Globus Software Development Roadmap

March 2006

Draft 1.0.3

1 Overview

Globus Toolkit development began in 1996 as part of an effort to provide the scientific community with a set of tools and services for building distributed applications. Thanks to the recent NSF award for the *Community Driven Improvement of Globus Software* (CDIGS), GT developers are able to continue this work in 2006.

CDIGS project goals:

1. Evolve and enhance GT functionality, performance, scalability and robustness to meet the requirements of its user community
2. Improve usability and manageability so as to decrease the cost and complexity of deploying, operating and using a GT-based infrastructure
3. Support major NSF users and communities
4. Expand the Globus user base

In order to solicit feedback and discussion regarding GT development plans, the CDIGS team is hosting a GT Roadmap Workshop in Chicago this March. This first draft of the Globus Software Development Roadmap, prepared as supporting material for the workshop, summarizes the current high-level development directions of this team.

2 Globus Background

To provide context, we provide some background information on Globus software, the Globus developer community, and the Globus user community.

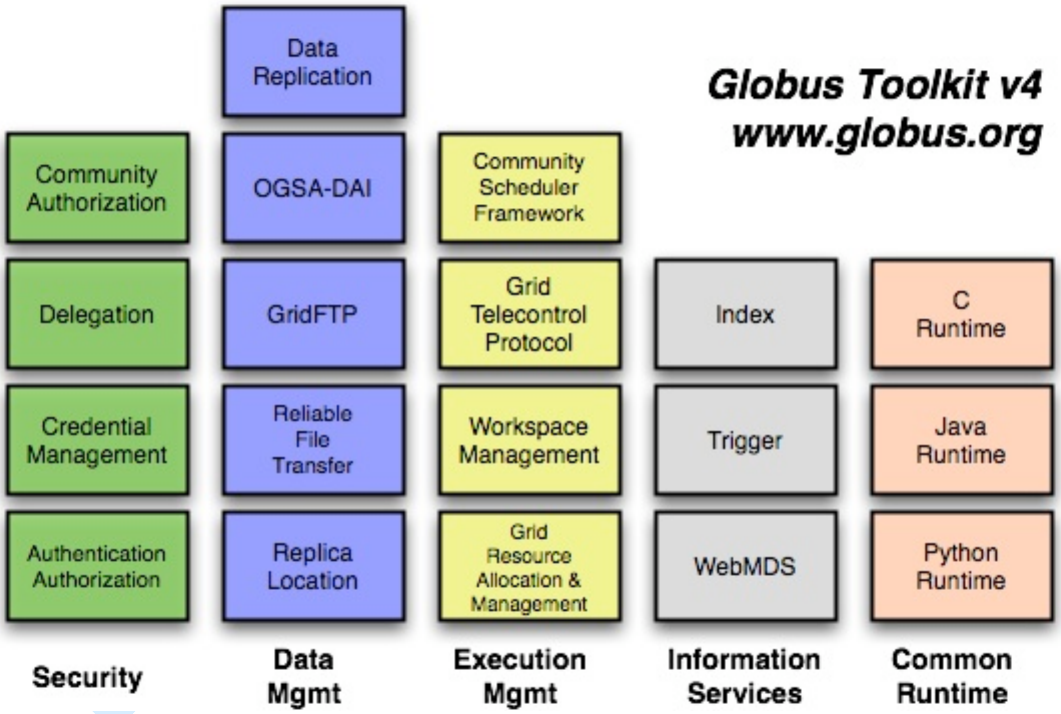
2.1 Globus Software

The Globus Toolkit (GT) is a collection of development tools, libraries and services for building Grid infrastructure, client libraries and end user tools for building Grid applications. The current Globus software release (Version 4.0.1) is the result of over ten years of development. Originally, GT was written in the C programming language and utilized a number of IETF and customized networking protocols. With the introduction of GT3, the toolkit took advantage of the increased adoption of service oriented architectures built on Web Services (WS) standards developed by the GGF, OASIS and W3C. This adoption continued in GT4. Because of its pervasiveness as an implementation language for Web services, Java is used for a significant number of the newer components (as well as Python).

The following figure provides a high-level view of the major components in the current GT release. GT Components are grouped into five main functional areas:

- **Common runtime** components provide low-level infrastructure for communication and threading, as well as libraries and tools for service and state management. The C, Java, and Python subsystems provide support for services written in C, Java, and Python, respectively.
- **Data services**, which address the requirements for locating, replicating, transferring and accessing data stored in either files or databases. A specific focus of components in this area is the management of very large scale datasets both in terms of number of data objects (i.e. files) and the size of those individual data objects.

- **Security** components address cross-cutting issues of authentication and authorization of both users and services. With GT4, the security implementation has moved from primarily a collection of libraries to a more services oriented structure, as illustrated by the introduction of standalone delegation and authorization services.
- **Execution** components provide for the deployment, execution and management of applications. Previous versions of the toolkit focused primarily on the execution of user-provided programs. However, newer services have generalized these functions to address deployment and management of community services as well as providing enhanced capability with respect to configuring and managing the execution environment in which applications and service run.
- **Information** components are concerned with the monitoring and discovery of both GT and non-GT services and capabilities. Starting with GT3 and continuing in GT4, many of the basic functions of state publication and management have been incorporated into the common runtime component of the system. The information components build on this by providing an extensible infrastructure for aggregating service state information, providing interfaces for publishing information from non-GT services, and notifying users when events of interest occur.



2.2 Globus Developer Community

The Globus developer community is large and distributed. Its work is made possible by an Apache-like community infrastructure, GlobDev, at http://dev.globus.org, which provides code repositories, email lists, bug tracking mechanisms, and so forth.

A list of contributors is published here. The following organizations have multiple Globus developers and/or have been involved in Globus development for a long time:

- University of Chicago and Argonne National Laboratory: staff have contributed in every major area of Globus development.
- A team at the University of Southern California's Information Sciences Institute: staff are active in Data Management, Information Services and Security.
- Lawrence Berkeley National Laboratory: staff lead work on the Python Common Runtime and related components.
- University of Edinburgh: staff lead work on OGSA-DAI and related components.
- NCSA: staff lead work on MyProxy and related components.
- Univa Corporation: staff have contributed in a range of areas

2.3 Globus User Community

In formulating the GT roadmap, we must inform our decisions based on the priorities and requirements of the GT user community. Because of its central role in this process, it is instructive to consider who the Globus user community is. One characterization would be to consider the level of interaction in which one has with Globus software. From this we can identify the following types of users:

- Infrastructure builders whose primary focus is in using the Globus software base as a means of constructing shared infrastructure for use by one of the other identified community members. Infrastructure builders are strongly driven by manageability, configuration, deployment, and operational issues.
- Service developers who use the Globus tools, libraries, and runtime service to build new Grid services and capabilities
- Application developers, who develop Grid applications. The distinction between application developers and service developer may be somewhat fuzzy, however, we will consider application developers be more focused on domain specific uses of the Globus toolkit, while service developers will tend to concentrate on more general purpose infrastructure components.
- End users who will develop no code, but may have extensive interaction with GT components, potentially without even realizing that they are using Globus.

Within each of these categories, the users come from a wide range of disciplines. GT has been used extensively in the physical sciences, including high energy physics, chemistry, geophysics, astrophysics, and biology. There has also been increased use of Globus to address problems in domains such as engineering design, financial services. One of the reasons that Globus is effective in such diverse application areas is its adherence to a layered architecture and a "toolkit" oriented approach. We would anticipate that directions outlined in this roadmap would continue this approach, avoiding whenever possible items directed toward a specific use case or application domain in favor of general purpose, reusable solutions.

2.4 The Globus License

From its inception, the Globus Tookit has been distributed under the terms of a liberal open source license, which essentially allows for unrestricted use in both academic and commercial applications. Prior releases of Globus were distributed with a license that was developed specifically for use with GT. While this license permitted virtually unrestricted use of the toolkit, there were some minor issues associated with the language. To further the growth of the GT user community, we addressed these issues in GT4.0.1 by adopting the Apache open source license in an unmodified form. Given the broad international adoption of Apache software, this change should serve to further broaden the use of Globus technology.

3 Summary of Recent Development Work

The Globus Toolkit Version 4 (GT4), first made available in April 2005, represents our latest stable release stream. Distinguishing features of GT4, as compared to previous toolkit releases, include:

- Transition to the Web Services Resource Framework (WSRF) as the standards base for managing service lifetime, state, and asynchronous notification
- Complete reengineering of the GRAM service with a focus on scalability and reliability
- Streamlined deployment and configuration
- A new open source GridFTP server implementation
- Across the board improvements in scalability, performance, and robustness
- Significantly improved documentation
- Inclusion of new services (some in technology preview mode) including a dynamic workspace service and a Data Replication Service.
- Support for Java, C and Python service and client implementations.

An August 2005 maintenance release (4.0.1) included bug fixes and performance improvements. Since that time, one additional release for the TeraGrid deployment provided limited additional GRAM and MDS functionality. All releases post-4.0.0 have been backward compatible, with no interface changes.

While not currently available as a general release, development of a number of important new features has also taken place in addition to bug fixes, better testing and other incremental improvements. These include more comprehensive use of attributes throughout the GSI, dynamic service deployment in the GT Java hosting environment, performance and scalability enhancements in MDS, and new clients. These changes either augment existing interfaces, or introduce new interfaces. This new functionality will be made available in future development releases.

4 Our Views on Short-Term Goals

We believe that short-term (next 12 month) Globus software development should emphasize the following general goals. These goals are based on the current state of GT implementations, observations on the trends in applications and production deployments, and current standards activities. There are a number of directions that we currently believe are going to be important and should be reflected in the priorities in the roadmap. We emphasize that these are high-level goals are meant to reflect general trends and in no way are these observations a substitute for the roadmap process that we are initiating with this document.

Of course, we believe that it is critical to continue to push on the goals of performance, robustness, scalability and usability. These goals take many different forms, from improving documentation, performance tuning, re-engineering critical path components, adopting latest versions of standards, etc. We see these general goals appearing frequently in the detailed roadmap items proposed below.

At a more high-level, we observe that there is increasing demand for more dynamic and adaptive configuration of Grid components. Ideally, it should be possible for a VO to deploy, configure, and manage its service infrastructure in a completely dynamic manner. We are starting to see some activities in this area: for example the introduction of the ability to dynamically deploy and instantiate a new service into a running GT4 container, and the ability to use GRAM to dynamically deploy nested VO-specific scheduling into the EGEE and OSG environments. We believe that it should be a priority to extend and enhance these basic capabilities, introducing support for advanced reservation, service and resource provisioning, for example. Configuration and management are another aspect of this type of dynamic adaptation. Hence, we believe that providing uniform interfaces, mechanisms and tools for manageability will also be needed.

Another major thrust should be the additional mechanisms to leverage institutional security infrastructure, particularly within the areas of policy specification, and associated auditing operations. For example, with the increasing acceptance of attribute authorities, such as Shibboleth, and the increased adoption of attribute security standards, such as SAML and XACML, we are seeing valuable institutional infrastructure that must be exploited. The work towards more uniform integration of attributes authorities throughout the security services is a step in this direction.

Finally, we observe that storage management continues to be a major driver for GT users. Along this front, we anticipate three high level directions. First is the integration of more managed storage, both in terms of space reservation, bandwidth management, and transfer scheduling. Second, there has been some support for data objects other than files with the integration of the DAI component into GT. We believe that further integration and exploitation of these capabilities will be important. Finally, we note that storage management tends to be an end-to-end system function, involving many components. This requires a solutions-oriented approach to storage management. We are starting to see some of this in GT4, for example with the introduction of the Data Replication Service (DRS), which combines the Replica Location Service (RLS), the Reliable File Transfer Service (RFT), and GridFTP. We believe that it will be important to continue developing high-level management capabilities for data.

We now present a more detailed view of current Globus development workplans. The next sections describe work underway by University of Chicago and USC staff in the areas of Common Runtime, Data Management, Execution Management, Information Services, Security, Documentation and Packaging.

5 Common Runtime Components

Common Runtime technologies provide low-level infrastructure for communication and threading, as well as libraries and tools for service and state management. This section describes the development status and plans for the following Common Runtime components: C WS Core, Java WS Core and XIO. The section begins with a brief overview describing current development activities and concludes with a summary list of the team’s near-term development tasks.

5.1 Current Status

For the **C WS Core** component, our near-term focus is on improving ease-of-use. We have made many improvements and will continue to work in this area. There is significant advantage in using the C WS core for clients due to the lack of JVM startup overhead. This is particularly advantageous when invoking the client repeatedly, such as from a script. We will focus on enabling that usage pattern and assisting communities, particularly GlobDev communities, in utilizing C WS core for clients. C WS Core is a fairly young implementation and there is a laundry list of features we would like to add, such as *XPath* integration, service sandboxing, *HTTP get* of schema with caching, global configuration/deployment, support for local service routing, additional secure conversation support, an improved authorization framework, and bindings improvements.

The *CoG JGlobus*, *Java WS Core* and *Core WS Schema* GlobDev projects all contribute to the production of GT4's **Java WS Core** component. Java WS Core is one of the more mature components in GT4, and thus far has proved to be a decent platform for building and hosting java services. The current and near-term focus of Java WS Core development is on maintenance as well as providing more advanced features. Future plans include support for additional WS standards, including WS-Enumeration, as well as an improved notification/subscription infrastructure. Such infrastructure improvements include notification pull support, notification throttling, and persistence of subscriptions to a database. Other advanced features, such as dynamic deployment, connection persistence, and support for SOAP with attachments have recently been completed and will appear in the future development releases.

Near-term work for the **XIO (eXtensible IO)** component revolves around user support, ease-of-use improvements, bug fixes, etc. We will be working on additional APIs and utilities to improve the usability of XIO. This includes support to make it easier to integrate third party libraries, blocking interfaces, an xio_copy library to handle copying between any two xio handles, etc. We have prototyped an xio_perf tool, similar to a stripped down iperf, that enables apples-to-apples performance comparisons across XIO drivers. We will continue to harden this as a testing tool and as a demonstrator of driver use. We will also continue to support, harden, and extend our existing driver set.

5.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Apply 3rd party software updates	robustness		CoG jglobus	4010
GridFTP client library improvements	performance, scalability		CoG jglobus & GridFTP	4011
Upgrade to the latest WSRF/WSN specs	functionality		Core WS schema	4012
WSDL/WS-I validation tests	robustness		Core WS schema	4013

Add JMS support for notifications	functionality		Java WS Core	4014
Add WS-Enumeration support	functionality		Java WS Core	4015
Enhance service-side notification infrastructure	performance, scalability		Java WS Core	4016
Improve efficiency of subscription code	performance, scalability		Java WS Core	4017
Apply 3rd party software updates	robustness		Java WS Core	4018
Improve remote debugging and diagnostics tools	usability, manageability		Java WS Core	4019
Add dynamic deployment	functionality		Java WS Core	4020
Driver descriptor	usability, manageability	ROMIO/PVFS	XIO	4054
Adaptive driver behavior	Usability, manageability		XIO	4055
Dynamic stack modifications	functionality		XIO	4056
Copy library	usability, manageability		XIO	4057
Pipe driver	usability, manageability		XIO	4058
Enhance driver SDK	usability, manageability		XIO	4059
Cultivate driver developer community	community expansion	MPICH	XIO	4060
Build helper utilities	community expansion	PVFS	XIO	4205
Add data management features	functionality	GridFTP developers	XIO	4206

6 Data Management

Data Management services and tools provide for the discovery, transfer and access of remote data in a variety of formats and storage systems. This section describes the development status and plans for the following Data Management components: Data Replication Service (DRS), GridFTP, Reliable File Transfer Service (RFT) and Replica Location Service (RLS). The section begins with a brief overview describing current development activities and concludes with a summary list of the team's near-term development tasks.

6.1 Current Status

The **Data Replication Service (DRS)** was first released as a Technical Preview component in the GT 4.0.0 release, and the GT4.0.1 release contains a major improvement of the service. Our current focus is on adding new functionality and improving scalability when handling multiple simultaneous client requests. Planned work includes extending the service to handle push-based as well as pull-based replication operations, adding persistence to the DRS resource to support failure recovery, and continued improvements to the scalability and robustness of the service. Users from several application communities have installed and used DRS and are providing feedback. We welcome new DRS users, and are interested in interacting with additional application communities to understand their replication requirements, driving further improvements to DRS.

GridFTP has just completed a major development cycle resulting in a brand new server released in GT 4.0. Our near term work revolves around user support, ease of use improvements, bug fixes, etc. We will be looking at adding support to protect the server from crashes due to OS resource exhaustion. Our next major development cycle will involve making GridFTP a "managed resource". We will be adding dynamic backend registration in striped servers, bandwidth throttling, interactions with storage reservations and management, and management of connections, including co-scheduling of the ends of a transfer.

The **Reliable File Transfer Service (RFT)** is a Web Services Resource Framework (WSRF) compliant service providing job scheduler-like functionality for data movement. It has been part of Globus Toolkit since the GT3 release. GT4's GRAM uses RFT for staging in executables and staging out results, and the Data Replication Service (DRS) also uses RFT for its file transfer needs. Our current focus is to improve robustness, scalability and throughput under heavy loads. We are currently working on achieving that, among other things, by adding support to reuse data transfer connections across multiple user requests. To improve ease of use we have recently added an option to use an embedded database that gets deployed and configured at installation time, enabling RFT to work out-of-the-box. Longer term we are interested in adding more "data-scheduler" like functionality to RFT, supporting priorities in transfers, and adding the ability to dynamically change the transfer request.

The **Replica Location Service (RLS)** has been a component in the Globus Toolkit for several years. The GT4 version of RLS represents a significant improvement over earlier versions of the service in terms of scalability, robustness, performance and documentation. Our near-term work for RLS includes completion of a WS-RF interface to the service. Longer term, we are interested in adding more fine-grained authorization to RLS servers, implementation of an embedded database back end and improving the modularity and memory management of RLS server code. Several projects currently use the RLS in their production Grids. We invite new users and new requirements for RLS functionality from application communities.

6.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Data Replication Service feature enhancements	functionality		Data replication	4004
Add persistence to the DRS resource	functionality		Data replication	4005
Push-based replication	functionality		Data replication	4006
Policy-based replication	functionality		Data replication	4007
Architectural requirements gathering	user support		Data replication	4008
Improve robustness of striped GridFTP server	robustness	TeraGrid	GridFTP	4049
Improve the resource management capabilities of GridFTP	usability, manageability		GridFTP	4051
Improve unit tests	robustness		GridFTP	4061
Improve performance on small-file transfers	performance, scalability	LIGO; astro physicists	GridFTP	4062
Embedded SQL database backend	usability, manageability		Replica Location	3995
Add WSRF interface	usability, manageability		Replica Location	3996
Database connection pooling	performance, scalability	TeraGrid	Replica Location	3997
Update documentation	usability, manageability		Replica Location	3998
Decouple SQL statements from code	usability, manageability		Replica Location	3999
Improve memory management	performance, scalability		Replica Location	4000
Add VO-aware authorization controls	functionality	TeraGrid	Replica Location	4001
Add WSRF-based configuration interface	usability, manageability		Replica Location	4002

Improve fine-grained authorization support	functionality		Replica Location	4003
Add ability to prioritize transfers	usability, manageability		RFT	3984
Add ability to schedule transfers	functionality		RFT	3985
Support additional transfer protocols	functionality		RFT	3986
Support multiple sources and partial file transfers	functionality		RFT	3987
Improve throughput performance	performance, scalability		RFT	4022

7 Execution Management

Execution Management components enable the deployment and management of execution environments, including sequential and parallel tasks, dynamically deployed services, dynamic accounts and virtual machines. This section describes the development status and plans for the following Execution Management components: Dynamic Accounts, GRAM and Virtual Workspaces. The section begins with a brief overview describing current development activities and concludes with a summary list of the team's near-term development tasks.

7.1 Current Status

The **Dynamic Accounts Service** (a.k.a. **WSS, WorkSpace Management Service)** was released as a Technical Preview in the GT4.0 release. The service allows an authorized Grid client to dynamically associate a Grid credential with a local Unix user account, and manage this association in terms of its lifetime and associated policies. This allows Grid clients to securely obtain on-demand access to remote resources. The current implementation works with account pools pre-created by the site administrator and provides ways of integrating site-specific management into the implementation. Our ongoing and planned work includes support for a database-backed pool account implementation, support for a C client, as well as new authorization modes. WSS has been deployed by the EGEE project and we are also working on providing support and continued feature development for this community. Several other communities (including OSG and TeraGrid) have also expressed interest in this functionality.

The **Grid Resource and Allocation Management (GRAM)** component provides the ability to submit and monitor jobs on remote resources. Both the pre-WS and WS versions of GRAM are included in GT4, with WS GRAM receiving priority development attention. The team's current focus is on providing support for users new to WS GRAM and gathering requirements on further refinements. Recent work with OSG has lead to adding important scenarios to our test suite and adding performance improvements to the code. Future development plans include adding support for GGF's JSDL.

The **Virtual Workspace Service** was released as a Technical Preview Java grid service in early fall 2005. The service allows an authorized Grid client to dynamically deploy a customized software environment on a remote resource in the Grid and associate it with a hard resource quota (CPU, memory, bandwidth). To achieve this, the current implementation uses virtual machines (specifically the open source Xen implementation) and associated tools. The benefits of this service include more efficient and more flexible use of resources as well as convenience to the users. The current and future work focuses on extending the concept of a workspace to represent a virtual cluster, scheduling and resource management of such clusters, implementing fine-grained resource enforcement, and providing secure ways of workspace deployment. The Workspace service has been deployed in production by OSG as part of the Edge Services project. Several other communities, including EGEE and TeraGrid, have also expressed interest.

7.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Maintain test suite	robustness	TeraGrid, OSG	GRAM	4036
Create validation test suite for users	user support	TeraGrid, OSG	GRAM	4038
Increase concurrency beyond the current 32,000 job limit	performance, scalability		GRAM	4039
Add throttling capabilities for high throughput situations	usability, manageability	TeraGrid, OSG	GRAM	4040
Improve the configuration and setup process	usability, manageability	TeraGrid	GRAM	4042
Optimize performance of processing individual jobs	performance & scalability		GRAM	4043
Support state-of-the-art job submission specs	community expansion	TeraGrid, OSG, GIN	GRAM	4044
Support advanced reservation	functionality	TeraGrid	GRAM	4045
Automatic resubmission of jobs	functionality	EGEE	GRAM	4046
Improve ability to specify job environment	usability, manageability	TeraGrid, EGEE	GRAM	4048

Optimize job throughput	performance, scalability	TeraGrid, OSG	GRAM	4050
Improve job metadata	usability, manageability	TeraGrid	GRAM	4047
Create C client	functionality	EGEE	Dynamic Accounts	4087
Small feature development	functionality, robustness	OSG, EGEE	Virtual Workspaces	4088
Provide support for resource pools	functionality, usability, manageability	OSG, EGEE	Virtual Workspaces	4089
Improve security for VMs	functionality	OSG, EGEE	Virtual Workspaces	4091
Enable creation of a virtual cluster	functionality	OSG, EGEE, potentially TeraGrid	Virtual Workspaces	4092
Enable fine-grained resource allocations	functionality	OSG, EGEE, potentially TeraGrid	Virtual Workspaces	4093
Build community image library	functionality	OSG, EGEE, educational projects	Virtual Workspaces	4094
VM configuration services	usability, manageability	OSG, PlanetLab, educational institutions	Virtual Workspaces	4095
Testbed guidelines and preparation	robustness, usability	OSG, EGEE	Virtual Workspaces	4096
Support VM image reconstruction	usability, manageability	OSG, EGEE, education outreach	Virtual Workspaces	4098
Virtual IP management	functionality, usability, manageability	OSG, EGEE	Virtual Workspaces	4099

8 Information Services

Information Services include high-level services, libraries, and tools for VO-wide resource monitoring and discovery and error notification. Key technologies include standard mechanisms for describing and accessing state, templates for additional information sources, tools for creating and managing higher-level aggregator services, a caching registry for service discovery, a warning service for error notification, and GUI tools. This section describes the development status and plans for our Information Services infrastructure, beginning with a brief overview describing current development activities and concludes with a summary list of the team's near-term development tasks.

8.1 Current Status

The **Monitoring and Discovery System (MDS)** has been a Globus Toolkit component since the first release and has continued to evolve and improve since then. In particular, the LDAP-based Index Service has been replaced by a more robust, WSRF-compliant version, and we have added additional higher-level functionality with the Trigger Service and with the development of a generic Aggregator Framework, which allows easier development of higher level services. We have also significantly improved the level of data collected through information providers, and made it easier for users to develop their own. We now also have a GUI component, WebMDS, to allow easy access to the data across sites. With these additions, we have seen the number of users grow significantly in the last year. Our near term work includes extending the higher-level functionality by adding an archiving service, database backend to the standard index service, and significant performance improvement for the index and query behavior. We are developing additional information providers, especially in terms of additional data from GT4 services and performance benchmark. We are also extending the interfaces to include C APIs and a WebMDS query page for resource selection.

8.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Improve MetaScheduling information	usability, manageability	TeraGrid	MDS	3971
Build performance benchmarks for clusters	performance, scalability	Heart modeling app, TeraGrid	MDS	3972
Add persistence to the Index service	functionality	OSG, TeraGrid	MDS	3973
Add a query interface for MetaScheduling info	usability, manageability	TeraGrid	MDS	3974
Additional Trigger Service documentation	user support	ESG, TeraGrid	MDS	3975
Create Archiver service	functionality		MDS	3976
Create information provider for GridFTP	performance, scalability	SCEC	MDS	3977
Create Inca information providers	functionality	TeraGrid	MDS	3978
Create a C API for the Index and Trigger services	community expansion		MDS	3979

Create clover test coverage reports for MDS infrastructure	robustness		MDS	3980
Improve Index service query performance	performance, scalability	Several requests from Globus user mailing lists	MDS	3981
Usage statistics	functionality	TeraGrid, ESG	MDS	4285
Trigger Service support for usage stats monitoring	functionality	globus toolkit	MDS	4286

9 Security

GT security infrastructure includes services and tools for providing VO-aware authentication and authorization, enabling identity establishment, delegation of authority and application of VO and local policy. This section describes the development status and plans for the Community Authorization Service (CAS), Delegation Service and general security infrastructure for GT. The section begins with a brief overview describing current development activities and concludes with a summary list of the team’s near-term development tasks.

9.1 Current Status

The Java-based **Community Authorization Service (CAS)** has been a component of the Globus Toolkit since GT 3.2. With the release of GT 4.0, CAS is now a WSRF-compliant service. Near-term development efforts include improving usability and manageability, such as adding a simplified administrative interface. Work is also underway to add an authorization interface that augments the existing pull interface. C client-side support is also planned. The CAS team is eager to form collaborations with community members interested in adopting the CAS-based security model for policy management.

GT4 saw the first release of the **Delegation Service**, which allows for delegation of credentials when using protocols that lack inherent support for delegation. Since the GT4 release, the Delegation Service has been used quite successfully by the GT components GRAM and RFT. Future work includes providing the ability to query for existing credentials. We also have plans to make configuration improvements such that the Delegation Service can work off short-term credentials. Key collaborations include working with the EGEE community to provide interoperable services and uniform interfaces. This is a relatively new component and we are keen on gathering additional requirements and feedback from the community.

A key focus for improvements in the general security infrastructure in GT4 include usability enhancements, such as improving error reporting and diagnostics, and simplification of security configuration and setup. Further we are planning various feature additions such as a logging and auditing framework, support for key discovery, and development of tools for ease of policy management. Recently an attribute-based authorization framework allowing for fine grained delegation of rights was added to the Java infrastructure, and we will be working towards defining and implementing a standardized callout to support this. Other identified goals are improvements to client side authorization and adding finer-grained security support for resource properties

9.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Simplify the CAS administration interface	usability, manageability		CAS/SAML Utilities	4063
Improve GSI-related error reporting and diagnostics	usability, manageability	TeraGrid		4064
Extend GT authentication mechanisms	community expansion	OTP, PIV and Kerberos users		4065
Simplify the process of obtaining and maintaining credentials	usability, manageability	TeraGrid		4066
Improve runtime security audit logging	functionality			4067
Create secure-logging service and audit framework	functionality			4068
Simplify security policy administration	usability, manageability	TeraGrid		4069
Add call-out interface for attribute-based authz and delegation	functionality	TeraGrid		4070
Add attribute service provider interface	functionality			4071
Add authz service provider interface	functionality	TeraGrid		4072
Centralize management and provisioning of security-related info	usability, manageability			4073

Simplify policy information publishing and discovery	usability, manageability	TeraGrid		4074
Simplify VO lifecycle management	usability, manageability			4075
Mirror client and server authz processing	functionality			4076
Make authz processing consistent across java and C implementations	robustness	TeraGrid		4077
Improve fine-grained authz of access to RPs	usability, manageability			4080
Provide context support for the roles and VO associated with service invocations	functionality			4081
Improve documentation, tutorials, best practices	usability, manageability			4082
Resolve interoperability issues between the EGEE and GT delegation services	community expansion	EGEE		4083

10 Documentation

10.1 Current Status

GT Manuals for the 4.0 release introduced a consistent organization of information for all components in the GT distribution. We also used an initial implementation of DocBook XML. The priority was providing 'buckets' for various types of information (interfaces, commandlines, configuration, etc) geared towards specific types of users (end-users, sys admins, developers) as well as higher level 'key concepts' guides. Users were happy with the breadth of information available for each component - as well as the improved usability of the website. In current and near term work, starting with the development docs for upcoming releases, the emphasis has shifted from simply collecting information in these buckets, to refining it for better readability (improved flow, more graphics, tweaking the organization, editing, improved key concepts). We're also getting more familiar with DocBook, enabling the addition of new features. Future enhancements to include: specialized indexes (for "how-to" guides, as well as general indexing), automatic glossaries, improved integration of information between components, and pdf output generation.

10.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Update docbook DTD	functionality		GT manuals	4023
Clean up docbook html	robustness		GT manuals	4024
Update general component-level doc structure	usability, manageability		GT manuals	4025
Update MDS doc structure	usability, manageability		GT manuals	4026
Update Security doc structure	usability, manageability		GT manuals	4027
Documentation requirements analysis	user support		GT manuals	4028
Standardize FAQs	usability, manageability		GT manuals	4029
Create an author primer	robustness		GT manuals	4030
Standardize API docs	usability, manageability		GT manuals	4031
Automatically update imbedded snippets of GT source	robustness		GT manuals	4032
Create PDF version of manuals	user support		GT manuals	4033

11 Packaging/builds

11.1 Current Status

With the release of GT 4.0.0, we introduced a new configure/make installer based on our GPT metadata. Users have indicated a higher level of satisfaction with the new GT installation experience. Current work focuses on expanding our range of binary platforms, as well as reducing build times from source. Our delivery methods have been expanded to include pacman for communities like OSG and TG; we are considering making that packaging option available to the community as a whole. We are also investigating the issue of making Windows-friendly distributions.

11.2 Potential Tasks: Summary

Additional detail regarding entries can be found in Appendix A, as well as bugzilla.globus.org.

Item Description	Benefit	Specific Beneficiaries	GlobDev Projects	ID
Create windows-friendly installers	user support		globus toolkit	4276
Create multi-processor-friendly installers	performance, scalability		globus toolkit	4277
Improve automated build/test reporting	robustness		globus toolkit	4278

Appendix A: Potential Tasks: Details

The following table contains additional details on proposed tasks. Tasks are sorted by ID.

Bug ID	Product	Short Description
3971	MDS4	MetaScheduling information MetaScheduling information including queue data from PBS, Torque and LSF, cluster monitoring data for Nagios, CluMon, Ganglia, and Hawkeye a. Includes adaptation to core information provider structure for ease of use and flexibility b. Initial primary user – TeraGrid c. Funding – TeraGrid GIG, NMI Performance Inside d. Status as of Dec 4 2005 – Campaigns written and staffed, preliminary demo and deployment available, full rollout expected early 2006 (JMS – write catchup campaign for LSF and Hawkeye) e. Meets CDIGS deliverable - I7 (Tools that facilitate the rapid integration of domain-specific information sources into GT information services infrastructure) and I3 (Create interfaces to external monitoring systems such as UDDI, Nagios, MonaLisa, NWS, Clarens, etc.)
3972	MDS4	Performance benchmarks for clusters Performance benchmarks for clusters, including MPI and PetSC a. Includes development of an information provider that can run on a backend node b. Initial primary user – Omar Ghattas heart modeling code/ TeraGrid c. Funding – Heart modeling Grant (joint with Keyes and Gropp) d. Status – Campaigns written (ref #) but not staffed e. Meets CDIGS deliverable - I7 (Tools that facilitate the rapid integration of domain-specific information sources into GT information services infrastructure)
3973	MDS4	Persistent Index Server Persistent Index Server a. Database backend for Index service b. Initial primary user – requested by Globus Consortium c. Funding – Globus Consortium through ISI d. Status - ?? e. Meets CDIGS deliverable - I5 (Improve scaling and performance of the GT aggregation services)
3974	MDS4	Query interface for metascheduling info MetaScheduling query interface for WebMDS – basic pull-down query page for users to make metascheduling decisions a. Requested by TG in lieu of their automatic metascheduling approaches b. Initial primary user- requested by TG, need to locate user group c. Funding – TeraGrid, CDIGS d. Status – unwritten campaign e. Meets CDIGS deliverable - I2 (Enhanced tools for querying and displaying

		aggregated data)
3975	MDS4	Trigger Service easy how-to Trigger Service easy how-to a. Incorporate what ESG has done to Trigger service back into release as patch on 4.0.1 b. Initial primary user – ESG, should recruit a second c. Funding – NMI performance inside, CDIGS d. Status – unwritten campaign e. Meets CDIGS deliverables - I2 (Enhanced tools for querying and displaying aggregated data) and I1 (Tools for streamlining deployment of grid information infrastructure)
3976	MDS4	Archiver Archiver a. Basic archive service for gt4 b. Initial primary user- need to evaluate c. Funding – NMI Perf Inside, CDIGS d. Status – unwritten campaign e. Meets CDIGS deliverable - I6 (New archiver service)
3977	MDS4	GridFTP performance data GridFTP performance data a. Information provider for GT4 GridFTP b. Initial primary user – SCEC c. Funding – NMI Perf Inside, CDIGS d. Status – unwritten campaign, need to discuss with SCEC, Bill A. e. Meets CDIGS deliverable - I4 (Improved information published by GT services)
3978	MDS4	Inca Information providers Inca Information providers a. Will need infrastructure for accounts b. Initial primary user – TG c. Funding – NMI Perf Inside, CDIGS, TeraGrid(?) d. Status – unwritten campaign, need to gather requirements e. Meets CDIGS deliverable - I4 (Improved information published by GT services), I3 (Create interfaces to external monitoring systems such as UDDI, Nagios, MonaLisa, NWS, Clarens, etc.), and I7 (Tools that facilitate the rapid integration of domain-specific information sources into GT information services infrastructure)
3979	MDS4	C-APIs for Index service and Trigger Service b. Initial primary user – need to find out c. Funding – NMI Perf Inside, CDIGS d. Status – unwritten campaign, need to gather requirements e. Meets CDIGS deliverable - I7 (Tools that facilitate the rapid integration of domain-specific information sources into GT information services infrastructure) and I1 (Tools for streamlining deployment of grid information infrastructure)
3980	MDS4	Clover coverage for MDS4

		Clover coverage for MDS4 a. Need to understand test coveage, then extend unit tests b. Initial primary user – CDIGS requirement and needed for Core Component status for GT Distribution Project c. Funding – CDIGS d. Status – unwritten campaign e. Meets CDIGS deliverable - I5 (Improve scaling and performance of the GT aggregation services)
3981	MDS4	Improve Index query performance Improve Index query performance a. Optimized protocols and interfaces for very large indexes b. Initial primary user – several requests from friends mailing lists form users of large indexes, including Pegasus – specific application group to be identified c. Funding – CDIGS d. Status – unwritten campaign e. Meets CDIGS deliverable - I5 (Improve scaling and performance of the GT aggregation services)
3984	RFT	Implement Priorities in transfers Currently, RFT treats all data transfer requests uniformly, but in practice, some transfers are more important to the VO or user than others. To address this, we will extend RFT to provide for transfer semantics based on priorities.
3985	RFT	RFT as a Data Scheduler. Incorporate functionality of a scheduler into RFT to schedule file transfers.
3986	RFT	Support for different transfer protocols Right now RFT supports only third-party gridftp transfers. It would be beneficial to users if RFT supports other transfer protocols like ftp,https,srb etc.
3987	RFT	Support for multiple sources and partial file transfers. A file can be located in more then one site. Users should be able to list all the sites in which the file is available. Two scenarios are possible here: 1. RFT can get a portion of file from each site 2. RFT tries to get a file from one site and if that fails get it from the next site in the list. Adding partial file transfer capabilities to RFT will make a high-level service invoke RFT to get a portion of file from a site.
3995	Replica Location	Embedded SQL database backend Implement an embedded SQL database backend: Allow a simple local deployment of the RLS server that does not require complex separate installation of a third- party database such as MySQL or

		PostgreSQL.
3996	Replica Location	WS-RF interface to RLS services Design a WS-RF interface to RLS services; develop a prototype of the interface; evaluate this and use it to drive standards development
3997	Replica Location	Database connection pooling Database connection pooling: investigate whether to implement this ourselves or whether we can get the functionality we need from supporting ODBC infrastructure
3998	Replica Location	Improve RLS documentation Improve RLS documentation: We need to update the documentation to reflect our recent greater emphasis on the use of binaries for installation. Also need to update database documentation, with particular emphasis on ODBC libraries such as UNIX ODBC. We will also review and update command line and API documentation.
3999	Replica Location	Decouple Data manipulation (SQL) statements Decouple Data manipulation (SQL) statements from code: Currently, data manipulation logic (SQL code) is embedded in rest of C code. Instead, we will put SQL code in configuration files that can be changed by end user or at installation time
4000	Replica Location	Memory management Memory Management: Revise memory management code to reuse memory instead of repeatedly allocating and freeing it, e.g., in connection structures.
4001	Replica Location	Separation of RLS content among VOs Explore adding functionality to extend the specification of a DN to specify a virtual organization name when connecting to an RLS server; enforce separation of RLS content among separate VOs, thus allowing multiple VOs to share RLS infrastructure while limiting access to mappings to users in a particular VO.
4002	Replica Location	Dynamic configuration of WS-RF RLS service Dynamic configuration of WS-RF RLS service: prototype the use of resource properties and subscription/notification mechanisms via the RLS WS-RF interface to support dynamic reconfiguration of RLS servers.
4003	Replica Location	Fine-grained authorization

		Explore using WS-RF security framework to provide additional fine-grained authorization support by associating policy identifiers with RLS entries. During this 6-12 month period, will iterate on the design issues and possibly do some initial prototyping, as driven by application requirements.
4004	Data Replication	DRS feature enhancements Gather user feedback on initial DRS implementation and use this to drive feature enhancements.
4005	Data Replication	Add persistence to the DRS resource Add persistence to the DRS resource; this will allow the state of DRS requests to be maintained after DRS failure.
4006	Data Replication	Push-based replication in DRS Define a push-based interface to the DRS and integrate this functionality into the DRS implementation.
4007	Data Replication	Policy-based replication services Policy-based replication services: design and prototype interfaces that allow a service to specify the replication policies it supports via resource properties. Policies include push-based vs. pull-based replication, which catalogs should be updated, failure recovery polices, etc.
4008	Data Replication	Architecture discussions on higher-level services Continued architecture discussions with application communities regarding requirements for higher-level data management services, including consistency services, validation services, etc.
4010	CoG jglobus	3rd Party Software Updates Description: Upgrade to the latest releases of 3rd party software such as BouncyCastle, Log4j, etc. Benefits: Latest bug fixes and new features provided by the 3rd party libraries.
4011	CoG jglobus	GridFTP Library Improvements Description: Improve the GridFTP library to use the new non-blocking I/O Java API to improve scalability and reduce the number of threads used by the library. Benefits: More robust and scalable GridFTP client library.
4012	Core WS Schema	Standard WSRF/WSN Specifications

		Description: Upgrade to the latest (official) WSRF/WSN specifications. Benefits: The code based on the official standard schemas instead of older drafts.
4013	Core WS Schema	WSDL/WS-I Validation Description: Automatic tests to perform WSDL and WS-I BP validation on the core schemas. Benefits: Properly formed and validated schemas. Also, quick detection of WSDL/WS-I validation problems in core schemas.
4014	Java WS Core	JMS Support Description: JMS support for notifications. Benefits: Through the JMS integration, Java WS Core will have support for reliable messaging (for example, the notification messages will not get lost) and also it will get the pull support. Also, since JMS is part of the J2EE specification, it will make it easier to integrate the GT services with the application servers.
4015	Java WS Core	WS-Enumeration Support Description: Provide an efficient way to iterate over large data sets such as resource properties, resource property document or query results. Benefits: The clients will be able to request and work on part of the data at a time instead of the whole set. That will reduce the memory used by the server and the clients as they will not have to serialize/deserialize one big SOAP message.
4016	Java WS Core	Notification Improvements Description: Enhance the server-side notification infrastructure with periodic notifications and a mechanism to throttle the rate at which notifications are sent. Improve the client-side notification library with support for pull based notifications, and a mechanism for receiving notifications that can fallback to polling (potentially with exponential back off) if push notifications cannot be delivered. Benefits: More scalable and robust server-side notification implementation. Clients will be able to receive notifications even if they are behind firewalls.
4017	Java WS Core	Subscription Improvements Description: Add database support to the subscription code. Improve efficiency of the subscription code especially when removing all subscriptions associated with a particular resource when that resource gets destroyed. Benefits: The database support will enable the services to keep their entire state in a database. Improved performance and efficiency of subscription code especially when

		destroying a resource with a large number of subscriptions.
4018	Java WS Core	3rd Party Software Updates Description: Upgrade to the latest releases of 3rd party software such as Axis, Xerces, Xalan, etc. Benefits: Latest bug fixes and new features provided by the 3rd party libraries.
4019	Java WS Core	Remote Debugging Description: Provide a way to remotely examine log files, change logging settings, get container information such as memory, number of threads, etc. Benefits: Ability to remotely monitor and debug the container and the services.
4020	Java WS Core	Dynamic Deployment I Description: Provide dynamic deployment capabilities to the Java WS Core standalone container. Benefits: Ability to deploy new services, undeploy existing services, or update services/code dynamically without restarting the container.
4022	RFT	Better Performance, Throughput and Scalability Alternative ways to submit huge transfer requests should be investigated to improve scalability. One way would be to send list of transfers as a binary data across wire. More optimizations can be done by reducing the frequency of database updates. This is important as it results in better performance. Interested users: Ann, Rob Schuler
4023	Documentation	Update DocBook DTD Description: Update the DocBook XML DTD to at least version 4.4. Benefits: Allows us to implement new features including better indexing, automatic glossaries and the ability to manipulate the display of TOC levels.
4024	Documentation	Clean up DocBook HTML Output Description: The HTML output of the DocBook documentation (GT version 4.0+) includes some odd markup - it reiterates the php header. When we have a newer version of DocBook, we should be able to reconfigure how the php headers are output so this does not happen. Benefits: Cleaner and easier-to-read code. Go further towards being XHTML-compliant.
4025	Documentation	Update component-level documentation structure - General Description: Since the first major release of our new documentation structure, there are a few structural issues that have been brought up

		that include: * (bug 2133) add fragment for non-WSDL (framework level) protocols * (bug 2134) update [component]_Interface_WSDL_Frag.xml fragment so that faults are further subdivided by operation * (Ian's request) investigate getting rid of fact sheets (or implementing them somewhere else, such as under the GTx.xFriends/ directory) * embed public interface frags directly into developer's guides (right now there is a section in the developer's guide that points to the standalone Public Interface Guide, but there's no reason why we can't go ahead and embed the frags within the guide, saving users a few clicks) * (bug 2675) command-line tools fragment should be broken out by end user vs admin using the vol number * speak to IPv6 compliance (bug 2232) Benefits: These updates either streamline existing content or add even more useful content per component.
4026	Documentation	Update component-level documentation structure - MDS Description: Since MDS has different requirements than most other technologies, need to analyze the structure of its documentation. We also need to add a component that documents usefulrp/information provider (bug 3872). Benefits: Clearer documentation - makes it easier to find information.
4027	Documentation	Update component-level documentation structure - Security Description: Since security has different requirements than most other technologies, need to analyze the structure of its documentation. Benefits: Clearer documentation - makes it easier to find information.
4028	Documentation	Analyze content for each component Description: For 4.0, we made a quantum leap in the amount of standardized content available for each component. But we need more analysis of the unique documentation needs of each component. We need to communicate with the different audiences (users, admins, developers) of each component to find out what topics they still need or are confused about and come up with 'recipe'-style documentation that we don't have yet. Some other content issues include: * improve key concepts guides (need a lot more graphics, less technical jargon - should start with 'beginner' info and then have a more advanced section for developers on supplemental, separate pages) * more diagrams (translating TC's sketches/power point presentations) * more tutorials and 'entry' documentation * more samples Benefits: More complete and comprehensive documentation that directly satisfies readers needs. Should enable more effective use of GT and fewer discuss messages.
4029	Documentation	Standardize FAQs

		Description: We need a process and place for FAQs per technology as well as general FAQs per stable version (bug 2236). Benefit: Cut down on discuss messages.
4030	Documentation	Primer for internal documentation processes Description: Need a primer that describes the doc structure and how to work with/write our documentation. Benefits: Helps Technology Coordinators be more efficient writers; clears up confusion about how the doc is structured or how to best use the most common DocBook tags.
4031	Documentation	Standardize API docs Description: Need to standardize how we process api docs, appoint people to be in charge of javadocs or doxygen processes at release time, improve the look and feel via stylesheet (bug 3356), group javadocs by component (bug 2142) and investigate ways of improving or even automating this process. Benefits: Avoids confusion at release time about who does what, improves usability of API.
4032	Documentation	Investigate automatically updating documentation with actual globus code Description: Investigate ways to automatically grab actual code for documentation purposes and insert into DocBook XML (for rps, wsdls, schema, etc...) Benefit: Cut down effort and time expended by Technology Coordinators in providing documentation. Code represented is more likely to be accurate.
4033	Documentation	Resolve DocBook errors for PDF creation Description: Correct errors for creating printable version (pdf). Benefits: Cuts down processing time and results in automatic creation of PDFs that handles URLs and cross-references well. PDFs have been requested for a long time.
4036	GRAM	Keep GRAM test suites up to date There are a number of test suites that are relied on to verify that GRAM is functioning and performing satisfactorily. These tests include unit tests, integration tests and performance tests. These tests must continue to be improved and kept up to date as GRAM evolves.
4037	GridFTP	Overall GridFTP Roadmap This bug exists simply to provide the "root" of the GridFTP Roadmap. It has dependancies on all the other Roadmap items, which

		have depenancies on the Campaigns, so the depenancy view of this bug provides a nice high level overview of the state of GridFTP development.
4038	GRAM	Make GRAM tests available to grid projects for use in their grid deployment testing Projects deploying GRAM in their grid infrastructure should be able to use the GRAM test suites to validate GRAM has been successfully deployed.
4039	GRAM	Improve scalability of the WS GRAM service WS GRAM max scalability (the number of jobs that it can be managing simultaneously) is currently limited to 32,000 jobs. That limit should be improved.
4040	GRAM	Improve throttling capabilities of the WS GRAM service Improve throttling capabilities of the WS GRAM service to ensure that the service runs acceptably under large job bursts / high throughput.
4042	GRAM	Improve the configuration and setup process of GRAM Add configuration options and features to WS GRAM to improve the installation/deployment/ configuration/setup process. This will make WS GRAM easier to deploy/incorporate into grid project software stacks.
4043	GRAM	Improve performance of processing an individual job Improving performance of an individual WS GRAM job. This would have obvious benefits to clients submitting large numbers of jobs. And would likely improve throughput, e.g. processing multiple jobs simultaneously.
4044	GRAM	Support state-of-the-art specifications related to job submission and management The Job Specification Description Language (JSDL) has been approved by GGF (http://www.ggf.org/ documents/GFD.56.pdf). We believe JSDL will be a standard that clients and tools will migrate to for grid job submissions. As such, WS GRAM should be extended to support jobs expressed as JSDL documents.
4045	GRAM	Support advanced reservation GRAM provides an API enabling clients to submit and monitor jobs to a variety of resource managers. Some resource managers support the ability to reserve computation resource for future use. GRAM in

		GT 4.0 does not support advanced reservation capabilities. GRAM should be extended to support the ability to make a reservation as well as specify a reservation for use in a job submission.
4046	GRAM	Dynamic service deployment/restart of persistent jobs A GRAM job is only executed once in the job's lifecycle. Execution completes on failure or normal termination. The proposal is to add the ability to GRAM to restart/resubmit the job before moving on from the execution phase of the job's lifecycle. New job directives will be needed to express the policy for which a job will be restarted to provide a level of persistence. For example, "restart/submit my job after an error, repeat for 5 times"
4047	GRAM	Investigate and improve per-job metadata There is information about a job that could be published by GRAM that would be beneficial to clients. For example, some clients submit jobs that are a daemon/service, then the client must connect to the daemon. Today the client has no way to discover on what node the resource manager started the daemon. Providing that information would make it easy for a client to locate and connect to the daemon. What other information does it make sense for GRAM to publish about a job: rusage, local RM job id, worker nodes allocated, ...? This could include providing tools and an interface for a job to publish metadata about itself. Today a client might have to consult multiple services/sources to get the information they need. At times this is done in ad hoc ways involving job wrappers or scripts.
4048	GRAM	Improve job environment management Make enhancements to WS GRAM to support the specification and creation of a job environment for use in a job submission. For example, setting job limits, creating a sym link, mounting a file system, creating sub directories, staging files based on site configuration/layout. In addition, creating a generic specification for using installed software, implemented by softenv http://www.ncsa.uiuc.edu/UserInfo/Resources/Hardware/CommonDoc/softenv.html or modules.
4049	GridFTP	Improve robustness and scalability of Striped GridFTP server Description: The Striped GridFTP server in GT 4.0.x requires that the front end (the process that external clients connect to and handles the control channel protocol) be configured with a static list of backends (the nodes that actually move the data) available to it. It will always use every available node in every striped transfer, and if any one of the configured backend nodes is down, the entire striped server is down. To improve upon this situation, we intend to: - investigate the

		scalability of the front end node. How many backends can a front end take? What happens if you have more? What is the best way to make multiple front end processes appear as a single entity to the outside world? - investigate making the backend nodes be able to dynamically register and de-register themselves with the frontend node. I.e., if a node disappears, it at most causes a failure of an ongoing transfer, but it then simply drops out of the resource pool for future transfers. - investigate ways of dynamically assigning resources to a request. This could be as simple as a fixed configuration, some sort of policy language / scripting thing, or a generic interface where arbitrary schedulers could be plugged in to make such decisions. Why is this important: These changes will make it easier to add or remove resources from your GridFTP pool trivial, easing administration (dynamic registration). The dynamic de-registration means the server is more robust to failure. Dynamic resource allocation can make your utilization of resources more efficient, allow you to throttle incoming resources, etc.. Communities we believe are interested in this: The TeraGrid, the Condor Project (Stork), OSG, and the high energy physics community in general, though this work makes the striped server more resilient to failures, so anyone running a striped server should benefit from it.
4050	GRAM	Improve performance of processing many simultaneous jobs This is one of the most important ways that WS GRAM is measured, job throughput. How many job can be processed per minute? Identify bottlenecks and invent ways to improve throughput.
4051	GridFTP	Add QoS and Managed Interfaces to GridFTP Description: This is a large topic and touches on several others, but essentially, at a very high level, the idea is to turn GridFTP into the resource manager for a storage system. I.e., it will provide the "plumbing" by which you can control transfers of data to and from your system. As a general guiding principle, we want to focus on enforcement mechanisms, develop clean, useful scheduling interfaces and then let the various schedulers plug in. This is closely related to the Storage Resource Manager (SRM) project, and when we are done, it should be trial to implment an SRM over this, if the interface is not already SRM. Some of the specific areas we plan to address: - bandwidth limiting. You should be able to specify a *maximum* bandwidth (no guarantees on a minimum). - disk space reservation. At a minimum, reserve space on a file by file basis. Investigate how to work with a generic reservation ID if there is a pre-existing reservation. - The level of complexity required basically precludes this being "force fit" into the existing GridFTP protocol, so we will be defining a new, probably SOAP based control channel protocol. -

		We will need some element in our system to have global knowledge of all requests to this resource (as opposed to the current circumstance, where each server runs completely independantly of each other). This implies queuing requests and is related to bug 4049 (front end / back end allocation). Why is this important: Large sites can be overwhelmed by having too many servers starting up, over subscription of disk bandwidth, etc.. Particularly when you have large workflows starting up, and you could be having thousands of processes competing for incoming bandwidth, storage space, etc.. Communities we believe are interested in this: Open Science Grid and the High Energy Physcis Community in general. Any large storage site that wants to manage access, similar to the way the manage access to compute resources.
4052	Toolkit Internals	Overall Toolkit Internals Roadmap This particular product is at the bottom of the C software stack. Changes here can have a significant impact. This code has been fairly stable for a long time, and changes only tend to be made here for platform portability reasons. We have no known feature requests, either from user communities or developers. Our current plan is to simply maintain this as necessary in terms of bug fixes.
4053	XIO	Overall XIO Roadmap This bug exists simply to provide the "root" of the XIO Roadmap. It has dependancies on all the other Roadmap items, which have depenancies on the Campaigns, so the depenancy view of this bug provides a nice high level overview of the state of XIO development.
4054	XIO	Investigate Driver Descriptor Description: This feature would enable specific features/semantics of a driver to be determined. For instance, ordered vs unordered data, reliable vs unreliable, etc.. Why is this important: It could allow applications to determine if a driver stack met its needs and could possibly allow an intelligent system to reconfigure the stack to meet its needs. Communities we think are interested: We had a request for determining POSIX compliance from the ROMIO/PVFS team, though that depends on what you are operating on. Internally, we believe it could enable some "intelligence" in higher level drivers and the xio_copy library (see the xio_copy roadmap item)
4055	XIO	Add the ability for one driver to query the stack for other drivers present Description: A driver could determine what other drivers are on the stack and possibly change their functionality accordingly. This would work in conjunction with the driver descriptor. A driver can use this

		feature to determine what other drivers are on the stack and then query the descriptor to determine what properties it has. Why is this important: It allows "intelligence" to be built in to higher level xio functions, such as xio_copy and eases the development requirements on developers trying to use the XIO framework. Communities we believed are interested: We propose this to as an "ease of use" feature.
4056	XIO	Allow alteration of the stack while the handle is open Description: This feature would allow the driver stack to be modified while the handle is open. We expect this to be a very difficult feature to add. Adding a driver to the stack could be alternately accomplished by using the pipe driver (see its roadmap item), but that still leaves removing a driver from the stack. Why is this important: It removes a constraint and provides flexibility. It would be useful in the xio_copy library. Communities we believe are interested in this: This is an internally generated idea, and until a specific need for *removing* a driver from the stack is found, we will likely opt for the pipe driver.
4057	XIO	xio_copy library Description: This is a library that would sit on top of XIO. Given two XIO handles, a source and a destination, it would very efficiently read from one and write to the other. This requires that the semantics of the two stacks be compatible. In the simplest case, it can simply be left to the developer to ensure that this is the case. However, using other features mentioned in this roadmap, the library could, in theory, query the semantics of the destination stack, and then dynamically add necessary drivers to the source stack to make them compatible. For instance, if the destination was ordered, and the source unordered, xio_copy could dynamically add an ordering driver to the stack. Why is this important: Reading from one source and writing to another is a very common scenario. If we could hide this behind a very simple, very efficient API, this would make developing higher level apps much easier. For instance, with this functionality, globus-url-copy could be implemented largely as an xio_copy between two handles. What communities are interested in this: This is an internally generated idea, though it would make globus-url-copy much easier to extend and maintain.
4058	XIO	develop an xio-pipe driver Description: The concept here is that this would be a transport driver (bottom of the stack) that simply passed data through from an existing stack. This could be used to add an additional driver onto an existing stack without actually altering the existing stack. You build a

		stack with the pipe driver, and the driver with the additional functionality required, say compression. The pipe driver is given the existing stack, say GSI/TCP, by using the handle associated with the stack containing the pipe driver, you effectively added compression to your existing GSI/TCP stack. Why is this important: It provides flexibility. You can alter the behavior of a stack, even if, for instance, you can't know all the requirements when you open the stack. What communities are interested in this: This is an internally generated idea, though it does provide flexibility and would be easier than trying to dynamically alter the stack after the handle is opened.
4059	XIO	Make developing new drivers easier Description: Drivers are complicated, and for reasons of efficiency, we made design decisions that will let developers "get themselves into trouble". We propose to functionality to make it easier to use from 3rd party APIs, particularly blocking APIs. Development of additional tools such as driver test suites, driver development workshops and tutorials, etc.. Why is this important: The XIO framework provides considerable power and flexibility, but ultimtely we need a community of driver developers for this to truly be successful. What communities are interested in this: Various potential driver developers we have talked to.
4060	XIO	Actively seek out and engage communities to develop drivers Description: XIO is a powerful framework and abstraction, but its power is multiplied by the addition of new drivers. We want to engage with, and assist key communities in the development of new drivers. Why is this important: The greater a selection of drivers there are available, the more useful XIO becomes. What communities are interested in this: The network protocol research community could benefit from using XIO, and any drivers developed makes their protocol available easily to anyone using XIO, including GridFTP. We also want to engage the MPICH/MPICH-G communities and to incorporate XIO.
4061	GridFTP	Continue to improve routine testing Description: Improved stability / quality of all components was a focus for GT4.0. We intend to improve upon the progress we have already made in these areas. This will include tasks such as: - ensuring that the client test suite is maintained, new tests are added as shortcomings are identified and as new features are added - working with the NMI test and build community we intend to add the ability to co-schedule two platforms and run tests between them. We will vary the architectures and versions of these platforms ensuring that we get cross version and cross platform validation. Why is this

		important: GridFTP is depended on in production environments and we must maintain the stability and existing functionality while extending our feature set. Communities we believe are interested in this: Everyone using GridFTP for more than one off experimentation.
4062	GridFTP	Improve the performance of GridFTP on lots of small files Desciption: Currently, as the size of the file drops, so does the overall performance of GridFTP. However, large data sets that consist of many small files is a common usage scenario. We intend to investigate options, potentially prototype one or more, and then choose one to implement for a stable release of GridFTP. The list of things we are considering includes, but is not limited to: - "on the fly tar" - This was, to the best of my knowledge, originally proposed by IBM in something they called dynamo. The idea is to make lots of small files into fewer, bigger files. The LIGO project is also working on this and we will engage with them as well. - GridFTP v2 defines MODE X which includes a transaction ID to enable pipelining of data on teh data channel. This is another option to investigate. - Writing a "pre-fetch" Data Storage Interface that could resond to a single file request by fetching multiple files (tarred or not) and then responding to subsequent requests from a local cache. This would be ideal if the access pattern is sequentially running through a list of files. Why is this important: Several communities have this usage scenario and are suffering from poor performance. Communities we believe are interested in this: LIGO, most of the astro physics communities.
4063	CAS/SAML utilities	Simplify the CAS administration interface
4064	GSI	Improve error reporting and diagnostics for GSI-related config and runtime problems
4065	GSI	Extend GT authentication mechanisms to include secure passwords, one-time passwords, PIV-tokens and Kerberos
4066	GSI	Simplify the process of obtaining credentials and configuring security settings
4067	GSI	Log events in the GT-runtime relevant to security audits
4068	GSI	Develop an Secure-Logging service and Audit framework to capture and corrolate events
4069	GSI	Simplify the process of administering security policy information like attributes, roles, access rules and delegation
4070	GSI	Define and implement a standardized authorization call-out interface that supports attribute-based authorization and delegation
4071	GSI	Define and implement a pluggable attribute service provider interface

4072	GSI	Define and implement a pluggable authorization service provider interface
4073	GSI	Simplify security and trust-root configuration through a directory/attribute-service driven approach
4074	GSI	Simplify the policy-driven publishing and discovery of policy information through negotiation protocols
4075	GSI	Simplify the VO life-cycle management through infrastructure services for identifiers and metadata
4076	GSI	Mirror the client and server authorization processing
4077	GSI	Consistent attribute and authorization processing in Webservices (Java&C) and GridFTP
4078	GSI	Deploy VM technologies to provide more secure use of keys/secrets
4080	GSI	Provide flexible admin facilities for access control of the resource properties
4081	GSI	Provide context support for roles and VO in the policy enforcement
4082	GSI	Keep improving our Documentation/Tutorials/Best practices
4083	GSI	Sync our Delegation Service implementation with the European/EGEE efforts
4087	Dynamic Accounts	C client for Workspace-DA (WSS) including query and integration with existing services tasks: - C client with create and query interface - integrate&test query into the GridFTP authorization callout
4088	Workspace Service	Small features and bug fixes (TP 1.1) current enhancements and bug fixes (as filed in bugzilla)
4089	Workspace Service	Remote startup of VMs within the same TCB - implement/integrate image propagation within TCB - restructure workspace back-end so that it can start VMs on remote nodes - implement a simple node pool database reflecting resource capability
4091	Workspace Service	Security for VMs GRIM-like credential generation for VM-as-edge-service, so that services can be trusted by remote entities
4092	Workspace Service	Virtual Cluster - intgrate/clean up/implement support for aggregate workspaces - add support for persistance of aggregate workspaces - enhancements to node pool database
4093	Workspace	Fine-grain enforcement for workspaces

	Service	- modify resource allocation to enable requesting of fine-grain resource allocations and implement back-end - enable management of resource allocations - refinements to node pool database
4094	Workspace Service	Create and populate a community image library - basic "client's view" (CDDLM-like) schema (as opposed to meta-data) - basic access mechanism
4095	Workspace Service	Workspace image creation environment prepare an environment (including but not limited to suitably > configured pieces of hardware) where users would be able to configure > their VMs (potentially based on a library of images
4096	Workspace Service	Miscellaneous testbed preparation activities
4098	Workspace Service	Workspace image reconstruction VM images can be huge and their content can be largely repetitive (same OS, same community-specific layer, different app). In other words, they could be taking a lot of time and storage, but don't have to. So the objective is to chop them up into reasonable pieces (called partitions), make some mods to workspace metadata schema so that we could (a) securely reconstruct the original VM image and (b) the partitions could be repetitive enough to be interchangable. Once we have that, a site could cache the most frequently used partitions (partial images) and deal only with the delta. Furthermore, if we could use this with RLS and other related fancy tools we could move those images around quite efficiently.
4099	Workspace Service	IP management for workspaces Every VM needs a presence on the network, the question is how to get there. On a very basic level this involves management of a small amount of public IPs assigned by the site admin as described by en enhancement somewhere around here. A more sophisticated solution would be to involve virtual private networks.
4131	GridShib	Author and embed SAML assertions in MyProxy based on non-SAML authentication to MyProxy Author and embed SAML assertions in MyProxy based on a non-SAML authentication (any method supported by MyProxy). The authentication assertion will be used for querying the attrbute authority, the GT module will look for the embedded assertion in the certificate used to authenticate with the container. An example profile for this is available here: https://authdev.it.ohio-

		state.edu/twiki/bin/view/GridShib/MyProxyNonBrowserAttributePull This is a building block also, this mechanism can be used in other profiles.
4132	GridShib	Author and embed SAML assertions in MyProxy based on SAML authentication to MyProxy Author and embed SAML assertions in MyProxy based on a SAML based authentication to MyProxy. As in roadmap entry #(X), the authentication assertion will be used for querying the attrbute authority, the GT module will look for the embedded assertion in the certificate used to authenticate with the container. This will be necessary for IdP first profiles, and will enable portal/browser based use cases that rely heavily on the current Shibboleth single sign on profiles.
4133	GridShib	GridShib for GT module enhancements Continue to enhance the GridShib for GT module: a) adding support for retrieving embedded SAML assertions from proxies to be used for attribute authority queries (see roadmap entries Bug 4131 and Bug 4132), b) adding SAML2 metadata support (both generation and consumption), and c) porting the module to be compatible with the GT4.2 authorization framework.
4134	GridShib	IdP NameMapper enhancements Continue to enhance the GridShib for Shibboleth module: add support for queries based on the authentication assertions in roadmap entries Bug 4131 and Bug 4132
4135	GridShib	User managed IdP name mapper Write a web-based application that allows a client to bind two of its identifiers together on the IdP side, making a persistent identifier (DN) available for attribute queries from the current GridShib for GT module. This is mainly in order to support integration with systems such as OpenIdP (which could have a large impact).
4205	XIO	Make XIO easier for developers to use Description: To improve the adoption and use of XIO directly by other applications (as opposed to via other Globus tools), we intend to implement certain common tasks as "helper" utilities / drivers / features. Why is this important: XIO is powerful, but in many cases it requires signifcant amounts of works to accomplish "ordinary" tasks. We need to improve this situation. Who is interested: All developers using XIO will benefit. We have been in discussions with several network protocol researchers. This would allow them to more easily

		integrate their protocols. Also, in discussions with the PVFS team, they indicated an interest in querying the characteristics of other drivers in the stack.
4206	XIO	Support Additional Features required by Globus Data Management Tools Description: As Globus expands its support for Data Management, particularly making data management a "first class" managed resource like computation, some of the features will end up being best implemented in the XIO layer. Why is this important: The single biggest user of XIO is the Globus Data Management tools. Additionally, any functionality added at this level can be leveraged by any program that wants to use an Open/Close/Read/Write interface. Who is interested: The GridFTP development team.
4256	Metrics	Reach agreement on a baseline set of evaluation metrics for Globus software There are many possible mechanisms for evaluating the relevance, usefulness, and quality of Globus software. The first step in a coherent strategy for measuring these aspects of Globus software is to define an initial set of metrics that we will use to begin tracking the state of the software and its use over time. We will undoubtedly change and expand this set over time, but it will be useful to begin with a specific set of measurements and start tracking them so that we can see how useful (or not) the initial set is and use it as a starting point for further exploration. Our intention is to focus the initial set of metrics on data that we are already producing, collecting, and archiving, rather than devising new types of data. Future work will likely involve new ways to analyze and report on this data and new types of data to collect, archive, and analyze.
4257	Metrics	Begin open reporting of evaluation metrics We need to begin reporting on the status (and history) of specific evaluation metrics for Globus software in an open manner. Anyone who is interested should be able to obtain reports generated by the Metrics project. The contents of these reports will be determined by the members of the Metrics project with guidance (and in many cases with direct support) from the community.
4261	Metrics	Assess the reliability of current usage reporting mechanisms A number of GT4 components include usage reporting functionality that sends usage reports to a central receiver for archiving and analysis. We currently have no data on how reliable this reporting mechanism is. In order to use this data in our evaluation metrics effectively, we need to know how reliable it is. Areas that may or

		may not have issues include: o The usage reporting code (aka sender code) that generates and sends usage reporting messages. o The receiver that receives usage reporting messages. o The transport protocol used by the usage reporting mechanism (currently based on UDP) and the types of network on which people run Globus software o The database in which the usage reporting messages are stored and archived. o The tools (of which there are already several) that analyze the data.
4262	Metrics	Explore analysis possibilities for GT4 usage reporting data A number of GT4 components include usage reporting functionality that sends usage reports to a central receiver for archiving and analysis. We know what data each component is reporting (this is included in the early draft metrics report developed by Lisa, Mats, and Jarek), but we have not yet had a systematic exploration of how useful this data is (for each component and as a whole) or what other kinds of usage reporting data might be important to have in order to answer key evaluation questions.
4263	Metrics	Do more automatic, routine analysis of evaluation metrics data GT4 components and the Globus support infrastructure (website, bugzilla, mailing lists, CVS, etc.) currently produce a great deal of data that might be of use in developing evaluation metrics. We currently are not making much use of this data because we do not systematically analyze the data that is generated, and in some cases we don't even archive the data. The Globus community is--in several areas--unable to make use of the data because we don't provide open access to it. We can correct these issues (while keeping the workload manageable) by systematically putting in place a broader set of automatically executed, periodic analysis mechanisms that produce reports that are openly available to the community.
4276	Installation	Windows-friendly installers We should have a windows installation that is ws-core + windows-friendly jars.
4277	Installation	Make -j friendly 4.1 distributions We should release a 4.1 release with new GRAM/MDS functionality. Installer should be make -j friendly
4278	Installation	Test reports online daily We should get our nightly/daily build and test results up in a friendly way. Ideally we would rebuild after each couple of commits.
4285	MDS4	Usage Statistics for Index and Trigger Services

		Collect usage statistics for the Index and Trigger Services. This will provide evaluation metrics for MDS services.
4286	MDS4	Trigger Service Support for Usage Stats Monitoring Provide trigger service support (providers and scripts) for the usage statistics monitoring described in bug 4284.