|  |  |
| --- | --- |
| **Statefullness** | |
| Stateless | Stateful: Object pools and/or |
| **Typing** | |
| Weakly typed in 1.4 or  Generics allow strong typing in 1.5+ | Strongly typed |
| **Integration with code** | |
| Configurable in properties files or  hard coded into code | Hard coded into code |
| **Exception Handling** | |
| Centralized uncaught exception handling forces exceptions to fail closed. | Developer is responsible for exception handling |
| **What a call looks like in the code (Manual Integration)** | |
| AccessController.isAuthorized("SpecialPolicy");  AccessController.isAuthorized("SpecialPolicy1", data); | AccessController.isAuthorized("DELETE\_FILE", new DeletePolicy("filename")); |
| **What a developer implements** | |
| public class SpecialPolicy1 extends ESAPIPolicy  {  public boolean evaluate(Object o)  {  //special rules go here  return t/f;  }  } | public class DeletePolicy extends ESAPIPolicyCollection  {  protected String filename;  public DeletePolicy(String filename) {  this.filename = filename;  }    public boolean evaluate (String data)  {  //filename validation rules  //return true or false.  }  } |
| **Integration** | |
| **Manual Integration Option**  AccessController.isAuthorized("SpecialPolicy");  AccessController.isAuthorized("SpecialPolicy1", data);  **Integration Option**  @policy KeyPolicyFile  @policy MethodToKeyPolicyFile  @policy AspectJ <cut stuff>  @Policy JSR115  @key StandardSimplePolcy  public class StandardSimplePolcy{  public boolean evaluate(Object<T> data) throws Exception {  //cast data  //do file look up stuff  }  }  **Properties file Integration**  policy.properties  SpecialPolicy=com.foo.model.policy.SpecialPolicy  SpecialPolicy1=com.foo.model.policy. StandardSimplePolcy  com.mypackage.myclass.mymethod=SpecialPolicy | **Manual integration**  AccessController.isAuthorized("DELETE\_FILE", new DeletePolicy("filename")); |