

gLexec  
(within the OSG and Fermilab)  
June 5, 2006

Keith Chadwick



## Background - Late Binding & Glide-Ins

---

Some Virtual Organizations are using grid job glide-ins as a mechanism to accomplish “late binding”.

In particular, CDF is using glide-ins\* on their recently commissioned glide-CAF cluster at Fermilab:

- A job is submitted to a Globus gatekeeper, when the job begins execution, it has guaranteed access to a CPU.
- At this point the so-called “pilot” job calls “home” and brings in a production job for execution.

\* The term glide-in is used generically, and does not necessarily imply the use of Condor.



## Glide-Ins can do Multiple Tasks

---

Nothing restricts the “pilot” job to running only one job - there may be reasons why the “pilot” job might run multiple jobs:

- Start stage-in of input datasets
- While staging input datasets run monte-carlo
- Pause monte-carlo to run analysis against input datasets
- Start stage-out of analysis results
- While staging out, resume monte-carlo

However:

Don Petravick: “Fermilab will view with extreme disfavor any “pilot” job which attempts to subvert the batch system scheduler by running multiple copies of CPU intensive code on systems with multiple CPUs.”



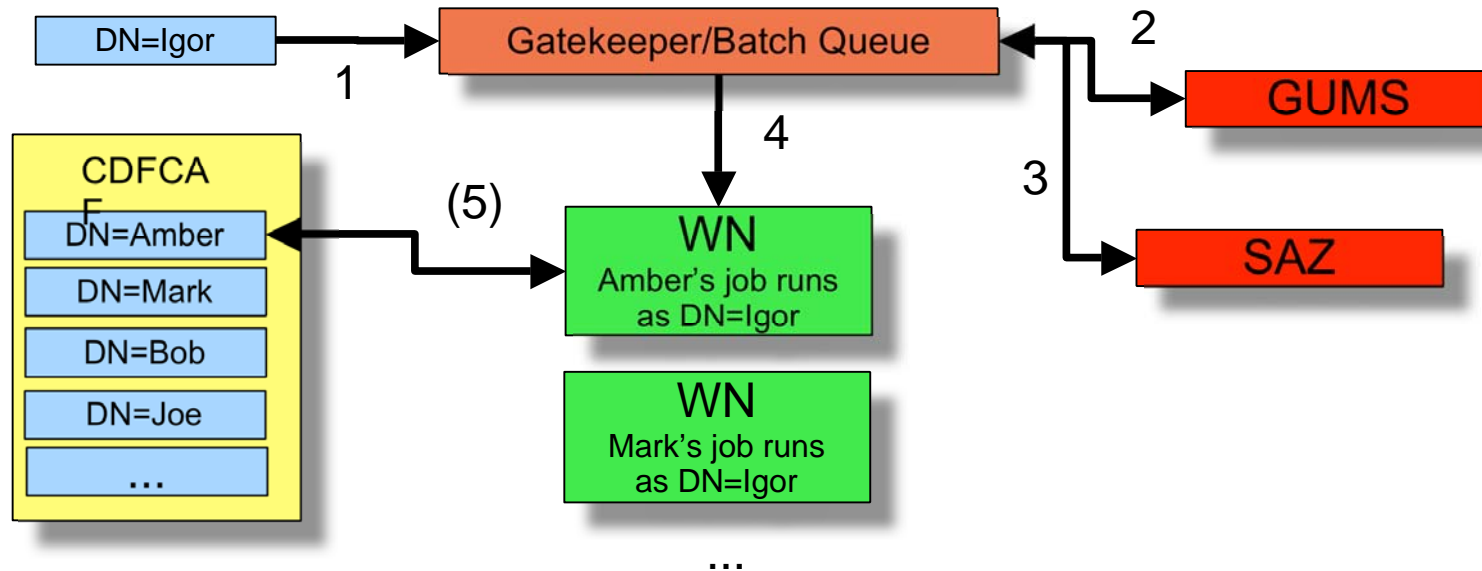
## What is the Issue?

---

Vicky White, the head of the Fermilab Computing Division raised several concerns with the use of grid glide-ins on the CDF glide-caf grid farm during the January 2006 OSG Collaboration Meeting held in Gainesville, FL:

- The “pilot” job does pass through the entire gamut (such as it is) of the Fermilab grid authorization and accounting infrastructure,
- The glide-in jobs run by the “pilot” job bypass the authorization and accounting infrastructure in the Globus gatekeeper
- Furthermore, the real user of the CPU resources was not immediately apparent to management.

## Current modus operandi



GUMS – DN to local UID mapper  
SAZ – site DN authorization



## gLexec - A Solution ?

---

gsEXEC and gLexec, both of which are based on the Apache suEXEC module, have been implemented

A “enhanced” gLexec has been selected to implement a solution to the glide-in problem.

Fermilab and NIKHEF are collaborating on the design and implementation of the enhanced gLexec which will use plug-ins for LCAS/LCMAPS or GUMS/SAZ callouts together with reuse of code from Prima.

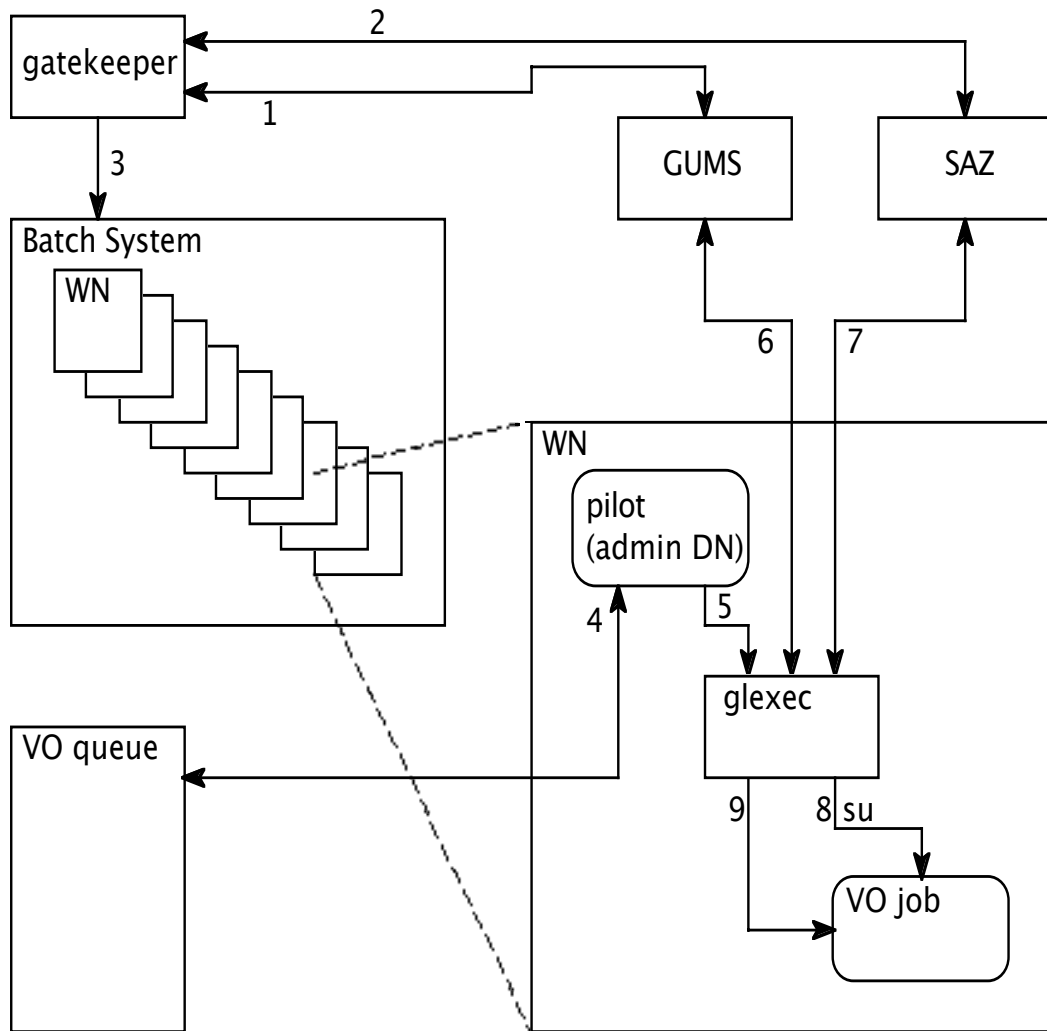
Miron Livny has agreed to implement a “stub” gLexec module in Condor.

# gLexec - Block Diagram

Block Diagram of job flow using gLexec:

Child jobs are vetted through both GUMS and SAZ.

gLexec can also communicate with accounting and auditing modules (if desired).



# Prima Software Components

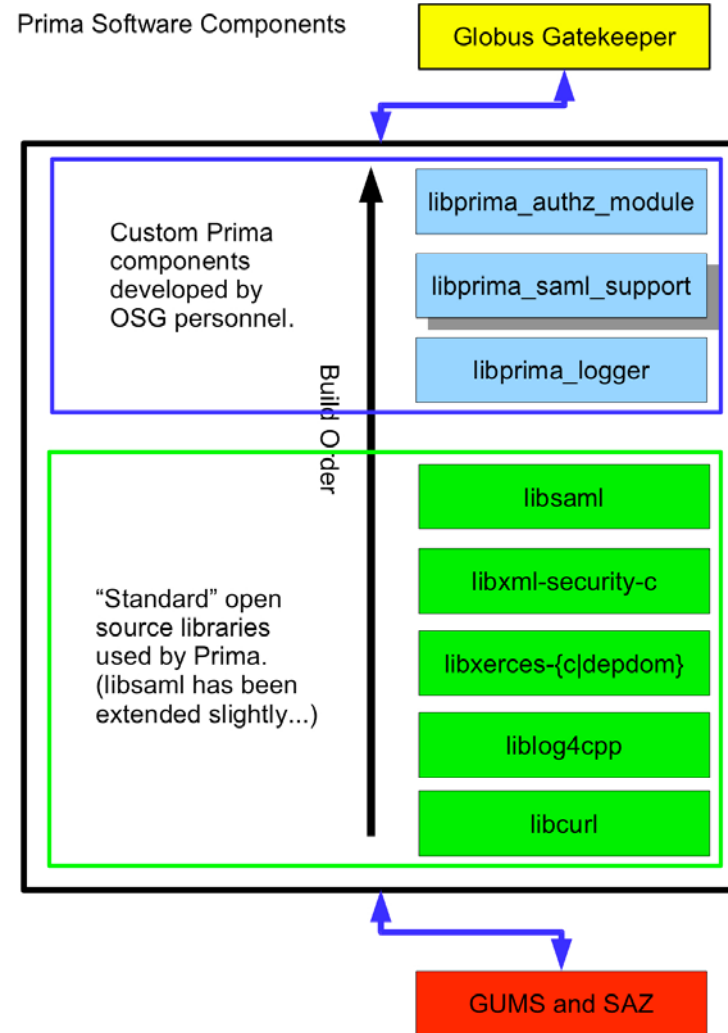
This diagram shows the current state of prima:

The black box surrounding the entire thing is prima. The components in the blue box are those developed originally by Markus Lorch and now maintained by Vikram, et al. OSG people. The green box contains several standard open source libraries that are largely unmodified, except for libsaml which Markus added an extension to help in communicating with the Gatekeeper.

The build order of these components is bottom to top.

Igor modified libprima\_authz and libprima\_saml\_support to remove the mutual dependency on each other, libprima\_authz included the libprima\_saml\_support header file, and a function in libprima\_saml\_support required some parameters from libprima\_authz. He separated these from each other so libprima\_saml\_support doesn't depend on libprima\_authz.

The Gatekeeper uses the authz callout to interface to the prima\_saml\_support shared object which uses libsaml to format the request, and the other green components to contact the GUMS server.



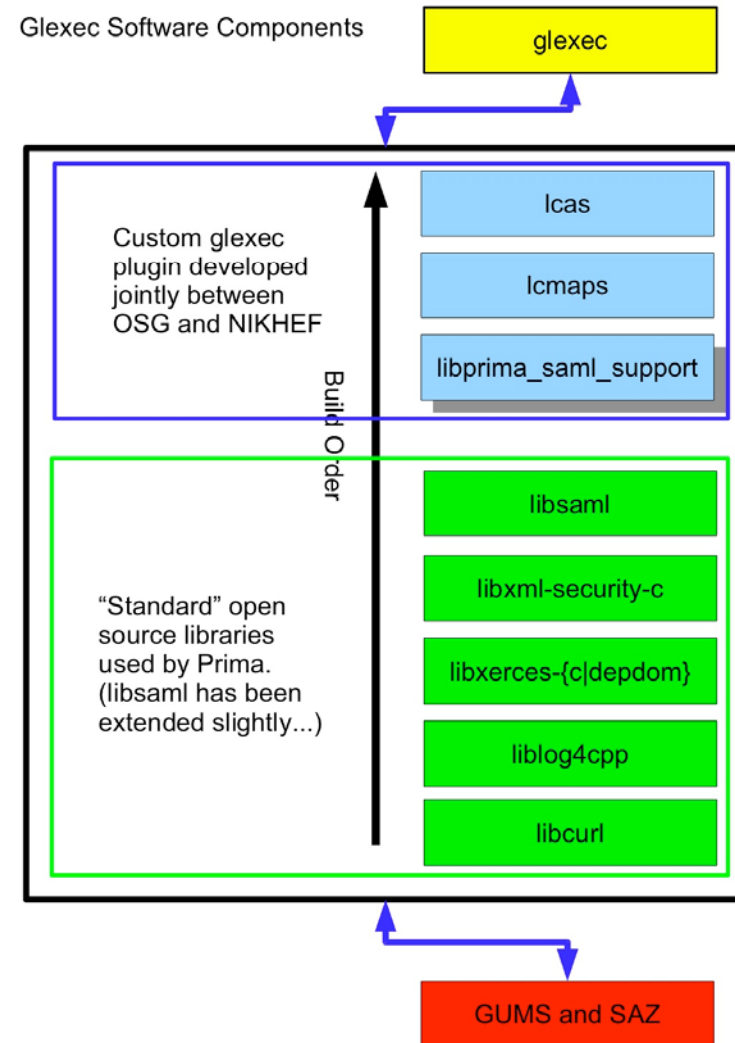


The diagram shows the current state of the gLexec interface to GUMS and/or SAZ:

The libprima\_authz component has been replaced by lcas and lcms plugins (developed by NIKHEF) which will pass the parameters to libprima\_saml\_support which will then function as before.

After Igor's changes, in the globus gatekeeper scenario, Vikram can compile everything up to and including libprima\_saml\_support, but the Makefile for the libprima\_authz is missing some things which he should have done either yesterday or today.

The goal is to make sure we can use the same code base and, ergo, not have maintain 2 sets of code - one for prima and one for glexec.





# Fermilab - Draft Pilot Job Definition

---

## Draft Pilot Job Definition:

A Pilot Job is defined as any job where a Job Manager submits a job request to a grid batch system, but the application that is executed on the Worker Nodes was created by, and the input and output data is owned by, another user.

## Draft Pilot Job Requirement:

A Pilot Job must correctly map the executed application and the input and output data to the actual owner of the application.



# Fermilab - Draft Pilot Job Policy

---

Draft Pilot Job Policy:

Any job that is discovered to be a Pilot Job which :

- Does not correctly perform the appropriate user mapping using the Site supplied utility (i.e., glexec),

or

- Lacks a pre-agreed Trust relationship with Fermilab approved by the CSEXEC

may be terminated immediately.

In addition, the DN of the pilot may be placed on the Site Black List until the situation has been rectified. Furthermore any existing trust relationship with that VO may be reexamined.



# Fermilab - Draft Trust Relationships Policy

---

Draft Trust Relationships Policy:

Glide-in operation at Fermilab is allowed only by Virtual Organizations that have a valid Trust relationship with Fermilab.

Trust relationships are granted and revoked by the Fermilab Computer Security Executive.

Virtual Organizations hosted at Fermilab (CDF, CMS, D0, MINOS, LQCD, etc) have an initial valid grandfathered trust relationship, based on the established basis for management and operational security controls that exists within Fermilab.

The granting of Trust relationships will be based on processes which have yet to defined.

It is a goal of the OSG that this process is not intended to be heavyweight or overly burdensome.



# Fermilab - Site and VO Trust Model

---

In the process of thinking about glide-ins, our thinking has morphed of a more general discussion of:

- "why a Site should trust a VO".
- "why a VO should trust a Site".

These questions are also foundational for the larger question of "Trust" across the grid.

At the present, such trust relationships only exist on the basis of, for lack of a better word, "small world" principles.

The veracity and completeness of your statements, your responsiveness, an understanding of the technology you are using, and so forth.

Companies like eBay, for example, make this more scalable, by evaluating the trust with a varieties of policies and technologies.

This is an interesting area for further discussion and exploration.

(see the talk on Trust and the draft OSG VO AUP for more discussion).



---

Any questions?