**Core J2EE Patterns**

A pattern is a proven solution to a problem in a context.

J2EE Patterns

* A pattern describes a proven solution to a recurring design problem

Why Use Patterns?

**They have been proven.** Patterns reflect the experience, knowledge and insights of developers who have successfully used these patterns in their own work.

**They are reusable.** Patterns provide a ready-made solution that can be adapted to different problems as necessary.

**Business Delegate**

* Hides the underlying implementation details of the business services such as lookup and access details of EJB architecture.
* reduces the coupling between presentation-tier clients and the system's business services
* The Business Delegate interacts with the Service Locator to locate business components
* The Business Delegate pattern uses Service Locator to gain access to the business service objects such as EJB objects
* separates the complexity of service location from the Business Delegate, leading to loose coupling and increased manageability

**Session Façade**

* The Session Facade is based on the Facade Design pattern.
* abstracts the underlying business object interactions
* **encapsulate the complexity of interactions between the business objects participating in a workflow**
* manages the business objects, and provides a uniform coarse-grained service access layer to clients
* manages the interactions between the business data and business service objects that participate in the workflow, and it encapsulates the business logic associated with the requirements
* **Thus, the session bean (representing the Session Facade) manages the relationships between business objects.**
* **The session bean also manages the life cycle of these participants by creating, locating (looking up), modifying, and deleting them as required by the workflow. In a complex application, the Session Facade may delegate this lifestyle management to a separate object. For example, to manage the lifestyle of participant session and entity beans, the Session Facade may delegate that work to a Service Locator object**
* The Session Facade is used by the Business Delegate when the client requests access to business services. The Business Delegate proxies or adapts the client request to a Session Facade that provides the requested service.
* The Session Facade can use the Service Locator pattern to reduce the code complexity and to exploit the benefits offered by the Service Locator

**Service Locator**

* abstract all JNDI usage and to hide the complexities of initial context creation, EJB home object lookup, and EJB object re-creation
* provide a single point of control, and improve performance by providing a **caching facility**