Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

IDIUM

Federal Ministry of Education and Research

Status of *PHYSnet* cluster integration

FIDIUM Collaboration Meeting | RWTH Aachen | 30 September – 1 October 2024

Johannes Haller, Johannes Lange, Daniel Savoiu, Hartmut Stadie

U Hamburg commitments in FIDIUM

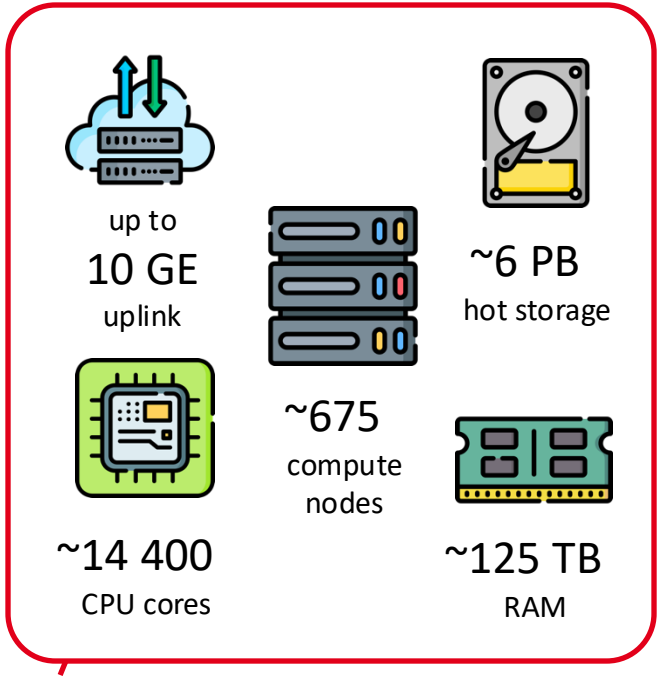
- Topic II – **Data lakes, distributed data, caching**
  - investigate and deploy data caching technologies
  - integrate dynamic data caches near newly integrated CPU resources

- Topic III – **Adaptation, testing, optimization**
  - deploy tools developed within FIDIUM to selected computing centers
  - integrate into production/analysis environments of HEP experiments
  - optimize to requirements for typical analysis workflows

PHYSnet cluster

compute resources shared by all institutes of physics faculty

- heterogeneous cluster, various queues for diverse applications:
  - ***idefix.q*** – mixed single-threaded applications
  - ***infinix.q*** – for multi-node applications using MPI + InfiniBand
  - ***obelix.q***, ***epyx.q*** – for large-memory applications
  - ***graphix.q*** – for GPU applications
- parts reserved for exclusive use by various project groups
  - high flexibility for tailoring to individual/group use-cases
  - can integrate dedicated resources for HEP applications
- adaptable to HEP workflows using containerization technologies



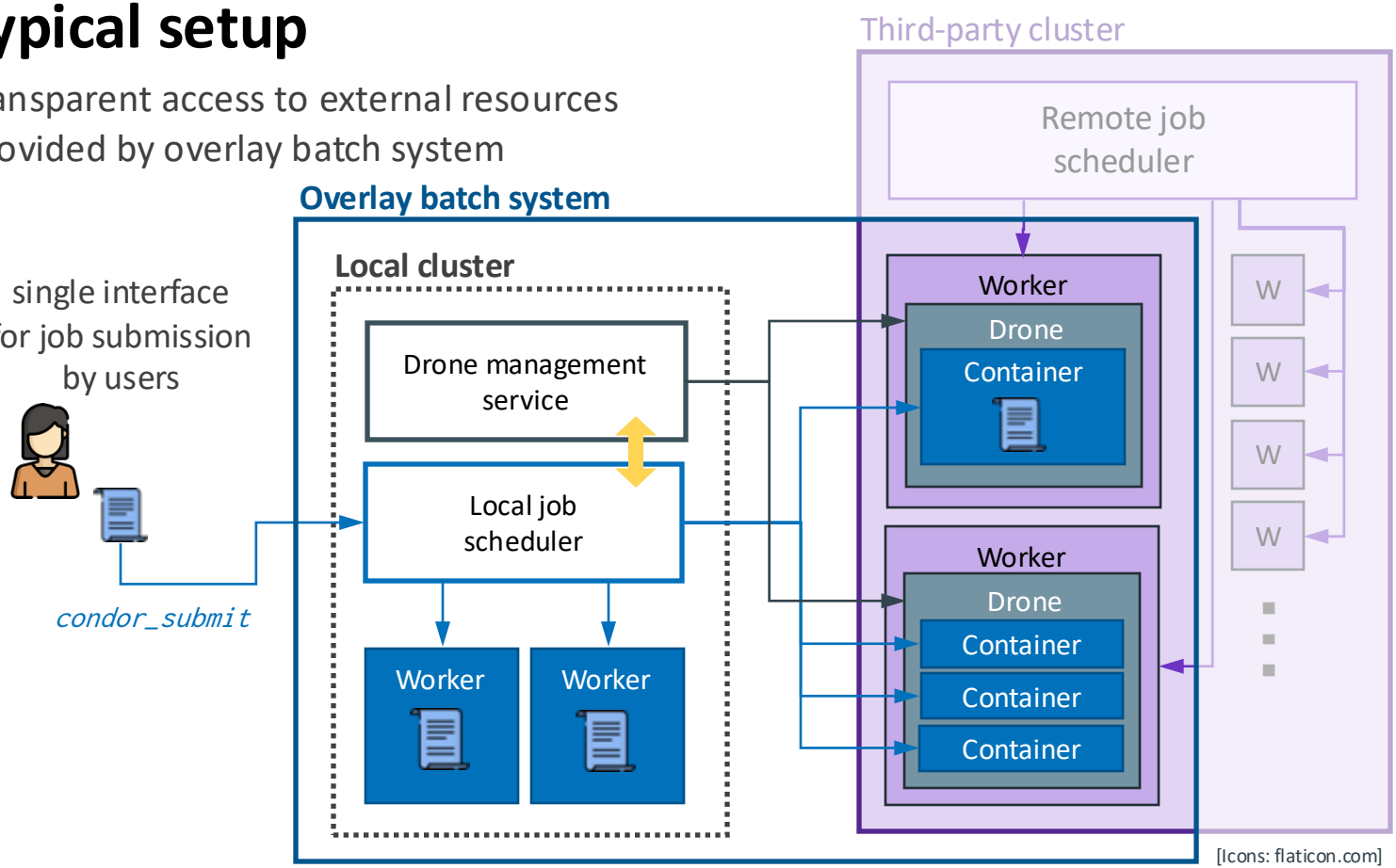
[Icons: flaticon.com]

	PHYSnet	Typical WLCG sites / *NAF*
OS	***Ubuntu***	***RedHat***-based (SLC/CentOS)
Batch system	***SGE****	***HTCondor***

) transition to **SLURM in progress*

Typical setup

transparent access to external resources provided by overlay batch system

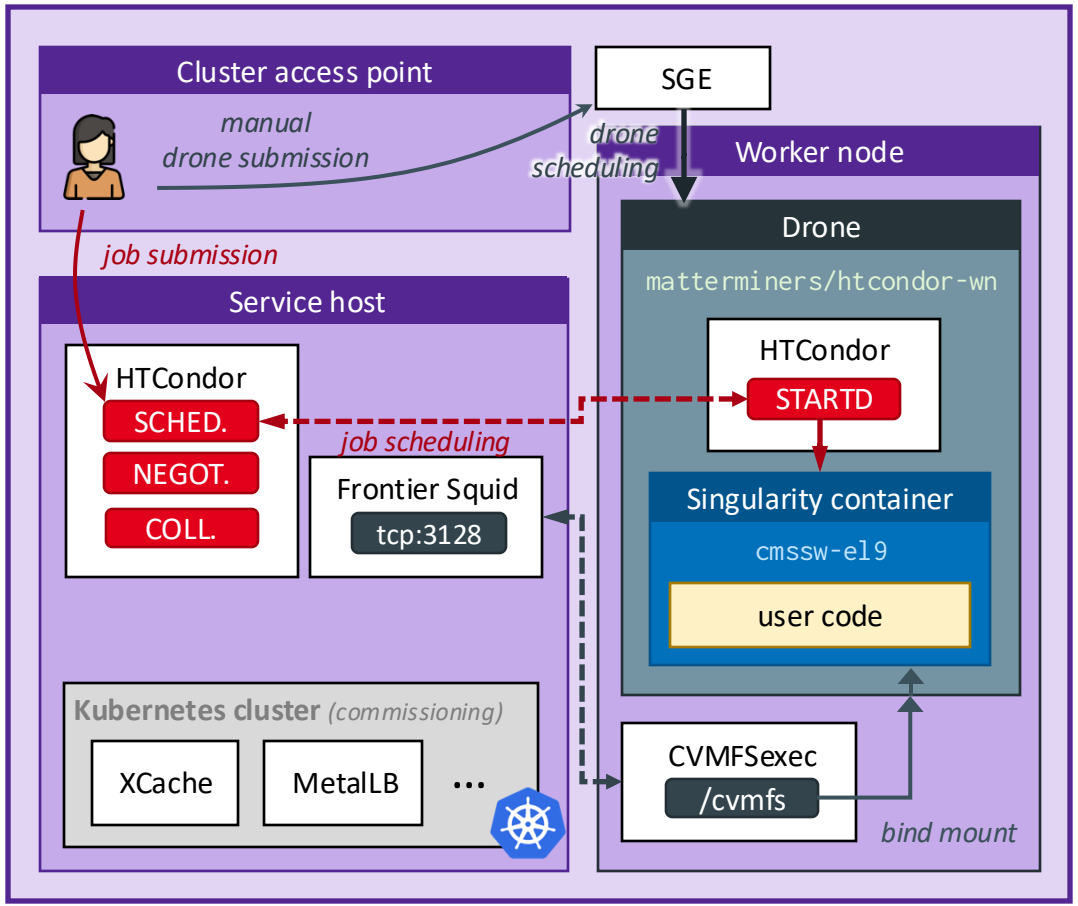


drone concept for acquiring and holding on to resources

remote jobs run in containers that emulate familiar environment

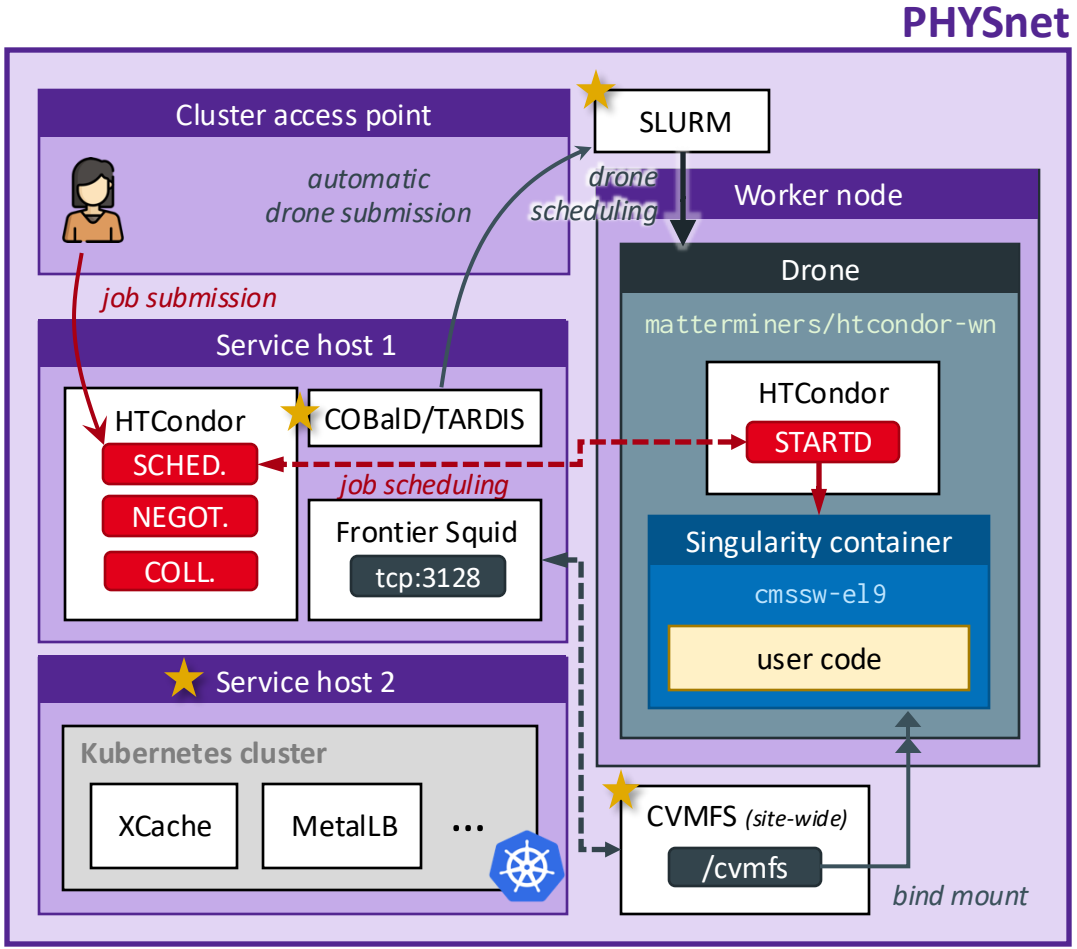
Current setup

- working setup for scheduling HEP analysis jobs to *PHYSnet* cluster
  - central ***HTCondor*** instance
  - jobs scheduled to drone containers provisioned via native ***SGE*** batch system
- obtained dedicated resources for hosting HEP-specific services
- unpacked container images taken from `/cvmfs/unpacked.cern.ch`
- moved to EL9 grid environment for job containers



Planned developments

- second dedicated host for caching
- CVMFS site-wide installation
  - alternatively, set up via ***cvmfsexec*** inside drone container
- adapt to *PHYSnet* transition to SLURM, expected to complete by the end of the year
- set up COBalD/TARDIS instance for automated drone management



HTCondor setup

Service host

- condor **v10.0** *(to match worker node)*
- system-wide installation, configured with ***central manager***, ***submit*** roles
- authentication via pool password
- **schedd** runs here to accept user jobs

Drone / Worker node

- ***matterminers/htcondor-wn*** container developed by KIT, provides ***HTCondor*** instance configured with ***execute*** role
  - **startd** runs inside drones & connects to other HTCondor daemons
  - dynamically updated configuration from external git repo using ***condor-git-config***
- jobs run in predefined singularity container `cmssw/el9`, from `/cvmfs/unpacked.cern.ch` with bind-mounted `/cvmfs`

Other developments

- set up ***ansible*** playbook with roles for all services on the central head node
  - useful for ***documentation*** & ***redeployment***
- continued development of grid submission tool ***grid-control***
  - adapted to ***python3.9*** and ***el9*** grid UI
  - implemented support for stage-in and stage-out via ***WebDAV*** protocol

Summary

- continued development of ***PHYSnet*** cluster setup for HEP analysis jobs
- obtained and commissioned dedicated resources for providing HEP-specific services (***CVMFS***, ***Frontier-Squid***, ***XCache***)
- set up ***HTCondor*** batch system for job submission on main service node

Next steps

- adapt to ***SLURM*** as soon as it is deployed at ***PHYSnet*** (transition in progress)
- improve provisioning of CVMFS, either by site-wide installation or via modified drone image
- large-scale testing of analysis workflows and collection of performance metrics
- COBalD/TARDIS, integration into overlay batch system at e.g. NAF

Thank you for your attention!