# INTEGRATION OF P2P RESOURCE DISCOVERY IN GRID (IPG)

By


**Rabail Javed**

(2003-NUST-BIT-42)


A project report submitted in partial fulfillment of
the requirement for the degree of
Bachelors in Information Technology

In


NUST Institute of Information Technology
National University of Sciences and Technology
Rawalpindi, Pakistan


(2007)

# CERTIFICATE

It is certified that the contents and form of thesis entitled **"Integration of P2P Resource Discovery in Grid (IPG)"** submitted by Ms. Rabail Javed have been found satisfactory for the requirement of the degree.

**Project Advisor:** ___________________

**Lecturer (Mr. Aatif Kamal)**

**Co-Advisor:** _________________________________

**Associate Professor (Dr. Hafiz Farooq)**

**Member:** _________________________________

**Lecturer (Mr. Abdul Ghafoor)**

**Member:** _________________________________

**Lecturer (Mr. Nauman Qureshi**)

# DEDICATION

I dedicate my unending efforts and accomplishment of this project to my supportive parents , my best friend, my loving brother and my advisor , who kept me motivated through out the course of this project and inspired me to make the impossible possible.

# ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my advisor Mr. Aatif kamal for his untiring efforts, assistance, inspiration, guidance, encouragement and support like my father at every stage of this project. I am highly thankful to my co-advisor Dr. Hafiz Farooq for his valuable suggestions and guidance throughout my thesis work.

I am especially grateful to my committee members Mr. Abdul Ghafoor and Mr. Nauman Qureshi for their motivation, support and help.

My special thanks to Mr. Nasir Mehmood , Mr. Kashif Sharif and Mr. Jamil Raza and Mr.Mujtiba Haider for their unending support, motivation and encouragement.

I express my gratitude to all of my teachers for their contributions in the enhancement of my knowledge and polishing my personality, all my professional skills owes to my teacher.

I offer my deepest thanks to my all my friends, especially my best friend Neelma Nadeem, Ayesha Rafique , Abdul Latif, Ali Raza Zaidi, Sumana Sattar and hummad anwer, who encouraged me to work harder and harder and for their moral support throughout my course and project work.

I offer my deepest love and indebtedness to my loving parents for their encouragement and prayers to Almighty Allah for my success.

I am thankful to many friends who encouraged me to work harder and harder and continued their moral support throughout my course and project work.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABSTRACT

Grid and P2P technologies are targeting different communities, they have different strengths and weaknesses, Grid follows a centralized approach for resource discovery and concentrates a lot on infrastructure, on the other hand P2P follows a decentralized approach for resource discovery and concentrates on failure handling rather than infrastructure.

Existing Grid resource discovery solutions are not scalable , one of the solutions proposed , is super peer model, but it only works for tightly coupled nodes, number of other communities are working on this issue for finding the synergies between Grid and P2P.

Above mentioned limitations of both the technologies depicts that a solution lies in the integration of P2P in Grid for efficient resource discovery. A service need to be developed that can help the interaction between P2P and Grid, and makes it possible to use resource discovery mechanism of P2P in Grid without making any changes in both the technologies. If both technologies are integrated, large scalable and resource efficient networks can be developed.

Integration is the main theme behind the project, which provides the motivation for the development of a service which can make the communication between Grid and P2P possible. IPG is an effort towards the integration of Grid and P2P networks, which uses a mapping technique for the integration purpose, it maps the Grid parameters into P2P parameters so that both can understand each other effectively and Grid can use P2P resource discovery mechanism efficiently.

IPG considers two scenarios; one is the Grid Wrapper Service, which interacts with P2P client through pipe establishment and with Grid Service through SOAP messages. It gets P2P end point messages from the P2P Client , map them into SOAP messages and send them to Grid Service and vice versa. Second is P2P Wrapper Service which interacts with Grid client through SOAP messages and with

P2P Service through pipes, It gets SOAP messages from Grid Client maps them to P2P end point messages.

Security issues are yet to be explored, A service prototype of the solution has been developed, but a complete system will be proposed in the future versions which will deal with the complete mapping of all the messages.

**Future of the IT lies in the integration of technologies rather than invention of new technologies. IPG is a step towards the integration of two competing technologies that is Grid and peer to peer.**

*Chapter 1*

# INTRODUCTION

Grid and P2P are targeting different communities, they have different strengths and weaknesses , Grid following a centralized approach for resource discovery and concentrates a lot on infrastructure, on the other hand P2P following a decentralized approach for resource discovery and concentrates on failure handling rather than infrastructure. This project will proves that if both technologies are integrated, large scalable and resource efficient networks can be developed.

In this project design and implementation of a service has been completed, This service is an effort towards the integration of Grid and P2P networks, which uses a mapping technique for the integration purpose, it maps the Grid parameters into P2P parameters so that both can understand each other effectively and Grid can use P2P resource discovery mechanism effectively and efficiently.

## 1.1 BACKGROUND

Web service is a model for the services that can be discovered, invoked and described using XML technologies such as SOAP, WSDL and UDDI. They use the brokered model such as Napster, Jini or CORBA but differ in terms of open standards.[1]

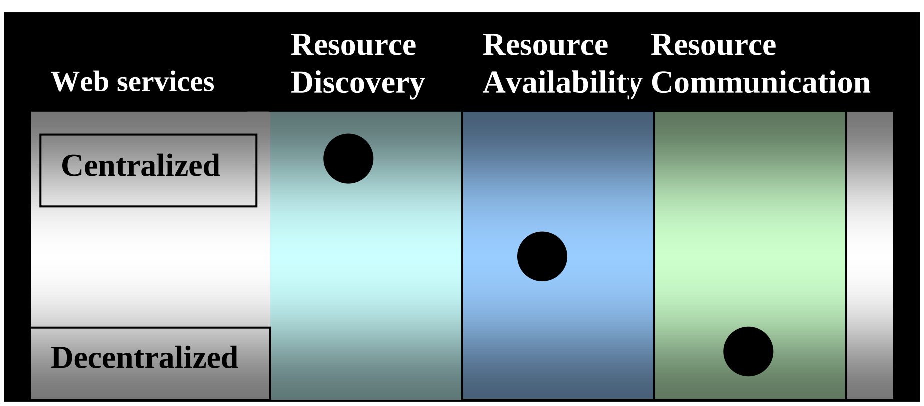



Figure 1.1 Taxonomy for web services[1]

If we take a look at Grid community, it is also adopting web services and OGSA is a step towards it. OGSA needed a distributed middleware on which they could base the architecture. If OGSA defined the interface, there has to be a standard way to invoke that method, if want our architecture standardized. Any distributed middleware could become the base of the architecture, but web services were the best choice, but they didn't meet one of the requirements of OGSA architecture that is the underlying middleware should be stateful.[2]

OGSA requires the notion of state to a Web services so that they can be monitored and referenced and that if the state is change, notifications could be sent to the applications. So WSRF entered into the scene, which specifies how we can make stateful web services which OGSA needs.

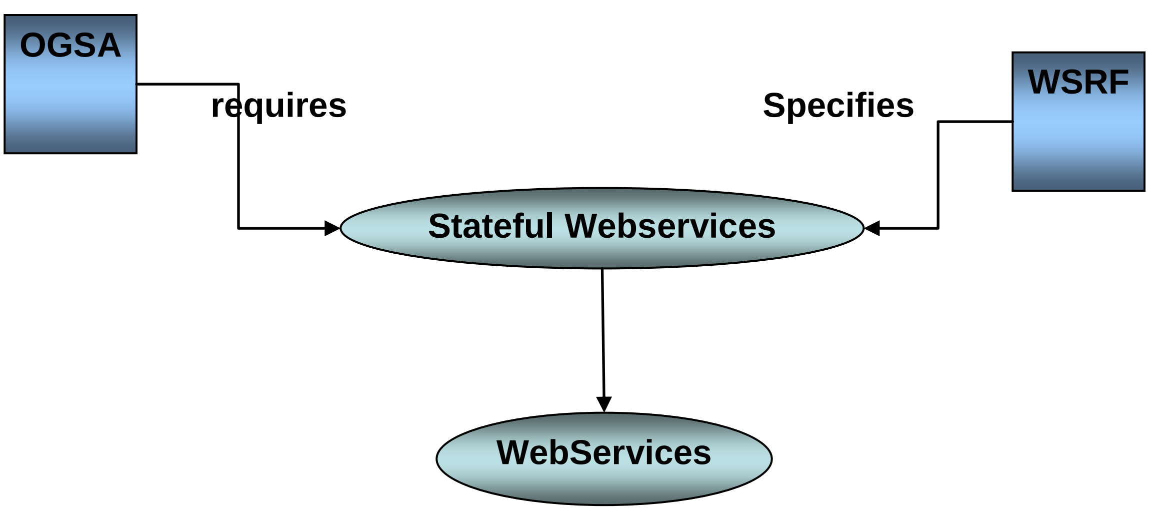



Figure 1.2: Relationship between OGSA, WSRF and Web Services [2]

In the existing Grid systems resource discovery is being handled using a centralized and hierarchical paradigm. As we talk about the degree of centralization , we donot mean to imply that there is only one server serving the information, rather we mean that there are a fixed number of servers providing the information which does not scale proportionately with the size of the network. These approaches require high level of trust and centralized control of resources, and also these approaches will become inadequate for scalable and future complex Grids, so more reliable and scalable solution for resource discovery is needed. Resource discovery in P2P on the other hand is totally decentralized.

## 1.2 PROBLEM STATEMENT

"Integration of P2P in Grid for **efficient resource discovery** and utilization in Grid environment.

"Need of a **Service**, which helps a Grid node to interact to P2P node."

## 1.3 PROJECT AIM

The aim of this project is to make the resource discovery in Grid more efficient using the P2P resource discovery mechanism. This will result in more scalable and fault tolerant Grids.

## 1.4 PROPOSED SOLUTION

One of the important research problems is the centralized and hierarchical resource discovery mechanism of Grids. All the current resource discovery approaches are designed to work with the current structure of VO and don't consider the scalability of future generation Grid networks.

As far as resource discovery techniques are concerned, number of techniques exists, some of them are based on super peer model, the focus of this project was to find out the resource discovery mechanism that is best applied to the Grid networks.

The resource discovery approaches proposed in this project are:

1) Development of a Grid/ P2P service

2) Extension of the previous P2P algorithms

The Grid/ P2P service was selected as a proposed solution to the stated problem in order to achieve the project aim and objectives.

## 1.5 METHODOLOGY

In this project, we developed a service, which will make the Grid resource discovery more scalable and efficient using the P2P resource discovery mechanism. This service will not make any changes to the existing architectures of Grid or P2P rather both middleware are used by it and provide a user defined API for the resource discovery in Grid. There are two scenarios, in first case Grid Client gives the parameters to the P2P Wrapper Service which acts as a middle translator wrapper service and converts the parameters in the form that P2P service can understand. In second scenario Grid Wrapper Service make the communication possible between Grid service and P2P client.

## 1.6 MOTIVATION

The main motivation behind the project and the development of this service is that is will not make any changes to either architectures, and can be used just like any other web service on the runtime, The intention is to make the Grids more efficient, self organizing, fault tolerant and scalable.

This project will is a step towards the innovation in the Grid and P2P systems by integrating the both technologies in a new way.Grids addresses the infrastructure but not the failure and P2P addresses the failure but the infrastructure both technologies can be integrated in order to compensate strengths and weaknesses of each other.

## 1.7 PROJECT DELIVERABLES

Following were categorized as project deliverables:

- Test application in Globus
- Test Application in JXTA

- Grid Wrapper Service
- P2P Wrapper Service
- Architectural model of the service
- UML Documentation of the service
- Implementation of the Service
- Project report encompassing the complete work carried out under the project

## 1.8 TOOLS AND TECHNIQUES USED

This project uses two major technologies of the distributed compurting

- JXTA (P2P)
- Globus Toolkit- 4.0.4 (Grid)

### 1.8.1 JXTA

JXTA is an open network computing platform designed for peer-to-peer (P2P) computing by way of providing the basic building blocks and services required to enable anything anywhere application connectivity.[7,10]

#### 1.8.1.1 Why JXTA

JXTA is preferred because it provides

- Interoperability
- Platform Independence
- Ubiquity

The JXTA protocols standardize the manner in which peers**:**

- Discover each other
- Self-organize into peer groups
- Advertise and discover network services
- Communicate with each other
- Monitor each other

### 1.8.1.2 JXTA Architecture


Figure 1.4 JXTA Architecture

**JXTA Core layer**

It encapsulates minimal and essential primitives that are common to P2P networking

- discovery
- transport (including firewall handling)
- the creation of peers and peer groups

- security primitives

**JXTA Services layer**

It includes network services that may not be absolutely necessary for a P2P network, but are common in P2P environments:

- searching and indexing
- directory
- storage systems
- file sharing
- distributed file systems
- resource aggregation and renting
- protocol translation
- authentication

**JXTA Application layer**

It includes implementation of integrated applications such as:

- P2P instant messaging
- document and resource sharing
- content management and delivery

### 1.8.2 Globus Toolkit 4

#### 1.8.2.2 Architecture

The Globus Toolkit 4 components are divided into five categories: Security, Data Management, Execution Management, Information Services, and the Common Runtime. [2]

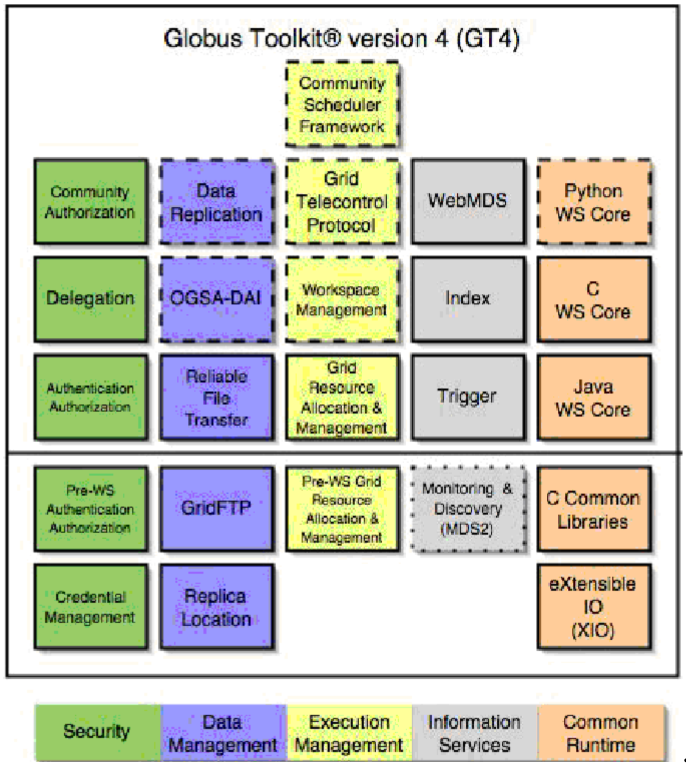



Figure 1.5 Globus toolkit 4 architecture [2]

### 1.8.1.2 GT4 Components

**Common Runtime**

The Common Runtime components consist of set of fundamental libraries and tools which are needed to build both WS and non-WS services.

**Security**

Security component, is based upon Grid Security Infrastructure, it actually ensures the secure communication.

**Data management**

This component helps to manage large sets of data in our virtual organization.

**Information services**

The Information Services or Monitoring and Discovery Services (MDS), used to discover and monitor resources in a virtual organization.

**Execution management**

Execution Management services actually deal with the initiation, monitoring, management, scheduling and coordination of executable programs.

## 1.9 TIME LINE

Various tasks carried out for completion of the research project, along with their estimated time frame, have been shown in Figure 1.4.

| ID | Task Name | Start | Finish | Duration |
|---|---|---|---|---|
| 1 | Background Study | 10/16/2006 | 2/28/2007 | 98d |
| 2 | Analysis | 1/8/2007 | 2/28/2007 | 38d |
| 3 | Design | 2/1/2007 | 2/28/2007 | 20d |
| 4 | Application Development | 3/1/2007 | 5/1/2007 | 44d |
| 5 | Testing | 5/1/2007 | 5/30/2007 | 22d |
| 6 | Research Paper | 2/28/2007 | 6/1/2007 | 68d |
| 7 | Documentation | 5/1/2007 | 6/15/2007 | 34d |

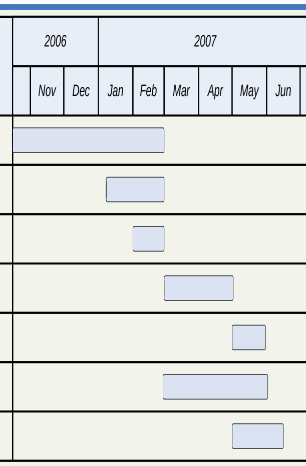



Figure 1.6: Project timeline

## 1.10 REPORT STRUCTURE

This document sheds light on all the aspects of the project, including the technique(s) being implemented and details of the developed tool. This document has been divided on the following pattern:

- Chapter 2: Literature Review. This chapter contains the existing techniques, encompassing their uses, advantages and disadvantages
- Chapter 3: Methodology. This chapter explains the approach used, development of service, system architecture and design of service
- Chapter 4: Results  This chapter includes the steps followed and the final results obtained
- Chapter 5: Conclusions. This chapter summarizes the whole project report
- Chapter 6: Recommendations: It discusses further improvements and recommendations in resource discovery service.

*Chapter 2*

# LITERATURE REVIEW

The review of literature for the project has been divided in the following modules:

1) On Death,Taxes and the Convergence of Peer to Peer and Grid Computing by Ian foster,Adriana Iamnitchi

2) A Super-Peer Model for Building Resource Discovery Services in Grids: Design and Simulation Analysis by Carlo Mastroianni, Domenico Talia, and Oreste Verta

3) A Java Based Architecture of P2P-Grid Middleware Hudzia B., Ellahi T.N., McDermott L. and Kechadi T.School of Computer Science and Informatics University College Dublin Dublin – Ireland benoit.hudzia, tariq.ellahi, liam.mcdermott, tahar.kechadig@ucd.ie

4) Web Services for Peer-to-Peer Resource Discovery on the Grid Domenico Talia and Paolo Trunfio DEIS, University of Calabria Via P. Bucci 41c, 87036 Rende, Italy ftalia,trunfiog@deis.unical.it

## 2.1 ON DEATH, TAXES AND CONVERGENCE OF P2P AND GRID COMPUTING

One of the major research issues is the resource sharing in large scale scalable distributed environments.P2P and Grid both intend to solve the same problem with the same approach that is resource sharing by the creation of overlay structures. But

**“Grid computing addresses infrastructure but not failure and P2P addresses failure but not infrastructure”. [3]**

### 2.1.1 Comparing Grid And P2P

#### 2.1.1.1 Target Communities and Incentives

Grid target professional and scientific communities e.g NASA and CERN but interest is growing from the commercial domain as well. P2P on the other hand is targeting mass users that scale in some times hundred of thousand of nodes .e.g Kazaa and SETI@home.

#### 2.1.1.2 Resources

Grid make use of diverse, powerful resources which are better connected with high degree of trust, centralized control and well defined policies that’s why Grid resources are always available. On the other hand P2P resources shows intermittent and variable behavior and they are less powerful (end user machines).

#### 2.1.1.3 Applications

Paper also discuss the applications areas of Grid and P2P, Some of the Grid applications are e.g the Host Page portal , which provides the remote access to supercomputer hardware and software, another solution is the numerical solution of the long-open “nug 30” quadratic optimization problem using hundreds of computers at many sites. NEESGrid system integrates earthquake engineering facilities into a national laboratory. In contrast, P2P provides more integrated kind of solution to specialize resource sharing problems, which is applicable currently on either computer cycles sharing or file sharing.

#### 2.1.1.4 Scale And Failure

Grid only accounts for modest no. of users and usually large activity is executed on Grid, and it doesn't address scalability and failure handling. P2P targets large no of users with large activity but less than Grid but it deals with scalability and fault tolerance issues.

#### 2.1.1.5 Services And Infrastructure

Although P2P deals with failure handling and scalability but it doesn't address infrastructure issue mainly due to the reason , there is no standardized protocol for P2P on the other hand Grid is more concerned about standard technologies and standard architecture i-e OGSA.

### 2.1.2 Future Directions

Future lies in the Integration of P2P and Grid computing because it will result in a ubiquitous solution which will have all these characteristics in it..

- Failure handling
- Scalable
- Persistent and multipurpose infrastructure
- Heterogeneity
- Robustness
- Performance
- Trust.
- Diverse discovery,
- Self configuration

## 2.2 A SUPER-PEER MODEL FOR BUILDING RESOURCE DISCOVERY SERVICES IN GRIDS: DESIGN AND SIMULATION

### 2.2.1 Integration

#### 2.2.1.1 Scalable Grids

Existing Grids will fail in the near future because they don't deal with the scalability issue, P2P decentralized resource discovery is one of the solutions to this problem.

#### 2.2.1.2 Super Model Approach

One solution has been proposed by the author is the use of super peers, these super peers serves single VO , and each VO is a cluster of tightly coupled nodes. Every VO has a super peer and all the super peers are connected to each other in P2P fashion., within the VO is a centralized search but between the VOs is the decentralized search in a P2P manner.[4,5]

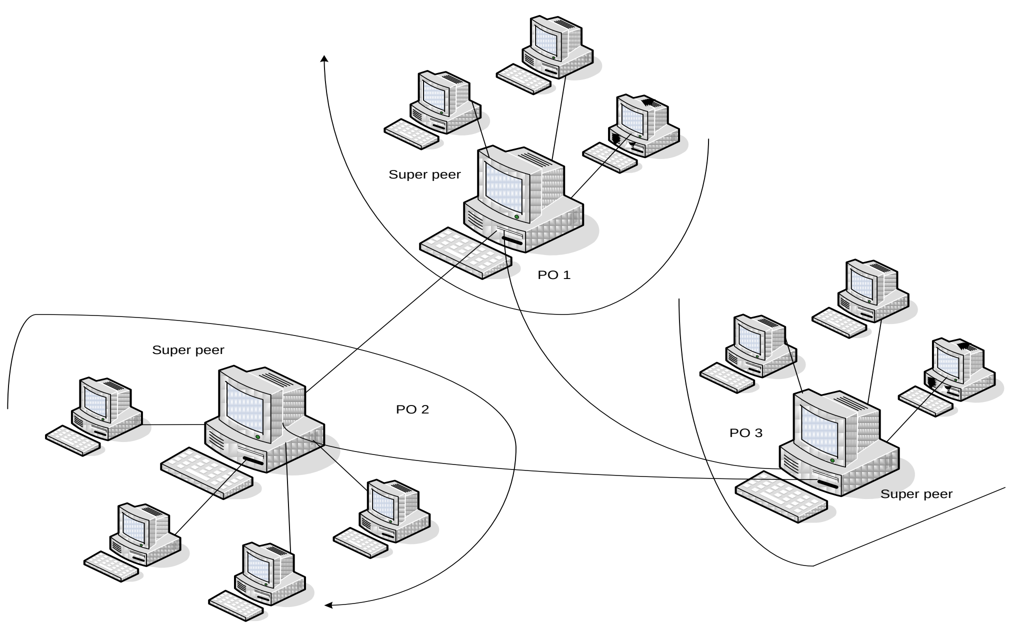



Figure 2.1: Super peer model [5]

### 2.2.2 Conclusion

Super peer model results in the integration of Grid an P2P network, it handles the whole model by membership management (membership of a node in a VO and resource discovery service

### 2.2.3 Problem With The Approach

After doing the critical analysis of this paper , super peer model approach works well with the current Grids but it would fail to work for the future Grids , because each VO is cluster of tightly coupled node and it wont work for the loosely coupled nodes.

## 2.3 A JAVA BASED ARCHITECTURE OF P2P-GRID MIDDLEWARE

Number of projects have been initiated with various visions but with the same goal of resource sharing, this paper present a new Grid system which address the data issues , called DGET (DATA Grid Environments and Tools), DGET make use of the advantages of both P2P and Grid systems. [12]

### 2.3.1 What Is The Research Problem The Paper Attempts To Address?

Resource Sharing is the main goal achieved by 2p an d Grid systems, discussing about the internet scale systems, there is a need of complex requirements of complex applications from the diverse disciplines, secondly the resources are also distributed and heterogeneous. Two systems are exploiting these views are Grid and P2P, their main objective pooling and coordination of large sets of distributed resources.

### 2.3.2 What Are The Claimed Contributions Of The Paper?

Major contribution of this paper is the development of DGET (Data Environments Tools), P2P Grid middleware which takes the advantage of both Grid and P2P, and prototype of this system is developed.

### 2.3.3 DGET Vs Grid Middleware

Globus, Legion and UNICORE are some of the middle wares of Grid, while comparing DGET and Grid middlewares, these points are notable

- Manual and static topology vs dynamic and self organizing topology
- Specialized Central servers vs decentralized resource discovery

And Specifically talking about DGET vs Globus , main differences are the security , resource accounting, Number of users and access control policies.

### 2.3.4 DGET Vs P2P Systems

P2P system lack sophisticated resource management and sophisticated security component. And they don't support migration, DGET on the other hand , has powerful resource management and powerful security mechanism (java security model) and supports Migration.

### 2.3.5 DGET Vs Hybrid Systems

Specifically talking about Our Grid and DGET, main differences are, DGET supports sophisticated resource discovery and migration of applications but Our Grid doesn't support these capabilities.

### 2.3.6 DGET Objectives

Main objectives behind the development of DGET is the uniform management of resources, ease of use and simplicity, fault tolerance and scalability , and these objectives are only possible , by the integration of Grid and P2P.

### 2.3.7 DGET Architecture

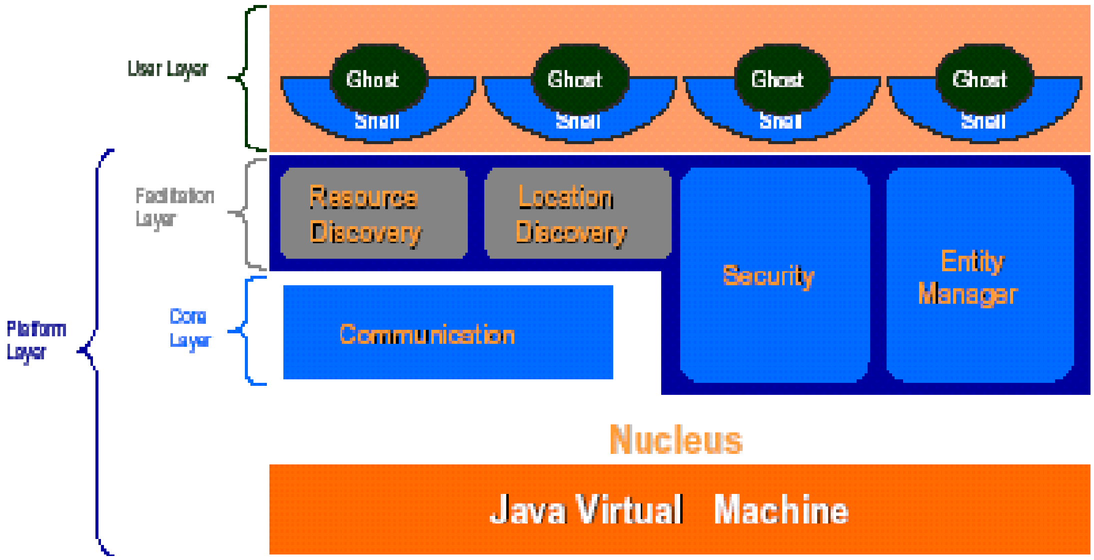



Figure 2.2: DGET Architecture[12]

### 2.3.8 Future Directions

- OGSA should support centralized and decentralized computing models.
- OGSA's requirements today, do not include all relevant requirements for P2P.

## 2.4 WEB SERVICES FOR PEER-TO-PEER RESOURCE DISCOVERY ON THE GRID

Today's Grids are based on centralized and hierarchical paradigms, but Grids should be decentralized in order to avoid the bottleneck issue and to address salability. On solution is use P2P protocols to develop decentralized Grid services, which will result in scalable Grids. This paper provides the architecture for using P2P resource discovery approach in Grid, and extends the model of GT3 information service. OGSA is exploited to define the P2P layer of specialized Grid services that enable resource discovery across different VOs in P2P manner. Author has also discussed a protocol named Gridnut for the communication between the Grid services at the P2P layer.[13]

OGSA defines Grid services as an extension of Web services and supports the integration of services and resources across distributed, heterogeneous, dynamic environments and community's .Web services and OGSA works in parallel to provide interoperability between loosely coupled services independent from the implementation, location or platform. WSRF had been proposed for even better and complete integration between web services and Grid services and introduced the concept of stateful web services for OGSA. .GT3 provides index services for the information about the resources; it is a Grid service which contains the information about the Ser of Grid services registered with it.

### 2.4.1 Proposed Architecture

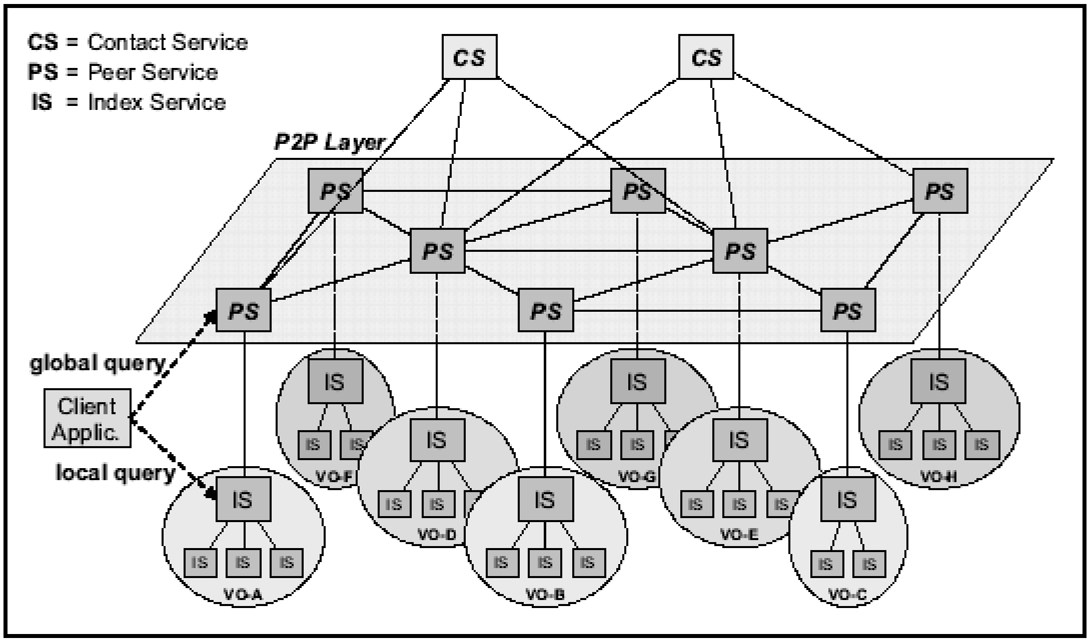



Figure 2.3: Peer service architecture [13]

A special P2P layer is defined on the top of the GT3 index service, two types of services have been developed , Peer service is used to perform the resource discovery and Contact services are used to help the peer to connect to each other.

There is a P2P service working for each VO. All of the peer services are connected to each other in P2P manner. A query message is sent to the peer service and it can only forward the query message to its neighbors, A query is processed by the peer service by the invocation of the index service of each VO . Response of the query message is sent back to the same path back.

### 2.4.2 Services Design

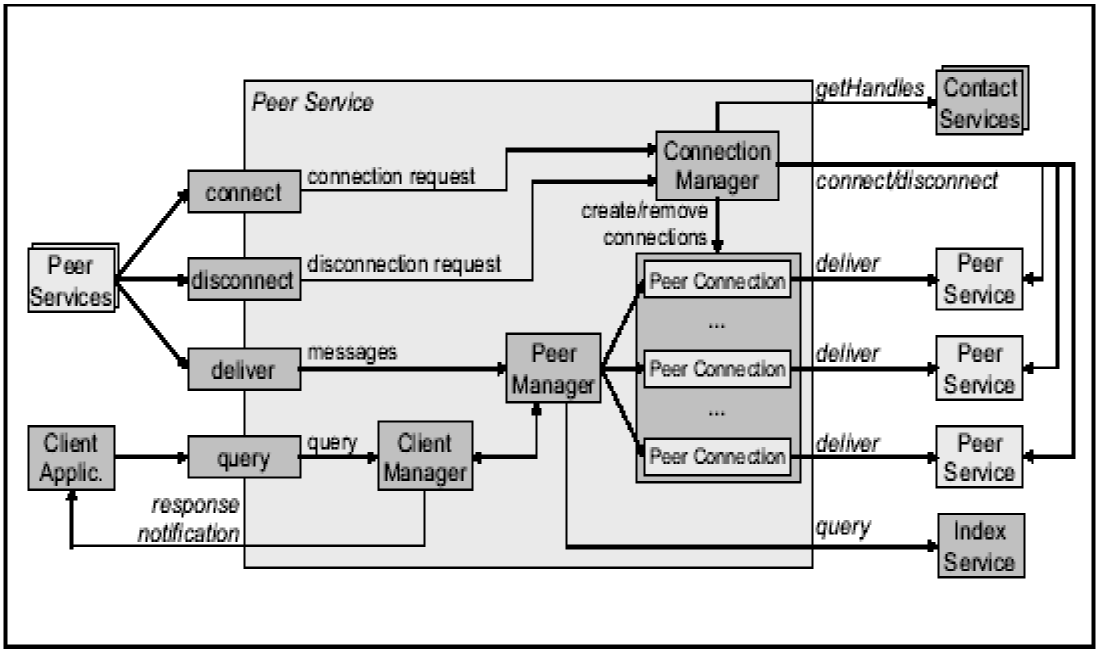



Figure 2.4: Peer Service design [13]

*Chapter 3*

# METHODOLOGY

In this project, we developed a service, which makes the Grid resource discovery more scalable and efficient using the P2P resource discovery mechanism. This service makes any changes to the existing architectures of Grid or P2P rather both middlewares are used by it and provide a user defined API for the resource discovery in Grid. Two services have been implemented to fulfill the goals of this project.

P2P wrapper service is developed to make the interaction possible between the Grid client and P2P service, Grid wrapper service is developed for the interaction between and P2P client and Grid service. These services take the parameters from either technologies, parse them, map them and covert them so that they are understandable.

## 3.1 System Architecture

Diagrammatic representation of the system architecture is shown in Figure 3.1 and 3.2. In Figure 3.1 Grid client communicates with the P2P service through P2P Wrapper service. P2P Wrapper service communicates with the Grid client in the form of SOAP messages, and with the P2P service through the establishment of pipes. It takes the parameters in the form of SOAP messages from a Grid client maps them into P2P end point message formats and vice versa.

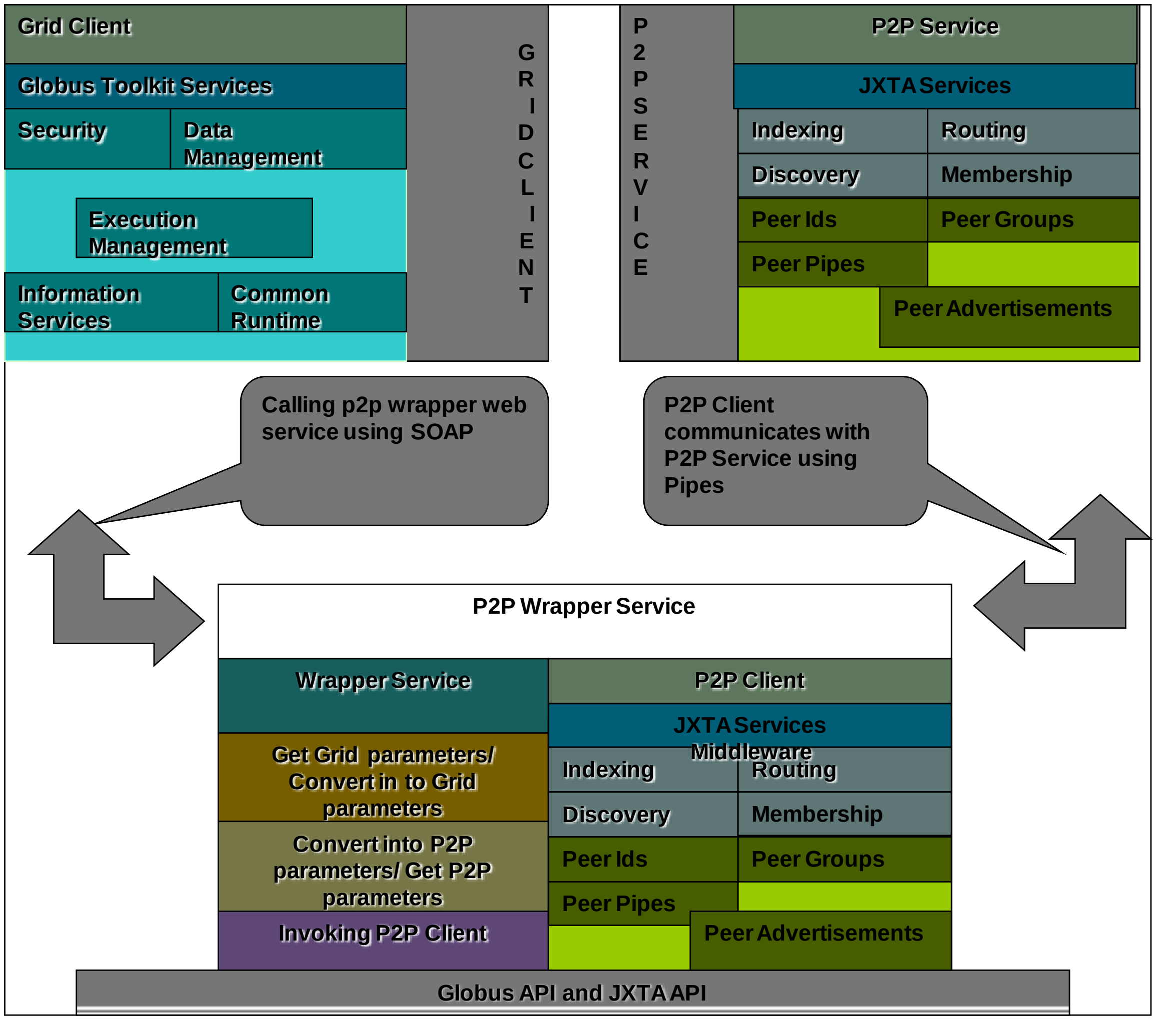



Figure 3.1 P2P Wrapper Service Architecture

In Figure 3.2 , Grid wrapper service proposed architecture has been discussed, This service makes the communication possible between Grid service and P2P client. P2P Client sends the query to get the services running under globus index , this query is received by the Grid Wrapper Service , it converts end point messages into SOAP messages and sends the query to the Globus index service to get all the services running under central index. Returned result will be mapped into P2P end point messages and forwarded to a P2P client.

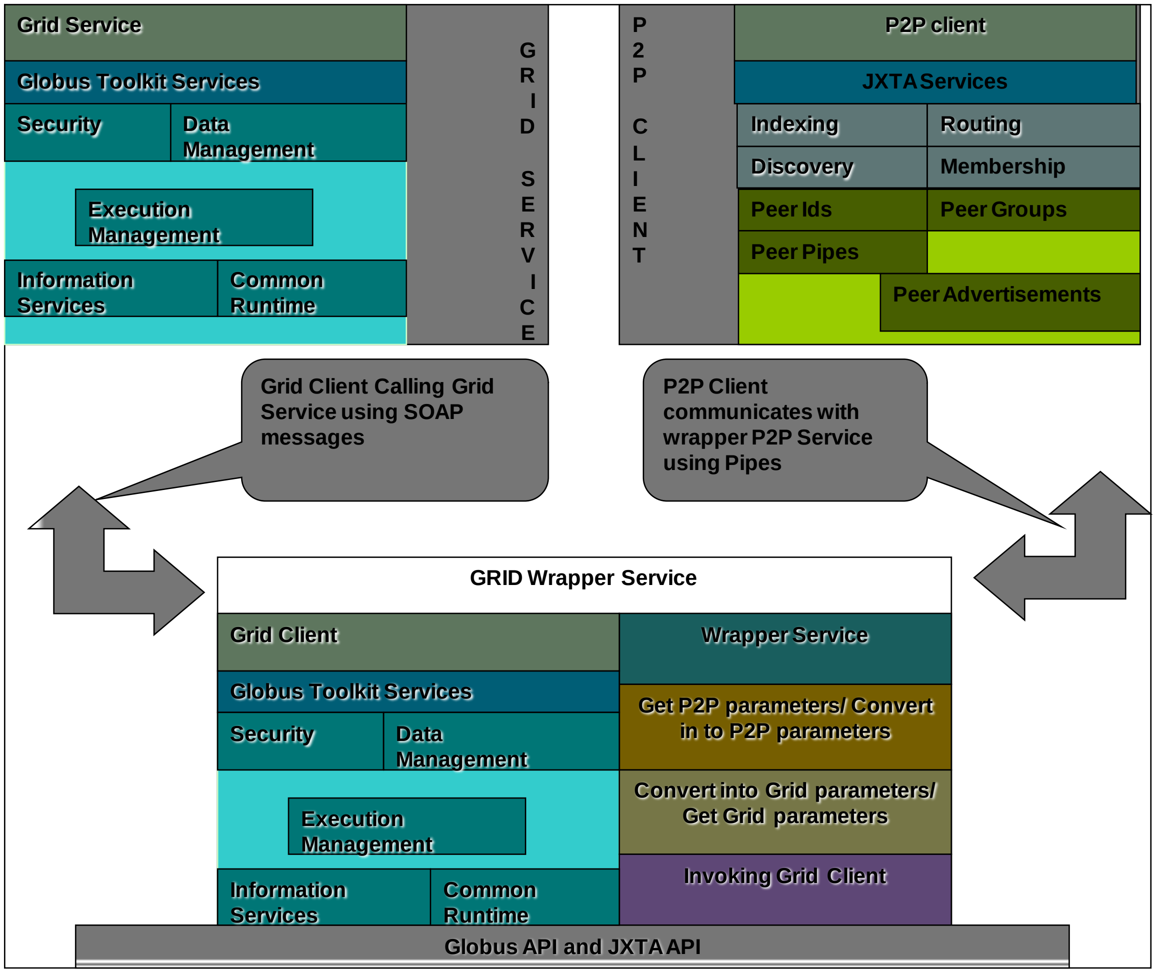



**Figure 3.2 Grid wrapper service Architecture**

## 3.2 System design

This section highlights the system design and the implementation details regarding P2P/Grid wrapper service. In order to explain the system design of P2P/Grid wrapper service use case diagrams, sequence diagrams and class diagrams have been employed.

### 3.2.1 Use Case Diagram

Use cases are description of the functionality of the system from the user's perspective. These diagrams were employed to show the functionality that the mapping service would provide and to show which services would communicate with the mapping service in some way to use that functionality. The use case diagrams of P2P wrapper service is shown in Figure 3.3a. and Grid Wrapper Service in Figure 3.3 b.

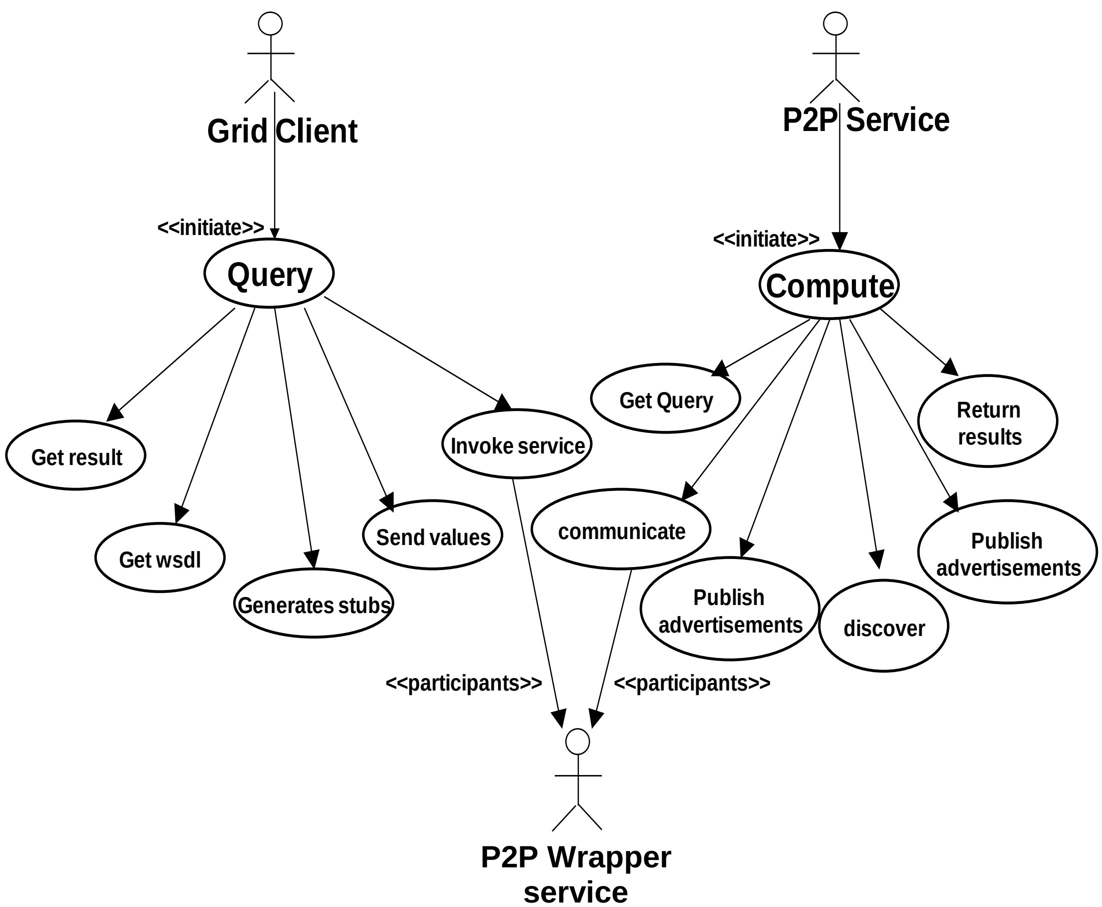



Figure 3.3 a Use Case Diagram

**Table 3.1 Query use case flow of events**

| | |
|---|---|
| **Use case Name** | Query |
| **Participating actors** | P2P Service,P2P wrapper service |
| **Flow of events** | 1. Gets the P2P wrapper service wsdl<br>2. Generate stubs<br>3. Input the parameters from the Grid client<br>4. Invoke P2P Wrapper Service<br>5. Get Results |
| **Entry Condition** | • Grid Client send parameters for the computation |
| **Exit Condition** | • Grid Client receives the response of the computation |
| **Includes** | Get Results, Generate WSDL, Invoke Service, Send values |
| **Extends** | ________ |

**Table 3.2 Compute use case flow of events**

| | |
|---|---|
| **Use case Name** | Compute |
| **Participating actors** | Grid Client, P2P wrapper service |
| **Flow of events** | 1. Publish service advertisement<br>2. Discover P2P wrapper services<br>3. Create input output pipes for the communication with the P2P wrapper service.<br>4. Gets the parameters<br>5. Computes<br>6. Return Results |
| **Entry Condition** | • P2P Wrapper service gives the parameters for the computation |
| **Exit Condition** | • P2P Service returns the results of the computation |
| **Includes** | Publish advertisements, Discover, get values, communicates, return results. |
| **Extends** | ________ |

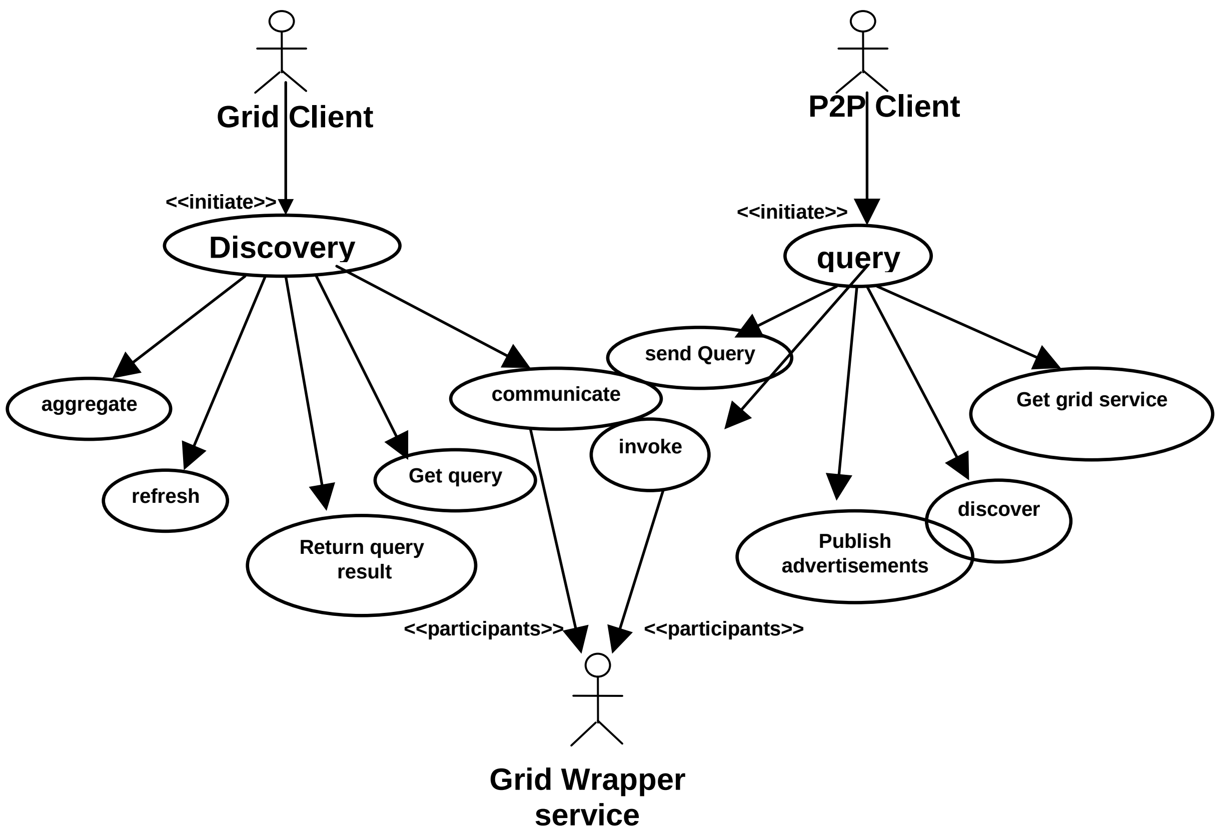



Figure 3.3 b Use Case Diagram

**Table 3.3 Discovery use case flow of events**

| | |
|---|---|
| **Use case Name** | Discovery |
| **Participating actors** | Grid Service, Grid Wrapper service |
| **Flow of events** | 1. Aggregates service information from all the services registered with globus index.<br>2. Gets query from the user<br>3. Communicates with the Grid wrapper service.<br>4. Returns the result of the query.<br>5. Refreshes the services information after some interval. |
| **Entry Condition** | • Grid Client gets the parameters from the client |
| **Exit Condition** | • Get the query resutls |
| **Includes** | Aggregate, get query, return query results, communicate, refresh |
| **Extends** | ________ |

**Table 3.4 Query use case flow of events**

| | |
|---|---|
| **Use case Name** | Query |
| **Participating actors** | Grid Service, Grid Wrapper service |
| **Flow of events** | 6. P2P Client publishes its advertisement.<br>7. Discovers Grid Wrapper Service.<br>8. Get the query for the Grid services from the P2P client.<br>9. P2P Client sends the query to get the Grid services .<br>10. Invokes Grid Wrapper service.<br>11. Returns the result of the query. |
| **Entry Condition** | • P2P Client gives the query |
| **Exit Condition** | • Get the results |
| **Includes** | Publishes Advertisement, get query, return query results, invokes, discover |
| **Extends** | ________ |

### 3.2.2 Class Diagrams

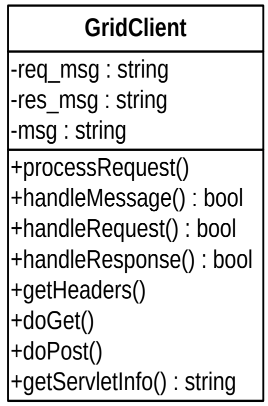



Figure 3.4 Grid Client Class Diagram

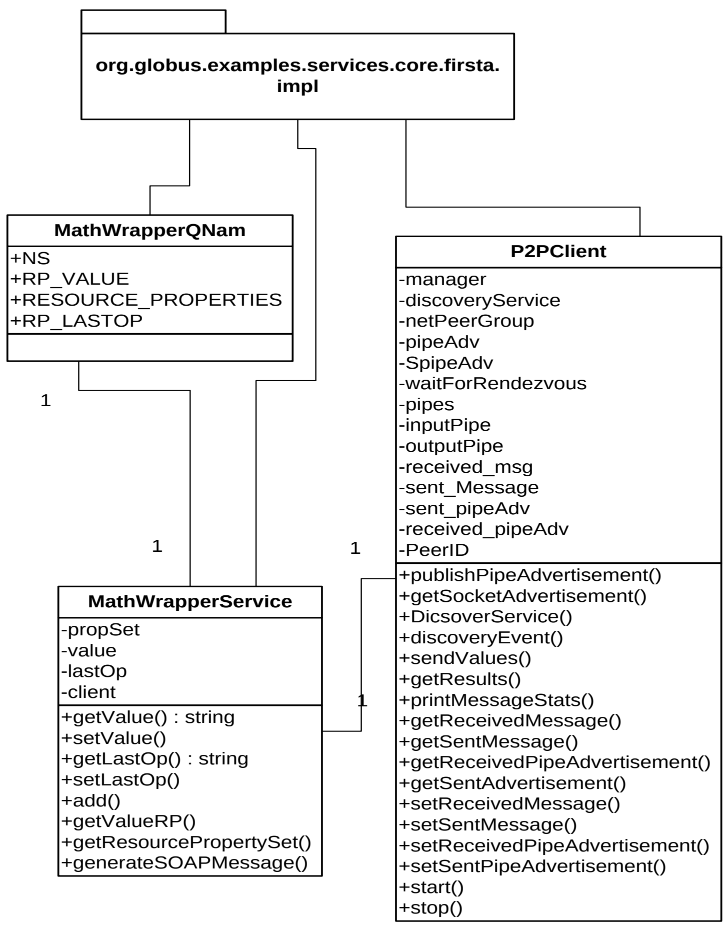



Figure 3.5 Grid Wrapper Service Class Diagram

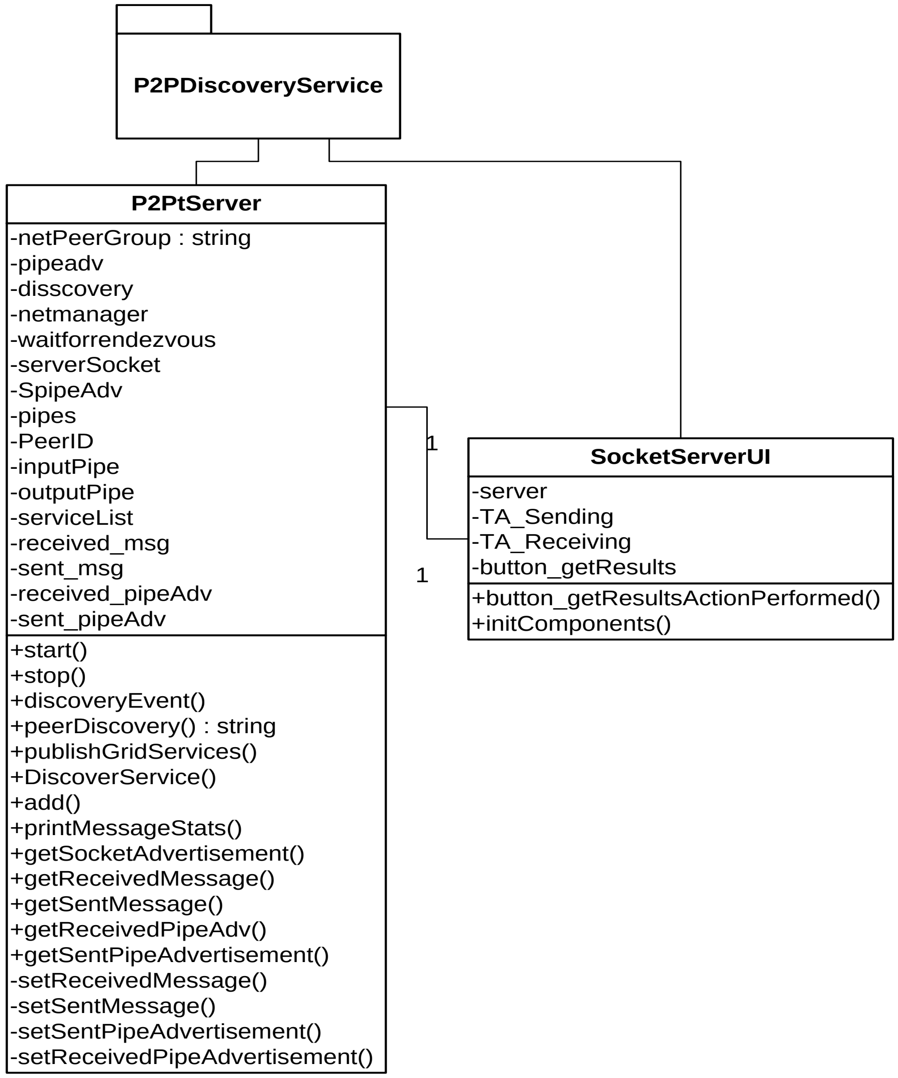



Figure 3.6 P2P Service Class Diagram

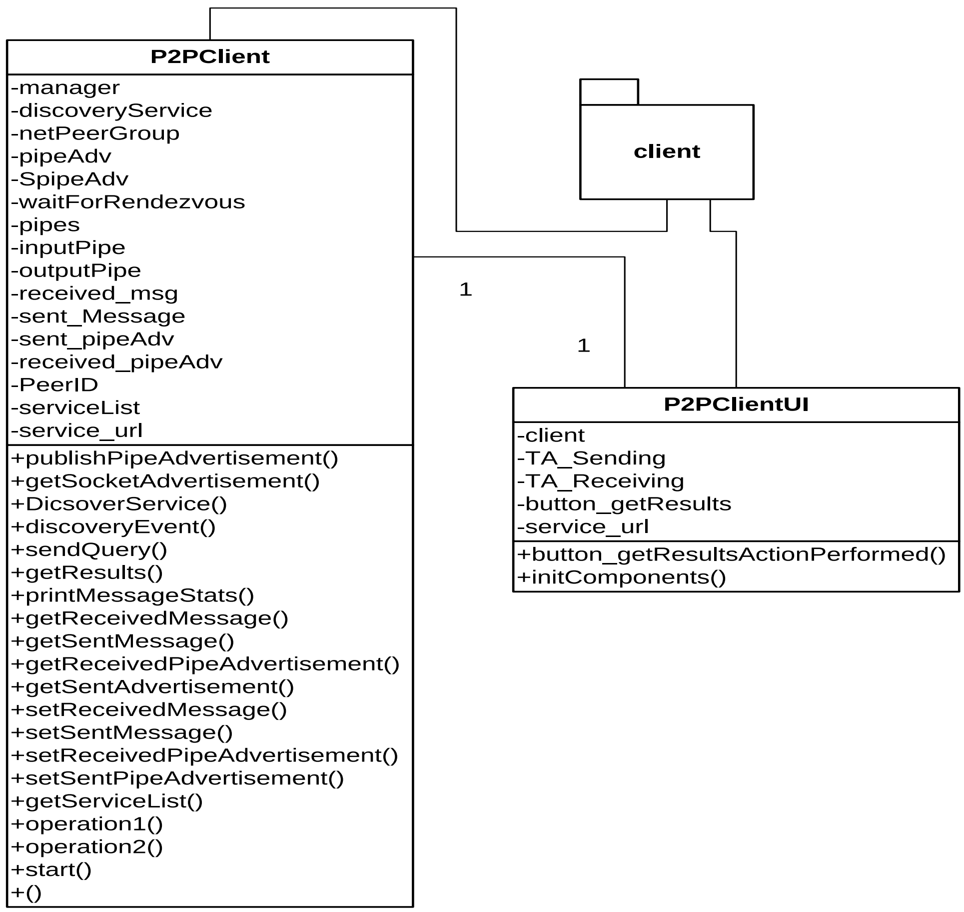



Figure 3.7 P2P Client Class Diagram

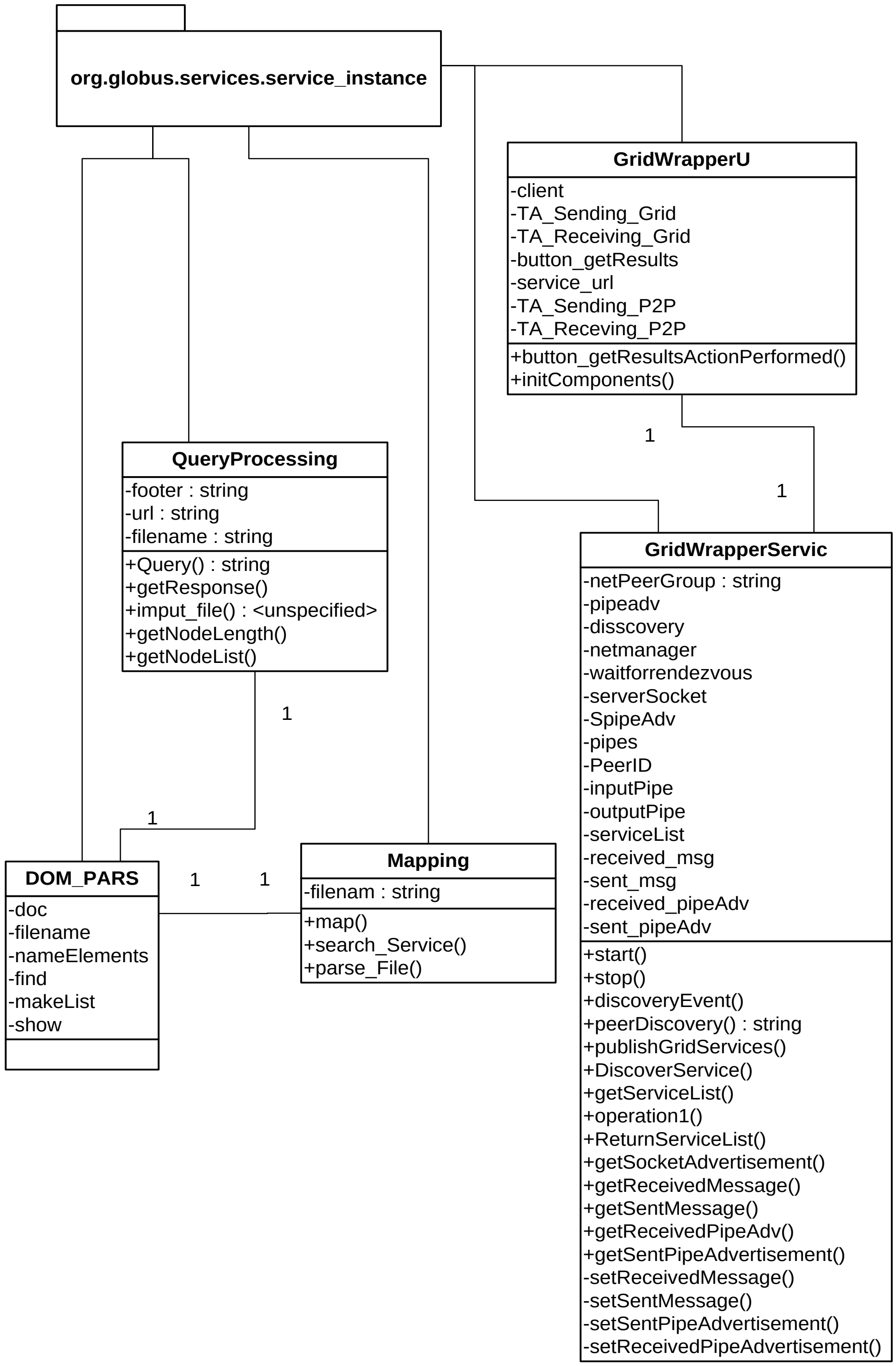



Figure 3.8 Grid Wrapper Service Class Diagram

*Chapter 4*

# RESULTS

This chapter highlights the findings and results of the project titled "Integration of P2P Resource Discovery in Grid". Two services are developed, One is Grid Wrapper Service and other is P2P Wrapper Service. For the result purposes, three machines are used. In the first case, machine 1 is running a P2P client , machine 2 a Grid Wrapper Service and machine 3 a Grid Service. In the second case machine 1 is running a Grid Client, machine 2 a P2P Wrapper Service and machine 3 a P2P Service.

## 4.1 GRID WRAPPER SERVICE AND RESULTS

Machine 1 is running a P2P Client as shown in the Figure 4.1; it publishes its own advertisement as shown in the sending text area and discovers the Grid Wrapper Service as shown in the receiving text area. After discovering each other, input and out put pipe is created, P2P client sends the message to Grid Wrapper service in the form of End Point Message Format, as shown in the sending text area.

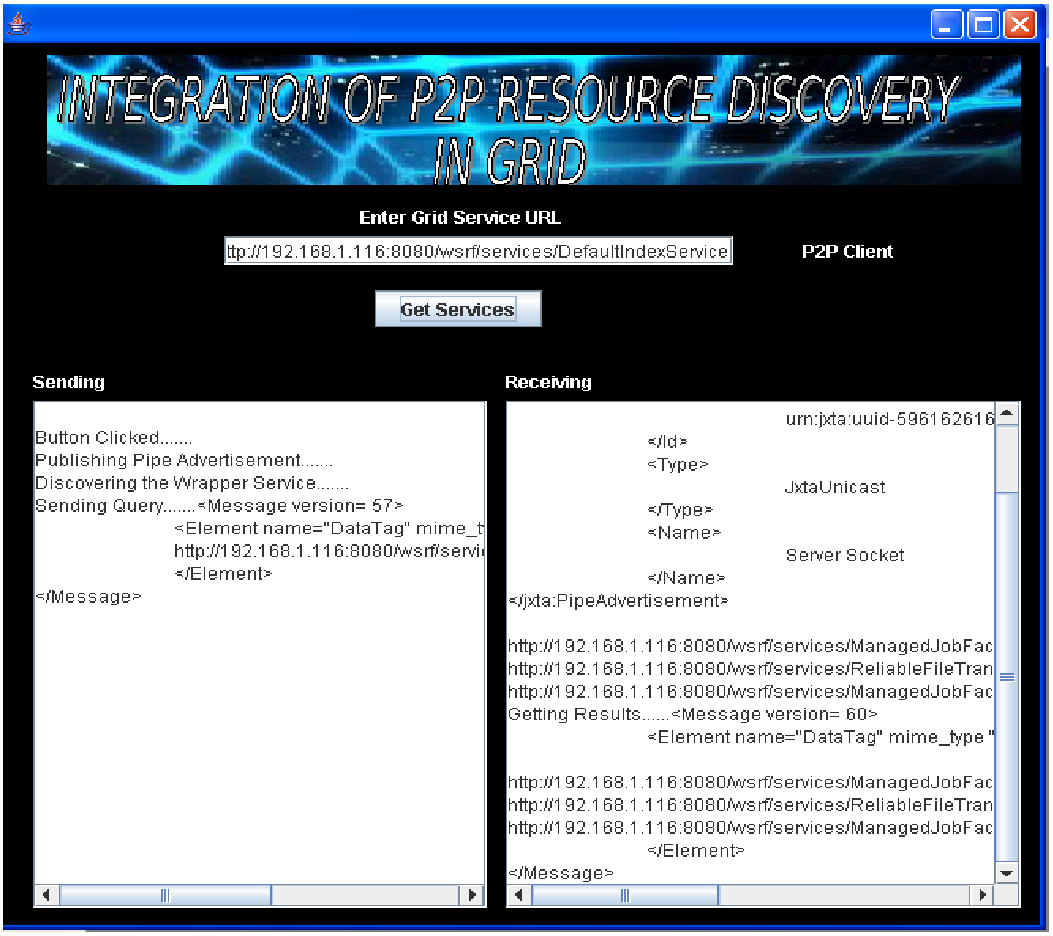



Figure 4.1 P2P Client Screen shot

Grid Wrapper Service discovers the P2P Client as shown in the “Receiving Message form P2P Client” text area. I also receives query in the form of End Point Message Format. Grid Wrapper Service parses the message, maps it into SOAP query and sends it to the Grid Service running on the machine 3 as shown in the “Sending the query to Grid Client” text area. Grid Wrapper Service gets the results in the form of SOAP messages, it parses them and maps them to End point P2P Message Format and sends the results to the P2P Client as shown in the “Sending Message to P2P Client” text area.

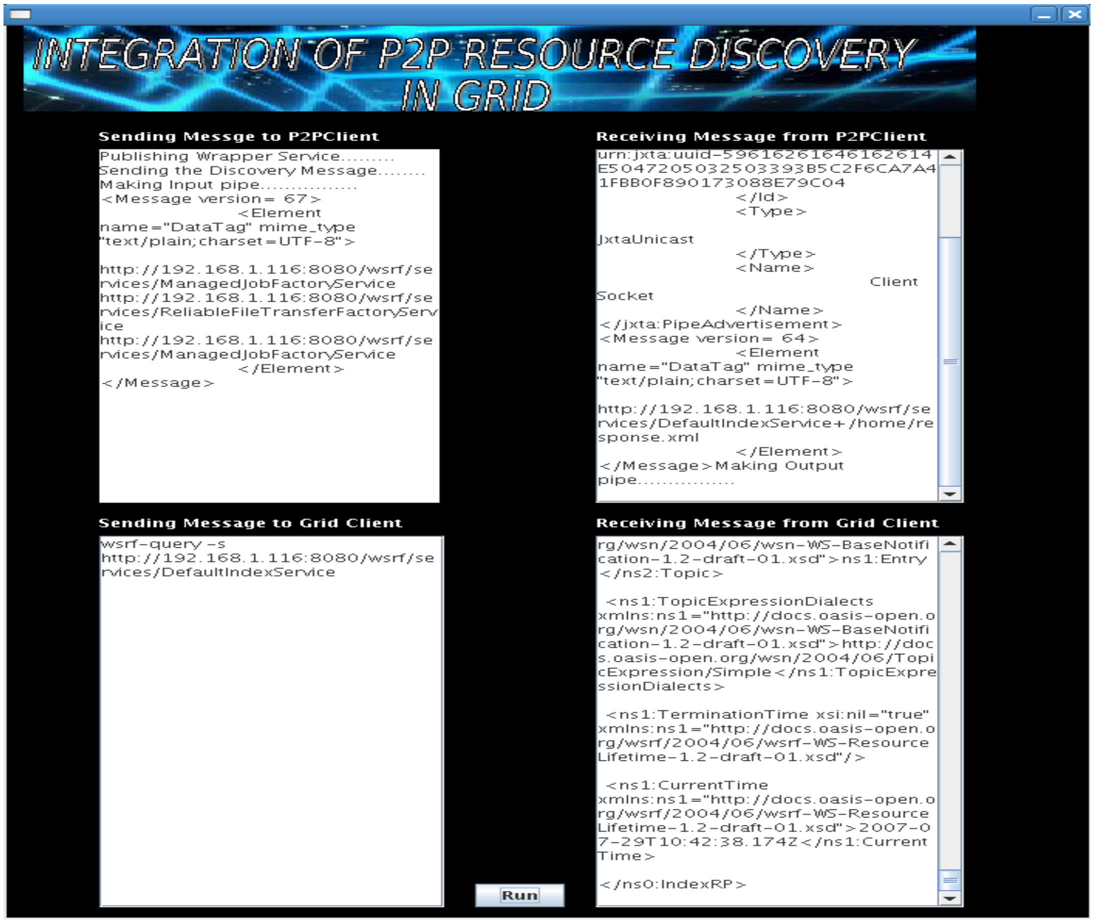



Figure 4.2 Grid Wrapper Service Screen Shot

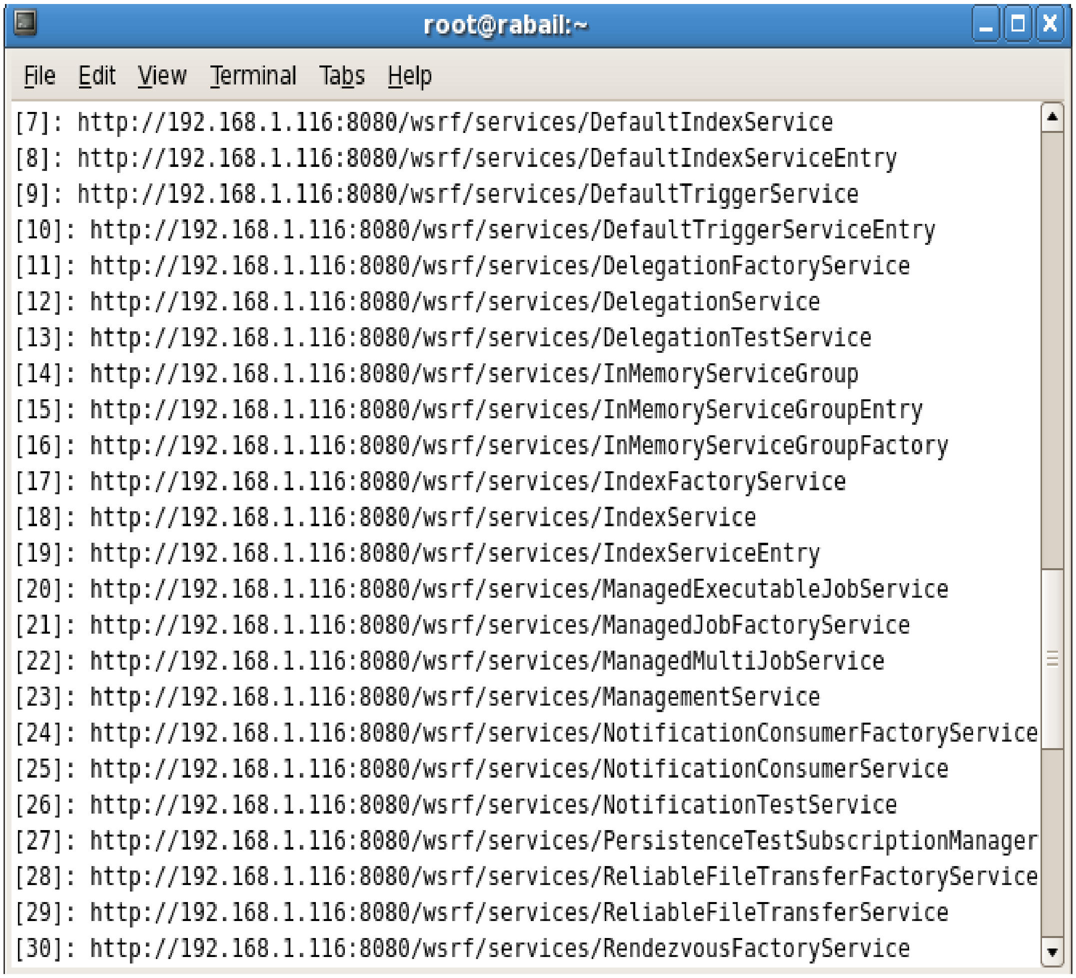



Figure 4.3 Globus Services running

## 4.2 P2P WRAPPER SERVICE AND RESULTS

P2P Math Service is shown in the Figure 4.4. It gets the parameter and simply doubles the number. P2P Service publishes it self as shown in the “Sending” text area, it discovers the P2P Wrapper Service as shown in the “Receiving” text area. It gets the number from P2P Wrapper Service and sends the result to the P2P wrapper service.

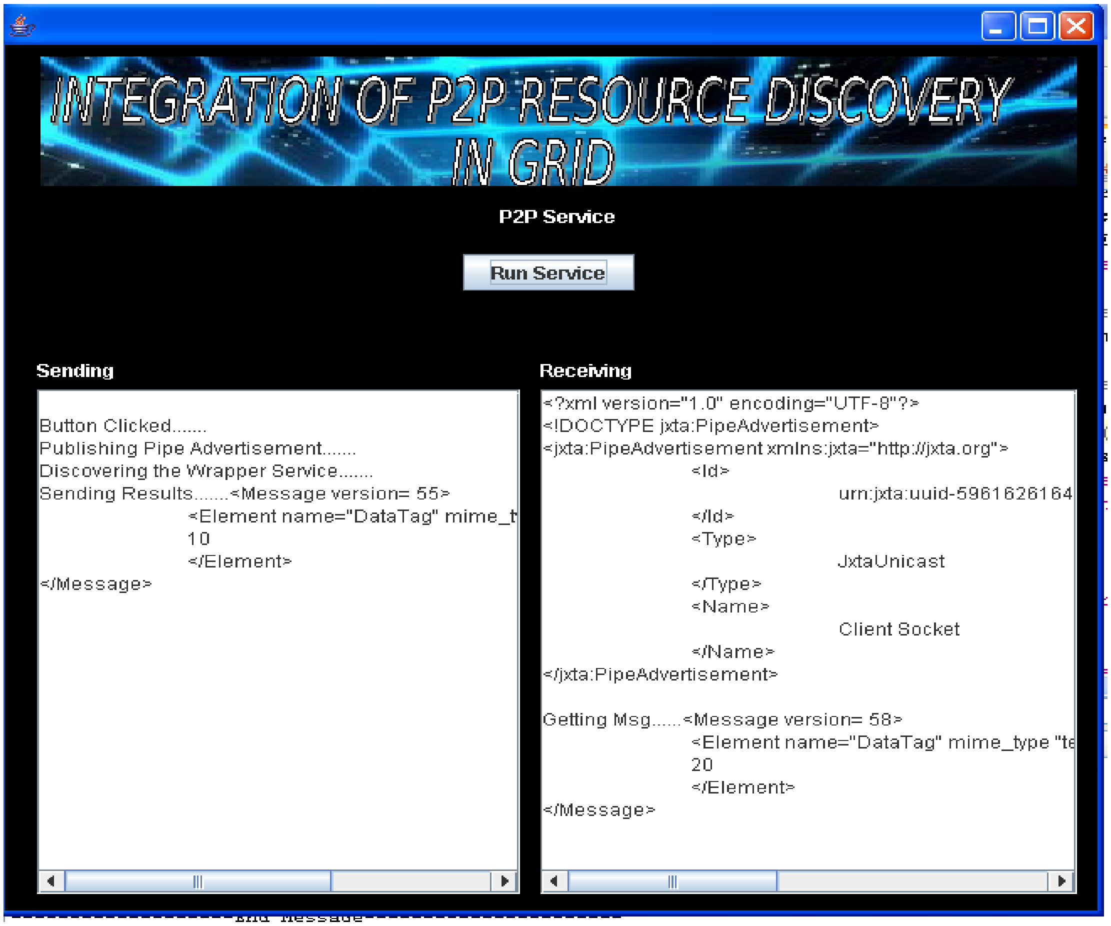
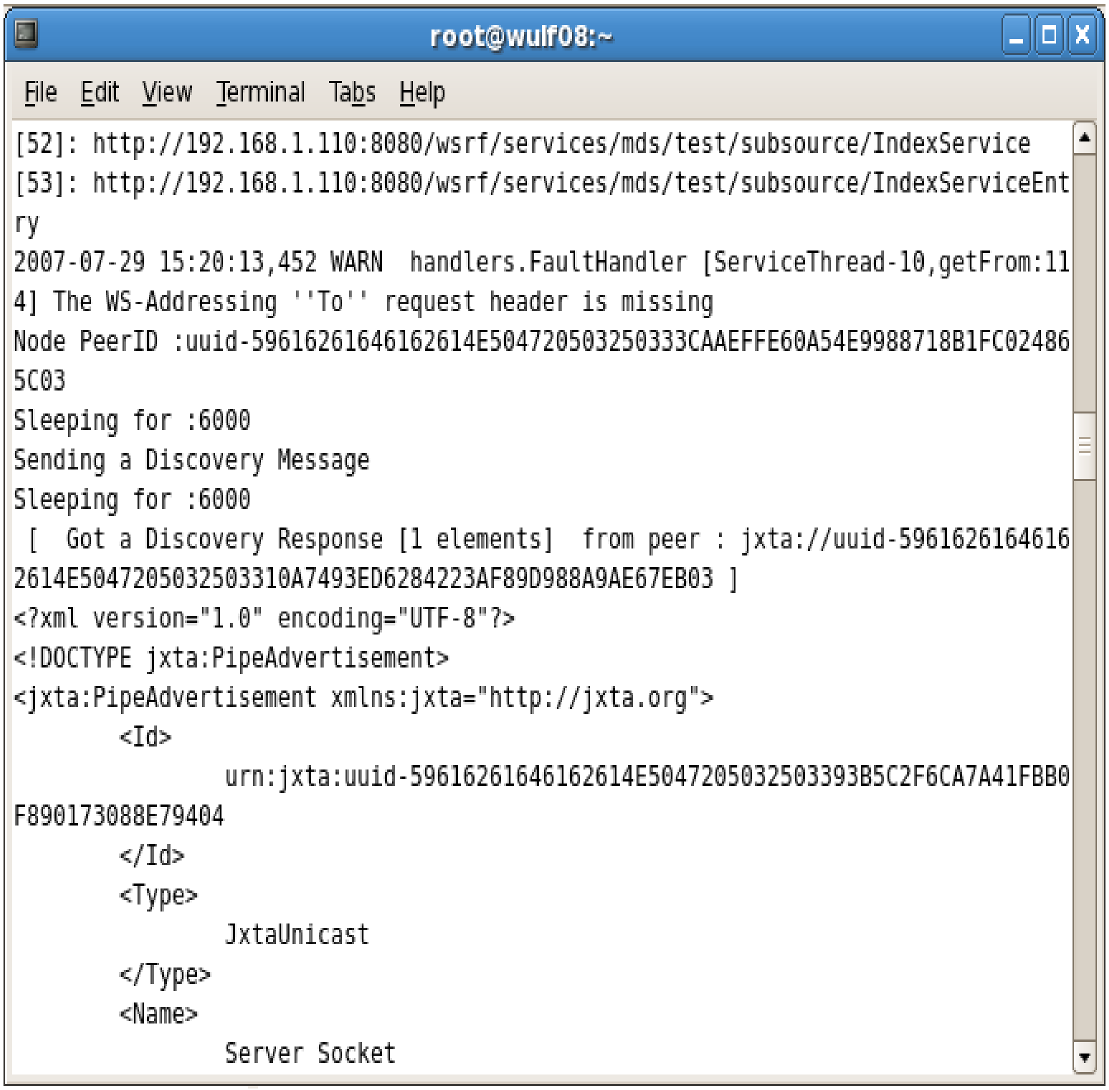



Figure 4.4 P2P Service Screen shot




P2P Wrapper service discovers the P2P Service pipe advertisement. Input output pipes are created. It receives the number "10" from Grid client; this message is mapped into P2P end message and will be sent to P2P Service as shown in the following figure.

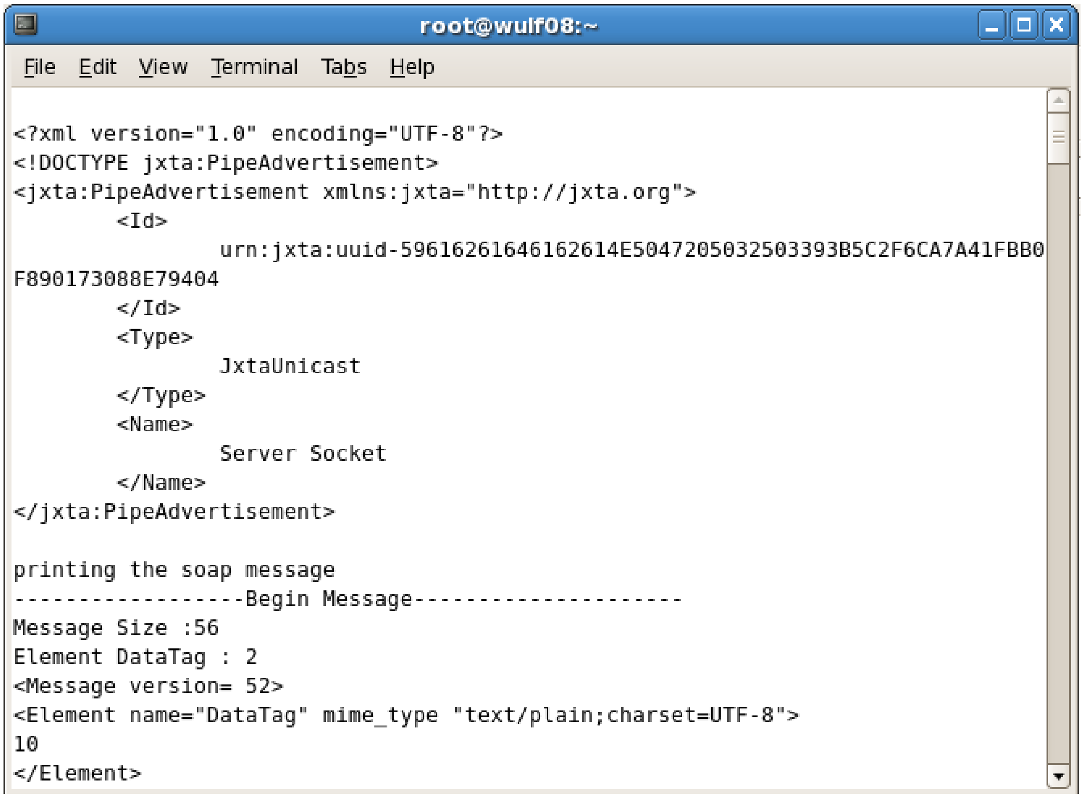



Figure 4.5 P2P Wrapper Service Screen shot

P2P Wrapper service gets the results from the P2P computation Service, in the form of End point messages and sends the results to the Grid Client.

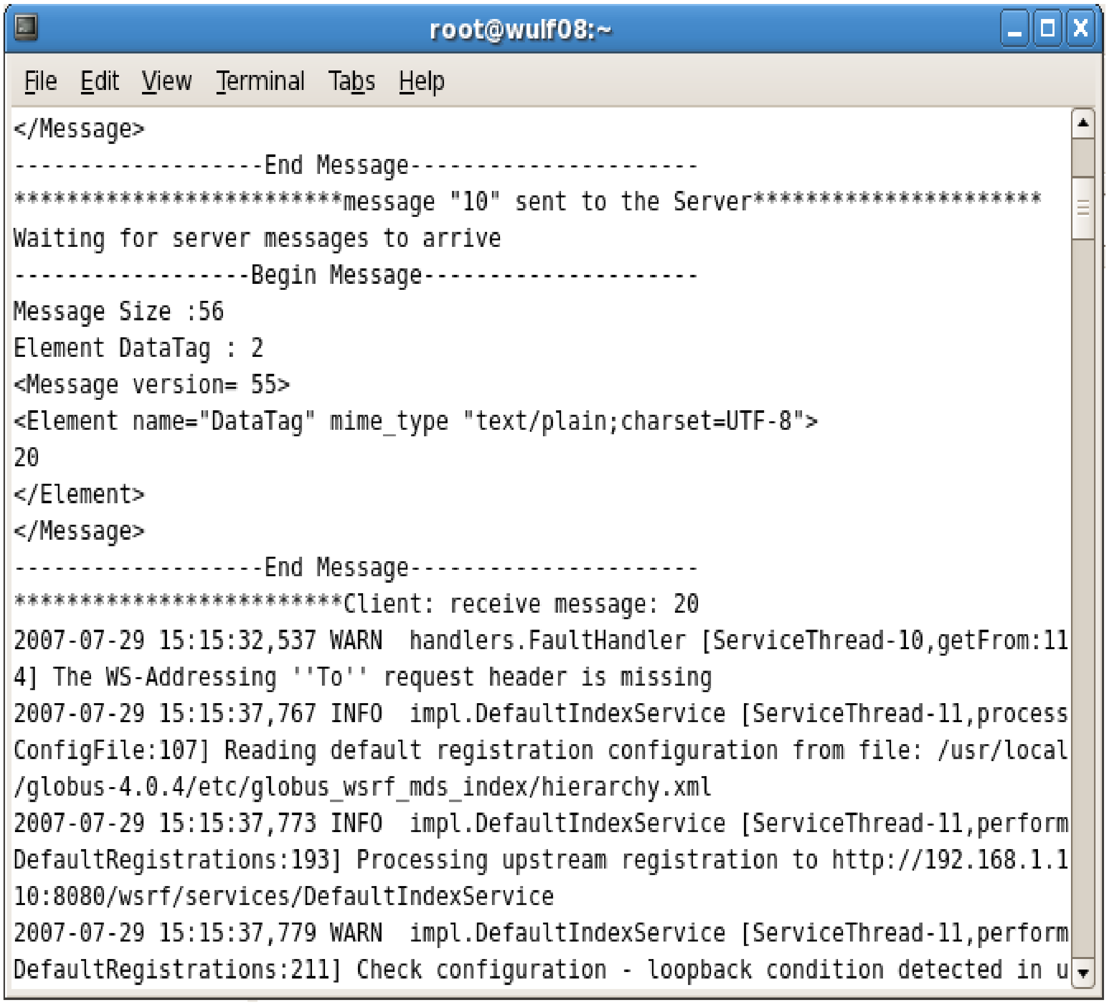



Figure 4.6 P2P Wrapper Service Screen shot

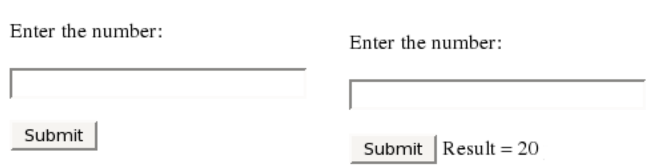



Figure 4.7 Grid Client Screen shot

Grid Client gets the result of the query after the complete mapping.

*Chapter 5*

# CONCLUSIONS & DISCUSSIONS

IPG is a step towards the innovation of Grid and P2P. A service has been developed to make the resource discovery and utilization efficient and scalable. A service which can make the communication possible between Grid and P2P. IPG proves that any service could be developed for this purpose. IPG provides simple, easy to use and easy to integrate with the existing technologies. Grid wrapper service could be used easily just like any other P2P service, by P2P Client, to communicate with any Grid Service, and P2P Wrapper could be used easily just like any other grid webservice to communicate with the P2P service, without making any changes to either of the technologies.

It is nevertheless an effort to converge two different communities and users of Grid and P2P , without making any changes in the architectures. Now whether its P2P or Grid, user of either technology can communicate with the service of either technology.

*Chapter 6*

# RECOMMENDATIONS

It is a step towards scalable Grid but there are number of disadvantages of this approach, only a prototype of the service has been implemented in this project comprising three machines, but it doesn't cover the security issue, it neither considers secure pipes in P2P nor it considers security in globus and signed certificates from Certification Authority in Grid. Secondly , just a prototype has been developed, number of improvements need to be made, A complete system have not been developed which should cater for all the elements of all the messages, third disadvantage is , this scheme is pretty slow , so the implementation should consider latency and other issues to make the communication mechanism faster. Even considering above mentioned disadvantages, IPG provides solution for the communication between Grid and P2P service. Nevertheless, this project is a step towards ubiquitous computing.

# REFERENCES

[1] Ian J. Taylor, (2004), From P2P to Web Services and Grids (Peers in a Client / Server world), Springer, 275

[2] Sotomayor B. , (2004) , The Globus Toolkit 4 Programmer's Tutorial, Borja Sotomayor, University of Chicago.

[3]Foster, I., Iamnitchi, A., (2003), On Death,Taxes and the Convergence of Peer to Peer and Grid Computing. 2nd International Workshop on Peer to Peer Systems, Berkeley, CA

[4] Iamnitchi, A., Foster,I., Daniel C.N, (2002), A peer to peer approach to resource location in Grid Enviroments. Proceedings of the 11th Symposium on High Performance Distributed Computing, Edinburgh, UK

[5] Mastriani,C., Talia,D., Verta A.,(2005) Super-Peer Model for Building Resource Discovery Services in Grids: Design and Simulation Analysis, EGC, Berlin, 132-143

[6]Loo,A.,The Future of Peer to Peer Computing.(2003), Communications of the ACM, New York , USA

[7] JXTA v2.0 Protocols Specification, March 18, 2007. <http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html>

[8] SOAP, April 10, 2007. <http://www.w3schools.com/soap/default.asp>

[9] SOAP-Envelope, April 10, 2007. <http://www.w3.org/2001/12/soap-envelope>

[10] JXTAProgGuide_v2.5, April 17, 2007. <www.jxta.org>

[11] Mastriani,C., Talia,D., Verta A.,(2005) Super-Peer Model for Building Resource Discovery Services in Grids: Design and Simulation Analysis, EGC, Berlin, 132-143

[12] Hudzia, B., McDermott, L., Ellahi, T.N. and **Kechadi, M-T**. (2006) A Java Based Architecture of P2P-Grid Middleware. International Conference on Parallel

and Distributed Processing Techniques and Applications (PDPTA'06), Las Vegas, Nevada, USA (June 26-29)

[13] Talia,D., Trunfio,P., (2004)Web Services for Peer-to-Peer Resource Discovery on the Grid, DEIS, University of Calabria, IEEE Internet Computing, Italy

# *Appendix-A*

**Distributed computing**

Distributed Computing is decentralised and parallel computing, using two or more computers communicating over a network to accomplish a common objective or task. The types of hardware, programming languages, operating systems and other resources may vary drastically. It is similar to computer clustering with the main difference being a wide geographic dispersion of the resources.

**Computer clusters**

A cluster consists of multiple stand-alone machines acting in parallel across a local high speed network. Distributed computing differs from cluster computing in that computers in a distributed computing environment are typically not exclusively running "group" tasks, whereas clustered computers are usually much more tightly coupled. Distributed computing also often consists of machines which are widely separated geographically.

**Grid computing**

A Grid uses the resources of many separate computers connected by a network (usually the Internet) to solve large-scale computation problems. Most use idle time on many thousands of computers throughout the world. Such arrangements permit handling of data that would otherwise require the power of expensive supercomputers or would have been impossible to analyze.

Grid computing or Grid clusters are a technology closely related to cluster computing. The key differences between Grids and traditional clusters are that Grids connect collections of computers which do not fully trust each other, and hence operate more like a computing utility than like a single computer. In addition, Grids typically support more heterogeneous collections than are commonly supported in clusters.

Grid computing is often confused with cluster computing. The key difference is that a cluster is a single set of nodes sitting in one location, while a Grid is composed of many clusters and other kinds of resources (e.g. networks, storage facilities).

**Resources**

One characteristic that currently distinguishes Grid computing from distributed computing is the abstraction of a 'distributed resource' into a **Grid resource**. One result of abstraction is that it allows resource substitution to be more easily accomplished. Some of the overhead associated with this flexibility is reflected in the middleware layer and the temporal latency associated with the access of a Grid (or any distributed) resource. This overhead, especially the temporal latency, must be evaluated in terms of the impact on computational performance when a Grid resource is employed.

Web based resources or Web based resource access is an appealing approach to Grid resource provisioning. A recent GGF (Global Grid Forum) Grid middleware evolutionary development "re-factored" the architecture/design of the Grid resource concept to reflect using the W3C WSDL (Web Service Description Language) to implement the concept of a WS-Resource. The stateless nature of the Web, while enhancing the ability to scale, can be a concern for applications that migrate from a stateful protocol for accessing resources to the Web-based stateless protocol. The GGF WS-Resource concept includes discussions on accommodating the statelessness associated with Web resources access.

# *Appendix-B*

## The JXTA Protocols

The JXTA protocols are a set of six protocols that have been specifically designed for ad hoc, pervasive, and multi-hop peer-to-peer (P2P) network computing. Using the JXTA protocols, peers can cooperate to form self-organized and self-configured peer groups independent of their positions in the network (edges, firewalls, network address translators, public vs. private address spaces), and without the need of a centralized management infrastructure.

The JXTA protocols are designed to have very low overhead, make few assumptions about the underlying network transport and impose few requirements on the peer environment, and yet are able to be used to deploy a wide variety of P2P applications and services in a highly unreliable and changing network environment.

Peers use the JXTA protocols to advertise their resources and to discover network resources (services, pipes, etc.) available from other peers. Peers form and join peer groups to create special relationships. Peers cooperate to route messages allowing for full peer connectivity. The JXTA protocols allow peers to communicate without the need to understand or manage the potentially complex and dynamic network topologies which are increasingly common.

The JXTA protocols allow peers to dynamically route messages across multiple network hops to any destination in the network (potentially traversing firewalls). Each message carries with it either a complete or partially ordered list of gateway peers through which the message might be routed. Intermediate peers in the route may assist the routing by using routes they know of to shorten or optimize the route a message is set to follow.

The JXTA protocols are composed of six protocols that work together to allow the discovery, organization, monitoring and communication between peers:

- Peer Resolver Protocol (PRP) is the mechanism by which a peer can send a query to one or more peers, and receive a response (or multiple responses) to the query. The PRP implements a query/response protocol. The response message is matched to the query via a unique id included in the message body. Queries can be directed to the whole group or to specific peers within the group.
- Peer Discovery Protocol (PDP) is the mechanism by which a peer can advertise its own resources, and discover the resources from other peers (peer groups, services, pipes and additional peers). Every peer resource is described and published using an advertisement. Advertisements are programming language-neutral metadata structures that describe network resources. Advertisements are represented as XML documents.
- Peer Information Protocol (PIP) is the mechanism by a which a peer may obtain status information about other peers. This can include state, uptime, traffic load, capabilities, and other information.
- Pipe Binding Protocol (PBP) is the mechanism by which a peer can establish a virtual communication channel or pipe between one or more peers. The PBP is used by a peer to bind two or more ends of the connection (pipe endpoints). Pipes provide the foundation communication mechanism between peers.
- Endpoint Routing Protocol (ERP) is the mechanism by which a peer can discover a route (sequence of hops) used to send a message to another peer. If a peer “A”wants to send a message to peer “C”, and there is no known direct route between “A”and “C”, then peer “A”needs to find intermediary peer(s) who will route the message to “C”. ERP is used to determine the route information. If the network topology changes and makes a previously used route unavailable, peers can use ERP to find an alternate route.

- Rendezvous Protocol (RVP) is the mechanism by which peers can subscribe or be a subscriber to a propagation service. Within a peer group, peers can be either rendezvous peers or peers that are listening to rendezvous peers. The Rendezvous Protocol allows a peer to send messages to all the listening instances of the service. The RVP is used by the Peer Resolver Protocol and by the Pipe Binding Protocol in order to propagate messages.

All of these protocols are implemented using a common messaging layer. This messaging layer is what binds the JXTA protocols to various network transports. (see Messages)

Each of the JXTA protocols is independent of the others. A peer is not required to implement all of the JXTA protocols to be a JXTA peer. A peer only implements the protocols that it needs to use. For example:

- A device may have all the necessary advertisements it uses pre-stored in memory, and therefore not need to implement the Peer Discovery Protocol.
- A peer may use a pre-configured set of router peers to route all its messages. Because the peer just sends messages to the known routers to be forwarded, it does not need to fully implement the Endpoint Routing Protocol.
- A peer may not need to obtain or wish to provide status information to other peers, hence the peer might not implement the Peer Information Protocol.

Each peer must implement two protocols in order to be addressable as a peer: the Peer Resolver Protocol and the Endpoint Routing Protocol. These two protocols and the advertisements, services and definitions they depend upon are known as the JXTA Core Specification. The JXTA Core Specification establishes the base infrastructure used by other services and applications.

The remaining JXTA protocols, services and advertisements are optional. JXTA implementations are not required to provide these services, but are strongly recommended to do so. Implementing these services provides greater interoperability with other implementations and broader functionality. These common JXTA services are known as the JXTA Standard Service.

**Figure 1. JXTA Protocols**

```
+-----------------------+     +-----------------------+
|  +-----------------+  |     |  +-----------------+  |
|  | Peer Information | <-------> | Peer Information |  |
|  +-----------------+  |     |  +-----------------+  |
|                       |     |                       |
|  +-----------------+  |     |  +-----------------+  |
|  | Peer Rendezvous  | <-------> | Peer Rendezvous  |  |
|  +-----------------+  |     |  +-----------------+  |
|                       |     |                       |
|  +-----------------+  |     |  +-----------------+  |
|  | Pipe Binding     | <-------> | Pipe Binding     |  |
|  +-----------------+  |     |  +-----------------+  |
|                       |     |                       |
|  +-----------------+  |     |  +-----------------+  |
|  | Peer Discovery   | <-------> | Peer Discovery   |  |
|  +-----------------+  |     |  +-----------------+  |
|Peer                   |     |Peer                   |
+-----------------------+     +-----------------------+
```

```
+------------------------+    +------------------------+
|  +-------------------+  |    |  +-------------------+  |
|  | Peer Resolver     | <-------> | Peer Resolver     |  |
|  +-------------------+  |    |  +-------------------+  |
|                    |    |                    |
|  +-------------------+  |    |  +-------------------+  |
|  | Endpoint Routing | <-------> | Endpoint Routing |  |
|  +-------------------+  |    |  +-------------------+  |
|Endpoint               |    |Endpoint               |
+------------------------+    +------------------------+


+------------------------+    +------------------------+
|      Transport         |    |      Transport         |
+------------------------+    +------------------------+
```

A peer may decide to cache advertisements discovered via the Peer Discovery Protocol for later usage. It is important to point out that caching is not required by the JXTA architecture, but caching can be an important optimization. By caching advertisements, a peer avoids the need to perform a new discovery each time it accesses a network resource. In highly-transient environment, performing the discovery is the only viable solution. In static environments, caching is more efficient.

A unique characteristic of P2P networks, like JXTA, is their ability to spontaneously replicate information toward end-users. Popular advertisements are likely to be replicated more often, making them easier to find as more copies become available. Peers do not have to return to the same peer to obtain the advertisements they seek. Instead of querying the original source of an advertisement, peers may query neighboring peers that have already cached the information. Each peer may potentially become an advertisement provider to any other peer.

The JXTA protocols have been designed to allow JXTA to be easily implemented on uni-directional links and asymmetric transports. In particular, many forms of wireless networking do not provide equal capability for devices to send and receive. JXTA permits any uni-directional link to be used when necessary, improving overall performance and network connectivity in the system. The intent is for the JXTA protocols to be as pervasive as possible, and easy to implement on any transport. Implementations on reliable and bi-directional transports such as TCP/IP or HTTP should lead to efficient bi-directional communications.

*Appendix-C*

# The Globus Toolkit 4

The Globus Toolkit is a software toolkit, developed by The Globus Alliance http://www.globus.org), which we can use to program Grid-based applications. The toolkit, first and foremost, includes quite a few *high-level services* that we can use to build Grid applications. These services, in fact, meet most of the abstract requirements set forth in OGSA. In other words, the Globus Toolkit includes a resource monitoring and discovery service, a job submission infrastructure, a security infrastructure, and data

management services (to name a few!). Since the working groups at GGF are still working on defining standard interfaces for these types of services, we can't say (at this point) that GT4 is an implementation of OGSA (although GT4 does implement some security specifications defined by GGF). However, it *is* a realization of the OGSA requirements and a sort of *de facto* standard for the Grid community while GGF

works on standardizing all the different services. Most of these services are implemented *on top of WSRF* (the toolkit also includes some services that are not implemented on top of WSRF and are called the *non-WS components*). The Globus Toolkit 4, in fact,

includes a complete implementation of the WSRF specification. This part of the toolkit (the WSRF implementation) is a very important part of the toolkit since nearly everything else is built on top of it.

However, it's worth noting that it's also a *very small* part of the toolkit. At this point, we'll repeat something we said at the very beginning of the tutorial: At the end of this tutorial you will know how to program stateful Web Services using GT4. This will allow you to progress towards using the higher-level services of the toolkit. However it is important to understand that *you cannot program Grid-based applications using only the Java WS Core included in this tutorial*. This tutorial should be approached as a stepping stone towards more powerful tooling, not as a definite guide on GT4 programming.