7. Implementation and Performance Evaluation

We implemented the proposed framework and validated its functioning through two case studies. The prototype implementation and these two case studies were then used to make an estimate of impact on performance for additional privacy protection offered by the framework. In this chapter, we present the implementation details, the performance figures obtained, and the analysis of the obtained performance figures.

7.1 COMPOSITION LANGUAGE USED

Several standard languages have been proposed in the literature for web services composition, such as BPEL4WS [9] and WSCI [10]. We used NuSOAP [59] for prototype implementation of our framework. NuSOAP is a powerful API developed for PHP platform. One of the key features of NuSOAP is the built-in WSDL support. The required libraries are contained in a file called nusoap.php.

7.2 PLATFORM AND ENVIRONMENT USED

As we used NuSOAP for prototype implementation of our framework and since NuSOAP is a powerful API developed for PHP platform, we implemented the framework on PHP platform with web services support for SOAP messaging and
WSDL documents. The experiments were conducted on Intel(R) PIV 1GHz machine, with separate machines for each service, communicating over a wired 100 Mbps switched network.

Appendix C contains the PHP process description of service functions of components and Appendix D contains the WSDL description of components services.

### 7.3 CASE STUDIES USED

We validated the framework's functioning through the two case studies discussed before. These are:

1. Online travel arrangement service and
2. Online health clinic service

### 7.4 PERFORMANCE EVALUATION

We then used the prototype implementation and the two case studies to make an estimate of impact on performance for additional privacy protection offered by the framework. For this, we compared the total time required for completing a service request both without using a CSP (customer making separate requests to individual service providers) and with using a CSP implemented with our framework. This was done for both the case studies – online travel arrangement service and online health clinic service. Figure 7.1 and Figure 7.2 respectively show the timings required to complete the services in the two cases for the two case studies.

Notice that the overhead involved (extra time required) in completing an online travel arrangement service request with the use of a CSP implemented with our framework is around 2 to 3 seconds. Similarly, overhead involved in completing an online health clinic service request with the use of a CSP implemented with our
framework is around 6 to 7 seconds. These overheads are reasonable and acceptable by customers because of the extra privacy protection that they enjoy due to the use of our framework-based CSP. The framework offers value-added service in terms of additional privacy protection to customer’s private information. Hence, tradeoff in performance of the order seen in our experiment for this additional security is worth it because it instills confidence in consumers for using web services. Such type of consumer confidence is key to the success of web services.

<table>
<thead>
<tr>
<th>Case study</th>
<th>Time taken without using a CSP (sec)</th>
<th>Time taken with using a CSP (sec)</th>
<th>Overhead in terms of percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online travel arrangement service</td>
<td>5.02</td>
<td>7.72</td>
<td>54%</td>
</tr>
</tbody>
</table>

**Figure 7.1.** Comparison of time required for completing an online travel arrangement service request without using a CSP and with using a CSP implemented with our framework.

<table>
<thead>
<tr>
<th>Case study</th>
<th>Time taken without using a CSP (sec)</th>
<th>Time taken with using a CSP (sec)</th>
<th>Overhead in terms of percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online health clinic service</td>
<td>11.25</td>
<td>18.02</td>
<td>60%</td>
</tr>
</tbody>
</table>

**Figure 7.2.** Comparison of time required for completing an online health clinic service request without using a CSP and with using a CSP implemented with our framework.