

Dive Into Seam 2

Ivan Li
(email2liyang@gmail.com)

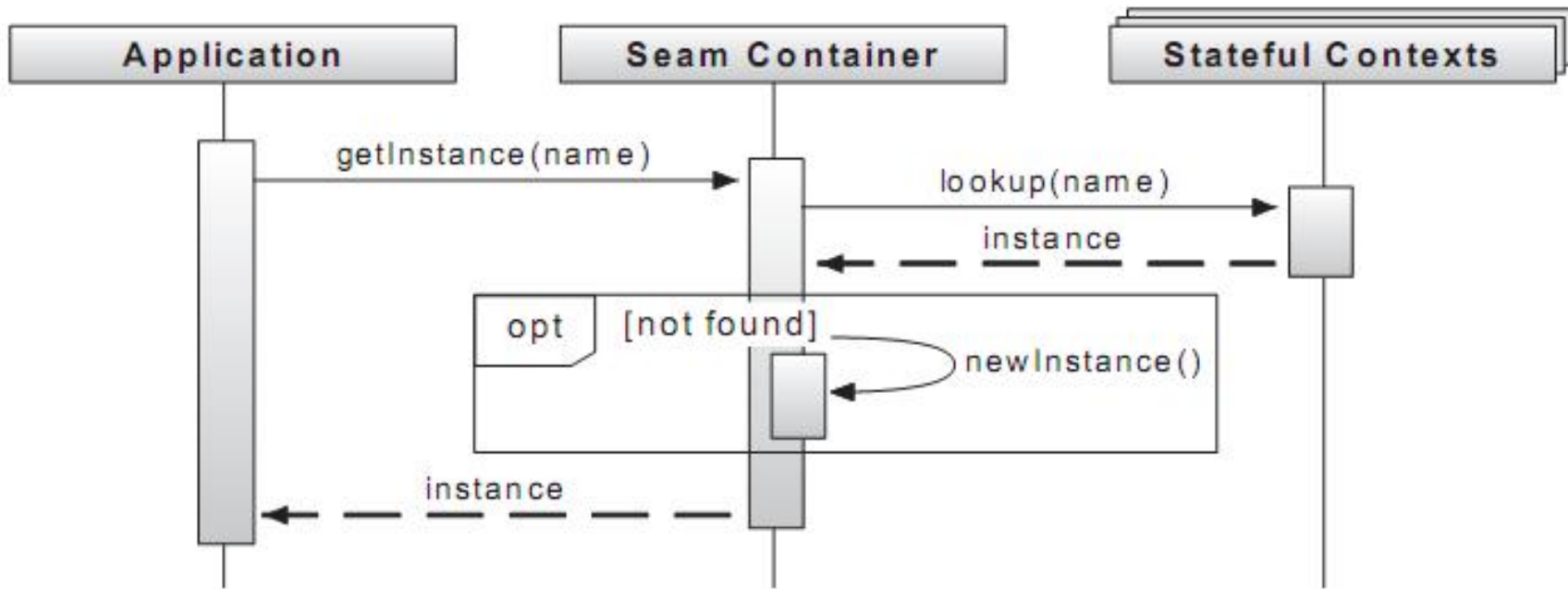
Agenda

- Components and Contexts
- Absolute Inversion Of Control
- The Conversation: Seam's Unit Of work
- More...

Components and Contexts

- Component VS Component Instance(Java class vs Java Object)
- Seam Component is stateful, Stored in Seam Context
- Seam Context Scope
 - Stateless -- Spring's Prototype Scope
 - Event -- Servlet Request Scope
 - Page -- Store in JSF Component Tree
 - Conversation -- Seam's Innovation
 - Session -- Servlet Session Scope
 - Business Process --multiple conversations for multiple user(BPM)
 - Application -- Servlet Application Scope
 - Unspecified --scope should be implied

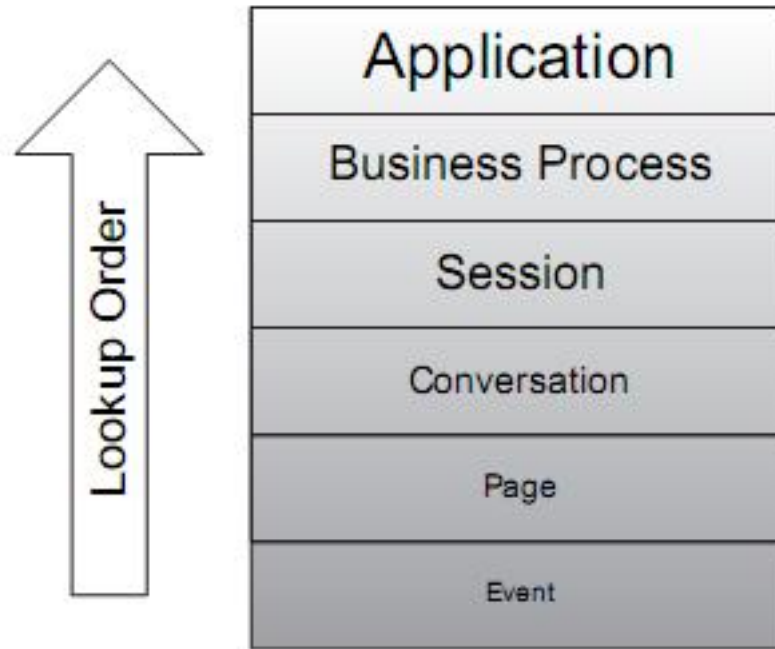
Components and Contexts



Sequence diagram of a component instance being requested from the Seam container

- Forget **New**, let seam create Instance for you
- Dynamic Proxy: [JavaSsist](#)

Components and Contexts



Seam Context Variable Lookup Order

How to Define Components

- Define Component By Annotations(recommended)

Annotation	What it does
<code>@Name</code>	Declares a class as a Seam component and assigns it a name. When the component name is requested, Seam instantiates a new instance of the class and binds it to the context variable with the same name.
<code>@Scope</code>	Specifies the default scope in which the context variable is stored when an instance of the component is instantiated.
<code>@Role (@Roles)</code>	Defines alternate name and scope pairings for the component that can be used to instantiate parallel instances of the same component for different purposes. Multiple <code>@Role</code> declarations are wrapped in a <code>@Roles</code> annotation.
<code>@Startup</code>	Instructs Seam to instantiate the component automatically when its assigned context starts (only applies to application-scoped and session-scoped components).
<code>@Namespace</code>	Binds a URI to a Java package and used to define custom XML namespaces in the Seam component descriptor.
<code>@Install</code>	Used to make installation of a component conditional or to provide a precedence to override another component definition. Conditions include the presence of a class on the classpath, the presence of another component, or the debug mode setting.
<code>@AutoCreate</code>	Tells Seam to create a new instance of the component when the component name is first requested, even if the calling code doesn't request for it to be created.

- Define Component By XML

- Some Class In jar, No access to Src

Seam Component's Life

- Loading component definitions
 - Scan component at startup
 - Store component definition in ApplicationScope
- Control When to @Install a Component
 - @Install
- Give Component Multiple Roles
 - @Role = @Name+@Scope
 - Get Component Instance multiple time
- Instantiating components at @Startup
 - Only SessionScope and ApplicationScope
- Component life-cycle callbacks
 - @Create VS @PostConstruct
 - @Destroy VS @PreDestroy
- Where all components go to die
 - The context it occupies ends
 - The context variable to which it is bound is assigned a null value
 - The instance is explicitly removed using the Seam API

What about EJB SessionBean?

- Who owns the sessionbean?
 - EJB Container?
 - Seam Container?
 - EJB Container owns the SessionBean
 - Seam Container borrow it from EJBContainer
 - Seam get EJB reference BY JNDILookUP
 - @Stateful
 - @Name("registerAction")
 - @JndiName("foo/FooAction/local")
- ```
public class FooActionBean implements FooAction { ... }
```

# Accessing components

| From where                                                     | How accessed                                    |
|----------------------------------------------------------------|-------------------------------------------------|
| JSF view                                                       | EL notation                                     |
| Seam annotation (e.g., @In, @Out)                              | Component name or EL notation                   |
| Java code: <code>Component.getInstance()</code>                | Component name or component class               |
| Java code: <code>Expressions.getInstance()</code>              | EL notation                                     |
| JPQL or HQL query                                              | EL notation                                     |
| Java properties (i18n bundle or component property)            | EL notation                                     |
| JavaScript using Seam Remoting                                 | Component name, stub instance, or EL notation   |
| Seam component descriptor (e.g., <code>components.xml</code> ) | Component name, component class, or EL notation |
| Seam page descriptor (e.g., <code>pages.xml</code> )           | EL notation                                     |
| jPDL page flow descriptor                                      | EL notation                                     |
| jBPM business process descriptor                               | EL notation                                     |
| Spring configuration file                                      | Component name or EL notation                   |

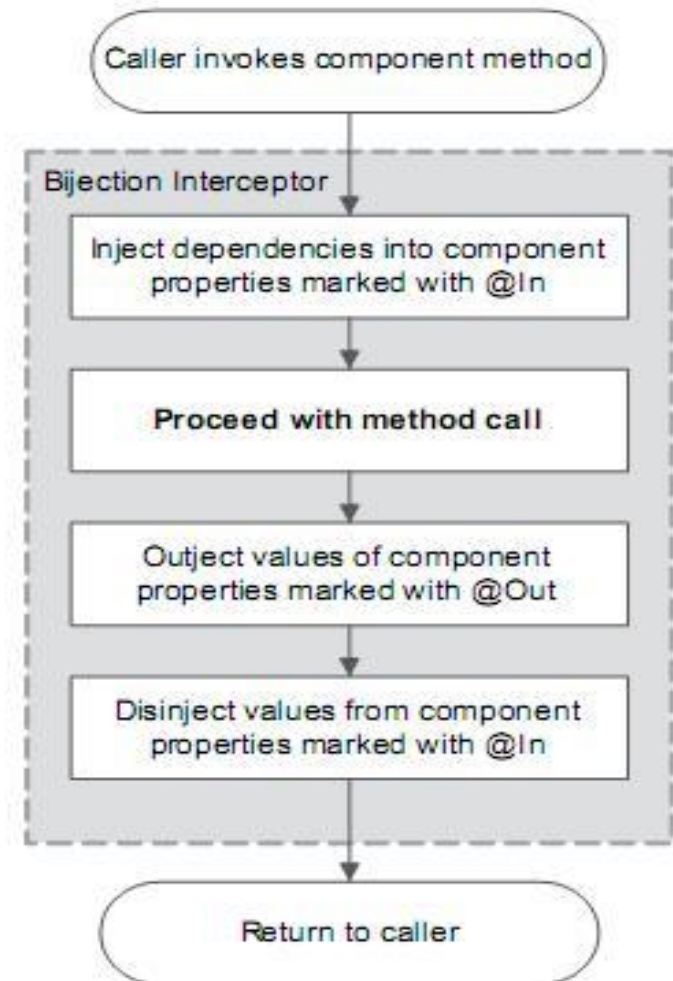
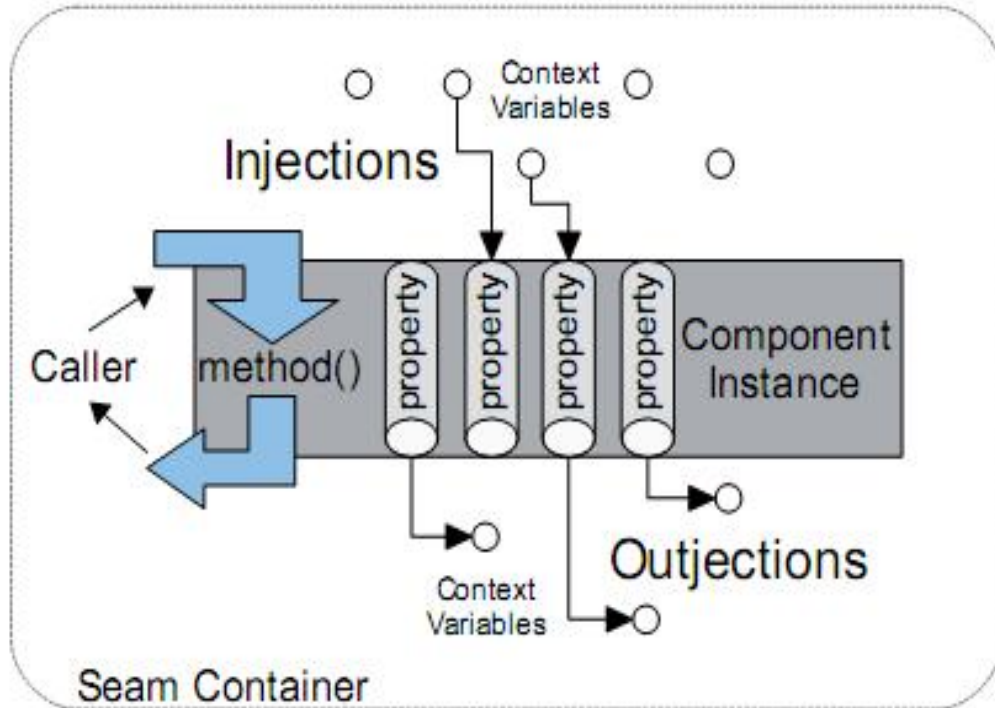
With SEAM's ENHANCED EL

# Absolute Inversion Of Control

- Introducing bijection
- Dynamic dependency @In-jection
- @Out-jecting context variables
- Bypassing bijection
- Component events
- Factory and manager components

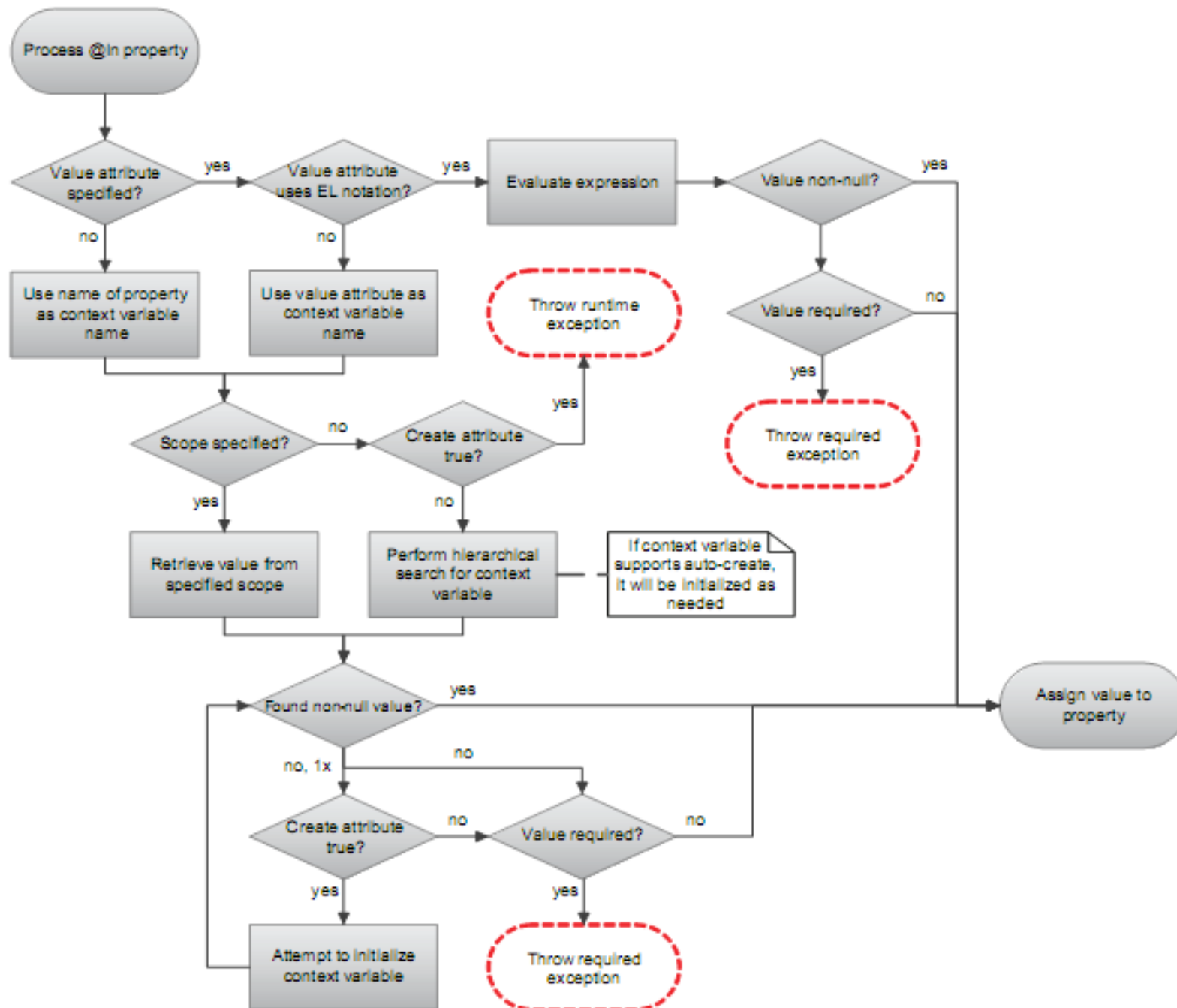
# Introducing bijection

- Bijection=Injection+OutJection
- Disinjection =Seam's way o tying up loose ends.

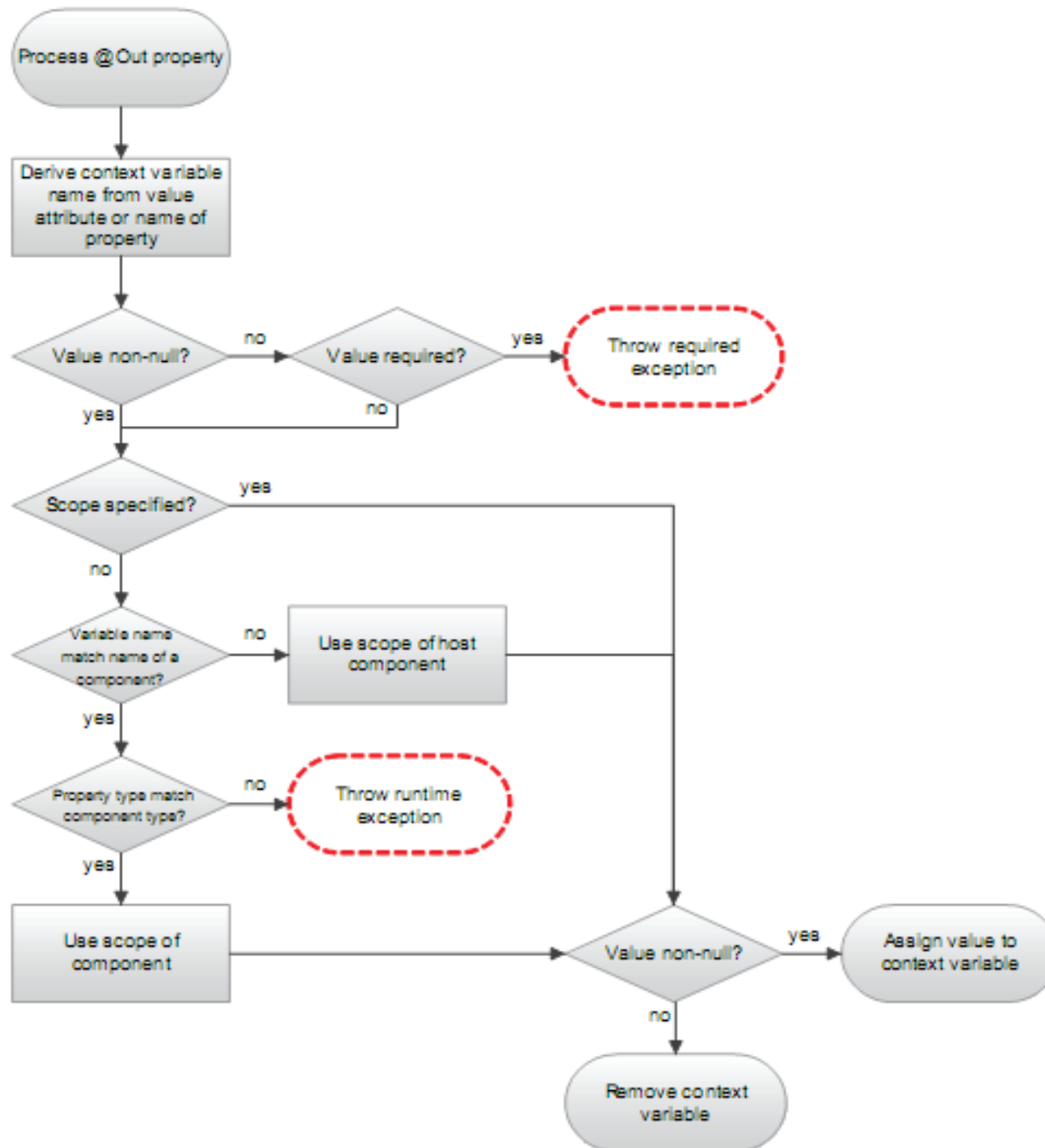


- By Interceptors

# Dynamic dependency @In-jection



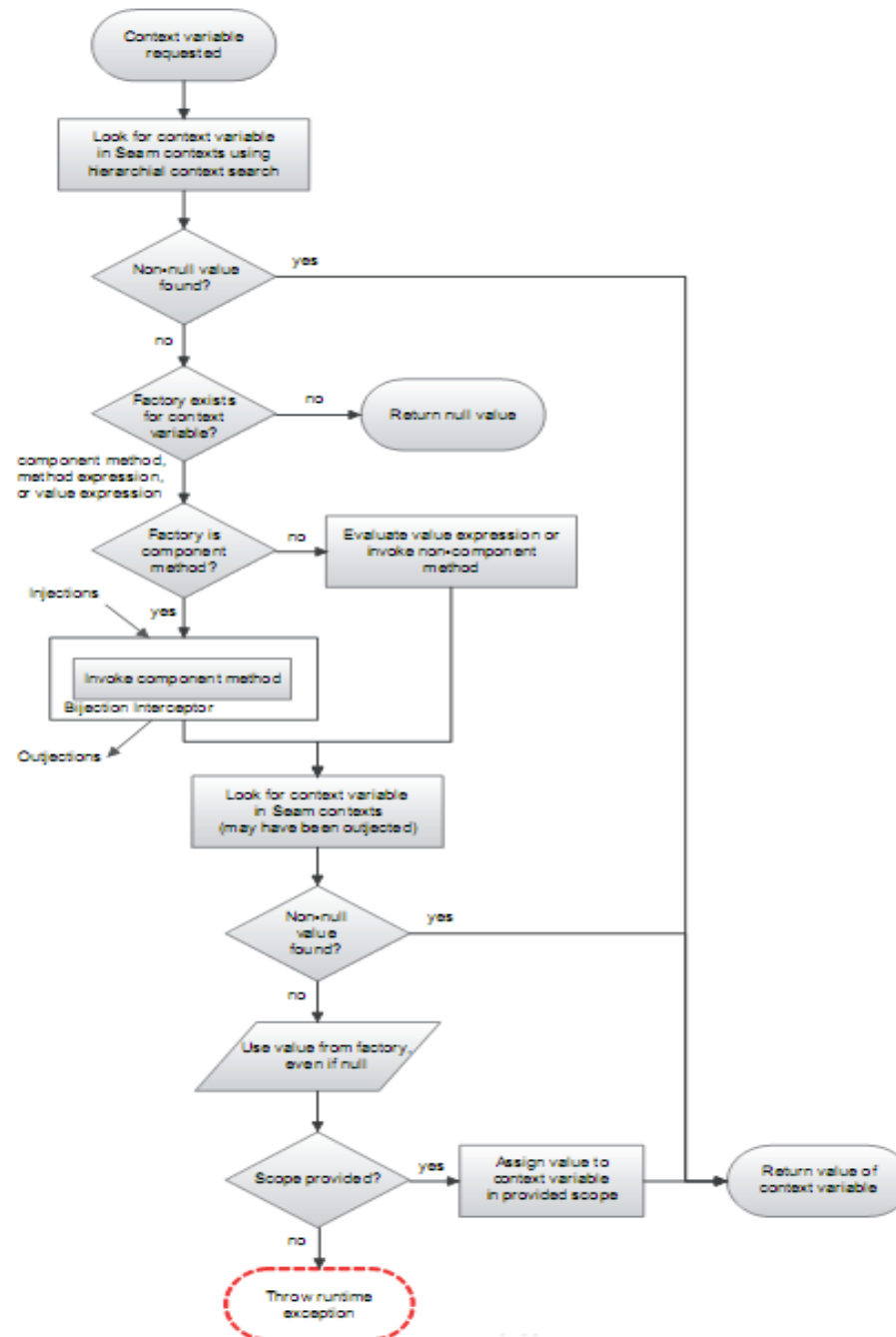
# @Out-jecting context variables



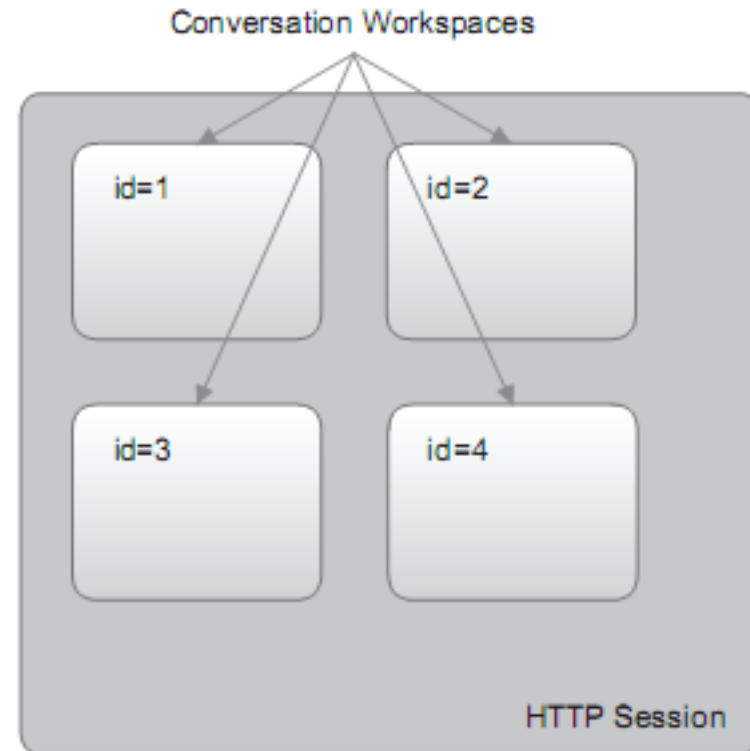
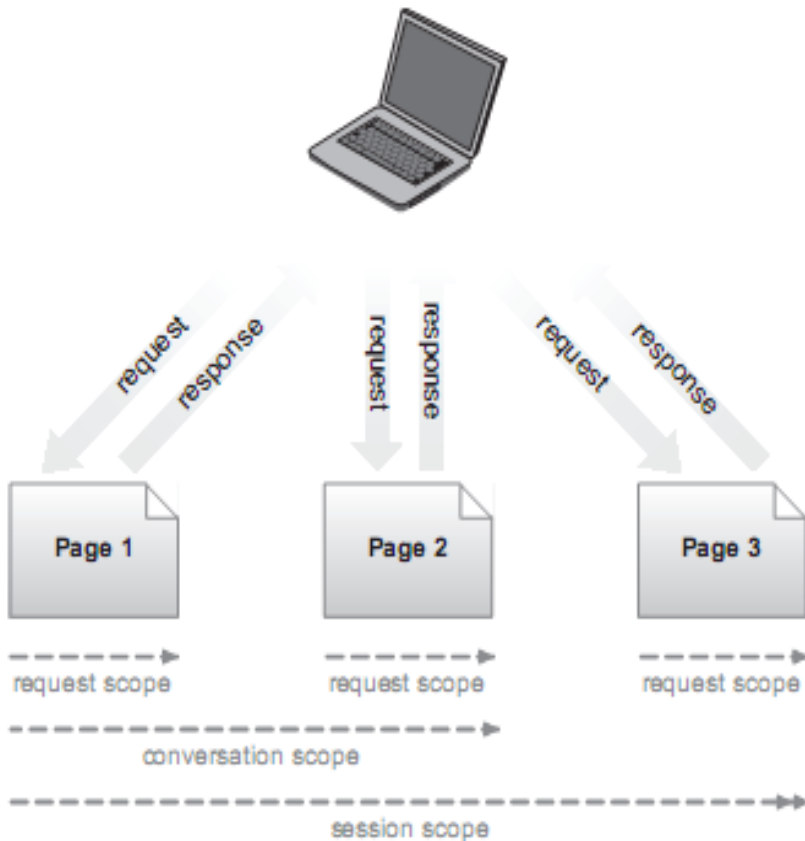
# Bypassing bijection And Component events

- Bypassing bijection
  - Internal Call
  - `@BypassInterceptors` will Disable Bijection
- Component Event
  - `@RaiseEvent("someEvent")`
  - `Events.instance().raiseEvent("someEvent");`
  - `@Observer("someEvent")`

# Factory and manager components



# The Conversation: Seam's Unit Of work



Conversation default Timeout:600000 msec

```
<core:manager conversation-timeout="3600000"/>
```

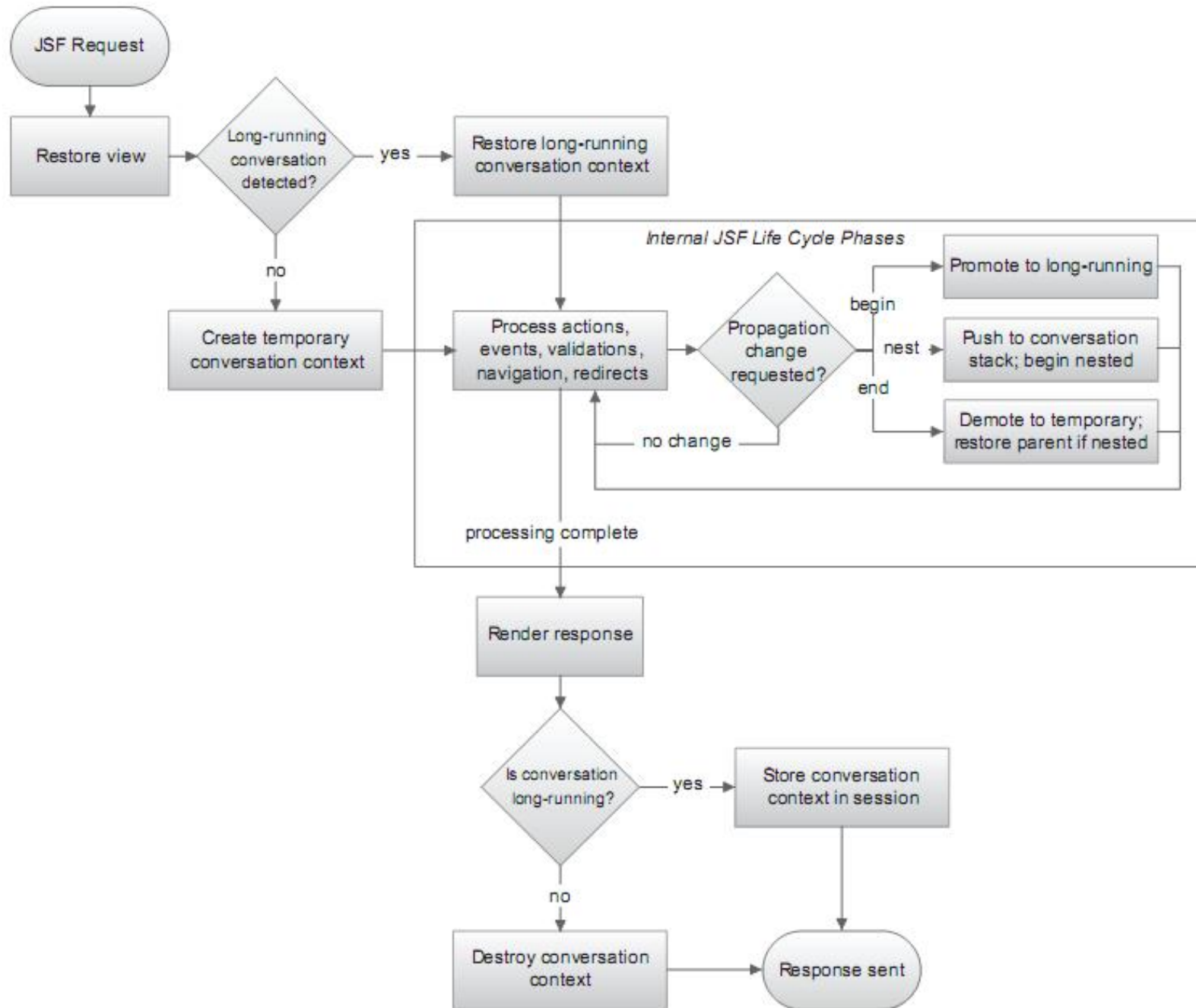
```
<page view-id="/coursewizard/holeData.xhtml"
timeout="7200000"/>
```

# Conversation's state

- Temporary
  - Restore View ->Render Response
  - Long-running Seed
- Long-running

| Propagation type | Description                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| begin            | Promotes a temporary conversation to a long-running conversation. An exception is thrown if a long-running conversation is already active.                                                                                                   |
| join             | Promotes a temporary conversation to a long-running conversation. No action is taken if a long-running conversation is already active.                                                                                                       |
| end              | Demotes a long-running conversation to a temporary conversation.                                                                                                                                                                             |
| nest             | If a long-running conversation is active, suspends it and adds a new, long-running conversation to the conversation stack. If a long-running conversation is not active, promotes the temporary conversation to a long-running conversation. |
| none             | Abandons the current conversation. The previous conversation is left intact and a temporary conversation is created to serve the incoming request.                                                                                           |

# Conversation Process



# Keeping the conversation going

- CONVERSATIONS ON POSTBACK
  - Store Conversation Token in JSF Tree
- RESTORING A CONVERSATION FROM A GET REQUEST
  - `<a href="preview.seam?conversationId=#{conversation.id}" target="_blank">Preview</a>`

# Ending a long-running conversation

- `<page view-id="/coursewizard/*">`
  - `<navigation>`
    - `<rule if-outcome="cancel">`
      - `<end-conversation/>`
    - `</rule>`
  - `</navigation>`
- `</page>`
- `@End`
- `ConversatonTimeout`

# Reference

- <http://www.seamframework.org/>
- <http://www.manning.com/SeaminAction>
- <http://www.infoq.com/cn/articles/jboss-seam>

Q&A