



CMS private cloud usage

pre-GDB, 09 July 2013

Mattia Cinquilli
IT-SDC-OL



Outline

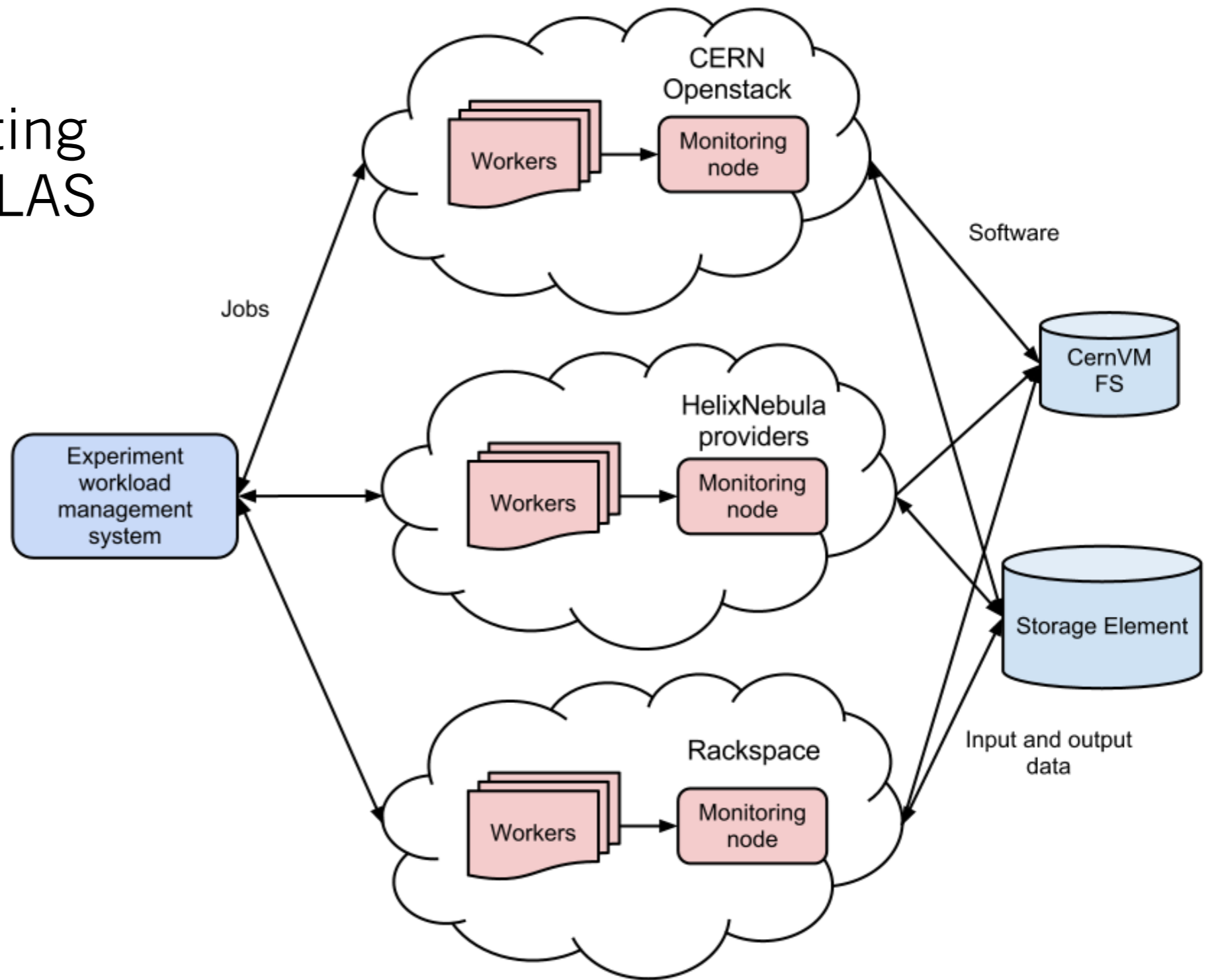
- Introduction
- AgileInfrastructure testing
- Contextualization
- DeltaCloud
- Summary

Introduction

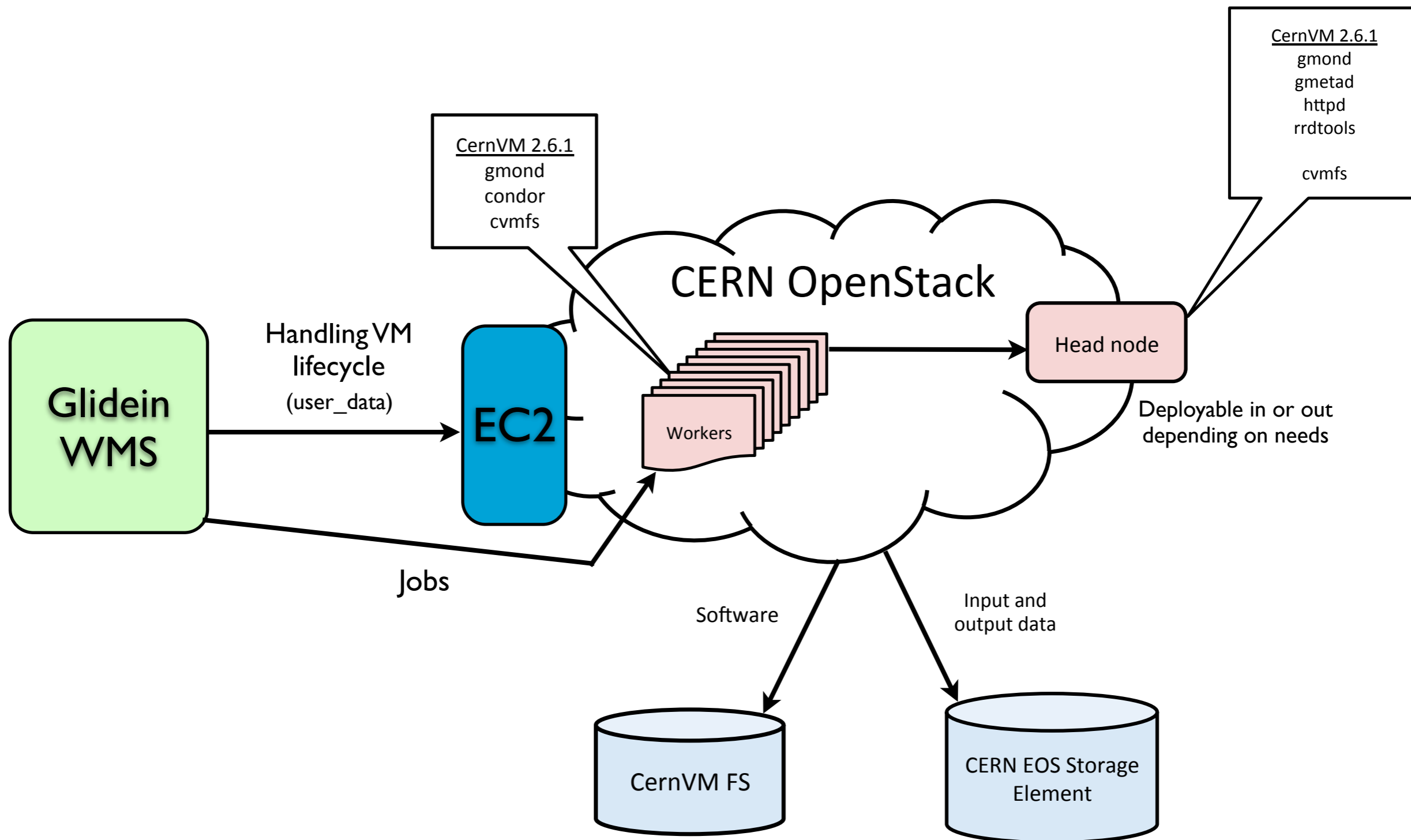
- Cloud offers several advantages
 - Clear separation between site/VO environment
 - Real machine under site's control, VM under VO control
 - Dynamic provisioning of resources
 - De facto standards
- CMS is evaluating several environments (private/public/opportunistic/...)
 - Commercial cloud resources to provide additional resources for peak use times?
 - HLT farm
 - Grid sites moving to cloud
 - Data preservation

Schema

- Parallel testing between ATLAS and CMS



CMS schema



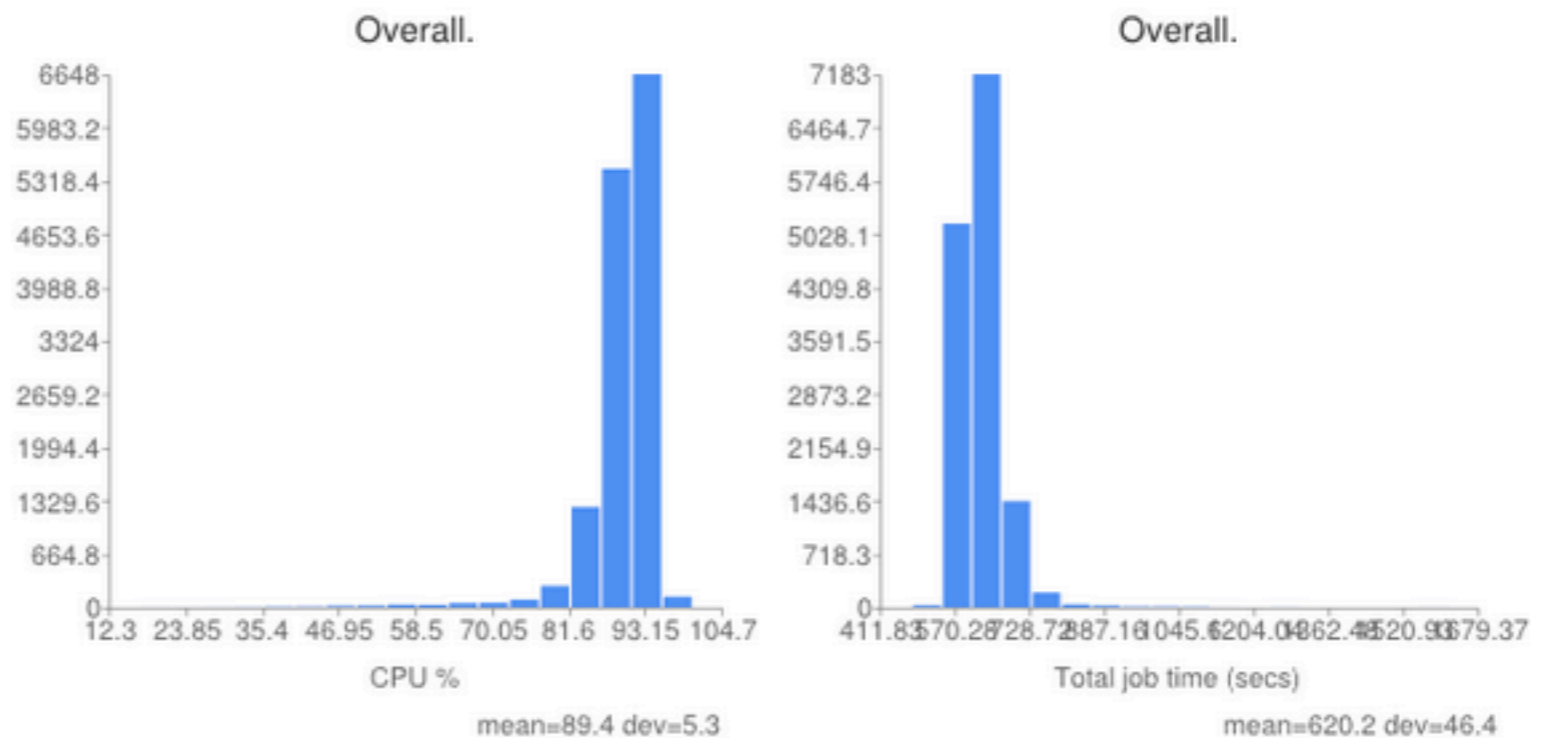
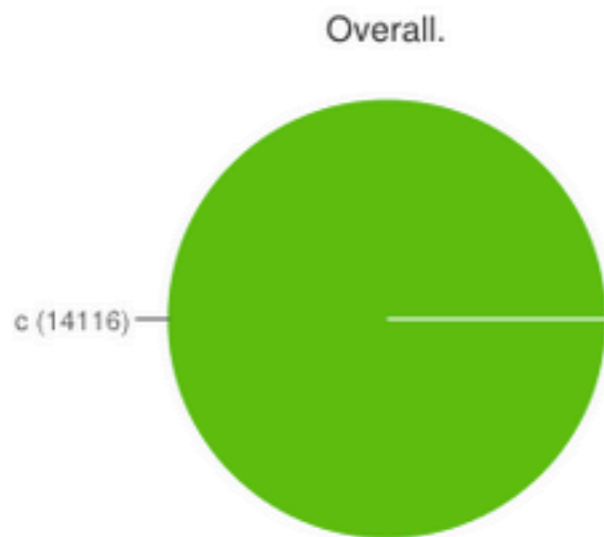
AgileInfrastructure Environment

- OpenStack based infrastructure
 - Folsom release
 - Both Nova and EC2 interfaces provided
- CernVM/SLC* images
 - CVMFS, Ganglia, Glidein bootstrap
 - but no dynamic contextualization (see later)
- Submission through WMAgent, CRAB-2 (HammerCloud)
 - GlideinWMS as scheduler automatically provisioning VM's (boot, delete VM's)
- CMS's quota == 800 cores

AgileInfrastructure Performances

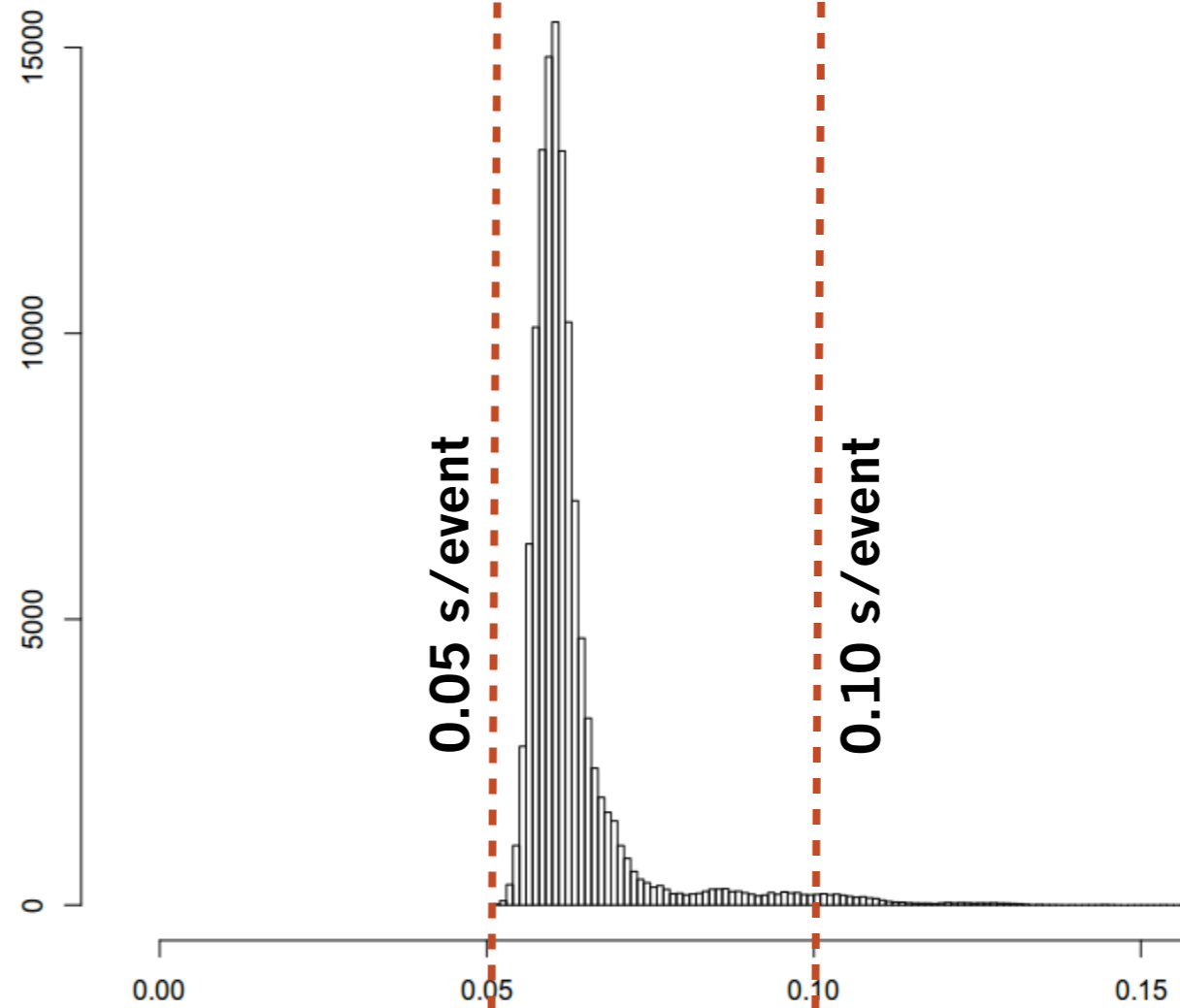
T2_CH_CERN_AI: job success rate 99.27%
T2_CH_CERN job success rate 86.13%

Input type: CMSSW_5_3_1
Input DS Patterns: /GenericTTbar/HC-CMSSW_5_3_1_START53_V5-v1/GEN-SIM-RECO
Ganga Job Template: glidein_test_ai.tpl
User code: pf2pat_cfg.py
Option file: empty
Template: glidein cloud AI
Logs

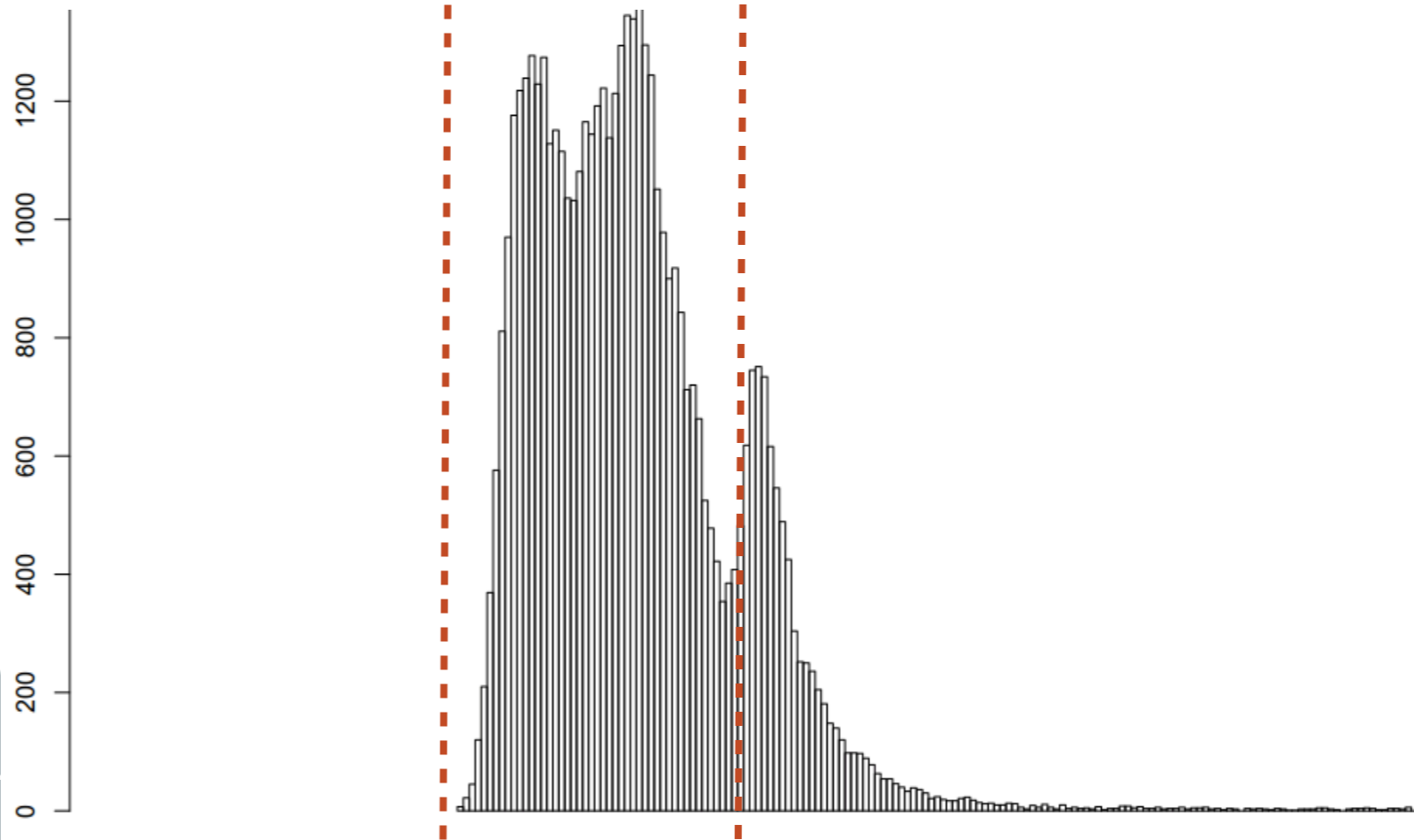


Time per event (s)

T2_CH_CERN_AI



T2_CH_CERN



Issues found

- Cern IaaS is getting more stable, just some minor issues
- Pilot-job framework extended to use a Cloud interface instead of a Grid interface
 - Moving from a model where we “conquer” resources to a model where we just “ask”
 - Policies must be adapted accordingly
 - Fair share in the cloud?
- Not all EC2 are equally implemented (e.g.: OpenStack vs AWS EC2 API's)
 - Need to converge on common functionalities/ implementations

Contextualization

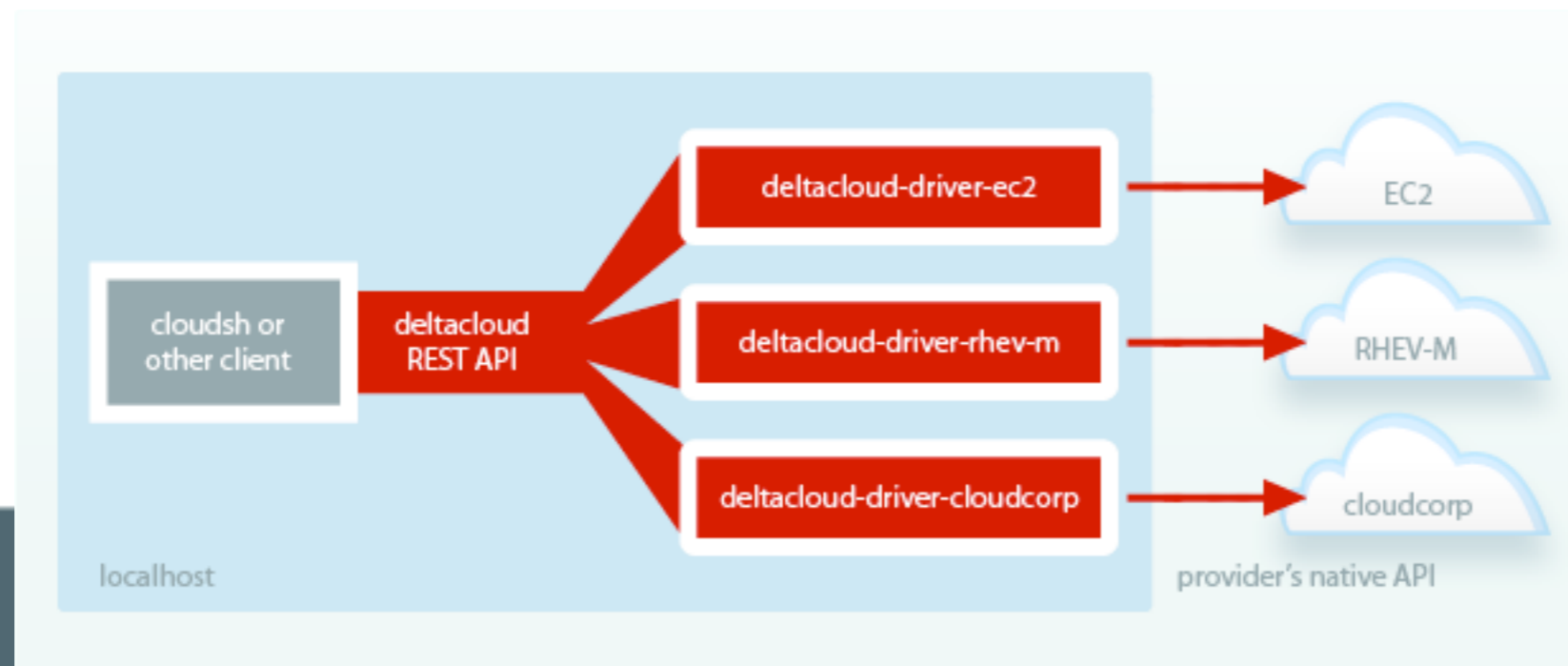
- Today CMS is using a golden image approach
 - Software/configuration are already in the image
 - Only glidein related parameters are dynamic
- Moving to a dynamic contextualization:
 - Software already pre-installed in the image
 - Share data to the VM and customize it at instantiation time (user_data)
 - for both: site and users
 - No image modification required
- First tests done with CernVM based on AMI

CloudInit

- Outside world is focusing on **CloudInit** to contextualize VM
 - Ubuntu project, portable in ~every OS
 - 1 Simple plug-in for every module to be configured
 - Very simple key-value input configuration
- **IT-SDC-OL developed following modules:**
 - condor, ganglia, cvmfs
 - *glidein bootstrap* (prototype)
- **Documentation:** <https://twiki.cern.ch/twiki/bin/view/LCG/CloudInit>
- Code shared with CernVM team

DeltaCloud

- Unique interface to interact with a wide range of cloud interfaces (Eucalyptus, OpenStack, Rackspace, ...)
- Exposes 3 different APIs: DeltaCloud, EC2, DMTF CIMI
- Status:
 - first installation done
 - functional evaluation looks good
 - able to interact with AI nova API
 - able to instantiate VM's and submit jobs through Condor



Summary

- CMS is getting ready to use private clouds
 - CERN private cloud and HLT are an optimal possibility to explore
 - Continue testing CERN private cloud (Agile IaaS)
 - Looking forward to increase the scale next autumn
- Trying to move to use well know tools ('de facto' standards)
 - Cloud-init for contextualization
 - EC2 as interface
- But: there is no general policy defined about which tools/interfaces a Grid site should possibly provide