

1. Introduction

In this project we will design, implement, deploy and test various versions of a single grid service from a basic service to sophisticated service with features such as logging and notification enabled. During the lecture classes we examined the description, architecture of and infrastructure supporting a Grid Service. For details of a grid service, grid service architecture (Open Grid Services Architecture: OGSA), grid services infrastructure (Open Grid services Infrastructure: OGSI) and the hosting environments see the comprehensive paper on this topic in [1]. The software that we will be using is the core of the Globus Toolkit 3.0.2. The core of the Globus can be downloaded from <http://www-unix.globus.org/toolkit/download.html#core> . The details of the core are available in a white paper on the core services at <http://www-unix.globus.org/core/> . This white paper also contains a javadoc-style Grid Services API description, User's Manual and a Programmer's Manual. The user's manual provides the instructions to compile, build, convert, deploy and test a grid service. The programmer's manual provides the details of writing a grid service, the various programming choices available, and deployment description. A samples directory in the core package provides a numerous examples illustrating the various grid services features.

[1] **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration.** I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002. (extended version of [Grid Services for Distributed System Integration](#)) [Citation, PDF]

2. Purpose (Goal of the Project)

Implement a suite of grid services ranging in complexity from a basic service to a sophisticated one for the weather service you implemented in the Project 1. The range of services in some ways represents the different qualities of service that a grid service can offer. Present the suite of services in a GUI interface for the user/client to choose, activate and execute. Prepare a GAR file of the deployment for later submission into CSE, CCR or Geneseo Grid.

3. Technology Requirements

Java 2 platform standard edition (J2SE 1.4 or later), Globus Tool Kit, core only.

4. Assignment

Write the weather service in Project 1 as a grid service. Let the weather service offer at least three functions/operations. The samples directory of the Core Globus download has many examples such as *counter* and *google*. The Counter example has sample code for a variety of implementations of the same counter. Among the features illustrated we are

interested in: basic, delegation, generate, notification, logging, secure and persistent. For the corresponding implementation of the weather service use this naming convention: <feature><service_name>Impl.java

For example: SecureWeatherImpl.java, LoggingWeatherImpl.java etc.
Reuse the GUI interface the core package provides to present your weather service.

4.1 Implementation Details

1. Download, install, configure and verify the core of the Globus Toolkit.
2. See the details of programming and building a grid service in a fine tutorial offered at <http://www.casa-sotomayor.net/gt3-tutorial/>
3. Build and test all the sample services that are provided in the downloaded package.
4. Implement the basic weather service and deploy it.
5. Test it with a simple command line client.
6. Modify the GUI to incorporate your basic weather service by adding GUI buttons and boxes. Test it.
7. Repeat the steps 3-5 for other improved versions of the weather service.
8. Prepare the GAR (grid archive file) for job submission into real grid. We will provide you the details later.

5. Report and Submission

See Project 1 for the Report that you need to prepare and for the submission details.

Due Date: December 8, 2003 by midnight. No extension will be given.