

# Policy Configuration Subcontract

The Policy Configuration Subcontract defines the interactions between container deployment tools and providers to support the translation of declarative Java EE authorization policy into policy statements within a Java SE Policy provider.

This subcontract also applies to the translation of authorization policy annotations that have an equivalent representation in Java EE deployment descriptor policy constructs (i.e., `security-constraint`, `method-permission`, `security-role-ref`, and `exclude-list` elements).

## 3.1 What a Java EE Platform's Deployment Tools Must Do

The `getPolicyConfigurationFactory` method must be used in every JRE to which the components of the application or module are being deployed to find or instantiate `PolicyConfigurationFactory` objects.

```
PolicyConfigurationFactory pcf =
    PolicyConfigurationFactory.getPolicyConfigurationFactory();
```

The `getPolicyConfiguration` method of the factories must be used to find or instantiate `PolicyConfiguration` objects corresponding to the application or modules being deployed.

```
String petContextID = "acme-pet-server /petstore";

PolicyConfiguration petPC =
    pcf.getPolicyConfiguration(petContextID,true);
```

The declarative authorization policy statements derived from the application or module deployment descriptor(s) must be translated to create instances of the corresponding `javax.security.jacc` Permission classes.

```
WebResourcePermission webPerm =
    new WebResourcePermission("/elephant", "GET");
```

Methods of the `PolicyConfiguration` interface must be used with the permissions resulting from the translation to create policy statements within the `PolicyConfiguration` objects.

```
petPC.addToRole("customer", webPerm);
```

The `PolicyConfiguration` objects must be linked such that the same principal-to-role mapping will be applied to all the modules of the application.

```
petPC.linkConfiguration(petFoodPC);
```

The `PolicyConfiguration` objects must be placed in `Service` such that they will be assimilated into the `Policy` providers used by the containers to which the application has been deployed.

```
petPC.commit();
```

Independent of this specification, J2EE deployment tools must translate and complete the declarative policy statements appearing in deployment descriptors into a form suitable for securing applications on the platform. On versions of the Java EE platform that require support for authorization policy annotations, the deployment tools must combine policy annotations in Java code with policy statements appearing in deployment descriptors to yield complete representations of authorization policy suitable for securing applications on the platform. The rules for combining authorization policy annotations with declarative policy statements are described in the versions of the EJB, Servlet, and Java EE platform specifications that require support for the annotations. Independent of whether annotations factor in the translation, the resulting policy statements may differ in form from the policy statements appearing in the deployment descriptors. The policy translation defined by this subcontract is described assuming that the policy statement form used by a platform is identical to that used to express policy in the deployment descriptors. Where this is not the case, the output of the translation must be equivalent to the translation that would occur if policy was completely specified in the deployment descriptors and the translation had proceeded directly from the deployment descriptors to the Java SE policy forms defined by this subcontract. Two translations are equivalent if they produce corresponding collections of unchecked, excluded, and role permissions, and all of the