

# *JSF vs Struts*

## Cosa è JSF?

JSF, Java Server Faces, è il framework J2EE RI per lo sviluppo del web layer delle applicazioni web server-side. JSF in particolare fornisce strumenti per la costruzione dell'interfaccia web e per la gestione del flusso di navigazione dell'applicazione.

---

---

# *JSF vs Struts*

## **Perché JSF?**

Tutti conoscono la Java Enterprise Edition come una piattaforma matura, solida e affidabile per la costruzione di applicazioni enterprise, con un solido back-end applicativo. Una delle critiche più comuni rivolte alla piattaforma è sempre stata la mancanza di un modello di riferimento per lo sviluppo dell'interfaccia che consentisse una maggiore produttività in questa attività.

---

---

# *JSF vs Struts*

## Perché JSF?

Con JSF questa lacuna viene completamente colmata, in quanto vengono messi a disposizione gli strumenti per lo sviluppo rapido dell'interfaccia web di applicazioni J2EE.

JSF quindi si occupa di indirizzare lo sviluppo dell'interfaccia dell'applicazione e non del livello di business-logic, per il quale restano validi tutti i concetti conosciuti.

---

---

# *JSF vs Struts*

## **Elementi JSF**

- Un insieme di componenti di interfaccia predefiniti.
  - Un modello di sviluppo "event-driven".
  - Un modello a componenti.
- 
-

# *JSF vs Struts*

## Servizi JSF

- Conversioni di tipo.
  - Validazione.
  - Binding tra campi di interfaccia e attributi di JavaBean.
  - Gestione del Flusso di Navigazione.
  - Rendering dei componenti.
  - Gestione degli errori.
  - Internazionalizzazione.
- 
-

# *JSF vs Struts*

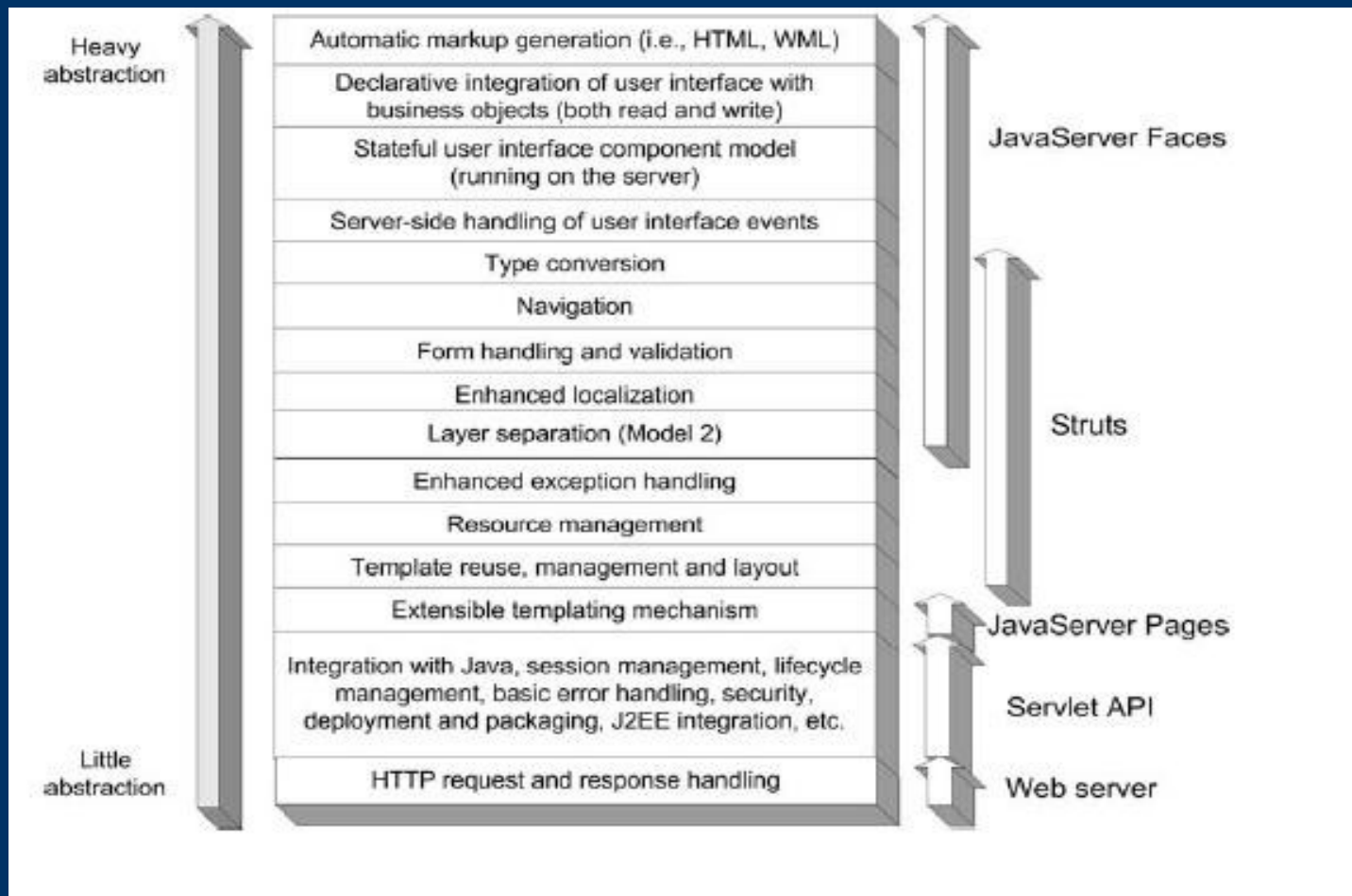
## **Perché scegliere JSF?**

JSF oltre a fornire servizi già presenti in altri framework introduce un livello di astrazione più elevato, analogamente a quanto fece Struts rispetto a Servlet e JSP.

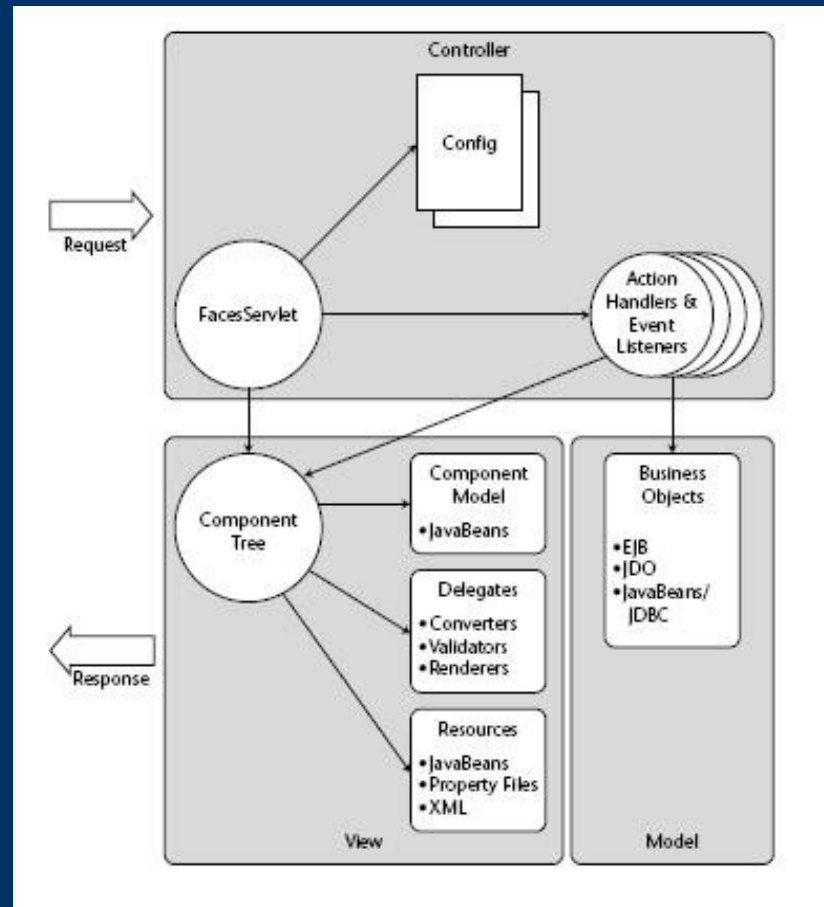
---

---

# JSF vs Struts



# JSF vs Struts



# *JSF VS Struts*

## **Perché scegliere JSF?**

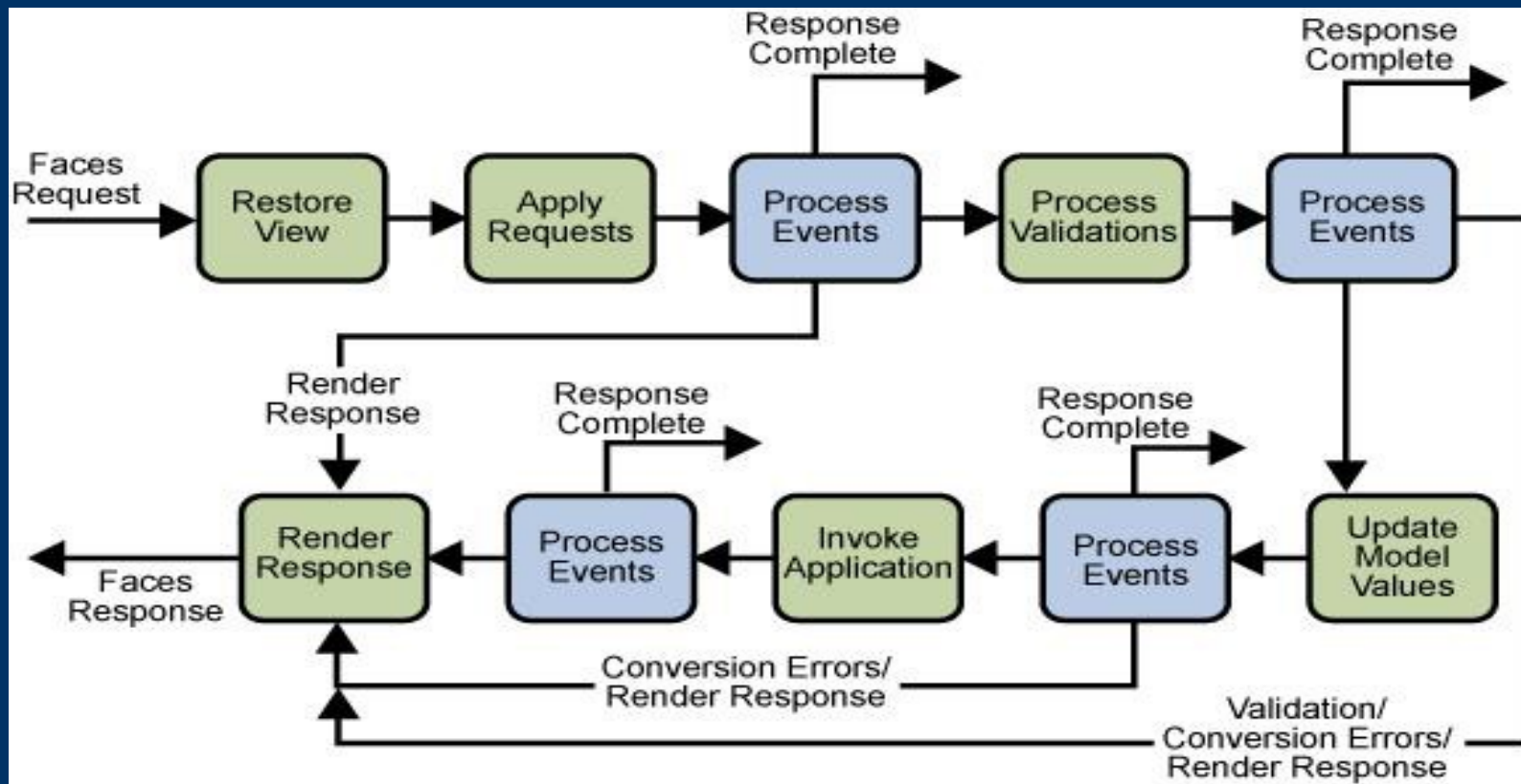
Struts ad esempio non fornisce un modello per la rappresentazione lato server degli elementi dell'interfaccia web e non ha un modello di sviluppo a eventi. JSF introduce questi concetti svincolando lo sviluppatore dalla gestione tramite codice di oggetti quali request e response, gestione che è invece ora totalmente a carico del framework.

---

---

# JSF vs Struts

## Ciclo di vita JSF?



# *JSF vs Struts*

## **Ciclo di vita JSF - Restore View**

In questa fase il framework acquisisce il component-tree della pagina richiesta se non è la prima richiesta alla stessa oppure ne crea uno nuovo nel caso di prima richiesta. È molto importante osservare che JSF mantiene lo stato dei componenti tra una richiesta e l'altra ad una vista per cui ad esempio mantiene i dati digitati in un form senza che lo sviluppatore se ne debba preoccupare.

---

---

# *JSF vs Struts*

## **Ciclo di vita JSF - Apply Requests**

In questa fase il framework memorizza i dati associati alla richiesta nei componenti associati ai vari elementi della pagina. Infatti ogni componente che può essere associato ad un input dell'utente ha un attributo `submitted value` che rappresenta il dato inviato dall'utente.

---

---

# *JSF vs Struts*

## **Ciclo di vita JSF - Process Validations**

In questa fase a tutti i componenti dell'albero che rappresenta la pagina in elaborazione viene richiesto di verificare che i valori presenti nella richiesta e memorizzati nella fase precedente sono validi quindi accettabili.

---

---

# *JSF vs Struts*

## **Ciclo di vita JSF - Update Model Values**

In questa fase viene associato il valore di un componente di una pagina, come ad esempio il valore di una text-box, ad un attributo di un JavaBean mediante una espressione simile a quelle tipiche di EL. Questi bean vengono detti backing-bean. I backing-bean sono dei JavaBean che consentono di interfacciare il livello di View dell'applicazione con quello di Model.

---

---

# *JSF vs Struts*

## **Ciclo di vita JSF - Invoke Application**

In questa fase quindi vengono eseguiti gli action methods dei backing-bean della vista e quindi viene eseguita la logica corrispondente all'azione effettuata dall'utente sull'interfaccia.

Quello che preme sottolineare ora e che è nella fase Invoke Application che tutti gli attributi dei backing-bean associati ai vari componenti della vista sono aggiornati e validi.

---

---

# *JSF vs Struts*

## **Ciclo di vita JSF - Render Response**

È l'ultima fase del ciclo , e si occupa di memorizzare lo stato della vista che si sta elaborando e di generare la risposta per il client.



# *JSF vs Struts*

**Componenti JSF**

MyFaces

Jboss RichFaces

ICEfaces

Oracle ADF Faces Component

