dev.globus.org and the Globus Incubator Process

Contact email: childers@mcs.anl.gov, jms@mcs.anl.gov
Contact web: http://dev.globus.org

Globus was first established as an open source software project in 1996. Since that time, the Globus development team has expanded from a few individuals to a distributed, international community. In response to this growth, the Globus community established dev.globus.org, a source code development infrastructure and meritocratic governance model to more easily expand the Globus community, enable additional projects to join Globus, and in general to make the process around Globus easier to understand and more transparent.

The Globus governance model and infrastructure are based on those of Apache Jakarta. Control over each individual software component (project) is in the hands of its most active and respected contributors (committers), with the Globus Management Committee (GMC) providing overall guidance and conflict resolution.

The infrastructure comprises repositories, email lists, Wikis, licenses and bug trackers configured to support per-project community access and management. Additional information about the conventions and guidelines can be found at http://dev.globus.org/wiki/Guidelines.

Current dev.globus Projects

Common Runtime Components

C Core Utilities: Used for maintaining machine portability, as well as some basic timer routines.
C WS Core: Provides core C runtime technologies for development of WS Grid services and clients.
CoG jglobus: Provides core Java non-WS runtime and security technologies.
Core WS Schema: Maintains the WSRF and WSN schema for the WS Core projects.
Java WS Core: Provides an implementation of the WSRF and WSN family of standards, as well as WS security technology and the Servicegroup implementation.
Python Core: Provides core Python runtime technologies for development of non-WS and WS Grid services and clients.
XIO: Provides a single API (open/close/read/write) that supports multiple wire protocols, with protocol implementations encapsulated as drivers.

Data Management

GridFTP: Provides high-performance, secure, reliable data transfer technologies optimized for high-bandwidth wide-area networks.
OGSA-DAI: Provides a data service framework for accessing and integrating data, and Grid-enabling databases.
Reliable File Transfer: Provides WS-based reliable data transfer technologies.
Replica Location: Provides data replication and discovery technologies.
Data Replication: Allows users to identify a set of desired files existing in their Grid environment, to make local replicas of those data files, and to register the new replicas.

Execution Management

GRAM: Enables users to locate, submit, monitor, and cancel remote jobs on Grid-based compute resources.
MPICH-G2: Provides a Grid-enabled implementation of the message passing interface (MPI) standard, and is based on MPICH.

Information Services

MDS4: Provides resource monitoring and discovery, including the Index and Trigger services and a visualizer, WebMDS.

Security

C Security: Provides C WS and non-WS security technologies.
CAS/SAML Utilities: Provides community authorization infrastructure.
Delegation Service: Provides a technology for delegating credentials to a host.
GSI-OpenSSH: Provides GSI authentication mechanisms for the SSH protocol 2.
MyProxy: Enables the storage and retrieval of X.509 credentials in a repository.

Distribution Projects

Globus Toolkit: Creates official Globus Toolkit distributions by integrating a select group of Globus technologies. The project strives to create state-of-the-art open source Grid toolkits of exceptional quality.

Documentation Projects

GT Release Manuals: Provides formalized documentation for the components in the Globus Toolkit distributions.

Incubator Process

The entry point for new project to join Globus is the Incubator Process, as overseen by the Incubation Management Project (IMP). The process consists of several stages:

1) A Candidate project is proposed to the IMP. Candidate proposals simply detail what the project is, and its benefit to Globus.

2) The IMP (and possibly the Globus Management Committee (GMC) discuss the project and vote on its acceptability. The IMP does not perform filtering on the basis of technical issues, but on the basis of the likeliness of the project becoming a successful meritocratic community.

3) Upon acceptance, the project, now a *Globus ProtoProject*, is assigned a mentor to bridge between the IMP and the project, and basic infrastructure is set up, including CVS/SVN space, wiki pages, mailing lists, licenses, and bugzilla space. A *Mentor* is assigned to assist in bridging between the project and the IMP, and to help the project with any infrastructure-related questions.

4) The ProtoProject will undergo reviews, at least quarterly, which will have one of three outcomes:

   a) The ProtoProject will remain in the Incubation state.

   b) The ProtoProject will be asked to retire due to lack of progress.

   c) The ProtoProject will "escalate" to become a full Globus project.

Incubator Projects

Dynamic Accounts: Allows a Grid client to dynamically assign Unix accounts on a remote resource based on PKI credentials and the authorization information they carry. The core of the Dynamic Accounts effort is the WorkSpace Service for Dynamic Accounts.

GridShib: Integrates a federated authorization infrastructure (Shibboleth) with Grid technology (the Globus Toolkit) to provide attribute-based authorization for distributed scientific communities.

GridWay: Enables large-scale, reliable and efficient sharing of computing resources using GRAM to interface to different distributed resource management systems and MDS for basic resource information.

gt-hs: Globus Toolkit Handle System provides a means for uniquely identifying structured data and other resources that can be used to store and retrieve state information about them.

MEDICUS: Medical Imaging and Computing for Unified Information Sharing is a project to federate medical imaging and computing resources for clinical and research applications.

Metrics: Measure the use of Globus software in terms of both quality and quantity. Quantity issues include how much is the software being used, by how many people, and how those people are distributed. Quality issues include how the software is being used, how useful it has been, and what the results of that use have been.

OGCE: Builds computational Grid and science gateway Web ports, Grid client tools, and supporting software. OGCE provides client environments for GT4, both WS and pre-WS, the Storage Resource Broker, and Condor. The portal environment also allows for the integration of collaboration tools and services, such as those provided by the Sakai project.

PURSe: Provides a set of tools for automating user registration and credential management, especially for portal-based systems.

ServMark: The integration of two performance evaluation tools, DiPerF and GrenchMark, which can be used for performance evaluation, namely the ability to test services in a distributed and scalable way and the capacity to generate and run dynamic test workloads with complex structure.

CoG Workflow: Provides an integrated but modular system that allows users to interact with workflows and monitor state through visual components.

Virtual Workspaces: Allows an authorized Grid client to deploy an environment described by the workspace meta-data on a specified resource quota. The environments that can be deployed in this way range from atomic workspaces to clusters and more complex constructs.