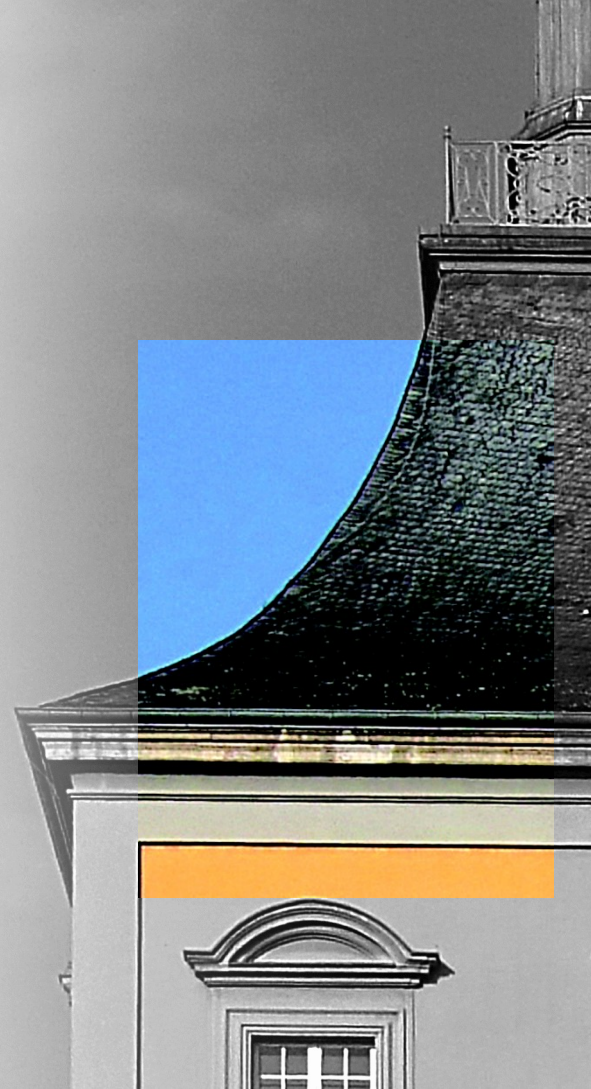
UNIVERSITÄT BONN

DYNAMIC INTEGRATION OF OPPORTUNISTIC COMPUTE RESOURCES

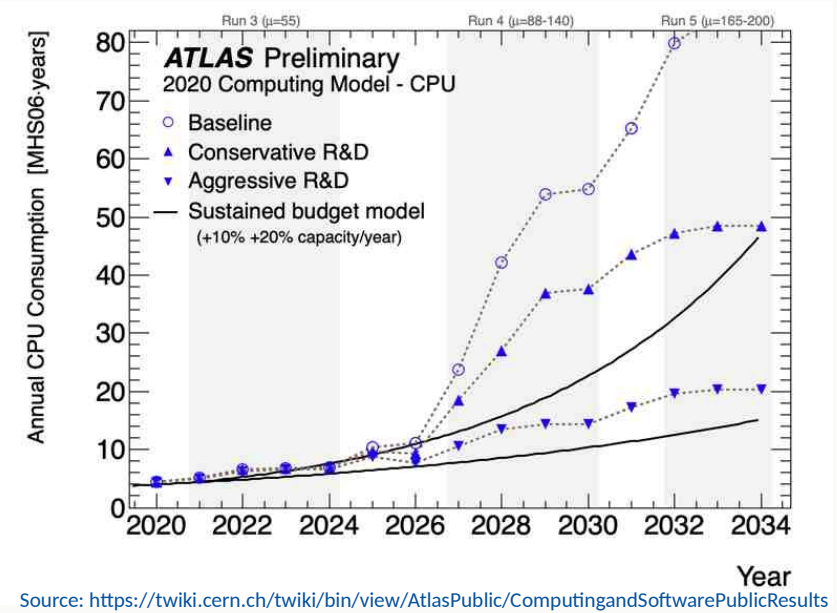
M. BÖHLER[1], R. CASPART[2], M. FISCHER[2],
O. FREYERMUTH[3], M. GIFFELS[2], S. KROBOTH[1], E. KÜHN[2],
M. SCHNEPF[2], F. VON CUBE[2], P. WIENEMANN[3]

[1]UNIVERSITY OF FREIBURG
[2]KARLSRUHE INSTITUTE OF TECHNOLOGY
[3]UNIVERSITY OF BONN



OPPORTUNISTIC RESOURCES

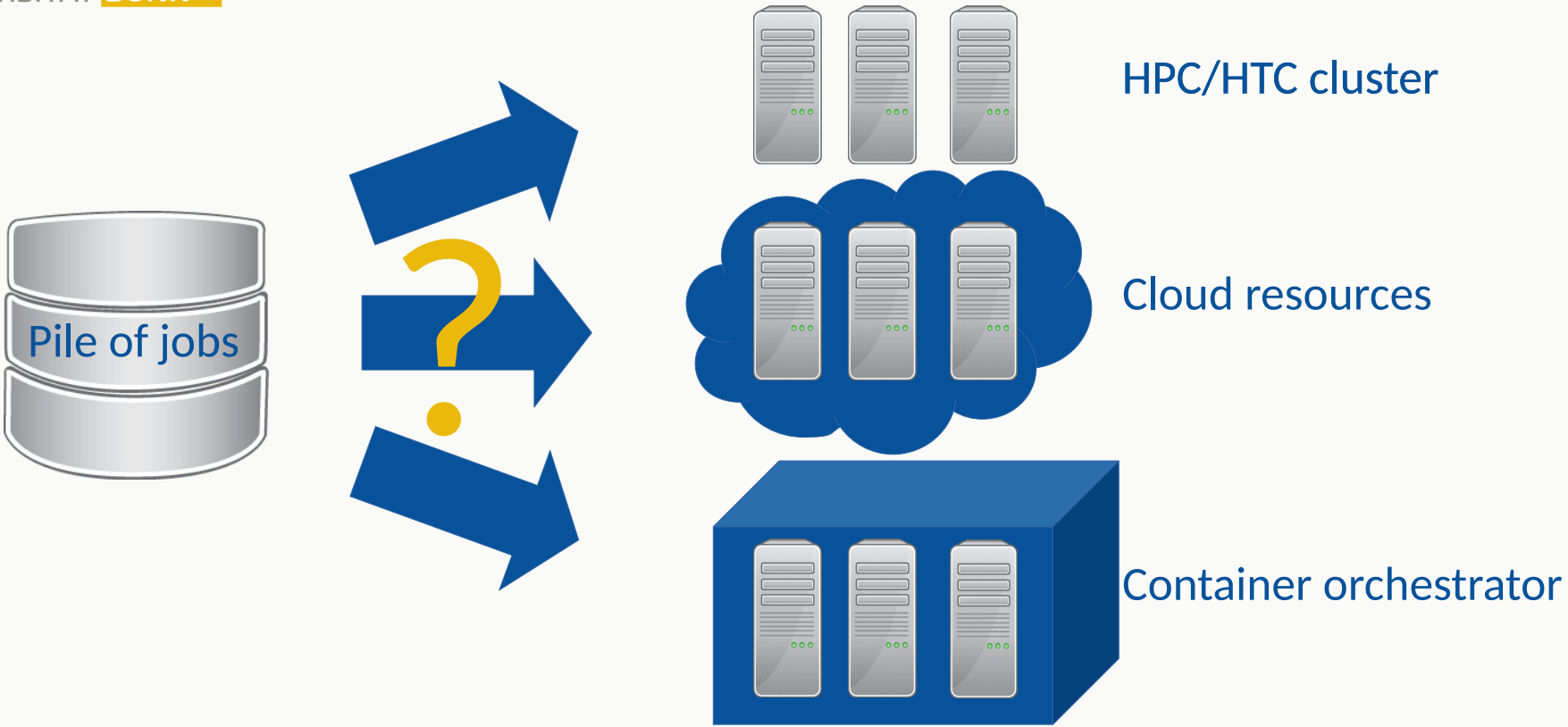
- Computing demands by HL-LHC are likely beyond what can be provided by dedicated resources (even if performance/price evolution is taken into account)
- Multiple approaches to address challenge
  - Temporarily use non-dedicated resources (HPC/HTC clusters, cloud resources, container orchestration suites, …)
  - Improve software/algorithms/ computing models
  - Investigate new technologies (quantum computing, …)
  - etc.



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

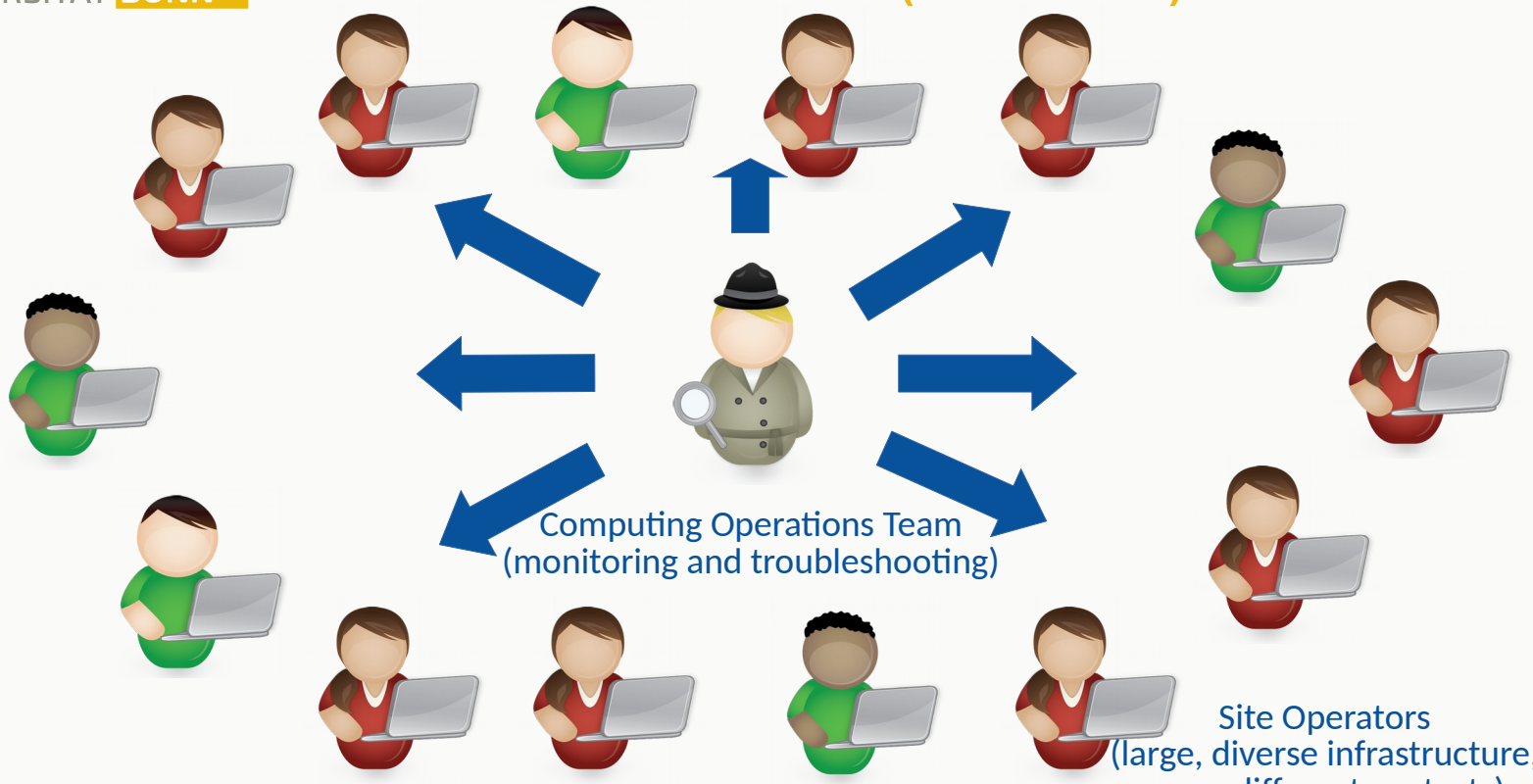
UNIVERSITÄT BONN

THE CHALLENGE



UNIVERSITÄT BONN

THE CHALLENGE (CONT'D)



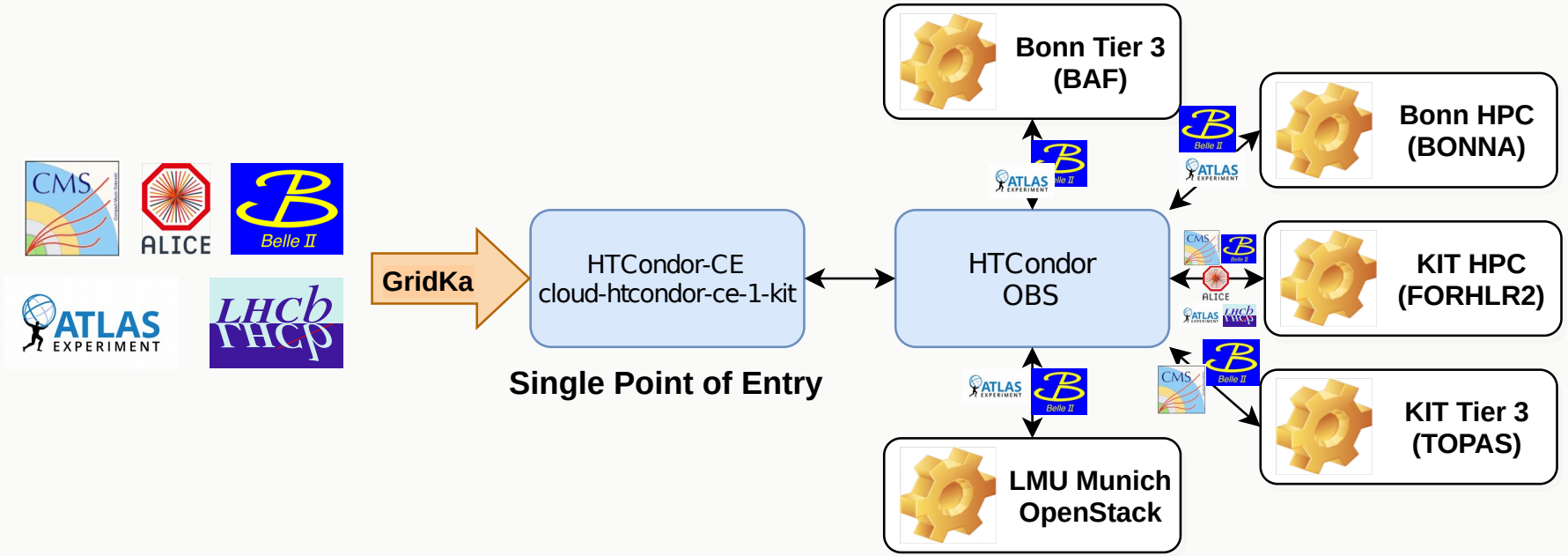
UNIVERSITÄT BONN

COBALD/TARDIS APPROACH

- Add abstraction layer between resource users and resource providers
- Hide heterogenous resources behind a single point of entry
- COBalD (COBalD – the Opportunistic Balancing Daemon)
  - Monitors usage of booked resources
  - Ramps up booked resources if they are well utilised
  - Reduces booked resources if they remain unused
- TARDIS (Transparent Adaptive Resource Dynamic Integration System)
  - Implements integration and management of resources provided by different systems (currently supported: OpenStack, CloudStack, Moab, Slurm, HTCondor, soon: Kubernetes) into overlay batch system (OBS)
- Available at `https://matterminers.github.io`,
  developed at KIT (M. Fischer, M. Giffels, E. Kühn, M. Schnepf, et al.)

COBALD/TARDIS APPROACH (CONT‘D)

COBalD/TARDIS interfaces overlay batch system (OBS) with local resource management systems



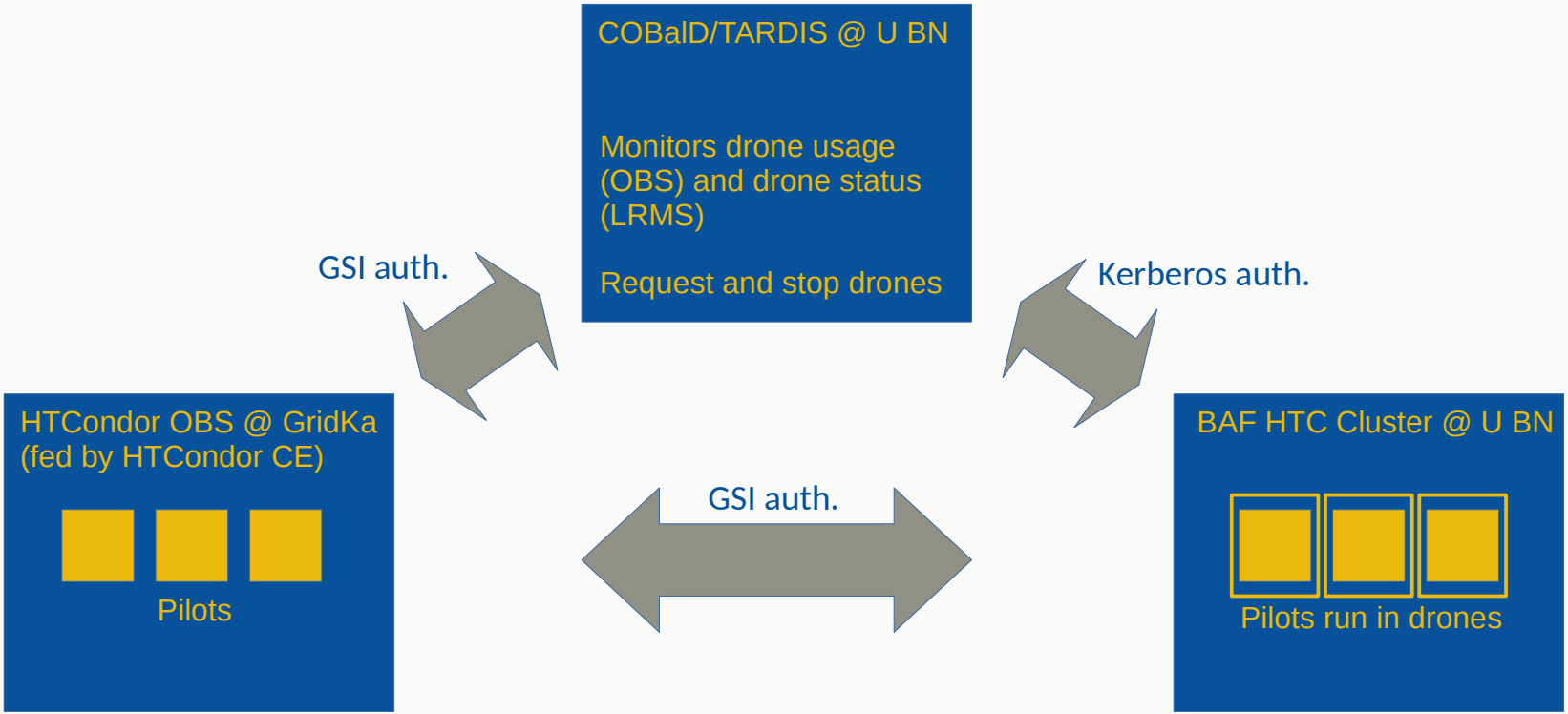
Source: M. Böhler et al., Transparent Integration of Opportunistic Resources into the WLCG Compute Infrastructure, submitted to CHEP 2021

UNIVERSITÄT BONN

PILOTS AND DRONES

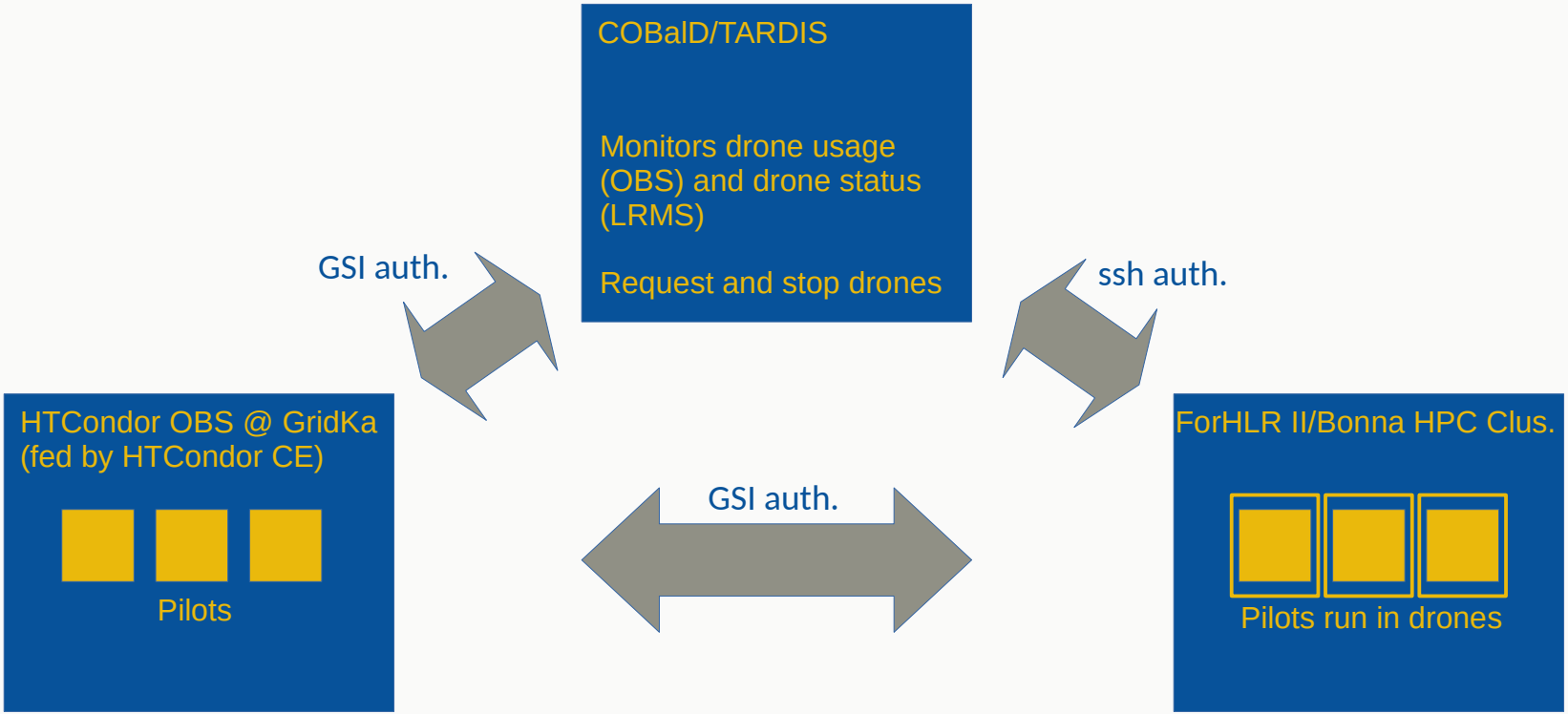
- Pilot = „Empty“ job skeleton which allocates resources (# cores, RAM, etc.)
  - E. g. ATLAS pilot: Also transfers input/output, pulls in payload when CPU becomes available (reduces latency)
- Drone = Generalized pilot started by TARDIS
  - Starts HTCondor execute daemons
  - Provides software environment
  - Integrates/removes resources into/from overlay batch system (OBS)

COBALD/TARDIS SETUP FOR BAF CLUSTER



UNIVERSITÄT BONN

SETUP FOR FORHLR II AND BONNA



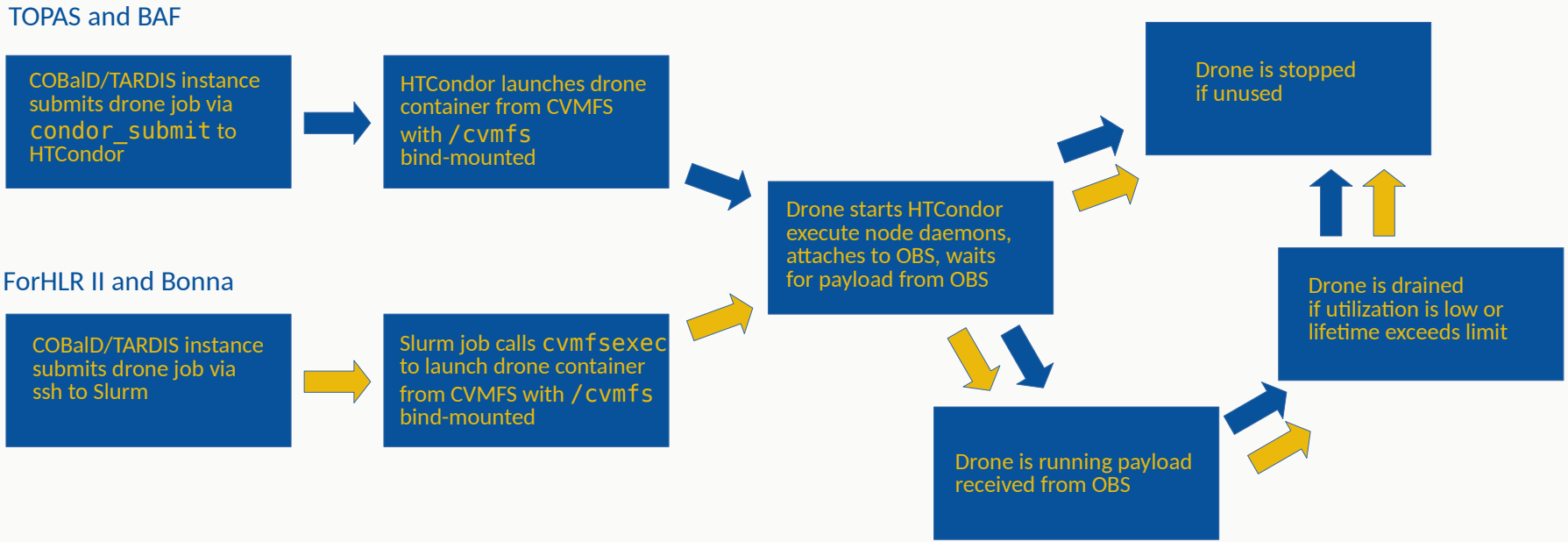
UNIVERSITÄT BONN

COBALD/TARDIS OPERATION

- Extremely lightweight compared to running grid compute element
- Only one instance per OBS needed. Still it turned out to be convenient to run an instance/service per resource provider (may run on the same host). May be run by OBS operator, resource provider or third party (depending on preferences).
- Configuration options to define
  - Metric to measure utilization (e. g. used/allocated CPU or memory)
  - When and how to start/stop new drones
  - Drone properties (e. g. # cores, memory, disk, lifetime [when draining should be triggered])
  - General settings like logging, etc.
- Shared port daemon needs to be reachable from drones to allow for draining
- Puppet module to automize setup: `https://github.com/unibonn/puppet-cobald`

UNIVERSITÄT BONN

DRONE LIFECYCLE EXAMPLES



UNIVERSITÄT BONN

CVMFSEXEC

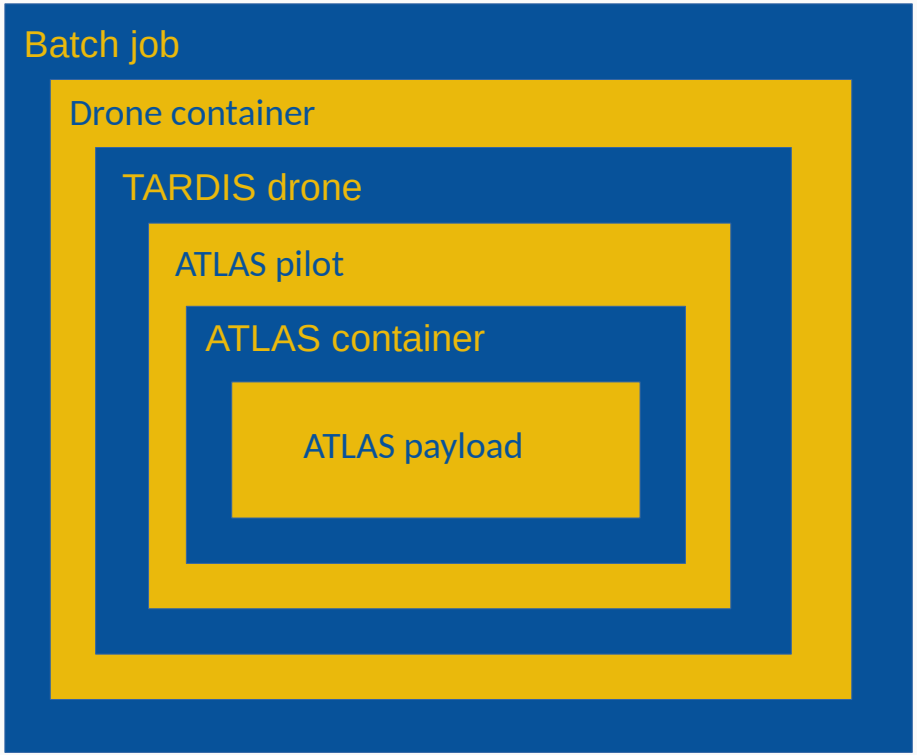
- `cvmfsexec`: Tool to mount CVMFS repositories as unprivileged user (written by Dave Dykstra, FNAL)
- Prerequisites*:
  - Either user namespaces and unprivileged namespace FUSE mounts (>= CentOS 7.8 or kernel >=4.18)
  - Or user namespaces and `fusermount` available
- Creates mount namespace to provide CVMFS repositories via `/cvmfs` mount point and runs given command in this namespace. If unprivileged namespace FUSE mounts are available, mounts are performed in additional process ID namespace such that kernel takes care of cleaning up mounts on process termination.

*There are more, less elegant options: See `https://github.com/cvmfs/cvmfsexec` for details.

UNIVERSITÄT BONN

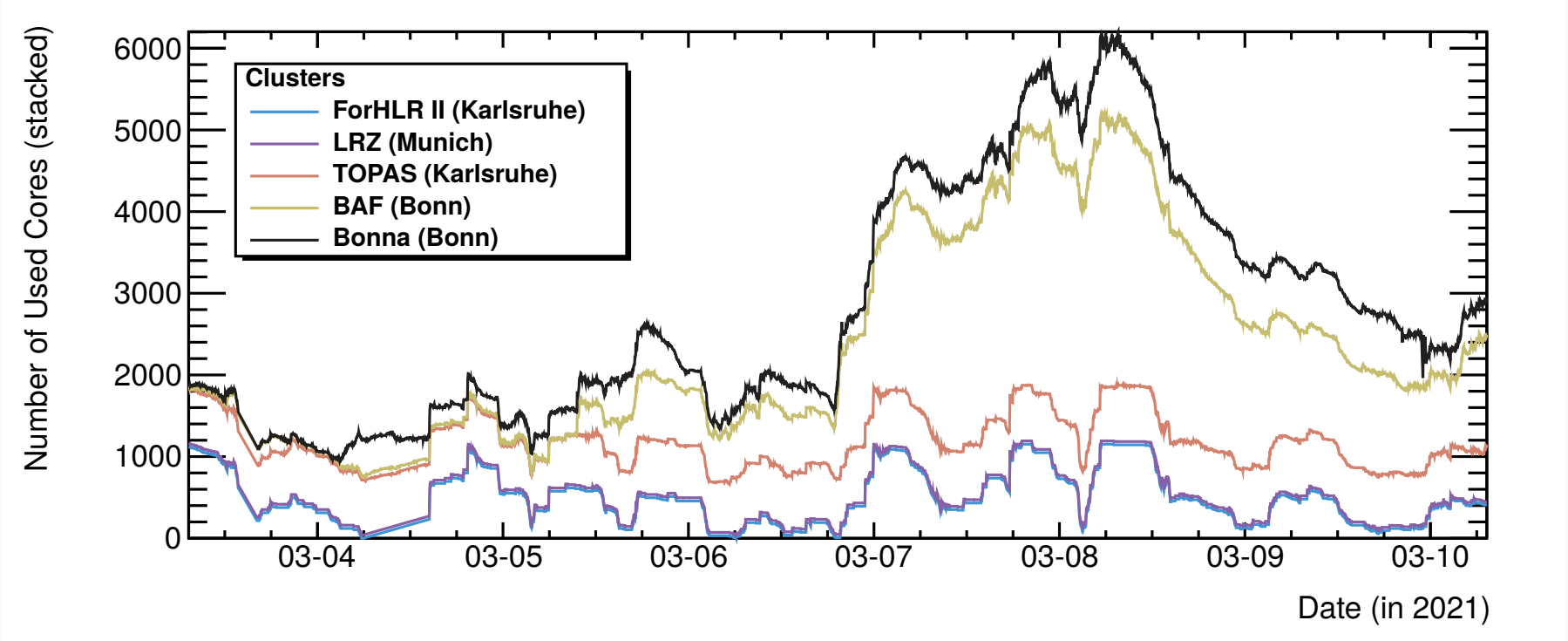
EXAMPLE JOB STRUCTURE

- Nested structure
- Containers to provide a common baseline environment for drones
- ATLAS containers to reduce site requirements (convenient for ATLAS)
- ATLAS pilots to improve throughput of ATLAS production system



UNIVERSITÄT BONN

OPPORTUNISTIC RESOURCE USAGE IN DE



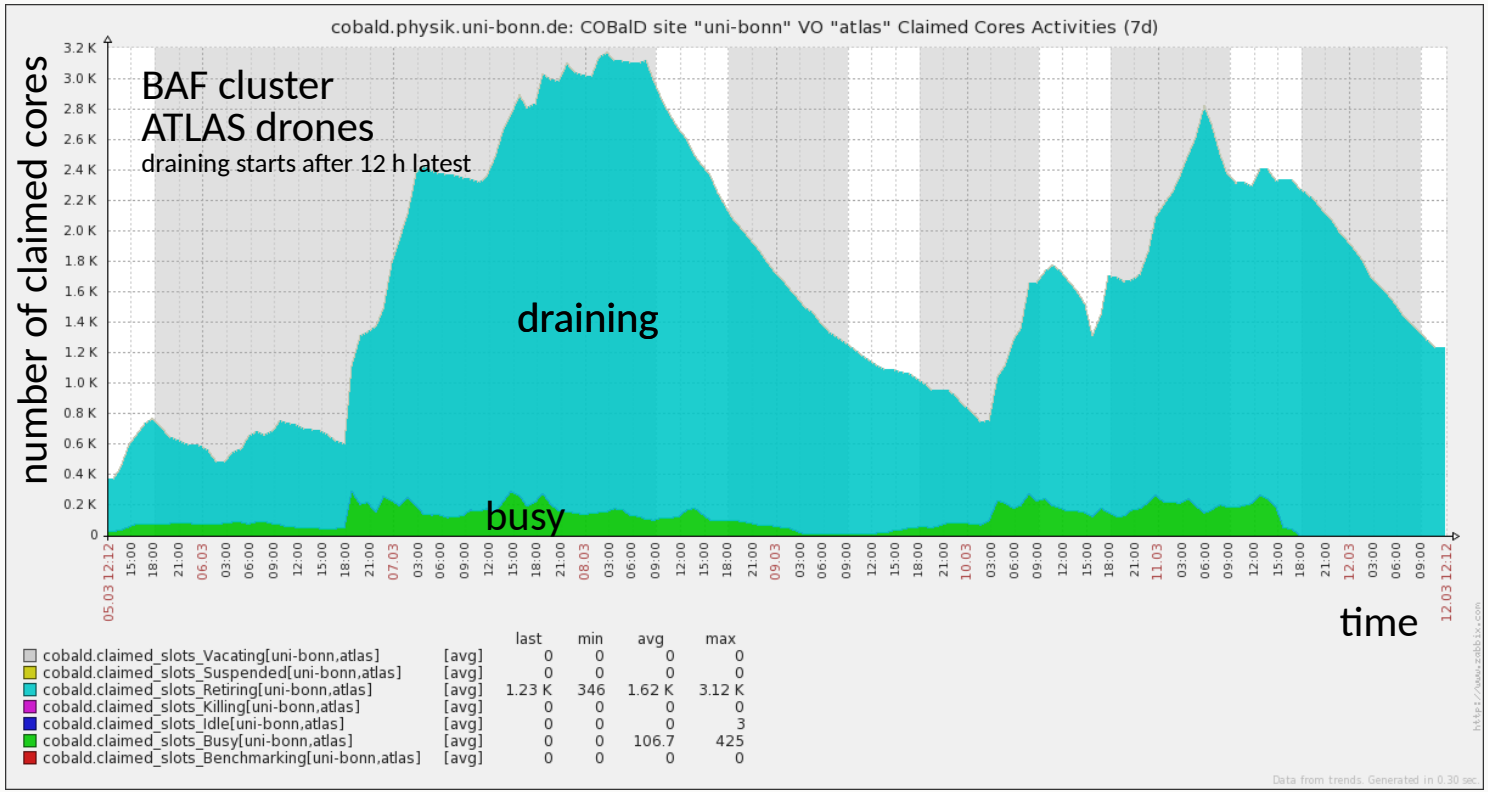
UNIVERSITÄT BONN

OPERATING MODELS

- Steady state
  - Provide a relatively small amount of resources continuously
- Backfilling with preemption (checkpoint, suspend or kill jobs)
  - Provide a substantial amount of resources if unused otherwise
  - Preempt jobs when slots are requested by local users
  - Either jobs still consume some types of resources (e. g. disk space, RAM) or payload is aborted
- Backfilling with graceful draining
  - Provide a substantial amount of resources if unused otherwise
  - Start draining early to ensure a continuous flow of slots becoming available for local users
  - Payload finishes successfully

UNIVERSITÄT BONN

„CONTINUOUS“ DRAINING



UNIVERSITÄT BONN

SUMMARY

- COBalD/TARDIS manages diverse opportunistic resources in a unifom way
- Concept successfully employed in production at a national scale
- Provided opportunistic resources provided by 5 sites in H2 2020 in the order of an average German university tier 2 centre
- Opportunistic resource usage harmonizes well with usage by local users
- For larger resource providers the network throughput during stage-in and stage-out phase of jobs can be sizable (at least for ATLAS and Belle jobs)
- More sites are welcome to join the club :-)