3. This JSR does not define or mandate a specific policy language to be used by providers. Each provider must define its own syntax, mechanisms, and administrative interfaces for granting permissions to principals.

4. The JSR does not require that providers support a policy syntax for granting to principals roles as collections of permissions.

5. Although the JSR is focused on defining permissions and policy for use by Java EE containers, we make no restrictions on the use of this information by other containers or applications, or on support by containers or providers of other permissions or policy.

6. It is not the intent of this JSR to extend or modify the Java EE authorization model to be equivalent to standard RBAC models for access control.

1.5 Running Without a SecurityManager

The following list defines changes to this contract that apply to containers running without a Java SE SecurityManager.

1. The restrictions defined in Section 3.3, “Permission to Configure Policy" need not be enforced. Also, the containers of the application server must not be denied permission to perform any operation that would have been permitted in the presence of a SecurityManager.

2. Such containers are not required (before dispatching a call) to associate an AccessControlContext with the call thread (as otherwise required by Section 4.1.3, “Pre-dispatch Decision" and Section 4.3.1, “EJB Pre-dispatch Decision").

3. When performing the operations defined in Section 4.7, “Checking AccessControlContext Independent Grants" and in Section 4.8, “Checking the Caller for a Permission", such containers must not employ the SecurityManager.checkPermission techniques defined in these sections.

4. When using the AccessController.checkPermission technique of Section 4.8, “Checking the Caller for a Permission", the calling container must ensure that the principals of the caller are contained in the AccessControlContext associated with the thread on which the call to checkPermission is made.