

Venice – A Lightweight Service Grid

Markus Hillenbrand, Joachim Götze, Paul Müller
University of Kaiserslautern, Germany
Department of Computer Science
{hillenbr, j_goetze, pmueller}@informatik.uni-kl.de

1. Introduction

What is the Grid? This question has been asked by many people and has been answered by many more. In 1969 Leonard Kleinrock imagined “the spread of *computer utilities*, which, like present electric and telephone utilities, will service individual homes and offices across the country” [1]. Ian Foster and Carl Kesselmann in 1998 tried to define a Grid that could help implement such computer utilities [2]. After several iterations, Ian Foster finally published his three point checklist [1] that must be met by a system to be called a Grid.

Today, a distinction between *Computing Grids*, *Data Grids* and *Service Grids* can be made to distinguish between the many Grid systems available. While Computing Grid like UNICORE [3], gLite [4] and Globus Toolkit [5] as well as Data Grids like the EU DataGrid Project [6] are deployed on a larger scale and used in world wide research projects, pure Service Grids offering arbitrary services to end-users are still in their infancy. Globus Toolkit 4 could be regarded as an early version of a Service Grid, but its legacy makes it neither lightweight nor easy to deploy or maintain.

A lightweight Service Grid should be easy to deploy, easy to maintain and easy to use. It should be built on open standards and be applicable to different usage scenarios – not only high performance computing or high throughput data processing. In the following, the *Venice Service Grid* will be outlined. Its roots are in a telephony project [7] that successfully made available SIP, H.323 and several supplementary services as Web services. The basic infrastructure is currently being reorganized and completed to create an open, dependable and flexible framework for deploying lightweight services to end-users.

2. Architecture

The services offered by the Venice Service Grid are implemented as Web services and can be divided into three distinct categories. *Management Services* are services necessary for the common usage and maintenance of users, resources, services, and other entities in the Grid. The *Information Services* are responsible for collecting, managing and distributing information needed by users or services. The *Application Services* finally are horizontal services that can be used by any application built on top of the framework and vertical services that are mostly valuable in a specific application domain. Common to all those ser-

vices is that they are equipped with at least one graphical user interface that can be started using the Software Deployment Service (see below). Additionally, every service can either be used separately or in a predefined workflow. In the following, the services already developed and currently in work will be shortly introduced while the services planned for the future will be outlined in the Future Work section.

2.1 Management Services

SSO (Single Sign-on Service). In order to access all services of a provider, a token-based single sign-on strategy is the most promising approach for a lightweight Service Grid. A user has to authenticate once and will then receive a token allowing him to proof his identity in any further communication by providing this token. If the user is accessing a service, this token will allow the user not only to proof his identity, but also to proof his right to access this service. As there may be several services involved in a usage scenario, it is very important to use an authentication and authorization strategy based on single sign-on. This strategy allows the user to be authenticated to any service inside the authentication domain without the need to enter his credentials a second time. A more detailed description of this token-based single sign-on solution developed for the Venice Service Grid can be found in [8].

MAB (Metering, Accounting, and Billing Service). It lies within the nature of a service-oriented architecture with a sophisticated authorization management to offer services that provide the same functionality with different quality of service for different users – according to their authorization data. And these services can be offered by different autonomous services providers. In the end, a flexible and open mechanism for metering and accounting service usage has to be provided as an infrastructural service. On top of that, a transparent and secure billing has to be guaranteed.

SDS (Software Deployment Service). In order to provide an easy-to-use application to the end-user, it is important to dispense the user from tasks like installing and/or updating software. A Software Deployment Service allows the user to use an up to date application without any further effort. Additionally, such a service enables a service provider not only to maintain a state of the art software infrastructure, but also to automatically and dynamically replicating certain services. This compensates for high load situations and establishes a replica management. As a result a Software Deployment Service reduces the admini-

stration effort on both sides the user and the service provider. A comprehensive overview of the Software Deployment Service already developed can be found in [9].

2.2 Information Services

DIS (Domain Information Service). This service is responsible for providing all necessary information for using services in a domain. Thus, the service is the starting point for any further service interaction and provides meta data about the single sign-on service, the information broker and the service domain itself.

WHOIS (Who Is Service). Using this service, it is possible to retrieve data about other users in any domain. The data available through this service has to be authorized by the affected user in order to respect the user's privacy. This service will be complemented by a presence service in the future.

IB (Information Broker). This service is a generic service responsible for brokering service meta data, user meta data and other arbitrary data sets. Its implementation is based on Peer-to-Peer technology in order to attain scalability and to overcome some shortcomings of UDDI and other central registries.

Service Information. Every Venice service has to implement a special interface providing information about the service itself (name, purpose, icon, authorization scheme, etc.). Only by implementing this interface, the service can be seamlessly integrated into the framework. It also addresses how to find a suitable GUI application for the service (e.g. via Java Web Start or a Web page).

2.3 (Horizontal) Application Services

Property Storage Service. This service securely stores tag/value pairs. As simple as it might sound, its functionality has a large potential, e.g. for storing user preferences, service properties or other arbitrary data. This service is massively used by other services within the framework in order to attain location transparency.

Time Keeper Service. This service can be used by other services or client programs to store measuring points consisting of a time stamp and a context. After measuring a sequence of data points, a time series analysis can be visualized and exported as image or text files.

Positioning Service. When it comes to locating physical resources (printers, scanners, laboratory equipment, etc.) in three dimensional space, this service provides a geometric and symbolic positioning scheme and a useful plugin mechanism for integrating several location sensing techniques (like GPS, RADAR, etc) into clients. This allows for finding the nearest suitable resource for a specific task.

2.4 (Vertical) Application Services

In a first application scenario, the framework has been used as an infrastructure for a Web services based Voice over IP application. Several services dealing with Voice over IP functionality have been developed and brought into

operation [7]. This work was funded by Siemens AG, Munich. In a current research project called DASIS [10], the framework is used to virtualize the devices, tools and workflows of an ordinary office. Here, a special service will be responsible for locating physical resources like printers and scanners and associate it with another service or workflow. This project is a joint venture with the Fraunhofer Institute for Experimental Software Engineering (IESE).

3. Future Work

Several application services will be developed in the future. A Publish/Subscribe service will enable applications and services to subscribe for topics of interest and receive notifications when an event of interest appears. In order to strengthen user interaction inside Venice and to demonstrate the possibilities of the Information Broker, a presence service [11] will be developed.

The integration of the Web Services Resource Framework (WSRF) will allow for separating resources from services and – on the long run – allow for compatibility with other WSRF-enabled products, like Globus Toolkit. A connection to Globus, Unicore, gLite and other Computing and Data Grids will make available computing power and large data stores to all services and users inside Venice. The goal is to treat e.g. a Globus node like a resource in Venice and thus enabling all Venice services to acquire additional computing power or storage.

References

- [1] Foster, I., *What is the Grid? A Three Point Checklist*, available online <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf> (2002)
- [2] Foster, I., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann (1998)
- [3] Unicore: <http://unicore.sourceforge.net/>
- [4] gLite: <http://glite.web.cern.ch/glite/>
- [5] Globus Toolkit: <http://www.globus.org>
- [6] EU DataGrid: <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [7] Hillenbrand M., Götze, J., Müller, P., *Voice over IP - Considerations for a Next Generation Architecture*, Proceedings of the 31st Euromicro Conference (2005)
- [8] Hillenbrand, M., Götze, J., Müller, J., Müller, P., *A Single Sign-On Framework for Web-Services-based Distributed Applications*, Proceedings of the 8th International Conference on Telecommunications ConTEL (2005)
- [9] Hillenbrand, M., Müller, P., Mihajloski, K., *A Software Deployment Service for Autonomous Computing Environments*, Proceedings of the International Conference on Intelligent Agents, Web Technology and Internet Commerce – IAWTIC (2004)
- [10] Project DASIS: <http://www.dasmod.de/>
- [11] Day, M., Rosenberg, J., Sugano, H., *RFC 2778: A Model for Presence and Instant Messaging*, (2000)