Before it dispatches a call to a web resource, the container must associate with the call thread an AccessControlContext containing the principals of (only) the target component's runAs identity (as defined in Section 4.5, "Component runAs Identity).

For the special case where the empty string must be substituted for the "/" pattern in the permission evaluation, all target related processing (including servlet mapping, filter mapping, welcome file processing, and form based login processing) must be performed using the original pattern, "/".

4.1.3 Application Embedded Privilege Test

When a call is made from a web resource to `isUserInRole(String roleName)` the implementation of this method must obtain a WebRoleRefPermission object with name corresponding to the servlet-name of the calling web resource and with actions equal to the roleName used in the call. For the special case where the call to `isUserInRole` is made from a web resource that is not mapped to a Servlet (i.e. by a `servlet-mapping`), the name of the WebRoleRefPermission must be the empty string. In either case, the implementation of the isUserInRole method must then use one of the methods described in Section 4.8, "Checking the Caller for a Permission" to determine if the WebRoleRefPermission has been granted to the caller. If a SecurityException is thrown in the permission determination, it must be caught, and the result of the determination must be that the permission is not granted to the caller. If it is determined that the WebRoleRefPermission has been granted to the caller, isUserInRole must return true. Otherwise the return value must be false.

4.2 Provider Support for Servlet Policy Enforcement

In support of the policy enforcement done by servlet containers, providers must implement the policy decision functionality defined in the following subsections.

4.2.1 Servlet Policy Decision Semantics

A Policy provider must use the combined policy statements of the default policy context (as defined in Section 4.10, "Default Policy Context") and of the policy context identified by calling PolicyContext.getContextID to determine if they imply the permission being checked. If one or more excluded policy statements imply the checked permission, the evaluation may terminate and the checked permission must be determined not to be granted. Otherwise, if one or more