

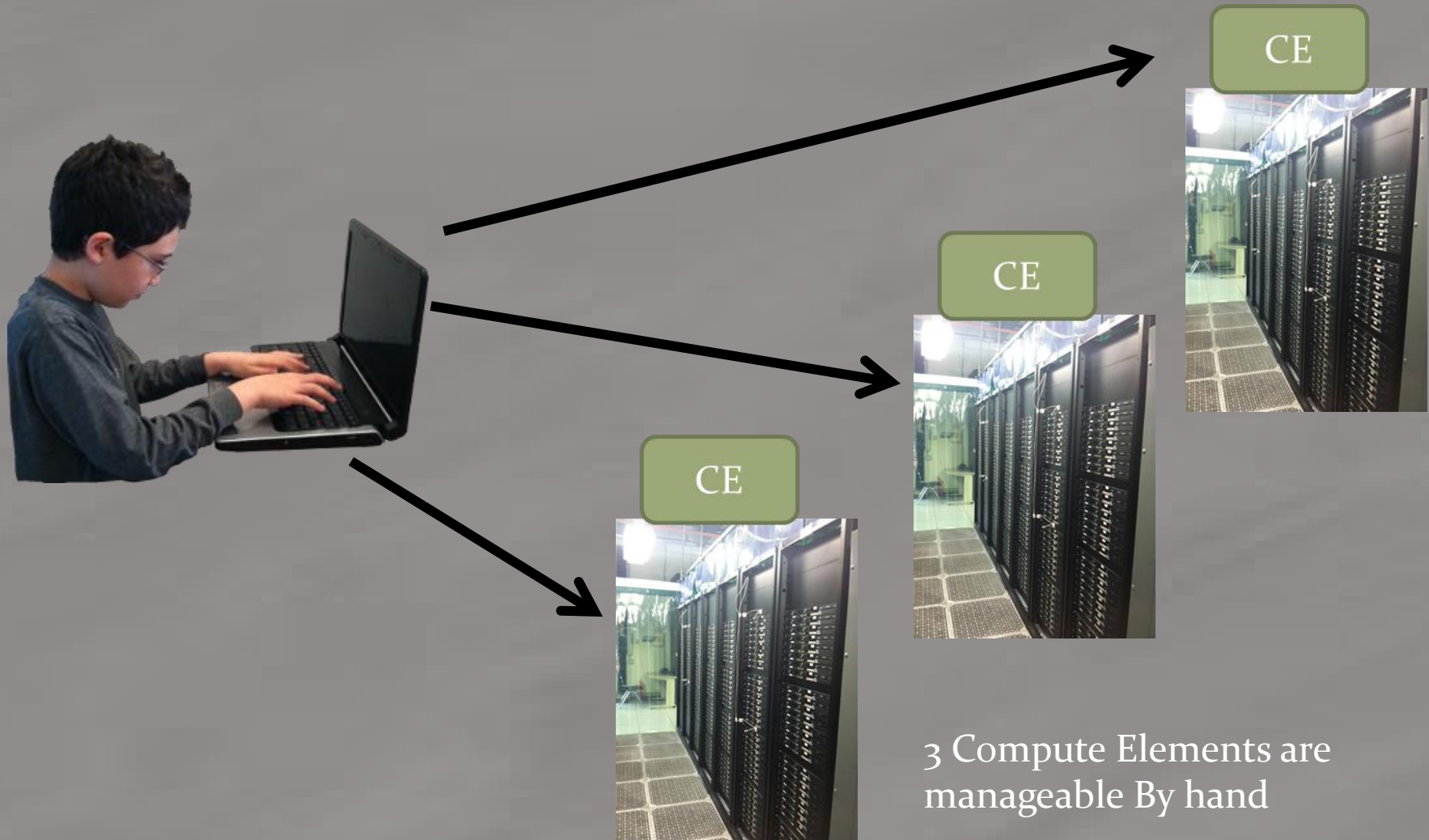
glideinWMS

Stakeholder Meeting

July 24, 2013

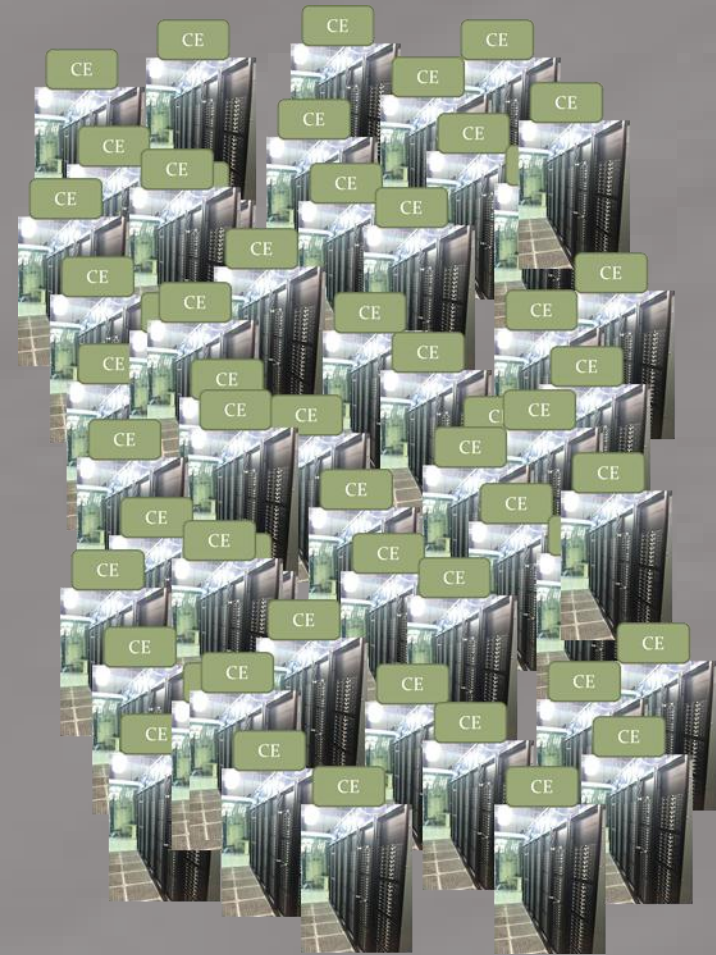
Burt Holzman

Challenges of Grid Computing: Distributed Compute Resources



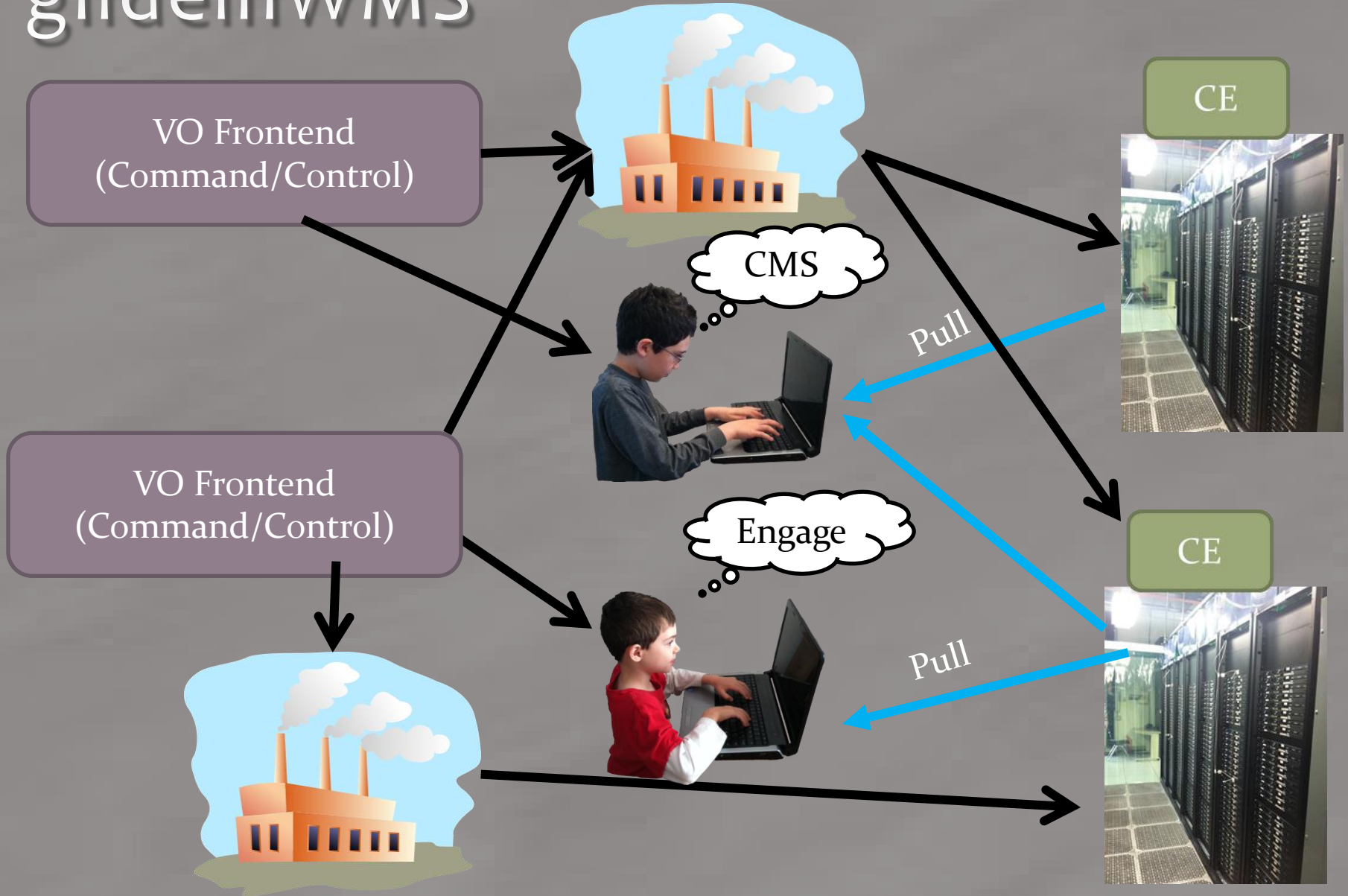
3 Compute Elements are
manageable By hand

Challenges of Grid Computing: Distributed Compute Resources



We need middleware – specifically
a Workload Management System
(and more specifically, “glideinWMS”)

glideinWMS

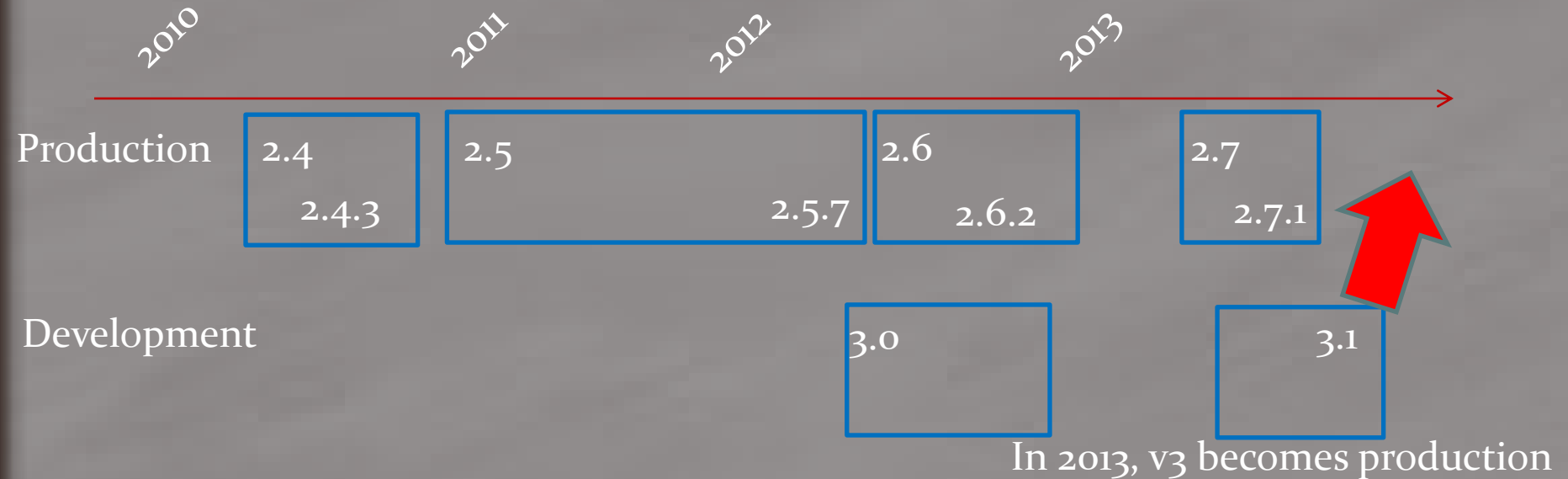


VO Frontend can talk to multiple factories

glideinWMS: quick facts

- glideinWMS is an open-source Fermilab Computing Sector product driven by CMS
- Heavy reliance on HTCondor from UW Madison and we work closely with them
- <http://tinyurl.com/glideinWMS>
- Contributors include:
 - Krista Larson (FNAL/Corral)
 - Parag Mhashilkar (FNAL/Corral)
 - Mats Rynge (ISI/USC/Corral)
 - Igor Sfiligoi (UCSD)
 - ~~Doug Strain (FNAL)~~
 - Anthony Tiradani (FNAL/CMS)
 - John Weigand (CMS)
 - Derek Weitzel (UNL)

glideinWMS: version timeline



- 2.4.x: privilege separation, aggregate monitoring, glexec control, glidein lifetime control
- 2.5.x: HTCondor TCP bulk updates, efficiency improvements, factory limits per frontend, excess glidein removal, shared ports, better user pool matchmaking
- 2.6.x: Better multislot support, ARC CE, more glidein lifetime controls, factory limits per frontend security class
- 2.7.x: Refactor for factory scaling, performance fixes, partitionable slot support
- 3.x: Cloud support, CorralWMS frontend support

Roadmap for glideinWMS

■ Scalability

- We can handle hundreds of destination CEs, but treat each permutation as a new CE
 - Refactored code from “one process per CE” v2.7, v3.1
- We can better scale HTCondor components (separate factory collector, for example)
- Work with HTCondor team on increasing HTCondor scalability (number of submitted jobs)

Roadmap for glideinWMS: 2/3

- Usability

- Easier installation and configuration

- We provide RPMs and tarballs, but it's far from push-button

- More consistent monitoring and logging (v3.1)

- Give more useful information in the hands of the VOs and users

- APFMon interface, for example (v3.2)

Roadmap for glideinWMS: 3/3

- Extensibility: beyond the grid



- This works **now** in v3.0, but is not our production release yet
 - Different cost models lead to new needed features: in progress



Extendability

- Sometimes this means making the cloud ready for you!
 - Parag is working on submission to Open Nebula (Fermicloud)
 - BH did some work on OpenStack:



All **My** Admin Documentation

Changes Important Changes Drafts Watched Changes Starred Changes

Search for owner:burt status:merged

	ID	Subject	Owner
▶ ☆	Id0278612	Remove instance_metadata_get_all* from db api (MERGED)	Burt Holzman
☆	Icca21ea2	Adds tests for tags in boto (EC2 API) (MERGED)	Burt Holzman
☆	Idb005b72	Change get_all_instance_metadata to use _get_instances_by_filters (MERGED)	Burt Holzman
☆	I88d40db3	Support Client Token for EC2 RunInstances (MERGED)	Burt Holzman
☆	Ibb2619e8	Sort output for unit tests in test_describe_tags before compare (MERGED)	Burt Holzman
☆	Ib5a593bd	Replace get_instance_metadata call in api.ec2.cloud._format_instances (MERGED)	Burt Holzman
☆	I1bf925c4	Change path for "Run one test" example in README.rst (MERGED)	Burt Holzman
☆	Idec460d8	Increase maximum URI size for EC2 API to 16k (MERGED)	Burt Holzman

What's Coming

glideinWMS 3.1 Production Release:
by end of July

(last show-stopping logger bug fixed yesterday!)

glideinWMS 3.1 Features

- Forward port of v2.7 features; v3.1 factory is backwards-compatible with v2 frontends
- Condor-C support
- Glideins can advertise error classads to user collector
- New tools: adding factory entries, converting v2 to v3 configurations
- Support for frontends to run external proxy creation/renewal scripts
- Disabling of glidein monitoring slots by default
- Better handling of system-level limits (e.g. rlimit nproc)
- Many bug fixes

```
[burt@baz gwms]$ git diff v3_0..v3_1_rc2 --stat | tail -1  
217 files changed, 13667 insertions(+), 5282 deletions(-)
```

(some) Stakeholder Requests

- OSG

- 2313: Add some pilot information to HTTP headers
- 2811: Plug-in XSLT architecture for frontends
- 3203: condor_q –analyze analogue for glideins
- 3204: Support file transfer plugins in pilot

- Fermilab

- 3686: Track batch system slot number through glidein startds

GlideinWMS 3.2 targets (1/2)

- Decreased memory usage
 - Better scaling led to lower steady-state memory consumption, but occasional big spikes
- Frontend blacklisting of entries
 - VOs want an easy way to stop matching jobs without tweaking configurations
- Periodic execution of validation scripts
 - Pilots only validate the worker node before the job executes – we could do better
- Partitionable slot metrics
 - The statistics on completed jobs is based around one slot per pilot, but multislot/partitionable pilots are configurable now

GlideinWMS 3.2 targets (2/2)

- Efficiency and performance improvements
 - We have refactoring work that has been waiting on a v3 production release
- Better High-Availability for frontends
 - Multiple frontends work, but will request more glideins than needed
- EC2 spot instances
 - This is the closest thing to “opportunistic” cloud computing that you pay for – grab cycles when they are the cheapest. We are working with HTCondor to support this in glideinWMS.
- Recover cloud logs if possible
 - Grabbing the console logs from VMs can help debugging
- Improvements to operations-level tools
 - E.g. letting `clone_glidein` pull from a central repository