Pilot Jobs

Oliver Keeble
CERN
What is a pilot job?

Virtual Organisation

User Job

VO Workload Management System or Job Queue

User Job

Pilot Job

Site Boundary

Grid Computing Service

Site or VO CE or traditional gatekeeper mechanism

LRMS Queue

VO Pilot Job (VO uid)

VO Pilot Job (VO uid)

Worker Node

User Job

Unix uid specific to User

gLexec
The life of a pilot job

Validate node

Any jobs in queue?

- Yes: Fetch user job (and credentials)
- No: Cleanup pilot area

Fetch user job (and credentials)

Execute job

Return output

Cleanup job area

Terminate
Problems Solved

- **Scheduling**
  - Priorities
    - JPWG anyone?
      - Different components currently interpret credentials inconsistently
      - Published information is not 100% reliable
  - Responsiveness

- **Matchmaking**
  - Can analyse environment before pulling payload
Problems Created

• Everyone runs as the same user
  – Accounting and auditing
  – Users can steal each other's data and interfere
  – Any user can run a pilot job
  – One user is (legally) accountable for all jobs
    – Site cannot ban particular users

• Proxies handled via unknown mechanisms

• VO infrastructure required

• VOs decide who runs, not the sites any more

• Pilot jobs which pull in jobs from a variety of users
  break the current Acceptable Use Policy in
  EGEE/WLCG/OSG(?)
gLExec

a thin layer
to change Unix domain credentials
based on grid identity and attribute information

you can think of it as:

• ‘a replacement for the gatekeeper’
• ‘a griddy version of Apache’s suexec’
• ‘a program wrapper around LCAS, LCMAPS or GUMS’

It is a small binary executed by the pilot job
Pilot job contacts its task queue and receives a payload + proxy
Then it executes

```
/opt/glite/sbin/glexec payload
```

with `GLEXEC_SOURCE_PROXY` pointing to the payload's proxy cert

- **The pilot job does not choose the user to switch to, it presents a proxy to the local authz mechanism**
- **Solves all the 'running as one user' problems**
- **Issues**
  - setuid binaries
    - gatekeeper already has this privilege (runs as root)
    - my workstation has 30 suid binaries on it!
  - Need distributed lcas (or GUMS or whatever)
  - Pilot infrastructure has to be modified to use it
Identity Mapping Mode – ‘just like on the CE’
- have the VO query (and by policy honour) all site policies
- actually change uid based on the true user’s grid identity
- enforce per-user isolation and auditing using uids and gids
- requires gLExec to have setuid capability

Non-Privileged Mode – declare only
- have the VO query (and by policy honour) all site policies
- do not actually change uid: no isolation or auditing per user
- the gLExec invocation will be logged, with the user identity
- does not require setuid powers – job keeps running in pilot space

‘Empty Shell’ – do nothing but execute the command…
Example Site Requirements
Enabling Grids for E-sciencE

• The UK(I) does not accept glexec in its present state and will not recommend its deployment.

• Before we will reconsider its deployment we require to see:
  – a satisfactory code inspection wrt security.
  – an AUP with VOs agreed in principle by sites (in general through GDB) and all 4 LHC VOs.
  – a satisfactory analysis of the effect on the supported batch systems and accounting of identity changes.
  – glexec should be simple and safe to configure. Simple so that all sysadmins can understand what they are doing and safe so that misconfigurations do not inadvertently lead to security exposures.
  – The three flavours of glexec should exist as separate binaries and not as one with a configuration switch. Each of these should satisfy all package dependencies. e.g., dependencies should express some requirement for an abstract "glexec" property, which is satisfied by "glexec-null" or "glexec-suid" etc.
The DIRAC Pilot Agent paradigm provides a reliable means to deploy Agents to the Grid

- LCG jobs are Pilot jobs for the DIRAC WMS
  - Actual workload management performed by DIRAC

- Pilot Agents are submitted on demand from the WMS
  - Reserves the resource for immediate use
    - Allows checking of the environment before job scheduling
    - Only bidirectional network traffic
    - Unidirectional connectivity

- Terminates gracefully if no work is available
Overview of the DIRAC WMS

Enabling Grids for E-sciencE

The diagram illustrates the architectural components of the DIRAC Workload Management System (WMS). The core components include:

- **Central Services**
  - Job Receiver
  - Sandbox Services
  - Job Monitoring Service

- **Data Optimizer**
- **Task Queue**
- **Job DB**

- **Interfaces**
  - DIRAC API
  - Production Manager Console

- **Worker Node**
  - Pilot Agent
  - Job Wrapper
  - Application
  - Watchdog

- **Worker Node Subsystems**
  - Application Subsystem
  - Watchdog Subsystem

- **Grid Resources**
  - CE
  - SE

- **Other Components**
  - LCG File Catalog
  - Agent Director
  - Agent Monitor
  - Matcher

The diagram shows the flow of data and interactions between these components, highlighting the structure and functionality of the DIRAC WMS.