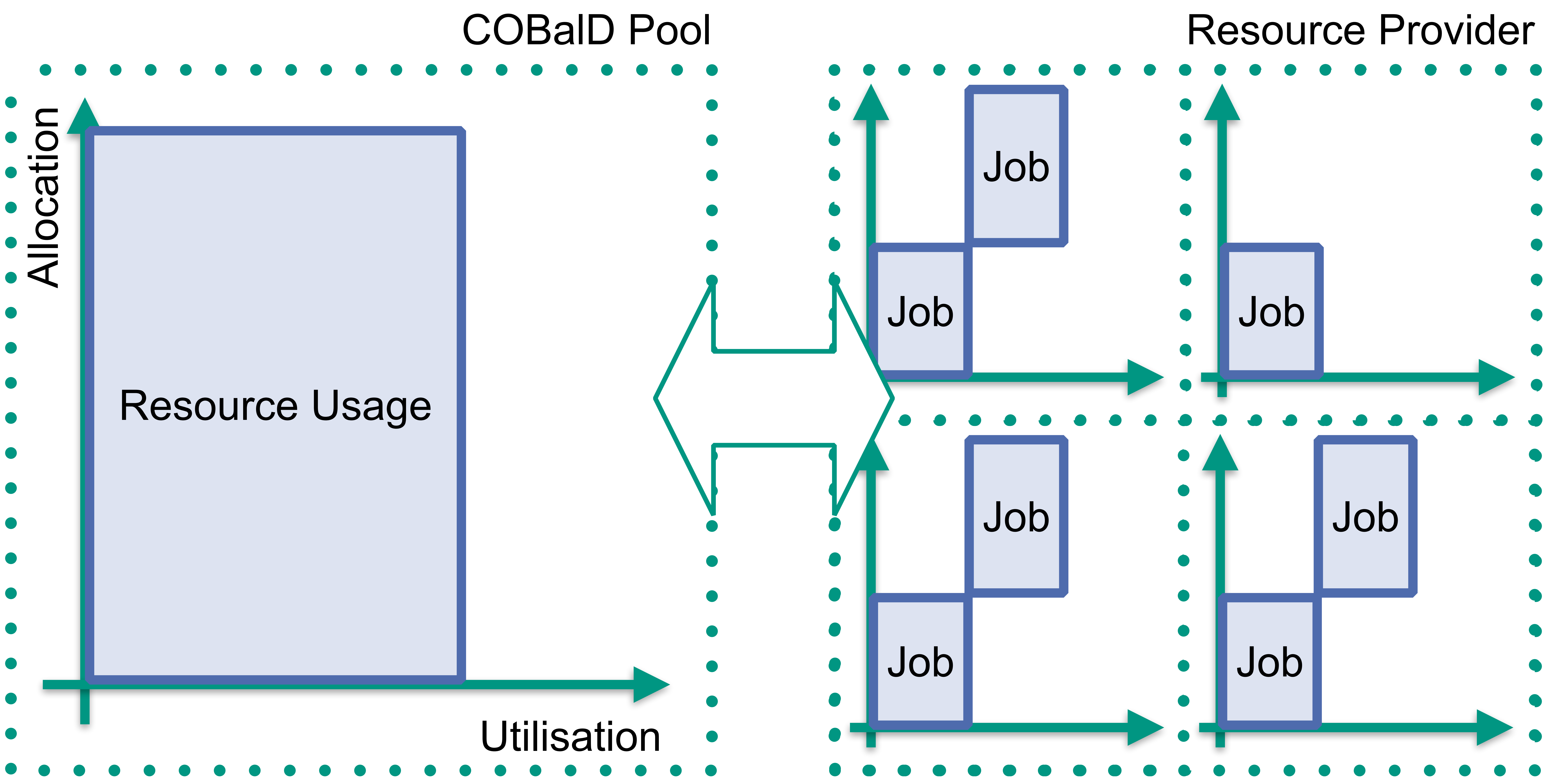
COBalD Resource Pool Model



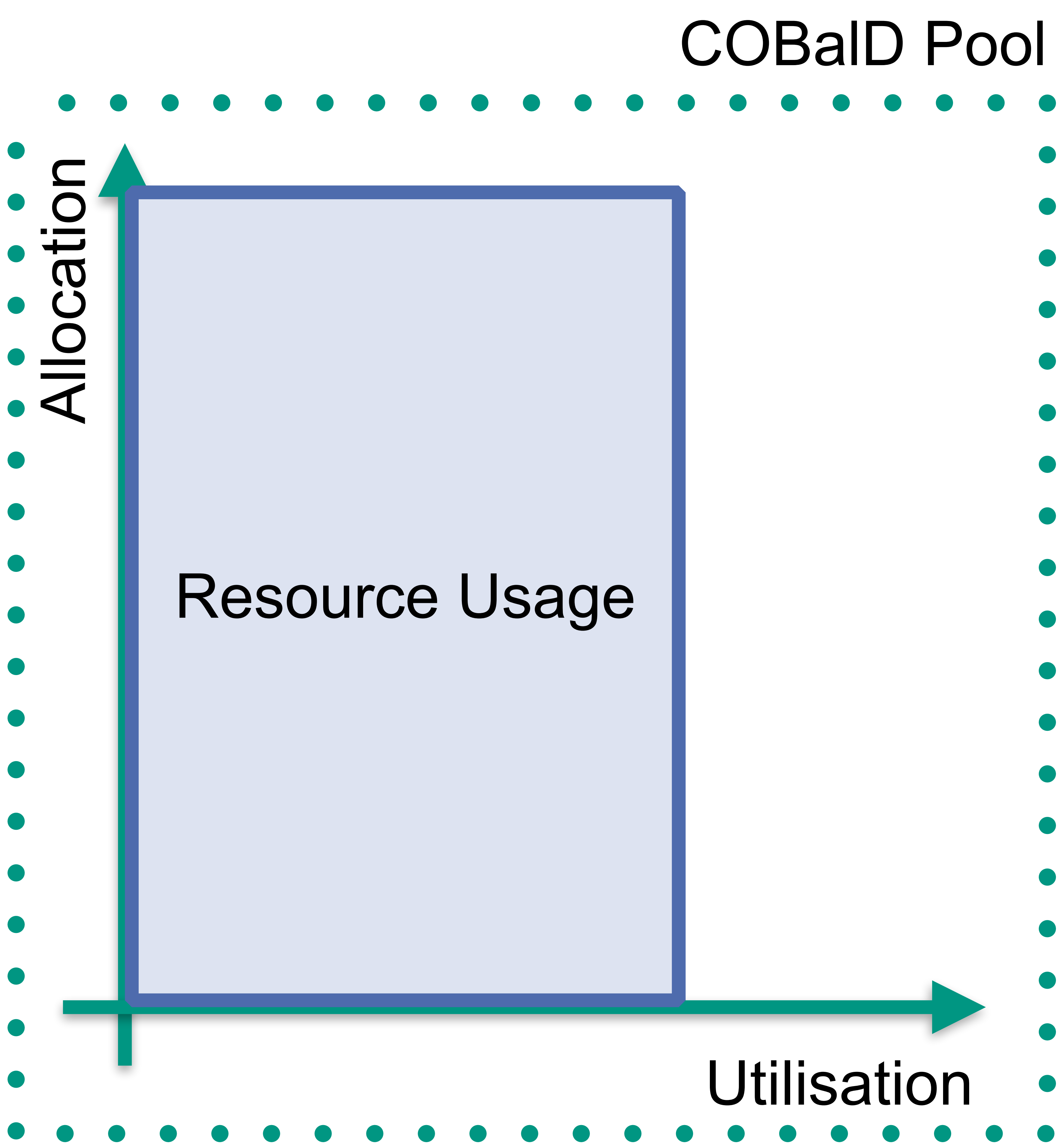
COBalD Resource Pool Model



COBalD Resource Pool Model



COBalD Resource Pool Model



COBalD Resource Pool Model

```
if utilisation < self.low_utilisation:
    return supply * self.high_scale
```