Transparent extension of the WLCG using the non-HEP side JURECA as an example

FIDIUM Collaboration Meeting 2022

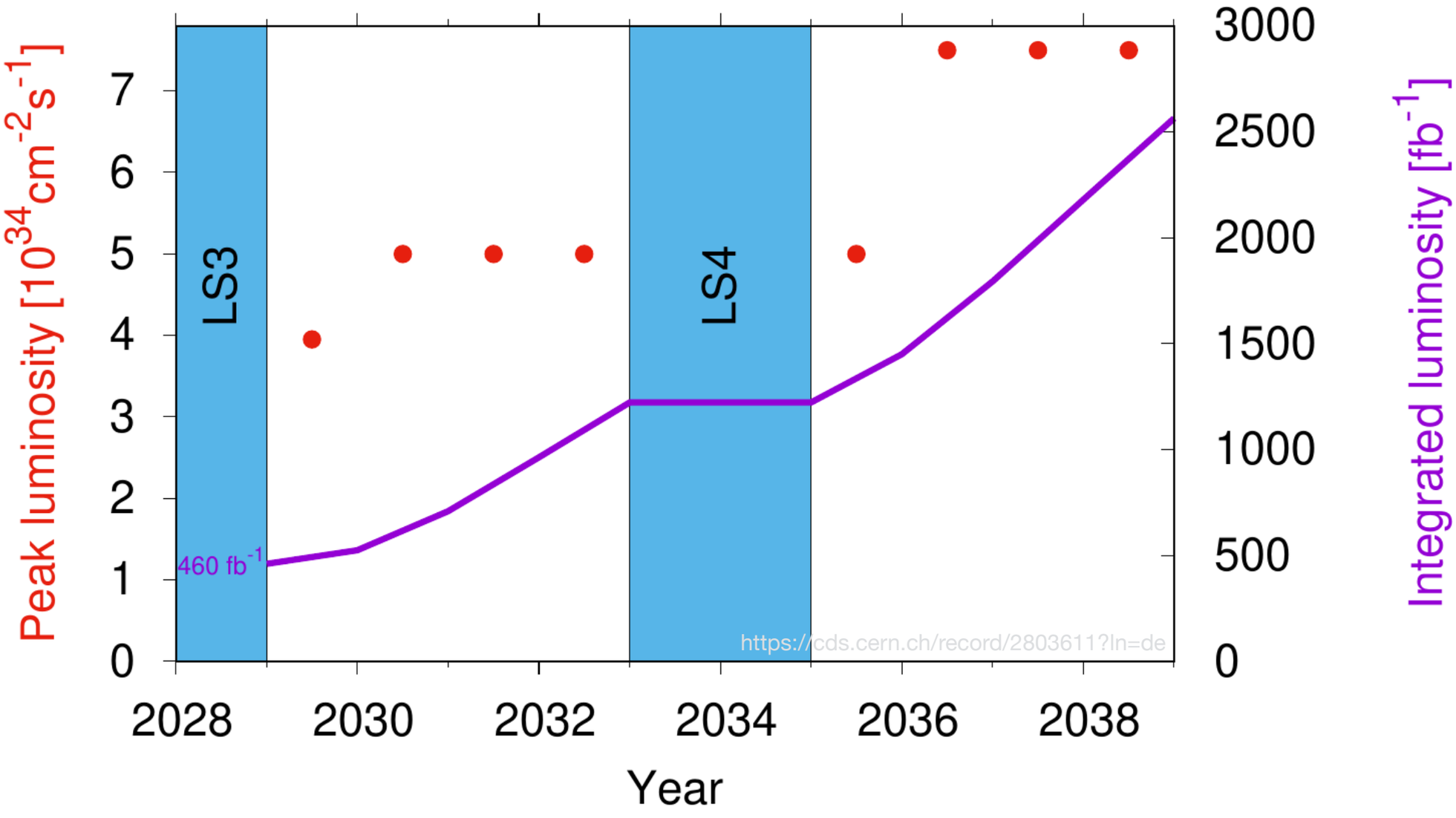
20.10.22

Manuel Giffels, Alexander Jung, Thomas Kreß, Thomas Madlener,
Andreas Nowack, Alexander Schmidt, Christoph Wissing

RWTH AACHEN UNIVERSITY

Motivation

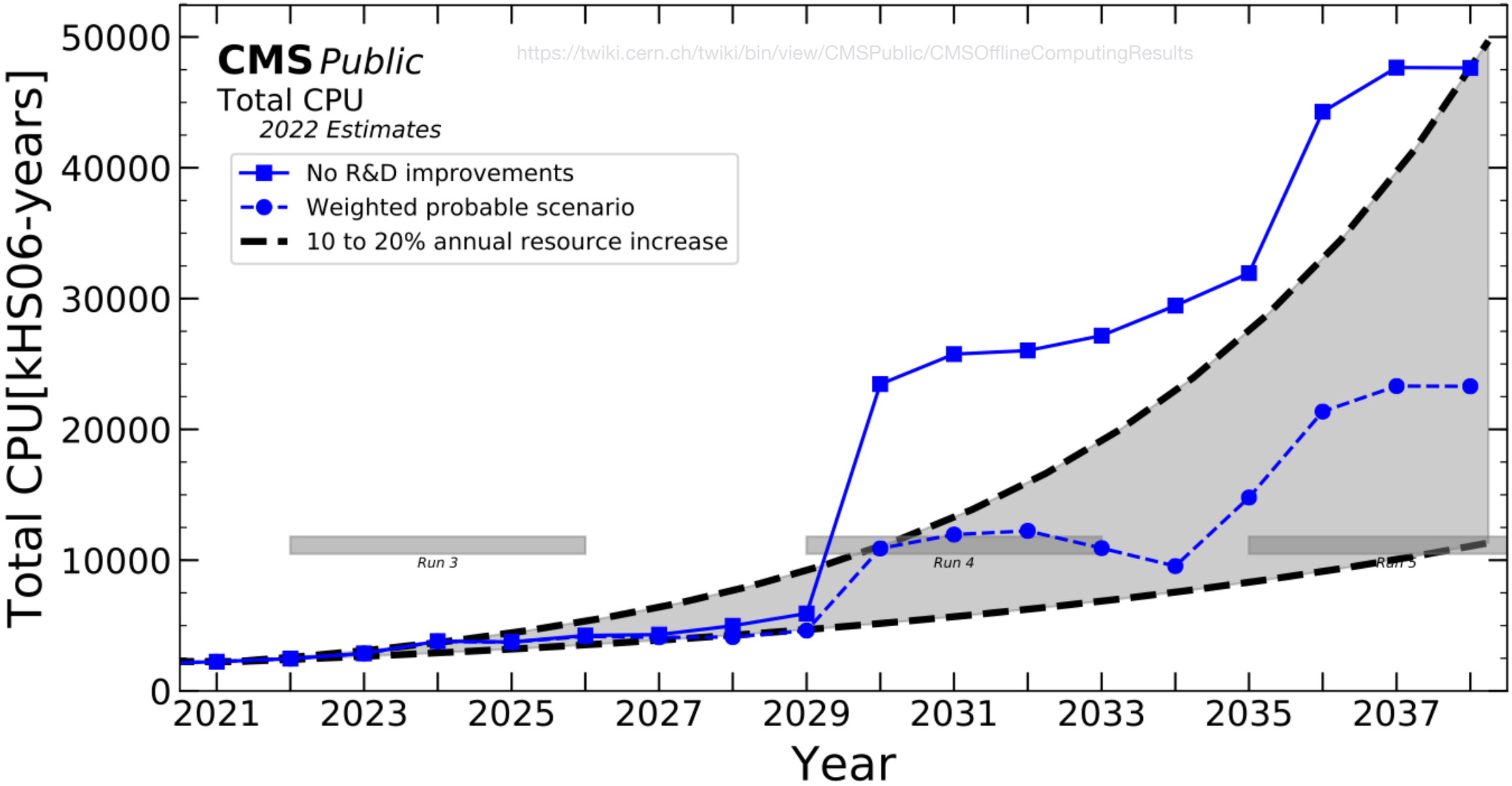
A. Jung - 20.10.22



- Estimated luminosity for Run 4 and beyond

- HL-LHC: challenges in the areas of data acquisition, processing, simulation, and analysis

Motivation



- Grid Resources may no longer be sufficient as of Run 4

- We need additional (opportunistic) resources

CMS's expectations for opportunistic resources

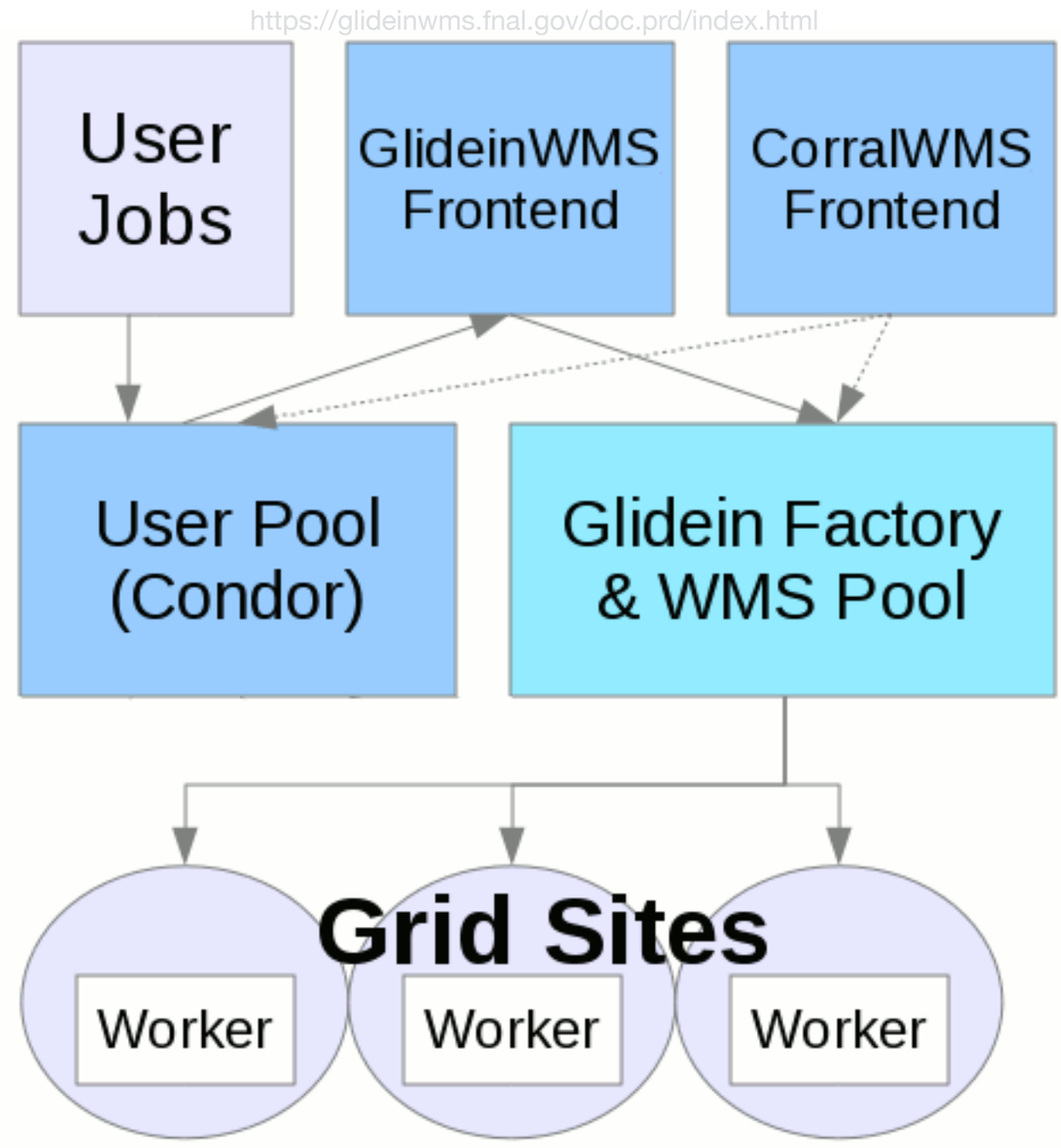
- Transparent for end users (integration into global CMS collector)
- Dynamic, without knowing/predicting future demand
- Expected WLCG environment is provided

Expected setup by grid jobs

- Connection to external site to store output
- CVMFS present to grant access to CMSSW
- Access to conditions database
- Potentially access to input data

Glidein/Drone/Pilot concept

- Placeholder jobs bind resources
- Environment is provided by a Singularity container
- Integration into overlaying batch system (OBS)
- OBS: central management of the actual jobs

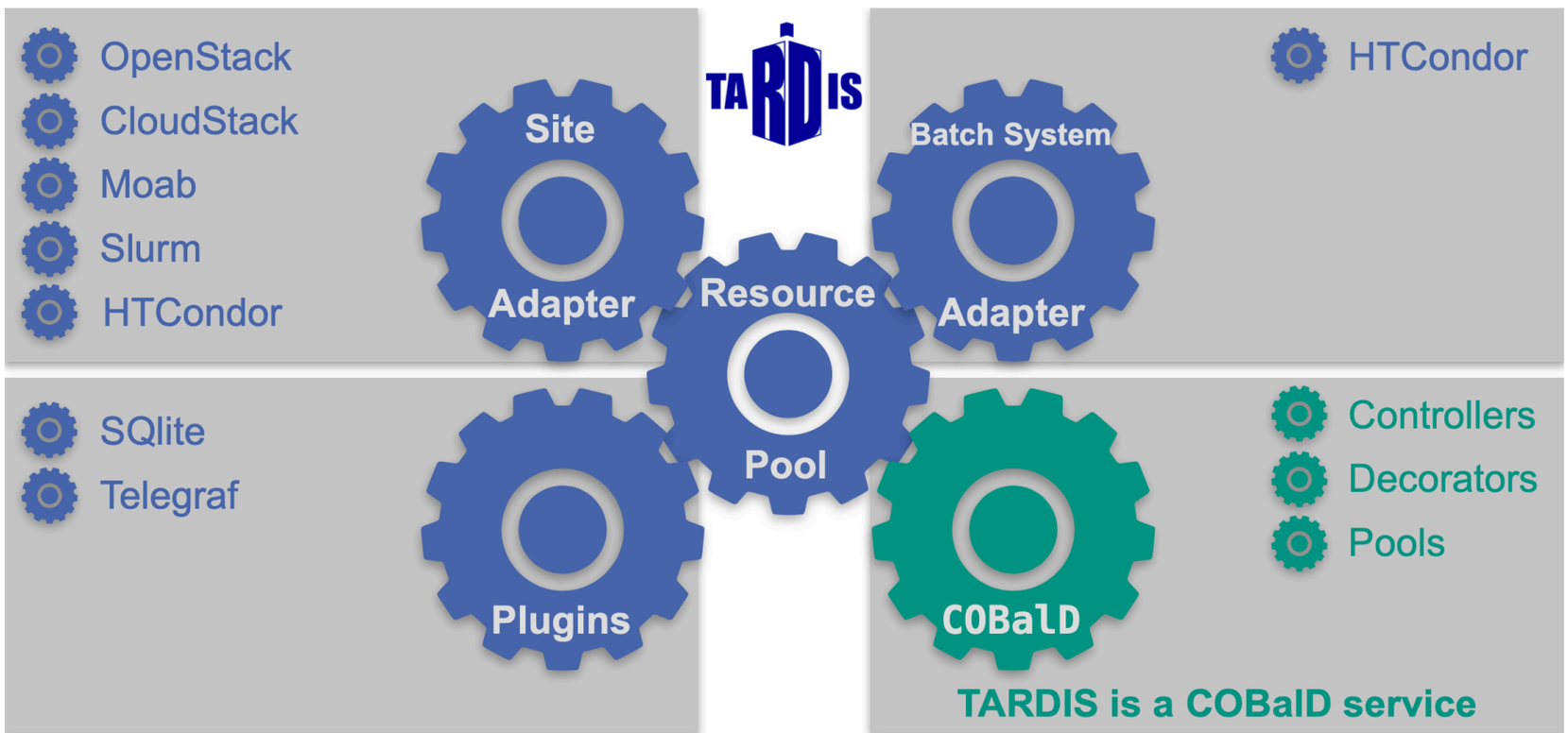


COBalD/TARDIS

- Automate the startup of glideins (manual still possible)

- COBald = Opportunistic Balancing Daemon
  - Monitors allocation and utilization of resources (e.g. RAM, CPUs…)
  - Forms abstract metrics
  - Requests more glideins or lets existing ones expire, based on metrics

- TARDIS = Transparent Adaptive Resource Dynamic Integration System
  - Interface between jobs, external resources, batch systems and COBalD



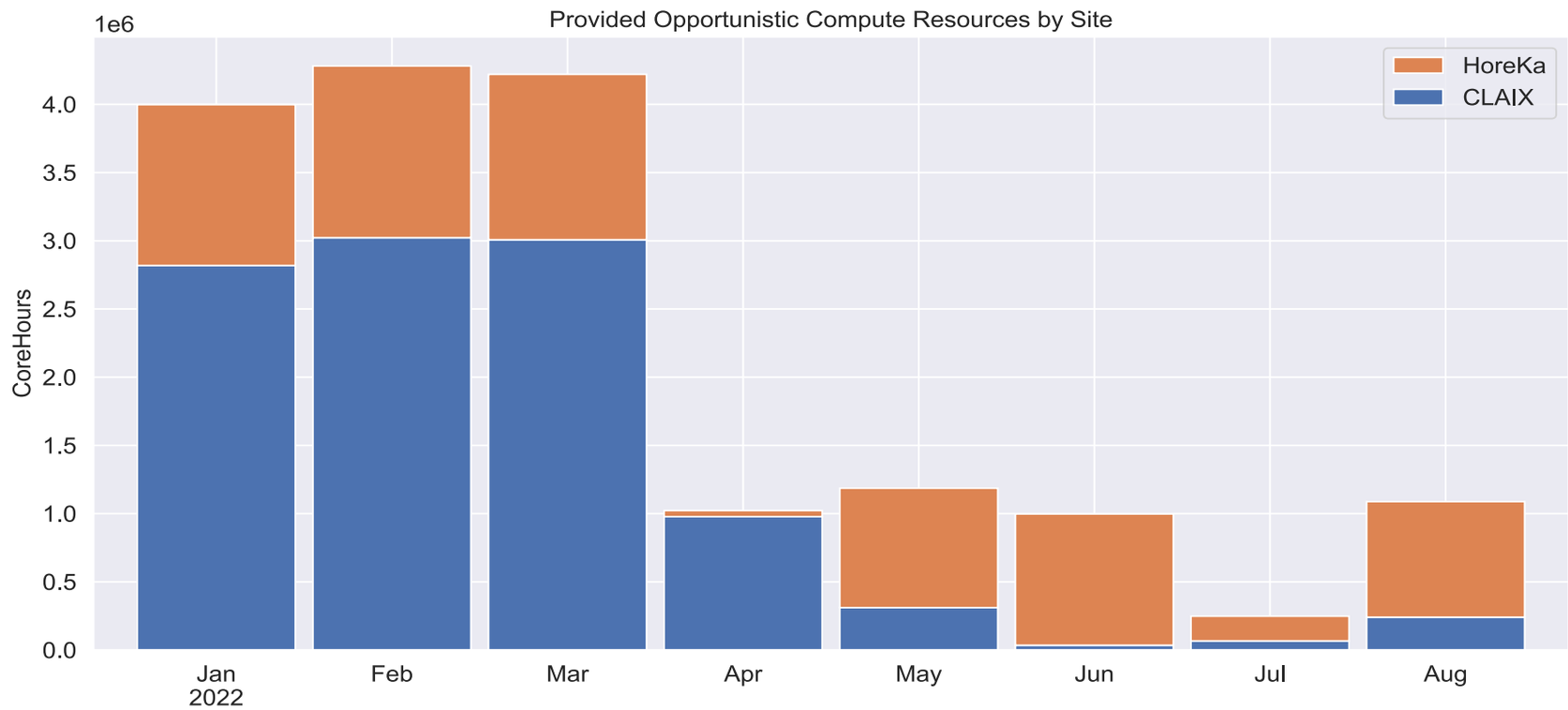
https://doi.org/10.1051/epjconf/202024507038

COBaID/TARDIS

- Already in use to extend German Tier 1 and Tier 2 to local HPC Centres

- KIT's Tier 1 extended to HoreKa
- RWTH's Tier 2 extended to CLAIX

} ~18 million corehours provided to CMS this year so far



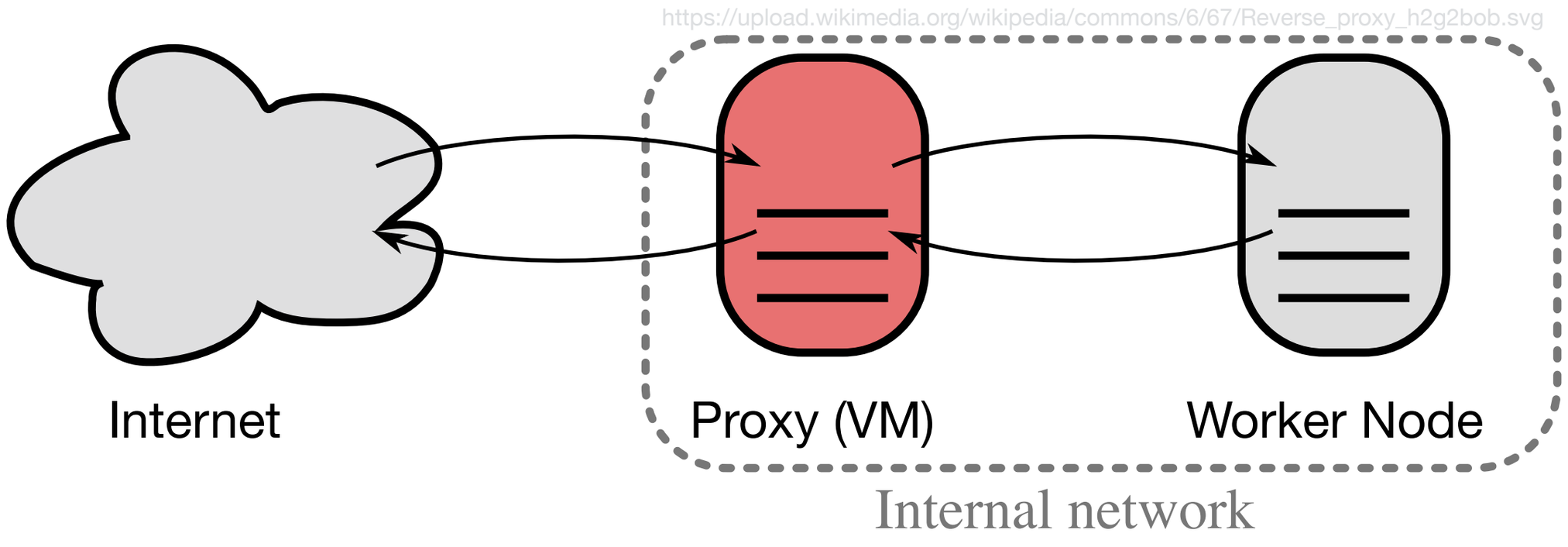
- Work in progress: Making the HPC Centre JURECA located at Forschungszentrum Jülich (FZJ) available

- 3 major challenges arise

Situation in Jülich (JURECA)

Strict firewall (almost ✅)

- Worker nodes have no connection to the outside world
- Worker nodes route all traffic via a virtual machine (VM) in the HDF Cloud
- Bandwidth Worker Nodes <-> VM: ~10Gbit/s
- Usage of ssh and proxychains from Worker Node (client) to VM (host)

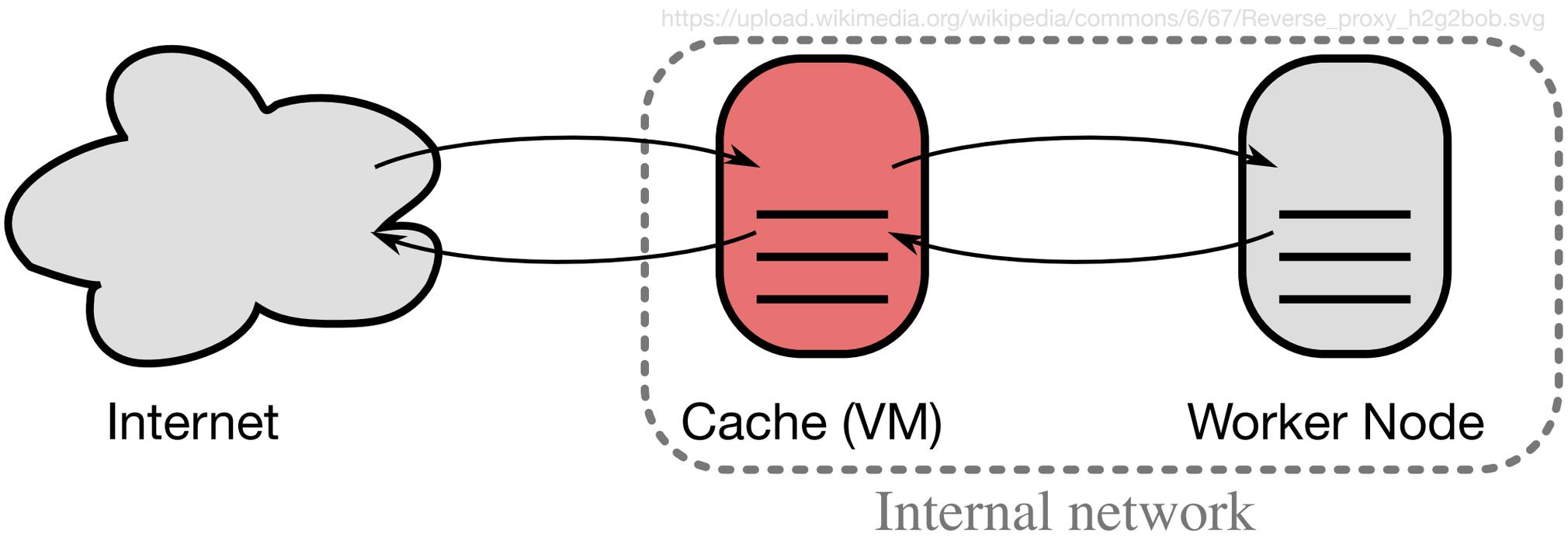


Environment (✅)

- CVMFS not present, but provided by prepared cvmfsexec tarball
- User jobs run inside singularity container
- Bind-mount unpacked tarball to /cvmfs/ inside container
- Bind-mount additional needed stuff inside container

Caching (✅)

- CVMFS traffic handled by Squid Proxy running on the VM
- Each glidein sets up its own cache on the worker node and can serve up to 256 payloads with it
- Caching of conditions database by Squid Proxy on VM
- Once cache is established: little traffic to the outside, much internal traffic



Situation in Jülich (JURECA)

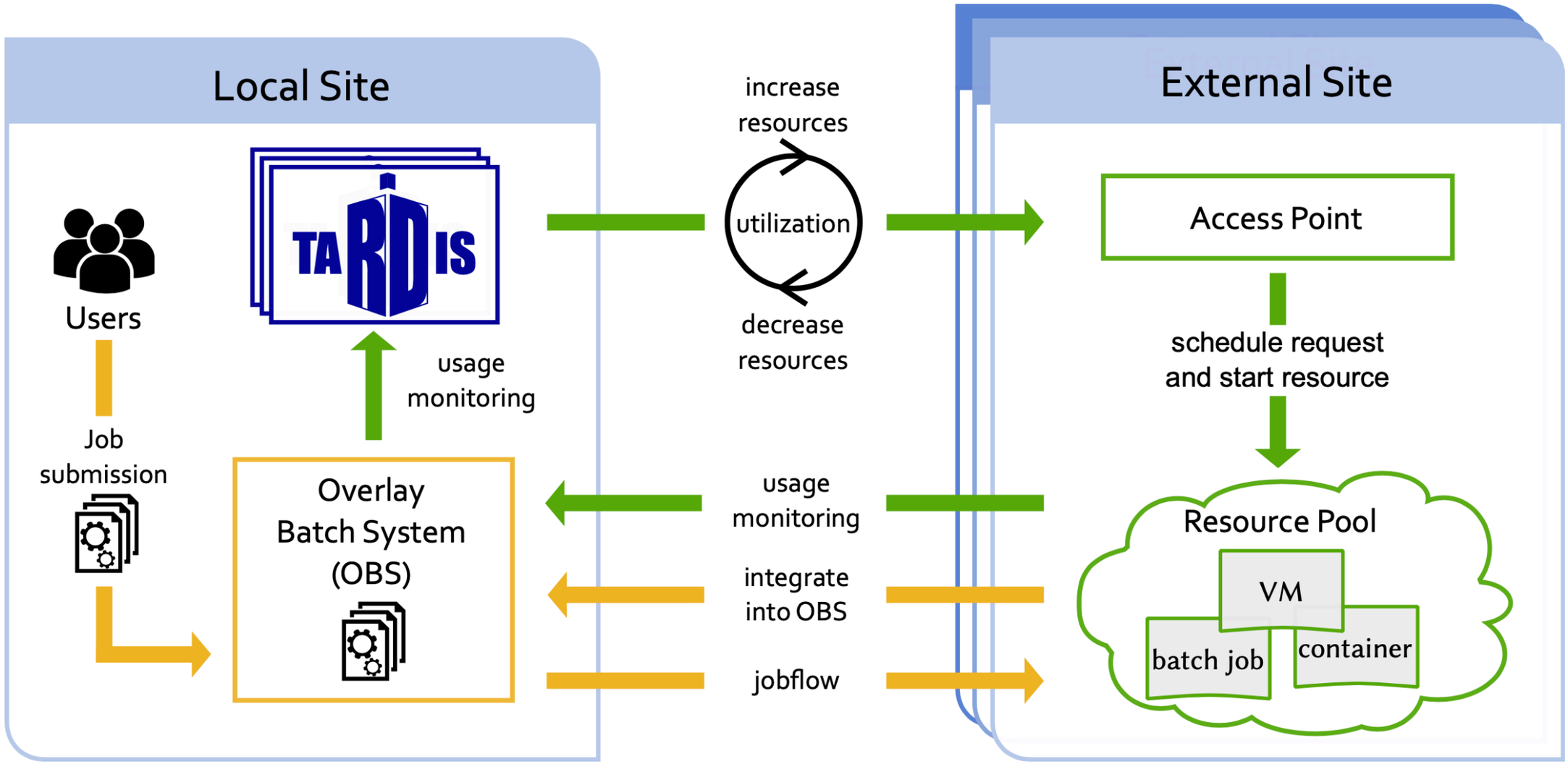
- It works so far, but…
- …connection glidein <-> HTCondor pool unstable
- Seemingly random outages between a few minutes and ~1h, connection comes back again
- Payload not effected

Conclusion and next steps

- Fixing the connection issues
- Ask the responsible admins for an exception in the fire wall
- Could maybe fix the connection problems
- Scale test with fully working setup/stress test

Backup

Schematic overview of TARDIS



https://doi.org/10.1051/epjconf/202024507038

COBalD's metrics

- Set I contains resource types (RAM, CPU,…)

- Allocation: measures how full a resource is/whether jobs still fit on it

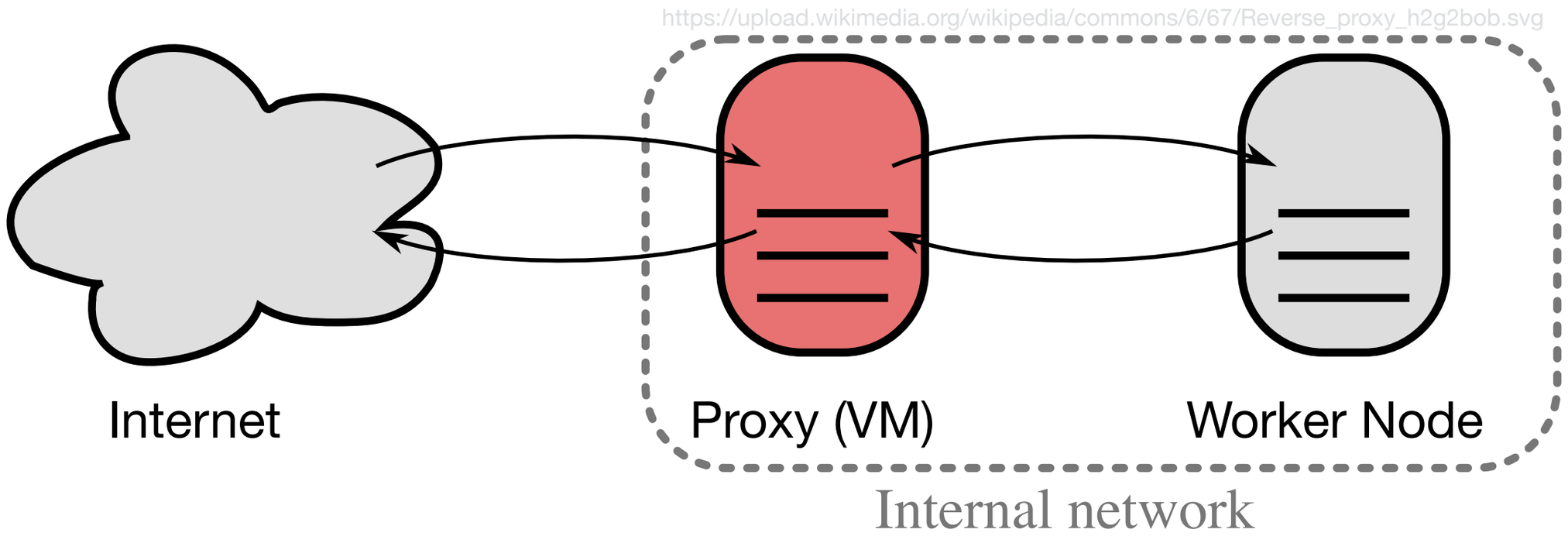
$$\text{allocation} = \max_{i \in I} \left(\frac{\text{used}(i)}{\text{requested}(i)} \right)$$

- Utilization: represents the suitability of the resource for the jobs (high utilization hints a good fit between jobs and requested resource)

$$\text{utilization} = \min_{i \in I} \left(\frac{\text{used}(i)}{\text{requested}(i)} \right)$$

Helmholtz Data Federation (HDF) Cloud

- Cloud computing platform offered by Jülich Supercomputing Centre (JSC)
- Resources produced as part of the Helmholtz Data Federation
- Provides access to the HPC file system among other things



- Connection made from port 5555 on Worker Node is forwarded to port 5432 on VM (inside container)

- Access to and caching of conditions database and CVFMS by Squid Proxy on VM via port 3306 (outside container)

(Frontier) Squid

- Based on standard Squid http proxy cache software
- Contains configuration defaults and bug fixes that are known to work well with the applications used on the grid
- Pre-configured for use by the Frontier distributed database caching system
- Used by ATLAS and CMS to access conditions data

Putting everything into a batch job

- Setup ssh tunnel
- Mount CVMFS repositories
- Launch singularity container and bind-mount CVMFS repositories to /cvmfs

„Setup“ on the worker node

- Enter the now running container

- Launch glidein
- …
- Profit

Work that is done inside the container

- Code available at https://gitlab.desy.de/thomas.madlener/cms-drp-jsc