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.

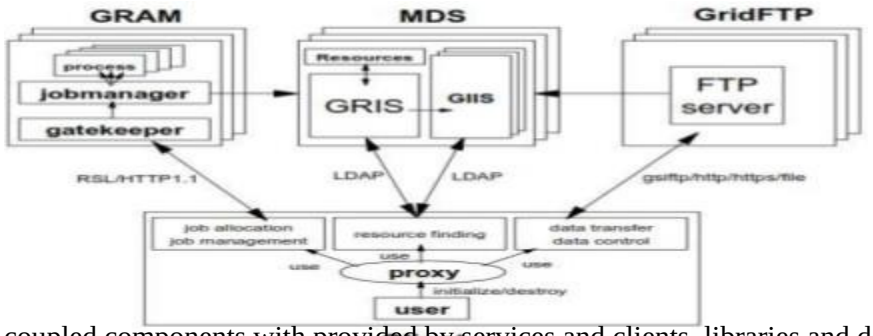
4.5 MAIN COMPONENTS AND PROGRAMMING MODEL:

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:

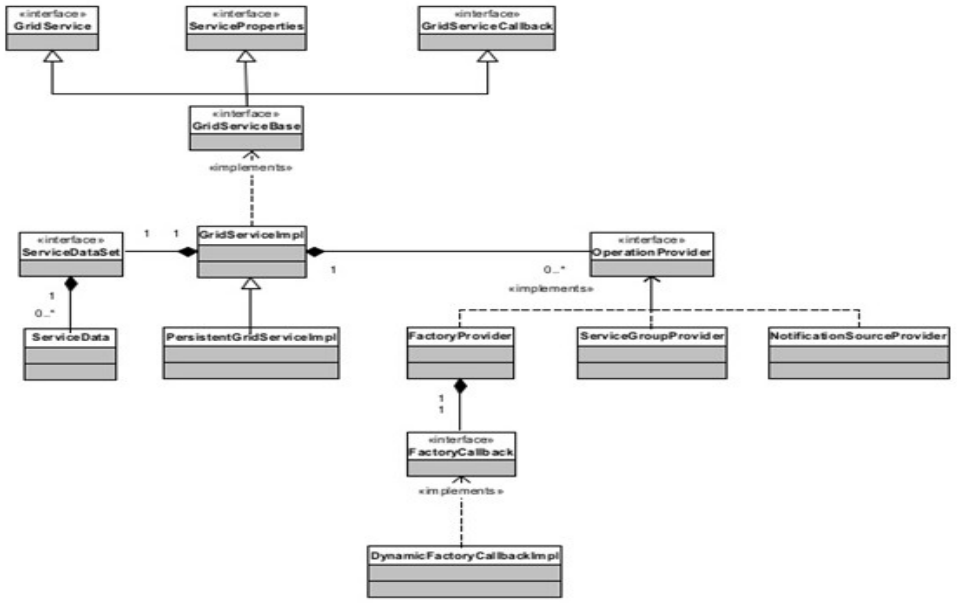
1. **Data transport and access:** It is a common protocol for secure, efficient, flexible, extensible data movement.
2. **Replica management architecture** : It is used for managing multiple copies of files and collection of files.
3. **GRAM:** The Globus Toolkit includes a set of service components 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**.



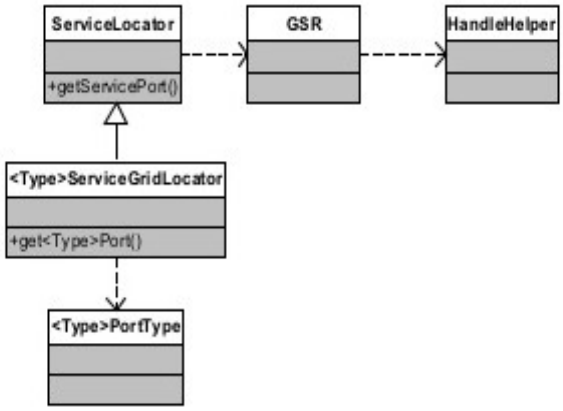
A set of loosely coupled components with provided by services and clients, libraries and development tools.

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.



- A Grid Service Base object is the base of all Grid services and implements the standard OGSI Grid Service Port Type.
- It also provides APIs to modify instance specific properties (either set by configuration or at run time), as well as APIs for querying and modifying service data.
- The base functionality can be seen as the functionality known at development time. At deployment and run time, additional functionality can be added in using Operation Providers, described in the next section.
- In GT3 Core we provide two implementations of the Grid Service Base interface. One to be used for transient (created by an OGSI Factory) Grid services called Grid Service Impl, and another one called Persistent Grid Service Impl to be used for persistent (created through configuration entries and always available in a container) Grid services.
    1. **Operation Providers:** A service can be created by simply extending from Grid Service Impl, or Persistent GridService Impl, but it is not recommended because of its limited flexibility. Recall that the base implementation 'locks' the service at development time into supporting a certain interface and it is then hard to reuse or customize the features of this service at deployment- and run time.
    2. **The GridService Callback** interface defines a number of lifecycle management callbacks that you can optionally implement to manage the state of your service
    3. **Factory Callback** can be implemented to provide custom factories for your services. It can, for instance, be used to create services in remote hosting environments.
    4. Most implementations are, however, likely to use the dynamic factory callback implementation we provide, which allows you to, through configuration, specify the implementation class that the factory should create.



- **Fig(a) Grid Service Client Programming Model**

- A Grid service client can be written directly on top of the JAX-RPC client APIs. However, for convenience and ease of use, we provide a number of utility classes simplifying GSH to GSR resolution, and GSR introspection.
- Further, we provide a custom stub generator extending the JAX-RPC stubs to integrate these utilities seamlessly into the client programming model.
- A client will typically get a handle from a registry, or through some out of band mechanism, to a well-known service instance, such as a factory.
- The handle is passed into a Service Locator that constructs a proxy, or stub, responsible for making the call using the network binding format defined in the WSDL for the service.
- The proxy is exposed using a standard JAX-RPC generated Port Type interface (sometimes referred to as Service Endpoint Interface). Note that this interface is identical to the one used on the server side to implement the service.

4.6 INTRODUCTION TO HADOOP FRAMEWORK

Part-A	**1.Name The Different Modules In Hadoop Framework(AU/April/May2017)** **2.What is the purpose of Heartbeat in hadoop(AU/Nov/Dec2017)** **3.What are the advantages of using Hadoop(AU/Nov/Dec/2016)**
Part-B	**1.Give a detailed note on hadoop Framework (AU/Nov/Dec 2016)**

Hadoop is the Apache Software Foundation top-level project that holds the various Hadoop subprojects that graduated from the Apache Incubator. The Hadoop project provides and supports the development of open source software that supplies a framework for the development of highly scalable distributed computing applications. The Hadoop framework handles the processing details, leaving developers free to focus on application logic.

The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing, including:

- **Hadoop** Core, our flagship sub-project, provides a distributed filesystem (HDFS) and support for the MapReduce distributed computing metaphor.
- **HBase** builds on Hadoop Core to provide a scalable, distributed database.