Federated Management Architecture Specification 1.0 Changelog
July 26, 2000

The following changes are PROPOSED:

- Add ALWAYS_ACTIVE modifier. Some services must be reactivated and remain active if they are a source of activity, such as explicit lease renewal. Section 4.14.5.
- Proxy is in the wrong package. It should be in the package javax.fma.common, not javax.fma.client.. Section 4.16.1.
- BaseServiceInfo and StationAddress need clone methods since the classes are mutable (FMA10-0093). Section 4.7.3 and Chapter 3. Neither should have toString() as it is defined appropriately by AbstractEntry. All null fields and noarg ctor are ordained to be consistent with other Jini entries.
- ConcurrentControllerException should be final. Section 9.4.
- ConcurrentThreadException should be final. Section 8.2.
- ConcurrentTransactionException should be final. Section 7.2.
- Remove unused public constructor for LogicalThreadID. Section 4.6.2.
- Make StationSecurityException final. Section 5.7.3.
- Add newPermissionCollection to AccessPermission so that implies() will work. Section 5.7.3.
- Remove redundant private constructor and add newPermissionCollection() so that implies will work. Section 4.8.2.
- Make Controller.Lock fields private. Section 16.3.
- add subscribeResponsibleAfter() to event service interface. Section 18.3.
- Add constructor to SerializationFailureException. Section 17.1.1.
- Add IllegalArgumentException to LogService.post() and search(). Section 17.1.2.
- Specify that implementations of Search shall throw an UnsupportedOperationException if the remove() operation is invoked . Section 17.1.3.2.
- Remove javax.fma.common.PersistenceStream as the marker interface of streams used for persistence. Section 10.2.2.
- Remove ContextMap field from javax.fma.common.Context. Add EmptyContext inner class to Context. Section 4.6.6.
- Change javax.fma.common.StationSecurityException to extend StationException rather than SecurityException. Section 5.7.3.
- The javadoc for the class LogMessage reads:
  ```
  ...
  * and category parameters are cloned.  Throwables are
  * not cloned under the assumption that all throwables
  * are imutable. If a particular throwable is not immutable,
  * a client’s attempt to log the message (via a
  * LogService.log() call) will throw a
  * LogMessage.SerializationFailureException
  * when the throwable cannot be serialized.)
  */
  ```

Since log() is specifically required NOT to throw any exceptions, this should read that the SerializationFailureException will REPLACE the unserializable throwable in the log message. Additionally, the constructor to LogMessage also requires an almost identical change.

Also, the javadoc to the inner class SerializationFailureException says:

```
/** Exception thrown indicating that a client has attempted
 * to post a LogMessage that contains
 * a throwable which cannot be serialized.
 */
```

This must be changed to reflect that this eexception is never thrown, only used as a replacement.

- Make SecurityService to be a Base Service. Security is defined to be an integral part of any FMA implementation. Functioning of the security framework is dependant on guaranteed existence of the SecurityService, which constitutes a definition of a Base Service. Thus, the SecurityService is suggested to be made a Base Service. If the SecurityService is not a BaseService, then it either needs to be pulled out of the specification, in which case the specification will be left underspecified on the authentication side or special provisions will have to be put in the specification outlining the instantiation of the SecurityService other than via dynamic deployment model, as dynamic deployment is not acceptable for the reasons of start up-dependencies in secure environment.

  Once the SecurityService is made a BaseService, the corresponding artifacts, such addition of getSecurityService() in ServiceFinder should be added.

- Move ProperLoginModule into a javax package. The specification defines a proper JAAS login module. This login module is used on the client-side to authenticate in FMA. However, because of the way ProperLoginModule has been specified, an implementation dependence has been introduced. Making the already well-defined ProperLoginModule a specification class would eliminate the implementation dependency.
- Add a class to the specification to provide for a generic “well-known” RemoteEventListener stub class. This would allow clients that use the EventService as subscribers or the SchedulingService to not have to provide class servers to provide the stub class used for remote communication of the RemoteEventListener. The specification should also dictate that all event sources, regardless of their type, have this well-known class locally available to them.