

# **Section 1: Overview**

## Table of Contents

<b>System Architecture Overview .....</b>	<b>3</b>
System Architecture.....	3
UI Tier/Layer .....	4
Business Tier/Layer .....	5
Application Servers .....	5
Hibernate.....	5
Math/Calculation Engine .....	6
Logical/Physical/Component Model .....	6

# System Architecture Overview

## System Architecture

The Oracle Insurance Policy Administration (OIPA) system interacts with several subsystems to provide a seamless and complete administration system. On the client side, the system is a lightweight browser-based subsystem. Both JavaScript and HTML orchestrate an expansive front-end. The JavaScript uses XML islands on the web page to provide speed by reducing the number of requests to the server. Style sheets provide the look and feel of the system and are maintained in central Cascading Style Sheet (CSS) files.

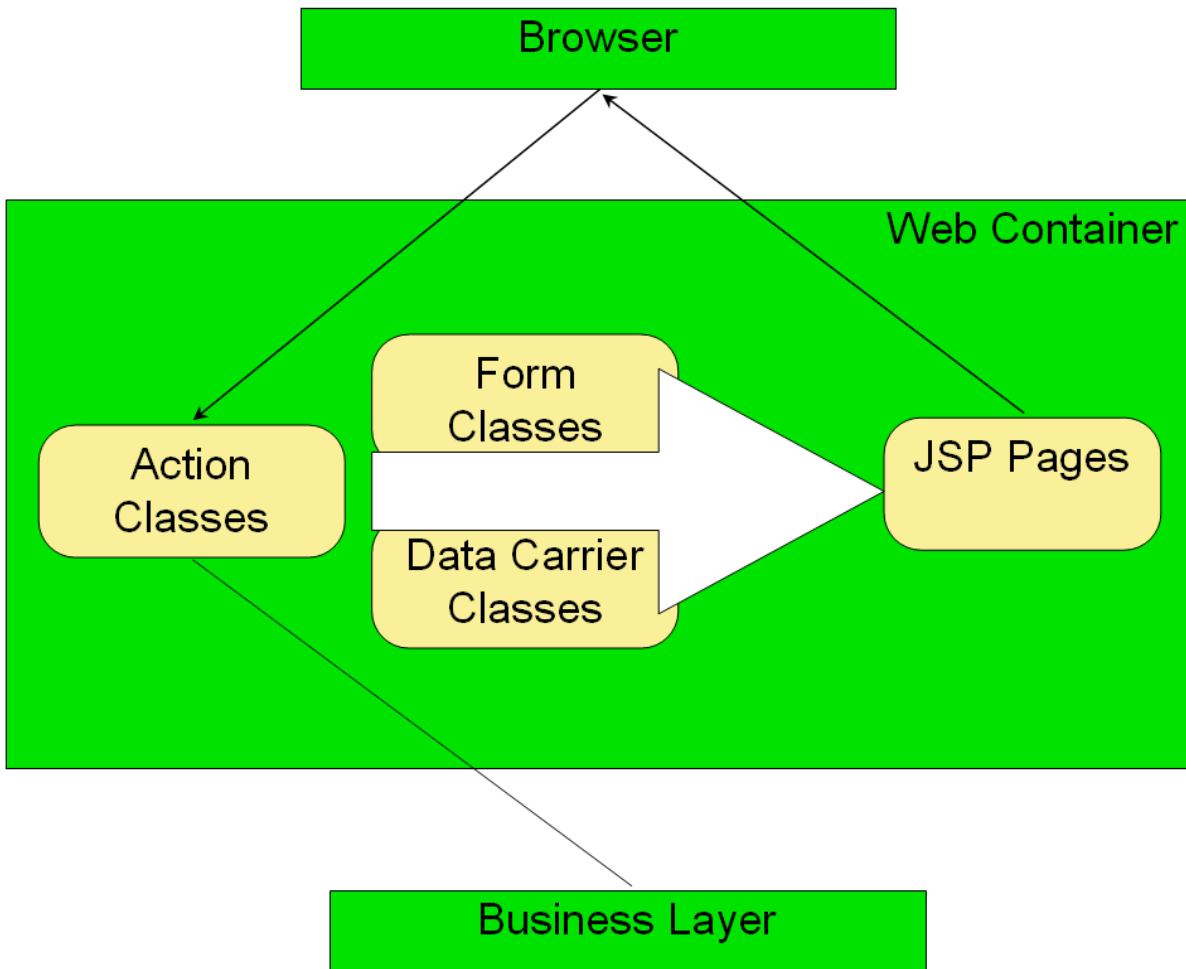
- Client-Side
- HTML (.ASP)
- JavaScript (.JSP/.JS)
- XML Islands (.JSP)
- Cascading Style Sheets (.CSS)

The server-side at the outset handles requests with the use of JSP pages. However, the JSP pages are simply a conduit for business processing and do NOT perform any business level functionality.

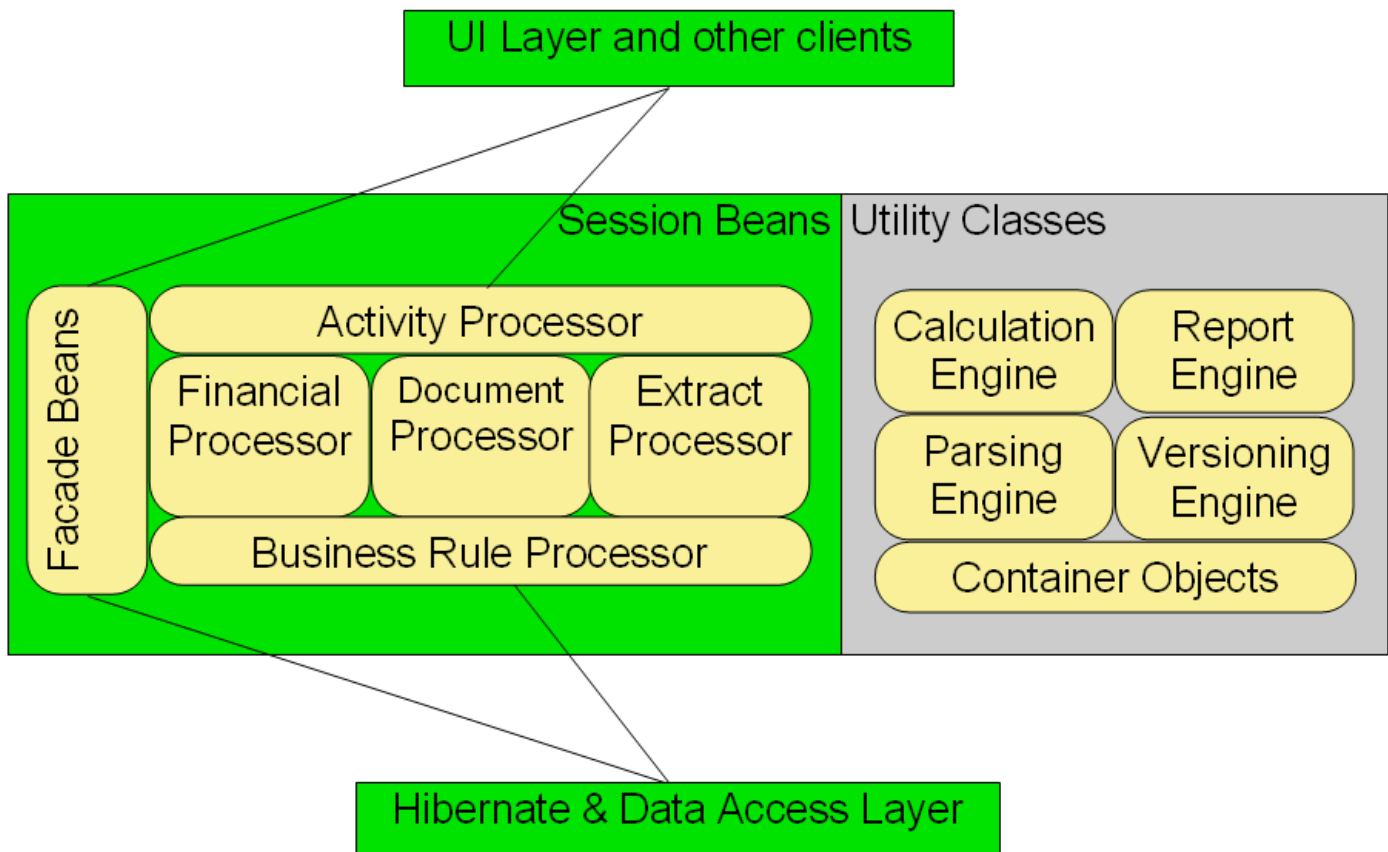
- Server-Side
- JSP-JavaScript (.JSP/.TXT)
- EJB (Enterprise Java Beans)
- Hibernate (Data Access and Persistence)
- Transact-SQL stored procedures
- Crystal Reports (.RPT)

The OIPA J2EE system is a scalable nTier web application in which the web server only pushes HTML, JavaScript, and XML to the client's browser. Using this programming model the OIPA architecture provides a model for developers to create a flexible and reusable application. By breaking up the system into tiers, OIPA developers only have to modify or add a specific layer, rather than rewrite the entire application, if they decide to change technologies or scale up.

## UI Tier/Layer



## Business Tier/Layer



## Application Servers

The fully scalable OIPA system supports WebLogic, WebSphere and JBoss. Within our J2EE environment, we create a single Enterprise Application Archive(EAR) file for all server products. Deployment descriptors are generated at build time for all of the application server products. OIPA does not use application server specific plug-ins. This allows for greater flexibility and more rapid development. All 3rd party .jars OIPA uses are bundled with our application.

## Hibernate

Hibernate is an Object to Relational mapping tool that allows XML mappings to be defined in connection between Java objects and database tables. This pluggable caching mechanism allows us to make use of a clusterable TreeCache. OIPA uses the cache to hold non-database information such as parsed XML. JTA integration ensures that objects are shared across the cluster.

## Math/Calculation Engine

Transaction Math is translated to an Intermediate script. These Intermediate scripts can be compiled to Java bytecode for speed. Compiled Intermediate script can be cached with the transaction much like the parsed XML. Translating to Intermediate script speeds the development of new features and new product capabilities are implemented with greater ease.

## Logical/Physical/Component Model

