

ACCESS FROM J2ME-ENABLED MOBILE DEVICES TO GRID SERVICES

PIOTR GRABOWSKI, BARTOSZ LEWANDOWSKI

*Poznan Supercomputing and Networking Center, piotrg@man.poznan.pl,
bartekel@man.poznan.pl, Noskowskiego 12/14 Poznan, 61-704 Poland*

MICHAEL RUSSELL

Max-Planck-Institut für Gravitationsphysik, russel@aei.mpg.de, Golm, Germany

The article examines the problem of giving the Grid users a possibility to access their applications and resources from any place using mobile devices. According to our approach the devices are incorporated as the clients of Grid services. Moreover, because of limitations of mobile devices this approach assumes adopting a gateway between the client and the Grid. These limitations forced us to pay special attention to building flexible user interfaces as well.

The paper also contains a description of several specialized mobile-oriented Grid services and the roles of a mobile client and gateway played in this services.

Finally, the document deals with issues related to obstacles and limitations that we have faced according to the chosen J2ME technology, and the ways we solved them.

ACCESS FOR MOBILE USERS IN GRID ENVIRONMENT

Grids and GridLab project introduction

The GridLab project is funded by the European Commission under the Fifth Framework Programme of the Information Society Technology, contract number IST-2001-32133 (see <http://www.gridlab.org/>).[1]. The GridLab project is developing an easy-to-use, flexible, generic and modular Grid Application Toolkit (GAT), enabling today's applications to make innovative use of global computing resources. The project is grounded by two principles, the co-development of infrastructure with real applications and user communities, leading to working scenarios, and dynamic use of grids, with self-aware simulations adapting to their changing environment. Grid Computing is an exciting buzzword in the computing world today. The GridLab project defines it to mean the exploitation of a varied set of networked computing resources, including large or small computers, PDAs, file servers and graphics devices. The networks could be anything from high speed ATM to wireless or modem connections. Exploiting these connected resources could, for example, enable large scale simulations not possible on a single supercomputer, aid computational work of geographically distributed collaborations,

simplify remote use of machines, and enable the new dynamic application scenarios the GridLab project proposes.

GridLab WP4 Portals and WP12 Access for Mobile Users introduction

The main goal of the Portals workpackage is to design, build and support the GridLab Portal. In the project, the portal team combines the lessons learned in the development of the Astrophysics Simulation Collaboratory (ASC) [8], a precursor to the GridLab Project, and the Grid Portal Development Kit [5], one of the earliest and most widely used research projects in the Grid portal community. The GridLab portal is based upon the new portal framework, GridSphere, developed by the GridLab portal team and available at its own project website at <http://www.gridisphere.org>. [2]. The primary goal of GridLab is to build a production environment that user communities can benefit from today and in the long-term future. In order to make this goal possible, the current production portal is based on the GridSphere portal framework. The GridSphere Portal is a Java Servlets/JSP [9] based framework that builds upon the **Java CoG Kit** [4], **Globus** [6], and **MyProxy** [7] to support such features as single sign-on, job submission, and data movement. As a production environment, the GridSphere Portal provides online tools for administering users of the portal, as well as the Grid resources and services users access. The GridSphere portal framework is based upon the portlet model. Essentially, a portlet specifies a "mini-window" within a Web page, where a Web page can consist of multiple portlets. Currently, the GridSphere framework is based upon the Portlet API similar to that of IBM's WebSphere. However, the portal team nearly succeeded having an implementation that conforms to the Portlets Java Specification Request (JSR 168) [15] which specifies the committee approved Portlet API.

Another GridLab workpackage which use some concepts of the portal workpackage and GridSphere facilities is the Access for Mobile Users workpackage (GridLab WP12). The main goal of this specialized workpackage is to make use of small and flexible mobile devices that are increasingly used for web access to various remote resources. This working package wants to provide grid access mechanisms for such devices. This requires adoption of the existing access technologies like portals for low bandwidth connectivity and low level end user hardware. The mobile nature of such devices also requires flexible session management and data synchronization. This workpackage is enhancing the scope of present grid environments to the emerging mobile domain. Utilizing the new higher bandwidth mobile interconnects, very useful and previously impossible scenarios of distributed and collaborative computing can be realized. To achieve this and taking into consideration some still existing constraints of mobile devices, the Access for Mobile Users workpackage is developing a set of applications in the client-server model with the J2ME CLDC/MIDP-client, and portlet server working with GridSphere.

PROJECT DEVELOPMENT

Grid services introduction

During the first months of the project duration, the WP12 team collected end-user requirements regarding access to the grid and grid environment itself. In case of typical grid application the mobile developer is faced with a large amount of data and heavy weight protocols that are used in connection between entities in the grid. In fact the data amount with some kind of scientific applications is a number in petabytes – this is true that only a fraction of it (numbers in megabytes) has to be presented to the user in a form of visualization for example. However, even in this case it is not possible to use the service by the mobile side as it is. There is no need to send the data that simply can not be utilized on the opposite side through the network. Additionally, the services in the GridLab project are developed in a form of web-service using OGSA Grid Security Infrastructure [11], Transport Layer Security (HTTTPG) [12] which is secure and straightforward. However, it requires much processor power.

Grid requirements and Mobile weaknesses

An additional requirement that was issued (besides using Web Services and secure client on the mobile side to talk with grid-services), is the need to enable some offline practice on the mobile side – in fact the mobile client should allow the end-user to work with limited functionality, even when explicitly off-line. Taking into consideration the above wishes, together with well-known mobile devices restrictions like limited memory, low screen resolution and colors, insufficient processor power, low network bandwidth and its intermittence, together with limited input possibilities, we decided to make use of those days newly emerged technology: Java 2 Platform, Micro Edition (J2ME) [10]. Although it suits our needs best, it still has/had some limitations. Java CLDC/MIDP 1.0 offers as standard only HTTP, the MIDP 2.0 version which provides HTTPS is not widely used (too few compliant devices yet). The 3-rd parties solution for webservices and SSL that we tested, even those light weight, were too heavy for simpler devices (the problems are connected mainly with memory shortages, suit size limitations and WAP [13] gateways/GSM providers policies).

Mobile Command Center – the gateway to the Grid

The aforementioned problems have convinced us to develop a gateway on the server side, which is working with other grid services. It is serving only the needed data in a form and size suitable for weaker devices, securing only the most important data. In our approach it is the Mobile Command Center - a web application working under GridSphere and acting for mobile devices as a gateway into the grid. Although the main problem seem to be solved, we still have some problems to overcome. One of the examples could be finding the most suitable keyboard mapping for a maximally wide range of devices. The main aspect of the problem is the need for custom keyboard mapping for each device group, without versioning of the Midlet application for each device. One of the possible

solutions for a sophisticated user could be the possibility for remapping the keyboard to suit his/her needs best. However, this solution can not be applied to programs used by a broad community; it still can help the most experienced users (who in fact our users are). The mobile command center is a central point of our architecture (see the Figure 1). All http requests sent from the J2ME MIDP enabled device are served there. If the request requires to call the external service (like Visualization Service, Message Box, Grid Authorization Service or MetaData Service), the request is translated into an appropriate form (e.g. GSI Grid Service GSOAP call) and forwarded to the external service. In this way the request regarding the list of message folders or messages of a given user is forwarded to the Message Box Web Service and the request for an optimal form of a given visualization is forwarded to the visualization service. The Message Box service is used for managing and sending short messages to users. The message can be sent as Email, SMS or even Fax to the end user; additionally, the messages are stored for future retrieval/management from a mobile device into the folder structured persistent repository. The Visualization Service for Mobiles is a part of a bigger application developed by GridLab workpackage 8 – Data and Visualization. The “mobile” interface part of this service allows to build PNG images from almost any source image format, the dimensions of an output picture can be set as exactly needed by a given device; also, the cropping function is provided. To allow performing easy “dataless” zooming, the interface function parameters for cropping are percentage values of the whole picture (this allows the client application to not even know the real dimensions of the source image) and the multiple crop is always performed on the first source image. Additionally, images prepared for small screen devices are antialiased for better presentation. Another service which is called from the Mobile Command Center is GridLab GRMS (Grid Resource Management System); this service allows the user to steer his/her applications: start, suspend, resume, cancel, migrate a job or just get history or information about it. All services in GridLab use GAS (Grid Authorization Service) and the Mobile Command Center also does it. Requests from a given user are checked if they can be forwarded to other services; also, some other service methods use GAS to get an authorization decision about various actions.

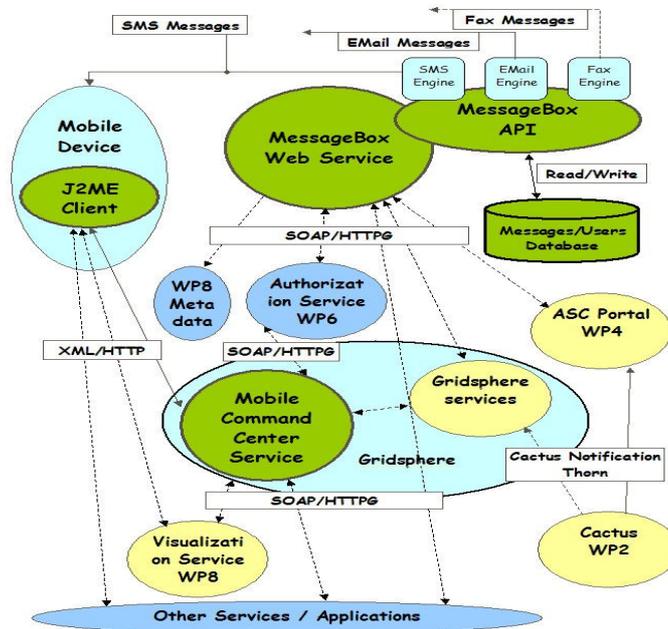


Figure 1. Information exchange between entities in GridLab – part affected by WP12 Access for Mobile Users workpackage activities.

All of the aforementioned services can be accessed from a mobile device via the Mobile Command Center. On the mobile side there is the J2ME MIDP 1.0 enabled device which is running our GridLab Mobile Client Midlet application. You can see what it looks like in the following figures (see Figure 2-5).

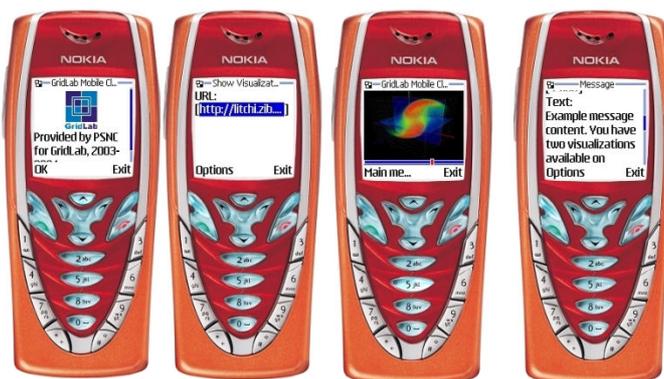


Figure 2. Example screen shots with GridLab Mobile Clients – Nokia 7210 MIDP SDK v1.0 emulator.

The client is a multithreaded Java application which sends HTTP requests to the server and displays responses in an appropriate way. To avoid authenticating the user with each request, there is a session maintaining (with cookies) implemented in our midlet. All typed data is stored in RMS record stores to avoid frequent retyping. After login to the server the user can choose which service he/she is going to use, choosing entries from the menu (e.g. displaying folders from the Message Box, Showing Visualization Service visualizations or displaying the list of user jobs managed by GRMS). There is a possibility to automatically parse messages for URLs with visualizations – the user does not have to type anything in this case. An application after finishing some tasks can send the information about it via the Message Box – the user gets this information as an SMS and using our client can browse his/her messages, choose the message he/she is interested in, and display the message. If the message contains information about visualizations, the user is notified about it and can display one of them with the client. If the visualization consists of more than one image, the navigation slider is displayed and the user can choose which ‘frame’ of visualization he/she wants to display. Each visualization frame can be cropped and zoomed. The zoom operation can be repeated numerous times, in this case the crop operation is always performed on the source image not currently displayed on the device – this allows us to prevent data loss (the source images are high resolution – even 1000% zoom on 100 by 100 pixel screen are clear and can be easily read by the user). The cropping procedure is also user-friendly – a resizable image sub-frame is displayed, the user can move it over the picture, resize the frame if needed and request the needed part of the picture with one key press, and the sub-picture is automatically enlarged to the device screen resolution.

Another part of the GridLab Mobile Client is connected with the GRMS service. The user can display a list of currently running jobs, start the job or cancel, suspend, migrate or resume the existing jobs. There is also a possibility to display job information or history.

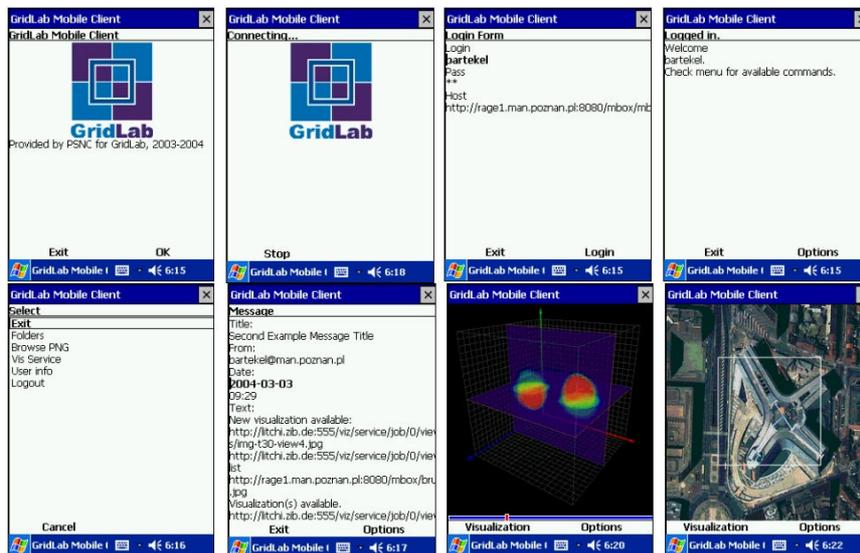


Figure 3. Example screen shots with GridLab Mobile Client – picture taken from Ipaq with running Jeode Java [14] and ME4SE package.

In the pictures in Figures 3 and 4 you can see how the user may login to the server, display his/her messages, choose visualization from the visualization set and zoom a part of the visualization. On devices that do not provide native J2ME MIDP support we are using custom device Java with the ME4SE package which allows to run J2ME midlets under Java SE environment.[3].

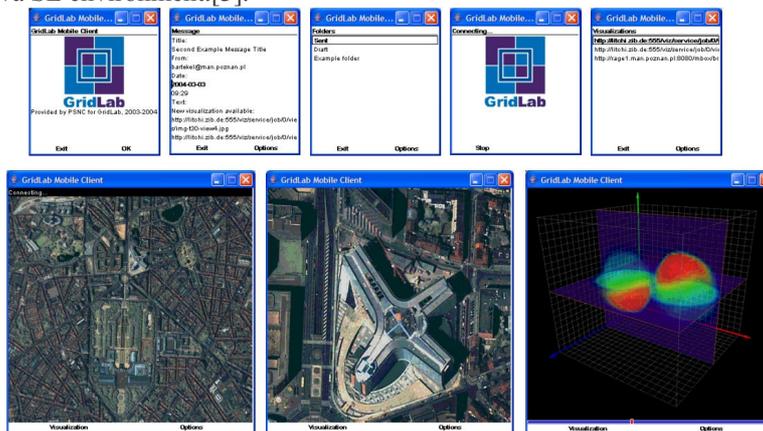


Figure 4. Example screen shots with GridLab Mobile Clients – picture taken from a standard screen device (like a laptop or a desktop computer) running Java 2 Platform SE and the ME4SE package.

As one can see, our work for PSNC and GridLab on Mobile Appliances is highly dependant on collaboration with other institutions (like HP/Compaq or SUN) and we make an extensive use of solutions provided by our consortium partners (GridSphere, GRMS, Visualization Service) and external community/companies (ME4SE). However, such strong dependence can cause many problems during development, we are strongly encouraged to do it as it is the only way not to 'reinvent the wheel' and give our users the best tool we can provide to access the Grid from a mobile device.

REFERENCES

- [1] GridLab project participants, <http://www.gridlab.org>.
- [2] Russell M., Novotny J., Wehrens O., "GridSpher", <http://www.gridsphere.org>.
- [3] Michael Kroll and Stefan Haustein, "ME4SE", <http://www.me4se.org>, (2003).
- [4] Java CoG Kit, <http://www-unix.globus.org/cog/>
- [5] Grid Portal Development Kit, <http://doesciencegrid.org/projects/GPDK/>
- [6] Globus, <http://www.globus.org/>
- [7] MyProxy, <http://grid.ncsa.uiuc.edu/myproxy/>
- [8] Astrophysics Simulation Collaboratory, <https://www.ascportal.org/>
- [9] Java Servlets/JSP, <http://java.sun.com/products/servlet/index.jsp>
- [10] Java 2 Platform, Micro Edition (J2ME), <http://java.sun.com/j2me/>
- [11] OGSA Grid Security Infrastructure, <http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/>
- [12] Transport Layer Security (HTTPG), http://www-unix.globus.org/toolkit/3.0/ogsa/docs/transport_security.html
- [13] WAP, <http://www.wapforum.com/>
- [14] Jeode, JVM Implementation of PersonalJava and EmbeddedJava Specifications, <http://www.insignia.com/>
- [15] Portlets Java Specification Request (JSR168), <http://www.jcp.org/en/jsr/detail?id=168>, <http://www.jcp.org/en/jsr/detail?id=168>