Pilot Job Frameworks

SURFsara – Online Grid Computing Course

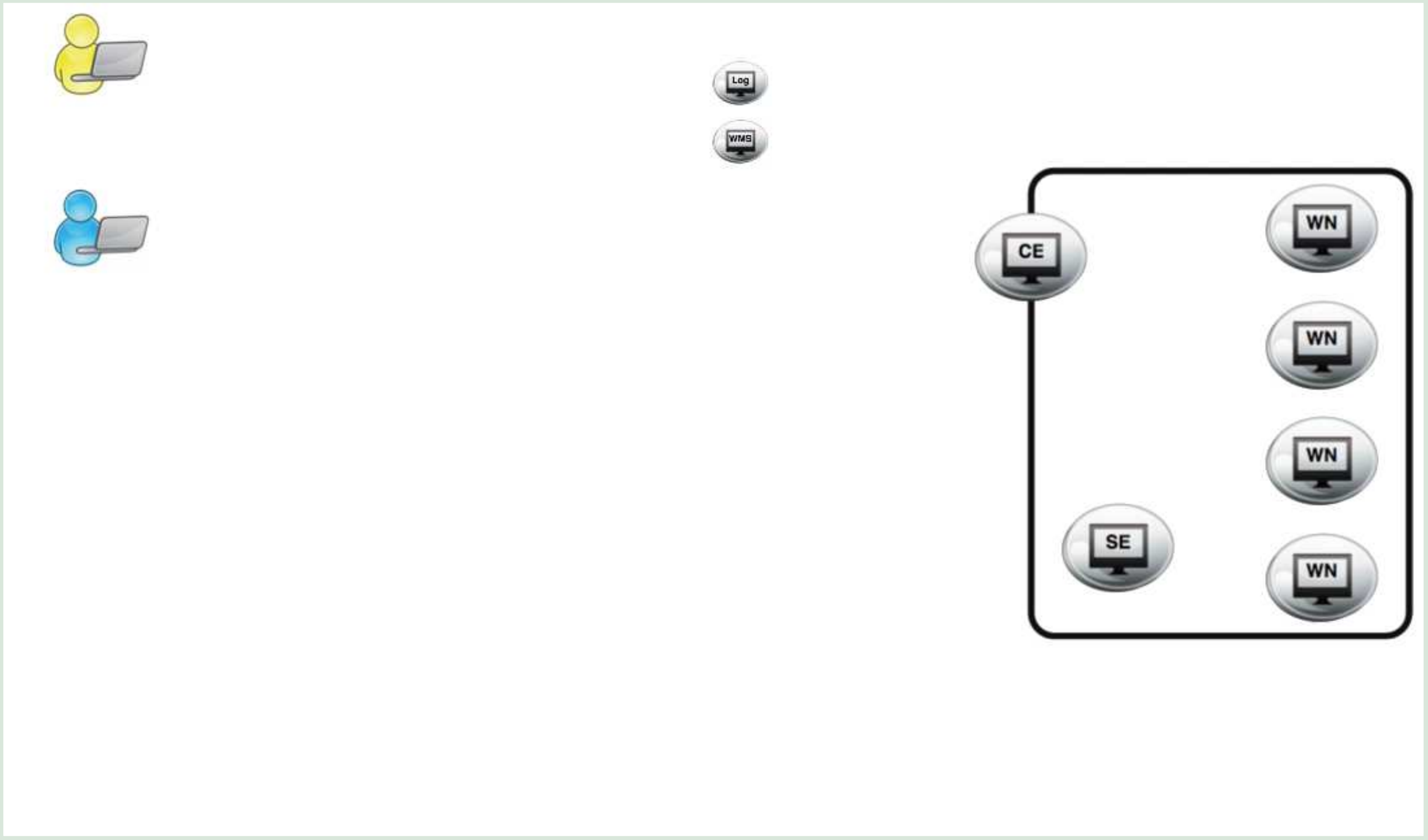
Outline

- ❑ What we did so far
- ❑ WMS challenges
- ❑ Pilot jobs

Job submission using the WMS

- ❑ Prepare software
- ❑ Create jdl
- ❑ Submit parametric job
- ❑ Find out which jobs failed
- ❑ Resubmit failed jobs

WMS animation



WMS challenges

- **Scheduling overhead**
  - Job matching requirements
  - Short run times
- **Scheduling inefficiencies**
  - Jobs locked in queue
- **Keeping track of failed jobs**
  - Failing jobs
  - Resubmissions

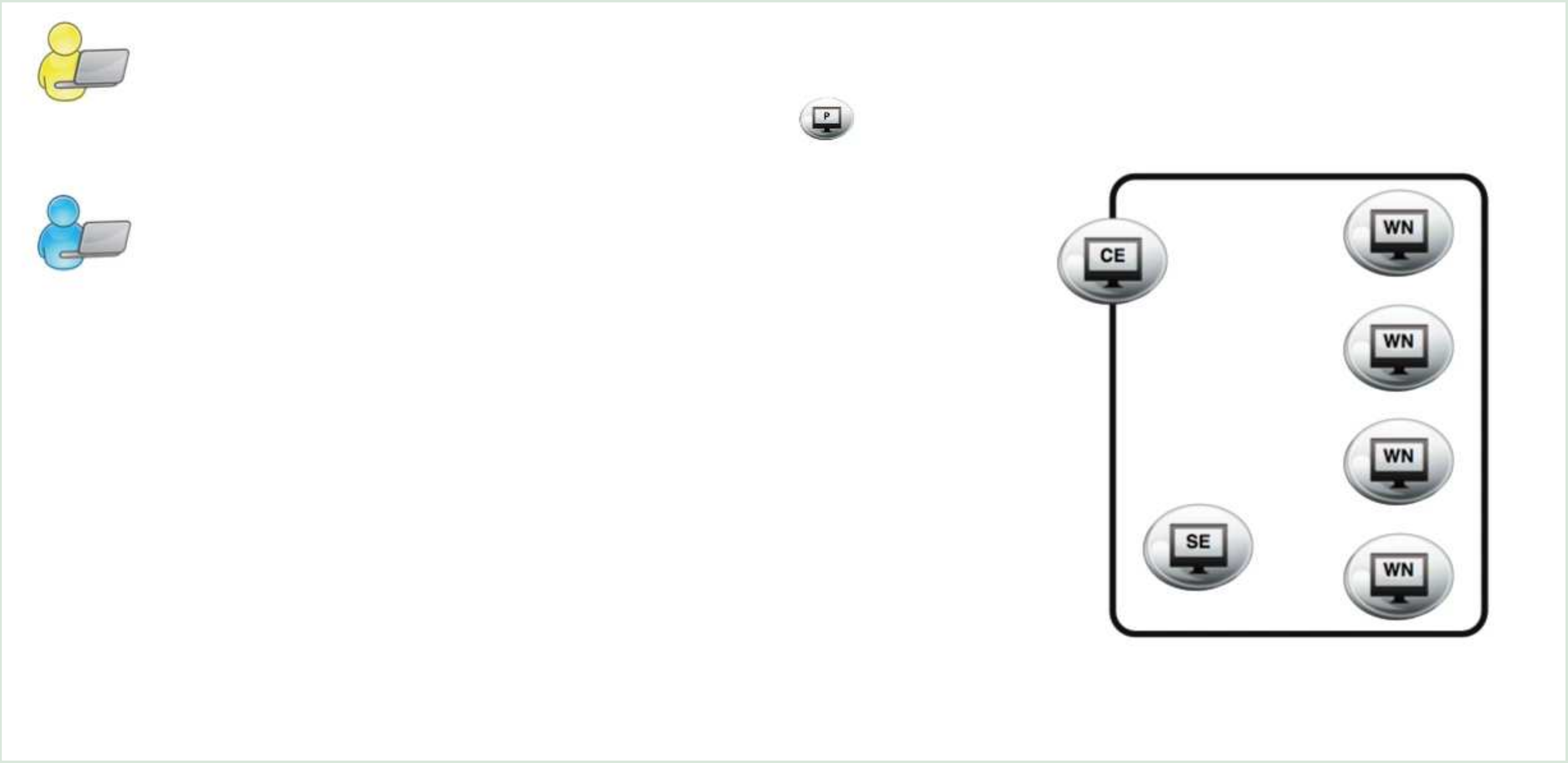
Other requirements

- **Saving small outputs**
  - Saving and retrieving difficulties
  - SRM latency
  - Hard to combine into larger blocks
- **Logging**
  - Matching arguments to outputs
  - Debugging

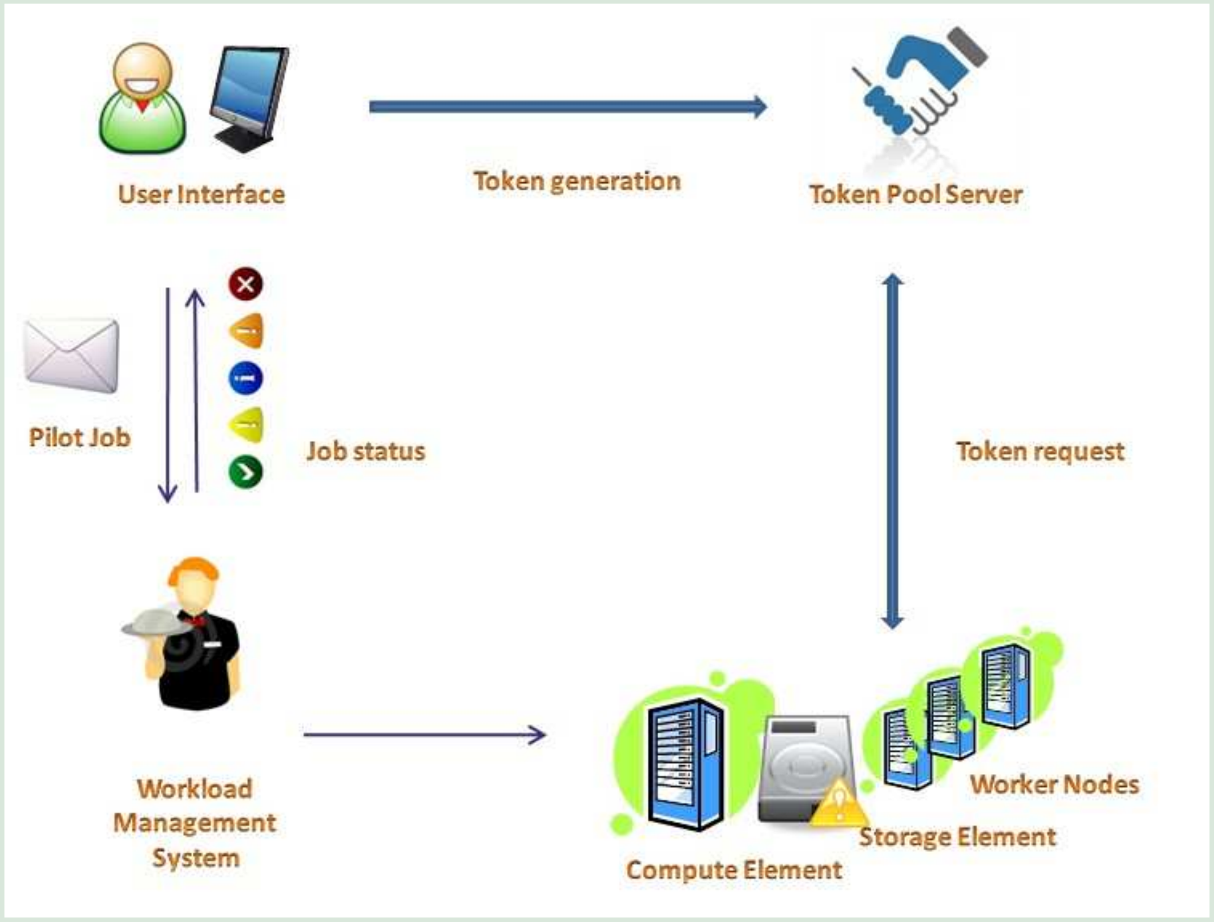
Pilot jobs

- ❑ Contain the basic information for running an application
- ❑ Rely on an external source to keep track of job tokens
- ❑ Fetch inputs at the latest possible moment
- ❑ Keep fetching new inputs until
  - ▪ all jobs are done
  - ▪ the job runs out of wall clock time
  - ▪ the job crashes

Pilot job animation



Pilot job life cycle



Tokens VS jobs

- Tokens: individual pieces of work
- Jobs: processes scheduled on worker nodes
- You could just as well use your laptop to act as a worker and just run the application locally!

Remarks

- Submitting jobs

- Resetting failed jobs