

# Opportunistic and HPC computing resources

*A. Tsaregorodtsev,  
CPPM-IN2P3-CNRS, Marseille,  
23 May, DIRAC User Workshop,  
CC/IN2P3, Lyon*



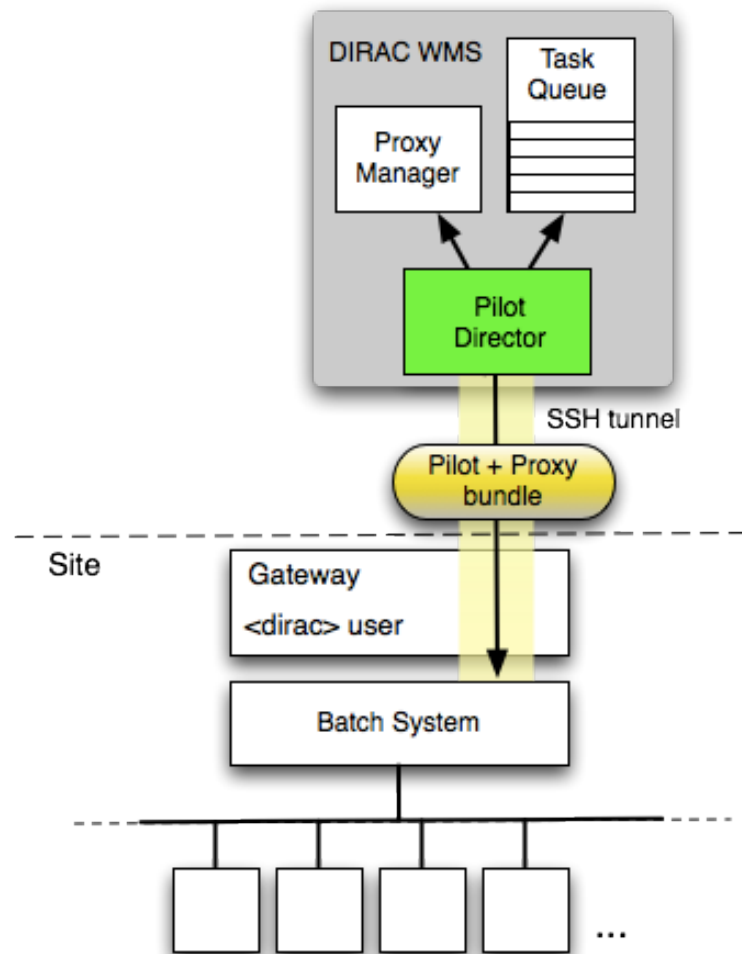
- ▶ Opportunistic resources
- ▶ SSH Computing Elements
- ▶ Boinc case
- ▶ HPC case
- ▶ Conclusions

- ▶ Opportunistic resources are all those which are not ensured by formal agreement with some grid infrastructure and pledges from resource providers
  
- ▶ It still can be grid resources at sites which do not make pledges to a community but do not object using it with low priority
  - ▶ When no high priority payloads are available
  
- ▶ Communities should be ready to use those grid resources
  - ▶ Resources should be configured for the community
  - ▶ Make effort to be not very specific in the payload requirements
  - ▶ Ensure small amount of pilots constantly waiting at opportunistic sites if there are suitable user payloads.

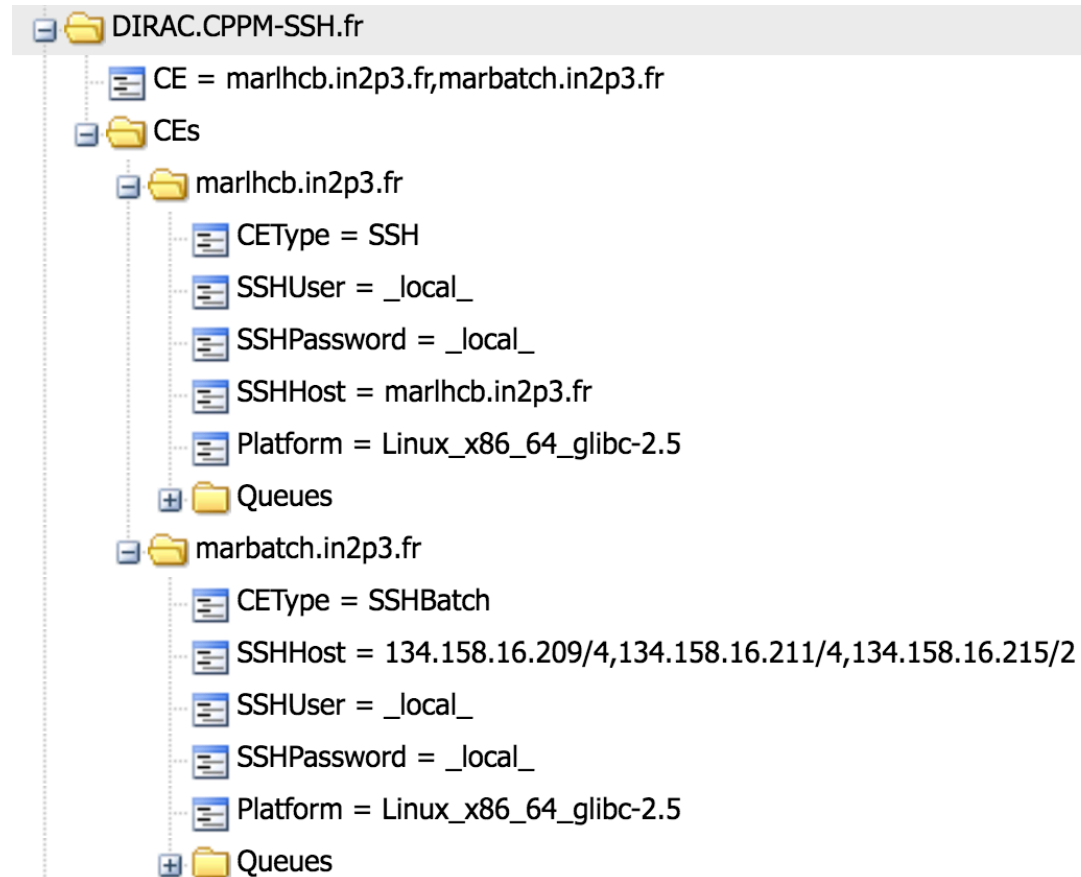
- ▶ DIRAC can offer the users of the common infrastructure, e.g. EGI, the possibility to connect their own computing resource with
  - ▶ no middleware
  - ▶ no BDII entry
  - ▶ no accounting reports

- ▶ All the computing resources where DIRAC pilots can be deployed can offer opportunities to user communities
- ▶ Pilot deployment is mimicking actions of a user running one's payloads
  - ▶ On a local host(s)
  - ▶ On a local batch system
  - ▶ User logs in via SSH and makes it payload running locally or via a “qsub” or equivalent command

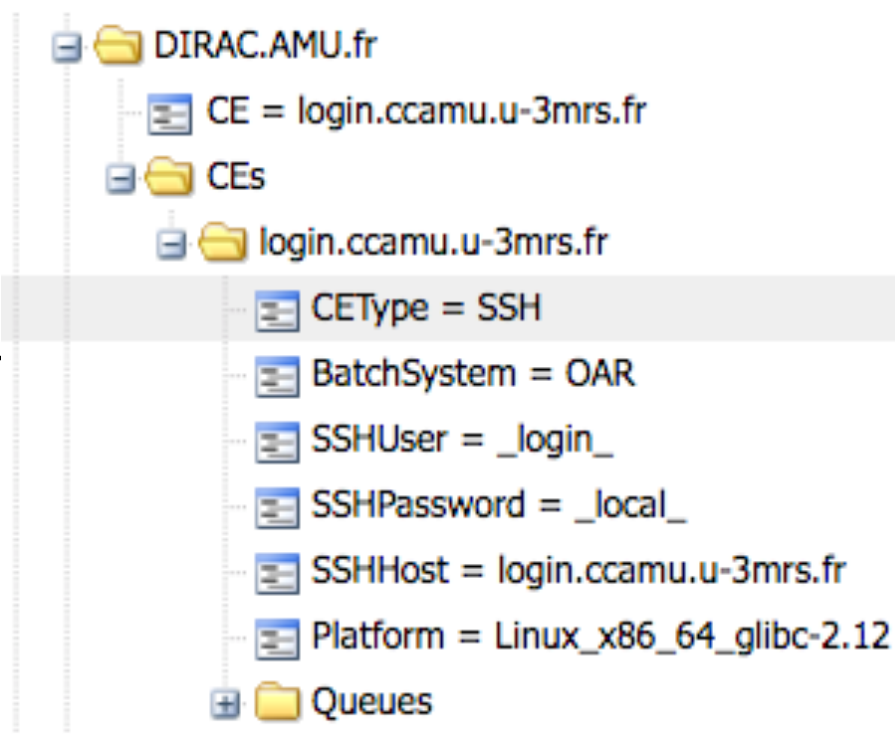
- ▶ **Off-site Pilot Director**
  - ▶ Site must only define a dedicated local user account
  - ▶ The payload submission through an SSH tunnel
  
- ▶ **The site can be:**
  - ▶ a single computer or several computers without any batch system
  - ▶ a computing cluster with a batch system
  
- ▶ **Pilots are sent as an executable self-extracting archive with the pilot proxy bundled in**
  
- ▶ **The user payload is executed with the owner credentials**
  - ▶ No security compromises with respect to external services



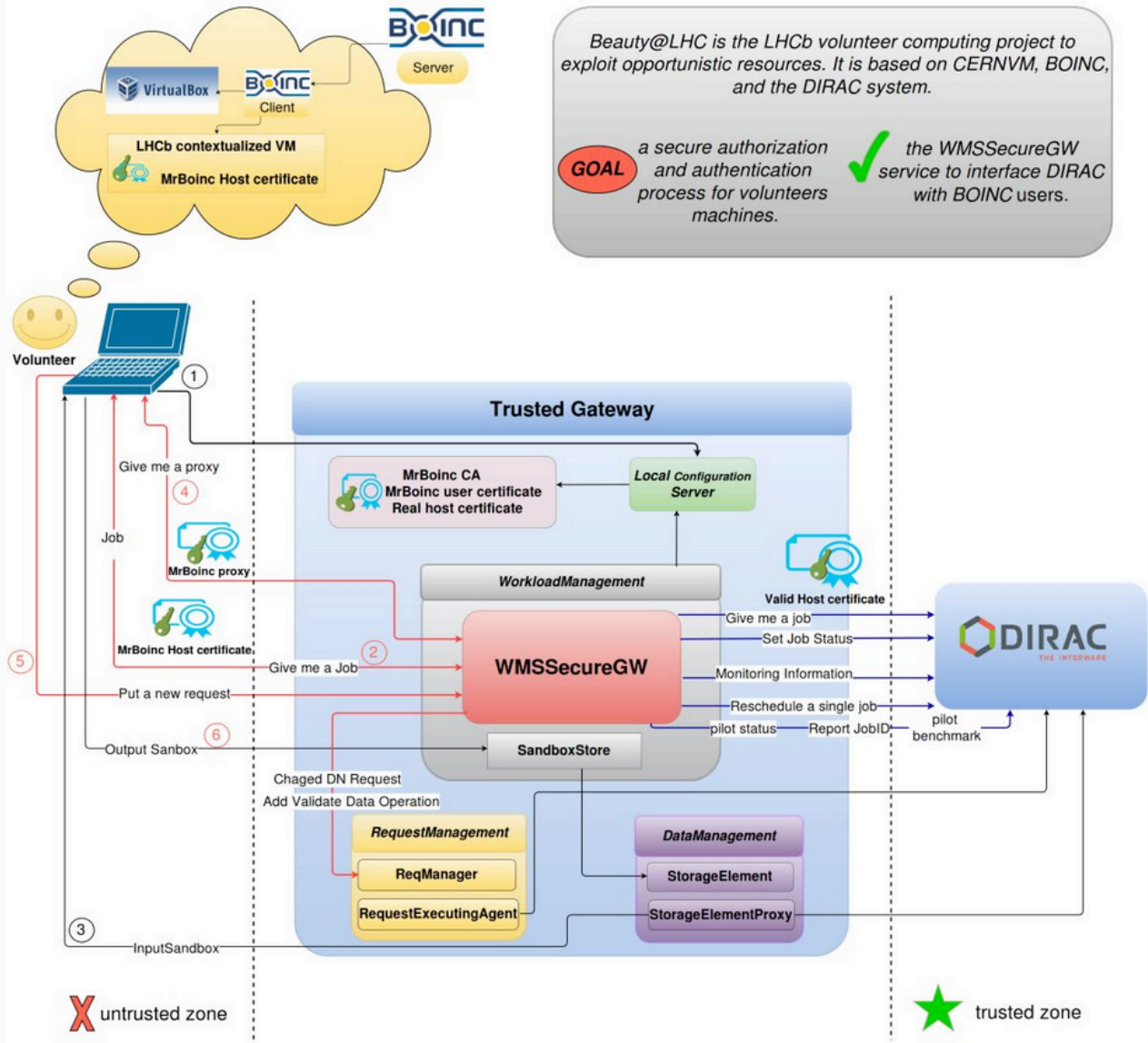
- ▶ SSH CE simplest case:
  - ▶ One host CE with one job slot
- ▶ SSHBatch CE
  - ▶ Several hosts form a CE
    - ▶ Same SSH login details
    - ▶ Number of job slots per host can be specified
- ▶ Pilots are sent as an executable self-extracting archive with the pilot proxy bundled in



- ▶ SSH login to the cluster interactive host
  - ▶ Copy several tools, e.g. BatchSystem plugin at the first time
- ▶ Submit pilots to the local cluster using a relevant BatchSystem plugin
  - ▶ Condor, GE, LSF, Torque
  - ▶ SLURM, OAR
- ▶ Pilots are sent as an executable self-extracting archive with the pilot proxy bundled in







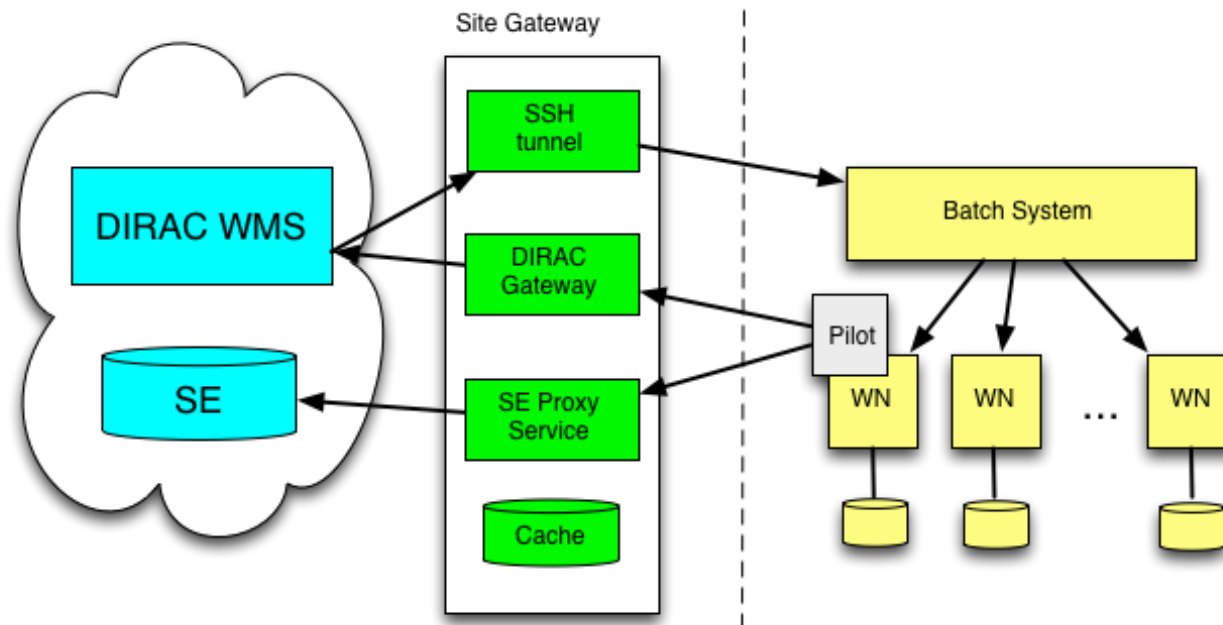
- ▶ Most of the building blocks are standard DIRAC components
- ▶ Specific Boinc components are extracted into the DIRAC extension package BoincDIRAC
  - ▶ <https://github.com/DIRACGrid/BoincDIRAC>
  - ▶ Mostly containing code for the WMSecureGW service
    - ▶ Redirection of all the DIRAC WMS interfaces needed by pilots
- ▶ BoincDIRAC is LHCb independent
  - ▶ Needs a developer to accomplish it as a general purpose solution
    - ▶ Started by Wenjing Wu, IHEP

- ▶ HPC clusters are usually not part of any distributed infrastructure
  - ▶ We have to agree that the HPC site delegates control of user jobs to the central DIRAC service
- ▶ We can easily use HPC centers if:
  - ▶ There is SSH login to the center interactive interface
  - ▶ Worker nodes have world-wide outbound connectivity
  - ▶ CVMFS available on worker nodes to deploy application software
  - ▶ The payloads have no requirements to use HPC special features, e.g. MPI, GPUs, etc

- ▶ SSH login is usually not a problem
- ▶ There are cases where there is rather an GSISSH login is required (LRZ)
  - ▶ This is controlled by `SSHType` parameter:  
`ssh` (default) or `gsissh`
- ▶ There are cases where one should login first into some local host and from there to the batch system interactive host
  - ▶ This is possible with using `SSHTunnel` parameter to specify the intermediate host

- ▶ DIRAC requires output-bound connectivity of worker nodes
  - ▶ To interact with central services
  - ▶ To download input and upload resulting data
- ▶ If no such connectivity, then more collaboration with the HPC site is necessary.
- ▶ DIRAC solution can be to install on the site gateway host
  - ▶ Gateway service
  - ▶ StorageElementProxy service

- ▶ Pilot communicates with the DIRAC central services through the Gateway proxy service
- ▶ Download Input Data and upload output data to the target SE through the SE proxy



- ▶ This case is similar to the Boinc case with no security related complications
- ▶ If StorageElement Proxy is a bottleneck, then local buffer SE can be used with the usual machinery for the asynchronous data upload with failure management:
  - ▶ ReqManager + RequestExecutingAgent

- ▶ If no way to strike a deal to install CVMFS on the HPC worker nodes, then other solutions that existed in pre-CVMFS era can be employed depending on site:
  - ▶ Preinstalling in a shared file system
  - ▶ Downloading for each job
  - ▶ ...



- ▶ There are many opportunities to grab non-dedicated resources in large grid infrastructures. User communities should be prepared for that
- ▶ In many cases SSH Computing Element solution can help to connect privately negotiated resources to the pool of dedicated resource managed by DIRAC
- ▶ BOINC solution can bring potentially a lot of resources but needs a special care about security in the untrusted environments
- ▶ HPC clusters may have many constraints but in almost all the cases a workaround solution can be found by putting together and properly configuring existing DIRAC components