specifications), components are configured such that they are assigned the identity of their caller (such as it is) as their runAs identity. Alternatively, a Deployer may choose to assign an environment specific identity as a component's runAs identity. In this case, the container must establish the specified identity as the component's runAs identity independent of the identity of the component's caller.

When a Deployer configures an environment specific component identity based on a deployment descriptor specification that the component run with an identity mapped to a role, those responsible for defining the principal-to-role mapping must ensure that the specified identity is mapped to the role.

A container establishes a component's runAs identity by associating an AccessControlContext with the component's thread of execution. The container must ensure that the AccessControlContext includes a SubjectDomainCombiner; and the container must protect the AccessControlContext associated with a running component such that, by default, the component is not granted permissions sufficient to modify the AccessControlContext.

## 4.6 Setting the Policy Context

A policy context identifier is set on a thread by calling the setContextID method on the PolicyContext utility class. The value of a thread's policy context identifier is *null* until the setContextID method is called. Before invoking Policy to evaluate a transport guarantee or to perform a pre-dispatch decision, and before dispatching into a Servlet or EJB component, a container must ensure that the thread's policy context identifier identifies the policy context corresponding to the instance of the module or application for which the operation is being performed.

Containers must be granted the "setPolicy" SecurityPermission independent of policy context identifier (or in all policy contexts) as they need this permission to set the policy context identifier.

### 4.6.1 Policy Context Handlers

This specification requires that containers register policy context handlers with the PolicyContext utility class such that Policy providers can invoke these handlers to obtain additional context to apply in their access decisions. Policy context handlers are objects that implement the PolicyContextHandler interface. To satisfy the requirements of this specification, containers are required to provide and register with the PolicyContext class the policy context handlers described in the following subsections. All of the required context handlers must[3] return the value null when activated outside of the scope of a container's processing of a

component request. In this context, the scope of a container's processing of a component request begins when the container asks policy to perform the corresponding pre-dispatch access decision and ends either when the access decision returns a failed authorization or when the dispatched request returns from the component to the container.

Policy providers must not call methods on or modify the objects returned by the context handlers if these actions will cause the container to fail in its processing of the associated request.

Containers may delay the registration of required context handlers until the first call to `PolicyContext.getHandlerKeys`, or for a specific handler, until the required context handler is activated (assuming `getHandlerKeys` has not been called). When a required context handler for which registration has been delayed is invoked, the container may return null, and must complete the registration of the handler before returning.

A provider that is dependent on a handler, should force registration of the handler in advance of the provider's processing of a component request for which the handler is required. This can be accomplished by invoking the required handler during initialization of the provider.

#### 4.6.1.1 Container Subject Policy Context Handler

All EJB and Servlet containers must register a PolicyContextHandler whose getContext method returns a javax.security.auth.Subject object when invoked with the key "javax.security.auth.Subject.container". When this handler is activated as the result of a policy decision performed by a container before dispatch into a component, this handler must return a Subject containing the principals and credentials of the "caller" of the component. When activated from the scope of a dispatched call, this handler must return a Subject containing the principals and credentials corresponding to the identity established by the container prior to the activation of the handler. The identity established by the container will either be the component's runAs identity or the caller's identity (e.g. when an EJB component calls `isCallerInRole`). In all cases, if the identity of the corresponding Subject has not been established or authenticated, this handler must return the value null.

---

3. Whether or not this requirement applies to an additional PolicyContextHandler depends on the definition of the handler.