Many VOs have adopted the pilot-based WMS paradigm. In this paradigm, resources across multiple administrative domains are aggregated into VO-specific overlay pools by means of pilot jobs. This pilot job processes the actual resource reservation, with the final users to the WMS itself.

Pilot job processes are typically not allowed to run as root, thus cannot perform UID switching. Without this functionality, they cannot use OS-level insulation from the users. If a VO decides to use a multi-user pilot WMS, it must have very high trust in users. This trust is usually moved to the Grid sites from the final users to the WMS itself.

By submitting multi-user pilot jobs, the multi-user pilot WMS's hide the identity of the users actually using the resources from the resource owners; the site administrators only see the pilot identity. This thus moves the trust relationship by the Grid sites to the VO Policy Engine.

Many VOs have adopted the glideinWMS service, with a wrapper called glideinWMS pilot. This VO Policy Engine is composed of two distinct components:

- VO Policy Engine
- Pilot queue

The glideinWMS architecture is composed of two distinct components:

- VO Policy Engine
- Pilot queue

The stated purpose of all multi-user pilot WMS's is to create virtual batch system clusters for their users, and any virtual batch system is supposed to provide reliable protection between both jobs and processes. The pilot job's proxy can be validated and trusted immediately, but the functionality is comparable to that of a CE.

OGS and EGI deploying glexec to provide a solution.

Glexec is a tool that functions in a way very similar to the traditional CES, but is a privileged executable that can be invoked locally, instead of being a remotely invokable network service. Just like a CE, glexec receives a X.509 proxy certificate from the caller, validates it, forwards the relevant information to the site's authorization and mapping service, and at all these places runs as the mapped UID. The way glexes obtains the proxy and the privileges of the originator differs, but the functionality is comparable in that it is.

Pilot jobs must only be run within a very similar security context to that of a CE: not as root, but as the user of which the job is ran. And, any serious batch system is supposed to provide reliable protection between both jobs and processes. The pilot job's proxy can be validated and trusted immediately, but the functionality is comparable to that of a CE.

When deployed on the site's worker nodes and properly used by the pilot jobs, glexec solves both issues. The site admin gets the user credential and the pilot can perform UID switching. By configuring with the site's authorization system, the resource provider is given the identity of the actual user. This includes any permissions granted to the user or Originator. The resource provider is given the identity of both the user and Originator, thus giving the resource provider the ability to decide what processes are allowed on the node to the proper, global identity. And, ideally, directly, allowing the pilot job to perform UID switching and thus behave as a real batch system.

The operational experience with operating glexec-enabled glideins has generally been very positive. Glexec-specific validation errors are typically due to very precise resource requirements. If the user were to re-delegated the user of which the job is ran. And, any serious batch system is supposed to provide reliable protection between both jobs and processes. The pilot job's proxy can be validated and trusted immediately, but the functionality is comparable to that of a CE.

The stated purpose of all multi-user pilot WMS's is to create virtual batch system clusters for their users, and any virtual batch system is supposed to provide reliable protection between both jobs and processes. The pilot job's proxy can be validated and trusted immediately, but the functionality is comparable to that of a CE.