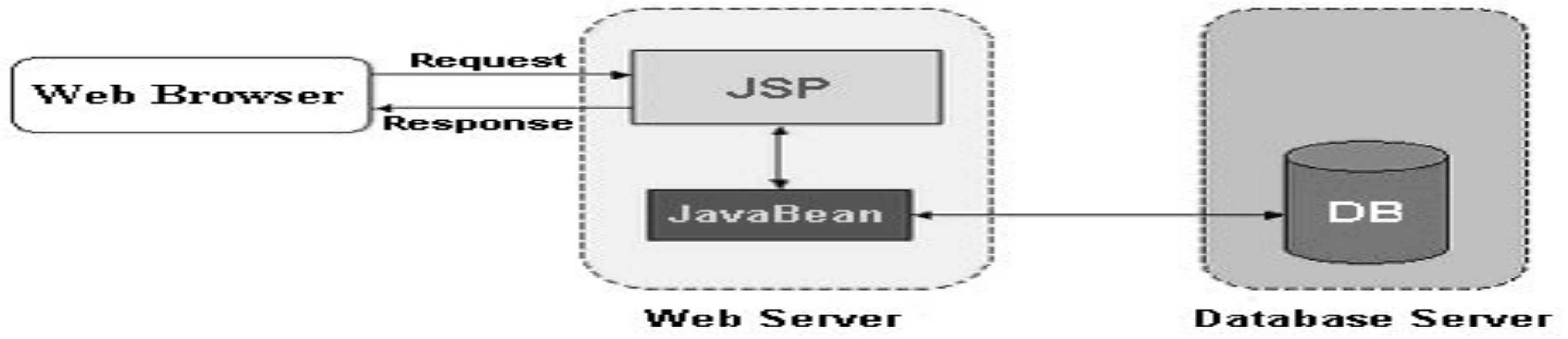
Integrating

Servlets, JSP, and JavaBeans

J2EE Architecture

- Model 1
  - JSP page is responsible for processing requests and sending back replies to clients
- Model 2
  - Integrates both servlets and JSP pages
  - Well known as the MVC (Model\View\Controller) paradigm

Original J2EE Architecture (Model 1)



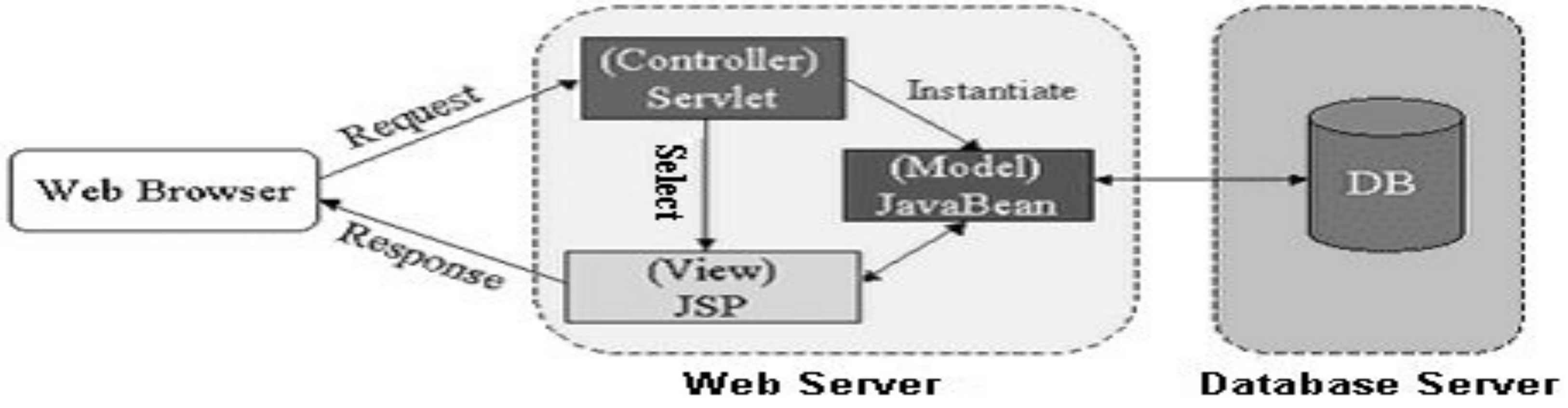
- Web browser directly accesses JSP pages
- Each JSP page processes its own inputs
- JSP includes logic to choose next page to return

× Hard to modify application flow

Model 1 : Main Problems

- A simple change can result in a cascade of changes in many different pages with unpredictable consequences
- Complexity grows fast as well, and what might originally look simple and straightforward can quickly turn into a big mess as you add more pieces

Model/View/Controller (Model 2)



- Introduces a controller between the browser and the JSP pages
- Controllers centralize logic for dispatching requests to next view
- Controllers handle view selection, which decouples JSP pages from one another
- View accesses the Model directly

Controlling Page Flow

- Controllers implement page flow
- JSPs do not refer to one another
- Controller dynamically chooses the "next" page
- Next view (page) to display depends on:
  - Results of any operation on the application model
  - Session state
  - URL and Request parameters
  - Form input

Implementing MVC

1. Define Bean to represent the data
2. Use a servlet to handle requests
   - Servlet reads request parameters, checks for missing and malformed data, etc.
3. Populate the beans
   - The servlets invokes business logic or data-access code to obtain the results. Results are placed in the bean that were defined in step 1.

Implementing MVC (Continued)

4. Store the beans in the request, session, or servlet context
   - The servlet calls setAttribute on the request, session or servlet context objects to store a reference to the beans that represent the results of the request
5. Forward the request to a JSP page
   - The servlet determines which JSP page is appropriate to the situation and uses the forward method of the RequestDispatcher to transfer control to that page.

Implementing MVC (Continued)

6. Extract the data from the beans
   - The JSP page accesses beans with jsp:useBean and a scope matching the location of step 4. The page then uses jsp:getProperty to output the bean properties.
   - The JSP page does not create or modify the bean; it merely extracts and displays data that the servlet created.

Applying MVC: Bank Account Balances

- Bank has Customers with property:
  - customer ID
  - name
  - balance
- Business Logic:
  - balance < 0   : delinquent customer page
  - balance < $10,000   : standard customer page
  - balance >= $10,000 : elite customer page

Applying MVC: Bank Account Balances

- Bean
  - BankCustomer
- Servlet that populates beans and forwards to appropriate JSP page
  - Read customer ID, calls data-access code to populate BankCustomer
  - Use current balance to decide on appropriate result page
- JSP pages to display results
  - Negative balance: warning page
  - Regular balance: standard page
  - High balance: page with advertisement added
  - Unknown customer ID: error page

Summary

- Use MVC (Model 2) approach when:
  - One submission will result in more than one basic look
  - Several pages have substantial common processing
- Architecture
  - A servlet answer the original request
  - Servlet does the real processing & stores results in beans
  - Servlets forwards to JSP page via forward method of RequestDispatcher
  - JSP page reads data from beans by means of jsp:useBean with appropriate scope