- T he modern data center is now full of virtualization and virtual machines.
- I n using solutions like Citrix's Provisioning Server or Unidesk's layering software technologies, administrators are able to take workload provisioning to a whole new level.
- I magine being able to set a process that will kick-start the creation of a new virtual server when one starts to get over-utilized.
- N ow, administrators can create truly automated virtual machine environments where each workload is monitored, managed and controlled.

Cloud layer.

- T his is a new and still emerging field. Still, some very large organizations are already deploying technologies like CloudStack, OpenStack, and even Open Nebula. Furthermore, they're tying these platforms in with big data management solutions like MapReduce and Hadoop.
- O rganizations can deploy distributed data centers and have the entire cloud layer managed by a cloud-control software platform.
- E ngineers are able to monitor workloads, how data is being distributed, and the health of the cloud infrastructure.
- T he great part about these technologies is that organizations can deploy a true private cloud, with as much control and redundancy as a public cloud instance.

Data center layer.

- Although entire data center automation technologies aren't quite here yet, we are seeing more robotics appear within the data center environment.
- Robotic arms already control massive tape libraries for Google and robotics automation is a thoroughly discussed concept among other large data center providers. Many experts agree that eventually, data center automation and robotics will likely make its way into the data center of tomorrow.
- The need to deploy more advanced cloud solution is only going to grow.
- More organizations of all verticals and sizes are seeing benefits of moving towards a cloud platform. At the end of the day, all of these resources, workloads and applications have to reside somewhere. That somewhere is always the data center.
- In working with modern data center technologies administrators strive to be as efficient and agile as possible.
- This means deploying new types of automation solutions which span the entire technology stack. Over the upcoming couple of years, automation and orchestration technologies will continue to become popular as the data center becomes an even more core piece for any organization.

UNIT IV PROGRAMMING MODEL

Open source grid middleware packages – Globus Toolkit (GT4) Architecture , Configuration – Usage of Globus – Main components and Programming model - Introduction to Hadoop Framework - Mapreduce, Input splitting, map and reduce functions, specifying input and output parameters, configuring and running a job – Design of Hadoop file system, HDFS concepts, command line and java interface, dataflow of File read & File write.

4.1 OPEN SOURCE GRID MIDDLEWARE PACKAGES

- The Open Grid Forum and Object Management are two well- formed organizations behind the standards
- Middleware is the software layer that connects software components. It lies between operating system and the applications.
- Grid middleware is specially designed a layer between hardware and software, enable the sharing of heterogeneous resources and managing virtual organizations created around the grid.

The popular grid middleware are

1. BOINC -Berkeley Open Infrastructure for Network Computing.
2. UNICORE - Middleware developed by the German grid computing community.
3. Globus (GT4) - A middleware library jointly developed by Argonne National Lab., Univ. of Chicago, and USC Information Science Institute, funded by DARPA, NSF, and NIH.
4. CGSP in ChinaGrid - The CGSP (ChinaGrid Support Platform) is a middleware library developed by 20 top universities in China as part of the Open source grid middleware
5. Condor-G - Originally developed at the Univ. of Wisconsin for general distributed computing, and later extended to Condor-G for grid job management.
6. Sun Grid Engine (SGE) - Developed by Sun Microsystems for business grid applications. Applied to private grids and local clusters within enterprises or campuses.
7. gLight -Born from the collaborative efforts of more than 80 people in 12 different academic and industrial research centers as part of the EGEE Project, gLite provided a framework for building grid applications tapping into the power of distributed computing and storage resources across the Internet.

Grid Standards and APIs

- The Open Grid Forum and Object Management Group are two well-formed organizations behind those standards.
- OGSA - Open Grid Services Architecture
- GLUE - Grid Laboratory Uniform Environment
- SAGA - Simple API for Grid Applications
- GSI - Grid Security Infrastructure
- OGSI - Open Grid Service Infrastructure
- WSRE - Web Service Resource Framework
- The grid standards have guided the development of several middleware libraries and API tools for grid computing.
- They are applied in both research grids and production grids today.
- Research grids tested include the

- ✓ EGEE,
- ✓ France Grilles,
- ✓ D-Grid (German),
- ✓ CNGrid (China),
- ✓ TeraGrid (USA), etc.

- Production grids built with the standards include the
  - ✓ EGEE,
  - ✓ INFN grid (Italian),
  - ✓ NorduGrid,
  - ✓ Sun Grid,
  - ✓ Techila, and
  - ✓ Xgrid
- Software Support and Middleware
- The middleware products enable the sharing of heterogeneous resources and managing virtual organizations created around the grid.
- Middleware glues the allocated resources with specific user applications.
- Popular grid middleware tools include the Globus Toolkits (USA), gLight, UNICORE (German), BOINC (Berkeley), CGSP (China), CondorG, and Sun Grid Engine.

4.2 GLOBUS TOOLKIT (GT4) ARCHITECTURE

Part A	**1.Name any 4 services of offered by GT4 (Nov/Dec 2016)** **2.Write the Significant use of GRAM (AU/April/May2017)**
Part B	**1.What is GT4? Describe in detail the components of GT4 with suitable diagram (Nov/Dec 2017)** **2.Draw and explain the Globus toolkit Architecture (AU/Nov/Dec 2016)**

- The Globus Toolkit, started in 1995 with funding from DARPA, is an open middleware library for the grid computing communities.

- The toolkit addresses common problems and issues related to
  - ✓ grid resource discovery,
  - ✓ management,
  - ✓ communication,
  - ✓ security,
  - ✓ fault detection, and
  - ✓ Portability.
- The software itself provides a variety of components and capabilities. The library includes a rich set of service implementations.
- The implemented software
  - ✓ Supports grid infrastructure management,
  - ✓ provides tools for building new web services in Java, C, and Python,
  - ✓ builds a powerful standard-based security infrastructure and client APIs
  - ✓ offers comprehensive command-line programs for accessing various grid services.
- The toolkit is for sharing of resources and services, in scientific and engineering applications.
- The shared resources can be computers, storage, data, services, networks, science instruments.

4.2.1 THE GT4 LIBRARY

- GT4 offers the middle-level core services in grid applications.
- The high-level services and tools, such as MPI, Condor-G, and Nirod/G, are developed by third parties.
- It is been for general purpose distributed computing applications.
- The local services, such as LSF, TCP, Linux, and Condor, are at the bottom level.

- The functional modules help users to discover available resources, move data between sites, manage user credentials.
- GT4 is based on industry-standard web service technologies.

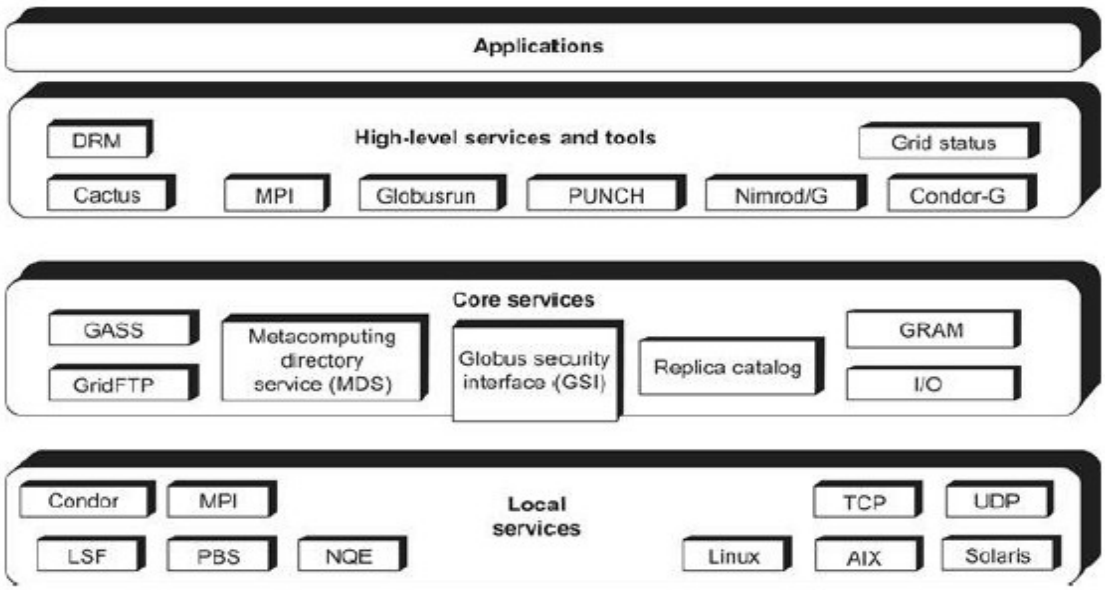


Fig :1 Globus Tookit GT4Architecture (supports distributed and cluster computing services)

4.2.2 FUNCTIONAL MODULES IN GLOBUS GT4 LIBRARY

Service Functionality	Module Name	Functional Description
Global Resource Allocation Manager	GRAM	Grid Resource Access and Management (HTTP-based)
Communication	Nexus	Unicast and multicast communication
Grid Security Infrastructure	GSI	Authentication and related security services
Monitory and Discovery Service	MDS	Distributed access to structure and state information
Health and Status	HBM	Heartbeat monitoring of system components
Global Access of Secondary Storage	GASS	Grid access of data in remote secondary storage
Grid File Transfer	GridFTP	Inter-node fast file transfer

- Nexus is used for collective communications and HBM for heartbeat monitoring of resource nodes.
- GridFTP is for speeding up internode file transfers.
- The module GASS is used for global access of secondary storage.

4.2.3 GLOBUS JOB WORKFLOW

- Typical job execution sequence proceeds as follows:
- The user delegates his credentials to a *delegation service*.
- The user submits a job request to GRAM with the delegation identifier as a parameter. GRAM parses the request, retrieves the user proxy certificate from the delegation service, and then acts on behalf of the user.
- GRAM sends a transfer request to the RFT (Reliable File Transfer),
- It applies GridFTP to bring in the necessary files.
- GRAM invokes a *local scheduler* via a GRAM adapter and the SEG (Scheduler Event Generator) initiates a set of user jobs.
- The local scheduler reports the job state to the SEG.
- Once the job is complete, GRAM uses RFT and GridFTP to stage out the resultant files.The grid monitors the progress of these operations and sends the user a notification when they succeed, fail, or are delayed.

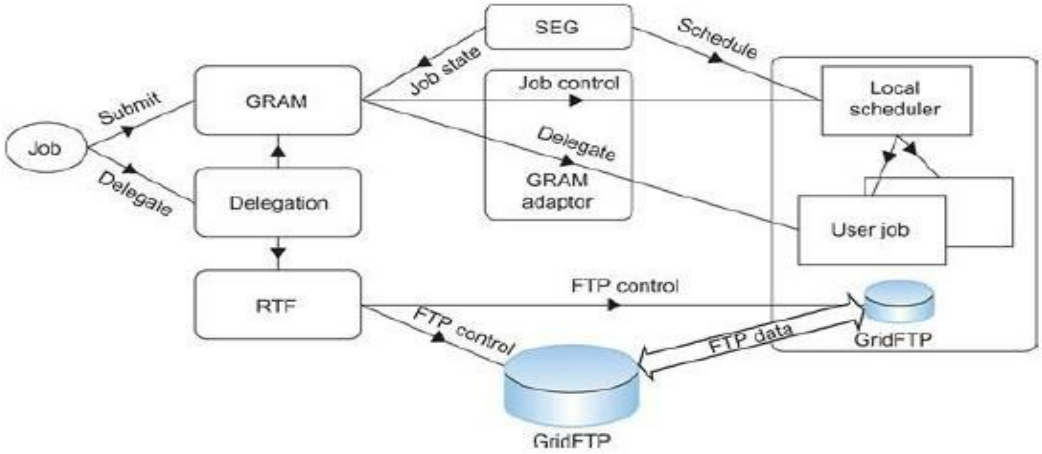
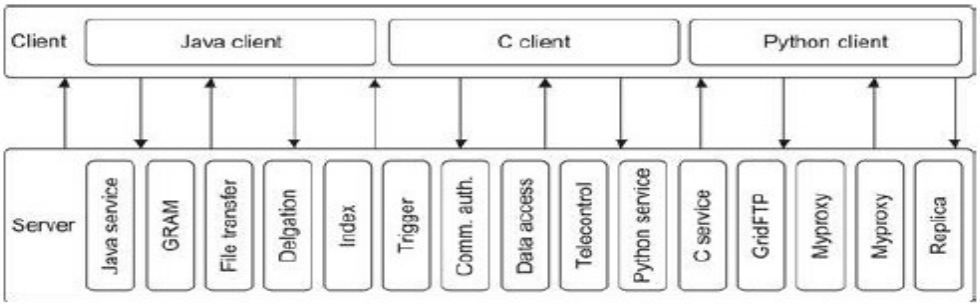


Fig :2 Globus job workflow among interactive functional modules

4.2.4 CLIENT-GLOBUS INTERACTIONS

- GT4 service programs are designed to support user applications.
- There are strong interactions between provider programs and user code.
- GT4 makes heavy use of industry-standard web service protocols and mechanisms in service description, discovery, access, authentication, authorization, and the like.
- GT4 makes extensive use of Java, C, and Python to write user code.
- Web service mechanisms define specific interfaces for grid computing.
- Web services provide flexible, extensible, and widely adopted XML-based interfaces.



- Client and GT4 server interactions; vertical boxes correspond to service programs and horizontal boxes represent the user codes.
- GT4 components do not, in general, address end-user needs directly. Instead, GT4 provides a set

of infrastructure services for accessing, monitoring, managing, and controlling access to infrastructure elements. The server code in the vertical boxes in corresponds to 15 grid services that are in heavy use in the GT4 library.

- These demand computational, communication, data, and storage resources. We must enable a range of end-user tools that provide the higher-level capabilities needed in specific user applications.
- Wherever possible, GT4 implements standards to facilitate construction of operable and reusable user code.
- Developers can use these services and libraries to build simple and complex systems quickly.
- A high-security subsystem addresses message protection, authentication, delegation, and authorization. The horizontal boxes in the client domain denote custom applications and/or third-party tools that access GT4 services. The toolkit programs provide a set of useful infrastructure services.

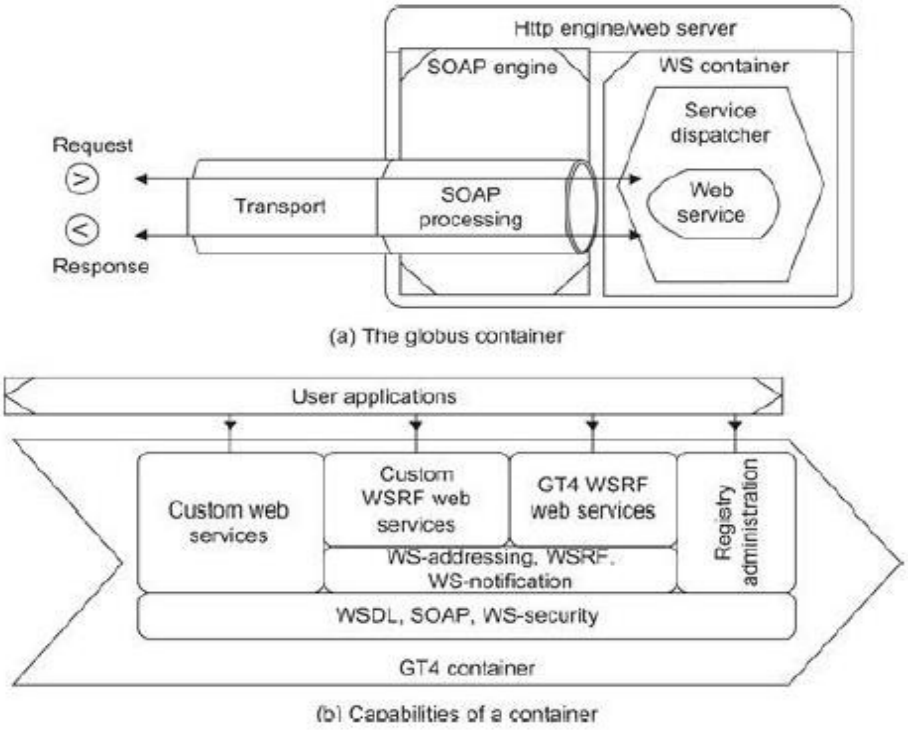


Fig:3 Globus container serving as a run time environment for implementing web services in grid

- Three containers are used to host user-developed services written in Java, Python, and C, respectively.
- These containers provide implementations of security, management, discovery, state management, and other mechanisms frequently required when building services.
- They extend open source service hosting environments with support for a range of useful web service specifications, including WSRF, WS-Notification, and WS-Security.
- A set of client libraries allow client programs in Java, C, and Python to invoke operations on both GT4 and user-developed services.
- In many cases, multiple interfaces provide different levels of control:

4.3 CONFIGURATION

step-by-step guide for preparing a machine and installing the Globus Toolkit as well as the steps required for a simple configuration

4.3.1 INSTALLATION AND CONFIGURATION OF GLOBUS TOOLKIT:

4.3.1.1 Globus toolkit Installation process:

- Create a new directory to install the Globus Toolkit. **If you need to delete your current installation, be sure to back up any configuration files, certificates and keys before doing so.**
- If you have previously run setup-gsi as root, there is no need to do so again, although it should not be harmful if you do.
- Any manual changes to configuration files in the Globus toolkit, $GLOBUS_LOCATION/etc will need to be migrated to the new installation.
- Host and service certificates not located in /etc/grid-security, such as the certificates used by MDS will have to be copied to a new location. More on this in the Verification section.

4.3.1.2 Environment Set up:

Before you build and install the Globus Toolkit, you need to setup your environment.
We will need to set the environment variable GLOBUS_LOCATION to the directory in which we plan to install the Globus Toolkit. You can do this using the following example for your shell:

- {csh} setenv GLOBUS_LOCATION <globus_install_dir>
- {bash} export GLOBUS_LOCATION=<globus_install_dir>

Examples:
{csh} setenv GPT_LOCATION <gpt_install_dir>
{bash} export GPT_LOCATION=<gpt_install_dir>

4.3.2 INSTALLING GPT:

The Globus Toolkit uses the Grid Packaging Toolkit (GPT) packaging software developed at NCSA. GPT is a multi-platform packaging system used to deploy Grid middleware. This release of the Globus Toolkit should be built using GPT 3.0.1.
Under the distribution and enter the following commands.

- % gzip -dc gpt-3.0.1-src.tar.gz | tar xf -
- % cd gpt-3.0.1
- % ./build_gpt

4.3.2.1 Binary Bundle Installation:

- This section will show you how to install a binary bundle. We will install the toolkit using the linux binary bundle for the Intel i686 platform.
- To install the binary bundle, run the following command:
  % $GPT_LOCATION/sbin/gpt-install \globus-all-2.4.0-i686-pc-linux-gnu-bin.tar.gz
- Before you can finish your install, you need to source the following file. To do so, first set your GLOBUS_LOCATION. Then, depending on your shell, run:
- **{csh} source $GLOBUS_LOCATION/etc/globus-user-env.csh**
- **{sh} . $GLOBUS_LOCATION/etc/globus-user-env.sh**
- Once the installation of the binary bundle is complete and you have sourced the above file, run the following commands to complete your installation:
- **% $GPT_LOCATION/sbin/gpt-postinstall**

4.3.3 Configuring and Installation: The first thing we need to do is complete the setup of GSI, the security software that Globus uses. To complete the setup of the GSI software, you need to run the following command as root to configure your /etc/grid-security directory:

- **% $GLOBUS_LOCATION/setup/globus/setup-gsi**

4.3.4 Verification:

The following procedure explains how to carry out testing of some of the basic functionality of the Globus Toolkit.

Step 1: Verifying your installation
To verify that your installation is coherent, i.e. that all package dependencies have been satisfied, run the command:
% $GPT_LOCATION/sbin/gpt-verify

Step 2: Obtaining certificates
Security is at the heart of Globus, and as such, you will not be able to test our Globus configuration until you have obtained a certificate for yourself.

{csh} source $GLOBUS_LOCATION/etc/globus-user-env.csh
{sh} . $GLOBUS_LOCATION/etc/globus-user-env.sh

Step 3: Testing your installation
When we have a user certificate, we can use the following tests to verify a working installation.

- Testing GRAM
- Testing MDS
- Testing GridFTP

4.4 USAGE OF GLOBUS:

- The open source Globus® Toolkit is a fundamental enabling technology for the "Grid," letting people share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy.
- The toolkit includes software services and libraries for resource monitoring, discovery, and management, plus security and file management. In addition to being a central part of science and engineering projects that total nearly a half-billion dollars internationally, the Globus Toolkit is a substrate on which leading IT companies are building significant commercial Grid products.
- The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop applications.
- Every organization has unique modes of operation, and collaboration between multiple organizations is hindered by incompatibility of resources such as data archives, computers, and networks.
- The Globus Toolkit has grown through an open-source strategy similar to the Linux operating system's, and distinct from proprietary attempts at resource-sharing software. This encourages broader, more rapid adoption and leads to greater technical innovation, as the open-source community provides continual enhancements to the product.

4.4.1 Purpose and Standard of Globus toolkit:

- The Globus Toolkit is an open source toolkit, freely available in source code form for use by anyone, including both commercial and non-commercial purposes.
- The Globus Toolkit has been designed and implemented to capitalize on and encourage use of existing standards from communities such as IETF, W3C, OASIS and GGF.
- The tools in the Globus Toolkit use these standards rather than creating new mechanisms that do the same things in different ways. Some examples of standard mechanisms used in the Globus Toolkit are listed below.
- SSL/TLS v1 (from OpenSSL) (IETF)
- LDAP v3 (from OpenLDAP) (IETF)

- X.509 Proxy Certificates (IETF)
- SOAP (W3C)
- HTTP (W3C)
- GridFTP v1.0 (GGF)
- OGSI v1.0 (GGF)

4.4.2 The Use of Globus Toolkit Components:

- Two Very Important Software Development Kits:

These SDKs (of particular interest to software developers) are critical for developing new Grid software services within the OGSA framework.

- Web Services Core (WS-Core) Implementation Used to develop and run OGSA-compliant Grid Services (available in Java and C/C++)
- Grid Security Infrastructure (GSI) Implementation Used to secure communication (e.g., between services and clients)

- A Variety of Basic Grid Services:

These services are popular among current Grid application and system builders. They provide uniform interfaces to the most typical types of system elements and they currently include both OGSA and non-OGSA implementations.

- Computing / Processing Power (GRAM)
- Data Management (GridFTP, DAI, RLS)
- Monitoring/Discovery (MDS)
- Authorization/Security (CAS)
- In development: Telecontrol (NTCP/GTCP), Metadata (MCS), Virtual Data (Chimera, Pegasus)

- Developer APIs:

All of the services and tools above include C/C++ libraries and Java classes for building Grid-aware applications and tools.

- Tools and Examples:

All of the services and tools above include tools and examples based on the developer APIs and associated services.

4.4.3 How to Use the Globus toolkit:

Very few of the components in the Toolkit include user interface elements that would be immediately useful to scientists, engineers, or other application users. One most certainly cannot get useful results

The Globus Toolkit can be extremely useful to application developers and system integrators. In order to make good use of the Toolkit, one must have a specific application or system in mind, preferably one that involves shared use of physical or logical resources from several administrative domains.

The following sections provide pointers to some of the Grid community's best tools and services including many that are not currently included the Globus Toolkit.

Security: describes a number of useful software tools for meeting the security requirements in Grid systems.

Monitoring and Discovery: describes software components that can provide monitoring and discovery features in Grid systems.

Computation: describes software tools that can be used to manage computational tasks in Grid applications.

Data: describes software tools that can be used to manage data and datasets in data-intensive applications.

Collaboration: describes software for facilitating and encouraging collaboration in distributed projects.

Packaging and Distribution: describes tools for helping to create integrated software distributions for

use in Grid projects.

4.4.4 Additional Globus toolkit Services:

- Globus toolkit 4 is used in WS protocols for service interactions.
- Globus toolkit 4 services work according to WSRF behaviour paradigms.
- The Toolkit's documentation, installation, configuration, and training resources are constantly improving through the efforts of the entire Grid community.

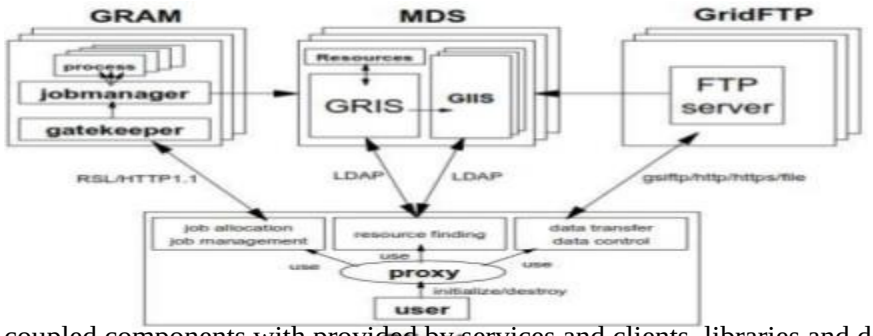
4.5 MAIN COMPONENTS AND PROGRAMMING MODEL:

4.5.1 Main Components of Globus toolkit:

The Globus Toolkit is a collection of reusable components to help application builders harness the grid.

There are six major Globus toolkit components:

1. **Data transport and access:** It is a common protocol for secure, efficient, flexible, extensible data movement.
2. **Replica management architecture** : It is used for managing multiple copies of files and collection of files.
3. **GRAM:** The Globus Toolkit includes a set of service components collectively referred to as the Globus Resource Allocation Manager (GRAM). GRAM simplifies the use of remote systems by providing a single standard interface for requesting and using remote system resources for the execution of "jobs".
4. **MDS**: The Monitoring and Discovery System (MDS) is the information services component of the Globus Toolkit and provides information about the available resources on the Grid and their status.
5. **GridFTP:** GridFTP-Lite is a light-weight GridFTP that uses a SSH based authentication mechanism instead of Grid Security Infrastructure (GSI). Even though GSI is quite powerful and provides single sign-on capabilities, it is quite complex to setup and maintain. GridFTP-Lite can be used as a standalone tool.
6. Proxy: That acts as an intermediary for requests from GRAM, MDS and GridFTP seeking resources from other **servers**.



A set of loosely coupled components with provided by services and clients, libraries and development tools.

4.5.2 The Programming Model of the Globus Toolkit:

- Java Programming Model focused on providing support for writing Java Grid services in GT3 Core.
- The main goal has been to make it as easy as possible to write our own services and deploy them into the container framework without having to worry about providing mandated OGSI functionality.