

Enhancing Distributed Systems through Web Services

Sergio Muscat

Department of Computer Science and AI,
University of Malta

1 Introduction

The concept of Web Services has, throughout the last few years, become one of the most discussed in the academic, as well as the business world. Many hail it as a revolutionary concept, while others look at it in a sceptic manner. The fact remains that the developers of the most diffused programming languages are giving great importance to integrating and supporting the creation and utilisation of Web Services.

The main advantage of Web Services is that they are based completely on XML, which gives them a very high degree of flexibility and, above all, platform independence. Systems written using one particular language, can transparently access exposed services on other systems written using different languages.

This particular feature of Web Services puts them in an ideal position to be utilised as the driving force behind distributed systems. The Internet is a jungle of computers, each with their particular features, operating system and hardware platform. Such a scenario makes it very difficult to create a system, distributed over various machines, which easily adapts to such an environment and communicates with other machines in this environment. Using Web Services, this problem is naturally solved.

This project aims at analysing the ways in which Web Services may be used to enhance distributed systems over the Internet framework. There are various issues which should be considered, including the software required and the security of data being transmitted.

2 Scenarios

This project will focus on two scenarios which can be easily applied to real-life situations. The final product will be a demonstration of each of these scenarios, applied to a particular business area.

2.1 Scenario 1: Web Services as Notification and Control Tools

In this simple scenario, Web Services are used as a remote control and notification tool for subscribers to the particular system. A small application located on the client's machine will poll the central system and retrieve any new notifications. The client will then be allowed to handle the notifications either through the client system, or via a web-based interface. Figure 1 illustrates the concept.

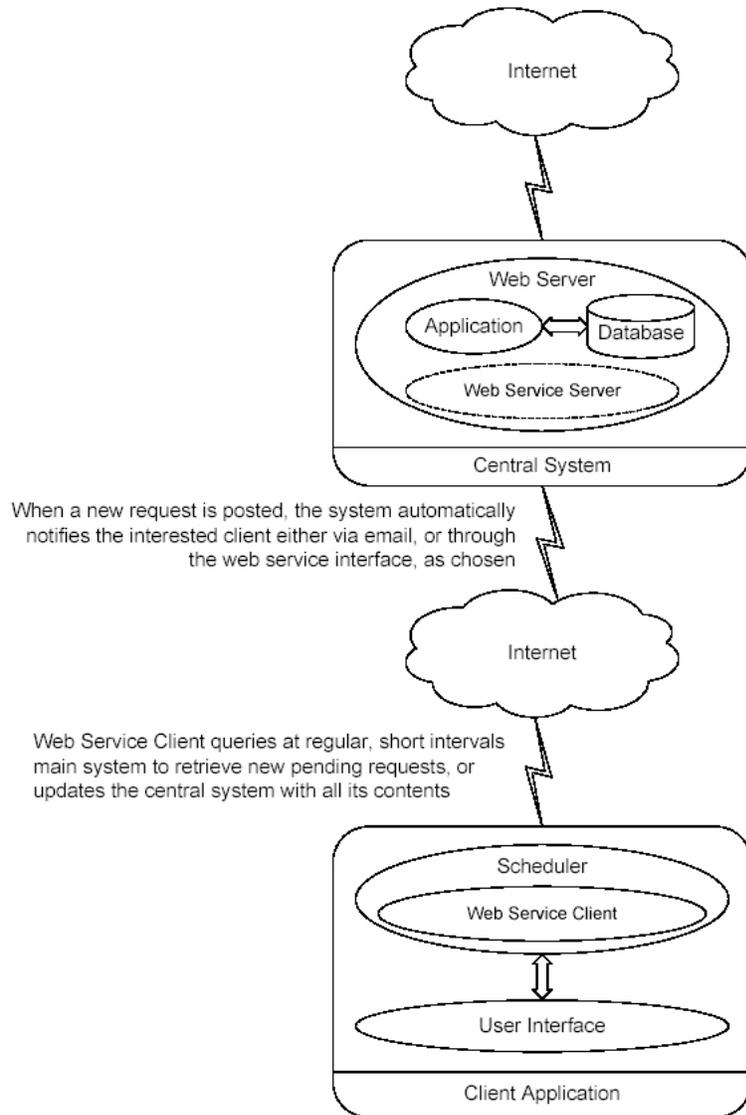


Fig. 1. Web Services as Notification and Control Tools

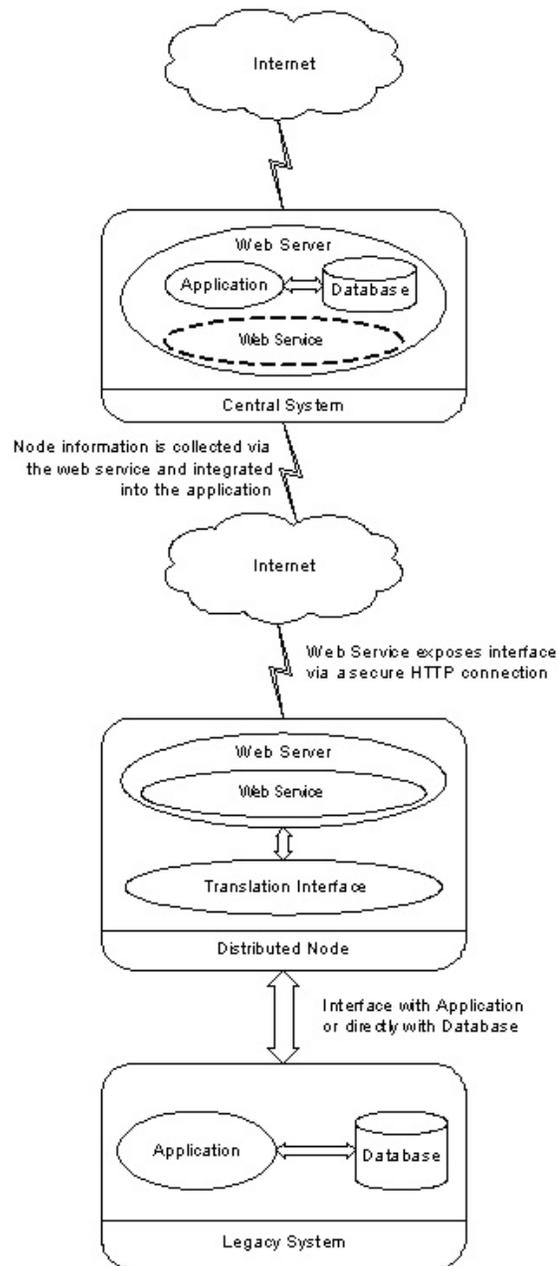


Fig. 2. Web Services as Integration Tools

2.2 Scenario 2: Web Services as Integration Tools

This more complex scenario puts Web Services as the main tool for remote integration and control. A central system is used to collect and disseminate information to and from various remote systems, integrating and displaying them as a single entity. This system allows third parties to utilise legacy systems for managing their information, while the distributed software node interfaces with such a system, extracting, translating and transmitting the information either automatically or on request from the central system. Web Services may be used in various degrees, depending on the requirements of the system, however, the main issue is whether the distributed node (as opposed to the central system) should either have the role of a client, or that of a server. The decision depends on whether such node should simply serve as a 'dumb' terminal which simply collects information and transmits it to the central system, or whether it should hold the information locally and only transmit the necessary details on request by the central system. The various aspects of this issue will be dealt with throughout this project, outlining the consequences, advantages and disadvantages entailed by each solution.

References

1. Securing and Managing XML Web Service Networks — Westbridge Technologies, 2002.
2. Introduction to Web Services — Systinet, 2002.
3. Web Services, the New Generation of Application Integration — Data Junction Corporation.
4. Web Services — SeeBeyond, 2002.
5. Web Services architecture overview, the next stage of evolution for e-business — IBM Web Services Architecture Team (karlgott@us.ibm.com)
<http://www-106.ibm.com/developerworks/web/library/w-ovr/>.
6. Building Asynchronous Applications and Web Services — Spiritsoft.