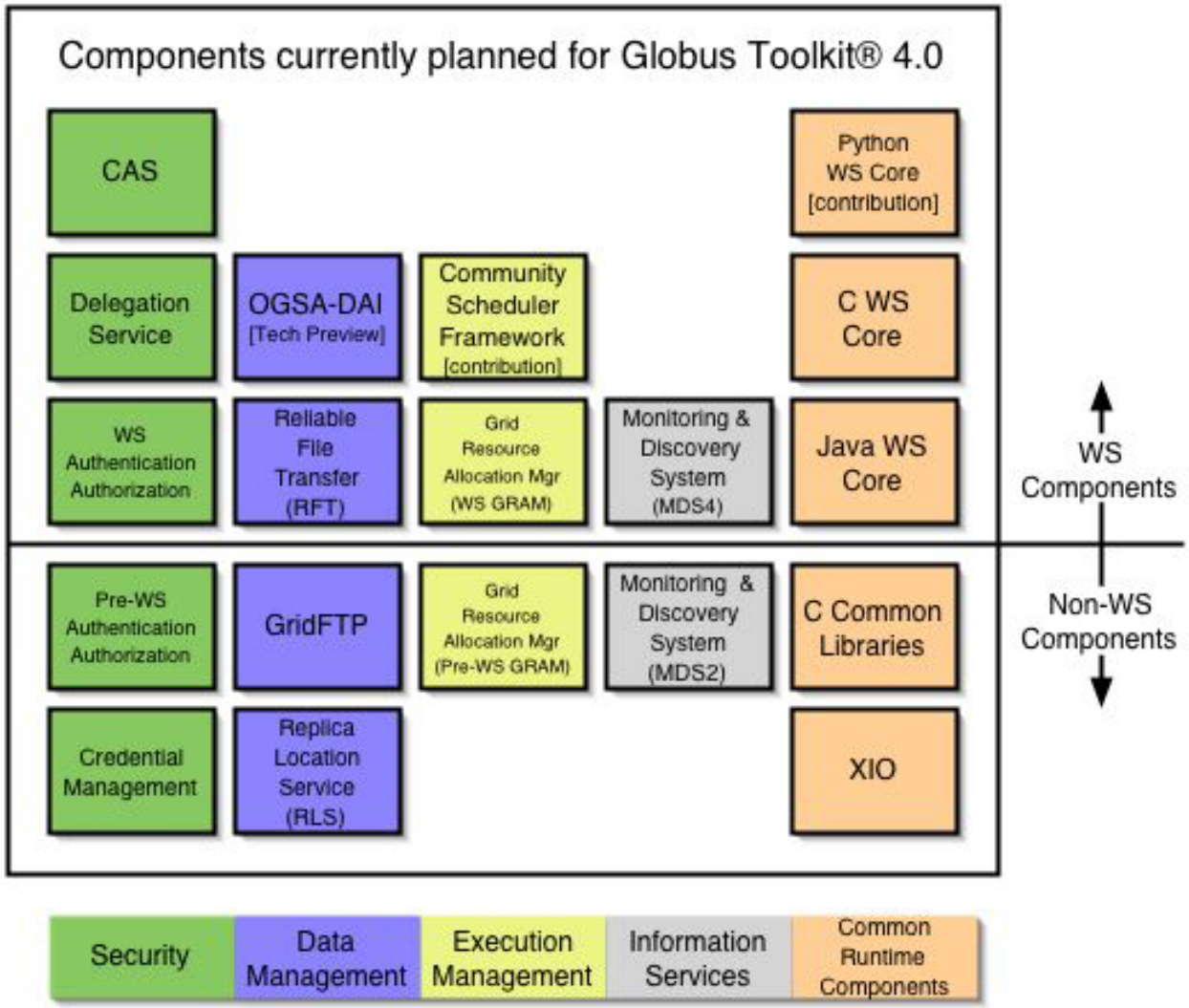
Overview

- Overview of the Toolkit
- Installation
- Packaging
- Security
- Resource Management
- Data Management
- Information Services


Components currently planned for Globus Toolkit® 4.0
CAS
Python WS Core [contribution]
Delegation Service
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
C Common Libraries
Non-WS Components
Credential Management
Replica Location Service (RLS)
XIO
Security
Data Management
Execution Management
Information Services
Common Runtime Components

Full Toolkit Installation Prerequisites

- Java 1.4.2
- Ant 1.5+
- C compiler
- GNU make/tar
- JDBC-compliant database (like postgres)
  - Required for RLS, RFT only
- Optional:
  - Junit (for unit tests)

Installing Java WS Core Only

- Core
  - Source: run ant dist
  - Binary: untar/unzip
- This installs a Java-only development platform

Full Installation

- ./configure --prefix=/usr/local/gt3.9.5
- make
- If you want to install RLS, must specify a path to the IODBC driver using --with-iodbc=/path/to/iodbc
- Supports sub-targets if you want to build only a subset of the toolkit. For instance:
  - make gridftp

Configuration

- The service-level configuration files are under $GL/etc/service_name
- A default set of values is supplied at installation
  - Format looks like:

```
<parameter name="bogusName"
  value="bogusValue" />
```

User Configuration

- Most user configuration is done in environment variables for the C code
  - Will be mentioned throughout presentation
  - Security options summarized at http://www.globus.org/security/config.html
- Java configuration done in $HOME/.globus/cog.properties
  - Create it if you want non-default behavior
  - Summarized at http://www.globus.org/cog/distribution/1.1/FAQ.TXT

Walkthrough Installation

- Install Java 1.4.2, ant 1.6.3, junit 3.8.1
- Create a “globus” account to own install
- Set JAVA_HOME, ANT_HOME
- Download and untar installer
- ./configure --prefix=/opt/gt3.9.4 && make

Security Configuration

- After installing the toolkit, we have the tools to acquire certificates
- Globus uses Grid Security Infrastructure (GSI) for security
  - PKI based, so we need X.509 certificates
- This will provide mutual authentication between services, single sign-on, and cross-organizational security
- http://www.globus.org/security/

Acquiring Certificates

- Best option: Acquire a certificate from an existing Certificate Authority (CA)
- Second-best: Setup a new CA for use while you're learning, and find a real CA for later use
- Last resort: Use a certificate from an untrustworthy source
- For all options, you must install the CA certificate on every machine you use

Certificate: Best Option

- You will have to prove your identity to the CA, then will receive a certificate
- You can check the TERENA Academic CA Repository for research/academic CAs
- The CA should have instructions for acquiring the CA certificate and installing it into /etc/grid-security/certificates

Certificate: SimpleCA

- Globus has a GPT package designed to manage a small CA for testing use
  - Based on CA scripts provided by openssl
- Every full installation has the capability to create a new CA
- If you perform multiple installations, create only **one** CA to use for the whole site
  - After that, you can practice using two if you want to see configuration requirements

Certificate: SimpleCA Continued

- Run $GLOBUS_LOCATION/setup/globus/setup-simple-ca
- Accept the defaults
- Your certificate authority is now installed in $HOME/.globus/simpleCA/
- A package is created for deployment on other machines:
  - globus_simple_ca_[CA-HASH]_setup.tar.gz

Certificate: SimpleCA Continued

- You may now request a certificate using grid-cert-request
- Then, on the machine where you installed the CA, run grid-ca-sign
  - $GLOBUS_LOCATION/bin/grid-ca-sign -in bogusin.pem -out bogusout.pem
- The file named in the –out flag should be sent back to the user who requested the certificate
- The signed cert should be placed in $HOME/.globus/usercert.pem

Certificate: Globus Certificate Service

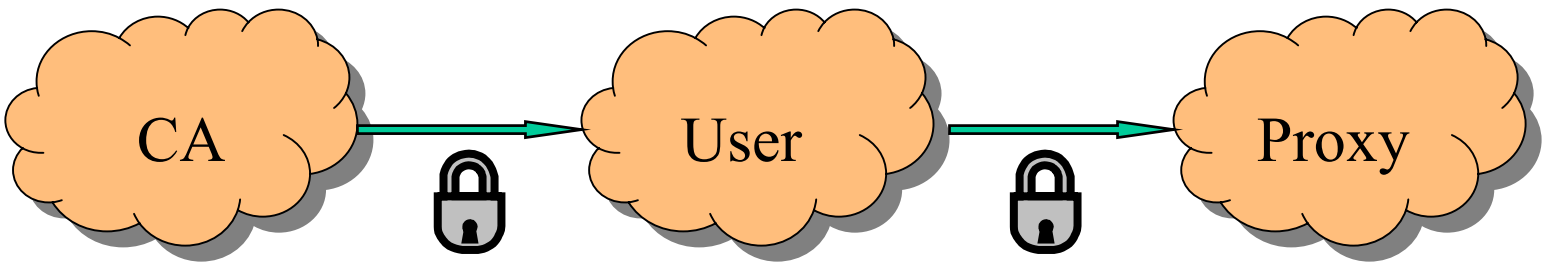
- Now there is an even more explicitly low-trust CA available at http://gcs.globus.org:8080/
- It auto-signs unique certificates with no validation
- This is the fastest way to start testing, but only useful for local testing

CA Certificates Review

- The CA certificate must wind up in /etc/grid-security/certificates
- If you don't have root, you can store those somewhere else
  - Set X509_CERT_DIR for C commands
  - Set "cacert" in ~/.globus/cog.properties for java commands

Proxy Certificates

- Once you have requested and received a certificate from a CA, you may create proxies
- Proxies are used in the mutual authentication process
  - By default, good for 12 hours
  - Stored on local disk



Proxy Certificates Continued

- Three ways of creating proxies:
  - grid-proxy-init (C commandline)
  - org.globus.tools.GridProxyInit (Java gui)
  - org.globus.tools.ProxyInit (Java cli)
- Warning! ProxyInit will echo password
- To change proxy file locations:
  - X509_USER_PROXY for C commandline
  - proxy= in cog.properties for Java
- Use –verify –debug to check that CA cert is properly installed

Walkthrough Installation (Continued)

- globus$ $GLOBUS_LOCATION/setup/globus/setup-simple-ca
- root# $GLOBUS_LOCATION/setup/globus_simple_ca_[hash]_setup/setup-gsi
- root# grid-cert-request –host
- bacon$ source $GLOBUS_LOCATION/etc/globus-user-env.sh
- bacon$ grid-cert-request
- globus$ grid-ca-sign

Container and Host Credentials

- You will have two copies of the hostcert
- One will be used by root-owned services and owned by root
  - /etc/grid-security/hostcert.pem
  - /etc/grid-security/hostkey.pem
- The other will be used by the container and owned by the globus user
  - /etc/grid-security/containercert.pem
  - /etc/grid-security/containerkey.pem

Running the GT4 Container

- Starting a container makes your installed gridservices accessible
- Run $GLOBUS_LOCATION/bin/globus-start-container
- It will print out a list of available services and keep control of the terminal
- If your machine's hostname is incorrect, add a logicalHost parameter to the globalConfiguration in $GL/etc/globus_wsrf_core/server-config.wsdd

Resource Management

- The Globus Resource Allocation Manager provides a single standard interface for requesting and using remote system resources
- Prerequisites:
  - A certificate for the container
  - A user certificate for the user
  - A sudo entry for running jobs as other users
  - A grid-mapfile for mapping certificate DNs to usernames

Resource Management: grid-mapfile

- GRAM is going to receive a certificate subject name
  - /O=Bogus/OU=Bogosity/CN=Bogus User
- It needs to know a local account name for that subject name
  - Specified in /etc/grid-security/grid-mapfile
  - "/O=Bogus/OU=Bogosity/CN=Bogus User" boguser

Resource Management: sudo

- Two lines to allow the globus user to run the $GL/libexec/globus-gridmap-and-execute command as any (non-privileged) user
  - Will never start a process as root
- The command will perform gridmap authorization of the requested user account, then execute either the jobmanager or the proxy tool

Resource Management: Submitting a Job

- You need to create a proxy of your certificate:
  - grid-proxy-init
- Once you have created a proxy, you may use the command line client:
  - bin/managed-job-run –f share/gram-client/test.xml
- The test job streams output back to $HOME/stdout

Resource Management: Jobmanagers

- The default ManagedJob service runs a job with fork/exec
- There are other packages for adding interfaces to scheduling systems like PBS, LSF, and Condor
- Select them at configure time with

  --enable-wsgram-{pbs, condor, …}
- Implemented as Perl modules in

  $GLOBUS_LOCATION/lib/perl/Globus/GRAM

Resource Management: Jobmanagers Continued

- The jobmanagers are perl modules conforming to the API at http://www.globus.org/gram
  - Under GT2 GRAM Documentation, "GRAM Job Manager Reference Manual"
- You may need to edit the perl module corresponding to your scheduler
  - Like lib/perl/Globus/Gram/pbs.pm for PBS
  - For instance, to use rsh vs. ssh

Data Management

- Data Management includes GridFTP, Replica Location Service (RLS), and Reliable File Transfer (RFT)
- Prerequisites:
  - Host certificate, user certificate
  - inetd/xinetd entry for GridFTP server
  - Database for RFT
  - Database for RLS

GridFTP

- GridFTP is a high-performance, secure, reliable data transfer protocol
- First, add an entry to /etc/services
  - Port 2811 is the IANA GridFTP port
- Then, add an entry to inetd/xinetd
  - $GLOBUS_LOCATION/sbin/in.ftpd
  - Also, make sure to set LD_LIBRARY_PATH in the environment

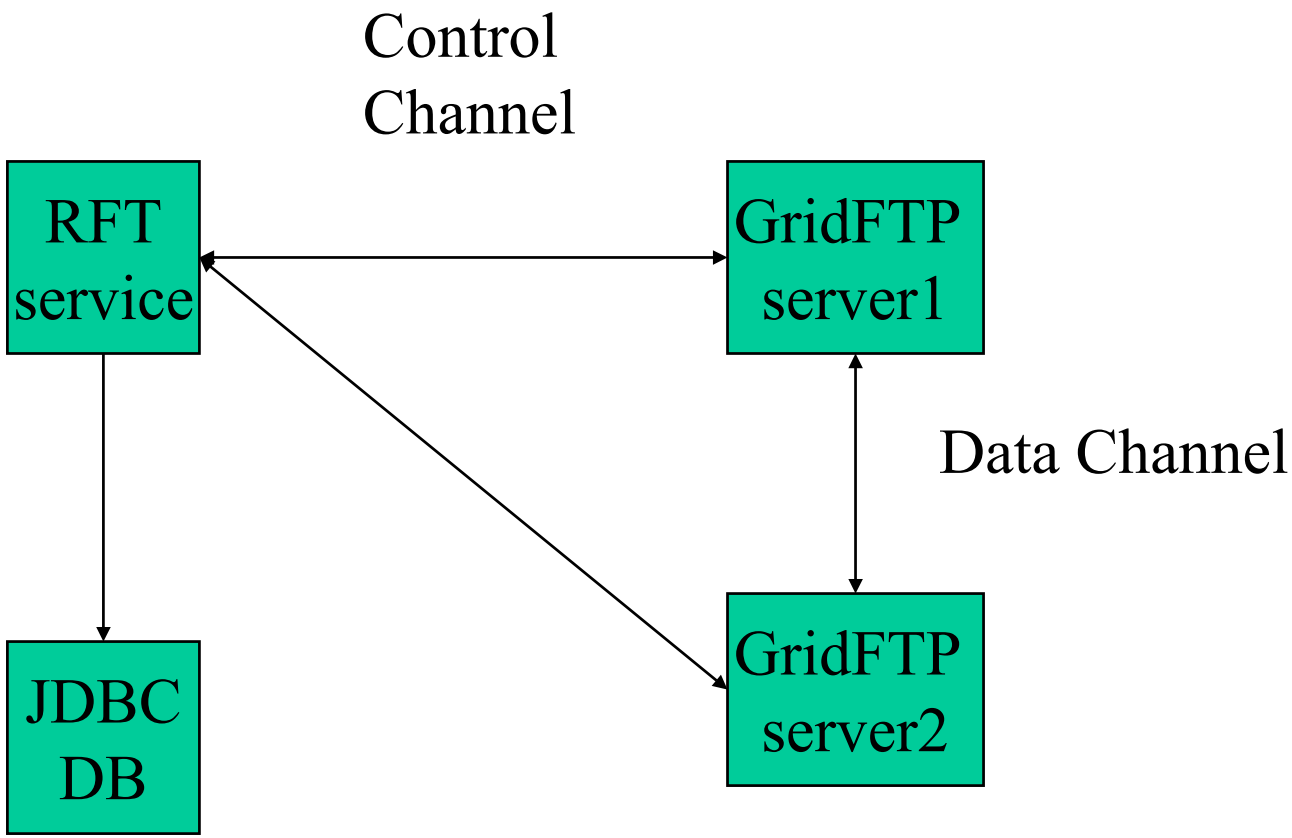
GridFTP Client

- globus-url-copy is the C commandline client
- First, create a proxy
- Then source etc/globus-user-env.{sh,csh}
  - This sets up your PATH and LD_LIBRARY_PATH
- Then, you can try copying a file:
  - globus-url-copy gsiftp://bogus.hostname/etc/group file:///tmp/copy-of-group

Reliable File Transfer

- RFT is a webservice for controlling and monitoring third-party transfers
- Setup a database for persistent data storage using the supplied share/globus_wsrf_rft/rft_schema.sql
- Edit the dbConfiguration in etc/globus_wsrf_rft/jndi-config.xml if you use a non-default database name
- share/globus_wsrf_rft_client contains a sample transfer for use with bin/rft

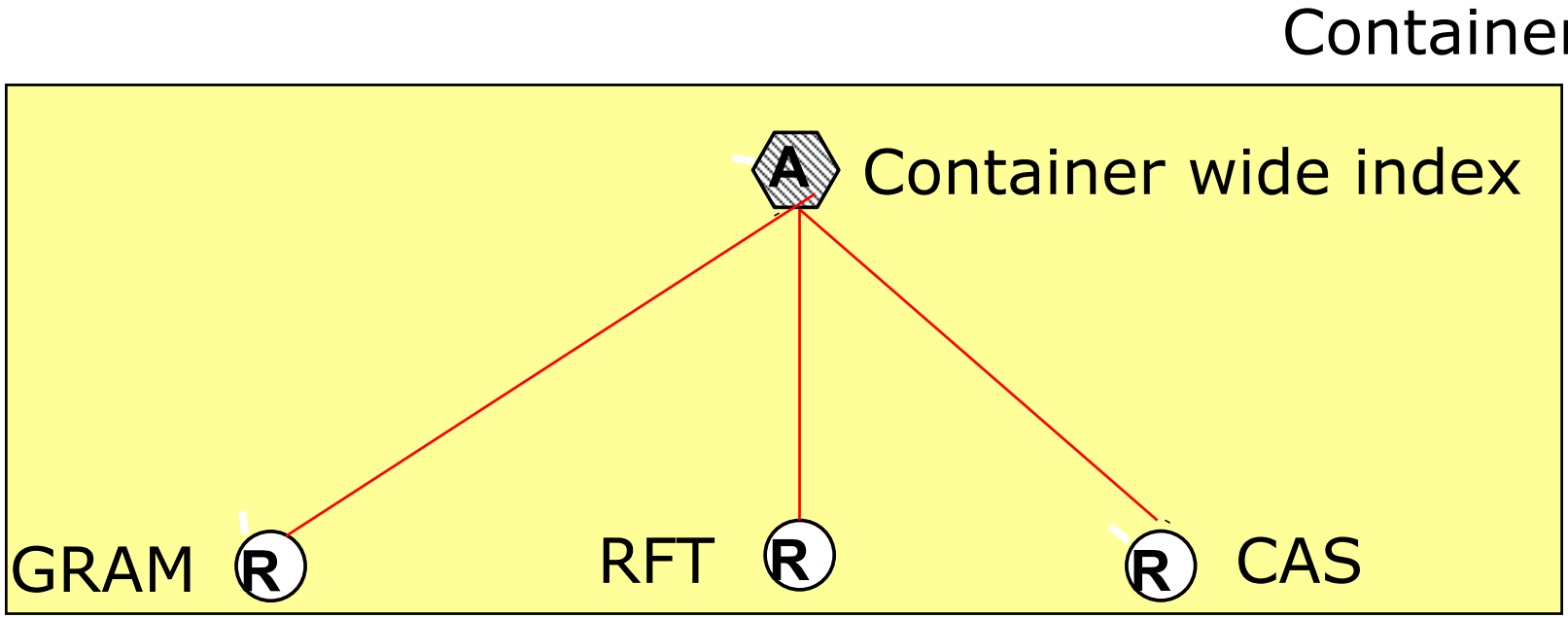
Reliable File Transfer



MDS components

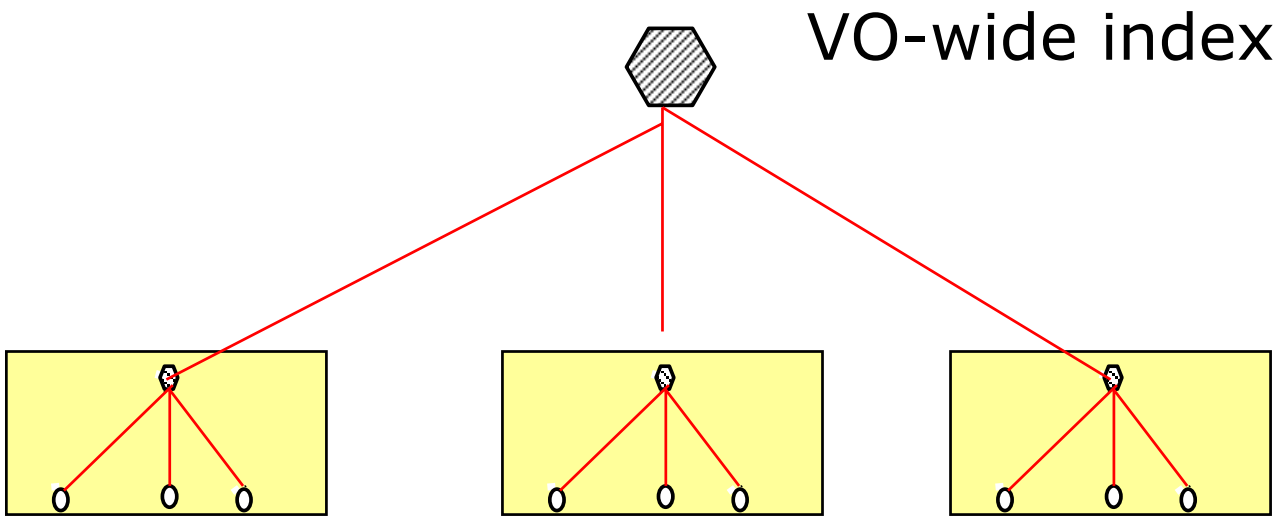
- Index service
  - Gathers information about several resources and publishes in one place
  - Container-level index gathers information about resources in that container
  - VO-level index gathers information about all resources in a VO
- WebMDS
  - Web-browser interface to MDS information
- GRAM cluster information
  - Gateways from Ganglia or Hawkeye

Containerwide index



- Each GT4 container has a local index
- Each service automatically registers to container index when correctly configured

VO-wide indexes



- Local indexes can be registered to VO wide indexes
- Config file at resource container or at VO index

WebMDS

- Web-based interface to display monitoring information

Registering a container index into a VO index

- Config file:

`$GL/etc//etc/globus_wsrf_mds_index/hierarchy.xml`

- At the VO index
  - Configure the URLs of container indexes
  - Add lines like:
  - <upstream>http://myresource.isi.edu:8080/wsrf/services/DefaultIndexService</upstream>
- At the resource containers
  - Configure the URLs of the VO indexes
  - Add lines like:
  - <downstream>http://myvo.org:8080/wsrf/services/DefaultIndexService</downstream>

Configuring GRAM to use a cluster monitoring system

- GRAM extracts and publishes cluster information from either Ganglia or Hawkeye
- $GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/gluerp.xml
- <defaultProvider> tag specifies whether to use Ganglia or Hawkeye or none.
- Uncomment appropriate example supplied in the config file

Some Possible Errors

- Didn't grid-proxy-init for client
  - Remember, proxy expires after 12 hours
- Couldn't find a valid trusted certificate directory
  - Populate /etc/grid-security/certificates
- /etc/grid-security/grid-mapfile
- Hostname does not match IP address
- Wrong database owner

Where To Go For More

- http://www.globus.org/toolkit/docs/development
  - Each component has a user's guide, developer's guide, and admin guide
- For installation, follow the instructions at http://www-unix.globus.org/toolkit/docs/development/3.9.5/admin