

Chapter 1

Assignment

In this chapter we will present the class that has been assigned to us. First, the lines of code to be analysed are presented without any comment. Afterwards, a brief description of the class role and function is presented. Obviously this is made by us and it is based only on the code and on the documentation provided by the authors.

1.1 Assigned class

The class assigned to us is called “**SecurityMechanismSelector**” and is located in the following path relative to the root of Glassfish project:

appserver/security/ejb.security/src/main/java/com/sun/enterprise/iiop/security/

The package which the class belongs to is: *com.sun.enterprise.iiop.security*

To present more clearly the class we include the first lines of code containing the documentation of the class and its declaration:

```
108  /**
109   * This class is responsible for making various decisions for selecting
110   * security information to be sent in the IIOP message based on target
111   * configuration and client policies.
112   * Note: This class can be called concurrently by multiple client threads.
113   * However, none of its methods need to be synchronized because the methods
114   * either do not modify state or are idempotent.
115   *
```

```

116  * @author Nithya Subramanian
117
118  */
119
120  @Service
121  @Singleton
122  public final class SecurityMechanismSelector implements PostConstruct {

```

Listing 1.1: Class documentation and declaration

1.1.1 Functional role

As the javadoc states, this class is responsible for selecting the security mechanism based on target configuration and client policies.

TO BE CONTINUED...

1.2 Assigned methods

The methods assigned to us are 4, that are presented below, with the entire code and with our description of their functional role.

1.2.1 First method

```

963  private Subject getSubjectFromSecurityCurrent()
964      throws SecurityMechanismException {
965      com.sun.enterprise.security.SecurityContext sc = null;
966      sc = com.sun.enterprise.security.SecurityContext.getCurrent();
967      if(sc == null) {
968          if(_logger.isLoggable(Level.FINE)) {
969              _logger.log(Level.FINE, "_SETTING_GUEST_---");
970          }
971          sc = com.sun.enterprise.security.SecurityContext.init();
972      }
973      if(sc == null) {
974          throw new SecurityMechanismException("Could_not_find_" +
975              "_security_information");
976      }
977      Subject s = sc.getSubject();
978      if(s == null) {
979          throw new SecurityMechanismException("Could_not_find_" +
980              "_subject_information_in_the_security_context.");

```

```
981     }
982     if (_logger.isLoggable(Level.FINE)) {
983         _logger.log(Level.FINE, "Subject_in_security_current:" + s);
984     }
985     return s;
986 }
```

Listing 1.2: First assigned method