Design and Implementation of Grid Based Application for Wireless Devices

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
In
Software Engineering

By:

Neha Sood
(80631013)

Under the supervision of:
Dr. (Mrs.) Seema Bawa
Professor & Head, CSED

JUNE 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "**Design and Implementation of Grid Based Applications for Wireless Devices"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Seema Bawa and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(*Neha Sood*)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Seema Bawa)
Professor & Head
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(SEEMA BAWA)
Professor & Head
Computer Science & Engineering. Department
Thapar University
Patiala

(R.K.SHARMA)
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. My greatest thanks are to my parents who bestowed ability and strength in me to complete this work.

This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. Seema Bawa.** Her enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. I am equally grateful to **Dr. Maninder Singh**, Assistant Professor, Computer Science & Engineering Department, a nice person, an excellent teacher and a well-credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions.

I am most indebted to the P.G coordinator **Mrs. Damandeep Kaur** for her help, assistance and suggestions during my work. I would like to express my sincere gratitude towards Ms. Anju Sharma, Ms. Shashi, full time research scholars at **Centre of Excellence in Grid Computing,** Computer Science and Engineering Department, Thapar University for their unending support.

I would like to express thanks to my friends Shilpi, Gurpreet, Maninder, Rohit, Kunal, Rajiv and Lokesh for co-operation and support. I am also very thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct–indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

(Neha Sood)

(80631013)

Abstract

Grid Computing is the new dimension after Parallel and Distributed Computing. Grid technologies provide mechanisms for sharing and coordinating the use of diverse resources. Grid enables the users all over the world to access available resources remotely as well. These resources include storage, processing power, licensed software and hardware. Today various wireless resources like PDA's, smart phones, laptops have also entered the Grid Computing scenario. These devices have added a new dimension to grid computing known as Wireless Grids.

Wireless Grids allows the ad-hoc sharing of resources on edge devices like PDA's, smart phones, laptops. A Wireless Grid is an augmentation of a wired grid that facilitates the exchange of information and the interaction between heterogeneous wireless devices. Integration of wireless devices with wired grid poses various challenges like mobility, low connectivity, processing power, battery life, security, lack of standard technologies etc.

In this thesis work, a communication channel has been designed and developed between wireless devices and wired grid to access resources. A mobile client is developed in J2ME, which interacts with the Grid gateway. Gateway is developed with the help of Servlets and it acts as an interface between wireless devices and grid services. Services have been developed in Java on Linux Platform and are deployed on Globus container. Client application is deployed and tested on N91 Wireless Device.

Organization of Thesis

Chapter 1 describes in detail about Grid Computing, It describes the evolution of Grid Computing from MetaComputing. Various components of grid infrastructure e.g. OGSA, GIIS, GRAM etc. are explained in brief. This chapter discusses the applications and benefits of Grid Computing. Later on, it introduces Wireless Grids.

Chapter 2 presents the Literature Review of Wireless Grids. It discusses the current state of art in the field Wireless Grids. This chapter covers various projects done on Wireless Grids.

Chapter 3 presents the Problem Formulation for this thesis work and explains the motivation of work.

Chapter 4 presents the complete framework of solution and explain it detail, all the components of GridSphere Framework. It covers all the installation and implementation steps for GridSphere and Wireless toolkit. It also covers the testing of application on N91 wireless device.

Chapter 5 presents the conclusion of this thesis work and discusses the future advancements that may take place in Wireless Grid.

Table of Contents

List of Figures

Chapter 1
Introduction

Grid technology provides the ability to store, share and analyze large volumes of data, ensuring that people have access to information at the right time, which can improve decision making, employee productivity and collaboration [3]. Grid is about virtualization, of both information and workload. As Grid computing evolved, the emphasis has shifted from the early view to the notion of Virtual Organization.

1.1 A Brief History of Grid Computing

Grid computing has its roots in the field of high-performance parallel computing, which has been successful on Massively Parallel Processor (MPP) systems. MPP systems utilize multiple CPUs within a single chassis to produce higher performance manifest through increased throughout. However, such systems become prohibitively expensive for large CPU configurations [19]. The ancestor of the Grid is MetaComputing, which links geographically diverse supercomputing resources via a high-speed network.

One of the first infrastructures in the area of MetaComputing was named Information Wide Area Year (I-WAY), which strongly influenced the subsequent Grid Computing activities. Actually, one of the researchers who lead the project I-WAY was Ian Foster who along with Carl Kellelman published a paper that clearly links the Globus Toolkit, which is currently the heart of many Grid projects, to MetaComputing. The term “Grid” was born at a workshop entitled “Building a Computational Grid” in 1997 at Argonne National Laboratory. The workshop was followed by the publication of the book “The Grid: Blueprint for a New Computing Infrastructure” in 1998. In the book, Ian Foster and Carl Kesselman wrote a definition: “A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities”.

In the article, “The Anatomy of the Grid” by Ian Foster in 2001, the authors refined the definition, considering that Grid computing is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

According to the checklist from Foster, the minimum properties of a Grid system are the following [4]:

• A Grid coordinates resources that are not subject to centralized control (e.g. resources owned by different companies or under the control of different administrative units) and at the same time addresses the issues of security, policy, payment, membership, and so forth that arise in these settings.

• A Grid uses standard, open, general-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, and resource discovery and resource access.

• A Grid delivers nontrivial service qualities, i.e. it is able to meet complex user demands.

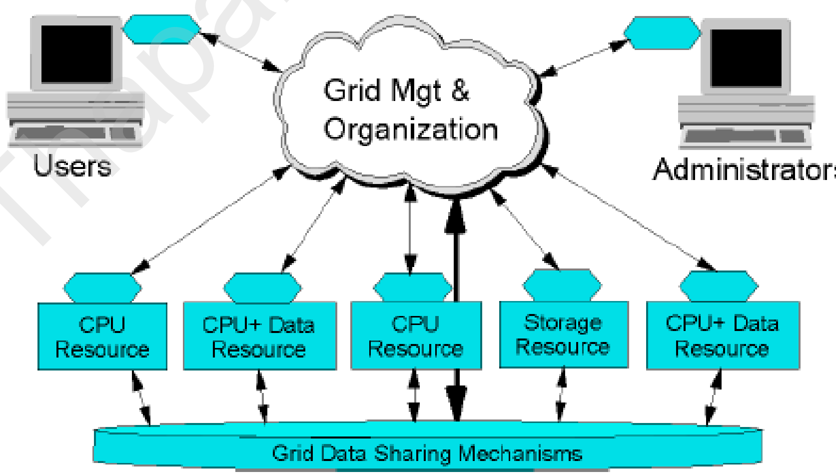


Figure1.1: Grid Computing Environment [3]

In Grid Computing, the key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. "The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization."

1.2 Grid Computing Infrastructure

In general, a Grid Computing infrastructure component must address several potentially complicated areas in many stages of the implementation. These areas are:

- Security
- Resource management
- Information services
- Data management

Let us further examine the significance of each of these above components:

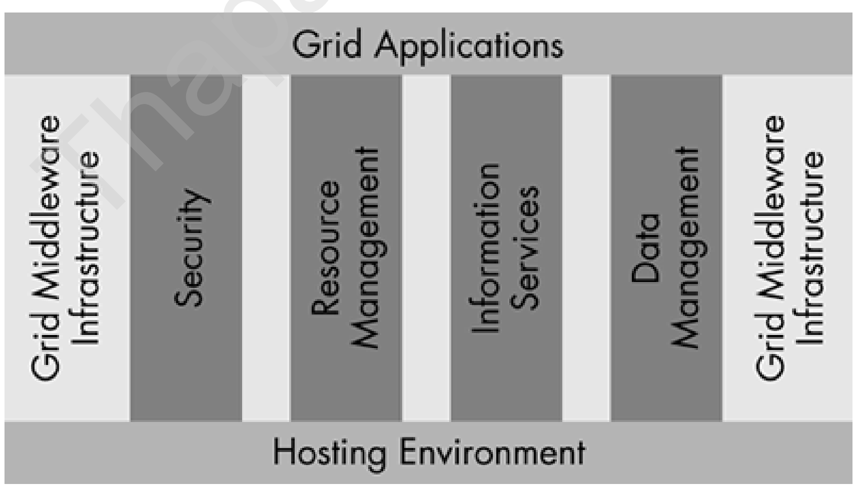


Figure 1.2: Grid Infrastructure Components [30]

Security

The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a Grid Computing environment. These computing resources are hosted in differing security domains and heterogeneous platforms. Simply speaking, our middleware solutions must address local security integration, secure identity mapping, secure access/authentication, secure federation, and trust management.

Resource Management

The tremendously large number and the heterogeneous potential of Grid Computing resources cause the resource management challenge to be a significant effort in Grid Computing environments. These resource management scenarios often include resource discovery, resource inventories, fault isolation, resource provisioning, resource monitoring, a variety of autonomic capabilities, and service-level management activities. The most interesting aspect of the resource management area is the selection of the correct resource from the grid resource pool, based on the service-level requirements, and then to efficiently provision them to facilitate user needs.

Information Services

Information services are fundamentally concentrated on providing valuable information respective to the Grid Computing infrastructure resources. These services leverage and entirely depend on the providers of information such as resource availability, capacity, and utilization, just to name a few. This information is valuable and mandatory feedback respective to the resources managers discussed earlier in this chapter. These information services enable service providers to most efficiently allocate resources for the variety of very specific tasks related to the Grid Computing infrastructure solution.

Data Management

Data forms the single most important asset in a Grid Computing system. This data may be input into the resource, and the results from the resource on the execution of a specific task. If the infrastructure is not designed properly, the data movement in a geographically distributed system can quickly cause scalability problems. It is well understood that the data must be near to the computation where it is used. This data movement in any Grid Computing environment requires absolutely secures data transfers, both to and from the respective resources. The current advances surrounding data management are tightly focusing on virtualized data storage mechanisms, such as storage area networks (*SAN*), network file systems, dedicated storage servers, and virtual databases. These virtualization mechanisms in data storage solutions and common access mechanisms (e.g., relational SQL, Web services, etc.) help developers and providers to design data management concepts into the Grid Computing infrastructure with much more flexibility than traditional approaches.

The Globus Alliance is a community of organizations and individuals developing fundamental technologies behind the “Grid”, which lets people share computing power, databases, instruments, and other on-line tools securely across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The Globus Toolkit includes software services and libraries for distributed security, resource management, monitoring and discovery, and data management. The Globus Toolkit is a middleware and middleware is backbone of Grid. All of these components of this middleware are required to develop Grid applications. [7]

- **GRAM**: The Globus Resource Allocation Manager maps requests expressed in a Resource Specification Language (RSL) into commands that local computer can understand.
- **GSI**: The Grid Security Infrastructure provides authentication services.
- **MDS**: The Monitoring and Discovery Service combine data discovery mechanisms with the Lightweight Directory Access Protocol (LDAP).
- **GRIS:** The Grid Resource Information Service provides information about resources, e.g. configuration, capabilities and status.
- **GIIS**: The Grid Index Information Service coordinates arbitrary GRIS services.
- **GASS**: The Global Access to Secondary Storage implements a variety of data access strategies, enabling programs running at remote locations to read and write local data.
- **GridFTP**: The GridFTP provides a high-performance, secure and robust data transfer mechanism.

Middleware helps to interact with resources but how to run jobs on them is a challenge.

1.2.1 OGSA (Open Service Grid Architecture)

The Open Grid Services Architecture extends the IP network services upward from physical transport of messages to a variety of services that have proven common in Grid applications. OGSA defines a service-oriented architecture, which is the key to effective virtualization of physical resources. OGSA services enable applications to address common Grid requirements for on-demand availability, system management, collaborative computing, and so on. The OGSA builds on existing Web service standards and extends these standards when needed.

Web services are a specific realization of a Service Oriented Architecture (SOA) in which various services interact with each other by exchanging messages in SOAP (Simple Object Access Protocol) (W3C, 24 June 2003) format while the contracts for the message exchanges that implement those interactions are described in WSDL (Web Service Definition Language).

. Web services have several advantages:

• Web services are platform-independent and language -independent, since they use standard XML languages.

• Most web services use HTTP for transmitting messages (e.g. service request, service response). This is a major advantage if building an Internet-scale application, since most of the Internet's proxies and firewalls will not mess with HTTP traffic.

• Web services are more adequate for loosely couple systems, where the client might have no prior knowledge of the web service until it actually invokes it. This is better suited to meet the demands of an Internet-wide application. The key to such success of web services is that it provides uniformity of interface; programs can discover and access resources without the need to write custom code for each specific resource. All the web services act like third party independent components, which are easily included in the applications to provide various functionalities. These web services reduce the code complexity since software only call those services.

OGSA needs to select a sort of widely used distributed middleware to build its own structure. In other words, when defining an interface that has a method, there has to be a Common and standard way to invoke that method. This base for the OGSA architecture could be any distributed middleware (e.g. COBRA, RMI, and RPC). Considering advantages that web services have compared to other technologies, OGSA chooses web services as the underlying technology finally. A grid application will usually consist of several different components. For example, a typical grid application could have:

• **VO Management Service**: To manage what nodes and users is part of each Virtual Organization.

• **Resource Discovery and Management Service:** So applications on the grid can discover resources that suit their needs, and then manage them.

• **Job Management Service**: So users can submit tasks (in the form of "jobs") to the Grid.

1.2.2 WSRF: The Web Services Resource Framework

WSRF is all about the state of a web service. By state, mean the data is not lost in continuous service invocations. Generally, services are stateless which means Web Services cannot remember information from one state to another. But Grid Applications do generally require statefulness. So, it is required that Web service somehow keeps state information. Giving Web services the ability to keep state information while still keeping them stateless seems like a complex problem. Fortunately, it's a problem with a very simple solution: simply keep the Web service and the state information completely separate. Instead of putting the state *in* the Web service (thus making it stateful, which is generally regarded as a bad thing) it is kept in a separate entity called a *resource*, which store all the state information.

Web services are a specific realization of a service oriented architecture in which various services registered in UDDI interacts with each other by exchanging messages in SOAP format while the contracts for the message exchange that implement those interactions are described in WSDL[19]. This description indicates three important specifications for the web service architecture:

- *SOAP* stands for *Simple Object Access Protocol*. SOAP is a simple XML-based protocol to let applications exchange information over HTTP. SOAP is platform independent and language independent. It is a protocol for accessing Web Service. It is a W3C standard. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP also specifies format of message exchange between client and server.

- *WSDL* stands for *Web Service Description Language*, which is an XML-formatted language used to describe capabilities of web service as collections of communication endpoints capable of exchanging messages. The operations and messages are described abstractly, and then bound to concrete network protocol and message format to define an endpoint.

- *UDDI* stands for *Universal Description Discovery and Integration*, an open industry sponsored by OASIS, is a platform –independent, XML-based registry which enables businesses to publish service listings and discover each other, and it has become the de facto standard web service repository. Although UDDI provides a weak discovery mechanism, it provides a data structure to detail a set of characteristics of web services that can help the process of web services discovery and selection.

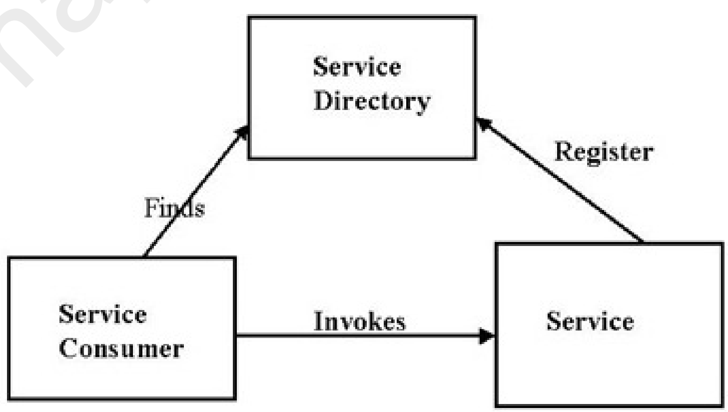


Figure 1.3 Service Oriented Architecture [19]

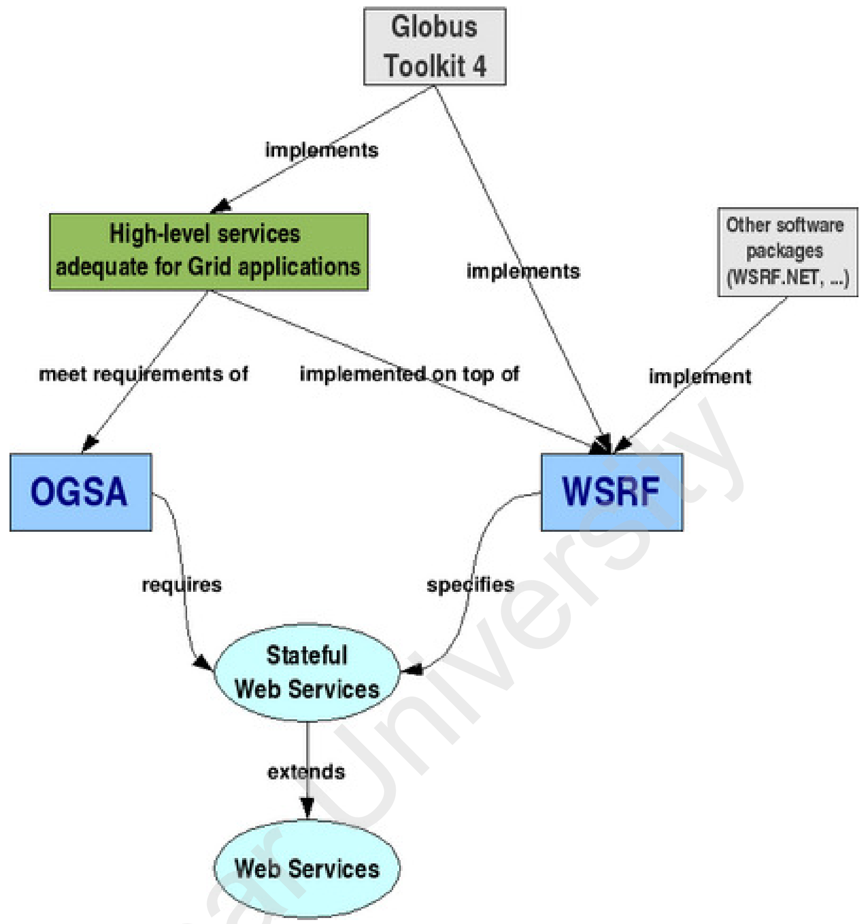


Figure 1.4 Relationships between OGSA, WSRF, Web Services and GT 4 [2]

The Grid community and the Web services communities initially worked without much coordination though Grid implementations often used existing Web technologies such as HTTP or public key based security. More recently, the two communities identified a set of common requirements, which has led to the formulation of a set of specifications known as the WS-Resource Framework, or WSRF. WSRF specifications define how to provide Grid capabilities using Web Services implementations.

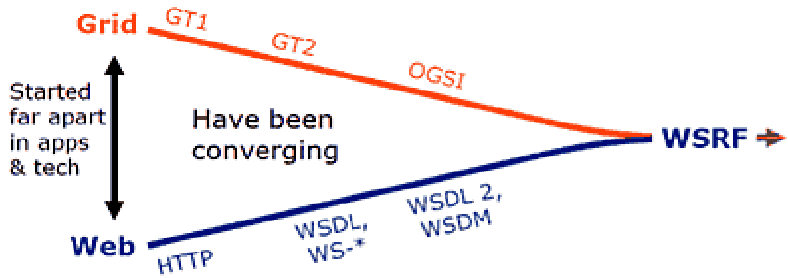


Figure 1.5: Evolution of WSRF [6]

Each resource has a unique *key*, so whenever software requires a *stateful interaction* with a Web service, it simply has to instruct the Web service to use a particular resource

1.3 Applications and Benefits of Grid Computing

Some applications of grid computing, particularly in the scientific and engineering arenas [7]

- Distributed Supercomputing/computational science
- High Capacity/throughput computing: large-scale simulation
- Content sharing, for example, sharing digital content among peers
- Remote software access
- Data-Intensive computing: medical instrumentation and mission-critical initiatives
- Collaborative computing (e-science-engineering)
- Utility Computing /service-oriented computing

Benefits:

- Enable resource sharing
- Make effective use computing resources, including platforms and data sets
- Reduce significantly the number of servers needed by (25-75%)

- Allow on-demand aggregation of resources at multiple sites
- Provide load smoothing across a set of platforms
- Enables the realization of a virtual data center

1.4 Emerging Trends in Grid Computing: Wireless Grids

In past several years, new computing models have emerged in Grid Computing. Grid serves the purpose of integrating all available ideal resources whether it is hardware or software. To integrate these static resources, middlewares are installed. These middlewares can be GLOBUS, ALCHEMI, and CONDOR etc. Thus Grid Computing has become an effective technology to compute big scientific and distributed applications. Today beside these static grid resources, users are equipped with various mobile devices. Wireless Grid Computing is the next step to Grid Computing where integration of mobile devices has been made possible. With ever decreasing size and lowering prices, mobiles are new grid resource. Among all the hardware resources available today, Mobile devices capture a big market. The major advantage of utilizing these mobile resources is there real time data collection feature. Since all these resources are embedded with sensors and cameras, more real time data can be collected increasing the efficiency and accuracy of distributed computing applications. Wireless Grid can be more easily understood from the following figure:

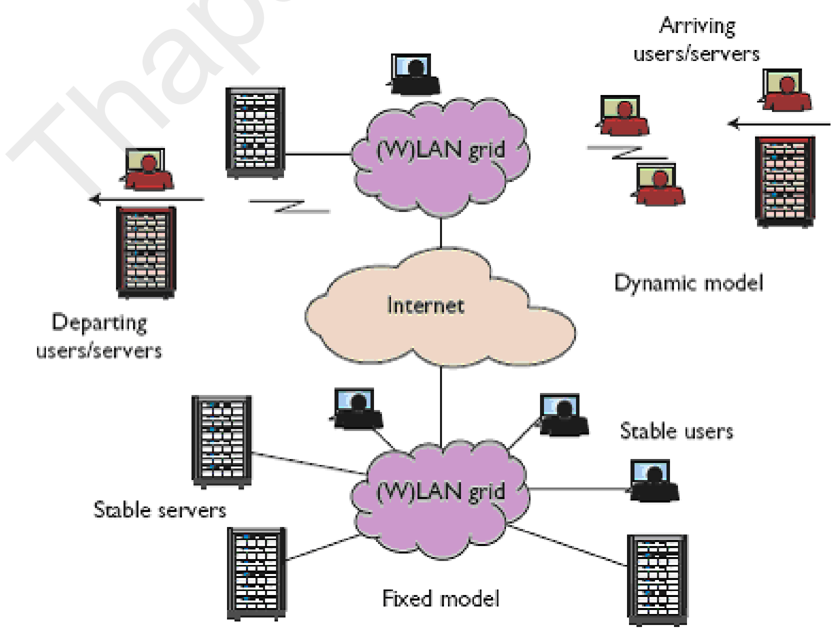


Figure 1.6 Interactions between Dynamic and Wired Grids [15]

In above diagram, interaction between dynamic and fixed wireless grids is established with the help of Internet as backbone. There are two types of wireless grids: those composed of unknown mobile users and devices engaged in ad hoc resource sharing and service creation in a particular location, and those composed of components with known identities managed within a stable institutional structure.

In some ways, wireless grids networks are similar to already found networks in connection with agricultural, military, transportation, air-quality, environmental, health, emergency, and security systems. Big pools of partners, which take benefit from this set up, are governments to very small enterprises by accessing various kind of information available. In other way around, it can be said that sometimes users and producers could sometimes be one and the same. Devices on the wireless grid will be not only mobile but also nomadic — shifting across institutional boundaries. Just as real-world nomadic cross-institutional boundaries and frequently move from one location to another, so do wireless devices.

1.5 Motivation

Grid Computing has emerged as an alternate way of having super computing capability. In Grid Computing, devices distributed all over the world contribute by giving either their hardware or software resources. Internet has played a major role in connecting these widely distributed resources. Various Grid Computing projects running all over the world need to gather information round the clock. Beside desktops, wireless devices are also equipped with better functionalities making them a viable resource for Grid Computing. Thus the motivation behind our work is utilize this large pool of wireless resources with high-end connectivity and developing applications for these devices. These applications will interact with grid services and will provide real-time data for various grid services. In Weather Forecasting Projects, these devices act as sensing devices thereby increasing the accuracy and reliability of results.

Chapter 2

Literature Review: An Introduction to Wireless Grid

Wireless Grid computing, is an extension of traditional distributed and desktop computing, seamlessly integrating various computing systems into our daily life, providing information and services anytime anywhere. With ever decreasing costs and increasing functionality in small sized chips, Wireless handheld devices enrich our daily lives and play vital roles in personal and business productivity. Today, Wireless Grid Computing is still an active and evolving research field, which achieved so far include Wireless Grid networking, Wireless Grid information access, support for adaptive applications, system-level energy saving techniques, and location sensitivity.

2.1 Evolution of Wireless Grids

When Personal Computers first brought closer to ordinary people, they won the first step toward making computers widely available, although it did not enable us to be aware of the full potential of information technology. With the appearance of networking, personal computing evolved to a distributed computing aspect. Distributed computing marked the next step toward Wireless Grid computing by introducing seamless access to remote information resources and communication with fault tolerance, high availability and security [2].

The integration of cellular technology with the Web and wireless LANs in the late 1990s led to the emergence of wireless grid computing. Both the size and price of Wireless Grid hardware device are falling continuously and this provides the new opportunity of building a distributed system with Wireless Grid clients. Wireless Grid computing is an important approach to information access and it prepares the way for the purpose of pervasive computing - “anytime, anywhere”. Pervasive computing is a superset of wireless grid computing [4]. The Wireless Grid computing goal of “anytime anywhere” connectivity is extended to “all the time everywhere” by integrating pervasiveness support technologies such as interoperability, scalability, smartness, and invisibility (Figure 2.1).

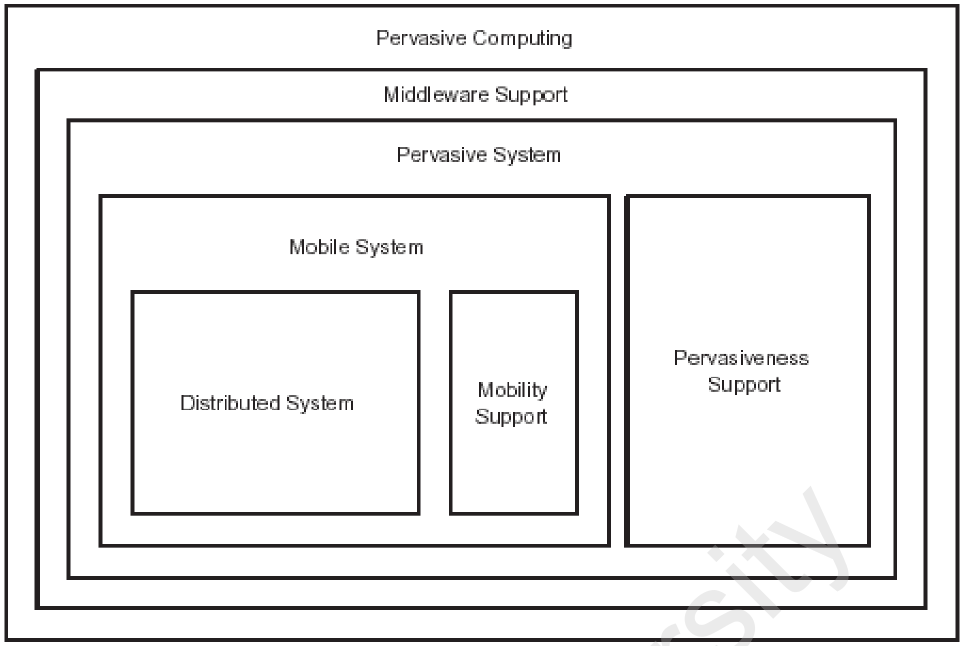


Figure 2.1 Relationships between Distributed and Wireless Grid Computing

2.2 Wireless Grid Computing Model

In order to build a pervasive computing environment, four broad areas are essential: device, networking, middleware, and application (Figure 2.2).

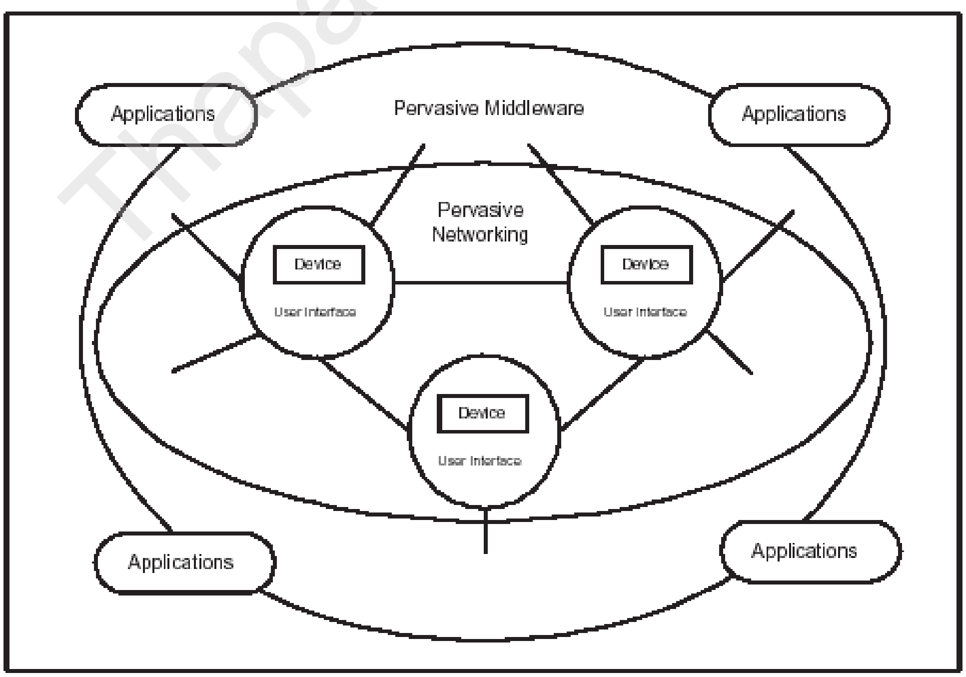


Figure 2.2: Pervasive Computing Framework [1]

Traditional input or output devices, and smart devices are the different device types, which can be contained, in an intelligent environment. The device could be an information source, or a processing centre, or even an agent of various users' actions. The way devices are considered important [2]. They are not only a repository of custom software managed by the user, but a form into an application or data space in addition With the proliferation of wireless devices, strong and robust pervasive networking will to be embedded in our life environment. Extending the current backbone infrastructure and technologies is not a viable way to meet such an anticipated demand. Global networks (e.g. Internet) will need to modify existing applications to completely integrate these pervasive computing devices into existing social systems.

The middleware is required to interface between pervasive computing system kernels (such as various hardware devices, low-level software in general) and the end applications running on different wireless devices. It can provide transparent, autonomous, and continued services, which solve problems arising from mobility and heterogeneity. The final key element of the pervasive computing system is application. The computing environment is the user's information-enhanced physical surroundings rather than a virtual space that exists to store and run software [3]. The application is not a piece of software written to exploit a device's capabilities, but a means by which a user performs a task.

2.3 Current Developments in Wireless Grids

Today, wide range of wireless devices is equipped with better features than ever. It is a well known fact that the number and types of mobile electronic devices worldwide is exploding: from laptops to personal media players, through the ever present mobile phone, there are literally billions of electronic wireless devices that can be connected to a network (just in 2006, more than 1 billion mobile phones were sold)[8]. Thus, it was only a matter of time that someone tries to harness this source of untapped power, be it for using its (rather limited) computing and data capabilities, or for accessing grid services from any place, at any time. Following are the two main current approaches to the task of "mobilizing" grid services:

- Using the mobile nodes just as clients or access points to the Grid.
- Using the mobile devices as conventional grid nodes where jobs (or parts of them) can be executed, thus offering mobile grid services.

Today, focus is on first approach .To fulfill the requirements of approach, till now three approaches have been followed

a) Proxy-Based Solution
b) Mobile-Grid Services

2.3.1 Proxy-Based Solution

One of the first approaches to be proposed to access the Grid from wireless devices was to use proxy-based solutions, as can be seen in various academic works [9].
Proxy-based solutions use what is called a proxy server (or gateway): a special (fixed) grid node, part of the core grid infrastructure, whose role is to handle and translate requests from mobile nodes, and forward them to the Grid, adapting the responses to the limited interfaces of mobile devices. Thus, the wireless devices requests (often using ad-hoc protocols) are handled by the proxy server, which invokes the necessary grid services and sends the response back to the wireless device. Initially, all Mobile Devices (MD) connected to the same proxy are nodes of the same Grid.

However, virtualization techniques (putting several virtual machines in the proxy) allow for isolation among wireless devices connected to the same proxy server. Regarding the technologies used for the implementations of these solutions, Java programming language (more concretely, J2ME) is often used to implement the clients on the MDs, web portals are also used for communicating with the proxy server.

The advantages of using this kind of solution is that one can decrease greatly the complexity of the client software to be run on the mobile devices, at the expense of increasing the complexity of the proxy's software. The increasing capabilities of mobile devices make this advantage less and less important, while its disadvantages (the proxy as a single point of failure and its scalability issues when the numbers of wireless devices grow) continue to be as important as ever.

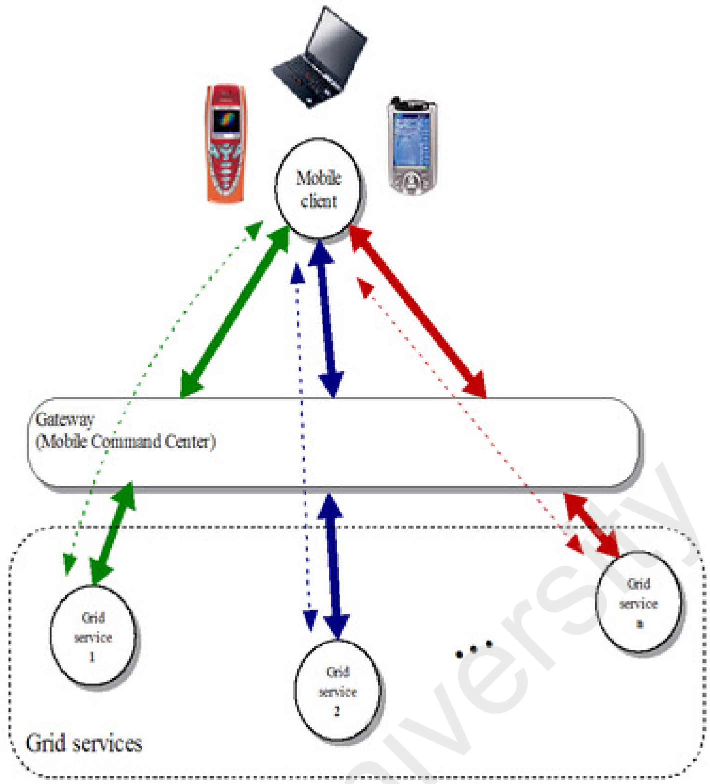


Figure 2.3: GridLab mobile accesses [9]

Various Client side and Server side technologies available for the application development are:

- Server side technologies

  a) *Servlets* are java classes that run on server end. Servlets have access to all resources on the server and are not limited by network access that is imposed on applets. Servlets support thin clients. Computation-intensive functionality can be moved over to the server side and accessed via Servlets instead of forcing the client to download the necessary classes to the client side.

  b) *Web Services* are the server side application components that communicate using open protocols. Web services are discovered using UDDI. XML is basis for web services. Web services are back-end logical components, which can be called by any application and results, are returned to the calling application. Web services can easily communicate with mobile devices through some J2ME application.

c) *Portlet* is java based web component managed by portal container. This solution can be use XSL transformations for serving contents to mobile side. Portlets can be hosted on frameworks e.g. GridSphere Portlet Framework.

- Client Side Technologies

There are plenty of technologies that are emerging in the mobile world nowadays. However, many of them work only on one or several hardware platforms. J2ME and WML browsers are among most successful technologies that give better offline work.

a) *J2ME* stands for JAVA 2 Platform Micro Edition. JAVA J2ME enabled devices (communication via HTTP/HTTPS/SOAP with server side) have small internal repositories that help them to work on database based applications. Profiles are defined for wireless devices which is a set of APIs specific to those devices. MIDP2.0 profile is supported by N91 wireless device. Devices are also categorized according to their hardware configurations, CLDC and CDC.

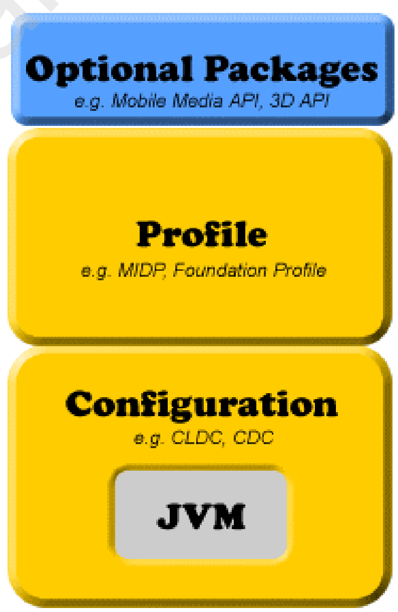


Figure 2.4: J2ME Stack.

2.3.2 Mobile access to Grid web services

Opposite to traditional grid access through a proxy server or gateway, there have been a number of works trying to probe other kind of solutions. Many tried to provide client (only) access to grid web services directly, on the basis of the following motivations [10]:

- Current mobile devices have sufficient resources (computational power and network connectivity) to allow direct access to grid web services.
- Compared to existing proxy-based approaches the usage of web services on the mobile device has only a small and acceptable impact on performance.
- Using standard grid web services allows for better scalability, interoperability and easier implementation of mobile grid clients.

Technologies used in this kind of approach include WSRF, GSI, J2ME, and J2ME Web Services API, kSOAP / kXML [9].

It is important to note that several of these initiatives end up using gateway solutions, not because of their higher efficiency, but on the grounds of the difficulty of developing mobile clients and applications. This fact shows that a good developer environment/framework and a coherent API are probably the major obstacles in the implementation of this kind of solutions.

2.3.3 Mobile Grid Services

The next logical step in the process of mobilization of the Grid would be considering mobile devices themselves as conventional grid nodes where grid services can be executed. This is a much less taken path that only a handful of researchers and projects have tried with mixed results.

2.3.3.1 Mobile OGSI.NET

Mobile OGSI.NET extends an implementation of Grid computing, OGSI.NET (which is a precedent of the more modern WSRF.NET), to mobile devices, on the basis of the paradigm that somehow the resource limited devices can collectively deliver the quality of service needed by the end user [11]. In fact, it allows for process execution and user-driven migration between MD's when a certain event occurs, e.g. the battery is very low or the mobile host is about to fail. Mobile OGSI.NET was implemented on top of the Microsoft PocketPC 2003 operating system and the .NET Compact Framework and at the time only supported PDA's, not mobile phones. Its architecture consists of the following three main layers [11]:

- The Monash University Mobile Web Server, which handles endpoint to endpoint message reception and transmission.
- The Grid Services Module, which handles grid services message parsing and multiplexes messages to the appropriate grid service.
- The Grid Services, which handles application logic and processing. In practice, just basic functionality was implemented; other important grid features like security or notifications have never been available. In fact, it seems that the interest in the project has waned, as not much beyond the work described in [2] has been done on the subject.

Furthermore, the next logical step has not been attempted, which would be updating this initiative to WSRF.NET, creating a "Mobile WSRF.NET". No interoperability tests of Mobile OGSI.NET with conventional grids have been conducted either, to out knowledge.

2.3.3.2 MoGrid

MoGrid [4, 5], a middleware developed by researchers at the Laboratório National de Computação Científica (LNCC) and the Pontifícia Universidade Católica do Riode Janeiro (PUC-Rio), in Brazil, aims to support mobile grid services, by providing means for resource discovery and task distribution in mobile ad-hoc networks. To that end, peer-to-peer techniques are used for resource discovery, and job submission is done through standard protocols like GridFTP. It also sports a so called "transparency layer", in order to isolate the connectivity problems typical in wireless scenarios and make them transparent to the grid applications and services. In its current form, the architecture does not provide the necessary security mechanisms for establishing privacy or trust. Also, it lacks any methods for interoperating with conventional (wired) grids, although plans for proxy-like interoperation with Globus have been mentioned.

Although this project shows some interesting concepts, its current status is uncertain. Some reference implementations were made in J2ME in 2005-2006, for publication purposes, especially in the peer-to-peer resource discovery area, and it seems that some amount of research is still going on in the subject of resource discovery and scheduling [6].

2.3.3.3 Akogrimo

Akogrimo (Access to Knowledge through the Grid in a mobile World) is a project funded by the EC under the FP6-IST program that ran from July 2004 until November 2007. It aims at defining and realizing a web services-based Mobile Grid Architecture ensuring the viability of new business models developed on it.

The participants of this collaborative platform can be mobile [1, 16]. Akogrimo focuses on a number of mobility aspects that can be integrated into grid environments, including:

- Context awareness (including presence information, location data, etc)
- Device mobility
- Session mobility (i.e. moving a session between two devices)
- Temporary disconnection of mobile devices
- Dynamic VOs
- Grid-controlled SIP calls

And these capabilities are shown through a series of implementations of demonstration applications, like for example e-Health applications (patient monitoring and emergency response) or disaster handling and crisis management scenarios. Akogrimo contemplates several kinds of services, most of them using the WSRF standard:

- Network services that communicate context information with higher layers and execute service needs from higher layers. This includes: (M) IPv6 components,
- Network QoS provisioning and Network management services.
- Network middleware that glues the Grid with a commercial mobile network. Auditing is another functionality of middleware in order to keep track of potential SLA violation. Services in this category include: Service discovery,

Session management, Context Management and A4C (Authentication, Authorization, Accounting, Auditing, and Charging).

- Grid infrastructure services: typical grid services adapted to the commercial mobile network: Data Management, Execution Management, SLA Management or Policy Management.
- Application support services that support domain-independent grid applications and allow for cross layer cooperation. This service include: VO Manager, Workflow enactment, SLA Definition and Operational VO Brokers.

In general, Akogrimo is a high-level architecture with little regard to the underlying grid infrastructure. In fact, its current implementation uses Globus and WSRF.NET as its main middlewares, although using other middlewares underneath Akogrimo should be possible (although the complexity of the architecture and services implemented over Globus makes it a rather daunting task).

The Akogrimo project is probably the only one big-scale effort into making the mobile Grid a reality. It has developed a very complete architecture, paying special attention to the industrial viability of their grid models (for telecom operators, service providers, etc.). This high-level approach and their election of Globus as the middleware makes Akogrimo share some of its weaknesses (e.g. setting up a working grid is a complex task). Also, although mobility and mobile devices are at the focus of the project, the actual implementation only covers laptops, not PDAs or mobile phones (probably because of the middleware, which is not suited for mobile devices).

2.4 Challenges of Integration of Wireless Devices and Grid Services

The combination of these two computing models enables us to step into a new realm of high performance Grid access through Wireless devices. At first glance, this combination does not seem appropriate. The client that needs to interact with Grid resources to accomplish a task is required to install and use Grid client end libraries. At present, the existing libraries are relatively resource intensive considering the limitations of Wireless devices. Moreover, most of the Grid applications are developed with the assumption that the end-systems possess client resources for the task at hand and their communication mechanism is reliable.

In the wireless grid environment, users can come into the environment with their Wireless devices and get access to services without any additional requirements. The computing environment needs to detect the user presence or leave and configure services automatically based on various context information. If Grid services are included in the Wireless Grid computing environment, they are required to be made context aware – Grid services can be customized at different cases. One of key features of such Grid environment [2] is highly dynamic; that is Wireless Grid users and Grid services can be integrated on-the-fly.

Architecture to support several Wireless devices in a computational grid must address [3] following limitations:

- Wireless Grid Elements Are Resource-Poor Relative to Static Elements
  - At any given cost and level of technology, considerations of weight, power, size, and ergonomics will exact a penalty in computational resources such as processor speed, memory size, and disk capacity. While Wireless Grid elements will undoubtedly improve in absolute ability, they will always be resource-poor relative to static elements.
- Mobility is inherently Hazardous

  A Wall Street stockbroker is more likely to be mugged on the streets of Manhattan and have his or her laptop stolen than to have the workstation in a locked office be physically subverted. Even if security is not a problem, portable computers are more vulnerable to loss or damage. Applications and the system, this approach permits individual applications to determine how best to adapt, but preserves the ability of the system to monitor resources and to enforce allocation decisions.
- Wireless Grid Connectivity is Highly Variable in Performance and Reliability

  Inside some buildings, a Wireless Grid element may have reliable, high- speed wireless LAN connectivity; but in other buildings, it may only have modem or integrated services digital network (ISDN) connectivity. Outdoors, it may have to rely on a low-bandwidth wireless wide area network (WAN) with gaps in coverage. More recently, research has begun for exploration of application-aware and context aware adaptation in wireless grid computing

- Wireless Grid Devices are poor at Storage and Computational Power

  Today, Wireless devices are coming up with increased storage capacity but that capacity is not enough to complete resource intensive applications.

2.5 Requirements

There is a computing model emerging from the scenarios that Wireless devices want to access Grid services to perform various tasks. As explained in the previous section, Wireless devices need the Grid for computation and integration, and the Grid needs Wireless devices to be the user interface to interact with the physical world. A dependent relationship exists between a number of Wireless devices and the resource-rich Grid environment. When considering how the vision of integrating Wireless devices into the Grid environment might be realized, a number of common requirements are observed [1]:

• A middleware is required to bring Wireless devices into the Grid in a flexible, open and interoperable way.

• A service description, discovery and composition mechanism is required to enable users to achieve their tasks by using available Grid services.

• Context-awareness is required to provide Grid services that are appropriate for the mobile user at the right time, in the right format, at the right device etc.

• A session management mechanism is required to support the characteristic of the mobility.

Chapter 3

Problem Formulation

During the review process, it has been observed that no standard application has been built that make communication between wired grid and wireless devices possible. There are various challenges of integration like mobility, limited connectivity, low computational power described in last chapter. To integrate mobile device, system should have following components:

- Middleware is the major component of Grid Architecture. It acts as the backbone of Grid Architecture. Middleware translates and transfers the request arriving from a workstation. A wireless device transfers its request through gateway to Grid. Various middleware are available e.g. Globus, Condor,
- Gateway is the component of wireless grid architecture which acts as mediator between request by wireless device and middleware. Gateway can be any web application, which is deployed on containers e.g. Tomcat.
- Client is a J2ME application, which works on either a CLDC or CDC wireless device. Client can also be some JSP application, which can easily call a web service at the end.

Wireless Devices e.g. PDA's have features like Wifi connectivity, WAP browsers, J2ME support, Connected Limited Device Configuration (CLDC), Connected Device Configuration (CDC), Better Storage Capacity, XML support. For every mobile device some profile is defined e.g. MIDP profile which provides APIs for particular functionality. Though most of the wireless devices are configured with these features, major limitation is mobility of theses devices, which limits the resource availability.

These devices communicate through protocols like SOAP over either http or https. All the functionalities to be provided to mobile client are provided in form of independent components called web services. These web services are deployed on grid portal and various wireless devices access them through grid portal. Gateway is responsible for providing an interface to these web services.

The main thrust is on the design of such applications which can access underlying grid services using some standard format and act as an interface of Grid to outer world. While here it discuss about standard format, it defines the standard methodology followed to integrate wireless devices and grid resources. There are two different ways to start the communication: First is to directly call the Grid Services and second is to access these grid services as well as web service deployed on container using some gateway application. In this thesis work, two cases are discussed and are analyzed proper considerations.

Here focus is on two specific issues:

1. How can client communicate with the gateway?

2. How this communication is made possible with the help of available software?

To implement and analyze, Simulator Wireless Toolkit has been used which provides environment and support for many CLDC and CDC devices. Thus it became possible test applications for all kind of devices. GridSphere Portal Framework is the container for all Grid Portlets. These Portlets act as gateway between wireless device and grid resources as well services. Client application is deployed and N91 wireless device.

Chapter 4

Design and Implementation of Wireless Grids

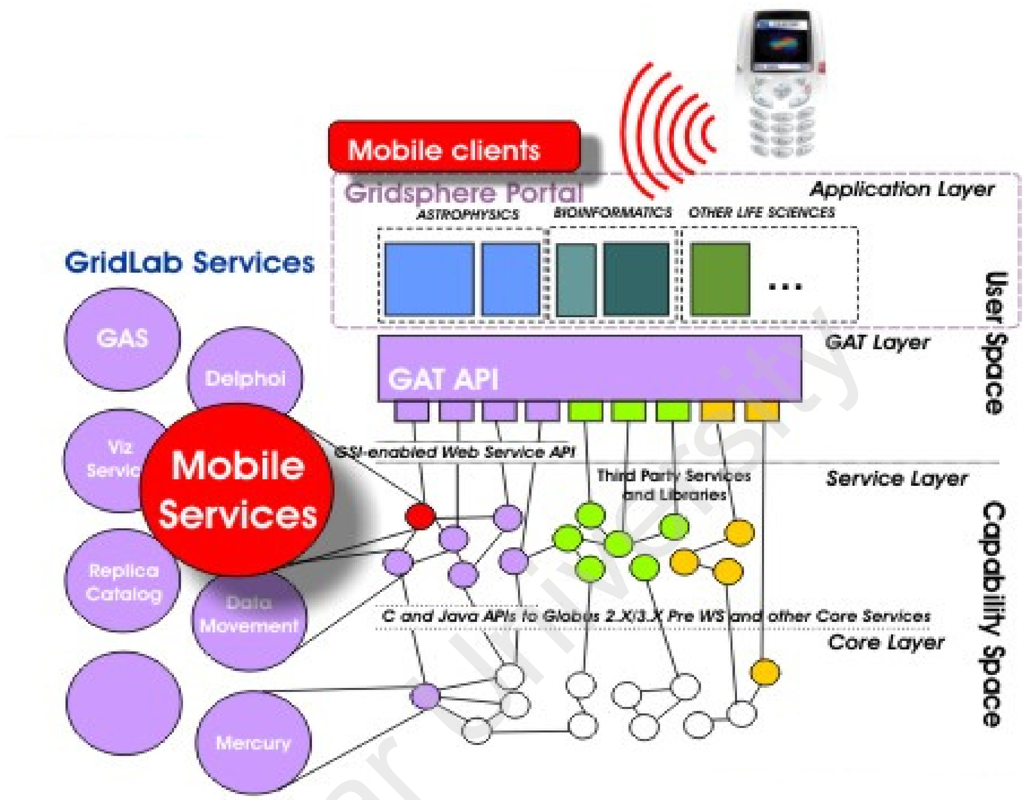
In this thesis work, Wireless Toolkit has been used to implement client application in J2ME and GridSphere Portal Framework acts as Gateway to client application. Client application is deployed on N91 wireless device. This supports MIDP2.0 and CLDC1.1 configuration.

Wireless Grid Configuration in real environment is a complex task since wireless devices are resource constrained. Simulator e.g. Wireless Toolkit provides mobile plug-in and APIs for the development of mobile application. These mobile applications are developed using J2ME (Java 2 Micro Edition). Most of the wireless devices have support for Java applications. Thus J2ME is an appropriate choice for development of client-end applications. These applications can easily access web services hosted on various containers. Web services are stand-alone logical components, which can be easily deployed on applications servers. GridSphere Portal framework can act as link to that web service.

Grid Portal provides an interface to grid resources. GridSphere is a Portal Framework, which provides a platform for the development of reusable portals. It includes a set of core portlets and portlet services that provide the basic infrastructure required for developing and administering Web portals. A key feature of the design of GridSphere is that it builds upon the web application repository (WAR) deployment model to support third-party portlet web applications. In this way, portlet developers can easily distribute and share their work with other portal projects that use GridSphere to support their portal development.

4.1 About GridSphere Portal Framework

The **GridSphere** portal framework provides an open-source portlet based Web portal. GridSphere enables developers to quickly develop and package third-party portlet web applications that can be run and administered within the GridSphere portlet container. This framework is a part of GridLab Project, which includes work packages for services like MessageBox Service, Notification Service. GridSphere framework has following Components:



Screenshot 4.1: GridSphere Portal Framework interaction with Mobile Client

Mobile Client is an application developed in Java. Various Mobile compatible services have been developed using this framework. This framework can also be used to deploy various other web applications.

4.2 Installation of GridSphere Portal Framework

GridSphere portal framework installation require following software packages:

- JDK 1.5
- Apache Ant 1.6.x
- Apache Tomcat 5.5.12
- GridPortlet 1.4

- GridSphere 2.1.5
- Installation of JDK 1.5

  JDK 1.5 is a minimum requirement for GridSphere 2.1.5 installation. It is made available by setting JAVA_HOME path in .bash_profile .

  # export JAVA_HOME=/usr/local/jdk1.5.05

- Installation of Apache Ant 1.6:

  Apache Ant 1.6+ is the basic requirement to make builds. To set up path in .bash_profile file. #export ANT_HOME=/usr/local/ant-1.6.5

- Installation of Tomcat 5.5.12

  Download and unpack tomcat from www.apache.org

  # export CATALINA_HOME=/usr/local/apache-tomcat-5.5.12

  Edit $CATALINA_HOME/conf/server.xml with appropriate port number. Download Xerces for Java and unpack it. Copy this file xercesimpl.jar to $CATALINA_HOME/common/endorsed/. The Portlet Manager Portlet requires the Tomcat manager webapp to be configured to allow the GridSphere user to manage web applications. Modify $CATALINA_HOME/conf/tomcat-users.xml and create a user with the manger and admin role.

  An example is shown:

  <user username="GridSphere" password="GridSphere" roles="manager, admin"/>

- Installation of GridSphere 2.1.5

  Download GridSphere 2.1.5 from www.GridSphere.com/download.html.

  Unpack the tar.gz file into a preferred location. Give the command

  # tar –xzvf <filename>

  Make sure that environment variables for $CATALINA_HOME and $JAVA_HOME are correct, and that Apache Ant binaries are on your $PATH.

  Headless Install

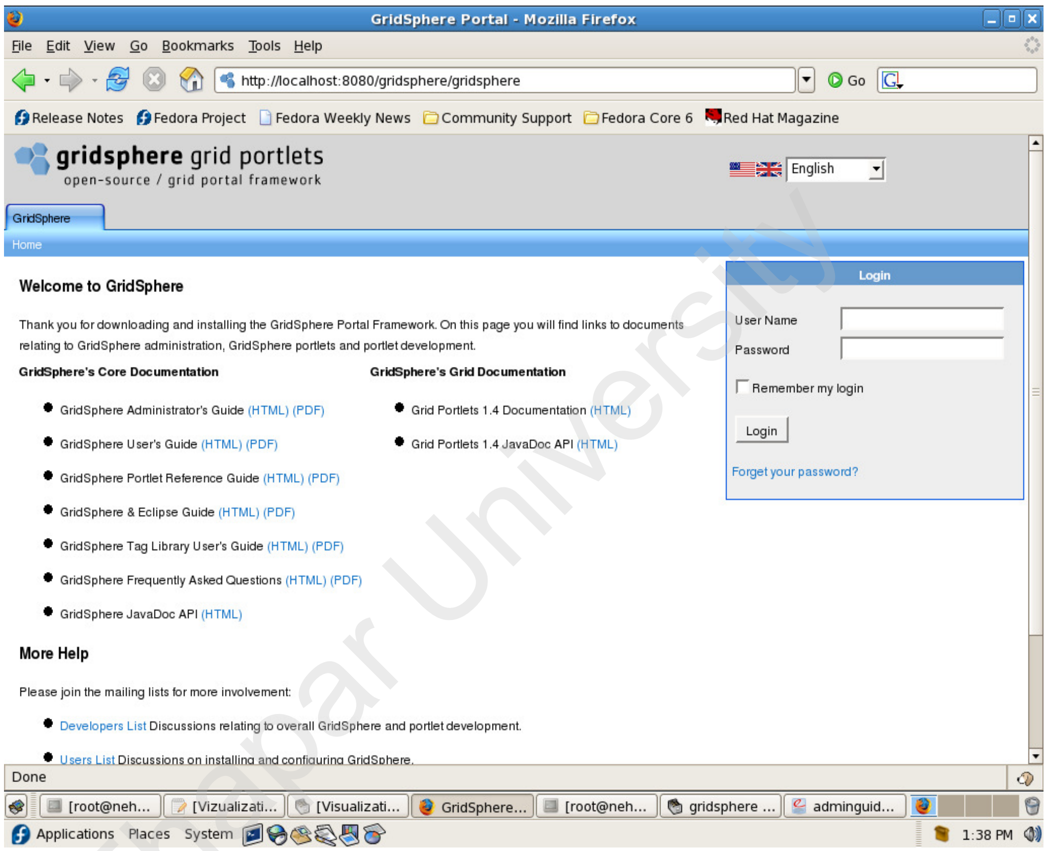
  From the root of the GridSphere source, type:

  # ant compile && ant deploy && ant create-database

  GUI Install

ant install

After setting all the environment variables, to check if GridSphere has been configured completely, check url http://localhost:8080/GridSphere/GridSphere .
If everything works fine, following page will be displayed:

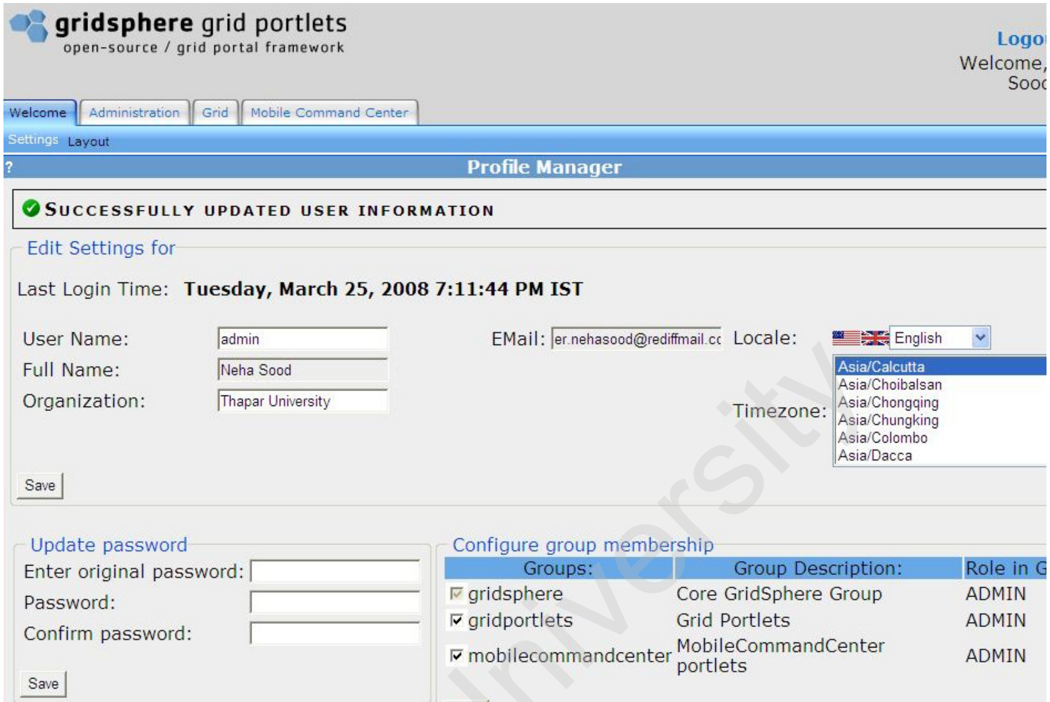


Screenshot 4.2: GridSphere Login Page

To add new portlets to the framework, download grid portlets from www.GridSphere.org and add these portlets to the portlets directory. Give the command

ant install

Grid Portlets are an essential component of GridSphere framework. Portlets are defined as visual components that can be assimilated into portal web pages. Portlets provide "mini-applications" that can either display informational content or provide access to other services. The GridSphere portal allows users to customize their workspace by adding and removing portlets as needed. In addition to standard window states like minimized or maximized, portlets can also provide various "modes" such as *view*, *edit*, *configure* and *help*



Screenshot 4.3: Welcome Screen of GridSphere Framework

Screenshot 4.4 Addition of resources in Resource.xml

This Framework will provide following functionalities:

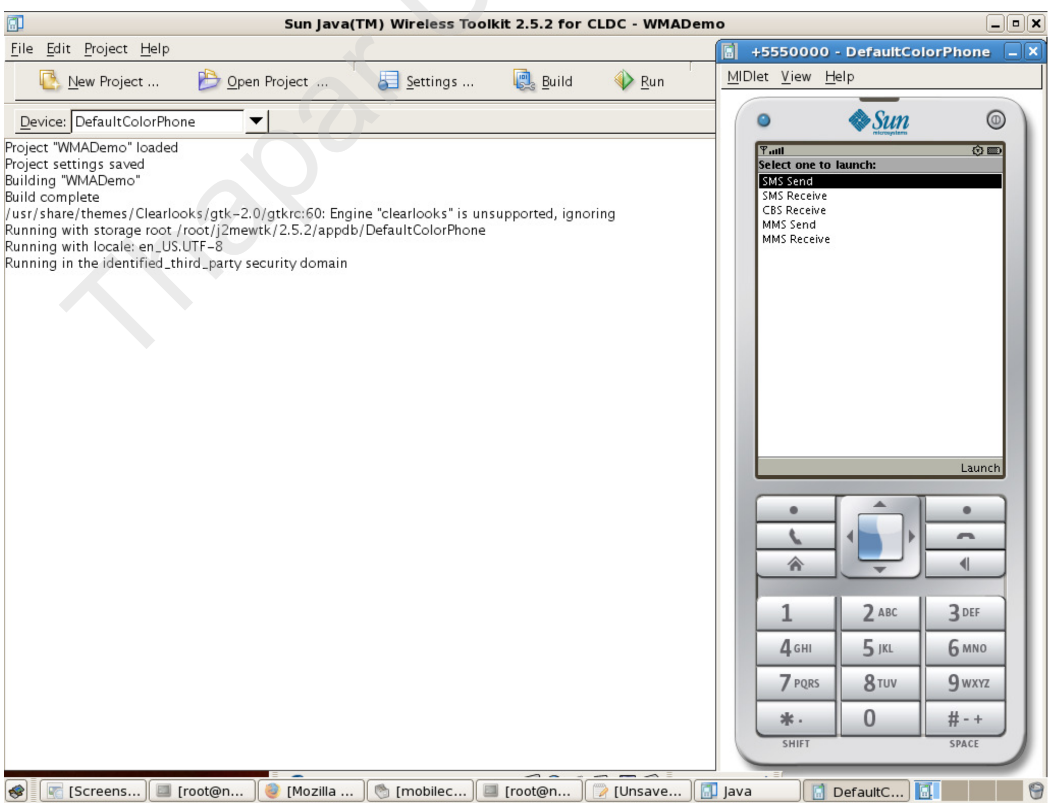
- Deployment of Web-Applications
- Deployment of Services
- Integration with various middlewares
- File Transfer using FTP

GridSphere is integrated Globus cluster. To add a resource, a resource.xml file is edited. This file can be edited from GridSphere framework as shown in figure 4.3 or it can be opened from ./webapp/WEB-INF/Resources.xml.

4.3 Installation of Wireless Toolkit

Download WTK2.1 from www.sun.com and unzip it into some folder. If the package is shell script, double click on the package name and it will run in terminal. Even if it still doest work, check the permissions given to this file. Change these permissions using following command

chmod +x <name of the file>



Screenshot 4.5 Wireless Toolkit 2.1

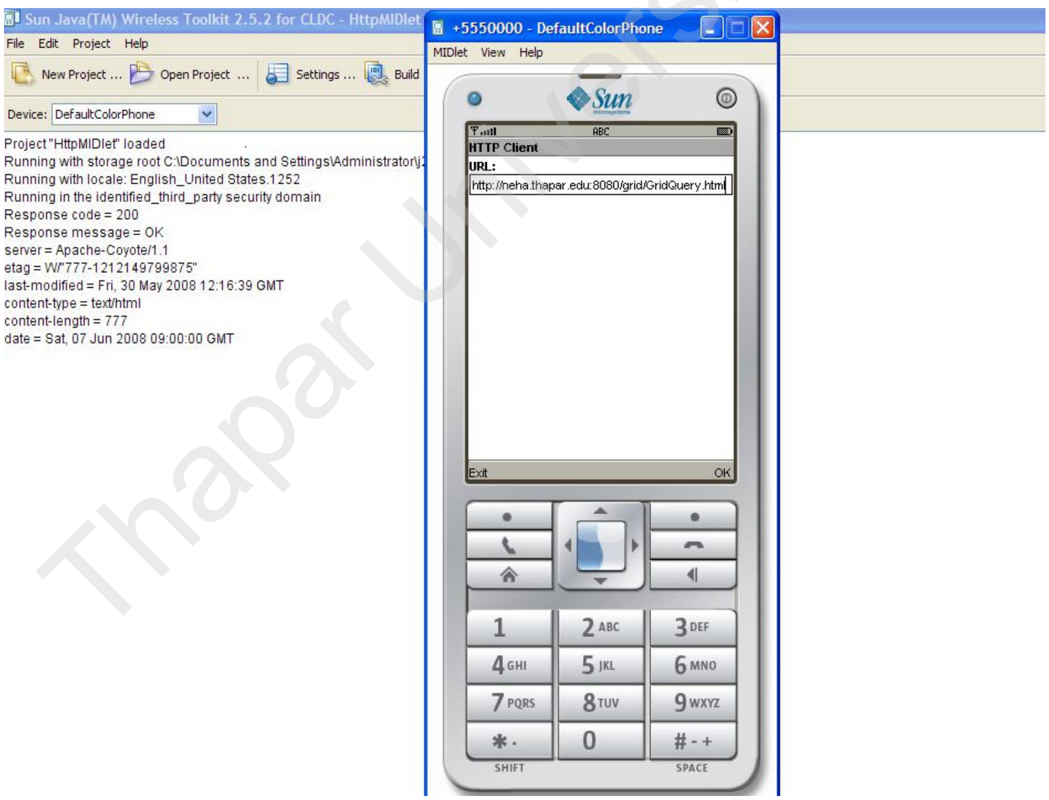
Installation of toolkit is wizard based where minimum requirement is JDK-1.5 and a directory will be created in the root directory. This directory will be named wtk2.1 and the /apps folder will contain all the applications to be run in wireless toolkit. To run wireless toolkit from fedora, go into /usr/local/wtk2.1/bin and give the command.

./ktoolbar

4.4 Experimental Results

To run application in simulator, following are the steps to be followed:

1. Create a Midlet in java and save as HttpMIDlet.java. Create a new project in Wireless Toolkit and name it HttpMIDlet.
2. Copy the source file into the src directory of Project in c:/Documents and Settings/Administrator/Wtk2.1/apps/HttpMIDlet/src.



Screenshot 4.6: Service call from Wireless Toolkit

3. Build the project and preverify it. In this step, HttpMIDlet.jad and HttpMIDlet.jar are created in /bin directory of project.
4. Copy these file on Mobile device and install these applications.

5. This application will call the Grid Database available at the back-end. Grid Database is dynamically updated with all the resources available. Pass the query string http://neha.thapar.edu:8080/grid/GridQuery.jsp.
6. This application is run on N91 Mobile Device.

Screenshot 4.7 Database Results obtained using N91

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The work presented in this Thesis gives an insight into the current developments in the field of Grid Computing and the development challenges in Wireless Grids.

The Thesis work mainly focuses on development of client version of grid services for wireless devices. Among wireless devices, focus is on PDA's, Smart Phones etc. These devices have resources constraints like less computational power, mobility, less battery life etc. Therefore these devices can act as clients only to grid services at present.

The client-side is successfully developed in J2ME and tested on Wireless Toolkit2.1. The application is deployed on N91 wireless device. The gateway is developed in Servlets and Service in Java on Linux Platform.

The main highlights of work are:

- Client-end application is developed in J2ME on MIDP 2.0 and CLDC 1.1 configuration. Most of the wireless devices e.g. PDA's work on these configurations. Therefore application can be successfully tested and deployed on large resource pool.
- Tools required for development are freely available for downloads. Simulators are available for PDA's eg. Blackberry, Palm devices which help to simulate applications in the absence of devices
- Wireless devices can easily access Servlets/JSP pages. Therefore gateway is developed Using Servlets.

5.2 Future Work

- Wireless Grid Computing is emerging and therefore focus of development and research. Major work can be done to make these wireless devices such as Smart Phones, Laptops, PDA's a computational node of Wired Grid.

- Current applications are tested on Smart Phones. It can be deployed and tested on PDA's having same configuration.
- Standards formats are yet to be decided for application developments.
- A standard and secure protocol will be developed because these devices share data in Ad-hoc networks.
- More bandwidth of the spectrum will be allocated to handle the low data transfer rate.

References

[1] Zhi Wang, Bo Yu, Qi Chen Chuanshan Gao, "Wireless Grid Computing over Mobile Ad-Hoc Networks with Mobile Agent", In the Proceedings of the First International Conference on Semantics, Knowledge and Grid(SKG 2005), ISBN 0769525342,2005

[2] Borja Sotomayor, "The Globus Toolkit 4 Programmer's Tutorial"

[3] IBM Red Book, Dec 2005 Edition. <http://www.redbooks.ibm.com>

[4] Ian Foster, "**What is the Grid? A Three Point Checklist",** Argonne National Laboratory & University of Chicago

[5] Piotr Grabowski, Bartek Lewandowski "GridLab Mobile Client User Guide v.1.0.1"

[6] Globus Toolkit Website www.globus.org/

[7] "A Networking Approach To Networking Grid Computing", D. Manoli, John Willey & Sons, Ltd, ISBN 9780471687566, 2005

[8] XtreemOS Integrated Project "Building and Promoting a linux-based operating system to support virtual organizations for next generation grids", project no. ist-033576

[9] GridLab Project website: http://www.gridlab.org

[10] http://mobdev.tknerr.de/wp-content/uploads/2006/09/mobilegridaccess.pdf.

[11] David C. Chu and Marty Humphery, "Mobile OGSI.NET: Grid Computing on mobile devices", In the Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing(associated with Supercomputing 2005),November 2004

[12] Lee W.Mcknight, James Howison and Scott Bradner, "Distributed Resource Sharing by Mobile, Nomadic, and Fixed Devices" Published by the IEEE Computer Society ISBN 10897801/04

[13] Piotr Grabowski, Bartek Lewandowski "Mobile Command Center Admin Guide v. 1.0.1"

[14] Globus website www.globus.org

[15] Sun Website www.sun.com

[16] http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html

[17] Piotr Grabowski, Bartek Lewandowski " GridLab WP12 Access for mobile users" GridLab Zakopane meeting, Dec. 2004

[18] Piotr Grabowski, Krzysztof Kurowski, Jarek Nabrzyski, Michael Russell, "Context sensitive mobile access to grid environments and VO workspaces" ,In the Proeedings of the 7th International Conference on Mobile Data Management(MDM'06), ISBN 0769525261, 2006

[19] Tao Guan, "A System Architecture to Provide Enhanced Grid Access for Mobile Devices", A thesis submitted For the Degree of Doctor of Philosphy, University of Southampton October 2007.

[20] David E. Milliard, Arouna Woukeu, Feng (Barry) Tao, Hugh C. Davis, "Experiences with Writing Grid Clients for Mobile Devices", School of Electronics and Computer Science

[21] Lee McKnight, Raed M. Sharif, Lidwein van de Wijngaert, "Wireless Grids: Assessing a New Technology from a User Perspective"

[22] "J2ME: The Complete Reference", James Keogh ,McGraw-Hill Publications

[23] Ian Foaster, Carl Kesselman, Steven Tuecke, "The Anatomy of the Grid, Enabling Scalable Virtual Organizations"

[24] The Global Grid Forum Web Site, http://www.gridforum.org

[25] The World Wide Web Consortium, http://www.w3c.org

[26] Jean Christophe Durand, "Grid Computing: A Conceptual and Practical Study", University de Lausanne, Nov 8, 2004

[27] J. Joseph, M. Ernest, and C. Fellenstein, "Evolution of grid computing architecture and grid adoption models", IBM System Journals, Vol 43, Number 4, 2004

[28] Akogrimo web site <http: //www.mobilegrids.org/. >

[29] Zhiqi Shen, Angela Goh, Chunyan Miao, Robert gay, "Agent Mediated Grid Services in e-learning" In the Proceedings of IEEE International Symposium on Cluster Computing and the Grid, 2004

[30] http://www.informit.com/articles/article.aspx?p=169508&seqNum=5

[31] http://www.devx.com/tips/Tip/5499

[32] http://web.princeton.edu/sites/isapps/jasig/2003winterMiami/presentations/SangShinportlet.pdf

List of Papers Accepted/Communicated

[1] Neha Sood, Seema Bawa, "Design and Implementation of Grid Based Applications for Wireless Grids" **at IEEE Asia-Pacific Services Computing Conference (IEEE APSCC 2008)** to be held in Jiaosi, Yilan, Taiwan from December 9-12, 2008 **[Communicated].**