**CORE SECURITY PATTERNS**

**CHRIS STEEL • RAMESH NAGAPPAN • RAY LAI**

**Web Tier Patterns**

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Enforcer</td>
<td>Delegate handling of authentication details</td>
</tr>
<tr>
<td>Authorization Enforcer</td>
<td>Delegate handling of authorization details</td>
</tr>
<tr>
<td>Interceptor</td>
<td>Use interceptors to perform security validation of input data</td>
</tr>
<tr>
<td>Validator</td>
<td>Centralize handling of security tasks in a base action class</td>
</tr>
<tr>
<td>Secure Base Action</td>
<td>Log sensitive data securely</td>
</tr>
<tr>
<td>Secure Service Proxy</td>
<td>Provide security to external systems using a proxy</td>
</tr>
<tr>
<td>Secure Session Manager</td>
<td>Centralize security session information handling</td>
</tr>
<tr>
<td>Web Agent Interceptor</td>
<td>Use an interceptor to provide security to web applications</td>
</tr>
</tbody>
</table>

**Business Tier Patterns**

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Interceptor</td>
<td>Capture security related events</td>
</tr>
<tr>
<td>Container Managed Security</td>
<td>Let the container manage security declaratively</td>
</tr>
<tr>
<td>Dynamic Service Management</td>
<td>Instrument code dynamically as needed</td>
</tr>
<tr>
<td>Obfuscated Transfer Object</td>
<td>Protect data passed around in transfer objects</td>
</tr>
<tr>
<td>Policy Delegate</td>
<td>Delegate security tasks to a helper</td>
</tr>
<tr>
<td>Secure Service Facade</td>
<td>Secure business tier access at the entry point in a Session Facade</td>
</tr>
<tr>
<td>Secure Session Object</td>
<td>Pass session state securely</td>
</tr>
</tbody>
</table>

**Web Service Tier Patterns**

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Inspector</td>
<td>Check for and verify XML security</td>
</tr>
<tr>
<td>Msg. Interceptor Gateway</td>
<td>Message intermediary that handles security tasks</td>
</tr>
<tr>
<td>Secure Message Router</td>
<td>Handles message security between multiple parties</td>
</tr>
</tbody>
</table>

**Identity Tier Patterns**

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assertion Builder</td>
<td>Defines how an identity can be built</td>
</tr>
<tr>
<td>Credential</td>
<td>Defines a mechanism for encapsulating credentials in tokens</td>
</tr>
<tr>
<td>Tokenizer</td>
<td>Describes how to sync passwords for security service provisioning</td>
</tr>
<tr>
<td>Password Synchronizer</td>
<td>How to handle legacy system SSO</td>
</tr>
<tr>
<td>SSO Delegator</td>
<td></td>
</tr>
</tbody>
</table>

**WWW.CORESECURITYPATTERNS.COM**

Copyright ©2005 - Steel, Nagappan, Lai