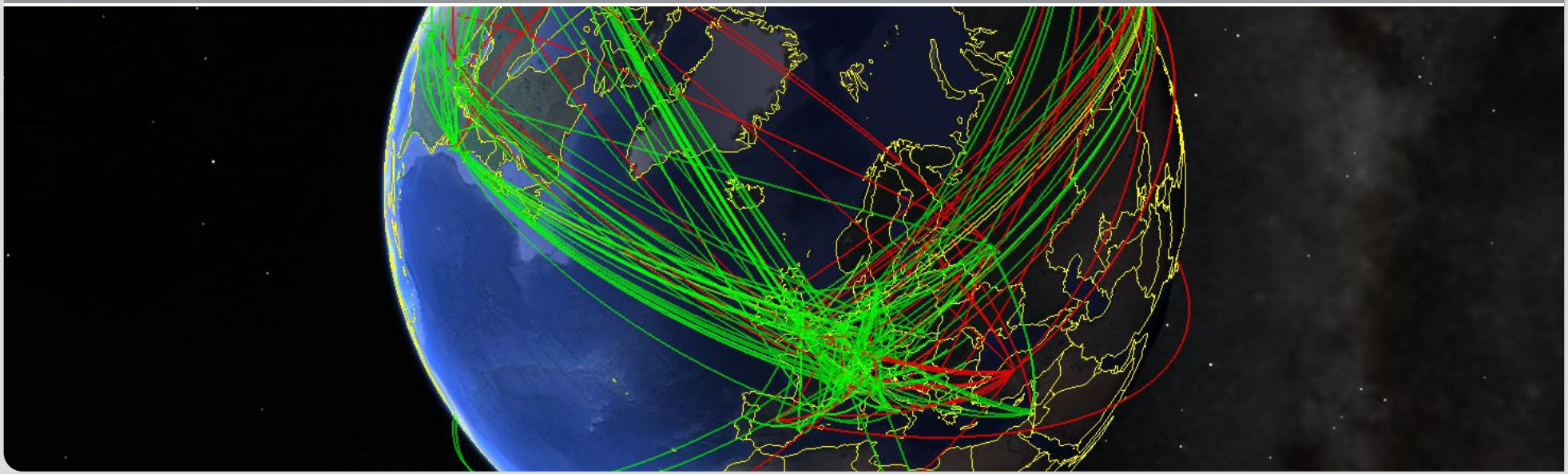
Karlsruhe Institute of Technology

ErUM Data: COBalD / TARDIS Workshop
COBalD / TARDIS @ KIT

Matthias Schnepf, René Caspart, R. Florian von Cube, Max Fischer, Eileen Kühn and Manuel Giffels

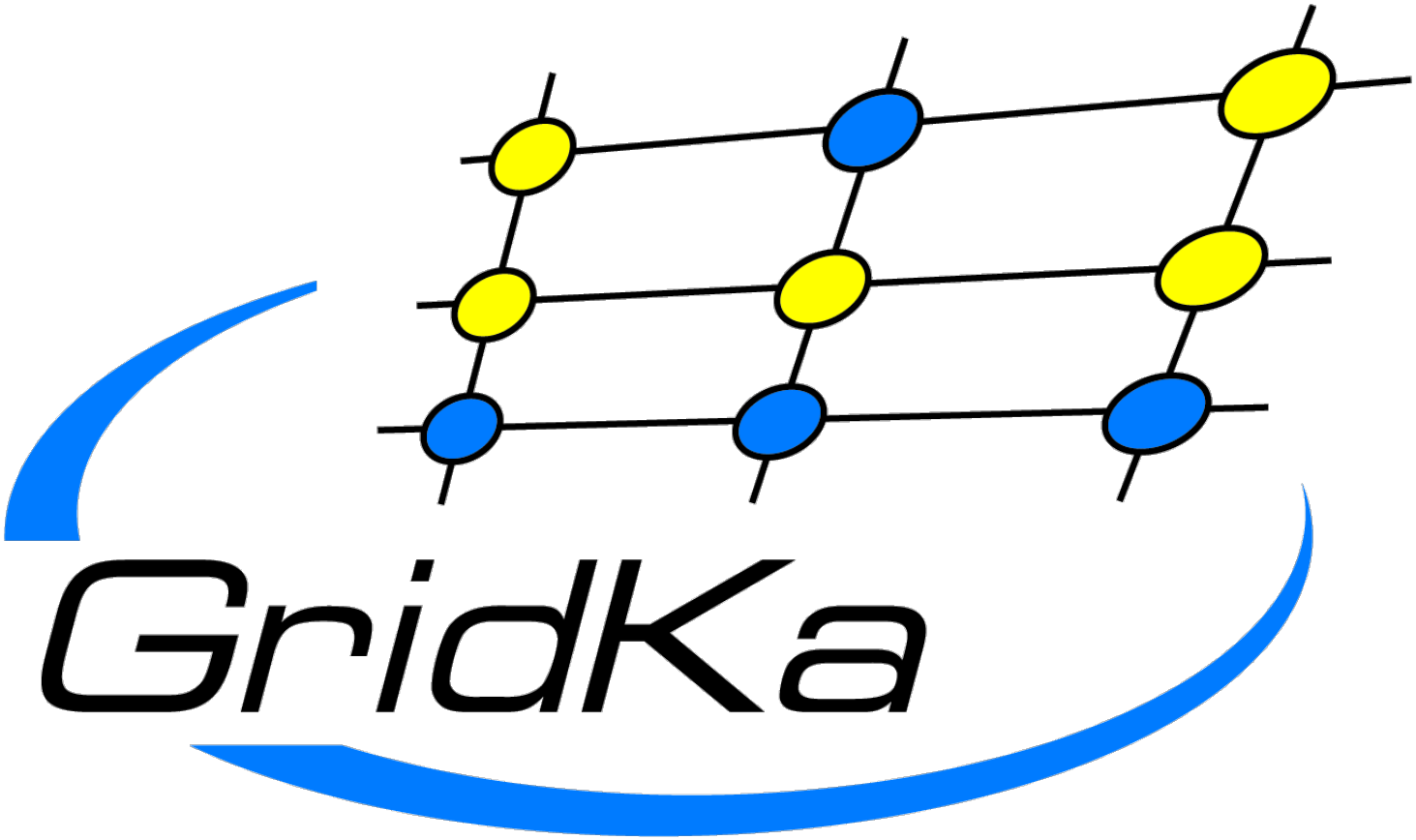
Karlsruhe Institute of Technology (KIT), SCC/ETP



Systems with TARDIS

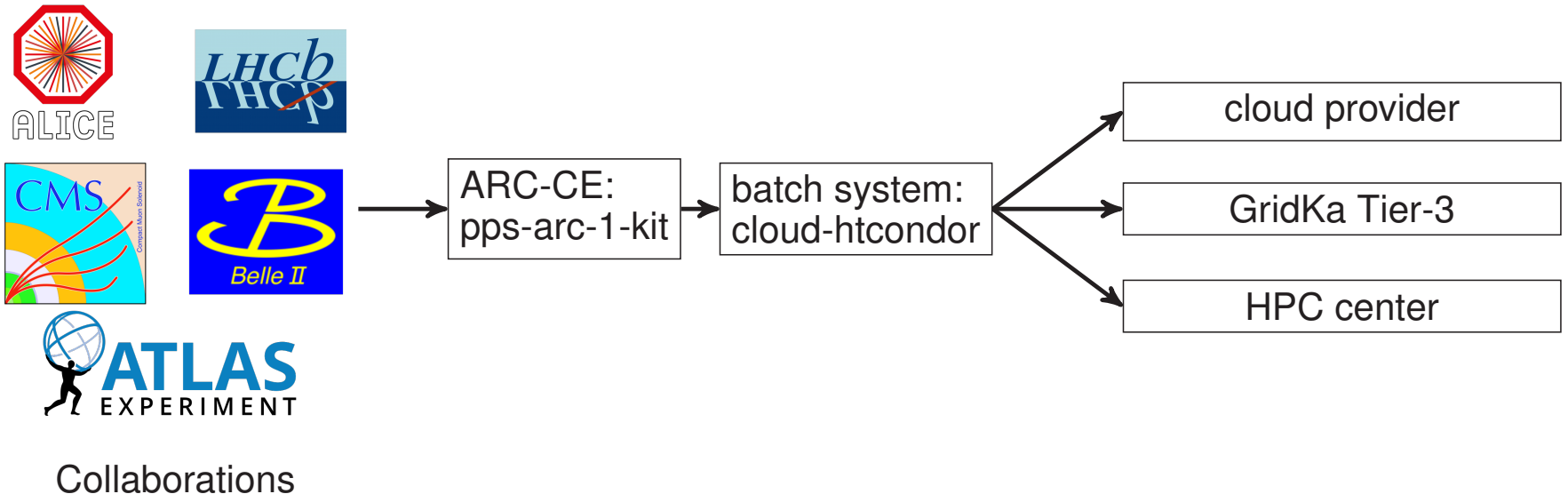
- GridKa cloud instance
  - cloud provider Exoscale: 300 CPU cores (during HNSciCloud project)
  - HPC Cluster ForHLR II: 400 CPU cores
  - high throughput Tier-3 cluster: 462 CPU cores
- ETP
  - HPC Cluster NEMO @ Freiburg: 8000 CPU cores
  - HPC Cluster ForHLR II | KIT: 260 CPU cores
  - GridKa School Instance: 800 CPU cores

TARDIS at GridKa



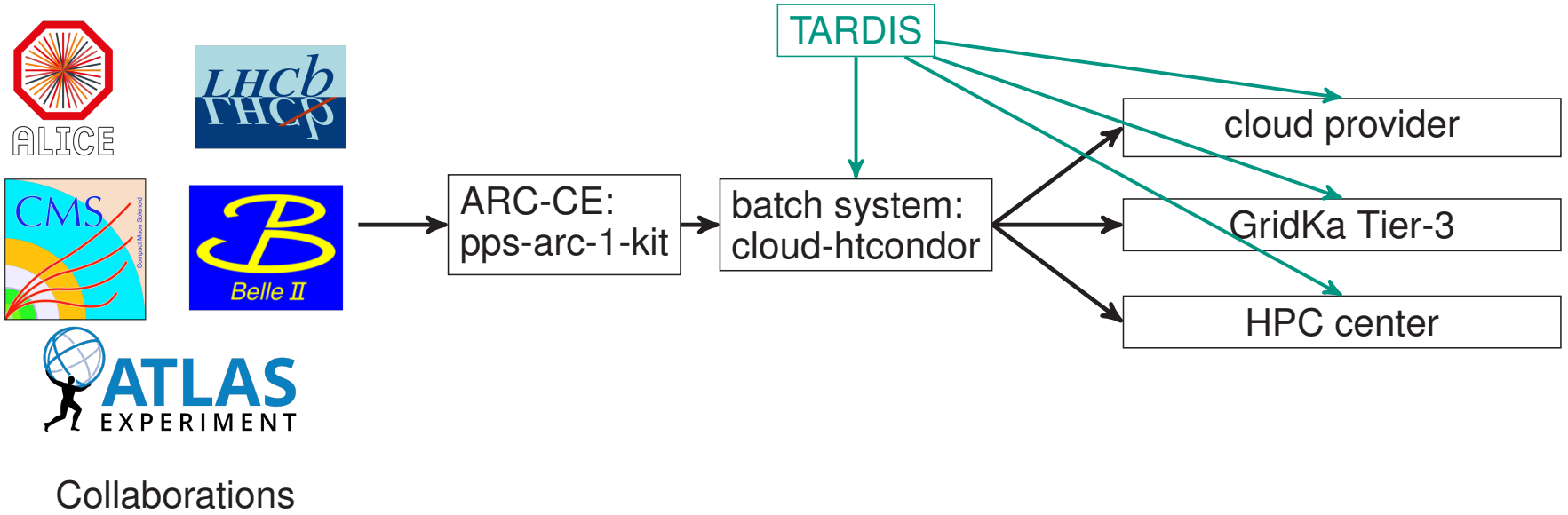
Current Setup

- entry point: `pps-arc-1-kit`
- HTCondor instance for opportunistic resources on `cloud-htcondor`
- resource manager TARDIS allocate and integrate resources via generalized pilots
  so-called drones (batch job, container, VM)



Current Setup

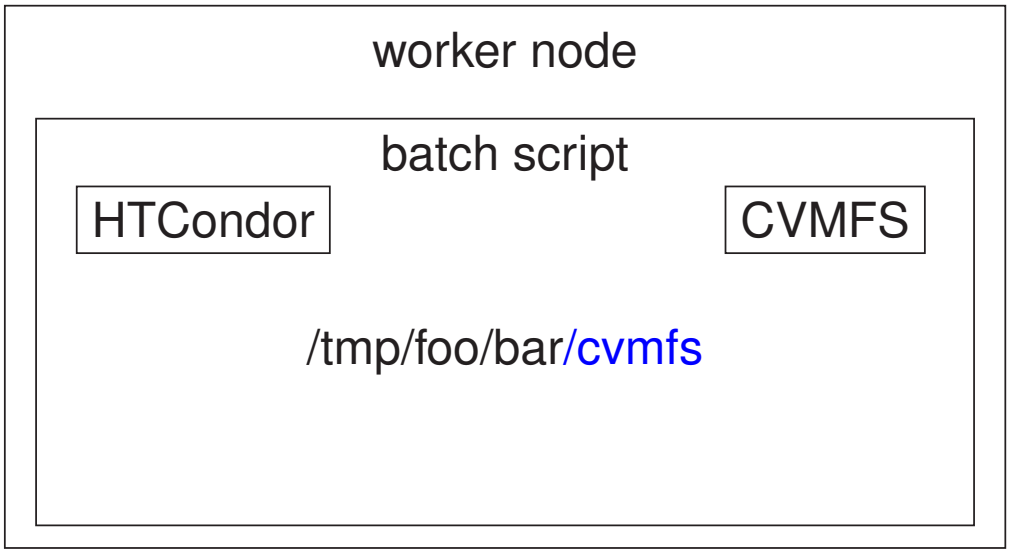
- entry point: `pps-arc-1-kit`
- HTCondor instance for opportunistic resources on `cloud-htcondor`
- resource manager TARDIS allocate and integrate resources via generalized pilots
  so-called drones (batch job, container, VM)



Collaborations

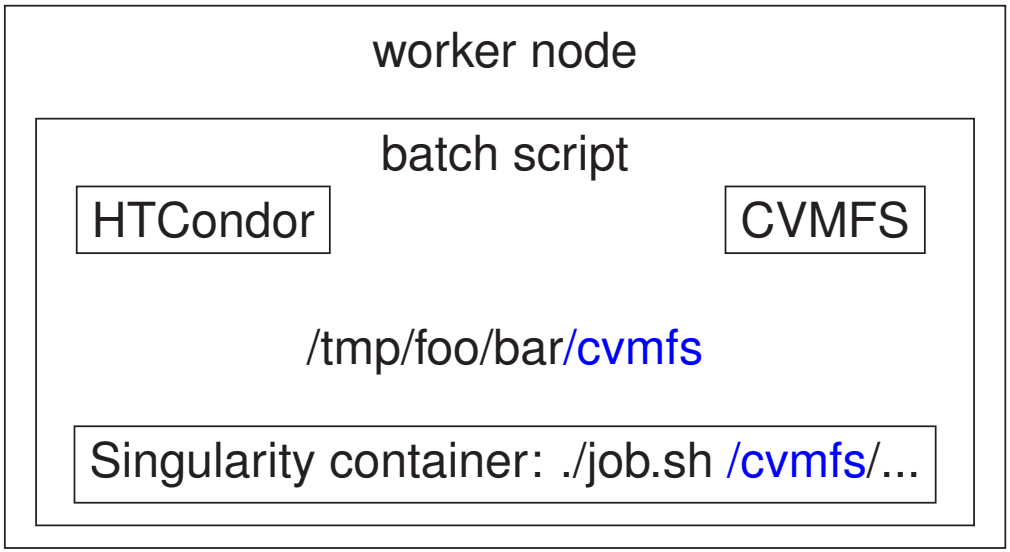
Drones at ForHLR II

- request resources via batch job
- worker node runs a batch script
  - HTCondor worker node
  - CVMFS
- HTCondor starts jobs in singularity container to provide CERN CentOS 7 environment



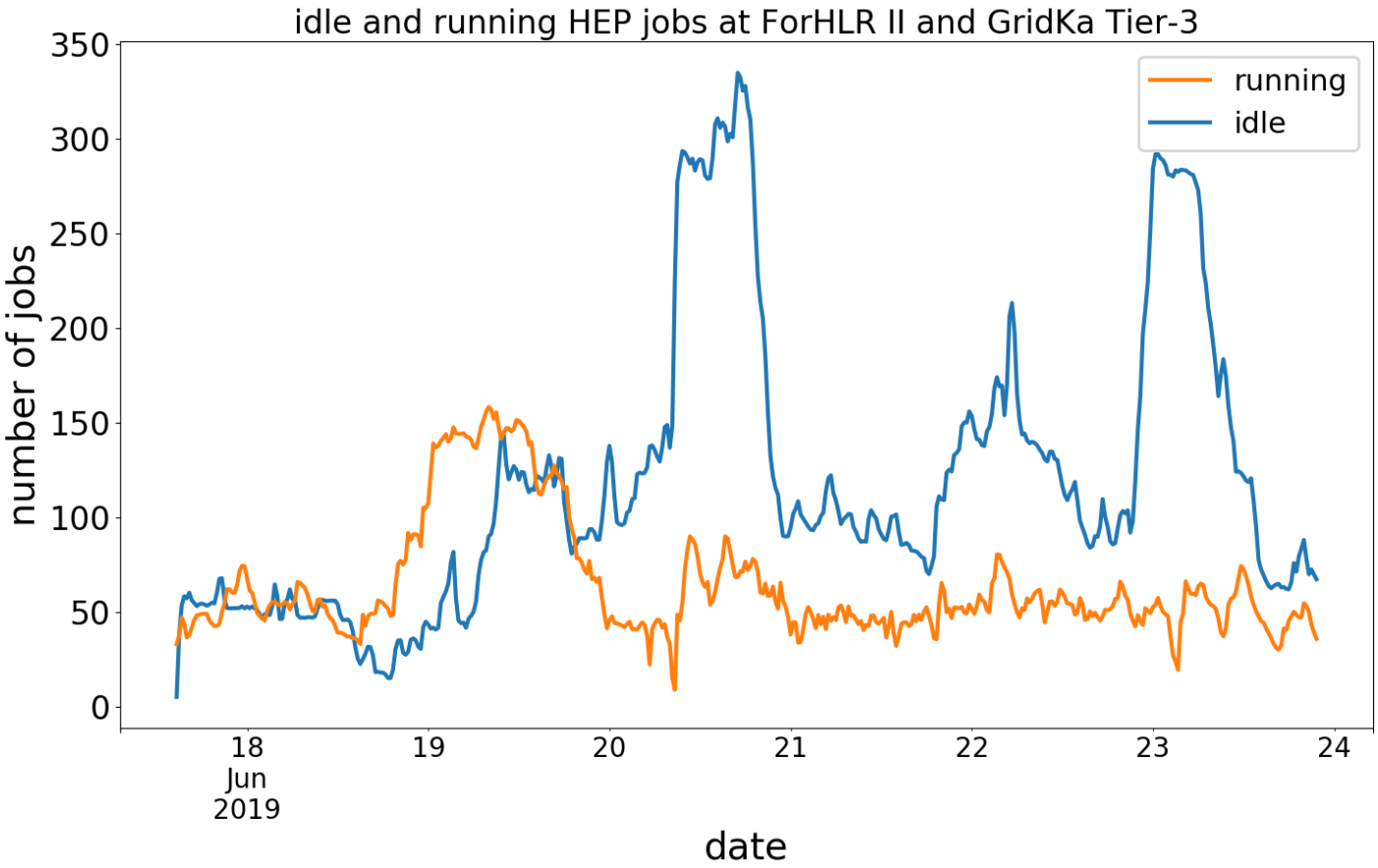
Drones at ForHLR II

- request resources via batch job
- worker node runs a batch script
  - HTCondor worker node
  - CVMFS
- HTCondor starts jobs in singularity container to provide CERN CentOS 7 environment



Current State

- integrate ForHLR II and GridKa Tier-3 resources
- management via TARDIS



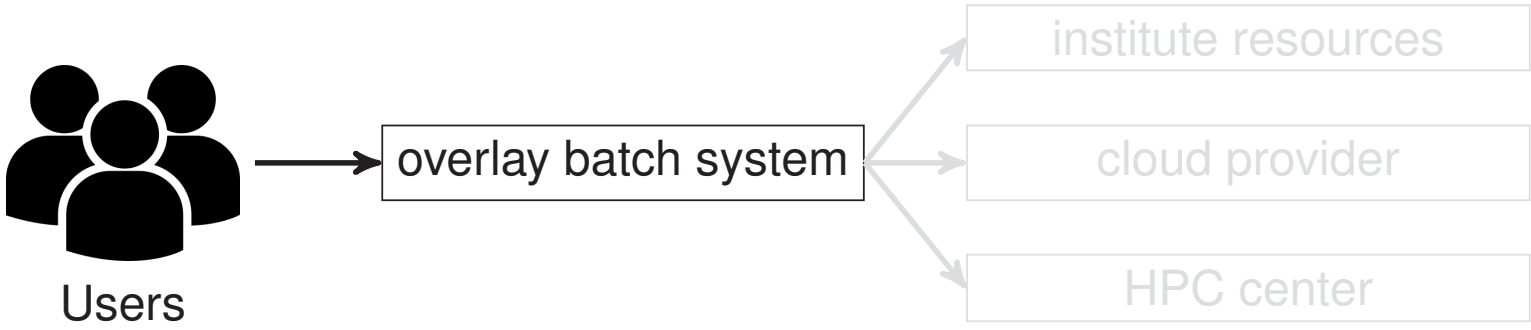
TARDIS at ETP

Karlsruhe Institute of Technology

Institute of Experimental Particle Physics

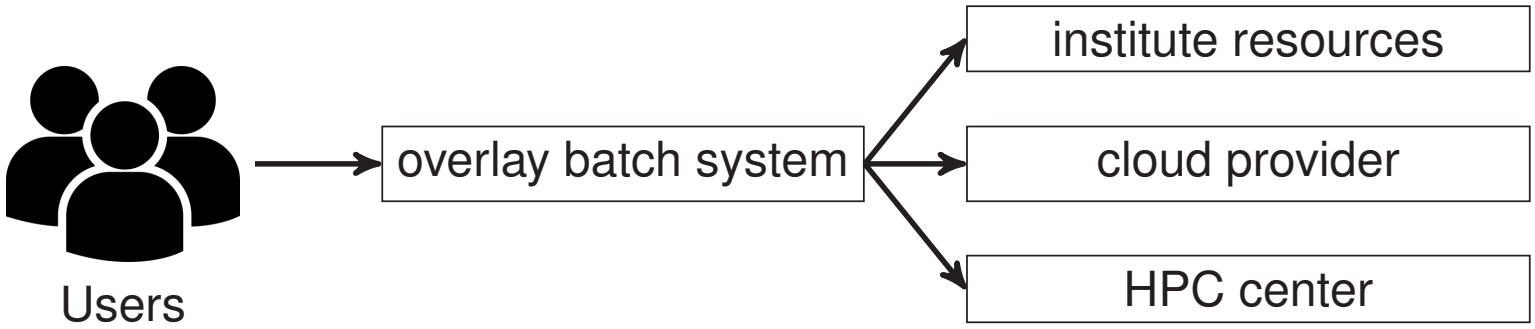
TARDIS at ETP

- schedule jobs to resources within an overlay batch system (OBS)
- users interact only with the OBS
- integrate resources from various resource providers on demand into a cloud



TARDIS at ETP

- schedule jobs to resources within an overlay batch system (OBS)
- users interact only with the OBS
- integrate resources from various resource providers on demand into a cloud



Resources at ETP

- static
  - usual worker nodes: 264 CPU cores
  - high throughput worker nodes: 84 CPU cores
  - desktop PCs (cycle stealing): 90 - 270 CPU cores
- included via flocking (connected HTCondor pools)
  - GridKa Tier-3: 462 CPU cores
- managed via TARDIS
  - OpenTelekomCloud: 500 CPU cores (during HNSciCloud Project)
  - HPC Cluster ForHLR II @ KIT: 260 CPU cores
  - GridKa School Instance: 800 CPU cores
  - HPC Cluster NEMO @ Freiburg: 8000 CPU cores

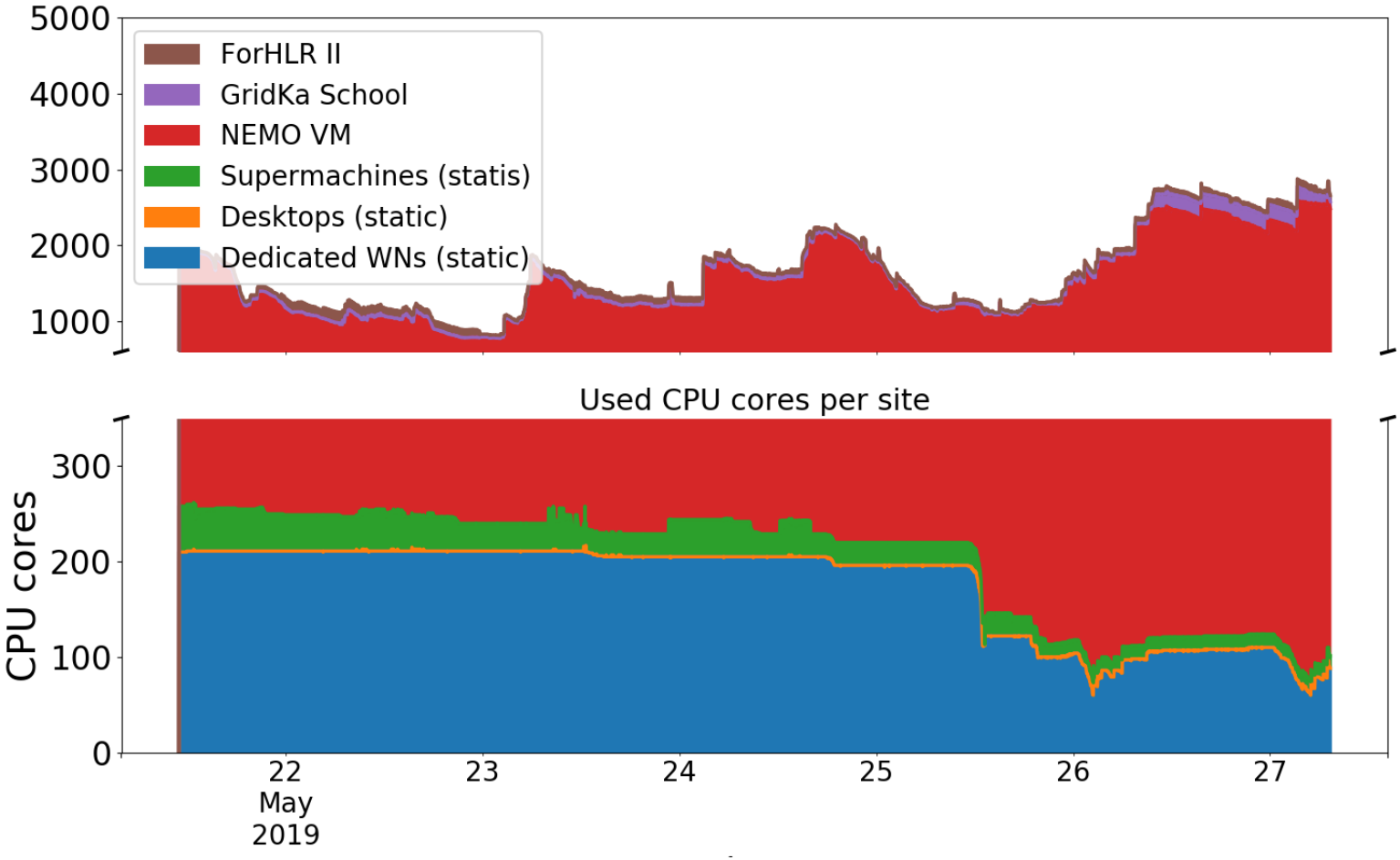


Resources at ETP

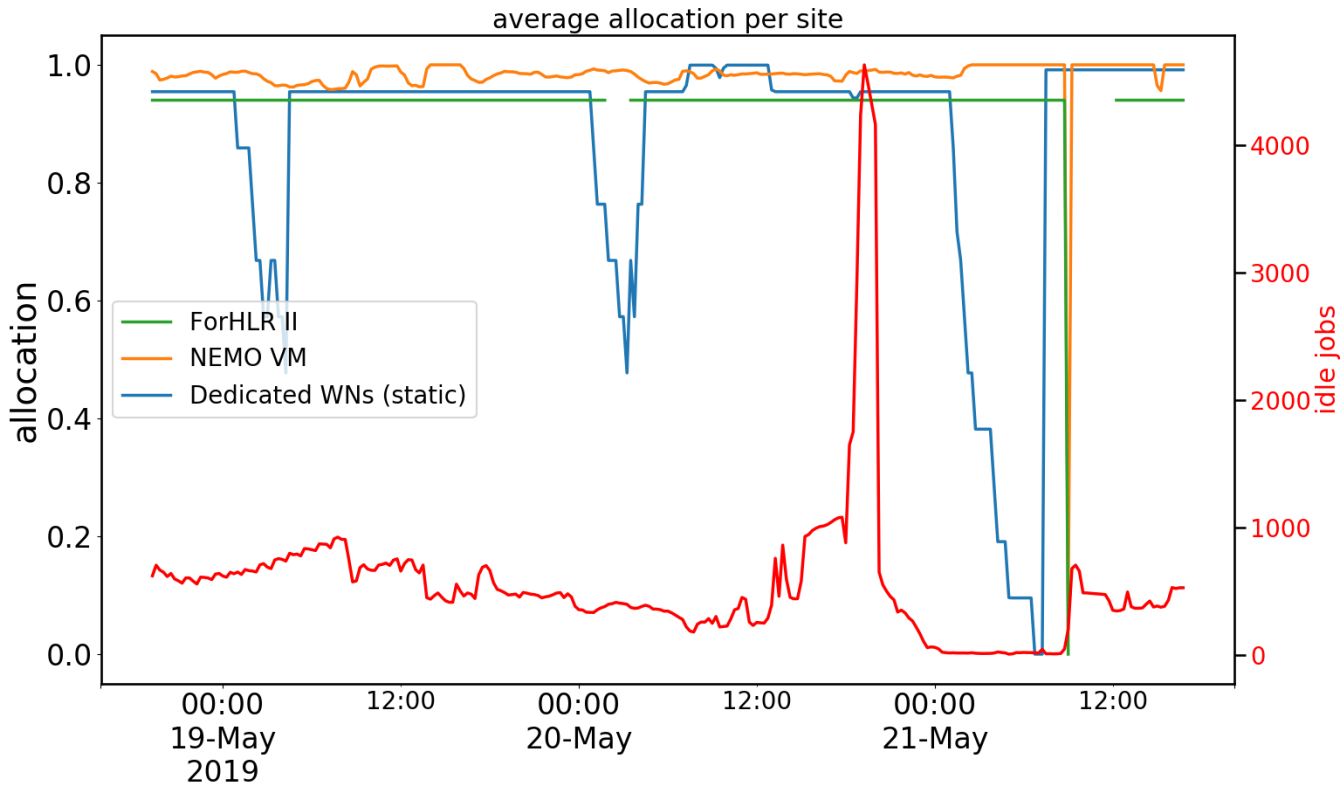
- static
  - usual worker nodes: 264 CPU cores
  - high throughput worker nodes: 84 CPU cores
  - desktop PCs (cycle stealing): 90 - 270 CPU cores
- included via flocking (connected HTCondor pools)
  - GridKa Tier-3: 462 CPU cores
- managed via TARDIS
  - OpenTelekomCloud: 500 CPU cores (during HNSciCloud Project)
  - HPC Cluster ForHLR II @ KIT: 260 CPU cores
  - GridKa School Instance: 800 CPU cores
  - HPC Cluster NEMO @ Freiburg: 8000 CPU cores




Resource Usage at ETP

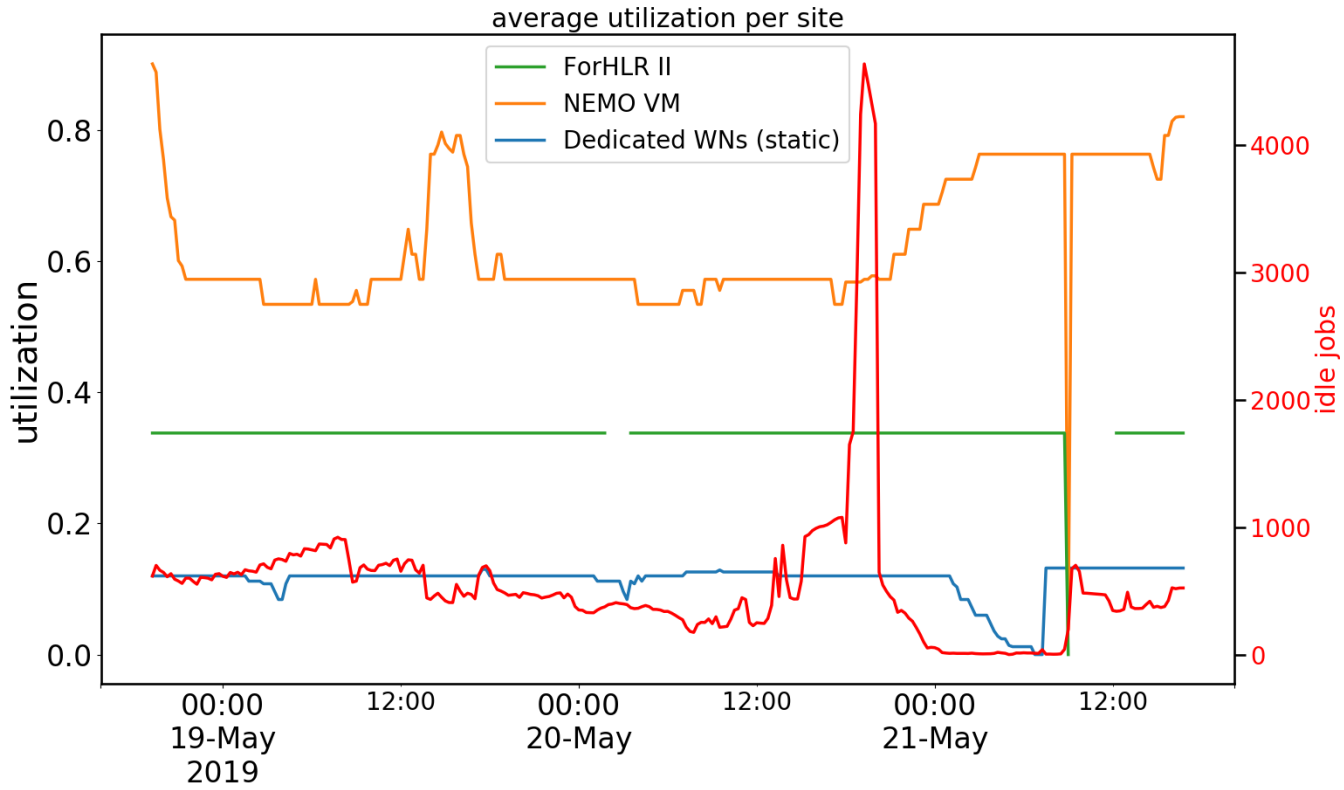


Allocation per Site at ETP



- demand and allocation (CPU, RAM, disk) are connected
- dynamic resources are more allocated than static resources with defragmentation

Utilization per Site at ETP



- low utilization (CPU, RAM, disk) $\Rightarrow$ a lot of resources are unused

Monitoring

- Telegraf service queries batch system information via python scripts
  github.com
  and send to InfluxDB service
- Grafana presents information from InfluxDB