# Seam

the next generation framework!

Salemi  EDV-Beratung

12/17/2006

Based on

JBOSS online presentations

# JSF is good but not perfect

- Managed bean concept doesn't decouple the layers
  - the code is more coupled
- Code doesn't support multi window within the same user session
- Memory leaks:
  - Backing beans stay in the session until the user log out
  - This is the cause for many bugs
- Page and application are weakly defined
  - Navigation rules are ad hoc
  - No flow rules for long running workflows

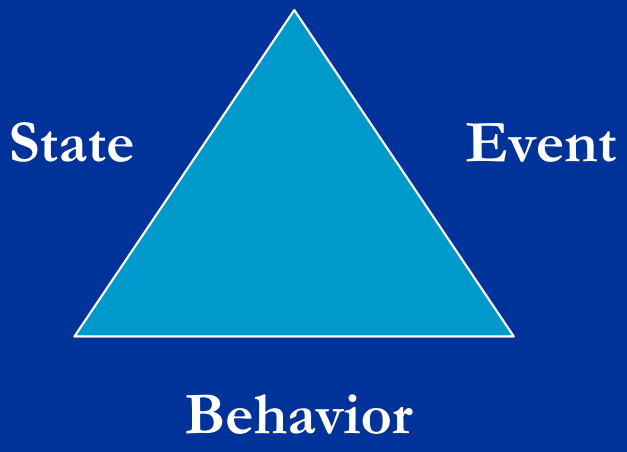
# Validation should be model-based

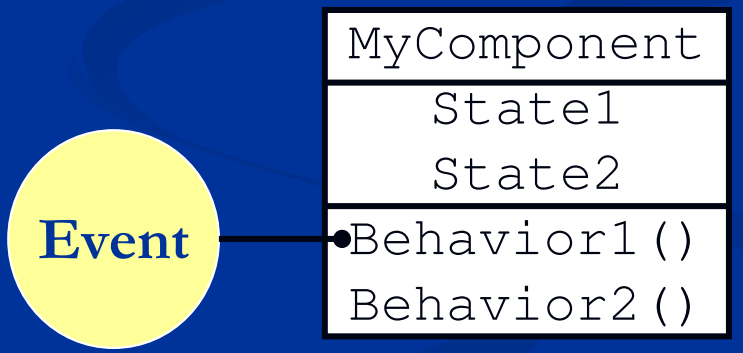
- Instead of validation in JSF use hibernate validation
  - before writing to DB the constraints are enforced
  - Seam can use them to validate user inputs
  - Hibernate can create the proper DDLs
- Express constraints on data model – e.g:

```
@Entity
pulic document{
        @id Long id;
        @length (max 100) String title;
        String content
        //getter and setters …
}
```

# Component in Seam

- Additional Dimension : Event Model
  - Send Messages to Components, Objects, ..
  - Allows to uncouple the components completely



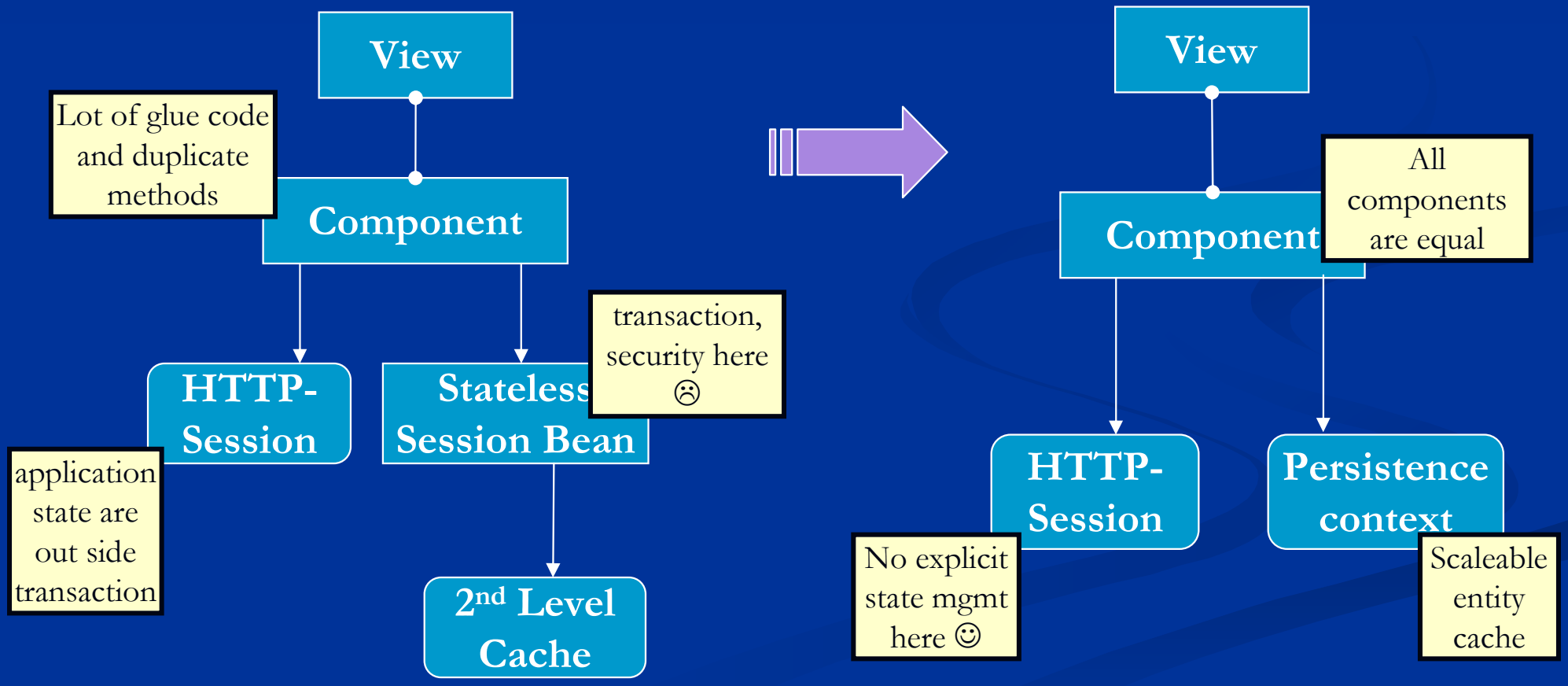



# What is a Seam Component

- Pretty much everything:
  - Stateless session bean
  - State full session bean
  - JPA Entity
  - Hibernate Entity
  - JavaBean

# Component Model improvement

- Move from disjoint to unified component model



# Seam Components example

```
@Stateful
@Name(“documenteditor”)
public class EditDocumentBean implement EditDocument{
@PersistenceContext
Private Enitymanager em;
@Begin
public Document get (Long id){
    …
}
@End @IfValid(outcome=REDISPLAY)
public String save (Document document){
    …
}
```

# Contextual Components

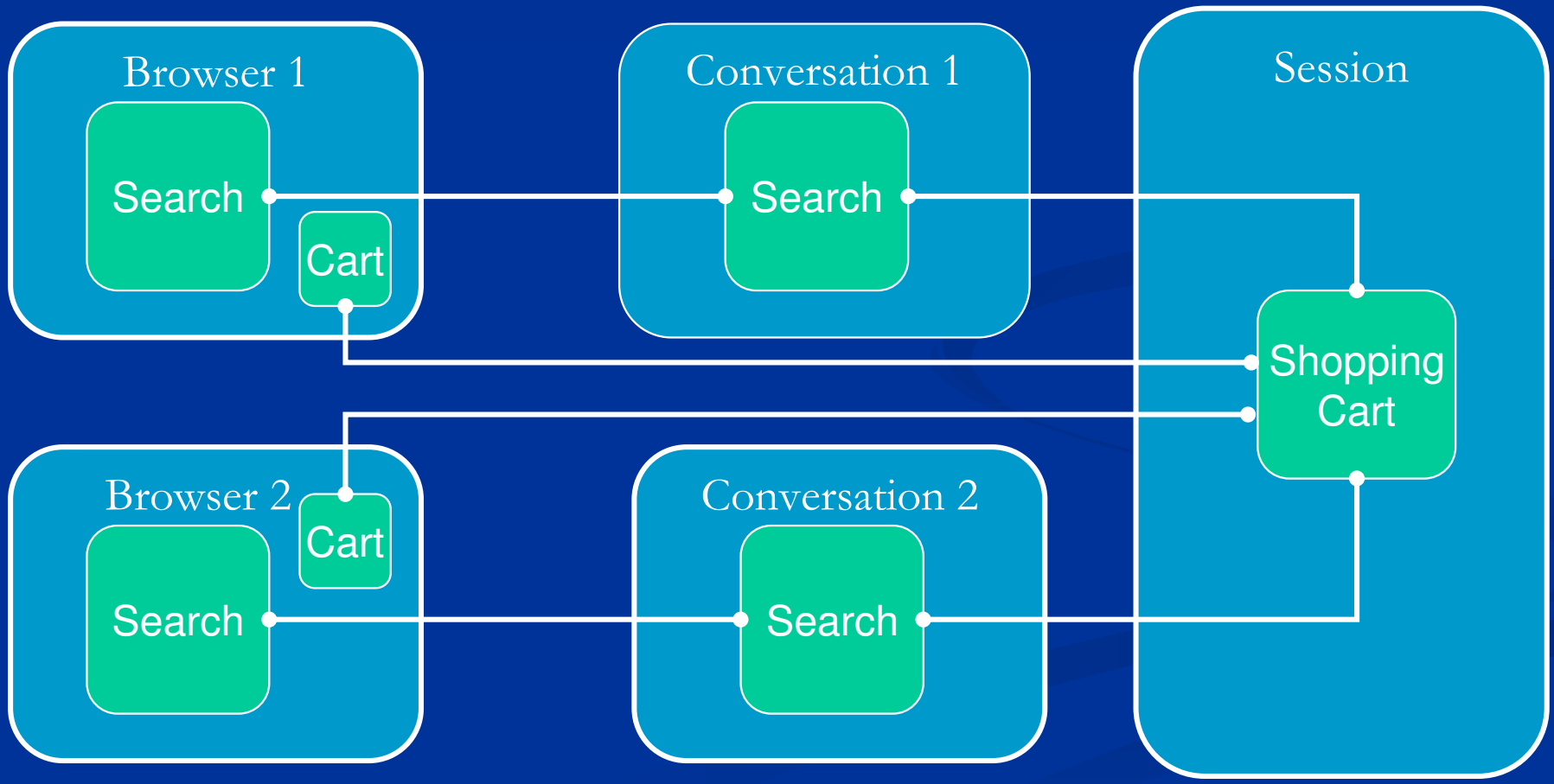
- Most problems are related to state management
  - Specification of Servlet context is not meaningful
  - EJB doesn't have strong state management model
  - There is a need of more powerful context concept
  - Logical context are meaningful for the application
- Mismatch between JSF and EJB 3 component model
  - Missing annotation in backing bean
  - EJB should be used as backing bean
- EJBs as backing bean make sense
  - Providing  event listener etc.
  - Entity beans provides data directly to form and accepts user input

# The Seam Context Model

- Seam defines a rich context model for state full components enabling container-management of application state. The context are:
  - Event
  - Conversation
  - Session
  - Process
  - Application
- Components are assigned to a scope using @scope annotation
- The highlighted "logical" context are demarcated by the application itself
  - @begin, @End

# Seam components are contextual

- Seams allows to have multiple search carts in different tabs - Seams hold everything consistent

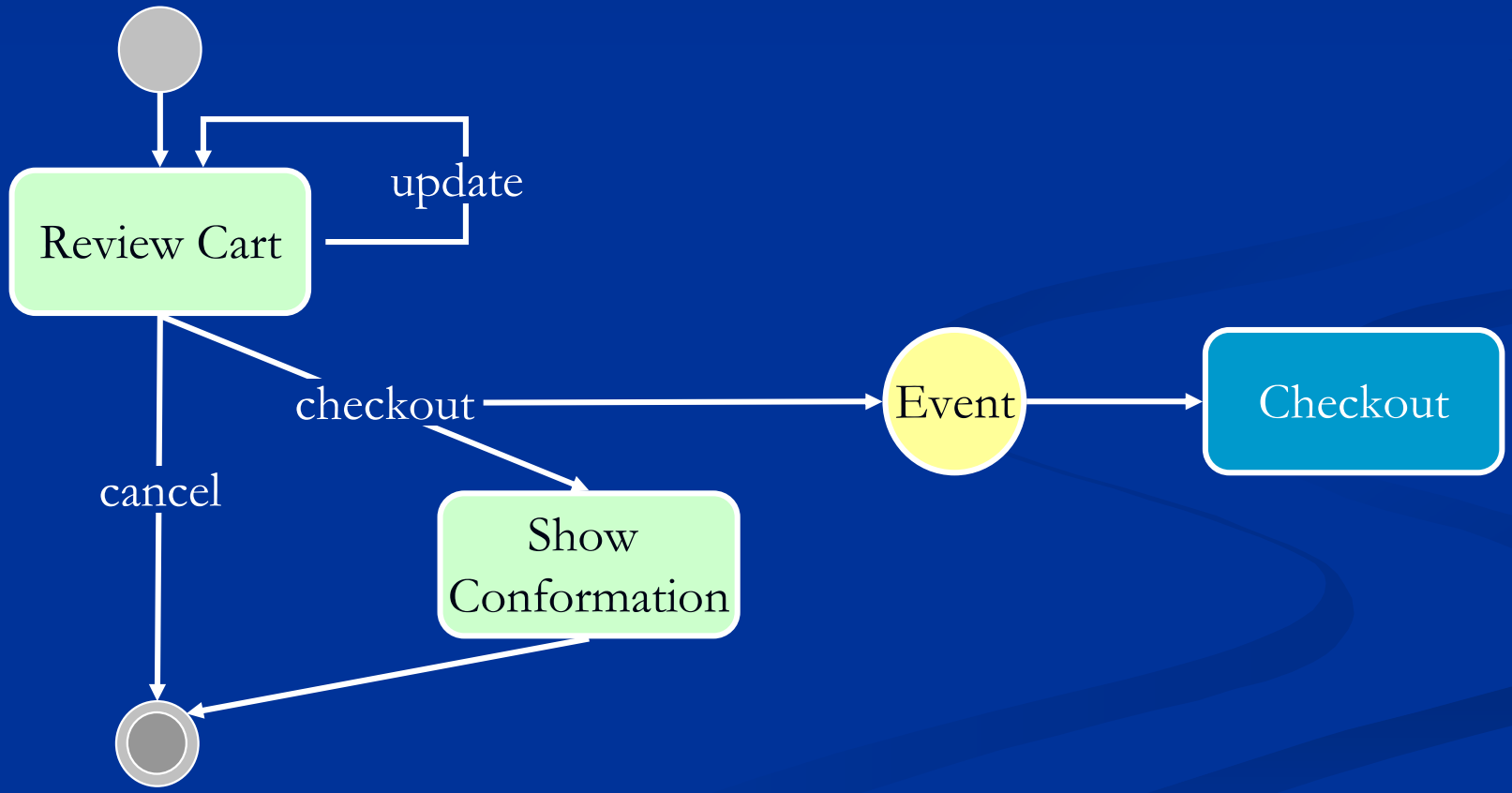


# Conversations

- Conversations are basic interactions the user has with application
- User interact with a page with the purpose to accomplish a logical unit of work (task)
- Seam understand the unit of conversation; it can mange the state of the component within this interaction (keep interactions separate) and drives the lifecycle according to the conversation
- State can be conversational
  - User open multiple windows to interact with the system
  - Don't need to implement code to distinguish between clicks (when two windows with same component for the same user)
  - we don't need to put information to session
  - no back button problem

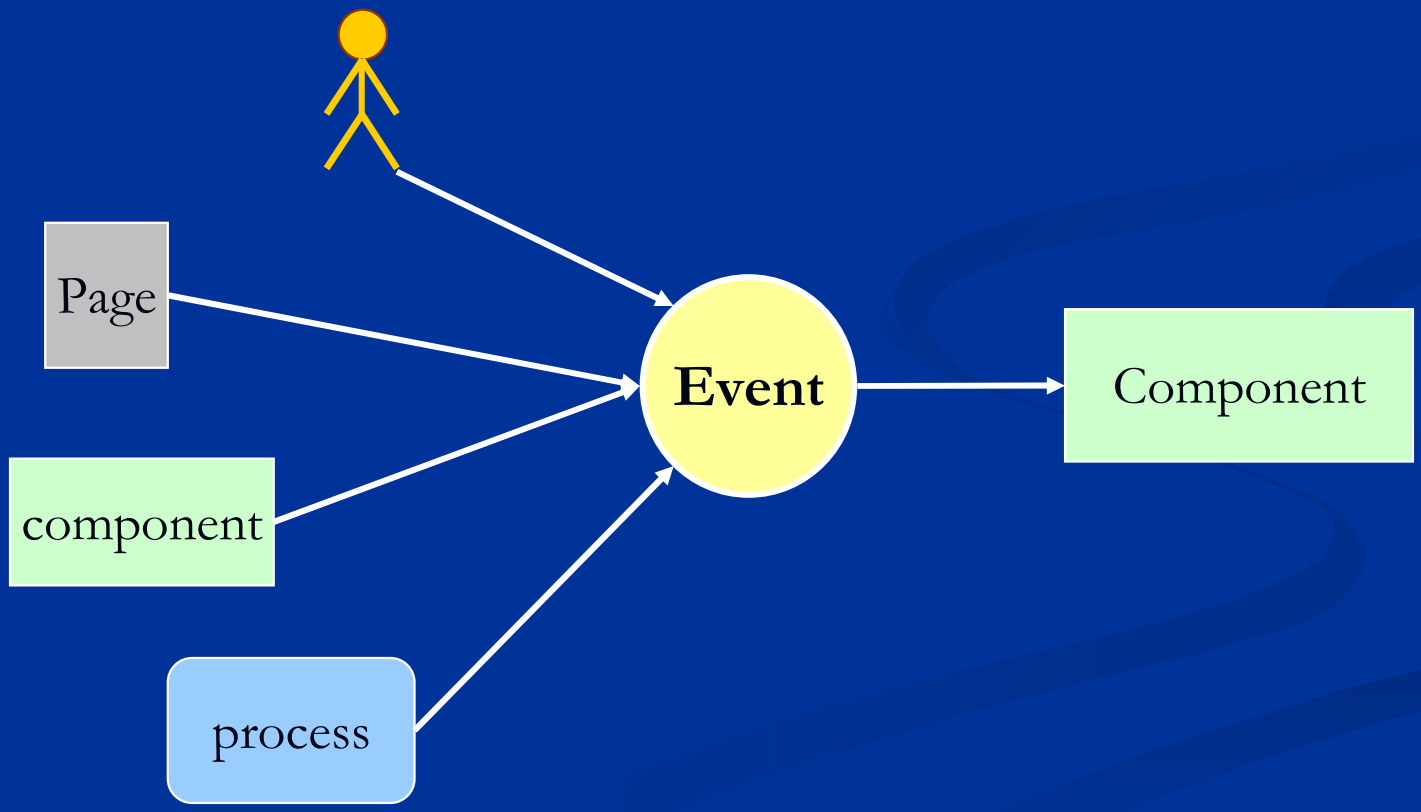
# Conversational Pageflow

- Pageflow inspect the components and changes the flow based of their state
- Pageflow can trigger event on transition



# Seam Event model

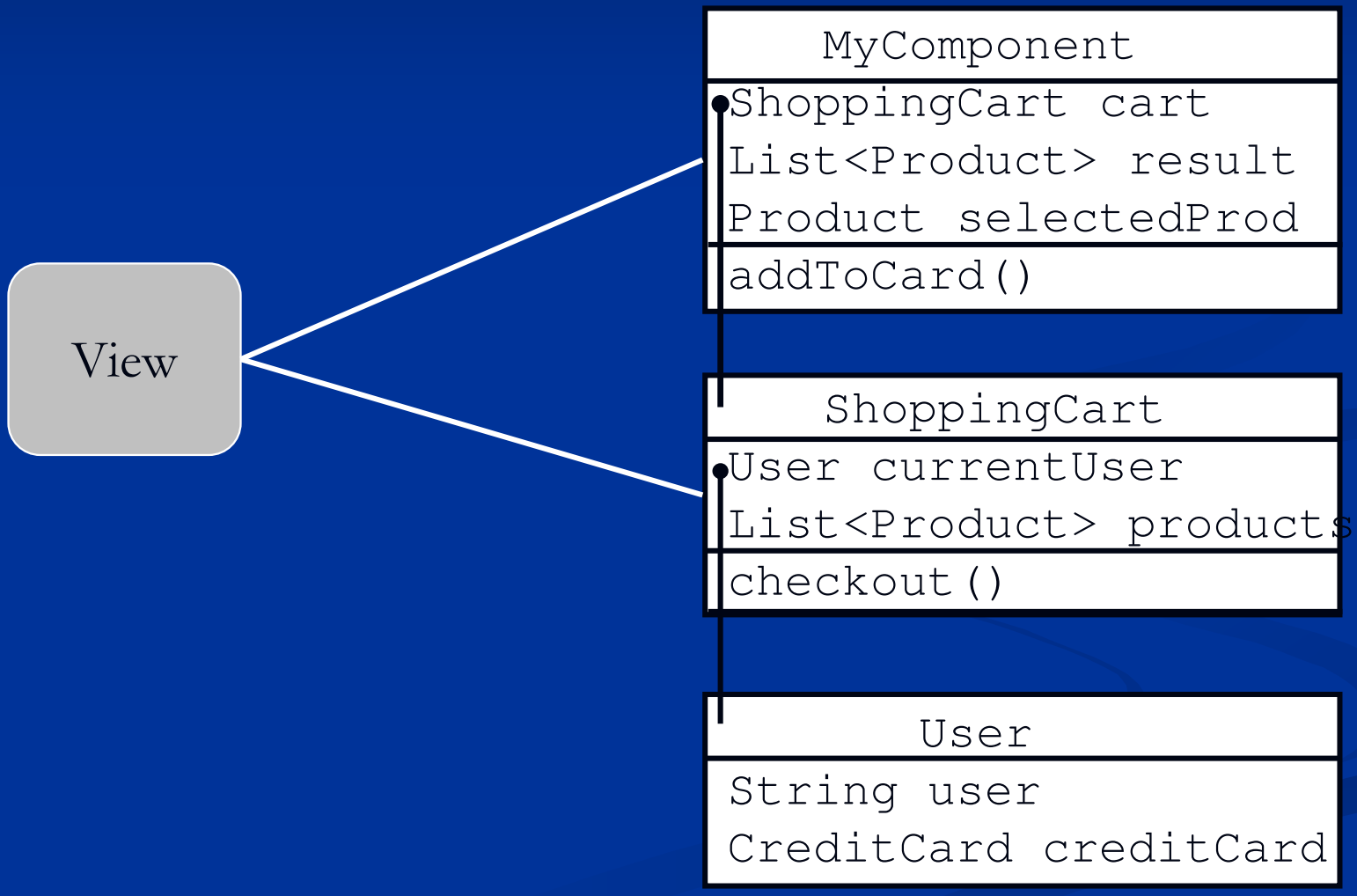
- Allows to uncouple the interaction between seam components and other objects



# Inversion of Control in Seam

- Most IoC-frameworks support only stateless dependency injection
  - Static, unidirectional, non-contextual
- Seams introduces state full dependency injection
  - dynamic
  - contextual
  - bidirectional

# State full Injection – an Example

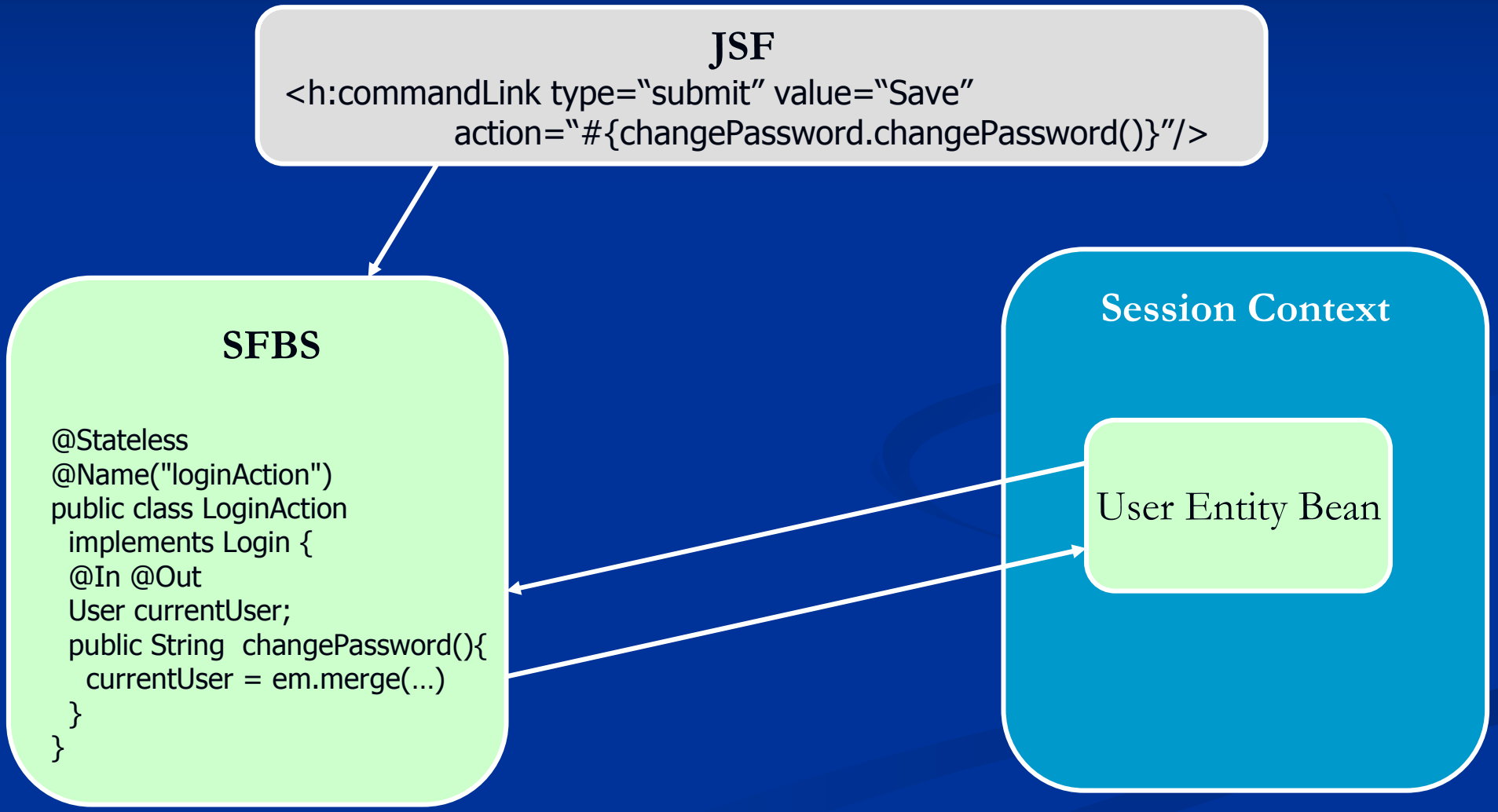


# Let look at IoC example

- @In injects the value of the contextual variable into the currentUser variable each time the component is invoked (@Out annotation is inverse)
- No dependency – aliasing a contextual variable into the namespace of a component

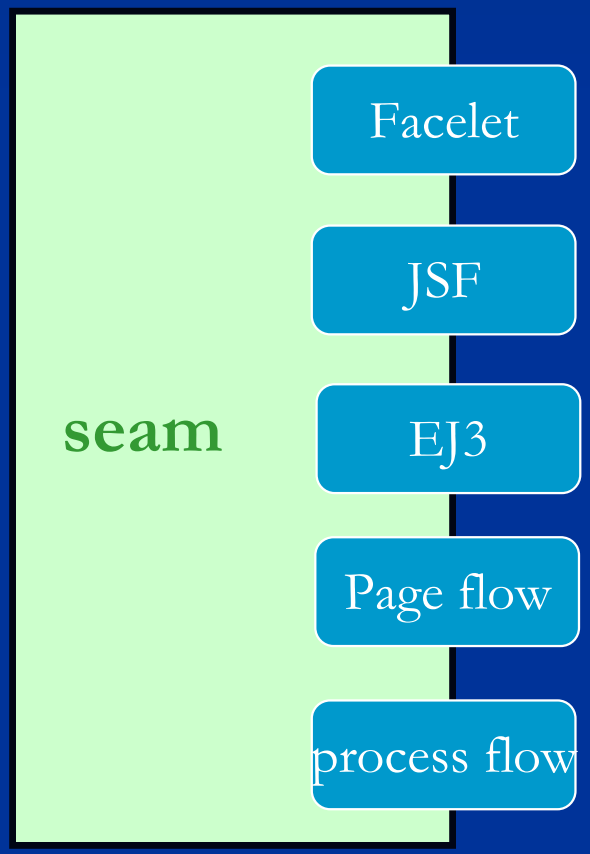
```
@Stateless
@Name("loginAction")
public class LoginAction implements Login {
@In @Out
User currentUser;
... }
```

# @In @Out


JSF
<h:commandLink type="submit" value="Save"
action="#{changePassword.changePassword()}"/>
SFBS
@Stateless
@Name("loginAction")
public class LoginAction
implements Login {
@In @Out
User currentUser;
public String changePassword(){
currentUser = em.merge(…)
}
}
Session Context
User Entity Bean

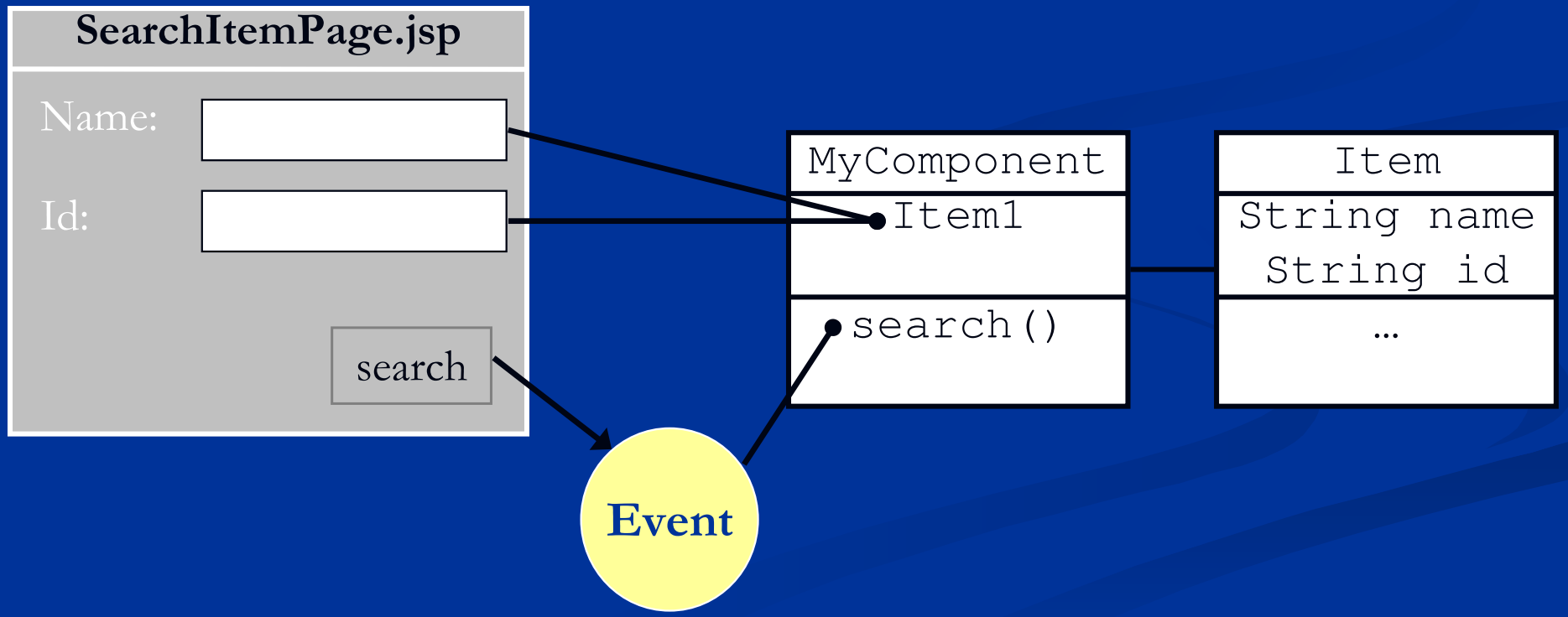
# Integrated Stack

- Wires technologies without writing code



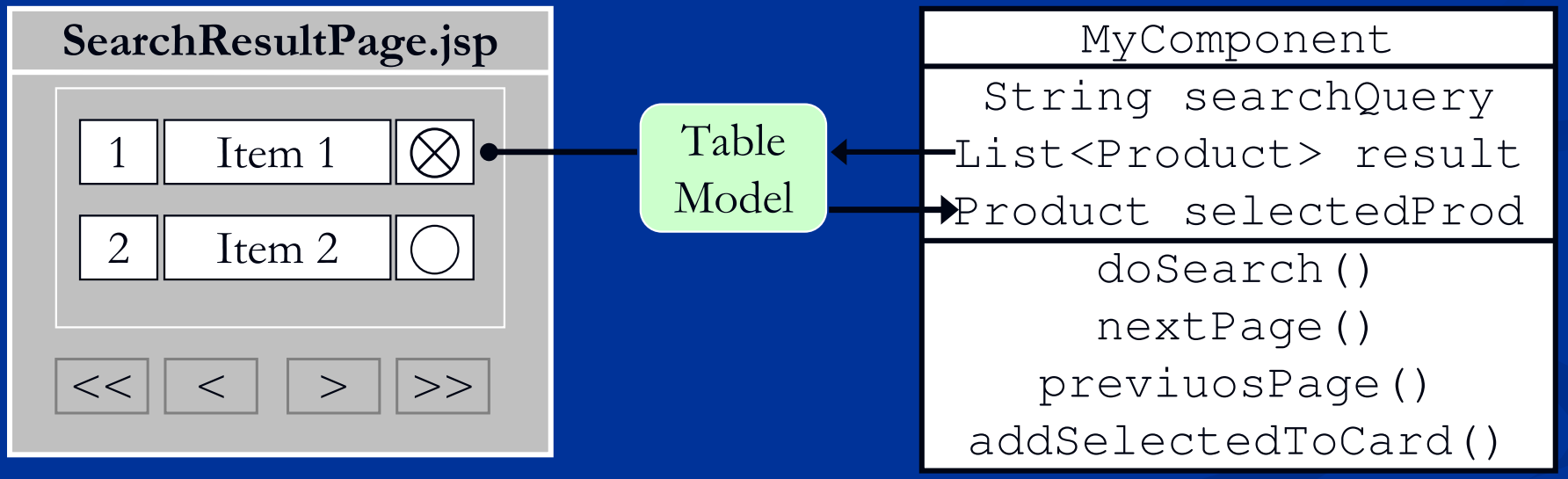
# Component based web development

- Biding from JSF directly to seams component
- Binding event as triggers to page flow or BP
- Binding UI component to seam components



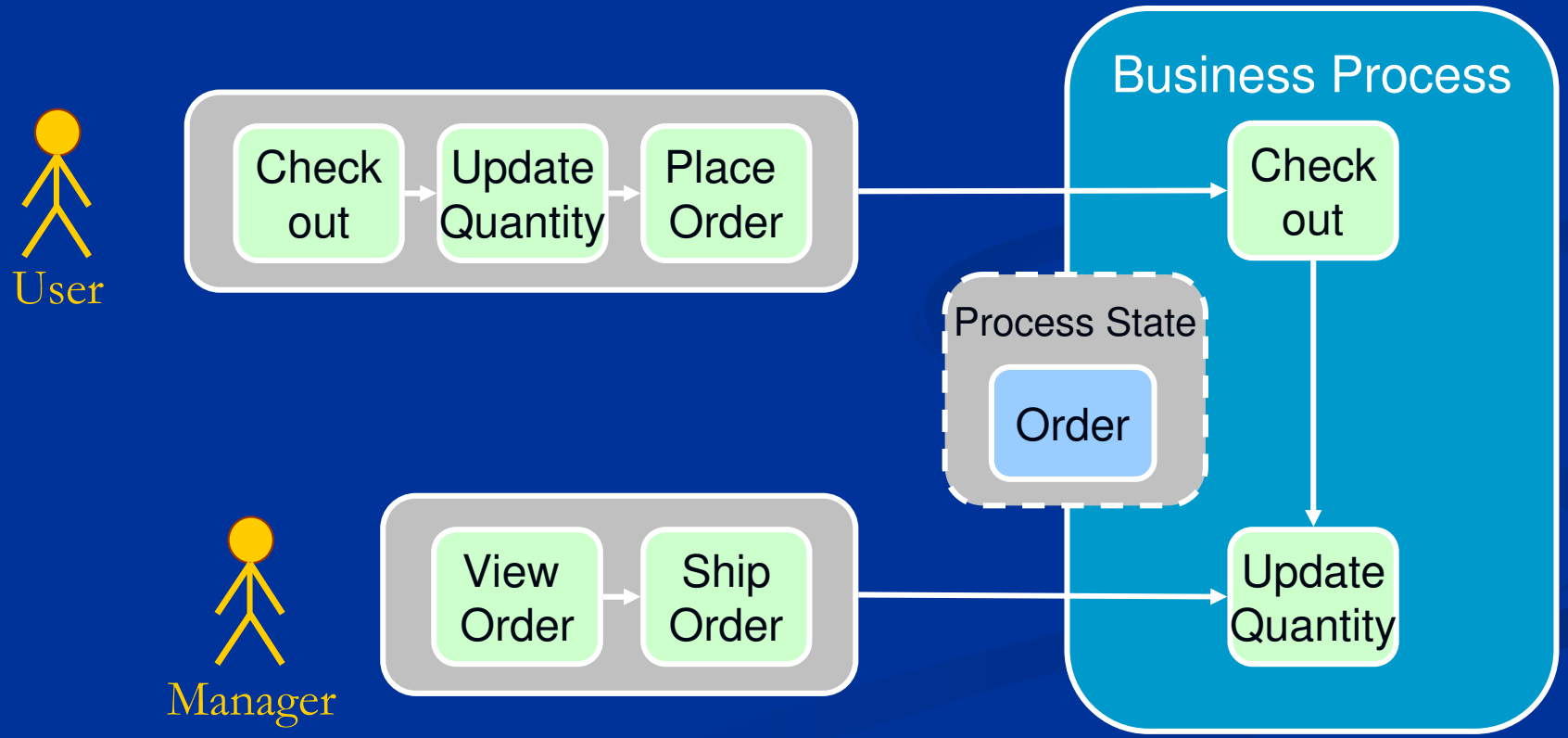
# UI Data Binding

- Map search seams component direct to table model no UI related backing beans needed to translate
- Data model annotation – transform list data to JSF data model



# Business Process

- A Business Process is a sequence of conversations with users with the system
- Processes have state that is shared between participant



# More Seam

- Seam supports conversational Ajax
- Seams supports automated testing
- Seam allows you to use facelets
- Any other Question?