

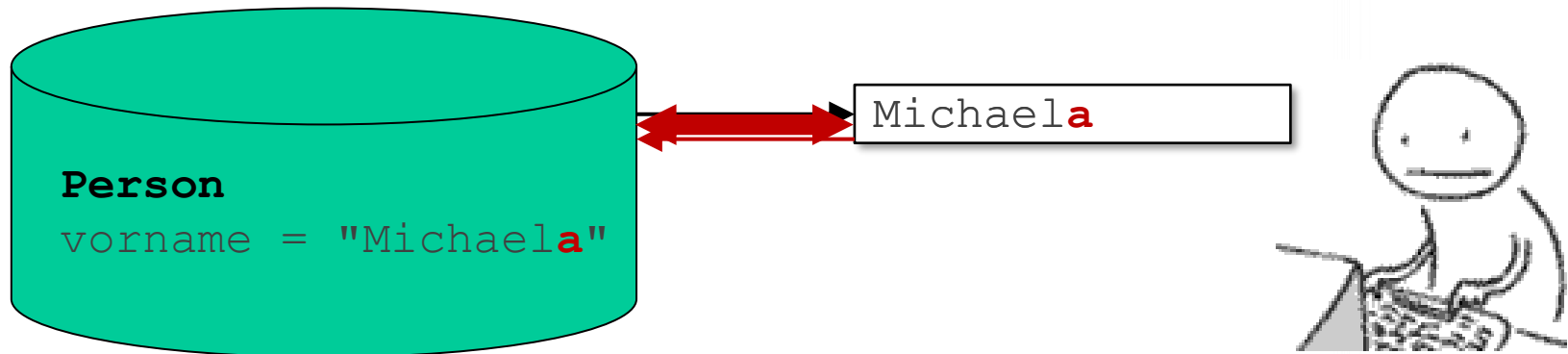


Java Server Faces Binding

- Definition
- Binding mit JSF
- JSF Managed Beans
- CDI Managed Beans

Definition

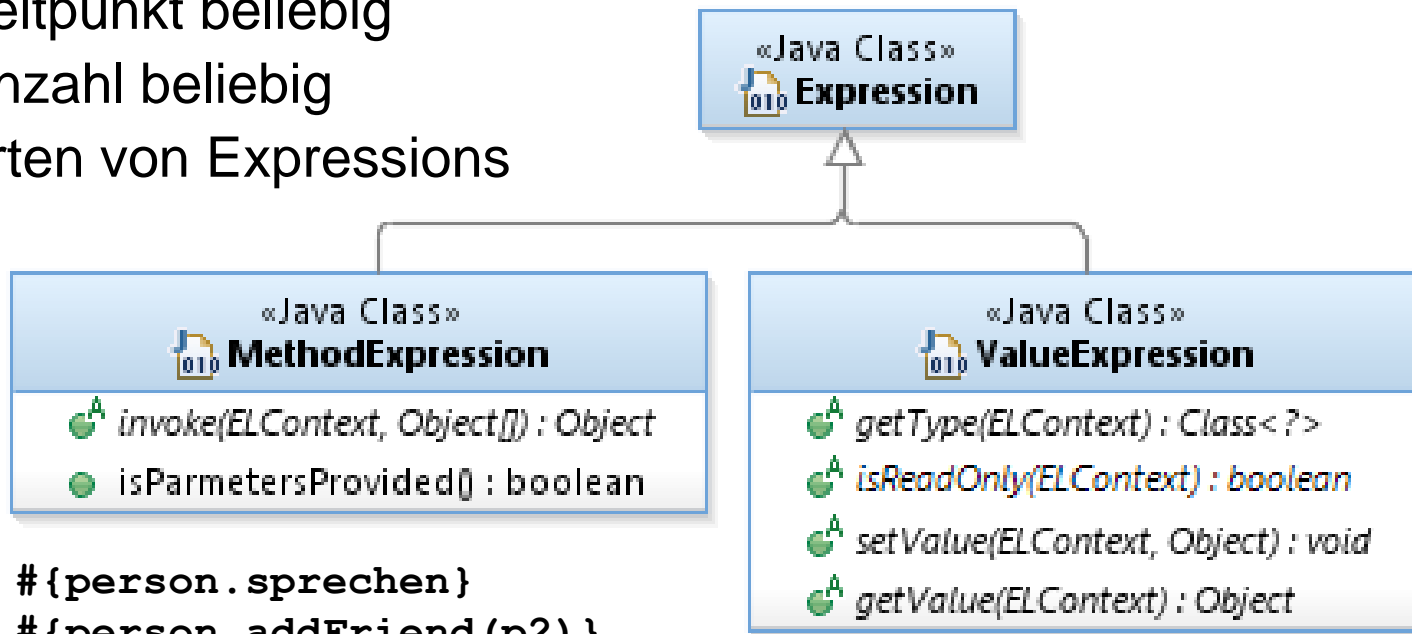
- Datendarstellung und -bearbeitung über Oberflächenkomponenten
- Datenherkunft und -ziel: Modellschicht
 - Java-Klassen
 - Datenbank
 - Web Service
- Synchronisierung zwischen Oberfläche und Modell erforderlich
 - Automatisierung über (Data) Binding



Binding mit JSF

Deferred Expressions

- Teil der Unified Expression Language
- Auswertung verzögert
 - Zeitpunkt beliebig
 - Anzahl beliebig
- 2 Arten von Expressions



`# {person.sprechen}`
`# {person.addFriend(p2)}`

- Referenz auf Methode

`# {person.name}` `# {1+1}`

- Referenz auf Attribut (Getter und Setter) oder Ausdruck

Binding mit JSF

Ausgabekomponenten

- Ausgabewert (`value`)
- Bedingte Anzeige (`rendered`)

```
<h:outputText value="#{person.name}"  
              rendered="#{page.view}" />
```

Eingabekomponenten

- Ein-/Ausgabewert (`value`)
- Bedingte Anzeige (`rendered`)
- Bedingte Aktivierung/Fokus (`disabled`)
- Bedingte Eingabesperre (`readonly`)

```
<h:inputText value="#{person.name}"  
             disabled="#{empty user.roles['editor']}" />
```

Binding mit JSF

Aktionskomponenten

- Aktionsmethode (`action`)

```
<h:commandButton action="#{person.sprechen}" />
```

Weitere Bindings

- Validierungsmethoden (siehe Validierung, Backing Beans)
- `ValueChangeListener`-Methode (siehe Event Handler)

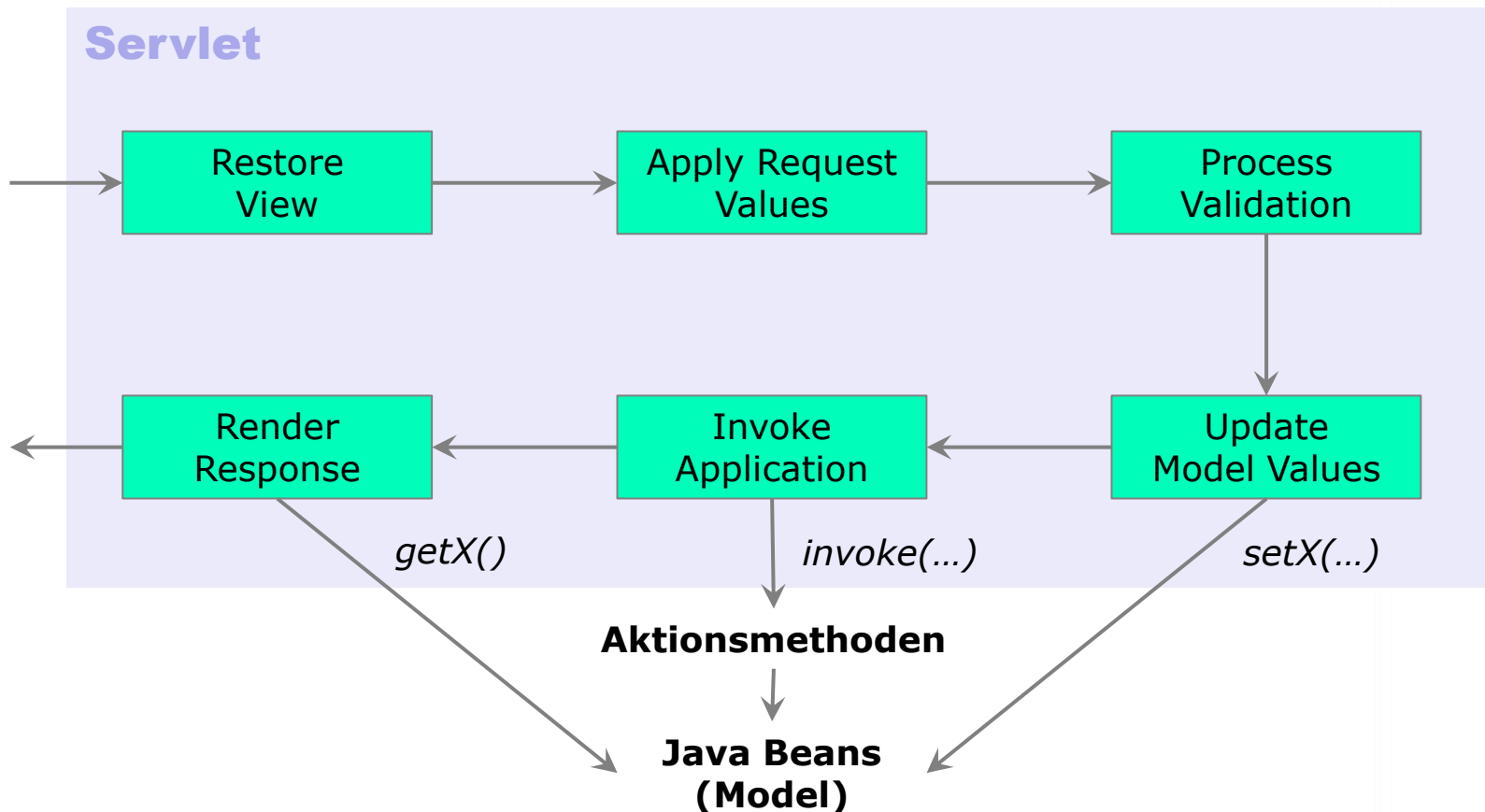
Binding mit JSF

Beispiel

```
<h:form>
  <h:outputLabel align="First Name:" for="firstName"/>
  <h:inputText id="firstName"
    value="#{person.firstName}"/>
  <br/>
  <h:outputLabel align="Last Name:" for="lastName"/>
  <h:inputText id="lastName"
    value="#{person.lastName}"/>
  <br/>
  <h:commandButton type="submit" value="Submit"
    action="#{registration.add}"/>
</h:form>
```

Binding mit JSF

Verwendung während der Requestverarbeitung



JSF Managed Beans

Binding über Deferred Expression

- Bindung an eine Bean Property
- Auflösen der Bean durch EL Variable Resolver
 - Standardauflösung
 - Implizite Objekte
 - Gültigkeitsbereiche
 - JSF Managed Bean
 - CDI Managed Bean

```
<h:outputText value="#{person.name}" />
```



???

JSF Managed Beans

Konzept

- Verwaltung von Objekten (Beans) durch Container
 - Lebenszyklus
 - Attribute (Dependency Injection)
- Bereitstellung in Gültigkeitsbereichen
- Erreichbarkeit in Facelets (u.a. für Binding)

Erstellung

- Implementieren einer Klasse
- Registrieren einer Instanz mit Namen, Klasse und Gültigkeitsbereich
 - Datei `faces-config.xml`
 - Annotation

JSF Managed Beans

```
@ManagedBean (name="booking")
@SessionScoped
public class Booking implements Serializable {

    private long bookingDate;

    public long getBookingDate() {
        return bookingDate;
    }

    public void setBookingDate(long date) {
        this.bookingDate = date;
    }

}
```

Alternative zu Annotation: Registrierung in faces-config.xml

```
<managed-bean>
    <managed-bean-name>booking</managed-bean-name>
    <managed-bean-class>de.ars.training.bo.Booking</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
```

JSF Managed Beans

Gültigkeitsbereiche

- Angabe für Managed Bean
- Beeinflussung der Lebensdauer einer Instanz
- Erweiterung durch eigene Gültigkeitsbereiche möglich

Gültigkeitsbereich	Beschreibung
Application (@ApplicationScoped)	Eine Instanz für gesamte Webapplikation
Session (@SessionScoped)	Eine Instanz pro Benutzer
View (@ViewScoped)	Eine Instanz pro Seite (und Benutzer), Ablage im Komponentenbaum
Request (@RequestScoped)	Eine Instanz pro Anfrage
None (@NoneScoped)	Keine Sichtbarkeit in Facelets, nur für Referenzierung in anderen Beans

JSF Managed Beans

Initialisierung von Managed Beans

- Initialisierung / Finalisierung mit `@PostConstruct` / `PreDestroy`
- Initialisierung von Attributen mit `@ManagedProperty`
 - Annotation von Attribut oder Setter
 - Dependency Injection zu anderer Bean möglich

```
@ManagedBean
public class Booking {

    @ManagedProperty(value=0)
    private long bookingDate;

    @ManagedProperty(value="#{debitor}")
    public void setDebitor(Debitor debitor) {
        [...]
    }
}
```

JSF Managed Beans

Initialisierung von Managed Beans

- Alternative zu Annotationen: `faces-config.xml`

```
<managed-bean>
```

```
  <managed-bean-name>booking</managed-bean-name>
```

```
  <managed-bean-class>de.ars.training.bo.Booking</managed-bean-class>
```

```
  <managed-bean-scope>session</managed-bean-scope>
```

```
  <managed-property>
```

```
    <property-name>bookingDate</property-name>
```

```
    <value>0</value>
```

```
  </managed-property>
```

Setter erforderlich!

```
  <managed-property>
```

```
    <property-name>debitor</property-name>
```

```
    <value>#{debitor}</value>
```

```
  </managed-property>
```

```
</managed-bean>
```

JSF Managed Beans

Initialisierung von Managed Beans

- Möglichkeiten für Konfigurationsdatei

```
<managed-property>  
  <property-name>debitor</property-name>  
  <value><null-value/></value>  
</managed-property>
```

NULL

```
<managed-property>  
  <property-name>categories</property-name>  
  <list-entries>  
    <value-class>java.lang.String</value-class>  
    <value>Java-Grundlagen</value>  
    <value>Java2 Enterprise Edition</value>  
  </list-entries>  
</managed-property>
```

Listen / Arrays

JSF Managed Beans

Initialisierung von Managed Beans

- Möglichkeiten für Konfigurationsdatei

```
<managed-property>
  <property-name>categories</property-name>
  <map-entries>
    <key-class>java.lang.Integer</key-class>
    <map-entry>
      <key>1</key>
      <value>Java-Grundlagen</value>
    </map-entry>
    <map-entry>
      <key>2</key>
      <value>Java2 Enterprise Edition</value>
    </map-entry>
  </map-entries>
</managed-property>
```

Maps

CDI Managed Beans

Verwendung von CDI Managed Beans

- Bean Management durch CDI Container flexibler
- `@javax.faces.bean.ManagedBean` **VS.** `@javax.annotation.ManagedBean`
- CDI definiert eigene Gültigkeitsbereiche (z.B. Conversation)
- CDI Managed Beans in JSF Managed Beans injizierbar

```
@javax.enterprise.context.RequestScoped
@Named("person") // Name im Facelet
public class Person {

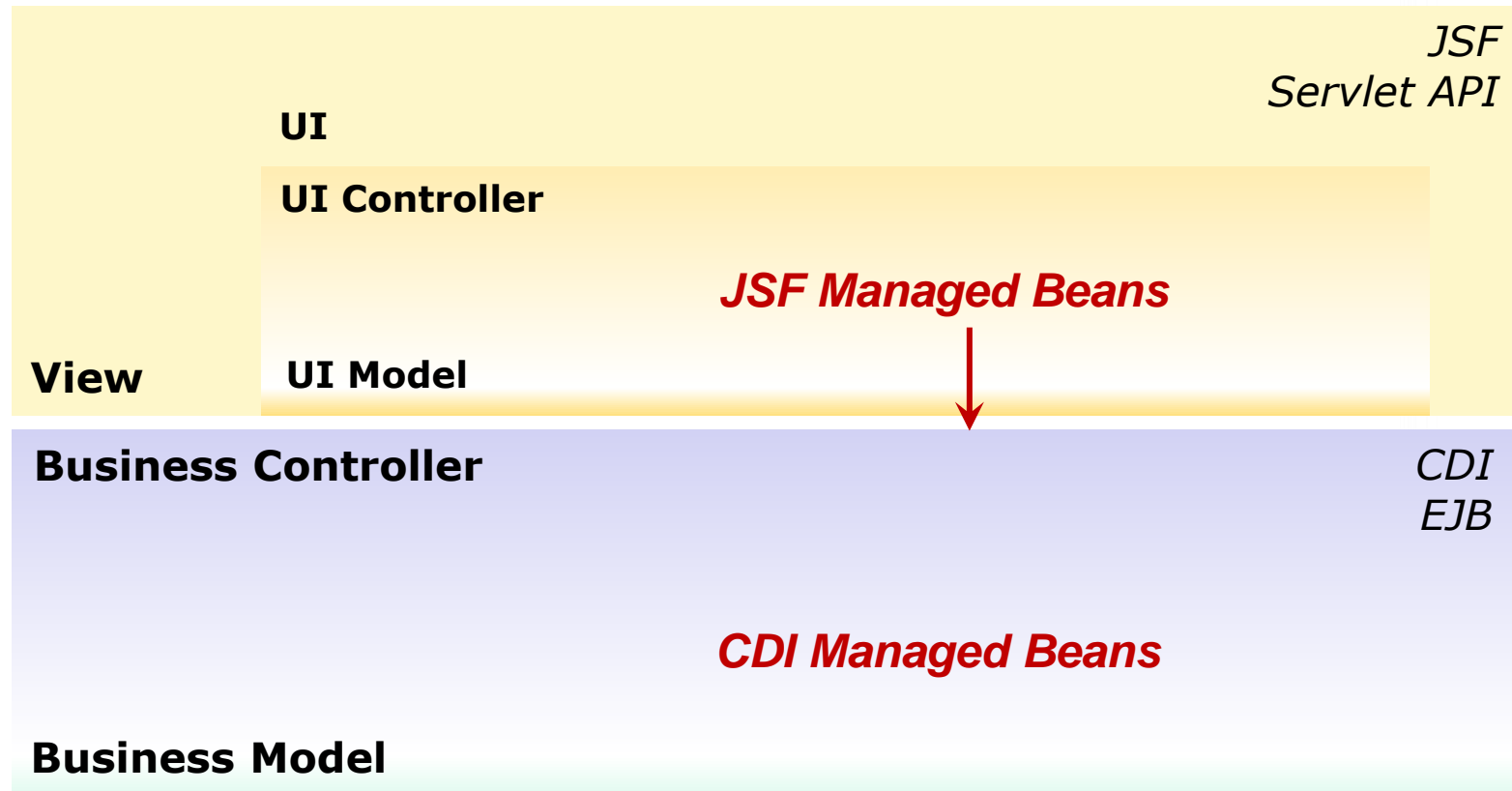
    [...]

}
```

Wann CDI, wann JSF Managed Beans?

CDI Managed Beans

Verwendung von CDI Managed Beans



Kontrollfragen

- Was bedeutet das Konzept „Binding“ und wie ist es in JSF realisiert?
- Wofür werden JSF Managed Beans verwendet?
- Welche Gültigkeitsbereiche gibt es für JSF Managed Beans? Was bedeuten diese Gültigkeitsbereiche?