Karlsruhe Institute of Technology

R&D for Future High Throughput Computing @ GridKa

TRIUMF-Helmholtz Workshop on Scientific Computing, DESY Hamburg - 16.09.2019

Manuel Giffels, René Caspart, Florian v. Cube, Tabea Fesenbecker, Max Fischer, Christoph Heidecker, Eileen Kühn, Matthias Schnepf

Institute for Particle Physics (ETP) & Steinbuch Centre for Computing (SCC)

- **German Tier-1 High Energy Physics and Astroparticle Data & Analysis Centre**
  - Support all four LHC experiments
  - Belle II, Pierre-Auger, several small communities
- **Joint R&D** with computer science towards HL-LHC
- **Resources**
  - Compute: ~29k cores
  - Disk: 37PB (used), Tape: 54 PB (used)
  - 100 Gb/s connection to LHCONE/ LHCOPN
- Among the largest and best performing T1s
- Annual international **GridKa School**
  - > 1800 participants since 2003
    https://gridka.school

R&D Environment and Backgrounds

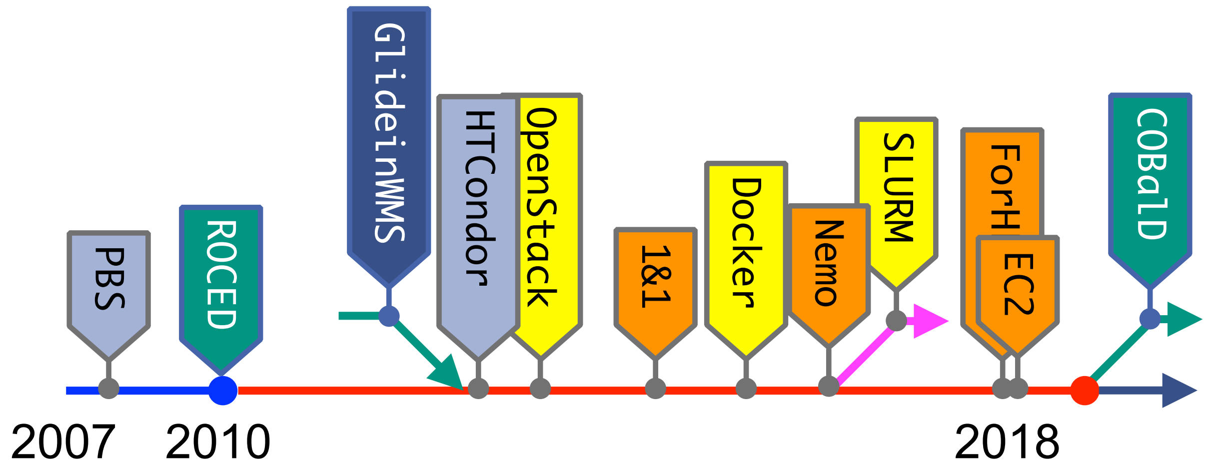
- Two collaborating HEP Computing Groups at KIT
  - SCC: GridKa Tier 1, focus on throughput and production systems
  - ETP: Institute Tier 3, focus on responsiveness and prototypes

R&D Environment and Backgrounds

- Two collaborating HEP Computing Groups at KIT
  - SCC: GridKa Tier 1, focus on throughput and production systems
  - ETP: Institute Tier 3, focus on responsiveness and prototypes
- The major topics of Research and Development
  - Dynamic on-demand processing resources via VMs/containers
    T. Hauth et al., On-demand provisioning of HEP Compute Resources on Clouds Sites and Shared HPC Centers, Journal of Physics **898**, 5 (2017)
  - Adaptive placement of input data via distributed coordinated caches
    M. Fischer et al., Opportunistic Data Locality for End User Data Analysis, Journal of Physics **898**, 5 (2017)

R&D Environment and Backgrounds

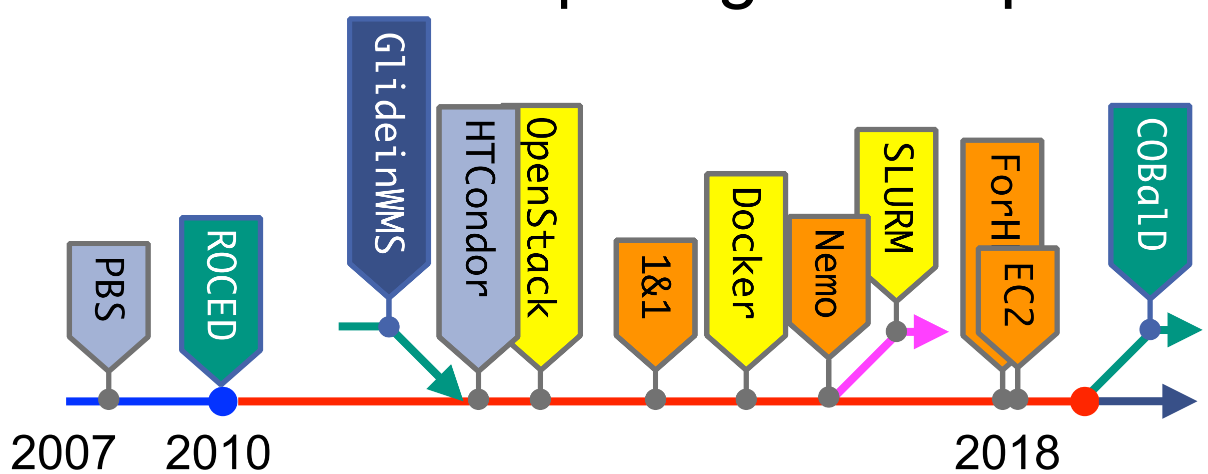
- Two collaborating HEP Computing Groups at KIT
  - SCC: GridKa Tier 1, focus on throughput and production systems
  - ETP: Institute Tier 3, focus on responsiveness and prototypes
- The major topics of Research and Development
  - Dynamic on-demand processing resources via VMs/containers
    T. Hauth et al., On-demand provisioning of HEP Compute Resources on Clouds Sites and Shared HPC Centers, Journal of Physics **898**, 5 (2017)
  - Adaptive placement of input data via distributed coordinated caches
    M. Fischer et al., Opportunistic Data Locality for End User Data Analysis, Journal of Physics **898**, 5 (2017)
- Longstanding involvement in computing development projects



Timeline of on-demand processing

R&D Environment and Backgrounds

- Two collaborating HEP Computing Groups at KIT
  - SCC: GridKa Tier 1, focus on throughput and production systems
  - ETP: Institute Tier 3, focus on responsiveness and prototypes
- The major topics of Research and Development
  - Dynamic on-demand processing resources via VMs/containers
    T. Hauth et al., On-demand provisioning of HEP Compute Resources on Clouds Sites and Shared HPC Centers, Journal of Physics **898**, 5 (2017)
  - Adaptive placement of input data via distributed coordinated caches
    M. Fischer et al., Opportunistic Data Locality for End User Data Analysis, Journal of Physics **898**, 5 (2017)
- Longstanding involvement in computing development projects



Timeline of on-demand processing

Opportunistic Resource

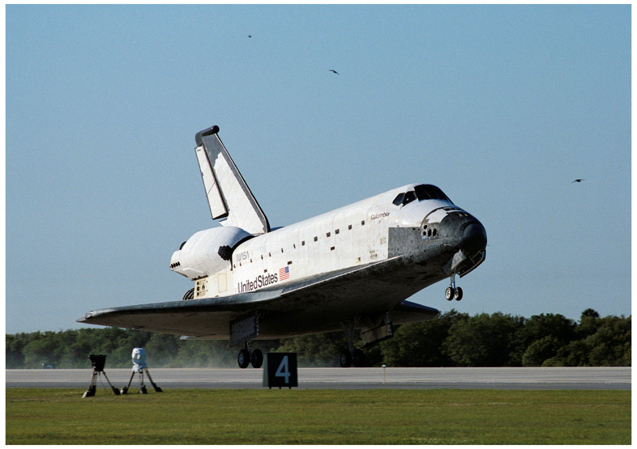
Any resources *not permanently dedicated to* but *temporarily available for* a specific task, user or group.

Opportunistic Resources and their Challenges

Very different to the traditional HEP environment:

Opportunistic Resources and their Challenges

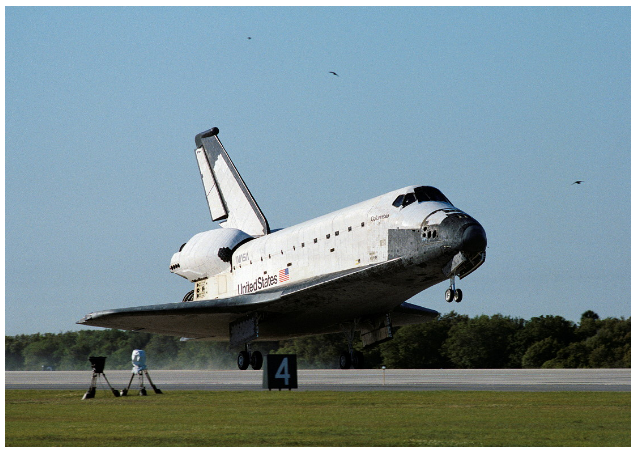
Very different to the traditional HEP environment:



HEP-Job

Opportunistic Resources and their Challenges

Very different to the traditional HEP environment:

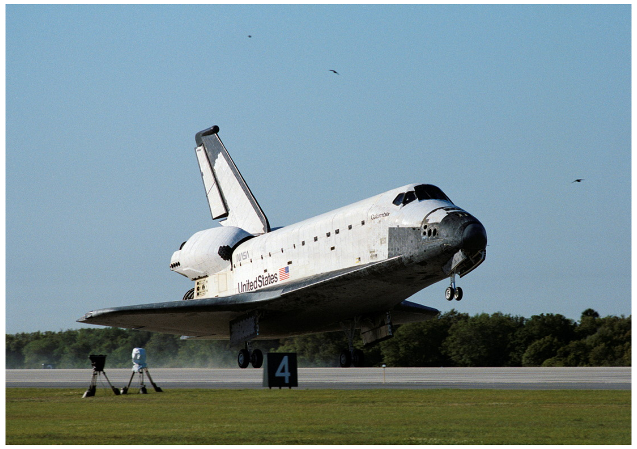


HEP-Job

HPC-Job

Opportunistic Resources and their Challenges

Very different to the traditional HEP environment:



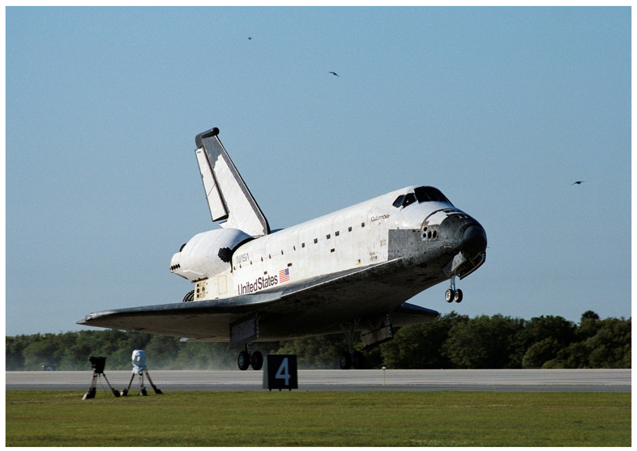
HEP-Job

Virtualisation/Container

HPC-Job

Opportunistic Resources and their Challenges

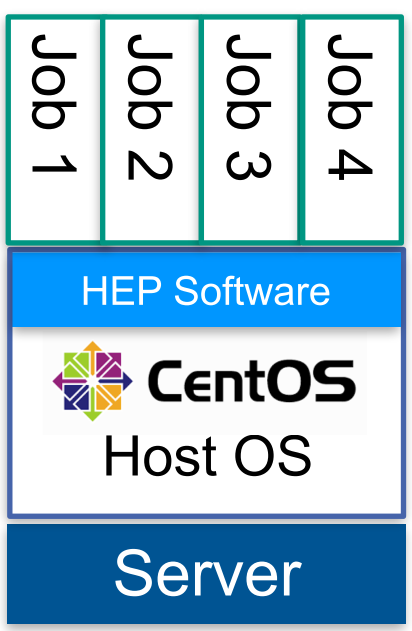
Very different to the traditional HEP environment:



HEP-Job

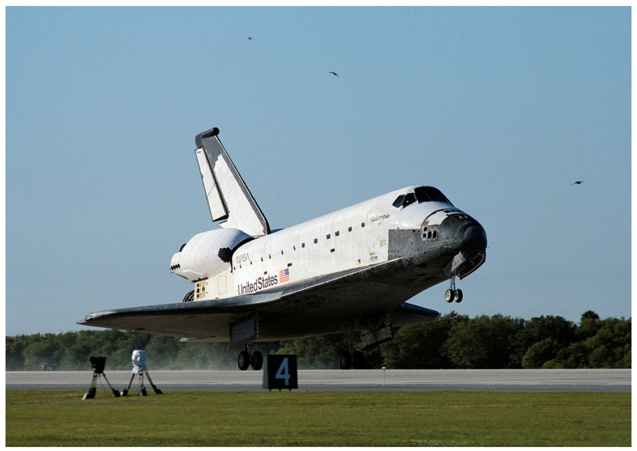
Virtualisation/Container

HPC-Job



Opportunistic Resources and their Challenges

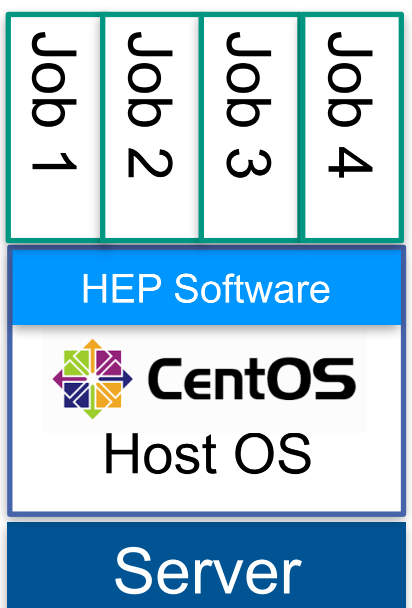
Very different to the traditional HEP environment:

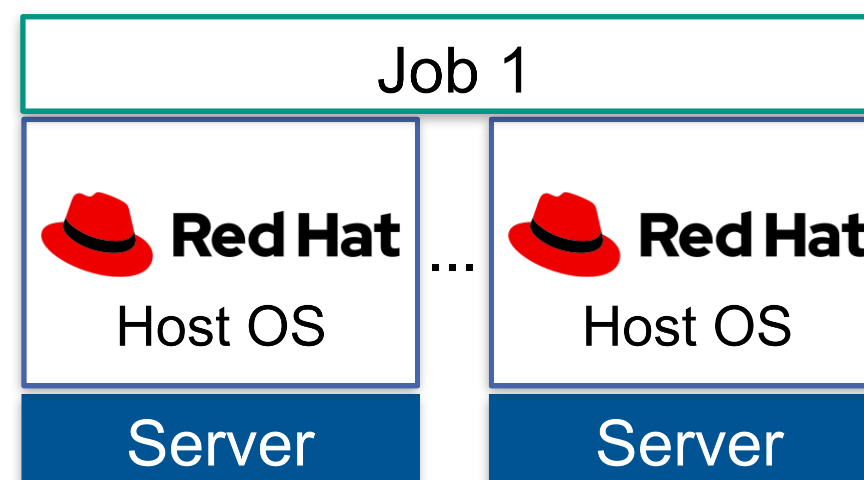


HEP-Job

Virtualisation/Container

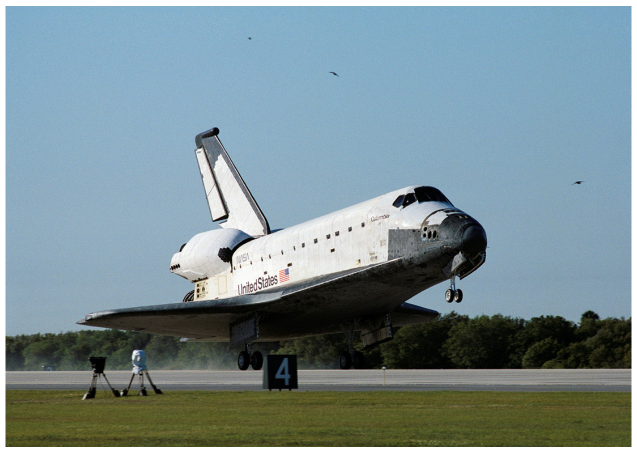
HPC-Job





Opportunistic Resources and their Challenges

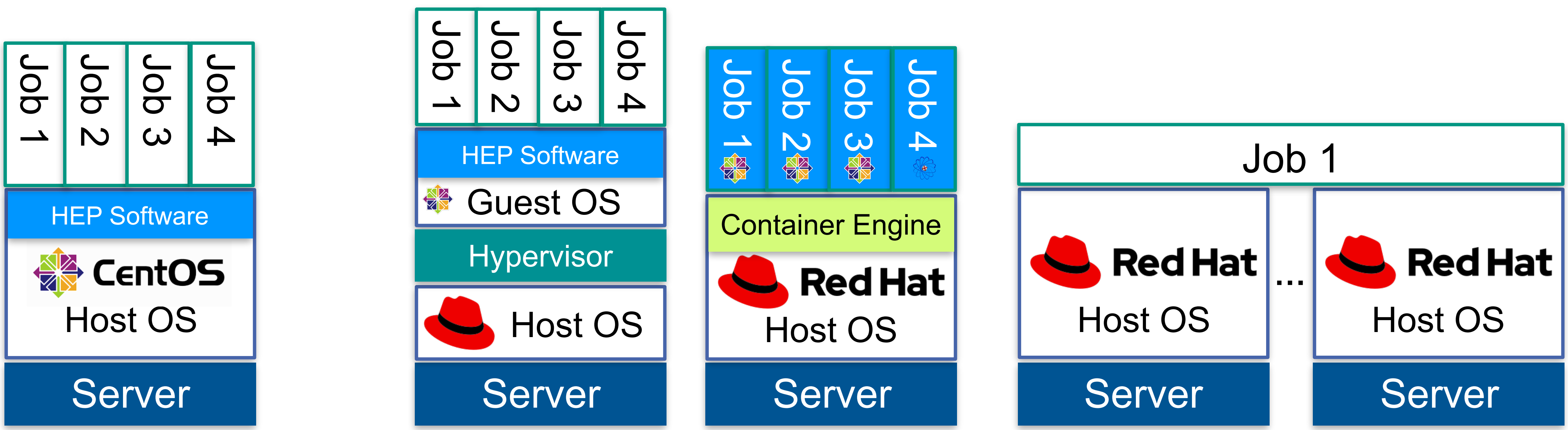
Very different to the traditional HEP environment:



HEP-Job

Virtualisation/Container

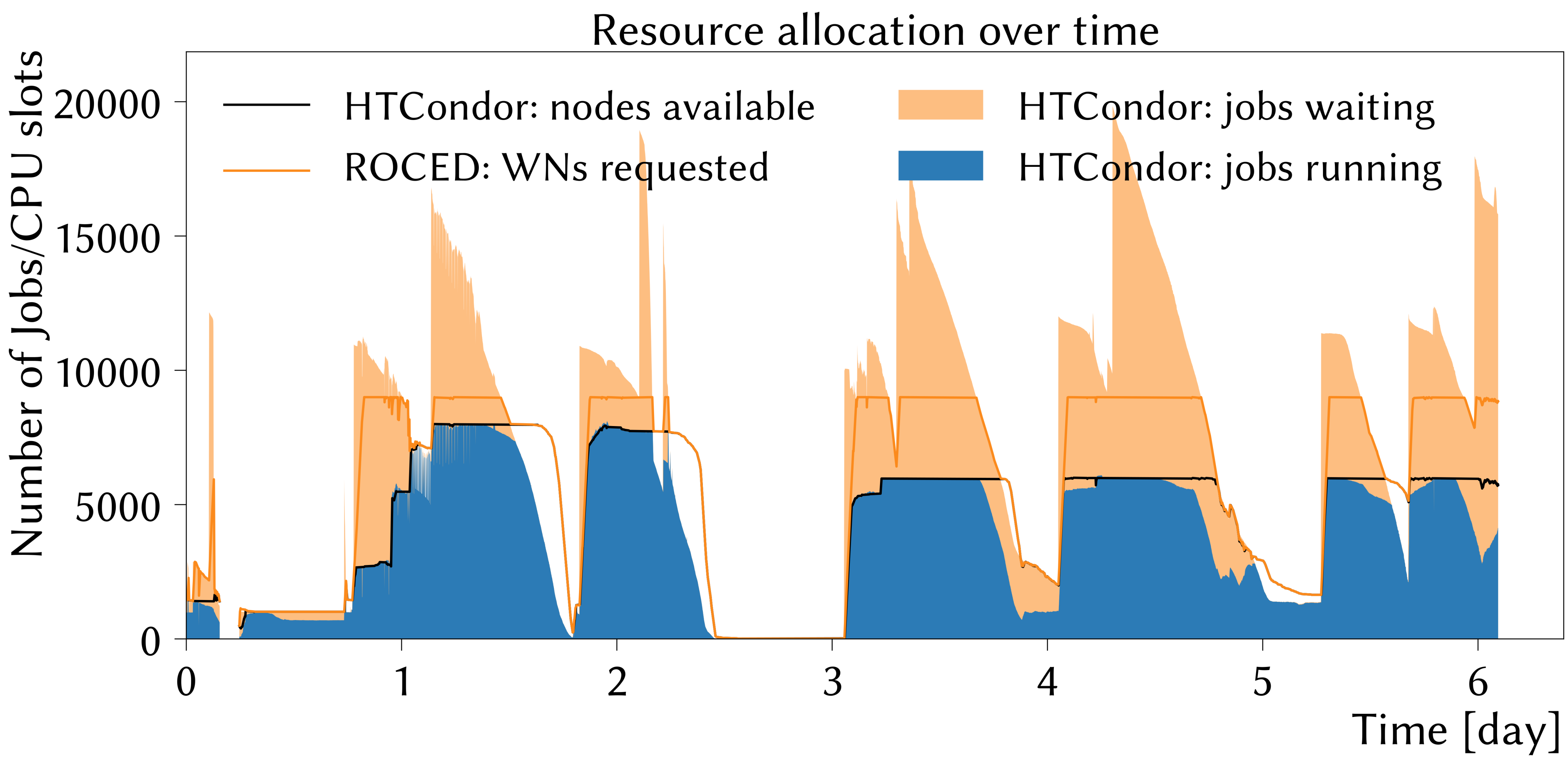
HPC-Job



➜ Virtualisation and containerisation techniques

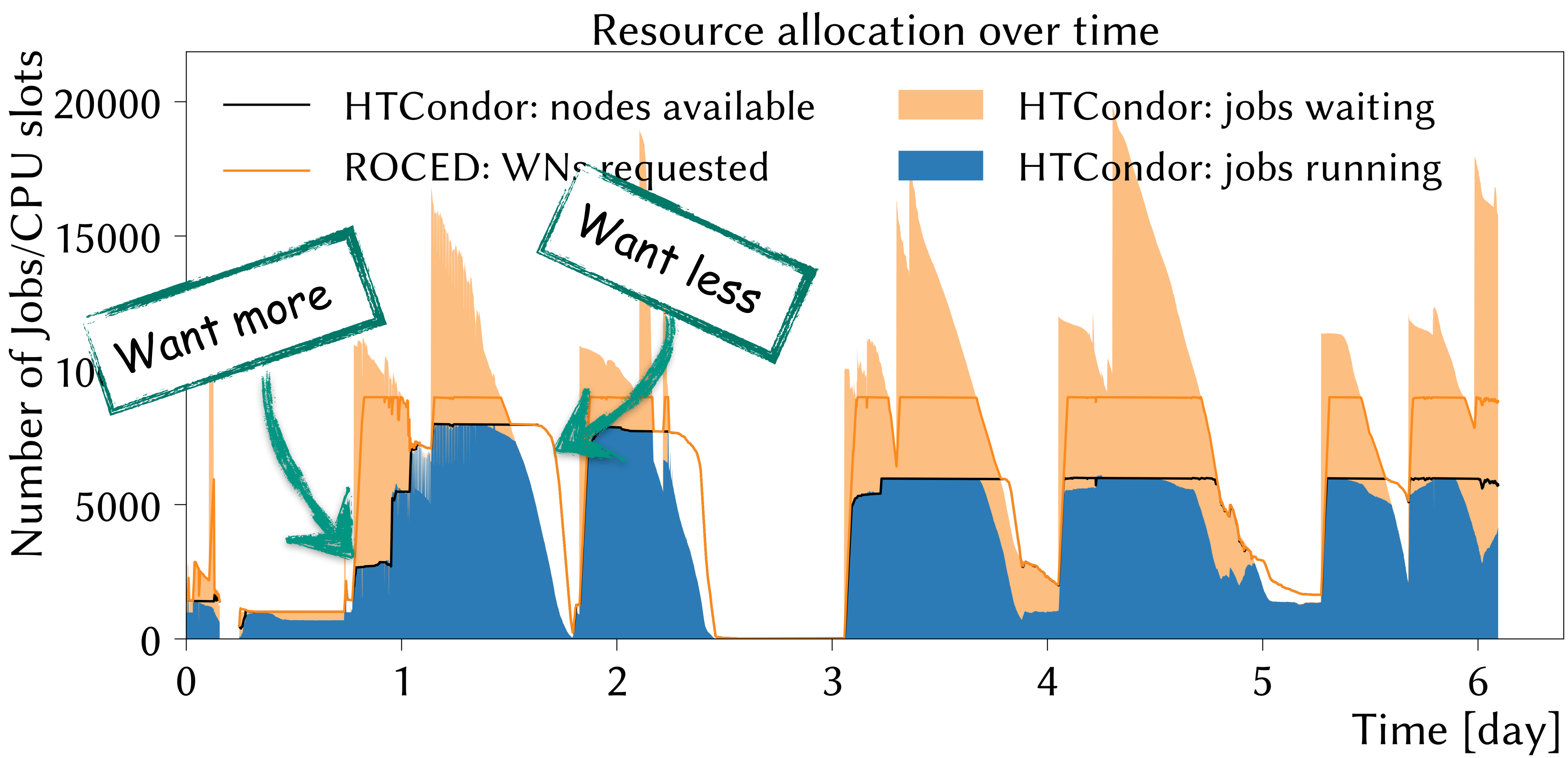
Opportunistic Resources and their Challenges

- Temporary availability of opportunistic resources
- Varying demand for opportunistic resource



Opportunistic Resources and their Challenges

- Temporary availability of opportunistic resources
- Varying demand for opportunistic resource

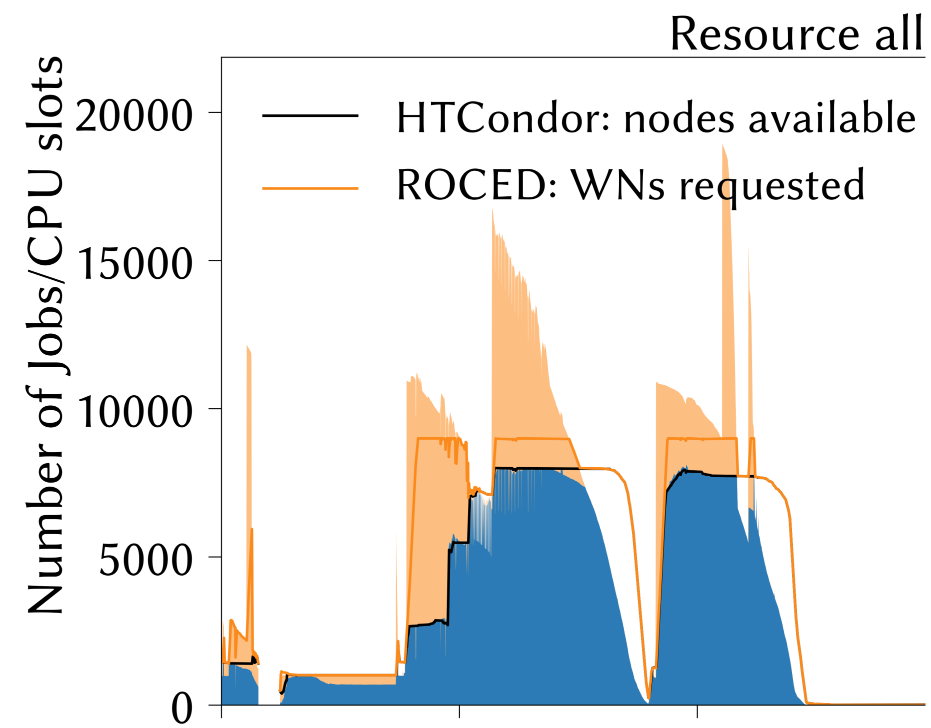


➜ Dynamic integration and workflow management

COBalD/TARDIS Resource Manager

Development at KIT:

- Following simplistic approach
  - Monitor resource utilisation
  - Increase well utilised resources
  - Reduce not used resources



COBalD

COBalD – the Opportunistic Balancing Daemon

COBalD is a lightweight framework to balance opportunistic resources: cloud bursting, container orchestration, allocation scaling and more. Its lightweight `model` for resources and their composition makes it easy to integrate custom resources and manage them at a large scale.

DOCUMENTATION | VERSION CONTROL | DOI

TARDIS – Resourcemanager

Transparent Adaptive Resource Dynamic Integration System

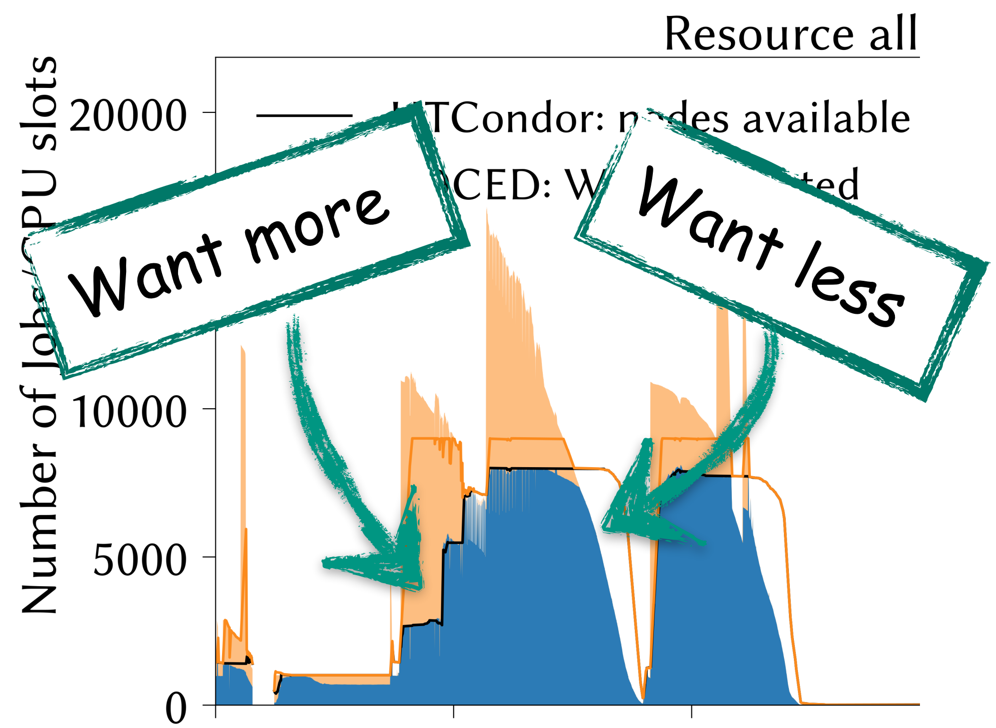
Transparent Adaptive Resource Dynamic Integration System enables the dynamic integration of resources provided by different resource providers into one overlay batch system.

DOCUMENTATION | VERSION CONTROL | DOI

COBalD/TARDIS Resource Manager

<u>Development at KIT:</u>

- Following simplistic approach
  - Monitor resource utilisation
  - Increase well utilised resources
  - Reduce not used resources



COBalD

COBalD – the Opportunistic Balancing Daemon

COBalD is a lightweight framework to balance opportunistic resources: cloud bursting, container orchestration, allocation scaling and more. Its lightweight `model` for resources and their composition makes it easy to integrate custom resources and manage them at a large scale.

DOCUMENTATION | VERSION CONTROL | DOI

TARDIS – Resourcemanager

Transparent Adaptive Resource Dynamic Integration System

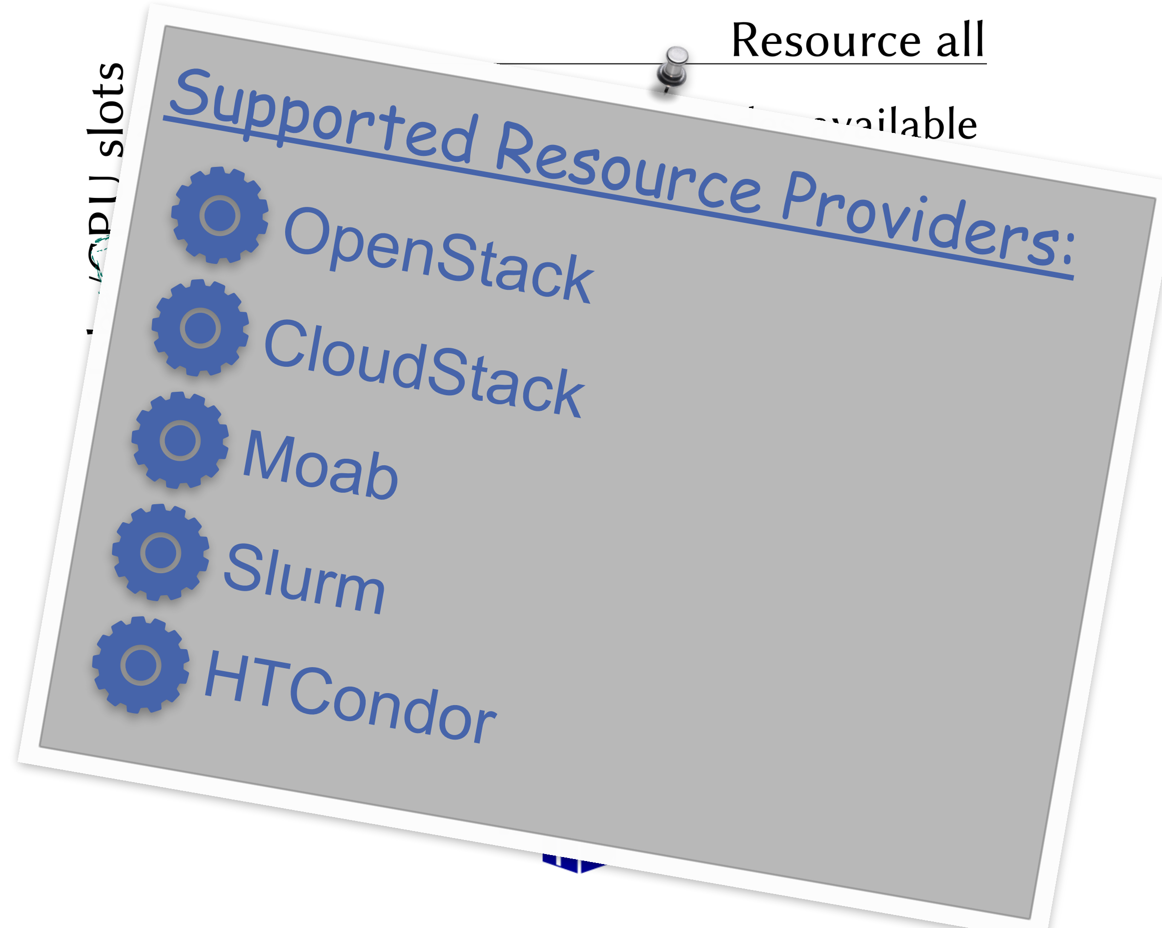
Transparent Adaptive Resource Dynamic Integration System enables the dynamic integration of resources provided by different resource providers into one overlay batch system.

DOCUMENTATION | VERSION CONTROL | DOI

COBalD/TARDIS Resource Manager

<u>Development at KIT:</u>

- Following simplistic approach
  - Monitor resource utilisation
  - Increase well utilised resources
  - Reduce not used resources



COBalD

COBalD – the Opportunistic Balancing Daemon

COBalD is a lightweight framework to balance opportunistic resources: cloud bursting, container orchestration, allocation scaling and more. Its lightweight `model` for resources and their composition makes it easy to integrate custom resources and manage them at a large scale.

DOCUMENTATION | VERSION CONTROL | DOI

TARDIS – Resourcemanager

Transparent Adaptive Resource Dynamic Integration System

Transparent Adaptive Resource Dynamic Integration System enables the dynamic integration of resources provided by different resource providers into one overlay batch system.

DOCUMENTATION | VERSION CONTROL | DOI

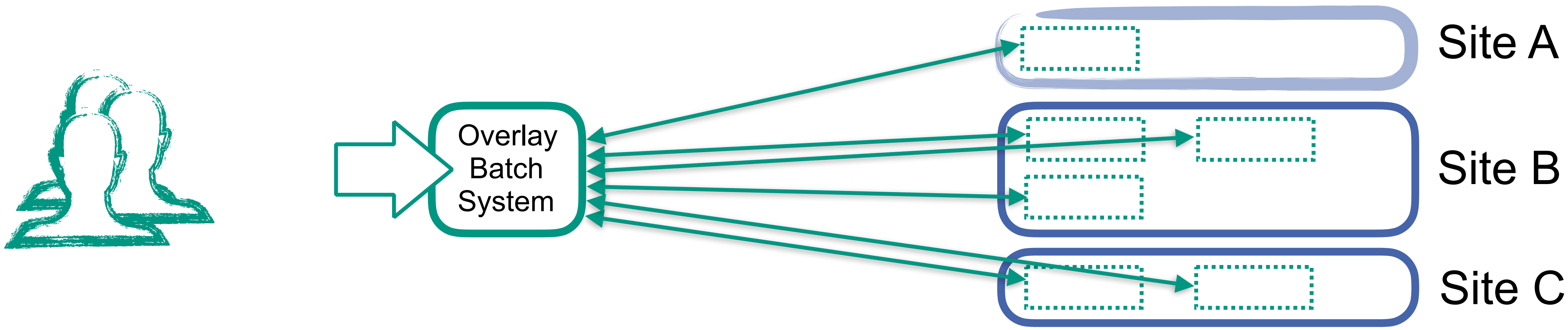
Opportunistic Resources and their Challenges

Opportunistic Resources and their Challenges

- Each opportunistic resource is different (very heterogenous system)

Opportunistic Resources and their Challenges

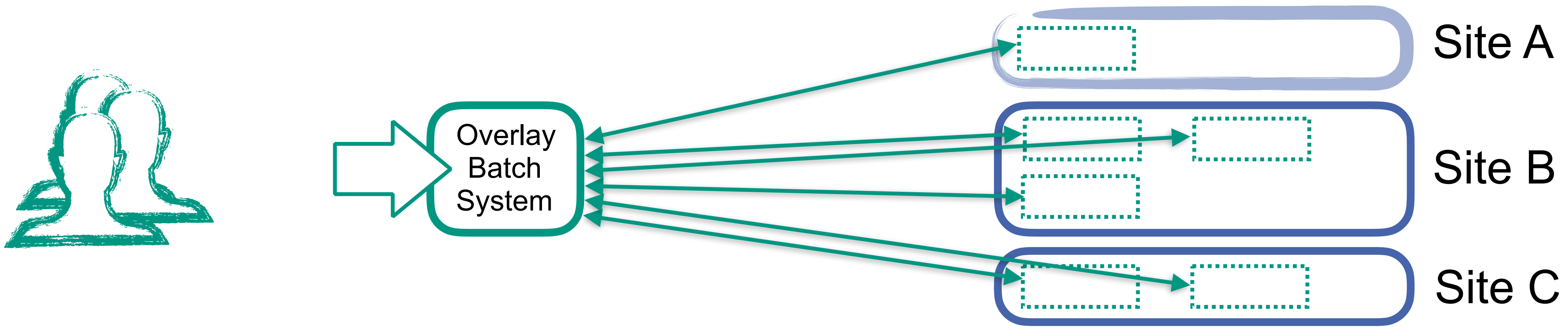
- Each opportunistic resource is different (very heterogenous system)
  - ➡ Hide complexity from users and computing operations of experiments



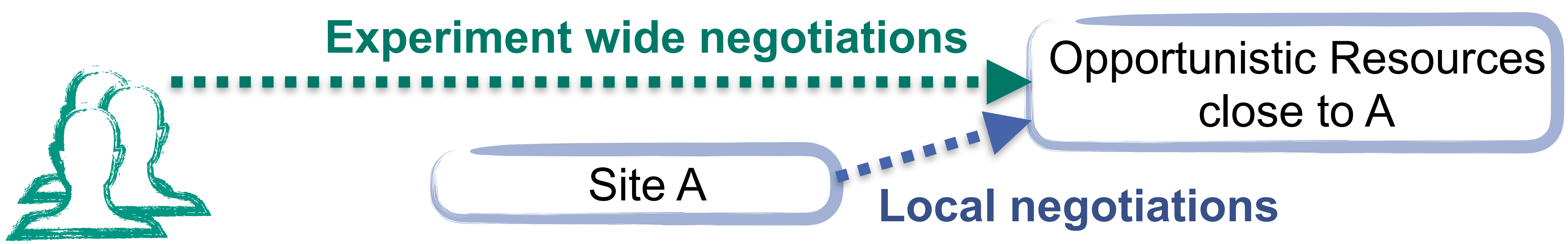
➜Transparent integration of resources needed

Opportunistic Resources and their Challenges

- Each opportunistic resource is different (very heterogenous system)
  - ➡ Hide complexity from users and computing operations of experiments

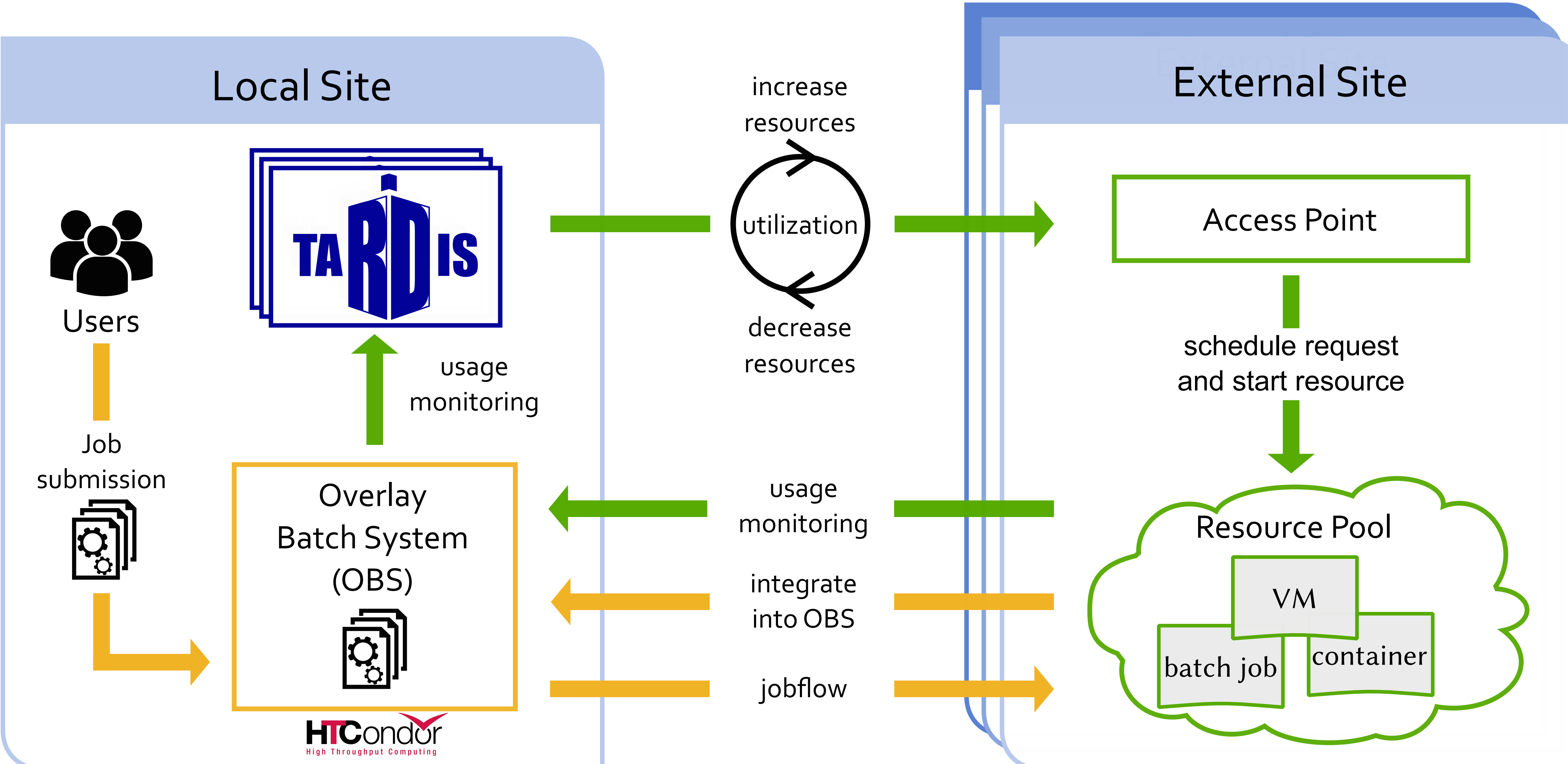


  - ➡ Local/regional negotiations with resource providers more promising than experiment wide negotiations



➜Transparent integration of resources needed

The Entire Picture

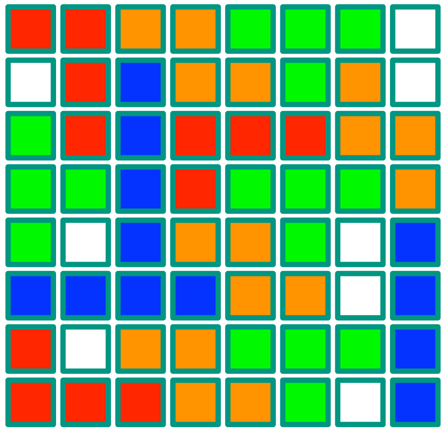


One TARDIS instance per site

Use case: Backfilling of HPC Resources

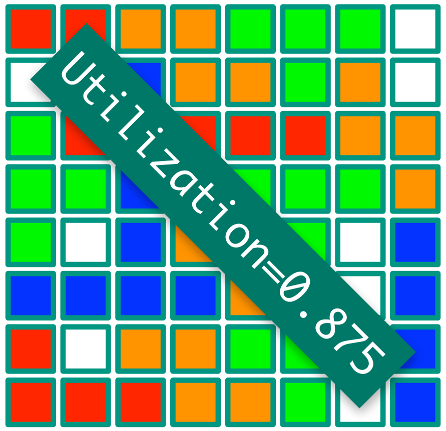
Use case: Backfilling of HPC Resources

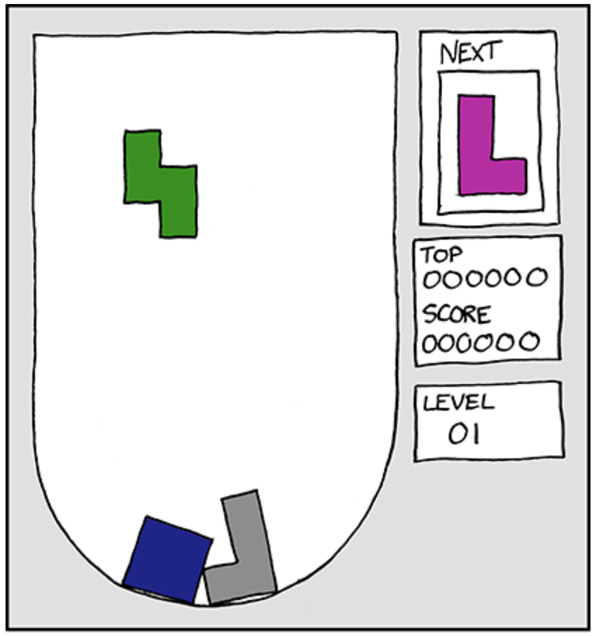
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)



Use case: Backfilling of HPC Resources

- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation

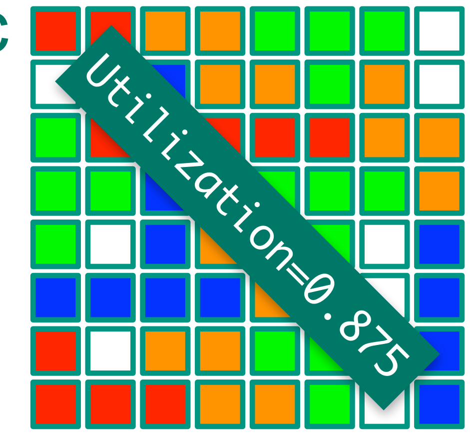


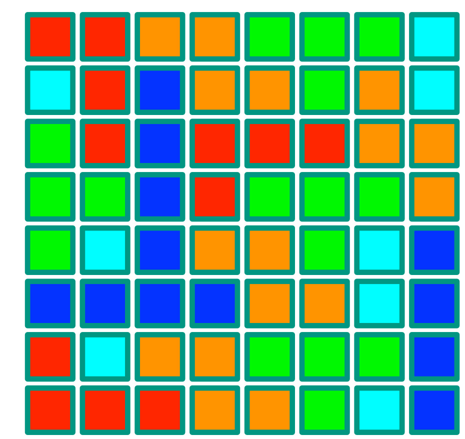


HELL ©XKCD

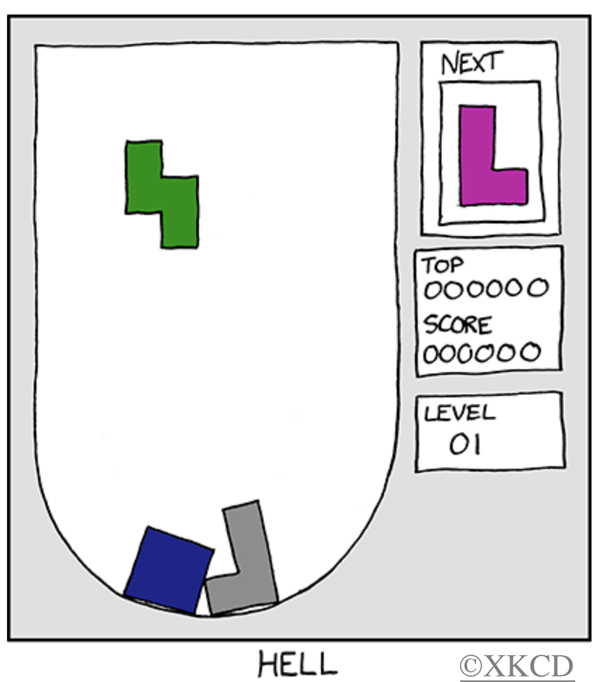
<u>Use case:</u> Backfilling of HPC Resources

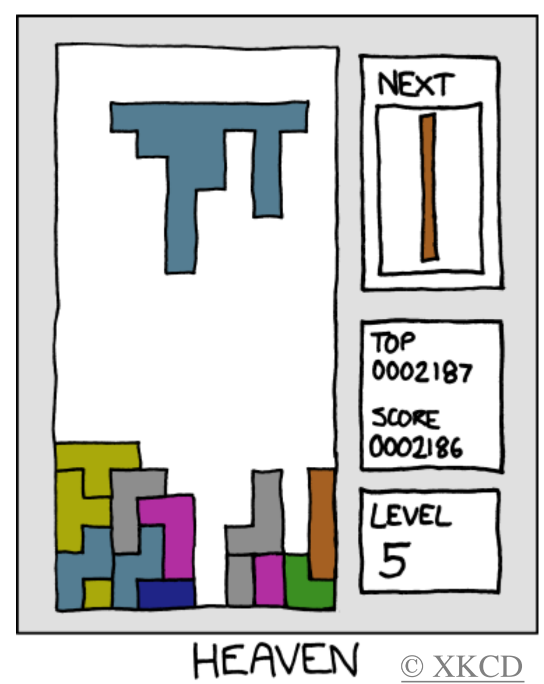
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization





HEP Job

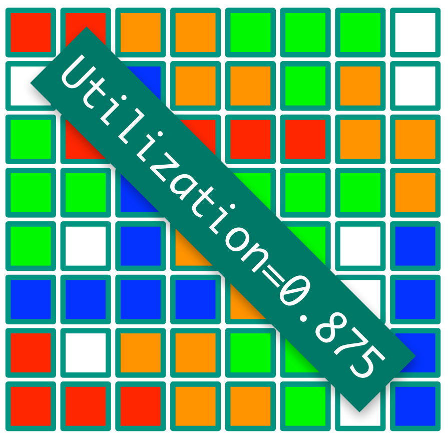


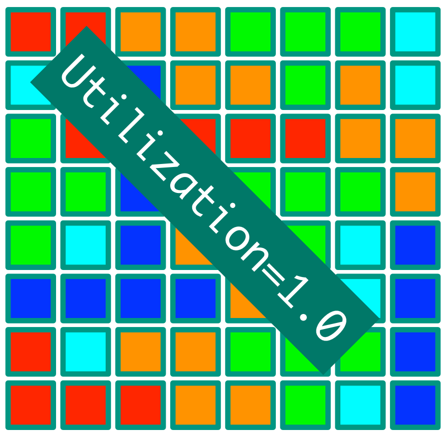


Karlsruhe Institute of Technology

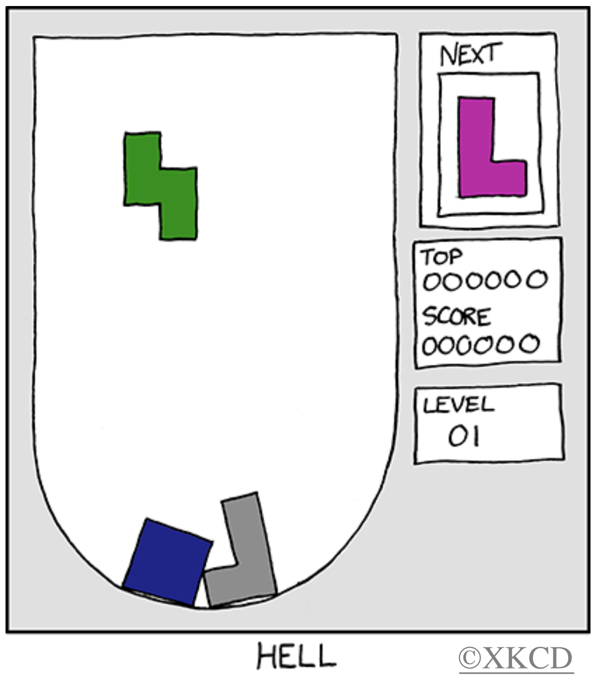
Use case: Backfilling of HPC Resources

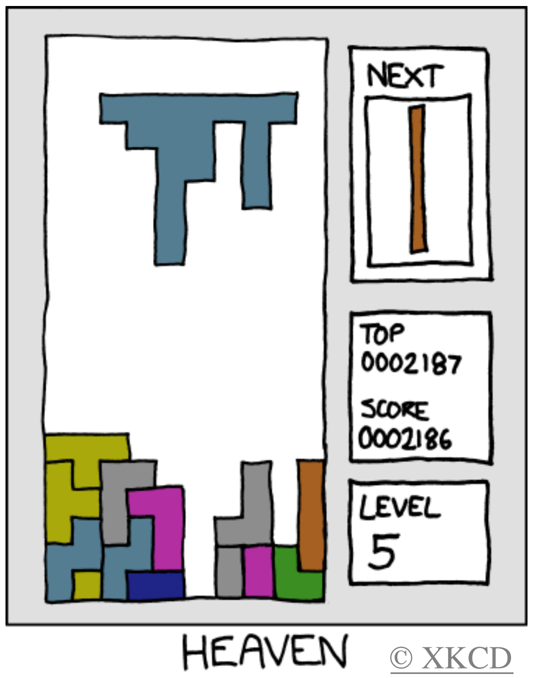
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization





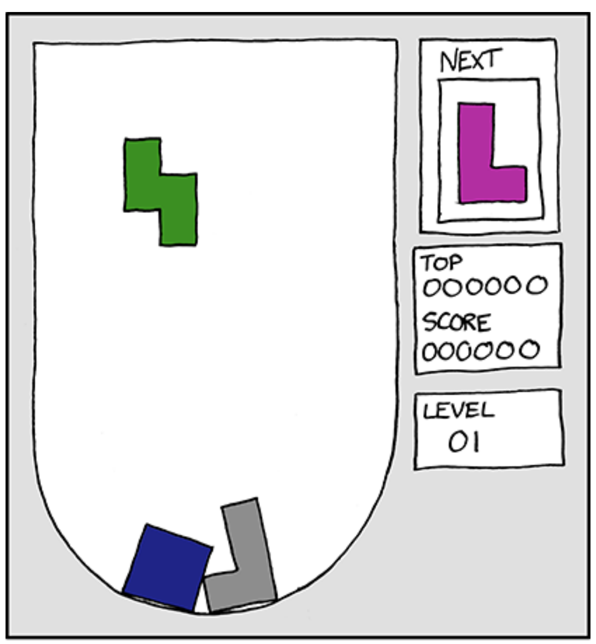
HEP Job



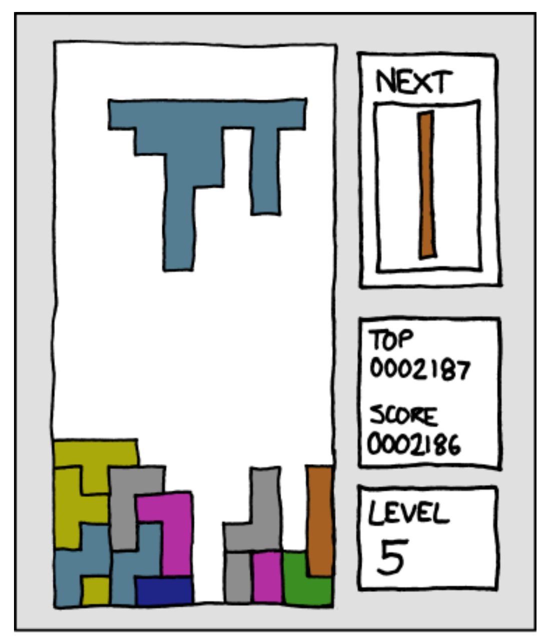


Use case: Backfilling of HPC Resources

- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization



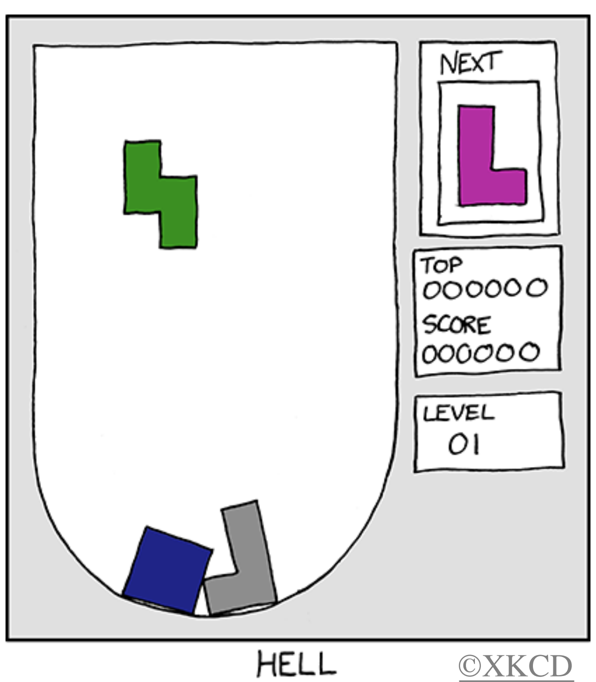
HELL ©XKCD

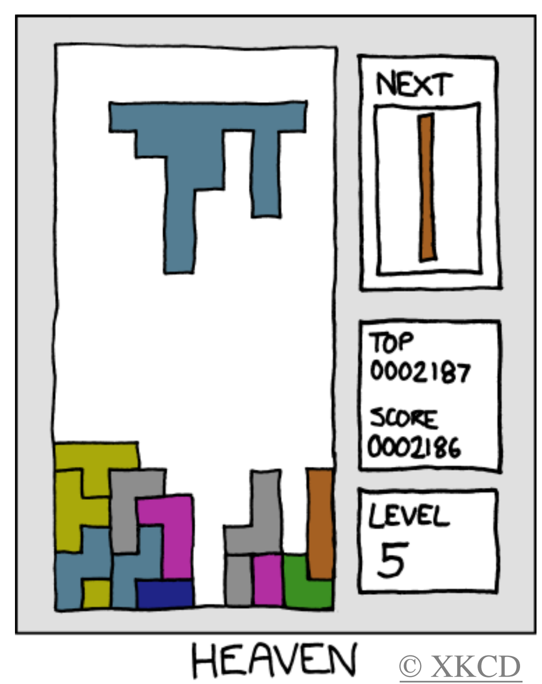


HEAVEN © XKCD

Use case: Backfilling of HPC Resources

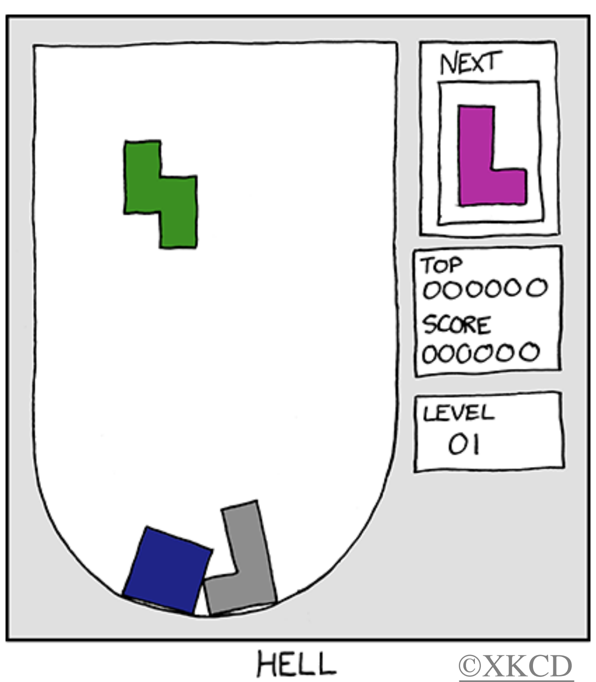
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization

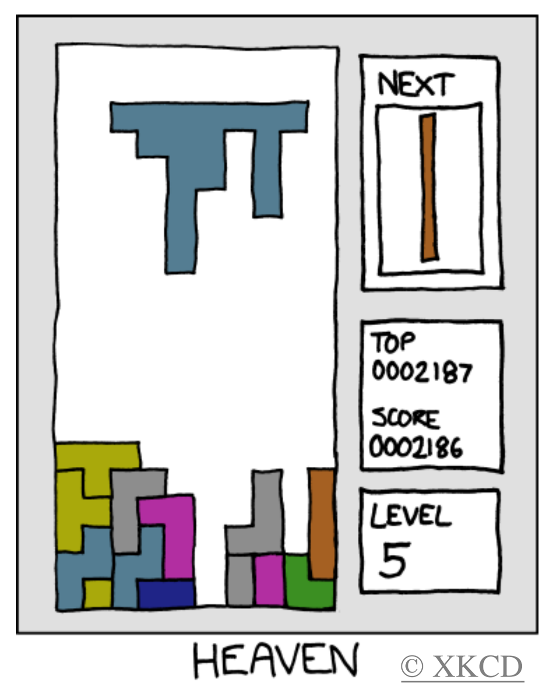




<u>Use case:</u> Backfilling of HPC Resources

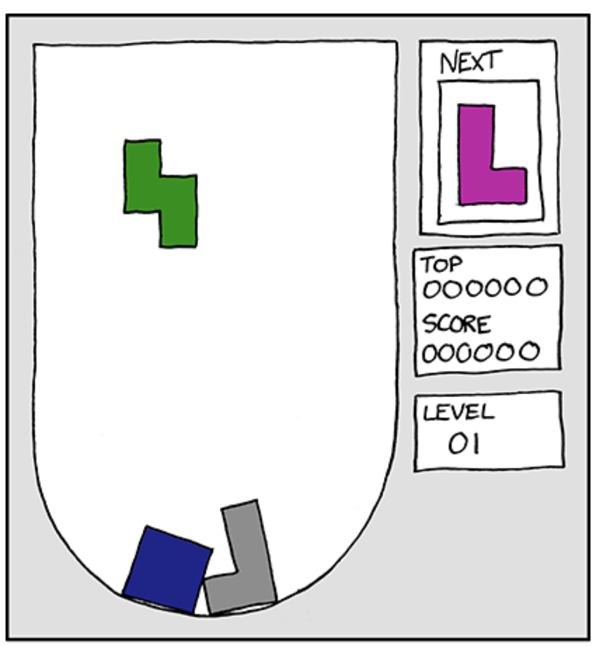
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization



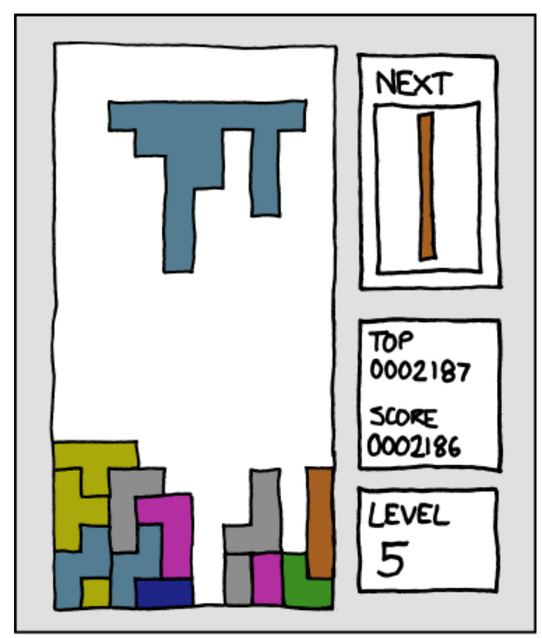


Use case: Backfilling of HPC Resources

- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization
- Enable pre-emption to free resources if needed



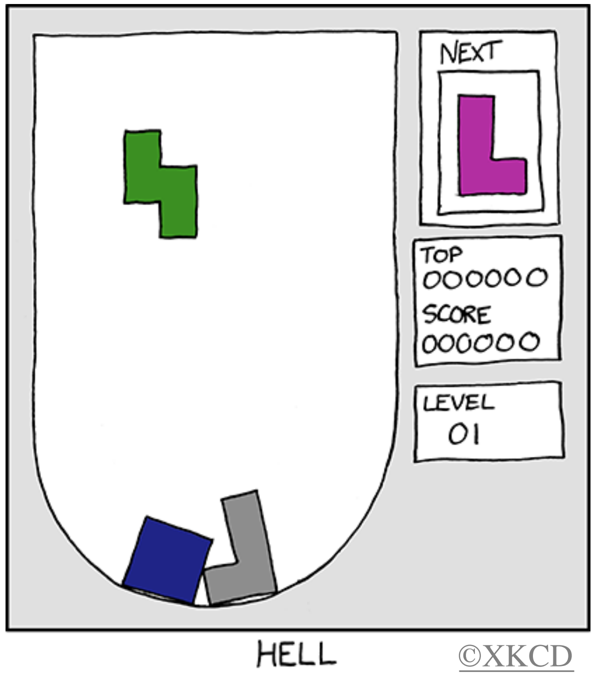
HELL ©XKCD

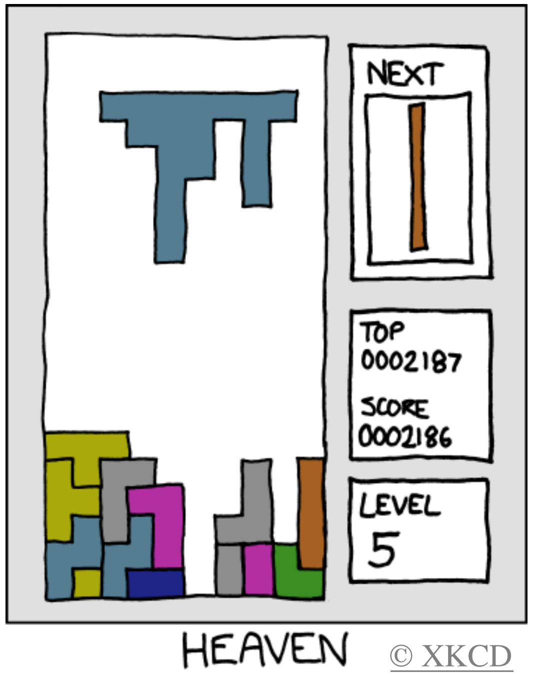


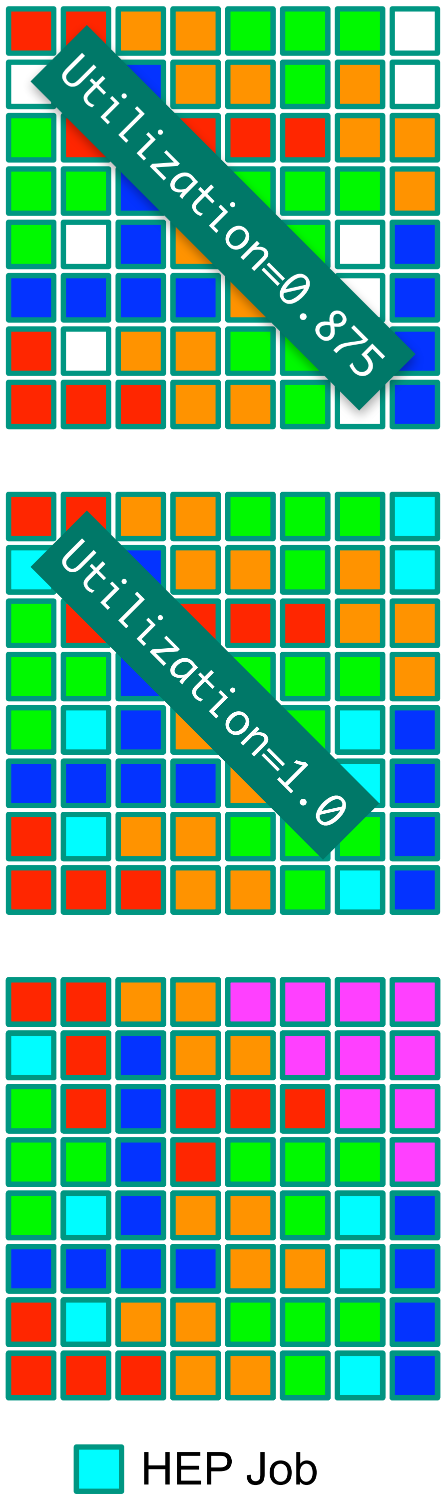
HEAVEN © XKCD

Use case: Backfilling of HPC Resources

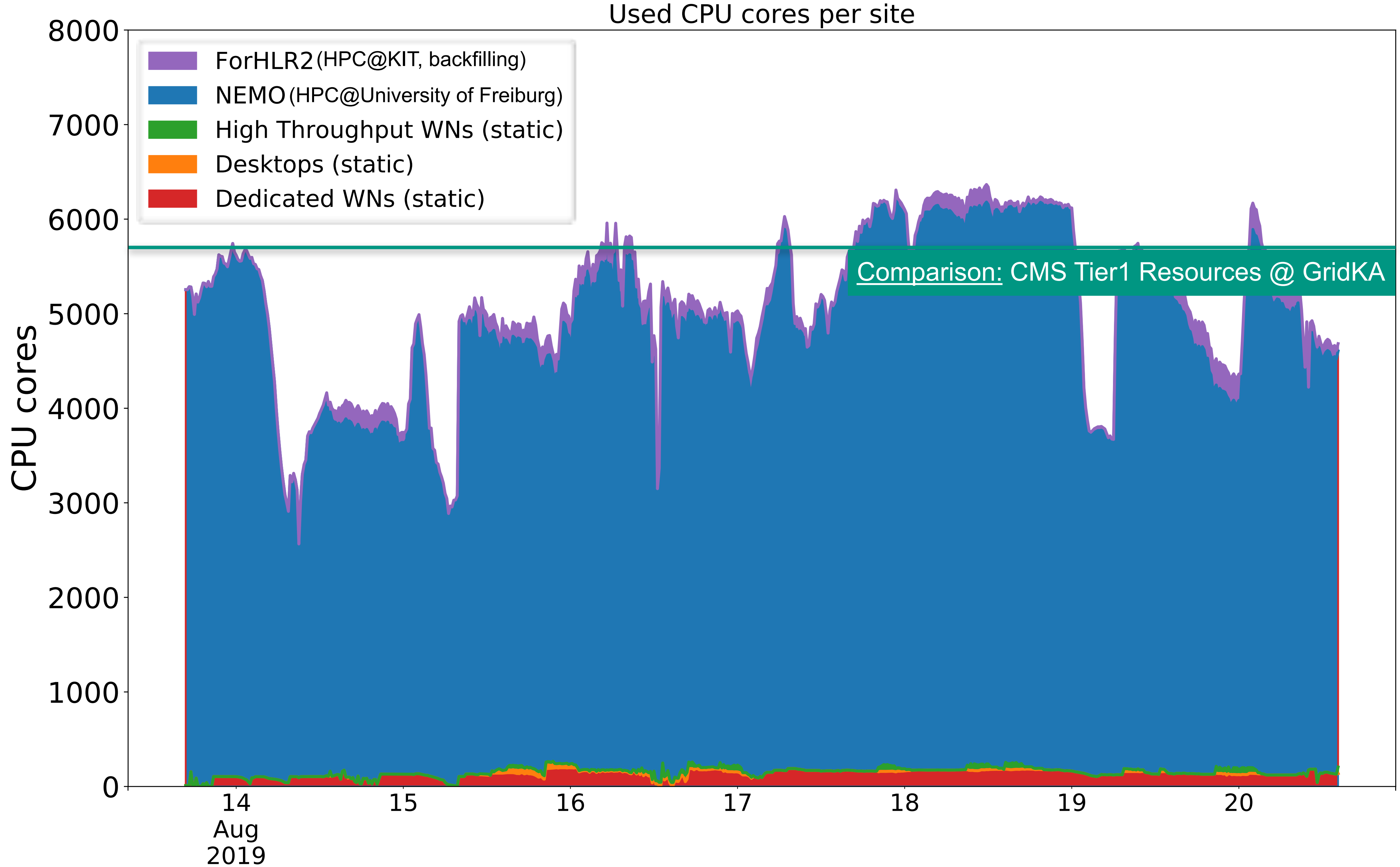
- HPC schedulers are optimized to schedule large scientific simulation and calculation workflows (Many cores/nodes)
- Leads inevitably to unused resources due to draining of machines and fragmentation
- Backfilling with small and short running HEP jobs increases the overall resource utilization
- Enable pre-emption to free resources if needed







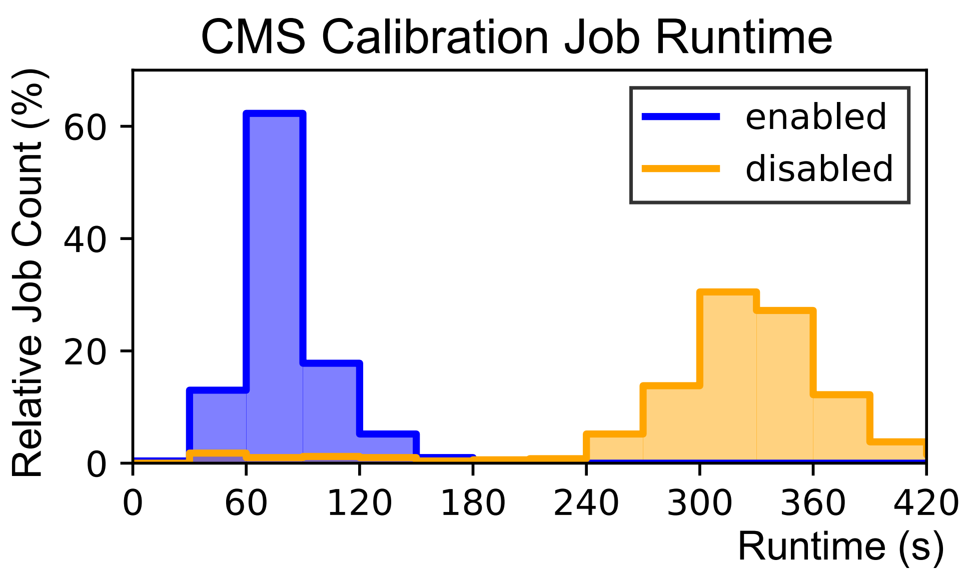
Opportunistic "TIER 1" for a Day



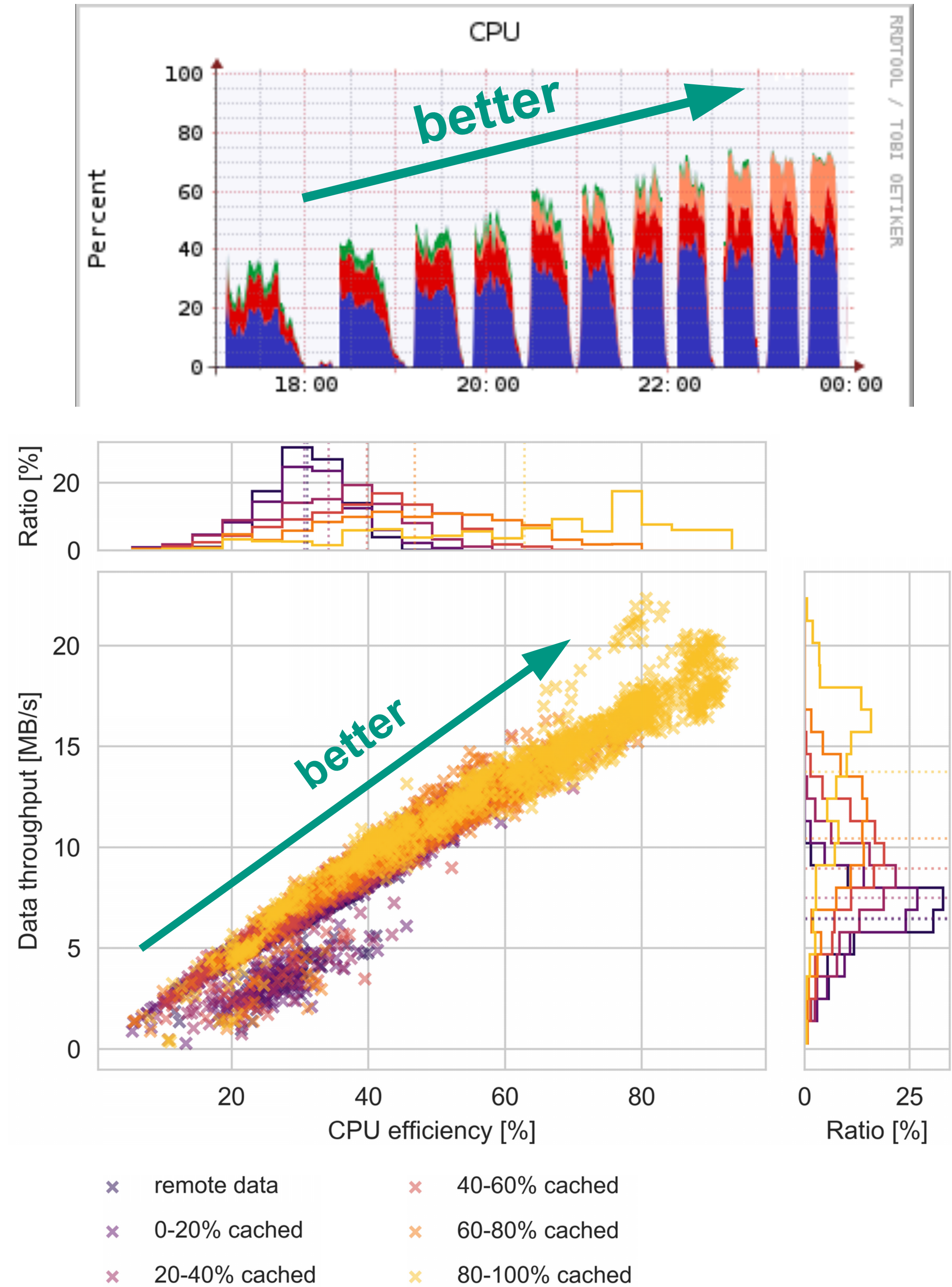
➜ Production ready software at scale in stable operation!

Data Caches

- CPU efficiency/runtime strongly depends on available I/O bandwidth
- Opportunistic sites have potentially slower WAN connections
- Many opportunistic sites offering access to fast storage systems (HPC, S3, etc.)
- Utilise data caches to enable opportunistic sites for recurrent I/O-intense workflows
- Transparent data access was also a hot topic in the Helix Nebula Science Cloud



CMS User Analysis



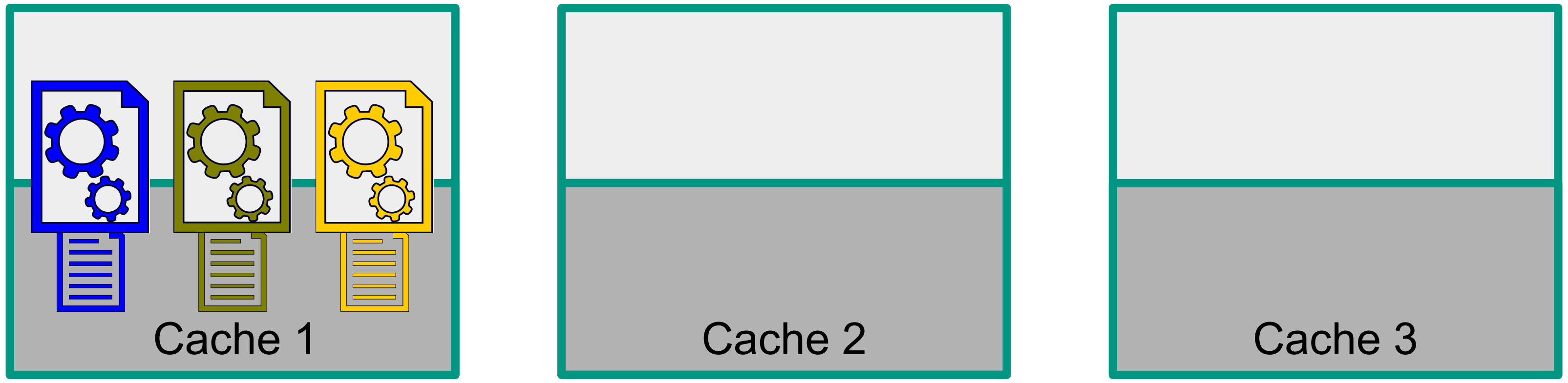
Caching

- Common solution for repeated access to the same data
- Cache data as close as possible to the CPU

Suitable for:

- HEP workflows that process the same data frequently
- CPU resources without permanent storage

Problematic on distributed resources with multiple caches



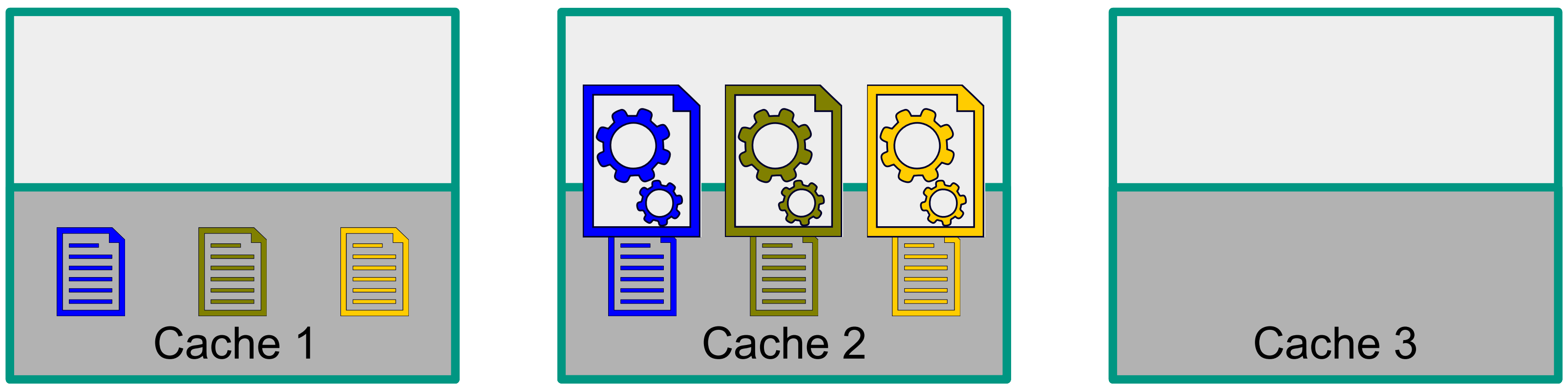
Caching

- Common solution for repeated access to the same data
- Cache data as close as possible to the CPU

<u>Suitable for:</u>

  - HEP workflows that process the same data frequently
  - CPU resources without permanent storage

Problematic on distributed resources with multiple caches



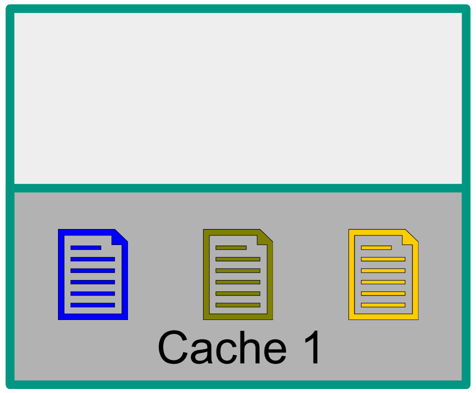
Caching

- Common solution for repeated access to the same data
- Cache data as close as possible to the CPU

Suitable for:

- HEP workflows that process the same data frequently
- CPU resources without permanent storage

Problematic on distributed resources with multiple caches



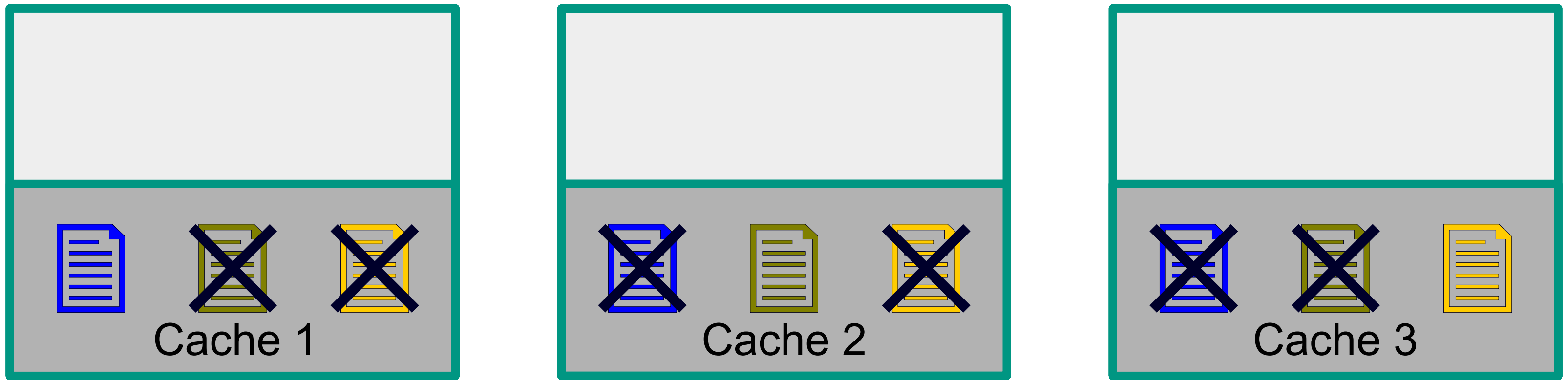
Caching

- Common solution for repeated access to the same data
- Cache data as close as possible to the CPU

<u>Suitable for:</u>

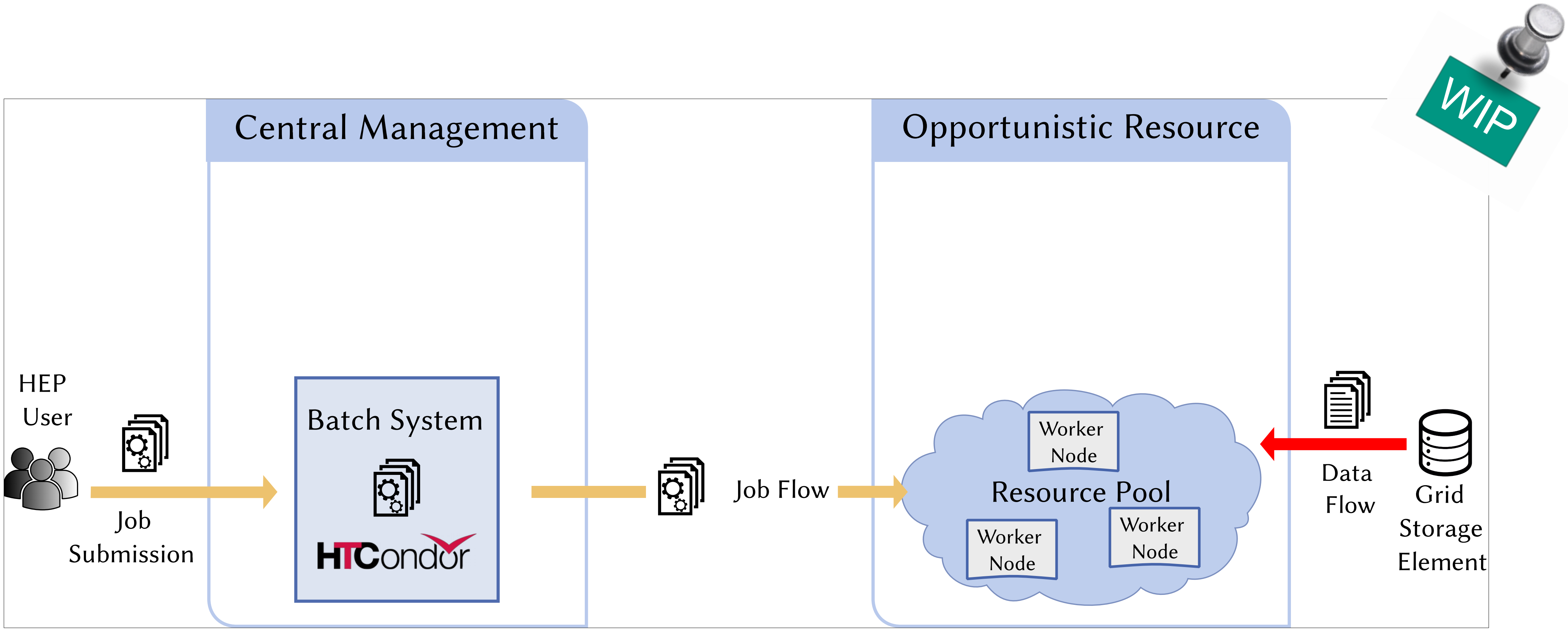
  - HEP workflows that process the same data frequently
  - CPU resources without permanent storage

Problematic on distributed resources with multiple caches



Waste of storage capacity due to replication of data!
Caches must be coordinated!

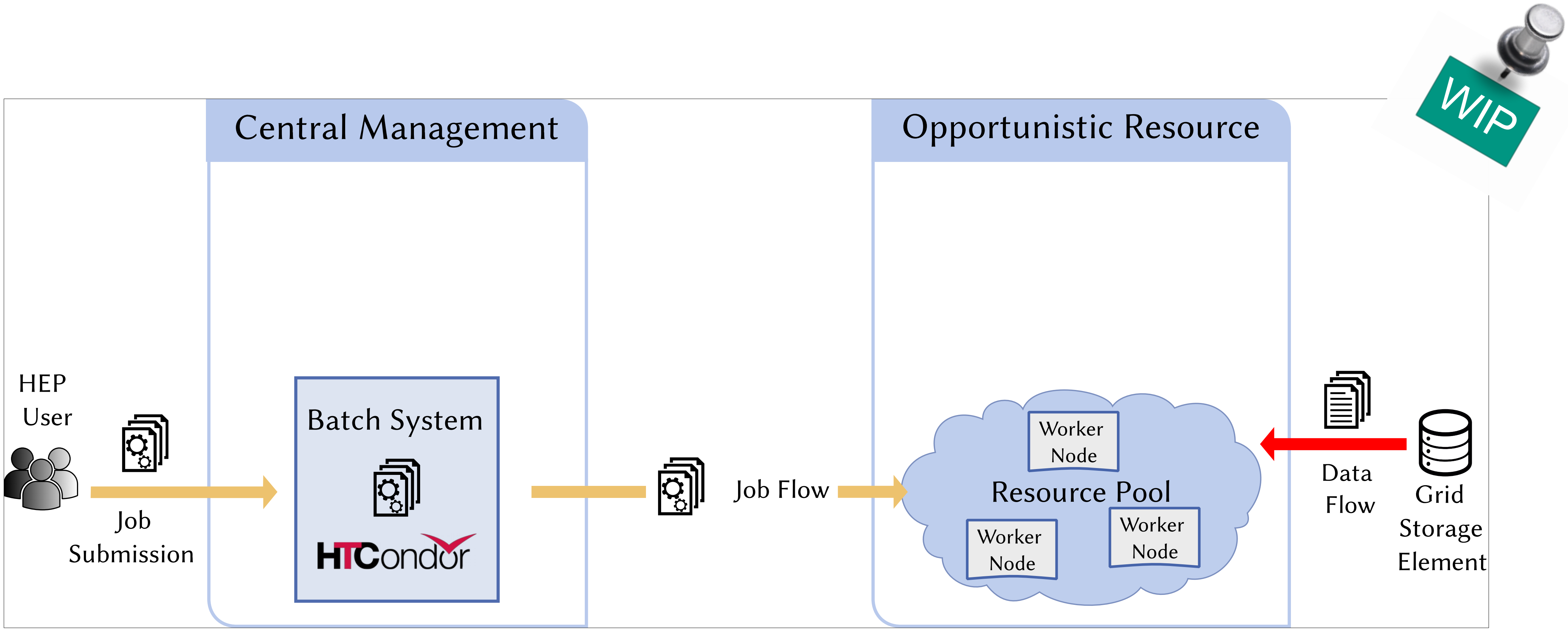
Coordinated Caching



Longterm experiences @ KIT:
"Data Locality via Coordinated Caching for Distributed Processing",
M. Fischer et al., J. Phys.: Conf. Ser.762 012011 (2016)

Coordinated Caching

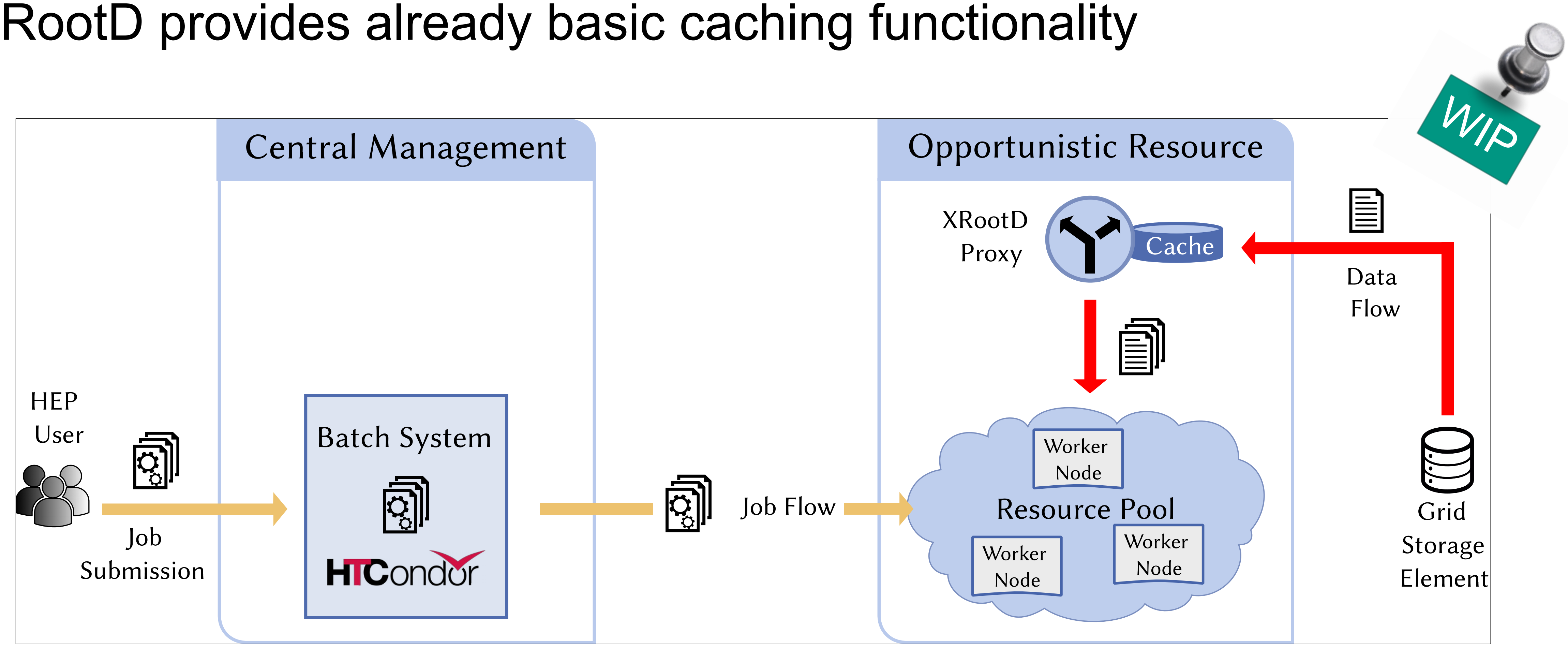
- HTCondor schedules jobs to resources



Longterm experiences @ KIT:
"Data Locality via Coordinated Caching for Distributed Processing",
M. Fischer et al., J. Phys.: Conf. Ser.762 012011 (2016)

Coordinated Caching

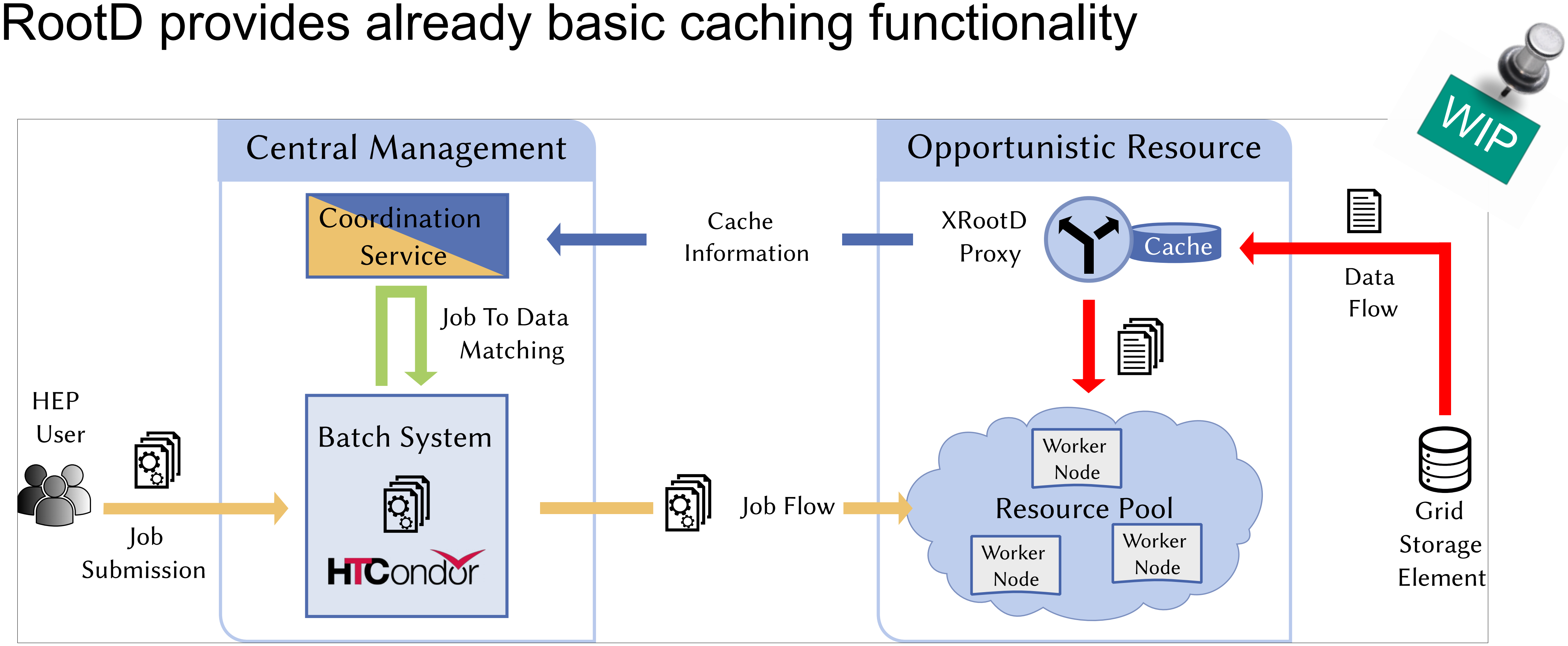
- HTCondor schedules jobs to resources
- XRootD provides already basic caching functionality



Longterm experiences @ KIT:
"Data Locality via Coordinated Caching for Distributed Processing",
M. Fischer et al., J. Phys.: Conf. Ser.762 012011 (2016)

Coordinated Caching

- HTCondor schedules jobs to resources
- XRootD provides already basic caching functionality

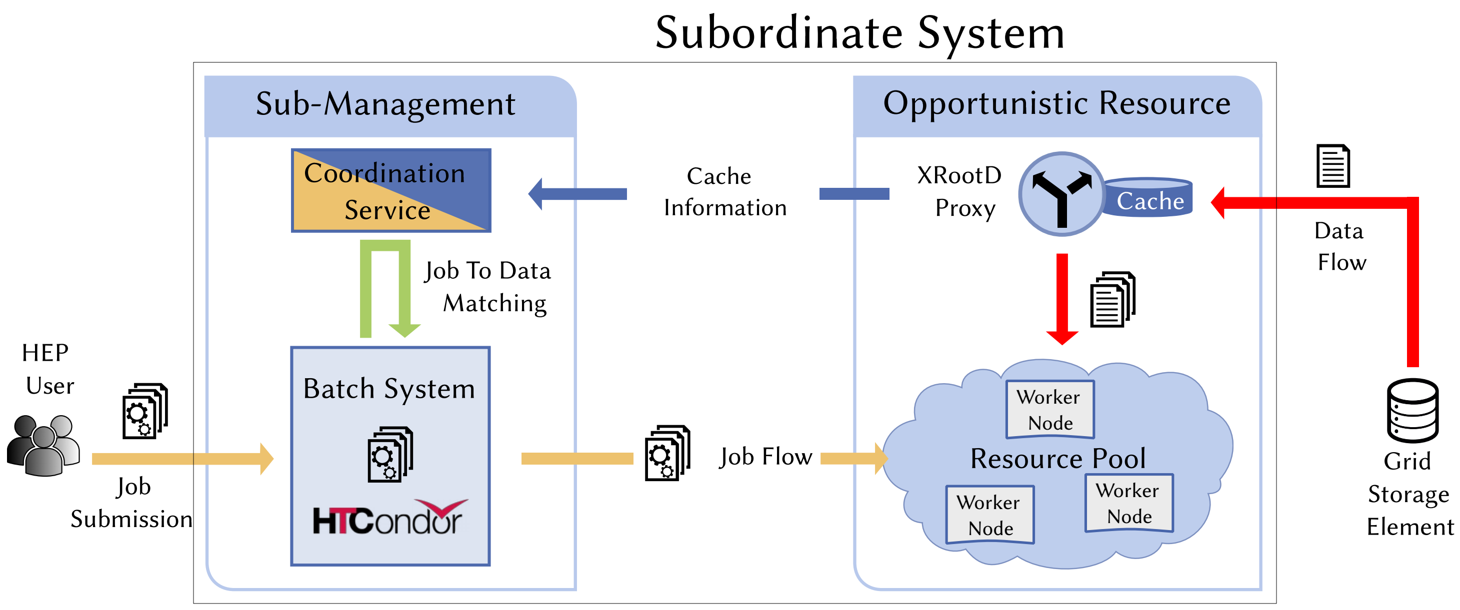


- NaviX coordination service currently under development at KIT
  - Implicit data placement via job scheduling
  - Schedule jobs to cached data

Longterm experiences @ KIT:
"Data Locality via Coordinated Caching for Distributed Processing",
M. Fischer et al., J. Phys.: Conf. Ser.762 012011 (2016)

Coordinated Caching

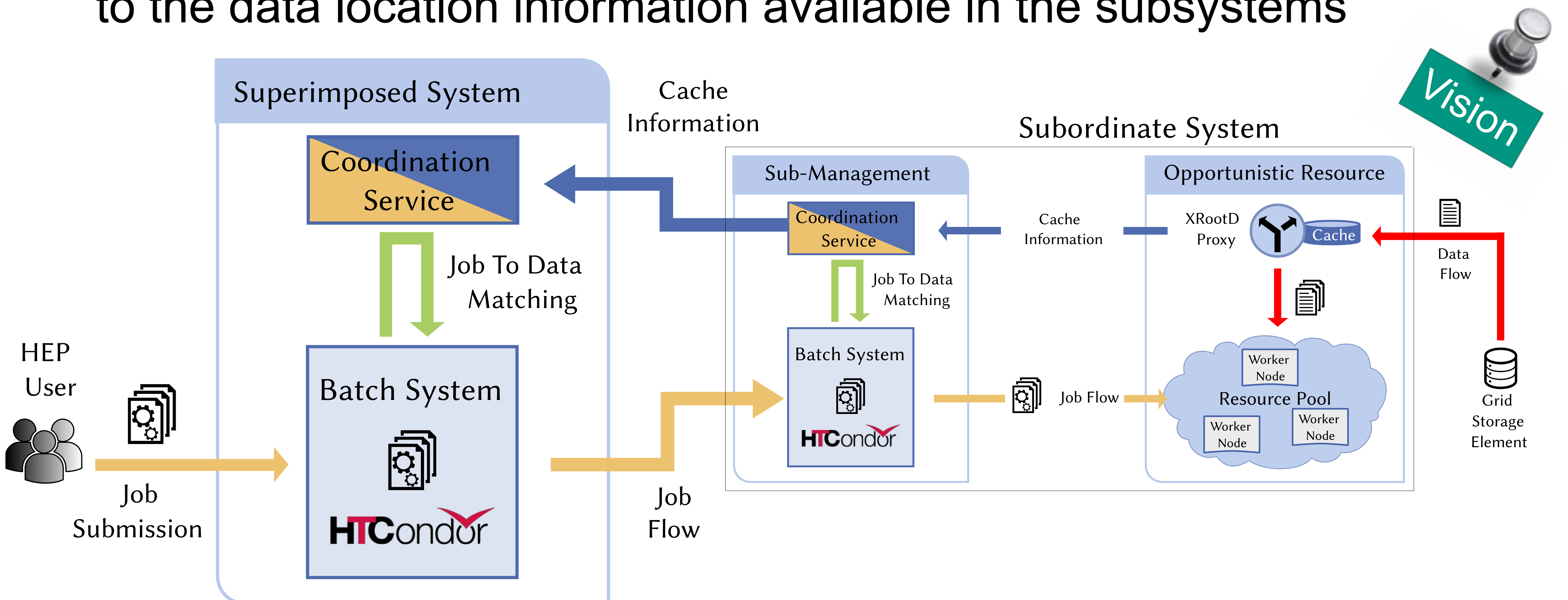
<u>Vision:</u> Build a hierarchical system of local, site and national caches



Scalability by Design

Vision: Build a hierarchical system of local, site and national caches

- XRootD and HTCondor take care of hierarchical upscaling
- Job-to-cache coordination can be performed on all levels with regard to the data location information available in the subsystems



... National collaboration | Regional collaboration | Institute resources | Local cluster

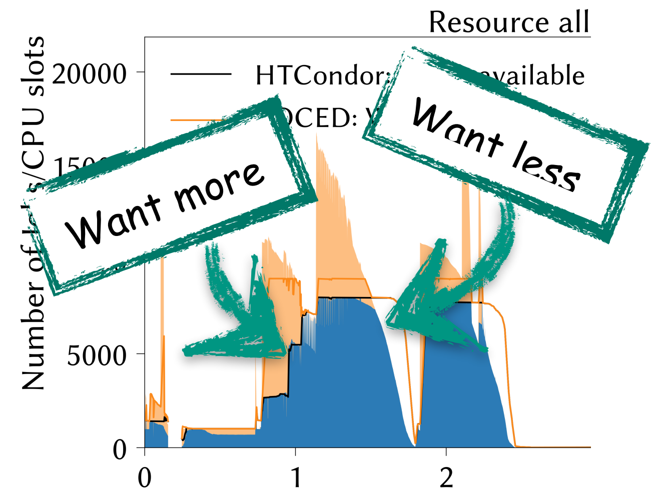
Conclusion

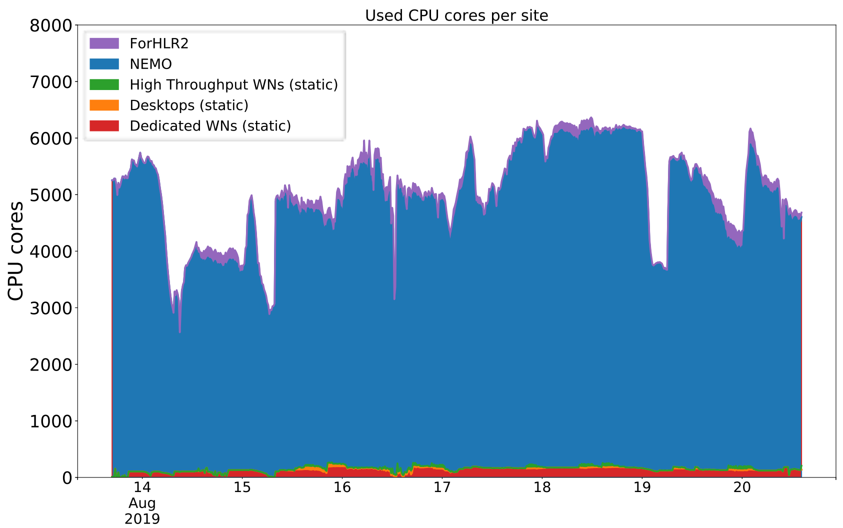
- **Dynamic on-demand provisioning of resources**
  - COBalD/TARDIS resource manager developed at KIT
  - Enables transparent and dynamic on-demand provisioning of opportunistic resources
  - Enables backfilling of HPC resources
  - Production ready software at scale

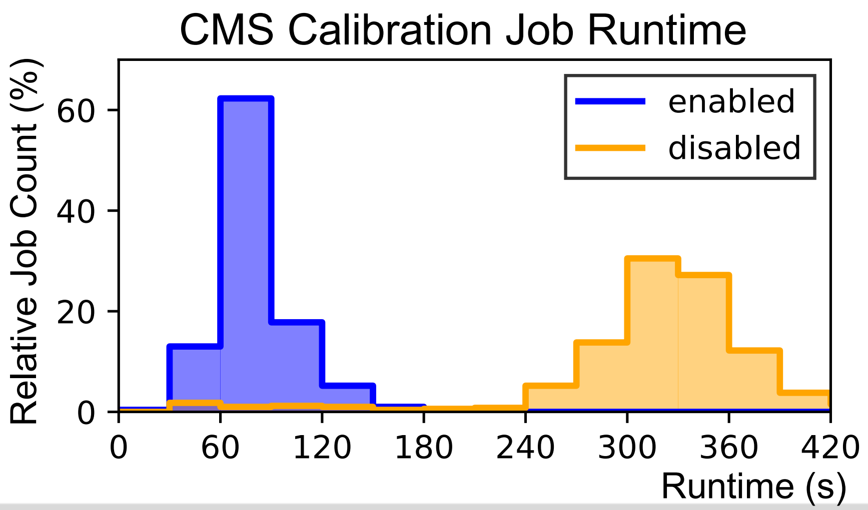
COBalD

TARDIS

- **Distributed coordinated caching**
  - NaviX coordination service in development at KIT
  - Working towards a scalable caching solution based upon HTCondor and XRootD
  - First prototype available

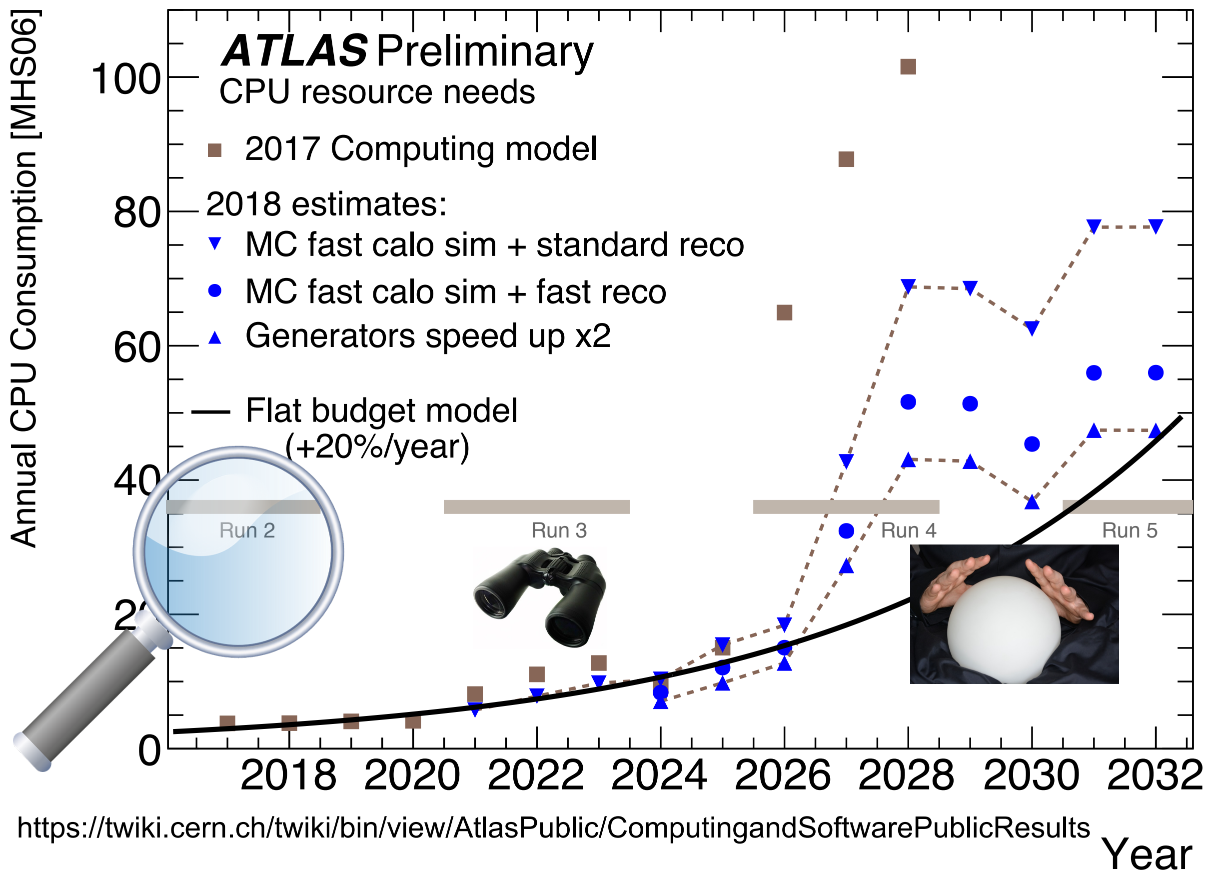






BACKUP

HEP Computing Challenges Ahead



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults

- ATLAS/CMS CPU resource estimates
- Assuming flat budget and 20% technology advance per year
- CPU and storage resource shortfall between needs and technology in 2027
- ⊕ Politically endorsed integration of cloud and high performance computing resources (US)
- → (R)evolution of computing model is required to master future challenges

<u>New interesting research topics:</u>

- Exploitation of modern technologies
- Improvement of algorithms and utilization of ML
- Dynamic integration of opportunistic resources (HPC, cloud, volunteer computing)
- Data lakes and data caching technologies

Transparent Integration: Overlay Batch System (OBS)

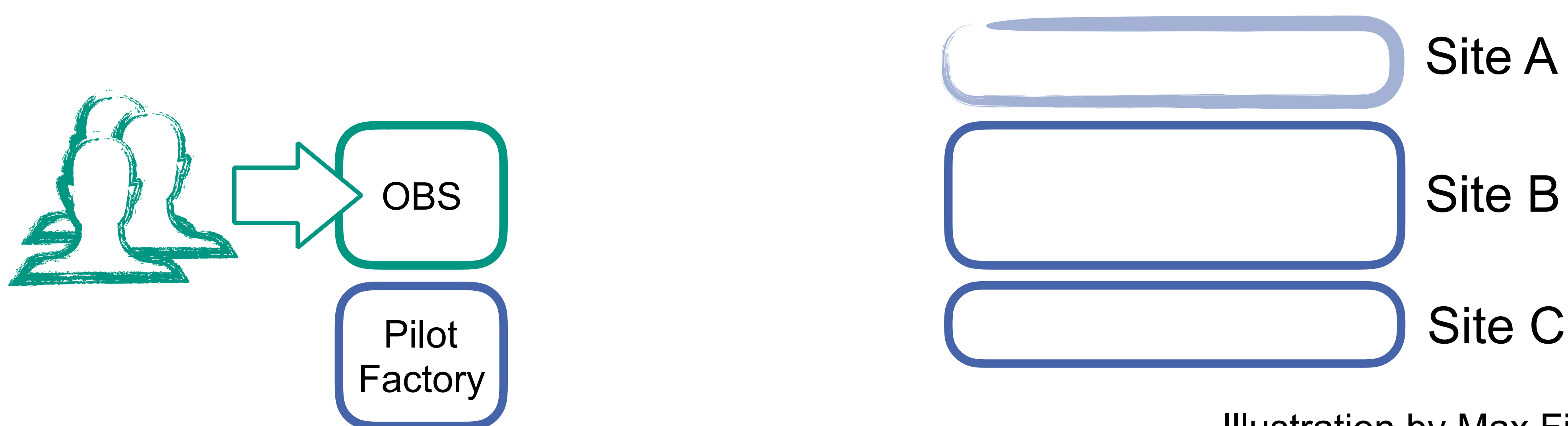


Illustration by Max Fischer (KIT)

Transparent Integration: Overlay Batch System (OBS)

Or how the Grid is used today:

Illustration by Max Fischer (KIT)

Transparent Integration: Overlay Batch System (OBS)

Or how the Grid is used today:

- Pilot factory submits placeholder jobs (pilots) to different sites

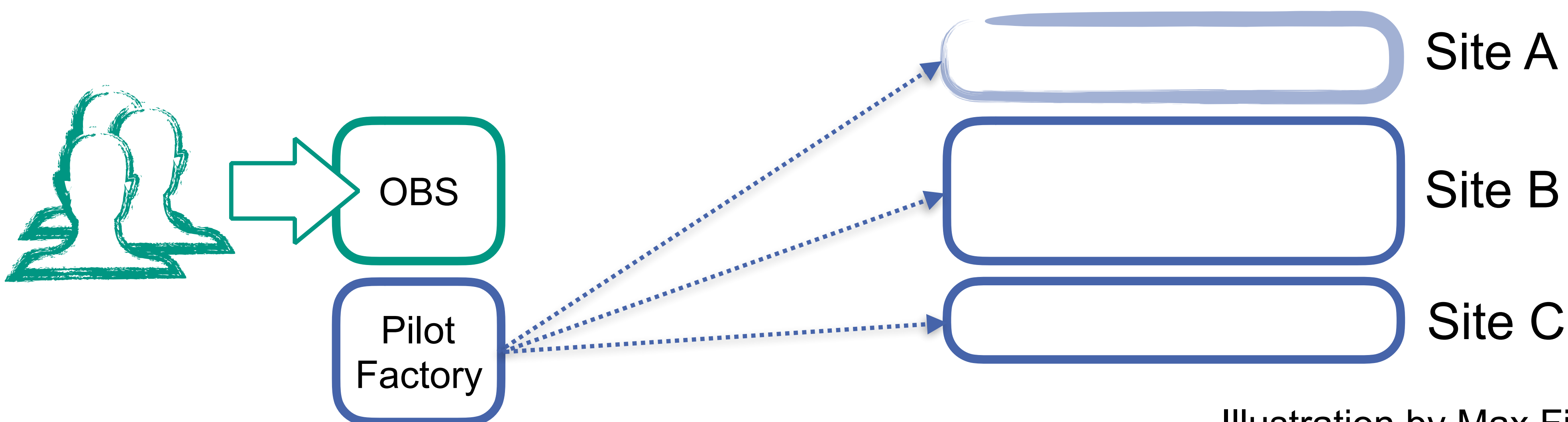


Illustration by Max Fischer (KIT)

Transparent Integration: Overlay Batch System (OBS)

Or how the Grid is used today:

- Pilot factory submits placeholder jobs (pilots) to different sites
- Pilot allocates resources at the site

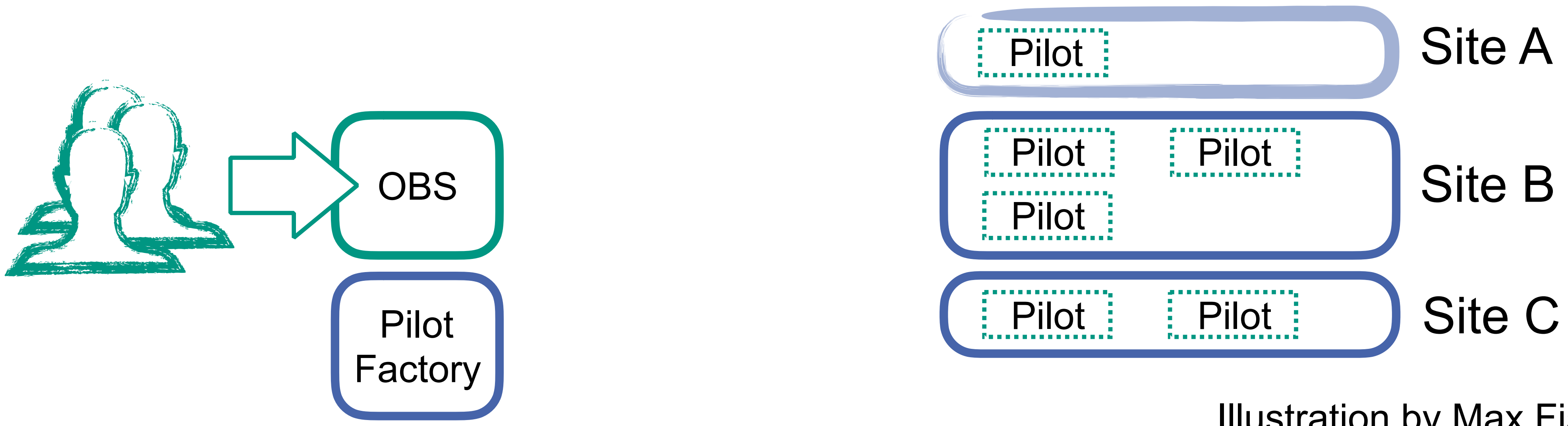


Illustration by Max Fischer (KIT)

Transparent Integration: Overlay Batch System (OBS)

Or how the Grid is used today:

- Pilot factory submits placeholder jobs (pilots) to different sites
- Pilot allocates resources at the site
- Resources are integrated into the OBS
- Workload is pulled from the OBS

➡ Users interact only with one single-point-of-entry the OBS

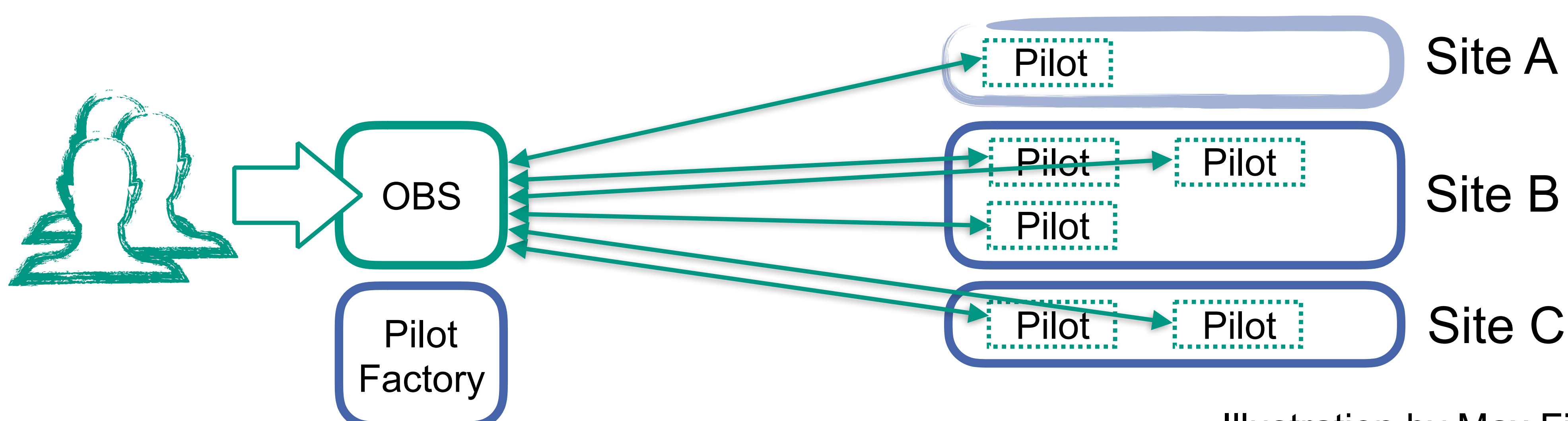
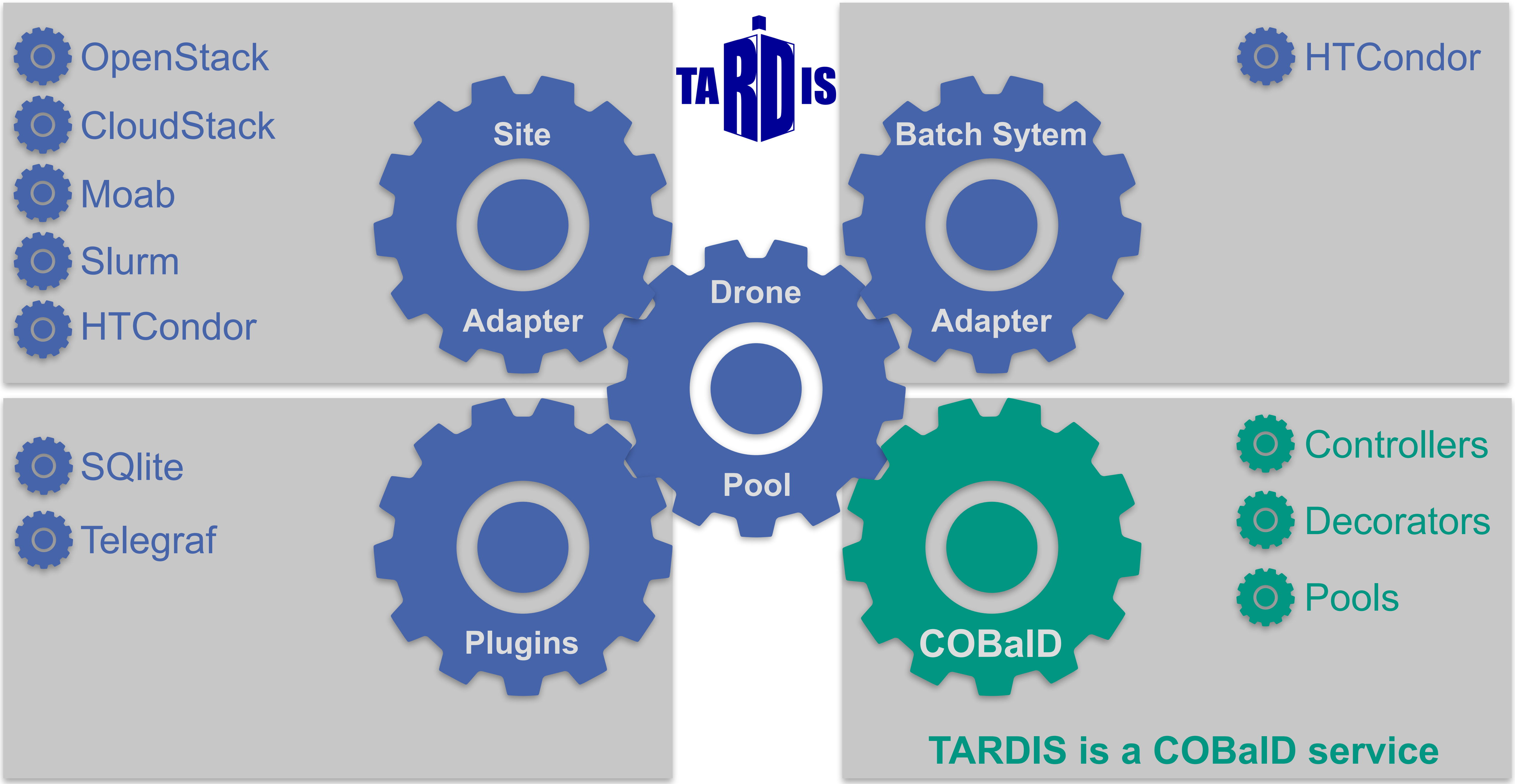
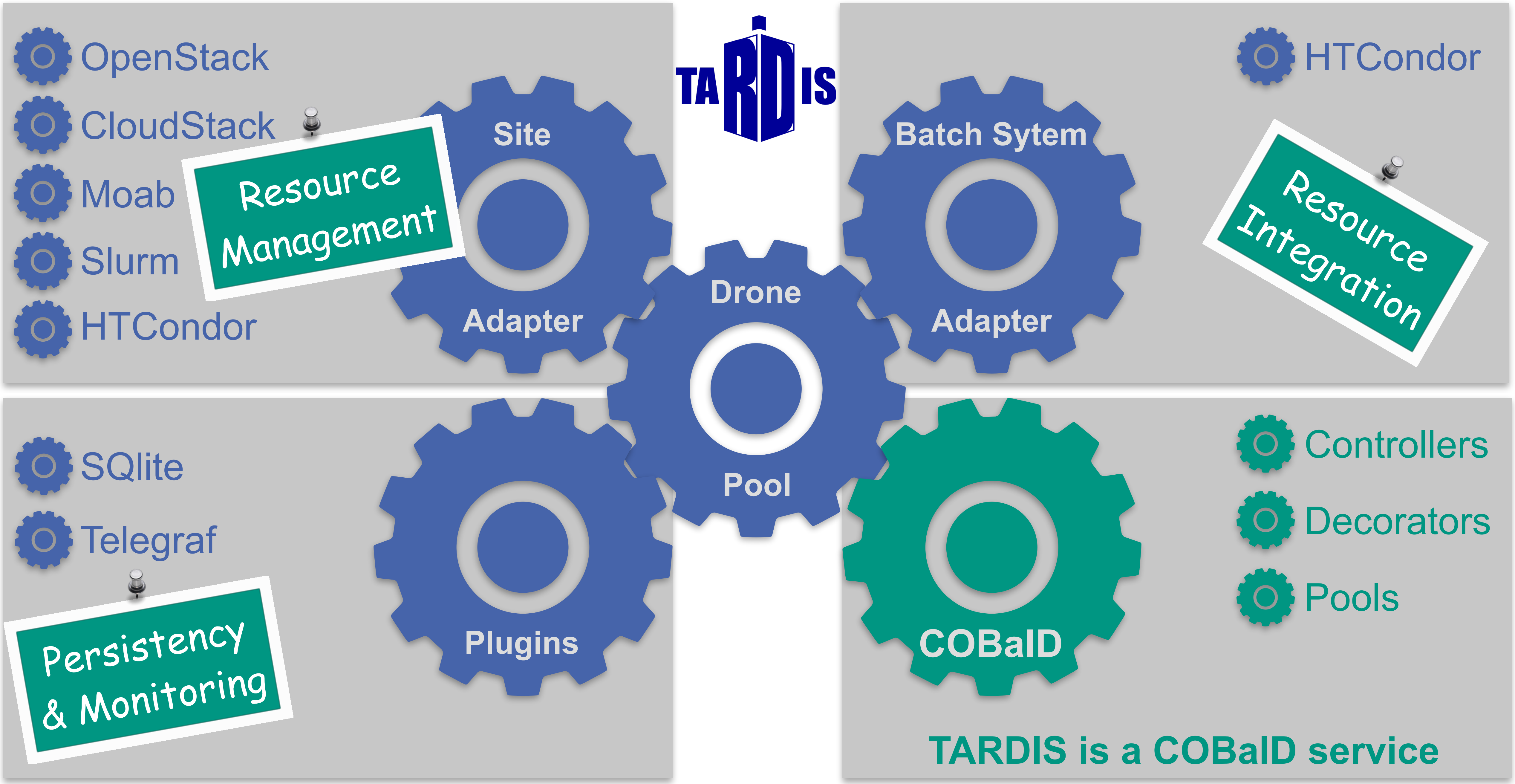


Illustration by Max Fischer (KIT)

Overview about COBalD/TARDIS

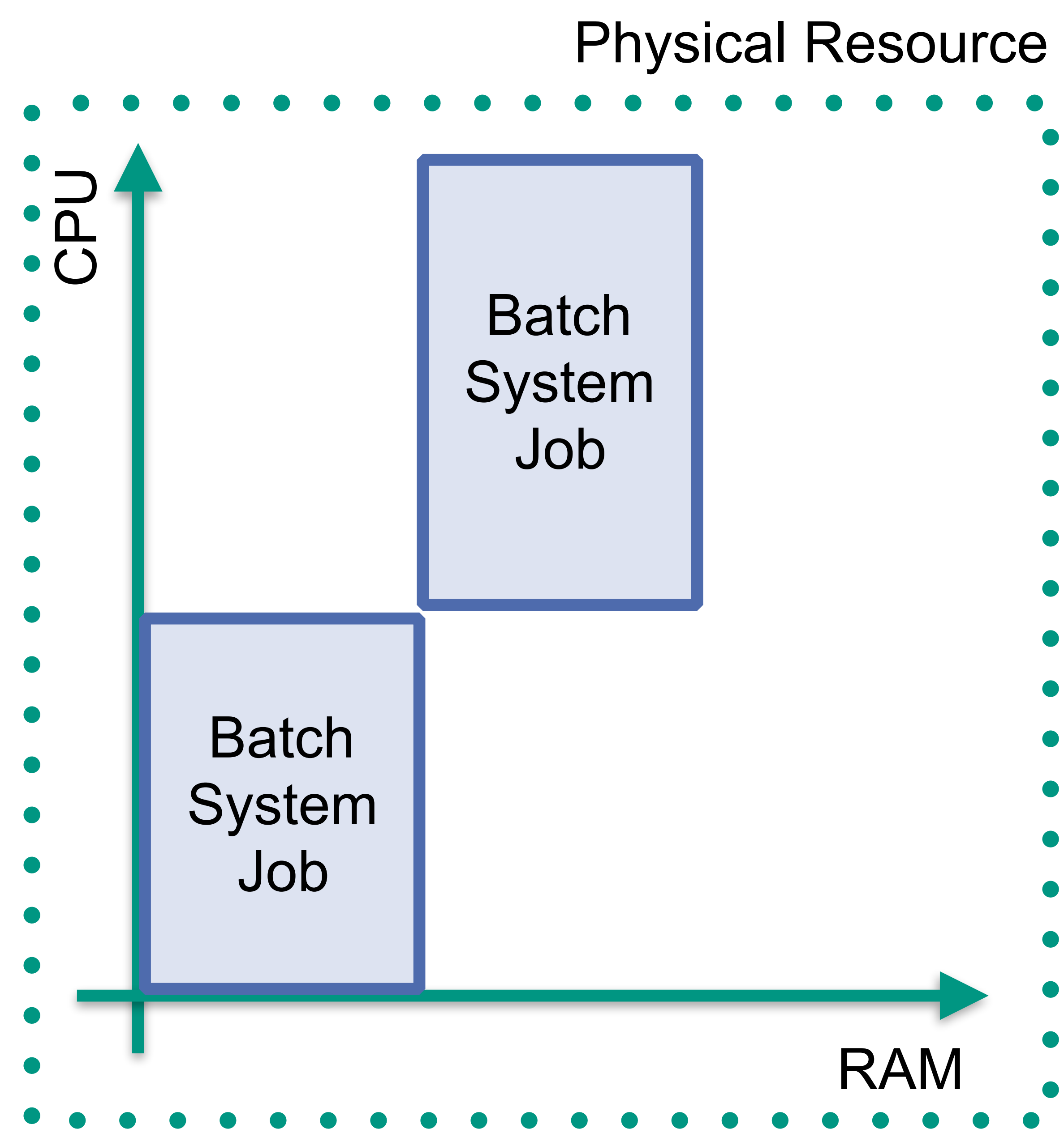


Overview about COBalD/TARDIS

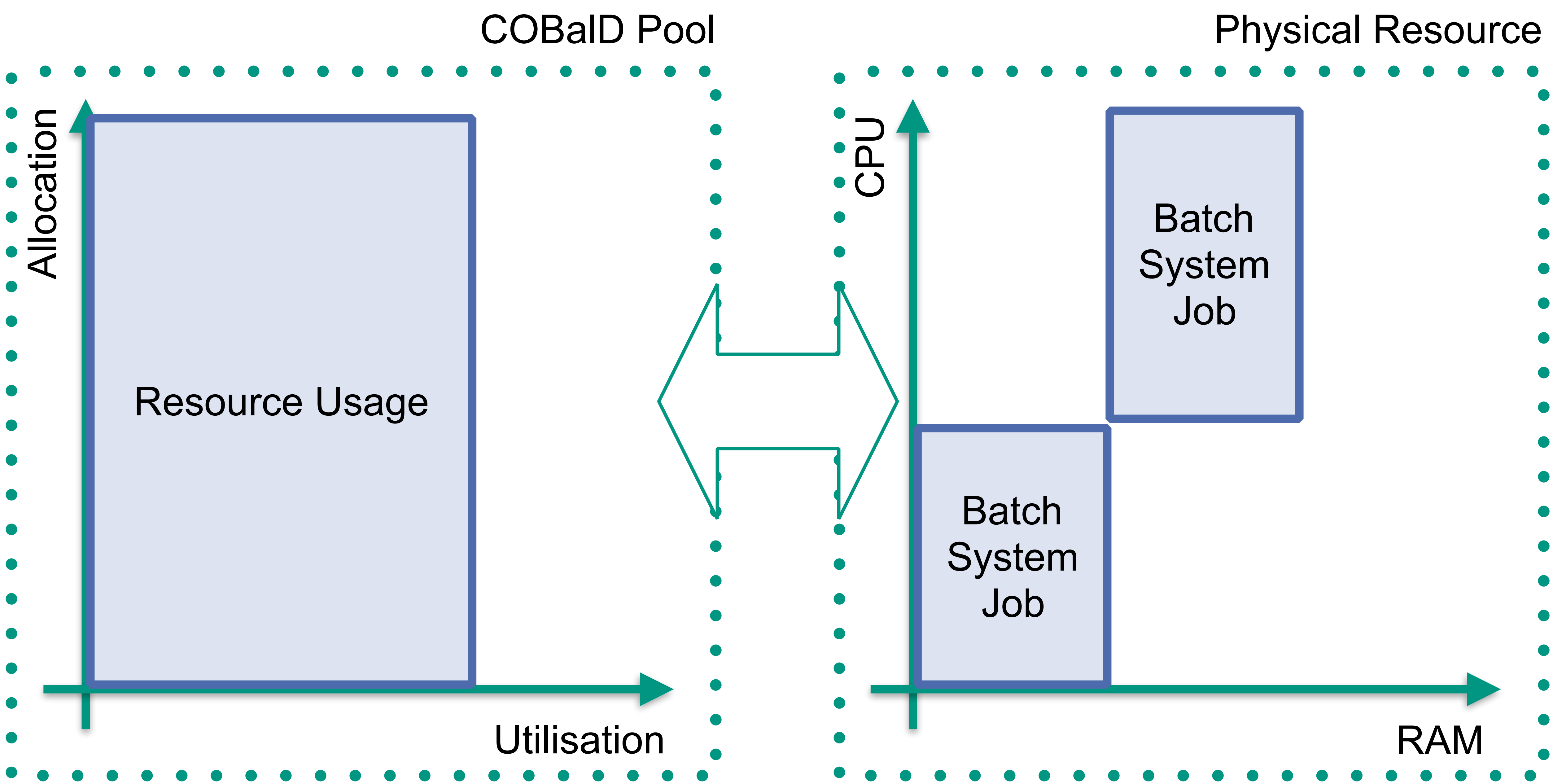


➜ Easily extendable by design through its modular structure

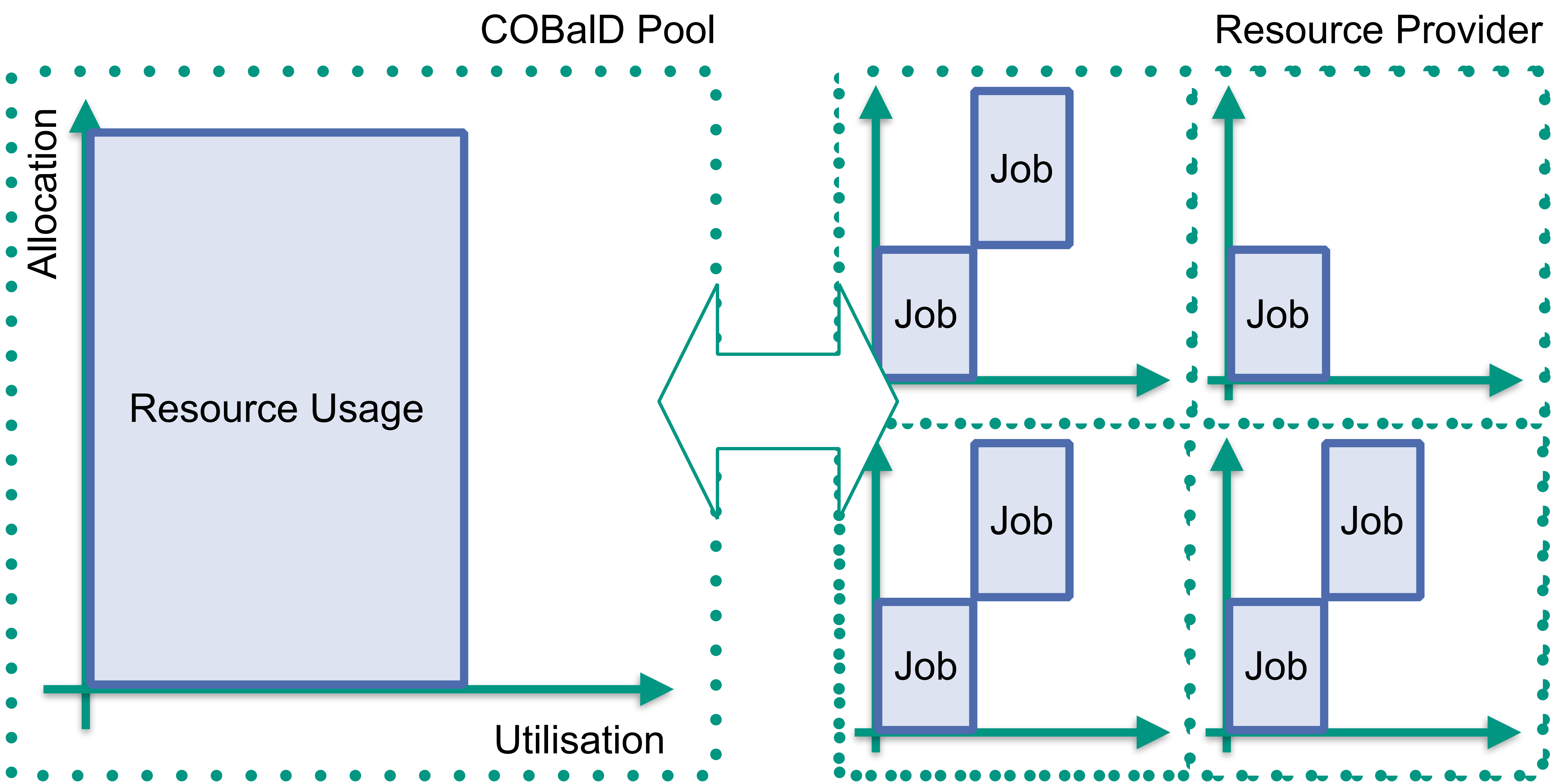
COBalD Resource Pool Model



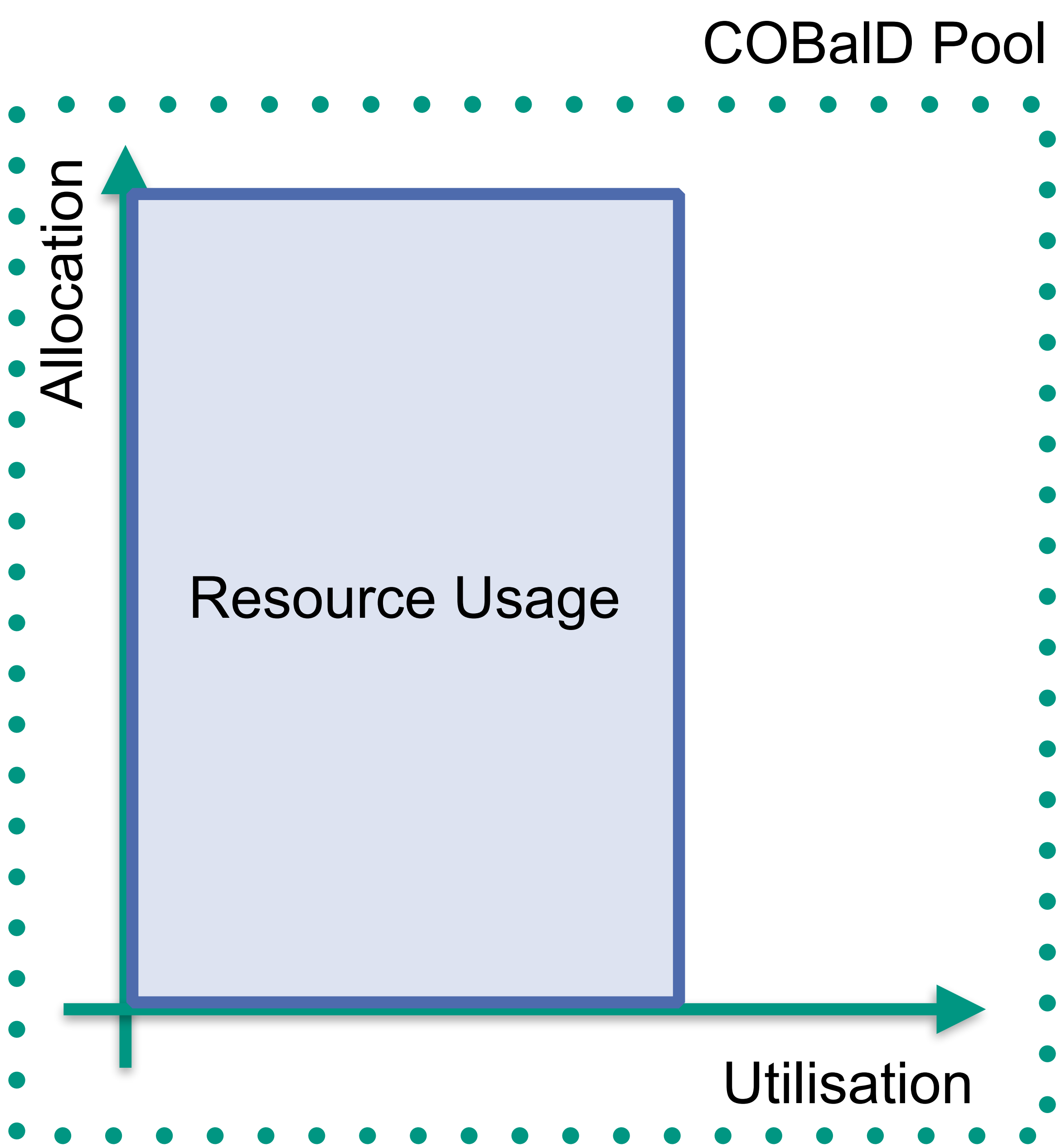
COBalD Resource Pool Model



COBalD Resource Pool Model

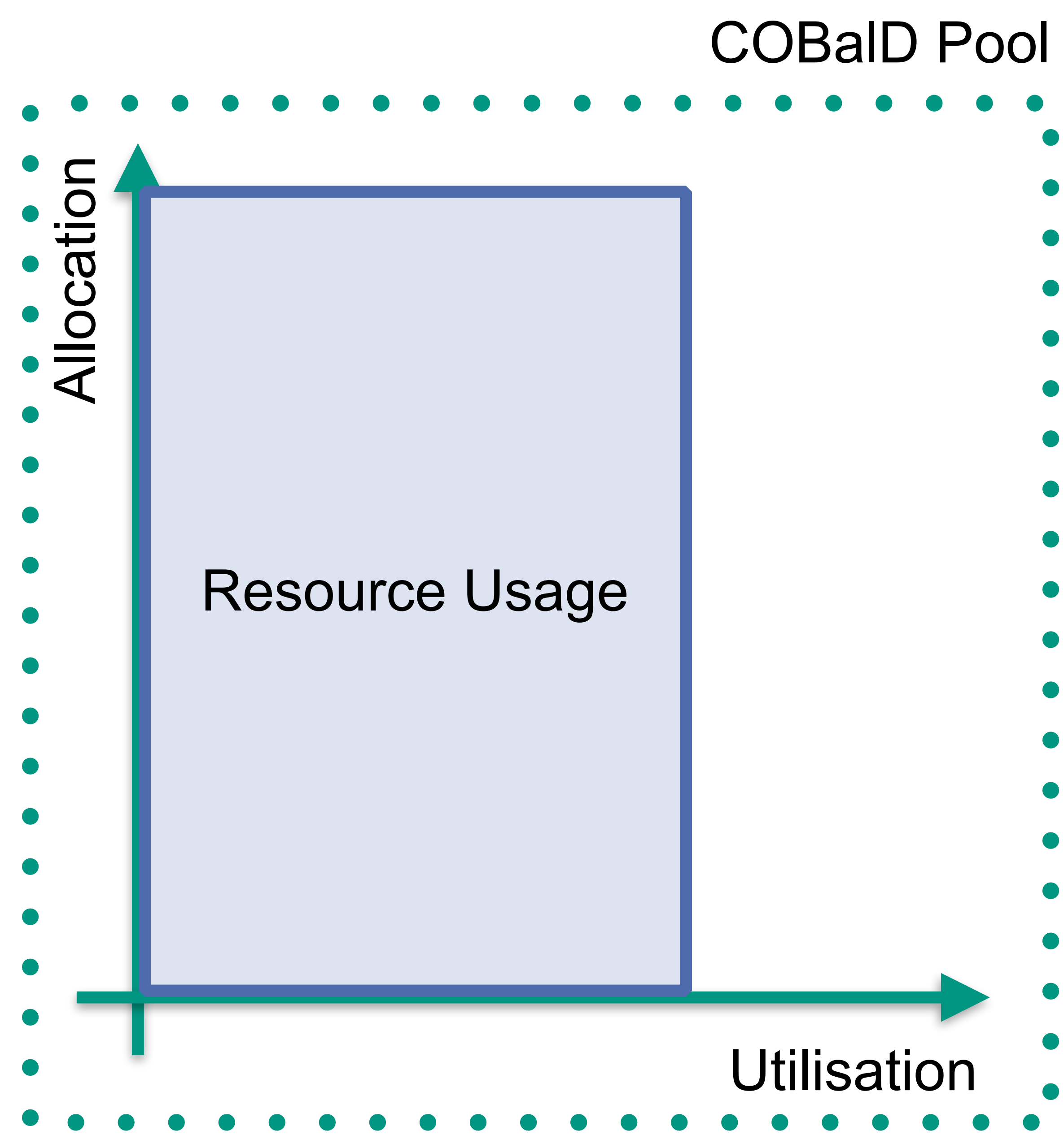


COBalD Resource Pool Model

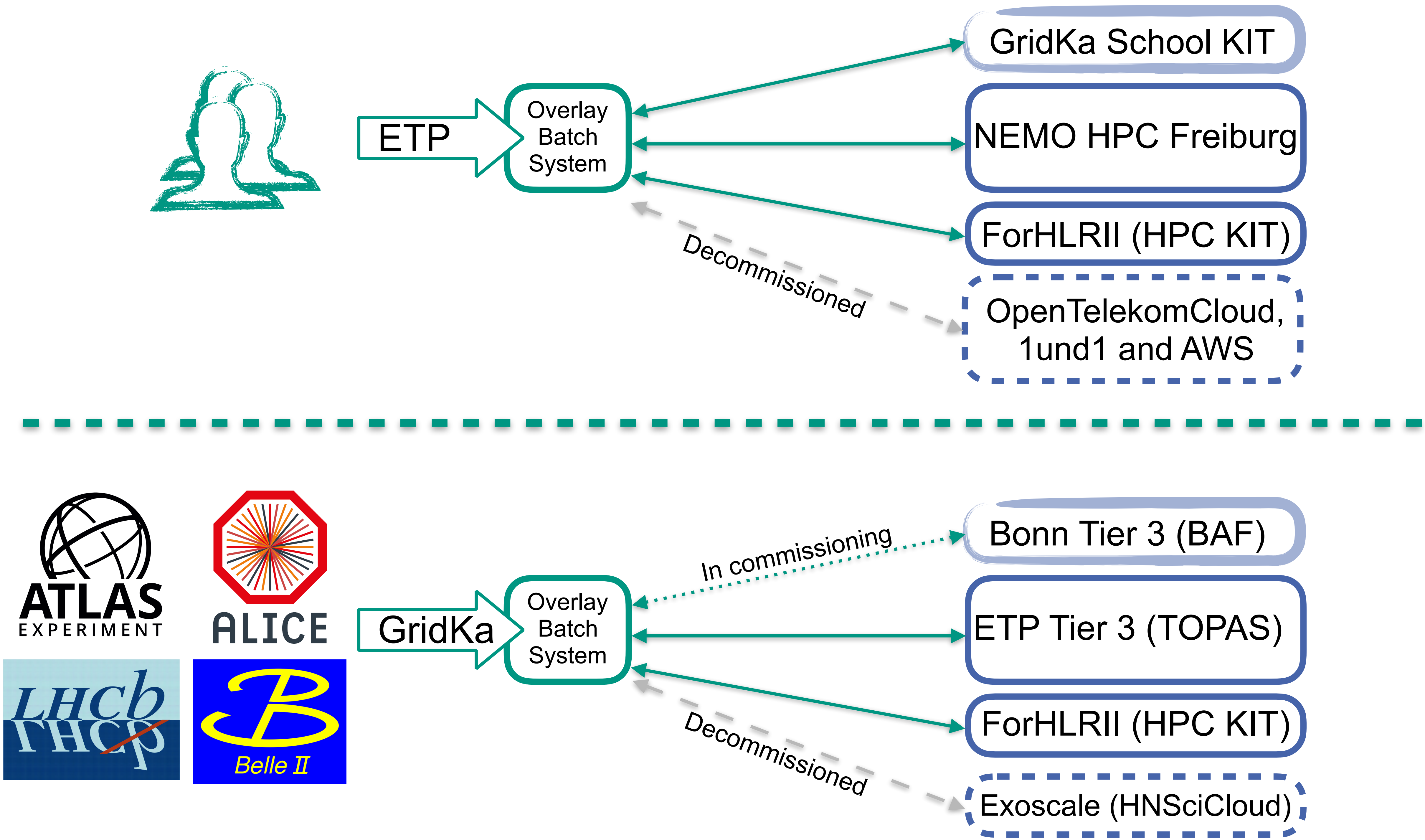


COBalD Resource Pool Model

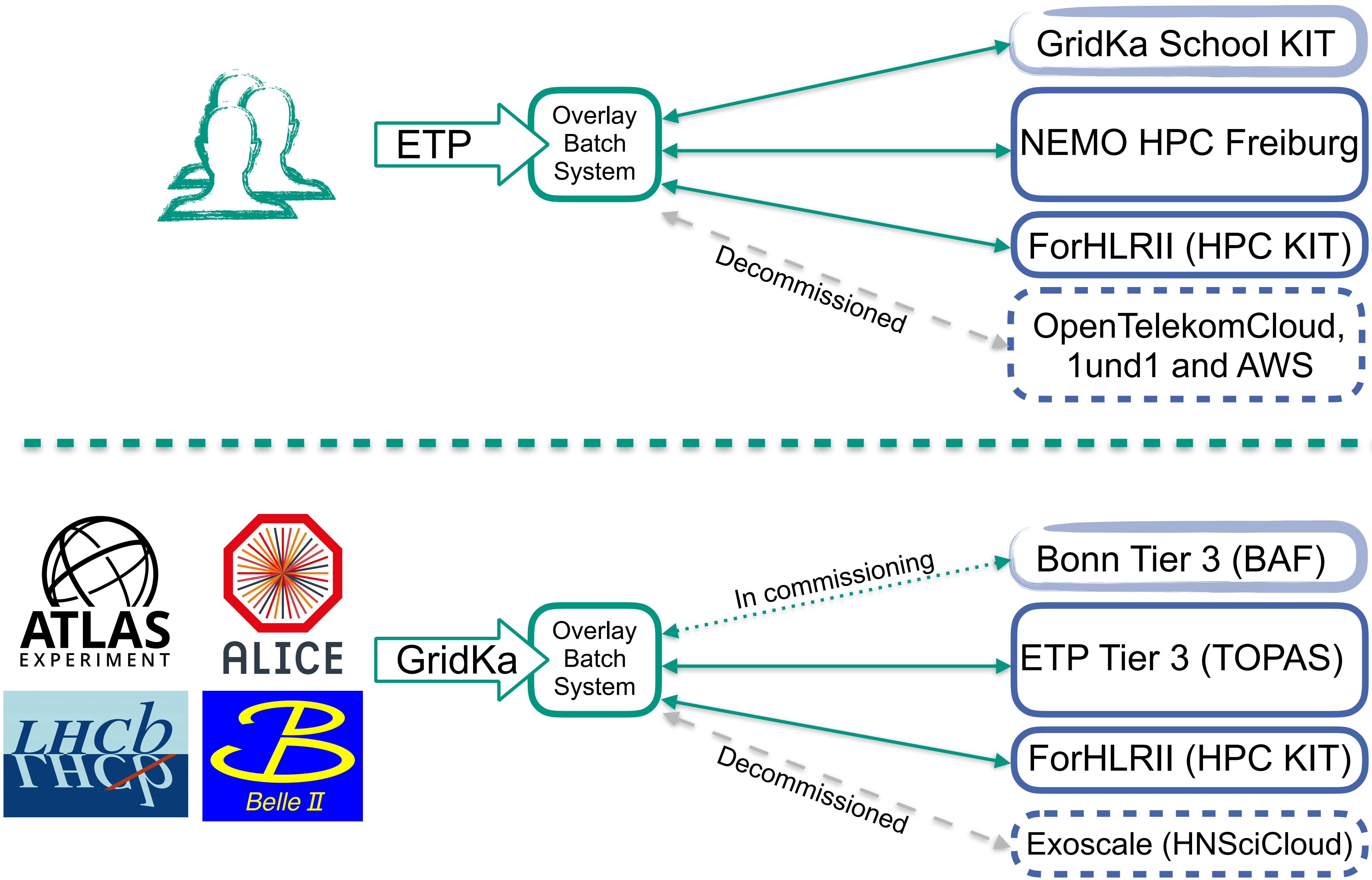
```
if utilisation < self.low_utilisation:
    return supply * self.low_scale
elif allocation > self.high_allocation:
    return supply * self.high_scale
```



Currently Available Opportunistic Resources

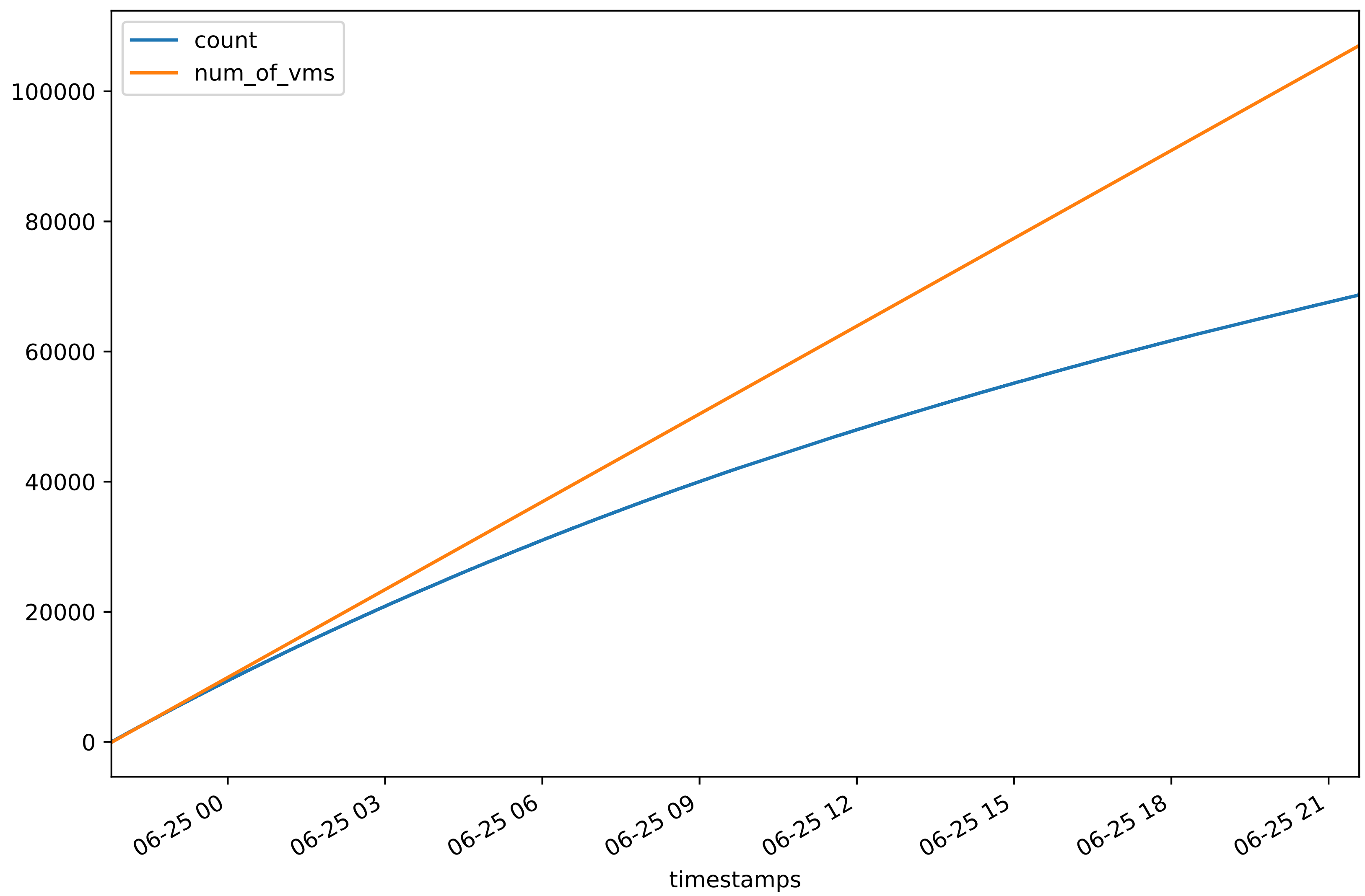


Currently Available Opportunistic Resources



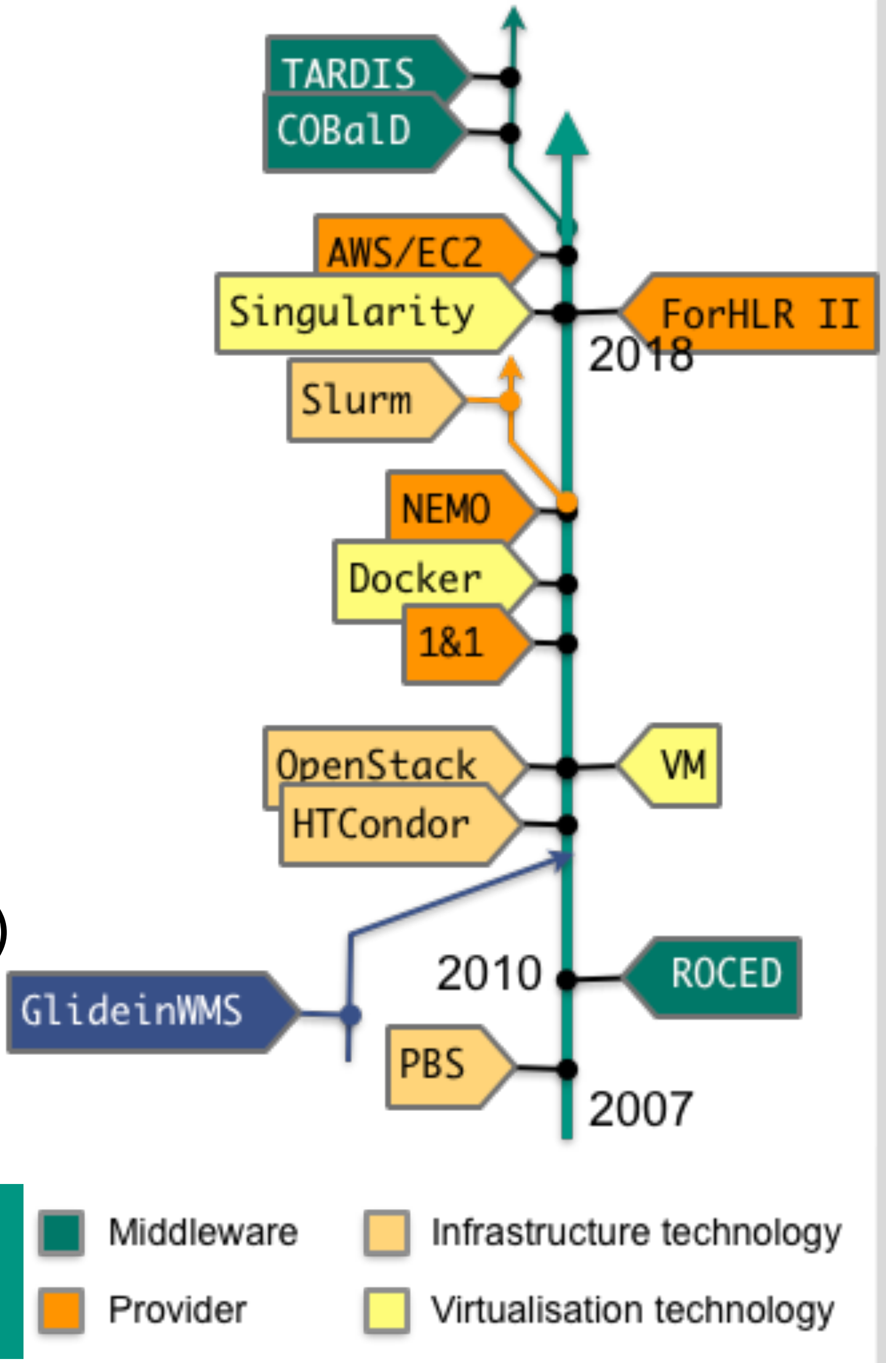
➜ Perfectly suited to transparently integrate various resources into WLCG computing

Scalability



Opportunistic Resources @ KIT (2007-present)

- Longterm experience with opportunistic resources, virtualization and containers
- Software development (Cloud Resource Manager) ViBatch, ROCED (➠COBalD/TARDIS)[1]
- Opportunistic Resources
  - Institute resources
    - Desktop cluster (Docker)
    - GridKa School Virtual Infrastructure (OpenStack)
  - HPC Cluster
    - ~~IC1@Uni Karlsruhe (ViBatch)~~
    - ForHLR II @ KIT (Singularity)
    - bwFORcluster NEMO @ Uni Freiburg (OpenStack)
  - Commercial cloud providers
    - AWS, 1&1 Cloud Services, OTC, ExoScale

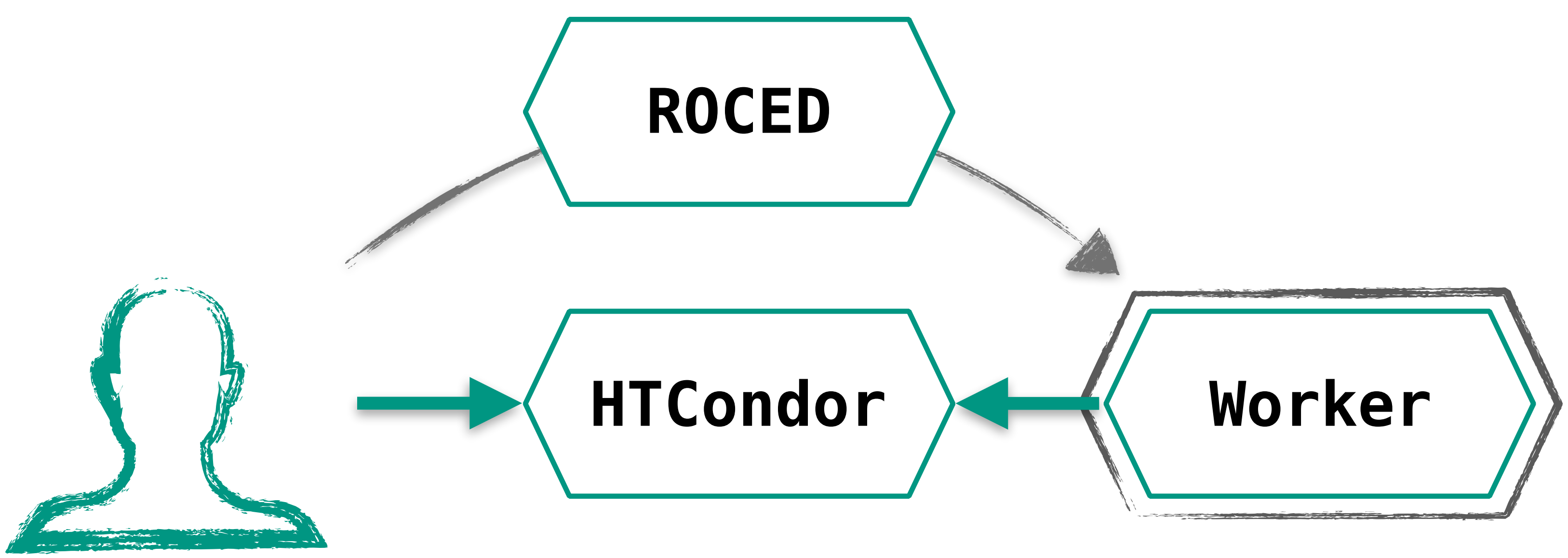


All dynamically and transparently integrated in a single HTCondor instance using ROCED and CoBalD/TARDIS

[1] COBalD - the Opportunistic Balancing Daemon (http://cobald.readthedocs.io/) by M. Fischer & E. Kühn
TARDIS - **T**ransparent **A**daptive **R**esource **D**ynamic **I**ntegration **S**ystem (https://github.com/giffels/tardis)

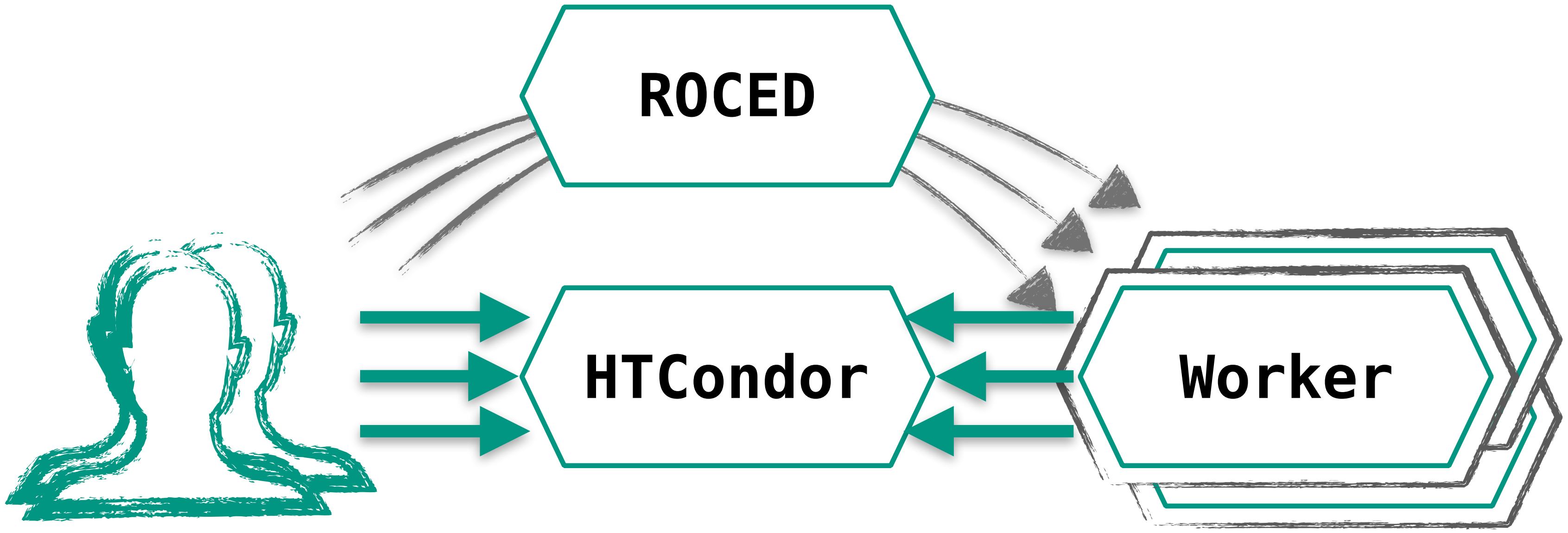
Illustration by E. Kuehn (KIT)

ROCED Resumé



Slide by Max Fischer (KIT)

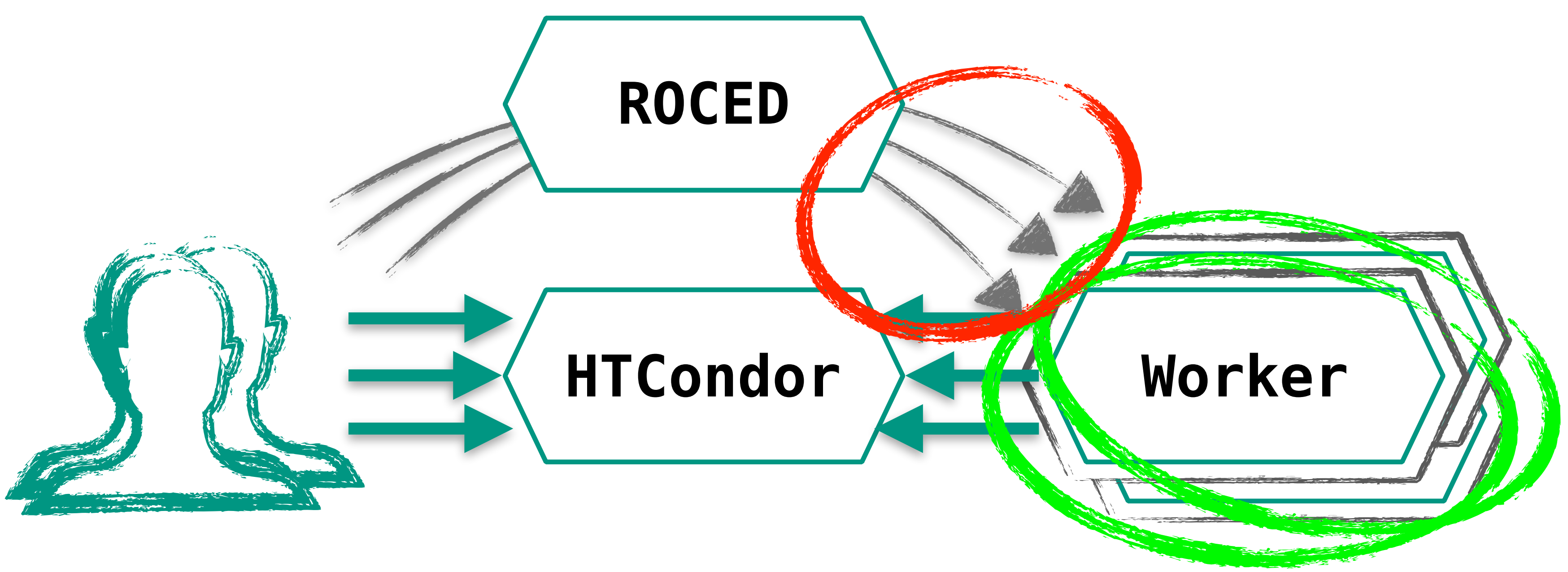
ROCED Resumé



Slide by Max Fischer (KIT)

ROCED Resumé

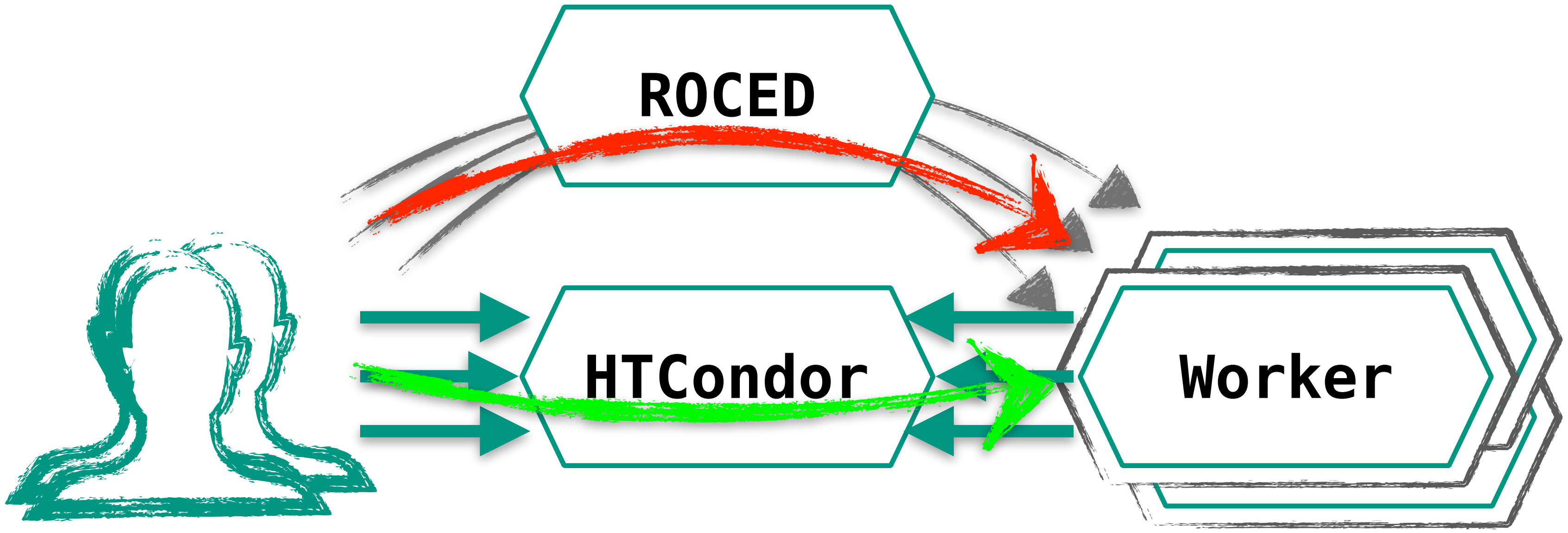
- Dynamic resources matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users



Slide by Max Fischer (KIT)

ROCED Resumé

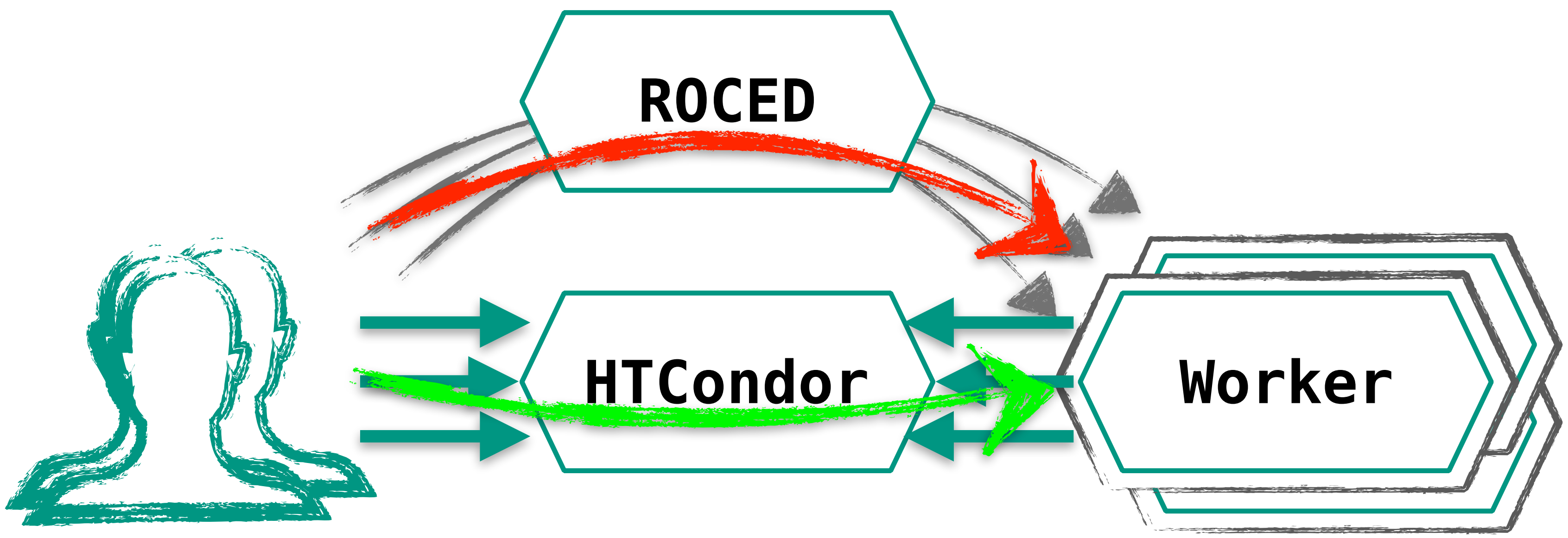
- Dynamic resources matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users
- Resource aggregation in overlay batch system
  - Unreliable to predict resources required for jobs
  - Efficient to integrate resources, then match jobs



Slide by Max Fischer (KIT)

ROCED Resumé

- Dynamic resources matching user demand
  - Trivial to support new providers for many users
  - Difficult to manage several providers for many users
- Resource aggregation in overlay batch system
  - Unreliable to predict resources required for jobs
  - Efficient to integrate resources, then match jobs
- Yet it really works!



Slide by Max Fischer (KIT)

Innovative Digital Technologies for Exploring Universe and Matter

- Joint proposal by HEP, Physics of Hadrons and Nuclei, Astroparticle Physics
- Covered Topics:
  - Development of technologies to utilize heterogeneous computing resources (Integration of Opportunistic Resources, Caching Technologies, Workflow Management)
  - Application and testing of those technologies in heterogenous computing resources
  - Deep Learning - Achieving knowledge through profound data-driven methods (Hardware-related Data Processing, Object Reconstruction, Simulation, Quality of Network Predictions)
  - Event reconstruction: Cost- and energy efficient utilization of computing resources (Alternative Algorithms and Architectures like GPUs)
- Funded in the scope of Digital Agenda programme (BMBF)

Proposal of:

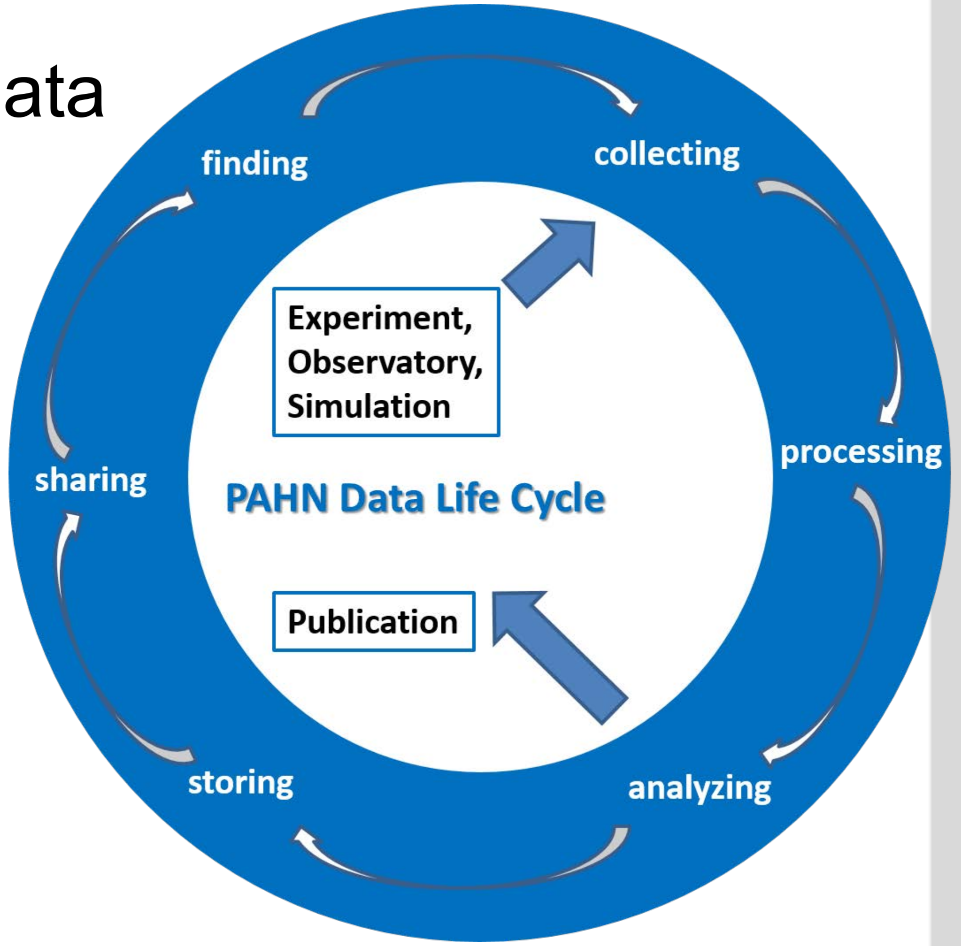
Associated Partners:

National Research Data Infrastructure (NFDI)

Particle, Astroparticle and Hadron & Nuclear Physics Accelerates the NFDI:

- Task areas:
  - Developing workflows and tools for data management
    - Tools to utilize heterogenous and HPC resources
    - …
  - FAIR Data Life Cycle Concepts and Open Data
  - Data analysis procedures and services
  - Real-time data analysis and selection

Proposal deadline in October!



Resources

- COBalD: http://cobald.readthedocs.io/
- ROCED: https://github.com/roced-scheduler/ROCED
- TARDIS: http://cobald-tardis.readthedocs.io/
- COBalD Simulation: https://git.scc.kit.edu/fq8360/cobalt_sim
- COBalD Demo: https://github.com/MaineKuehn/cobald_demo