Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Federal Ministry
of Education
and Research

Status of *PHYSnet* cluster integration & test of analysis workflows

DCMS meeting | 27 April 2023

Johannes Lange, Daniel Savoiu, Hartmut Stadie

Reminder: *PHYSnet* cluster @ UHH

compute resources shared by all institutes of physics faculty

- heterogeneous, multiple pools/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
- want to use these resources for HEP workflows
  - requires adaption using ***containerization*** technologies
  - transparent integration into HEP-specific infrastructure

up to
10 GE
uplink

~6 PB
hot storage

~675
compute
nodes

~14 400
CPU cores

~125 TB
RAM

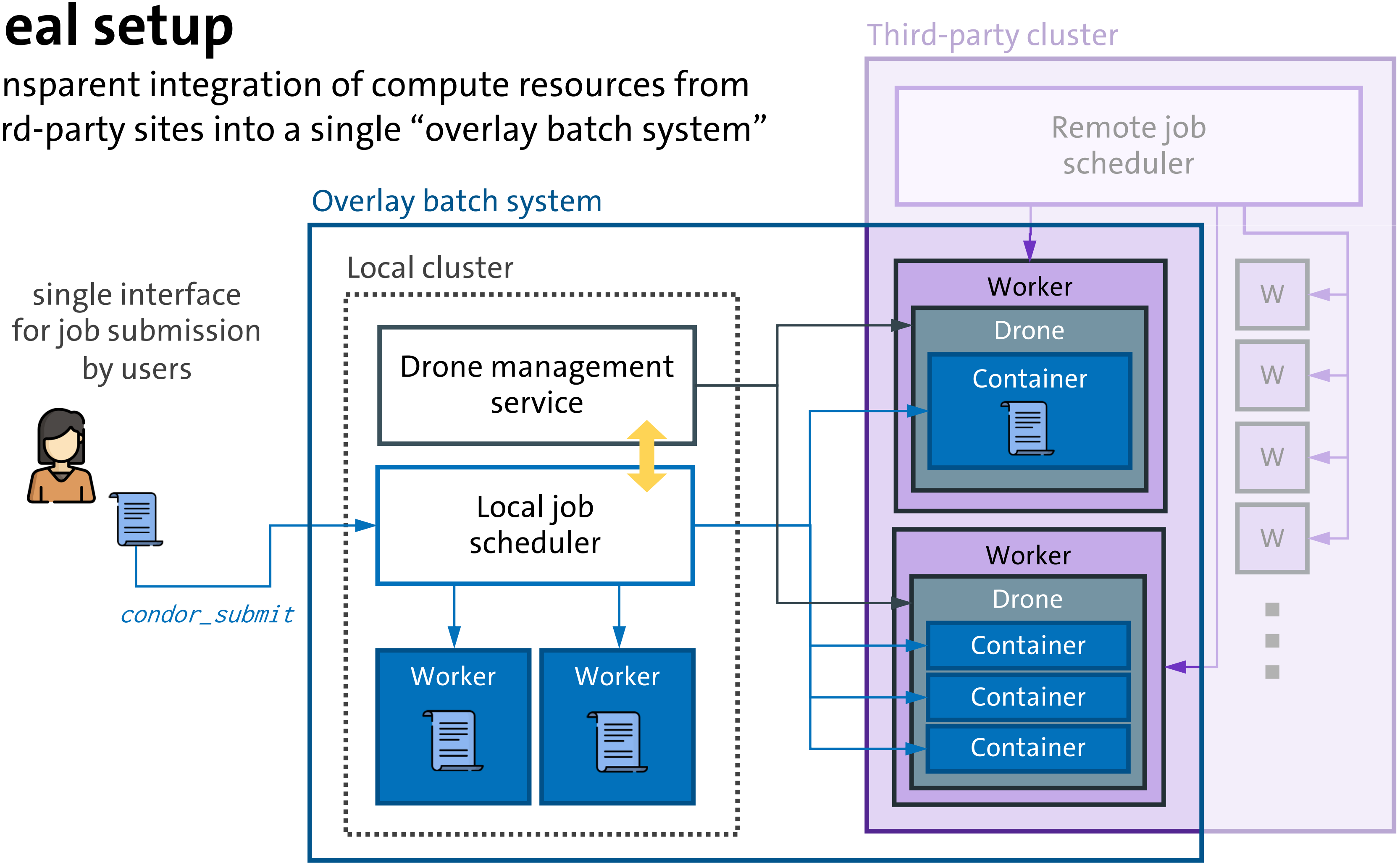
[Icons: flaticon.com]

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

*(transition to **SLURM** planned for this year)*

Ideal setup

transparent integration of compute resources from third-party sites into a single "overlay batch system"



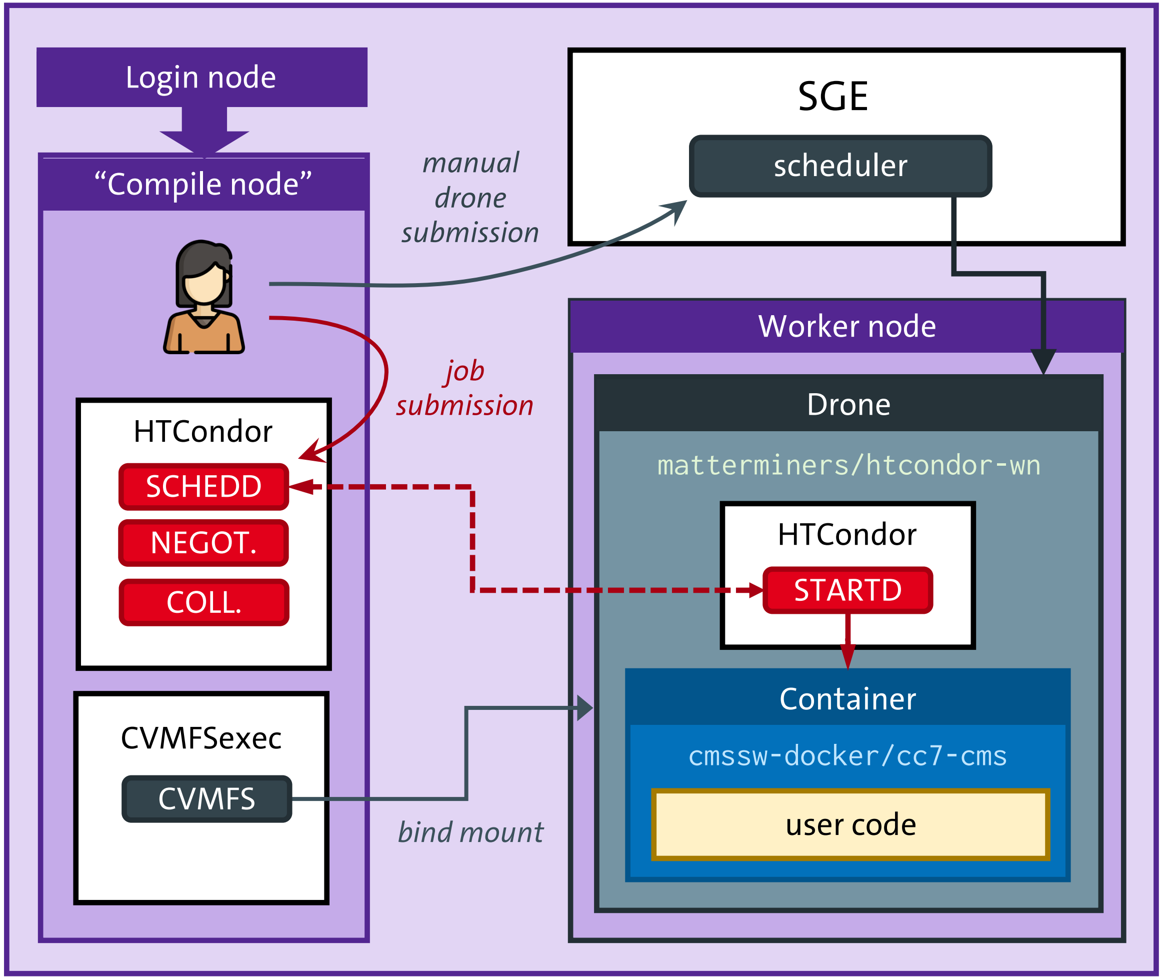
drone concept for acquiring and holding on to resources

remote jobs run in containers that emulate familiar environment

[Icons: flaticon.com]

Current setup at PHYSnet

- *for now*: small dedicated ***HTCondor*** instance
  - **schedd** running on general-purpose "compile node" as a central manager
- ***drones*** submitted to local SGE batch system as long-running jobs
  - **startd** runs inside drones & connects to other HTCondor daemons
- ***CernVM-File System*** (CVMFS) mounted in userspace using [***cvmfsexec***]
  - **bind-mounted** at `/cvmfs` inside drone container
- all components running ***without elevated privileges***



Container sources

- unpacked container images taken from `/cvmfs/unpacked.cern.ch`
  - for *drones*: ***htcondor-wn*** image developed by KIT
  - for *job containers*: standard CMS CentOS 7 image ***cc7-cms***
- ***htcondor-wn*** provides flexibility to dynamically reconfigure drones
  - using ***ansible*** + ***condor-git-config*** to reconfigure HTCondor without needing to restart container

Simple workflows at *PHYSnet*

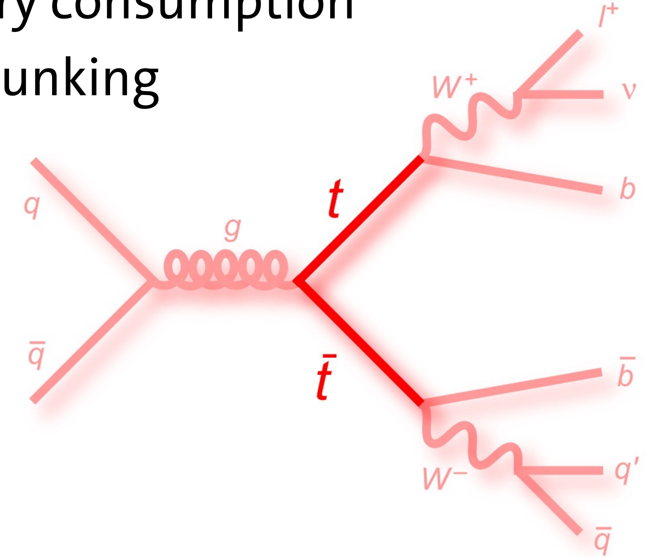
- ***simple file transfer*** from/to grid storage elements via ***gfal2*** libraries + ***X.509*** authentication
  - works without problems, used to benchmark file transfer to various grid sites

- typical EDM file processing with ***CMSSW***
  - precompiled user analysis code can run inside drones using CMS-specific containers
  - actual running over input files requires valid SITECONF, investigating possible solutions

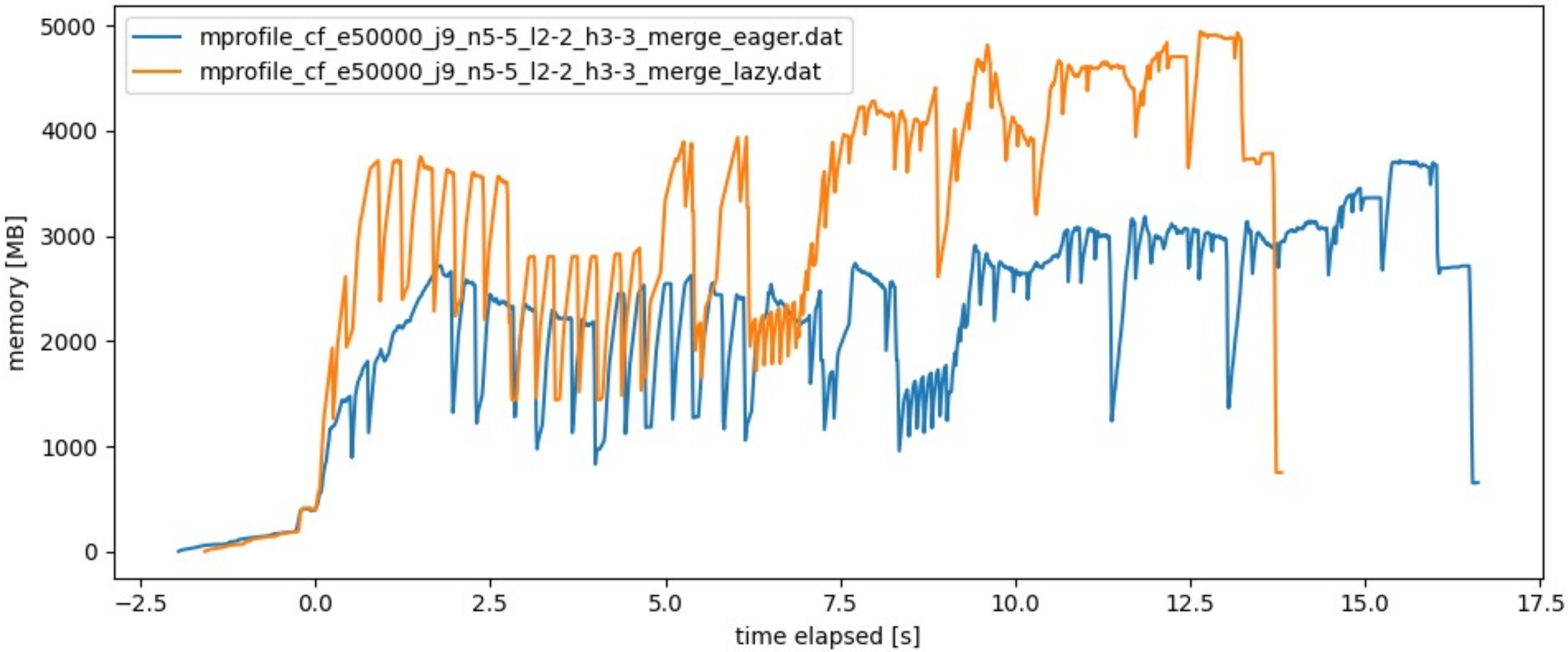
- *planned*: tests with modern workflows using new columnar framework ***columnflow***
  - array-at-a-time computation instead of event-at-a-time
  - complete orchestration/job management with ***HTCondor*** backend
  - largely experiment-agnostic, reads in flat *n*-tuples in a variety of formats (ROOT, Apache Arrow/Parquet)
  - Run-3 CMS analyses based on NanoAOD are in development at UHH2, plan to use these workflows in future benchmarks

Excursion: Optimizing columnar event reconstruction

- use case: ttbar reconstruction
  - challenging due to large combinatorics of *N* jets per event
    [$O(3^N)$ possible assignments of jets to leptonic/hadronic decay]
- ***columnflow*** implementation uses ***AwkwardArray*** + chunked processing
  - ~100 000 events at a time → need to optimize memory consumption
  - *ansatz*: factoring combinations by multiplicity, sub-chunking
- evaluate profiled memory allocations in analysis code to compare different implementations



Excursion: Optimizing columnar event reconstruction



- *here*: “**lazy**” vs “**eager**” merging of results from “sub-reconstructions”

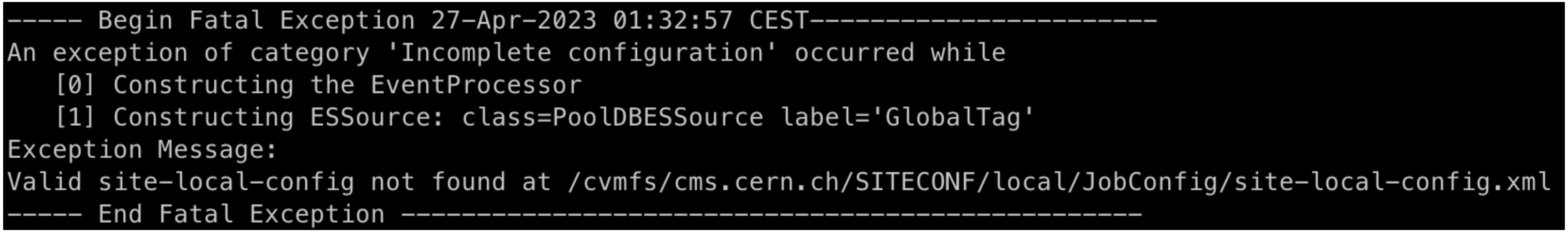
Summary, issues & outlook

- working test setup for HEP job submission at *PHYSnet* faculty cluster
- using container images/tools provided by CERN or DCMS groups (KIT)
- typical analysis workflows tested, some issues to be resolved (SITECONF)

- ***HTCondor*** configuration using dynamic partitionable slots sometimes leads to failed job matches → to be investigated
- currently all components are running on *PHYSnet* infrastructure → offload HTCondor central manager and scheduler to outside
- using pool password for authentication → switch to token-based authentication
- separate CVMFS instance per job → site-wide installation
- drones are started manually → automate using COBalD/TARDIS (after transition from SGE to SLURM)

Backup

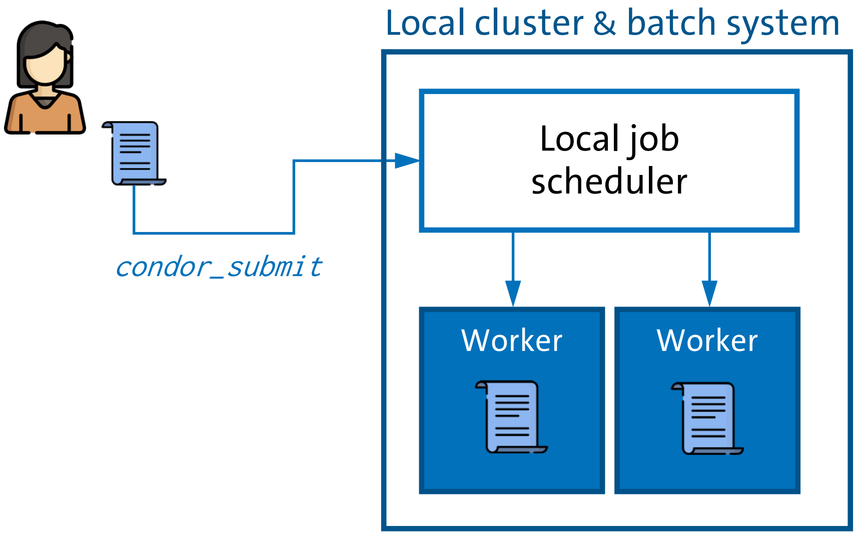
CMSSW local site config error



```
----- Begin Fatal Exception 27-Apr-2023 01:32:57 CEST-----------------------
An exception of category 'Incomplete configuration' occurred while
   [0] Constructing the EventProcessor
   [1] Constructing ESSource: class=PoolDBESSource label='GlobalTag'
Exception Message:
Valid site-local-config not found at /cvmfs/cms.cern.ch/SITECONF/local/JobConfig/site-local-config.xml
----- End Fatal Exception -------------------------------------------------
```

Standard use case

users submit analysis jobs to local resources



- user jobs written to run on fixed software environment provided by the cluster
  - e.g. ***CentOS 7*** + ***CMSSW***

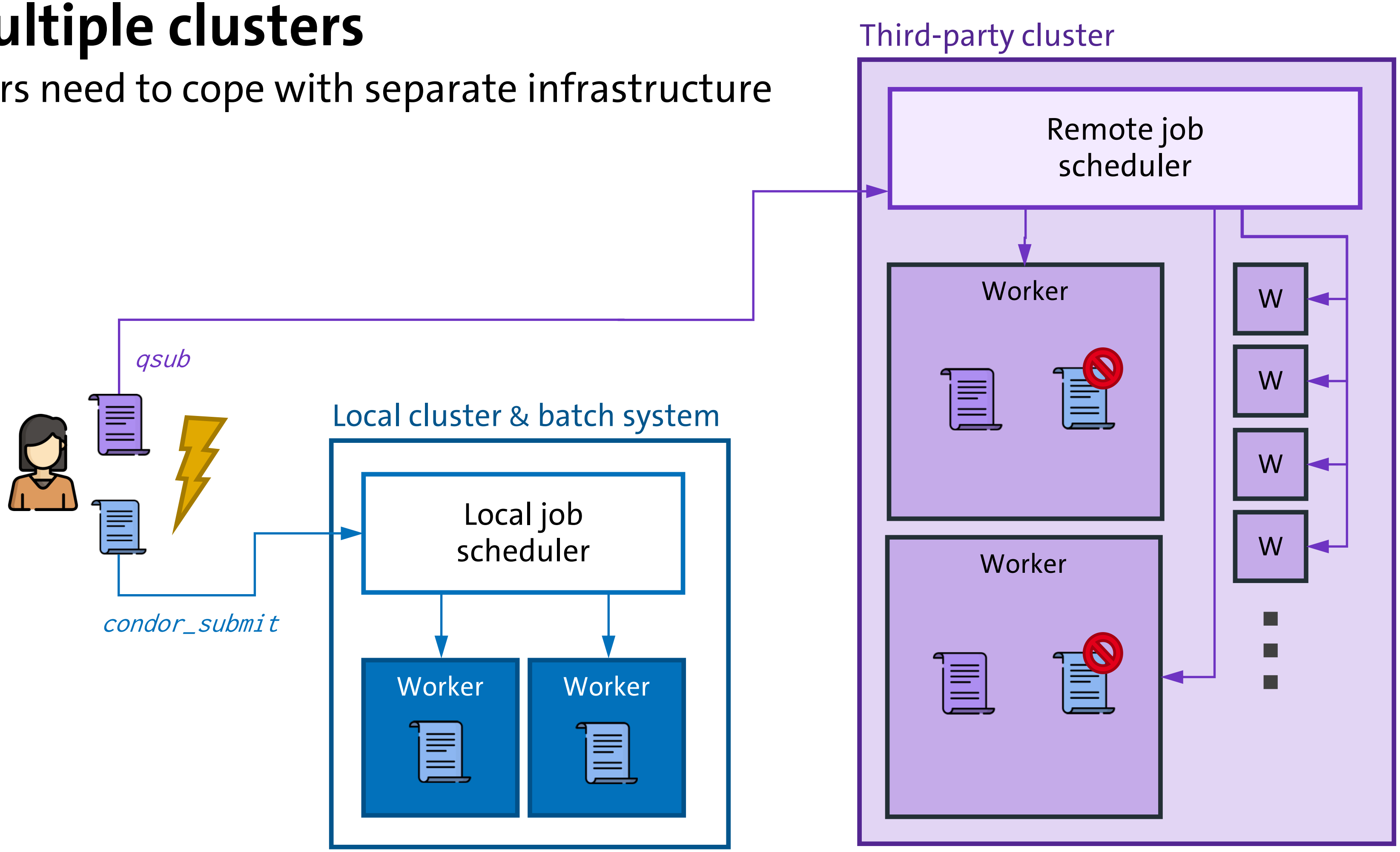
- users submit jobs to local cluster
  - scheduled to run on local resources “owned” by the cluster

- what if more resources are needed?

[Icons: flaticon.com]

Multiple clusters

users need to cope with separate infrastructure

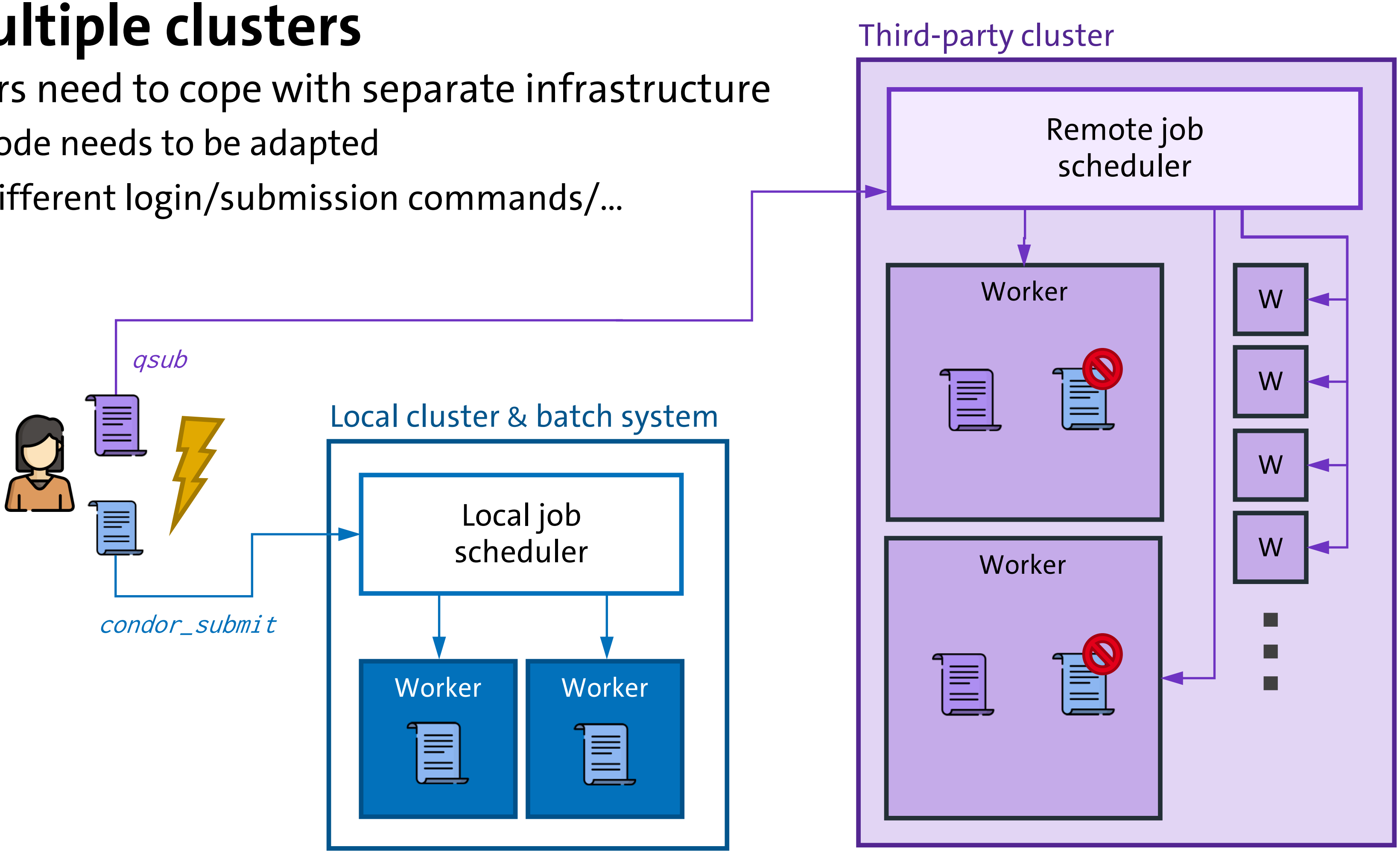


[Icons: flaticon.com]

Multiple clusters

users need to cope with separate infrastructure

- code needs to be adapted
- different login/submission commands/...

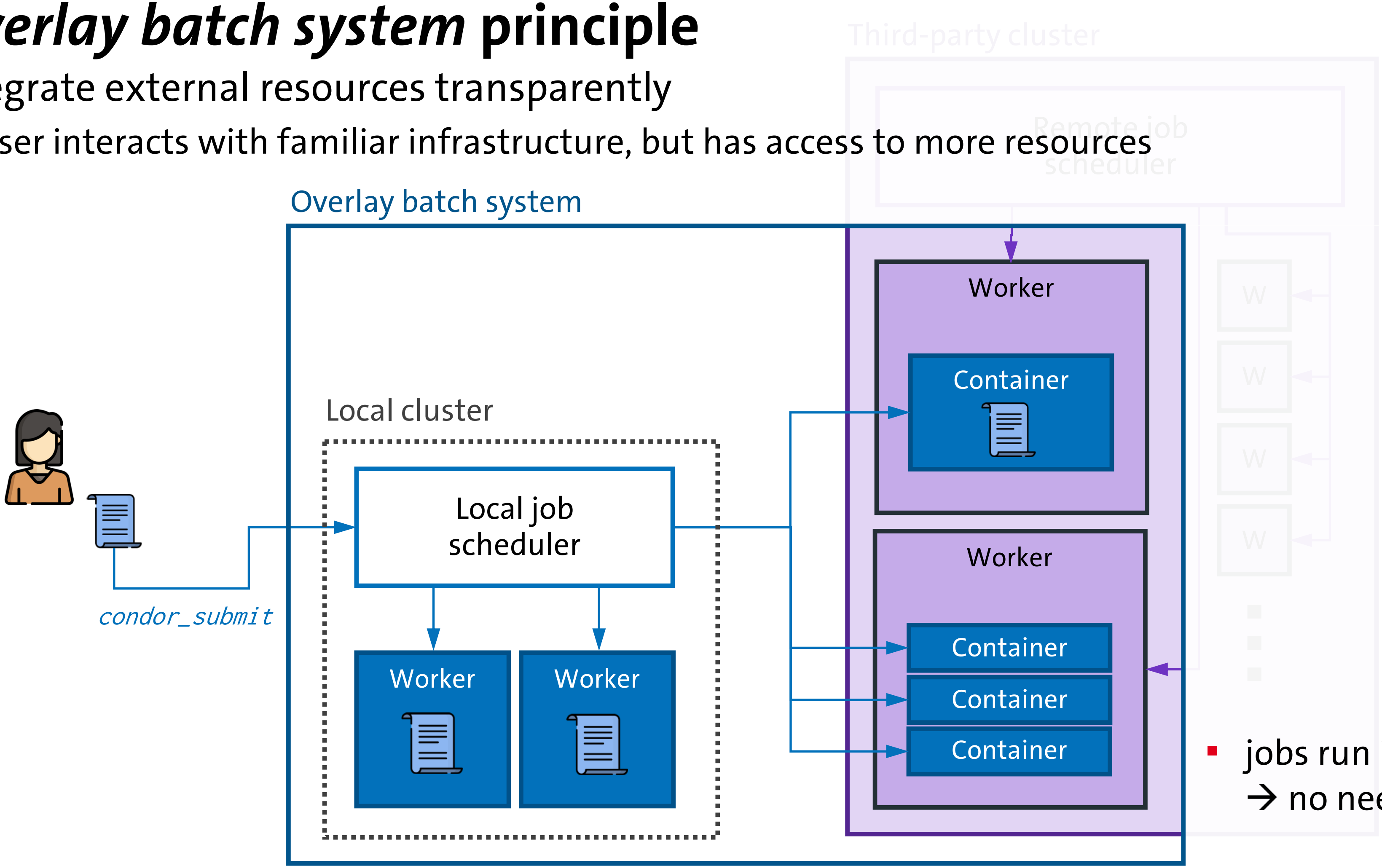


[Icons: flaticon.com]

Overlay batch system principle

integrate external resources transparently

- user interacts with familiar infrastructure, but has access to more resources



- jobs run in containers
  → no need to adapt code

[Icons: flaticon.com]