Pilot Job Frameworks Review

- GDB working group mandated by WLCG MB on Jan. 22, 2008

- Mission
  - Review security issues in the pilot job framework of each experiment
    - Pilot jobs are taken as multi-user in this context
  - Define a minimum set of security requirements
  - Advise on improvements
    - Per framework or common to all
  - Report to GDB and MB
    - Time frame is a few months
- Members
  - ALICE: Predrag Buncic
  - ATLAS: Torre Wenaus
  - CMS: Igor Sfiligoi
  - LHCb: Andrei Tsaregorodtsev
  - WLCG: Maarten Litmaath (chair)
  - EGEE: David Groep
  - FNAL: Eileen Berman
  - GridPP: Mingchao Ma
  - OSG: Mine Altunay

Status

- 6 phone conferences held
- Discussion on mailing list
- Each experiment is to provide a document about their system
  - LHCb were the first
  - CMS were next
  - ALICE and ATLAS not ready yet
    - ATLAS have a lot of documentation on PanDA on their TWiki pages
      - https://twiki.cern.ch/twiki/bin/view/Atlas/PanDA
- A security questionnaire has been discussed
  - Agreement on the relevance/scope of a question was not always evident
  - Each document should provide the answers for an experiment
    - Adequacy judged per experiment, no formal criteria
    - LHCb answers deemed satisfactory, CMS to provide more details later

Questionnaire v1.0 (1/2)

- Describe in a schematic way all components of the system.
  - If a component needs to use IPC to talk to another component for any reason, describe what kind of authentication, authorization, integrity and/or privacy mechanisms are in place. If configurable, specify the typical, minimum and maximum protection you can get.

- Describe how user proxies are handled from the moment a user submits a task to the central task queue to the moment that the user task runs on a WN, through any intermediate storage.

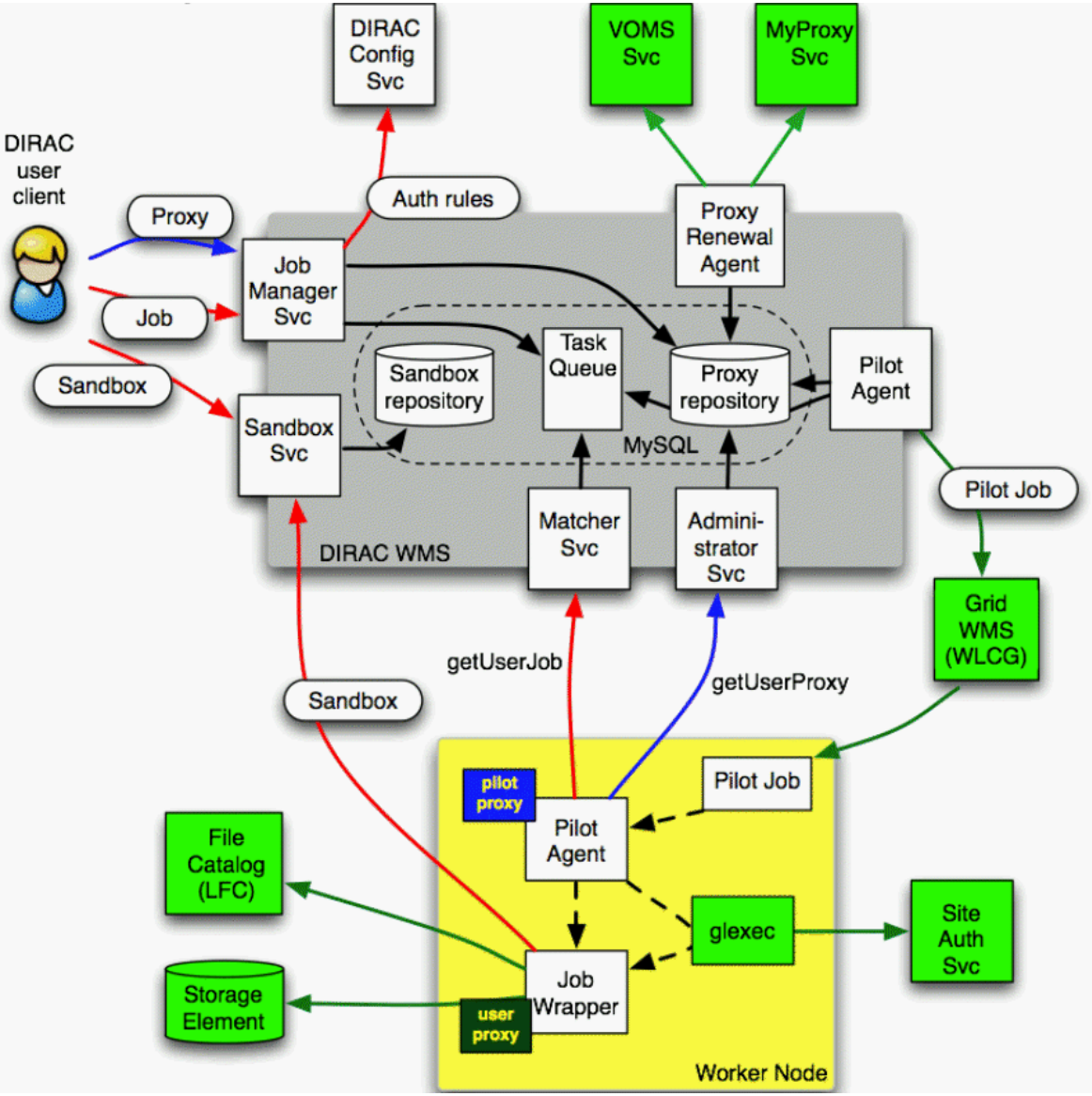
- What happens around the identity change on the WN, e.g. how is each task sandboxed and to what extent?
- How can running processes be accounted to the correct user?
- How is a task spawned on the WN and how is it destroyed?

- How can a site be blocked?

Questionnaire v1.0 (2/2)

- What site security processes are applied to the machine(s) running the WMS? [Here WMS means the VO WMS, not the gLite WMS.]
  - Who is allowed access to the machine(s) on which the service(s) run, and how do they obtain access?
  - How are authorized individuals authenticated on the machine(s)?
  - What is the process for keeping the service(s) and OS patched and up-to-date, especially with respect to security patches?
  - Do you have an identified security contact?
  - Describe the incident response plan to deal with security incidents and reports of unauthorized use?
  - What services (in general) run on the machine(s) that offer the WMS service?
  - What processes exist to maintain audit logs (e.g. for use during an incident)?
  - What monitoring exists on the machine(s) to aid detection of security incidents or unauthorized use?
- Can you limit the users that can submit jobs to the VO WMS? How?

LHCb DIRAC WMS



DIRAC WMS workflow

- Workload preparation on UI (input files, JDL)
- Workload submission to DIRAC WMS
- Requirements and priority analysis → insertion into Central Task Queue
- Grid submission of pilot job with original requirements
  - Using special credential (VOMS role)
- Start of pilot on WN
  - Install Job Agent, check WN capacity and environment
- Request highest-priority matching workload from Task Queue
  - Download associated limited user proxy
  - Execute job wrapper through glexec
    - Preparation of input data, installation of missing LHCb software as needed
    - Parallel execution of user workload and watchdog process
    - Upload output data
  - Cleanup of user workload and proxy
  - Report resource consumption to DIRAC accounting system
- Get another workload if enough remaining CPU time

LHCb document

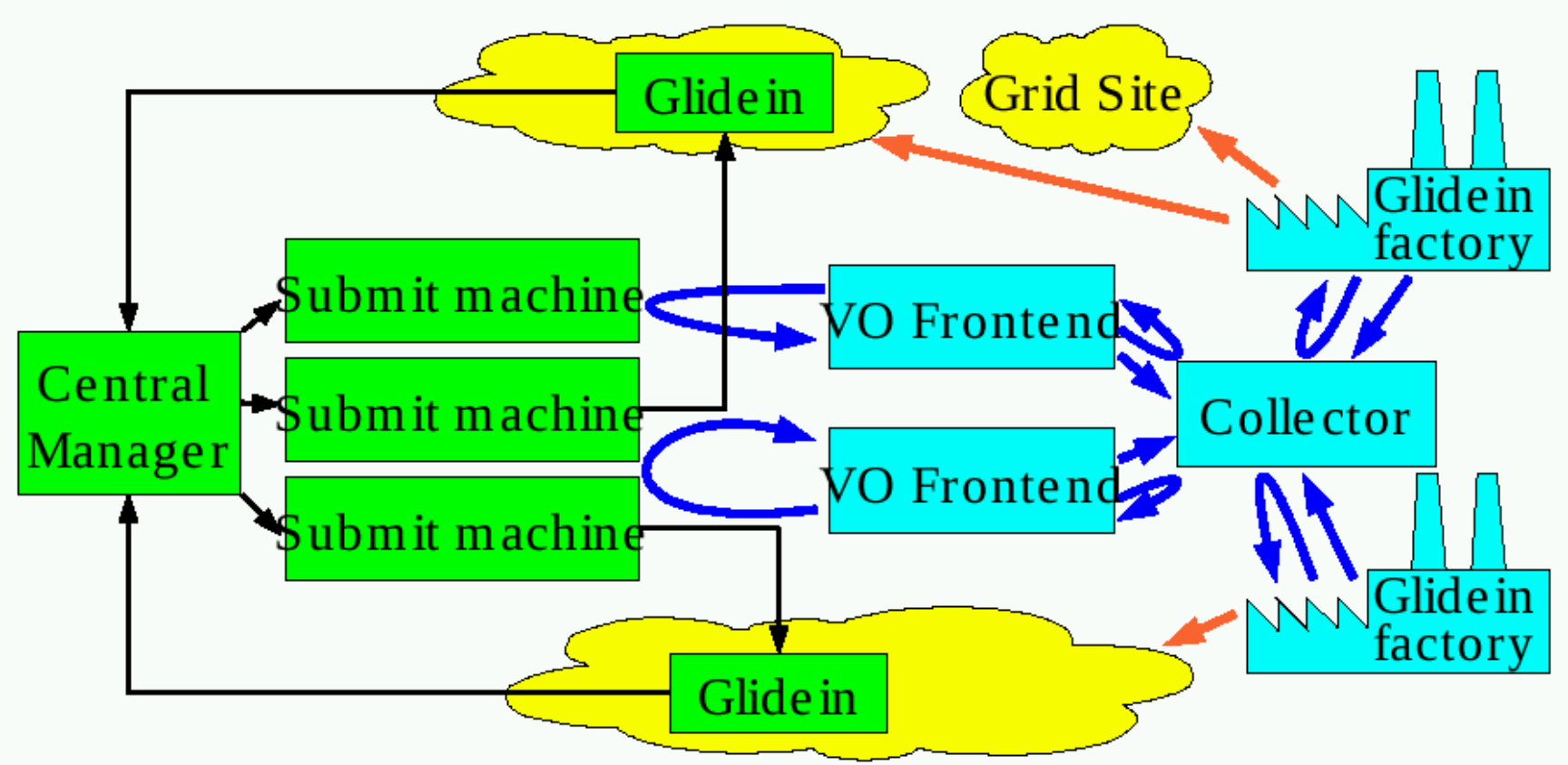
- "Workload Management with Pilot Agents in DIRAC"
  - http://indico.cern.ch/materialDisplay.py?sessionId=4&materialId=0&confId=20230

1. Overview
2. Workflow in the DIRAC WMS
3. User authentication and authorization
4. Brief DISET overview [DISET = DIRAC SEcure Transport]
5. Job submission to the DIRAC WMS
6. Proxy handling in the DIRAC WMS
7. Pilot Job
8. Running user job
9. Job monitoring, user interaction with a job
10. User job accounting
11. Pilot Job Questionnaire
12. References

DIRAC WMS remarks

- DISET provides secure communication channels
- DIRAC WMS restricts users and can block them
- It can block sites
  - A mask is applied to pilot job submission and to workload matching
- It is trusted by myproxy.cern.ch
- It is hosted by the CERN IT department according to IT regulations
  - Access restrictions, security updates, incident response procedures
- A pilot job cannot renew its own proxy
  - It can only download limited proxies for user workloads

CMS glideinWMS schematics



CMS glideinWMS workflow

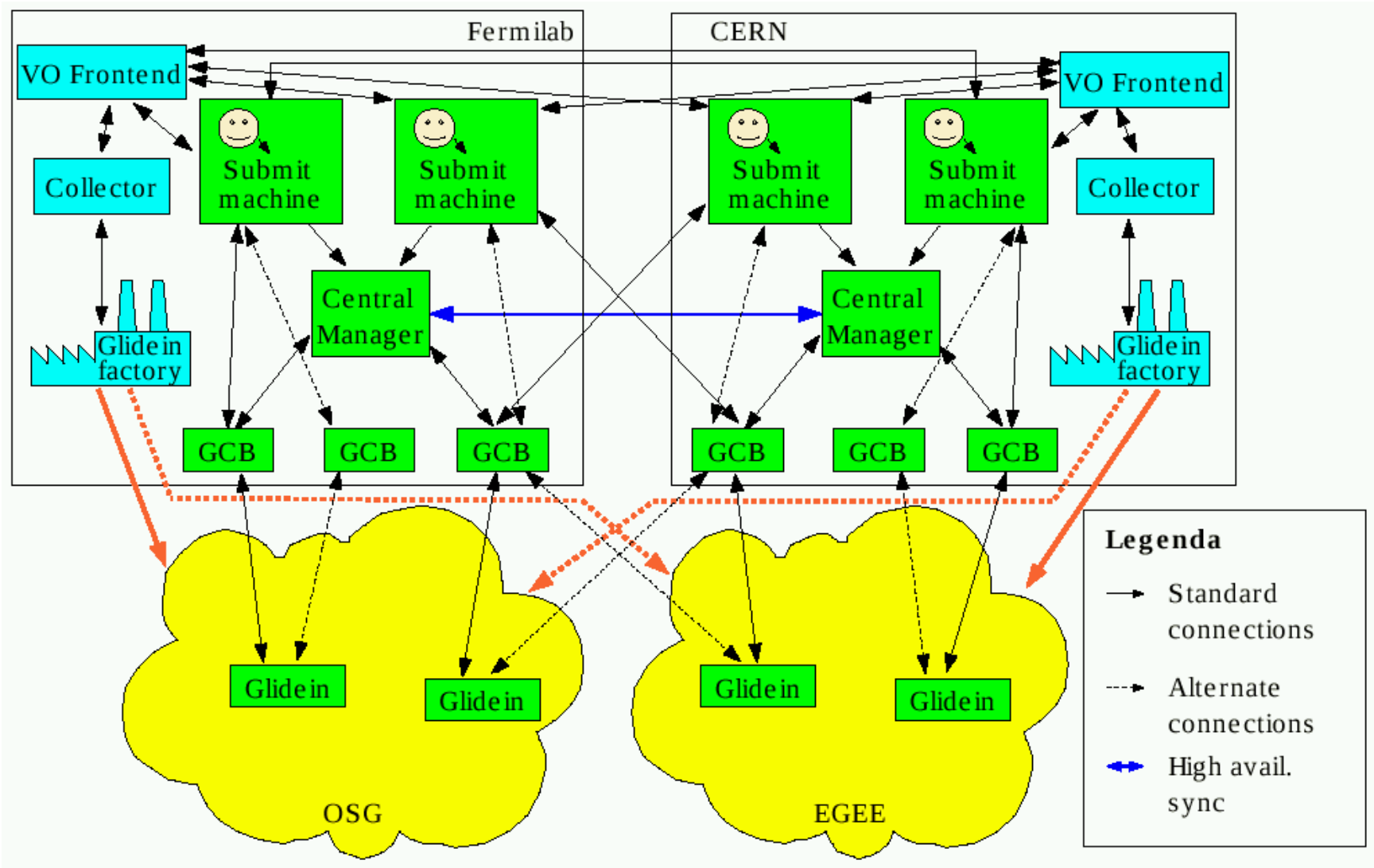
- System relies on standard Condor components and communication

- User jobs are submitted to a set of condor_schedd on submit machines
  - Waiting jobs are advertized in a Central Manager
- Glidein factories submit pilot jobs a.k.a. glideins
- VO frontends regulate the number of glideins to be submitted
  - Based on the number of jobs waiting in the condor_schedd queues

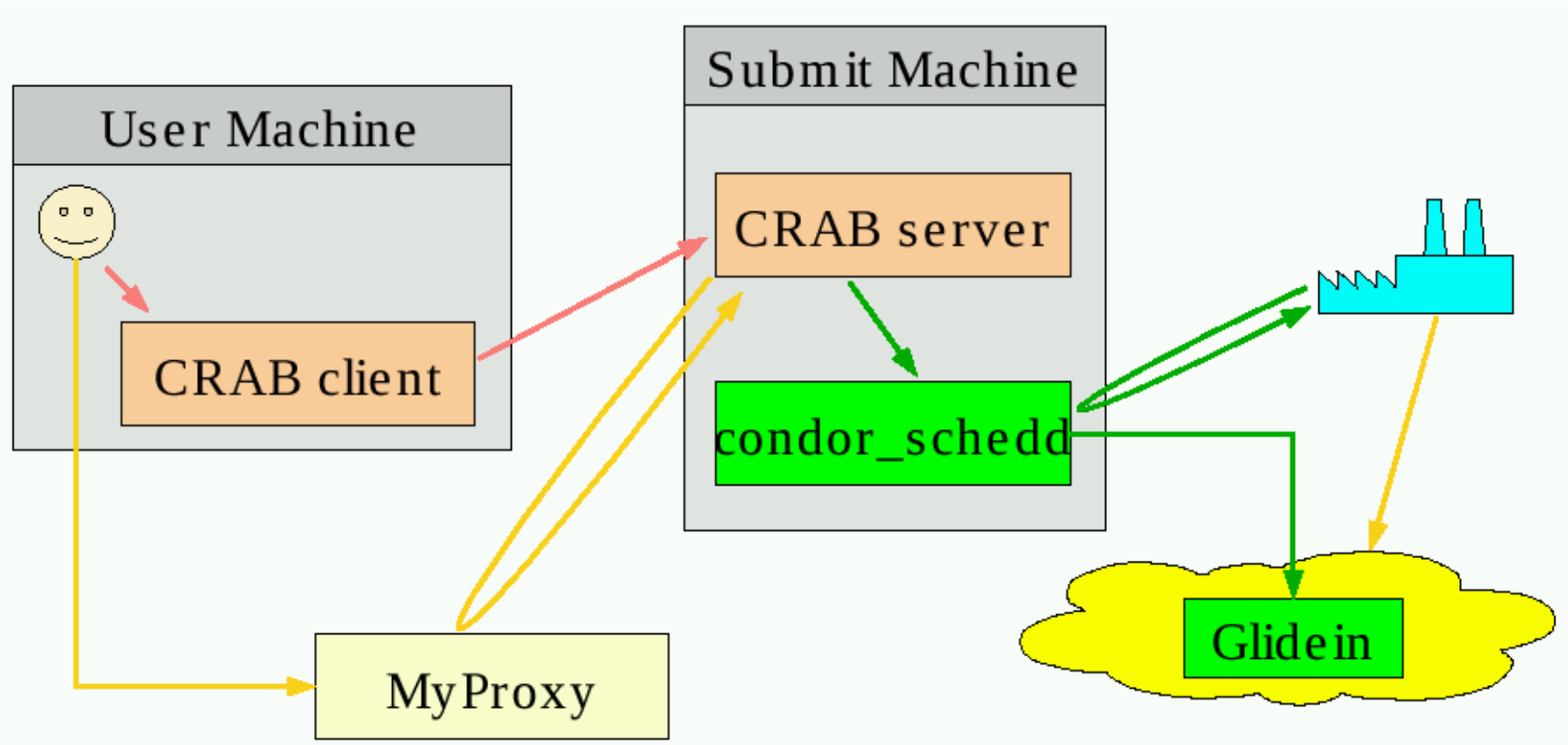
- A condor_collector is used as a dashboard for message exchange.
- Generic Connection Brokers can provide connectivity across firewalls and in NAT environments

- A glidein contacts the Central Manager for a job matching the WN
- The user job is spawned through glexec and cleaned up afterwards
  - The glidein may then wait a while for another job or exit

Envisioned CMS production system



Envisioned CMS analysis instance



CMS document

- "Workload management with glideinWMS"
  - http://indico.cern.ch/materialDisplay.py?sessionId=4&materialId=0&confId=20230

1. Introduction
2. Structural overview
   1. The Condor pool
   2. Glidein handling
   3. Working in a firewalled world
   4. Credentials handling
3. Deployment scenarios by USCMS
   1. Current prototype production installation
   2. Planned worldwide production installation
   3. Planned analysis installation
4. Conclusions

- Appendix
  - Pilot Job Frameworks questionnaire

CMS glideinWMS remarks

- Condor provides secure communication channels

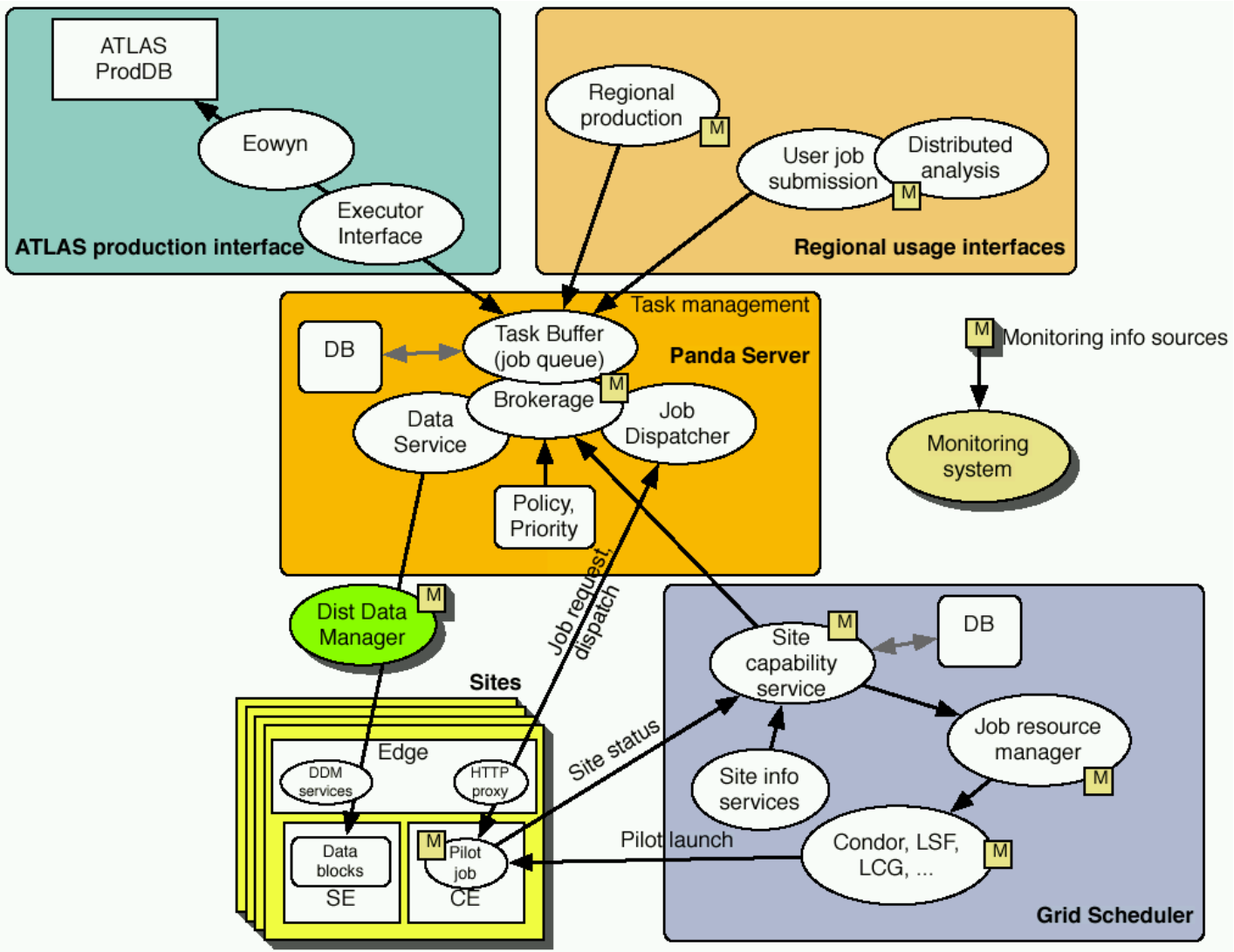
- Both production and analysis instances can restrict/block users
- Sites can be blocked
  - Using IP tables or Condor functionality

- A submit machine is trusted by an associated MyProxy server
- Users will not login onto submit machine, but interact with CRAB server
  - CRAB = CMS Remote Analysis Builder

- Submit machines etc. hosted by T0/T1/T2 sites
  - Need access restrictions, security updates, incident response procedures

- Condor does not yet support limited delegation
  - Accepted as feature request

ATLAS PanDA architecture

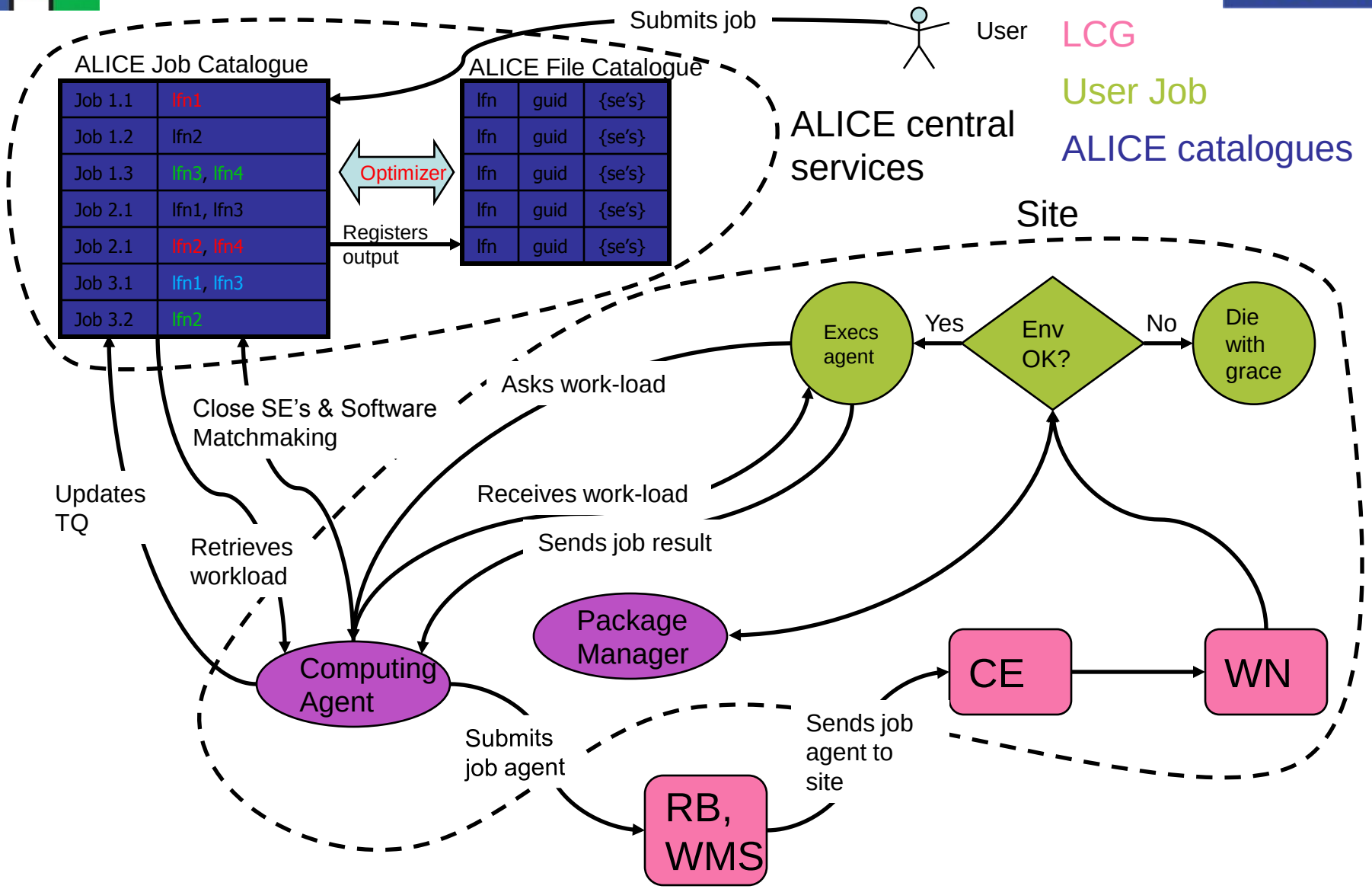


PanDA workflow

- PanDA server receives user job definitions with data requirements
- Jobs are inserted into global work queue
- Brokerage module prioritizes jobs and assigns them to sites
- Required input data is dispatched to the chosen site
  - Interaction with ATLAS Distributed Data Management system
  - Jobs are released to dispatcher when input data has been made available
- Delivery of pilot jobs to WNs managed by independent subsystem
- Pilot jobs contact job dispatcher for work
  - If no appropriate work is available, the pilot may pause or exit

ALICE job submission in LCG

VO-Box
LCG
User Job
ALICE catalogues



ALICE services