The ALICE software release validation cluster

Dario Berzano CERN (Genève, CH) • **Mikolaj Krzewicki** FIAS (Frankfurt, DE)

dario.berzano@cern.ch • mkrzewic@compeng.uni-frankfurt.de

Goals

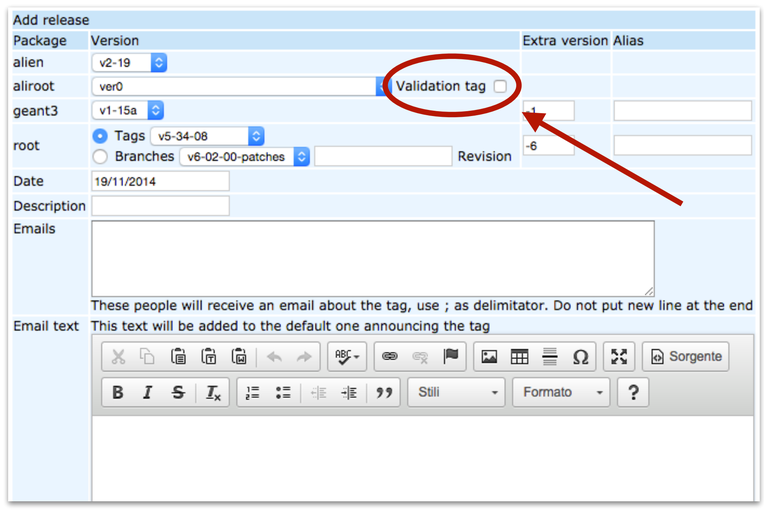
- Start and review results easily: **integrated** with current workflow
- Fully reproducible **everywhere** in a consistent environment

Steering components

- **Workflow manager:** launches all release validation batch jobs and dispatches head node VM
- **Release Validation daemon:** integration with build process, fetch and **publish** AliRoot binaries, **submit** jobs and send **notifications**
- Source and RPMs: github.com/dberzano/cern-alice-relval

Software release validation in one click

Complex process fully transparent to the librarian: one click procedure



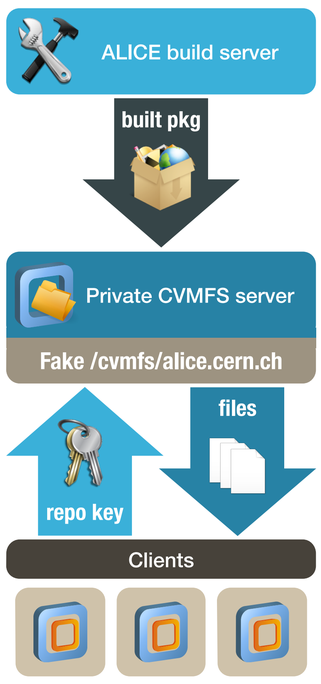
- Dead easy: just **check a box**
- Fresh cluster of VMs launched transparently
- Validation **config** distributed with the software

Key technologies: the CernVM Ecosystem

CernVM provides a full toolchain for publishing software, deploying and scaling clusters of VMs and ensuring the runtime environment consistency

Private CVMFS server

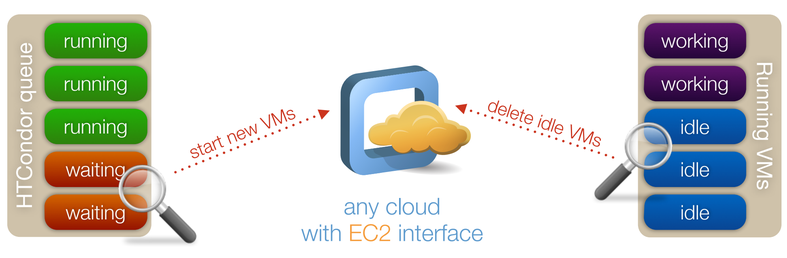
Validation tags and dependencies are published on a private CVMFS server



- Do not litter official CVMFS server
- Private server pretends to be **alice.cern.ch**
- Clients mount on **/cvmfs/alice.cern.ch**: prod environment
- Special **key** injected to clients for auth

elastiq: adaptive VM deployment

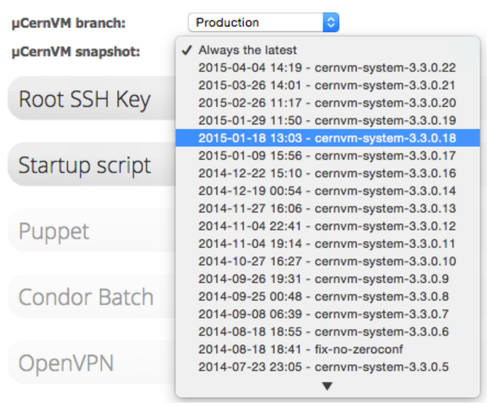
Worker nodes are VMs deployed on demand by monitoring jobs queue



- Validation jobs on HTCondor
- Initially zero nodes: **elastiq** monitors queue and starts/deletes VM workers
- **EC2 API:** works on any cloud
- Source, RPMs, debs, manual (very easy): github.com/dberzano/elastiq

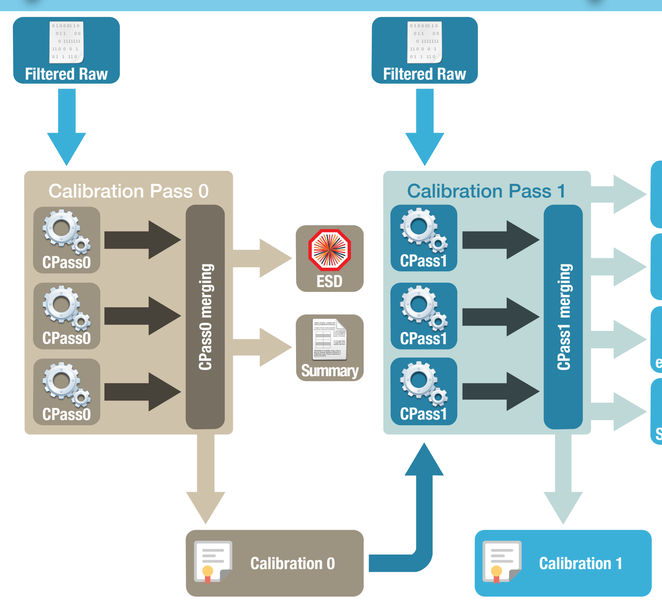
Consistency with CernVM

Every validation starts a fresh cluster of VMs with a pinned CernVM snapshot



- Cluster environment 100% re-runnable on any cloud, at any **future** time
- **Configure** cluster of CernVMs via web: cernvm-online.cern.ch

Batch jobs workflow and dependencies



- Full reconstruction workflow: batch jobs with **dependencies**
- Steering program generates deps automatically on input files
- **elastiq:** num of running VMs varies during validation (merging)

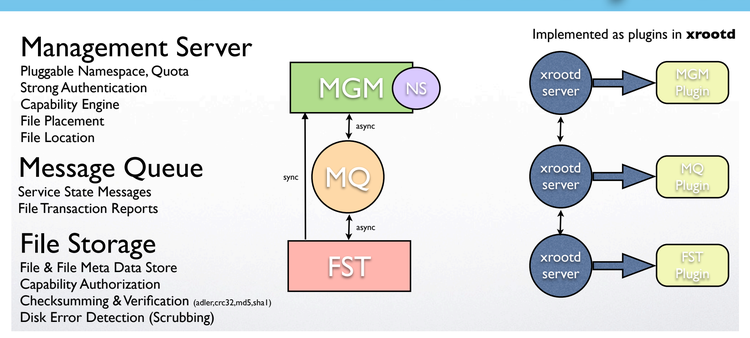
Makeflow

```
part1 part2 part3: input.data split.py
        ./split.py input.data

out1: part1 mysim.exe
        ./mysim.exe part1 >out1

out2: part2 mysim.exe
        ./mysim.exe part2 >out2

out3: part3 mysim.exe
        ./mysim.exe part3 >out3

result: out1 out2 out3 join.py
        ./join.py out1 out2 out3 > result
```

ccl.cse.nd.edu/software/makeflow

- Submits jobs in the correct order on HTCondor
- Dependencies manifest inspired by **Makefiles**

Distributed filesystem: EOS



EOS is from CERN IT-DSS
eos.cern.ch

- EOS FUSE client is in CernVM: **mounted** POSIX filesystem
- Reference raw data: **1.1 TB** fully read from mount point
- Results written on mounted EOS: **80-250 GB per validation**

Review results with a web browser

A notification email is sent to the reviewers with a link to the results