

## 9 Standards, Best Practices and Patterns

### 9.1 Purpose

The Standards, Best Practices and Patterns section describes the Standards, Best Practices and Patterns that will be used in development of the Broker.

### 9.2 Standards/Best Practices Chart

Standard/Best Practice	Use in UI SIDES
JSR 109/1.1	Not Used – Deprecated to JSR 921
JSR 921	Used
Use of SAAJ	Not Used – Use of MTOM instead
OASIS WS-Security 1.0	Used
Web Services – Interoperability (WS-I)	Used
JSR-181	Not Used
JAXM/JSR-67	Not Used
JAXR	Not Used
JAX-RPC	Not Used
JAXB	Used
JSR-93	Not Used
JSR-101	Used
JSR-112	Not Used
JSR-127	Not Used
JSR-152	Used
JSR-153/JSR-220	Not Used
JSR-154	Used
JSR-168	Not Used

### 9.3 Patterns

#### 9.3.1 Model-View-Controller (MVC) Pattern

Model-View-Controller (MVC) - This is a classic design pattern often used by applications that need the ability to maintain multiple views of the same data. The MVC pattern hinges on a clean separation of objects into one of three categories — models for maintaining data, views for displaying all or a portion of the data, and controllers for handling events that affect the model or view(s).

Because of this separation, multiple views and controllers can interface with the same model. Even new types of views and controllers that never existed before can interface with a model without forcing a change in the model design.

### 9.3.2 Factory Pattern

The factory pattern, like other creational patterns, deals with the problem of creating objects (products) without specifying the exact class of object that will be created. The factory pattern handles this problem by defining a separate method for creating the objects, which subclasses can then override to specify the derived type of product that will be created. More generally, the term *factory method* is often used to refer to any method whose main purpose is creation of objects.

### 9.3.3 Data Transfer Object Pattern

Data Transfer Objects (DTO), formerly known as Value Objects or VO, are a design pattern used to transfer data between software application subsystems. DTOs are often used in conjunction with Data Access Objects to retrieve data from a database. DTOs do not have any behavior except for storage and retrieval of its own data.

### 9.3.4 Data Access Object Pattern

A Data Access Object (DAO) is an object that provides an abstract interface to some type of database or persistence mechanism, providing some specific operations without exposing details of the database. This isolation separates the concerns of what data accesses the application needs, in terms of domain-specific objects and data types (the public interface of the DAO), and how these needs can be satisfied with a specific DBMS, database schema, etc. (the implementation of the DAO).

### 9.3.5 HTTP Request/Response Pattern

HTTP Request/Response, also known as request-reply, is a message exchange pattern in which a requestor sends a request message to a replier system which receives and processes the request, ultimately returning a message in response. This is a simple, but powerful messaging pattern which allows two applications to have a two-way conversation with one another over a channel.

### 9.3.6 Singleton Pattern

The singleton pattern is a design pattern that is used to restrict instantiation of a class to one object. This is useful when exactly one object is needed to coordinate actions across the system. Sometimes it is generalized to systems that operate more efficiently when only one or a few objects exist. It is also considered an anti-pattern since it is often used as a euphemism for global variable.

### 9.3.7 Template Method Pattern

## **Version 9 – Design Baseline – 12-08-2009**

**Copyright © 2008 - 2009 National Association of State Workforce Agencies. All Rights Reserved.**

The Template Method pattern lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure. The algorithm is written as an abstract class and pieces can be redefined in classes that subclass it.