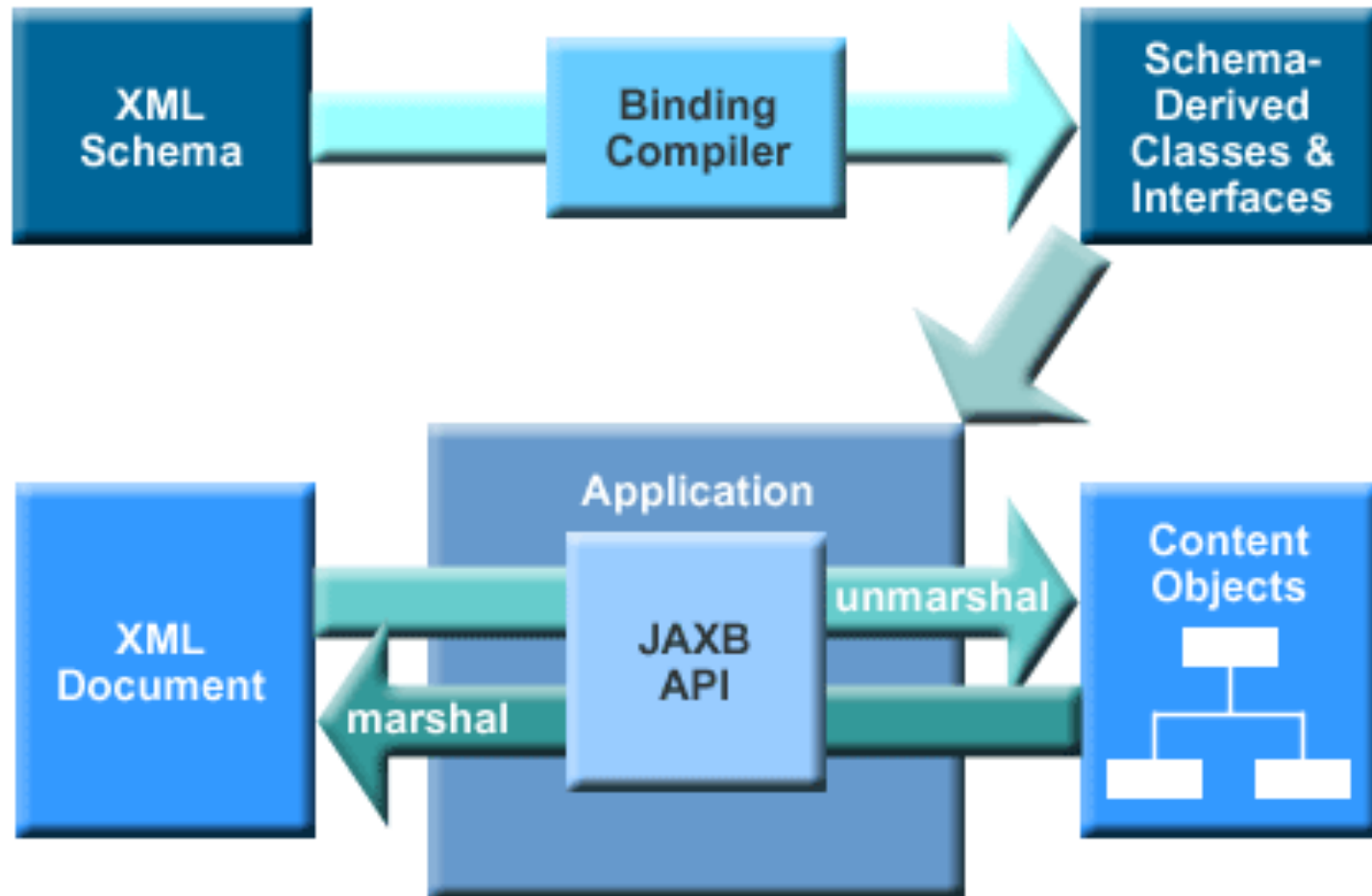


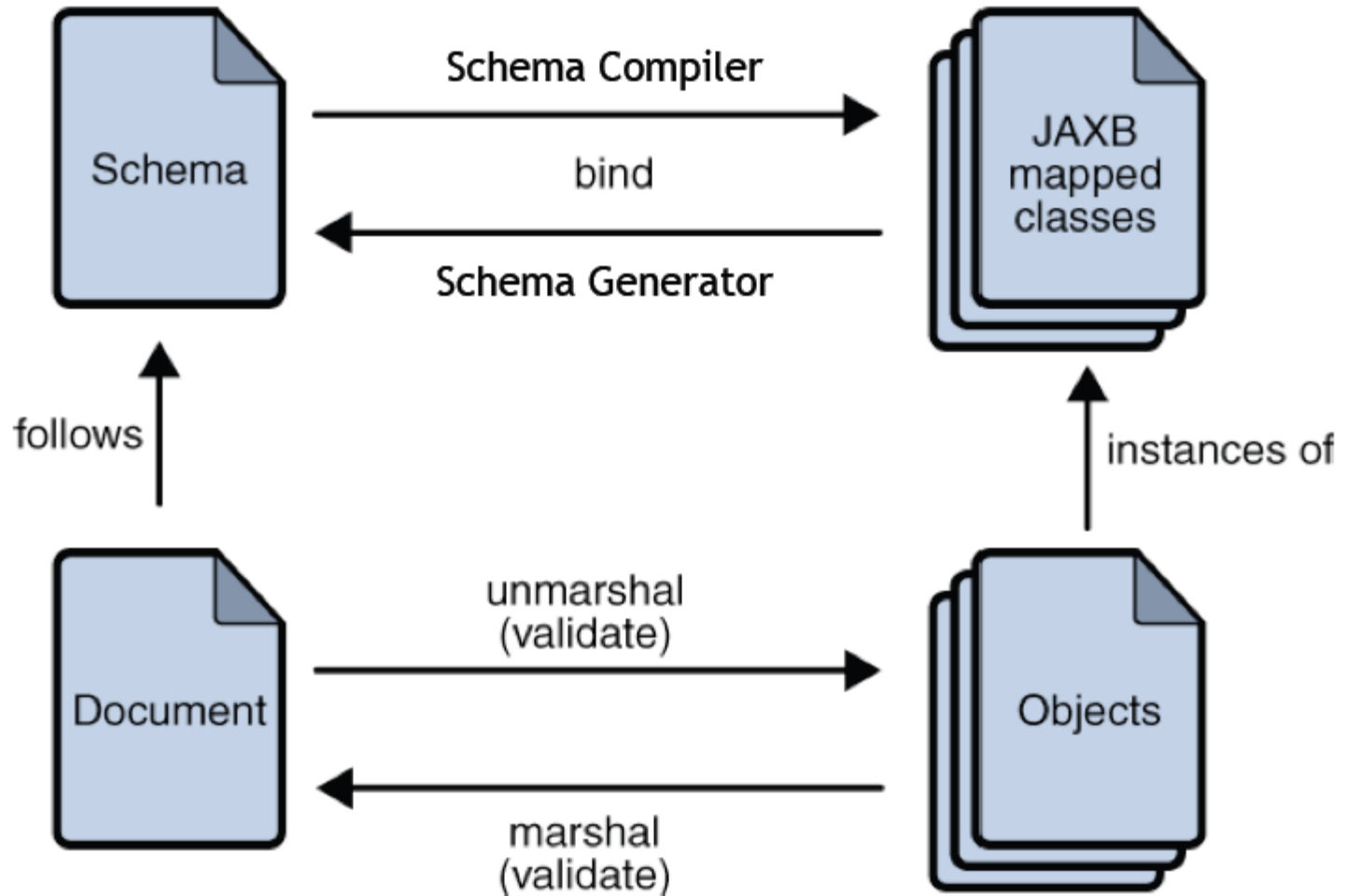
Java EE 6

Java Architecture for XML Binding - JAXB

JAXB



JAXB



Mapping eigener Klassen

- Bindung anpassbar über JAXB-Annotations
- Annotierbare Java-Programm-Elemente:
 - Felder, Properties, Klassen, Enums, Packages
- Dokumentation:

<http://java.sun.com/javase/6/docs/api/index.html>

package javax.xml.bind.annotation



@XMLElement

- Java-Elemente: Felder, Properties
- Verwendung: bindet ein Feld oder ein Property an eine XML-Element-Deklaration
- Eigenschaften:
 - name: String
 - namespace: String
 - required: boolean(false → minOccurs=0)
 - type: Class
 - defaultValue: String(→ default=“...“)



@XMLType

- Java-Elemente: Klassen, Enums
- Verwendung:
 - bindet eine Klasse oder Enum an einen XML-Typ
 - Reihenfolge der Unterelemente bestimmbar
- Eigenschaften
 - name: String
 - namespace: String
 - propOrder: String[] (→ <xsd:sequence>; leeres Array → <xsd:any>)



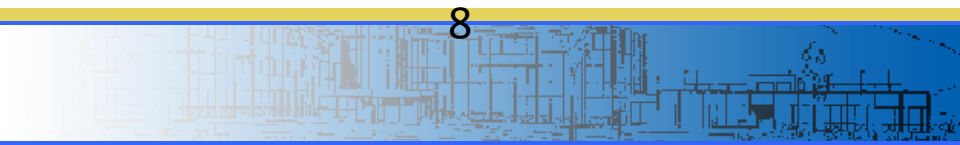
@XMLAccessType

- Java-Elemente: Klassen, Packages
- Verwendung: bestimmt, welche Felder und Properties(JavaBean-konforme Felder) an XML-Deklarationen gebunden werden
- Eigenschaften:
 - `XmlAccessType.FIELD`: bindet alle Java- Elemente ungeachtet ihrer Sichtbarkeit
 - `XmlAccessType.NONE`: bindet keines der Java-Elemente
 - `XmlAccessType.PROPERTY`: bindet alle Properties
 - `XmlAccessType.PUBLIC_MEMBER`: Default, bindet alle public Felder und Properties



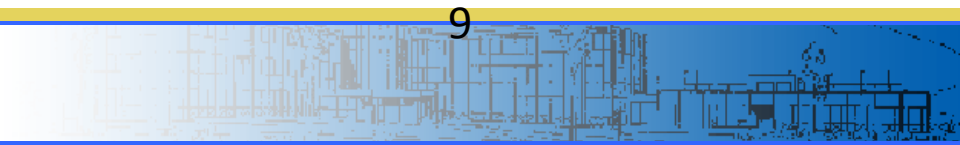
@XMLTransient

- Java-Elemente: Felder, Properties
- Verwendung: schließt ein Feld oder eine Property von der Bindung aus (z. B. um Namenskonflikte aufzulösen)



@XMLAttribute

- Java-Elemente: Felder, Properties
- Verwendung: bindet ein Feld oder eine Property an eine XML-Attribut-Deklaration
- Eigenschaften
 - name: String
 - namespace: String
 - required: boolean(true → use="required")



```

@Entity
@Table(name = "PROJECT")
@XmlAccessorType(XmlAccessType.FIELD) // Field access
@XmlRootElement(name="myProject") // XML Root
@XmlType(propOrder = {"id","position","name","projectitemCollection"}) // Order of output
@NamedQueries({
    @NamedQuery(name = "Project.findAll", query = "SELECT p FROM Project p"),
    @NamedQuery(name = "Project.findById", query = "SELECT p FROM Project p WHERE p.id = :id"),
    @NamedQuery(name = "Project.findByName", query = "SELECT p FROM Project p WHERE p.name = :name"),
    @NamedQuery(name = "Project.findByPosition", query = "SELECT p FROM Project p WHERE p.position = :position"))})
public class Project implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    // @Basic(optional = false)
    // @NotNull
    @XmlElement(name="myID")
    @Column(name = "ID")
    private Long id;

    @XmlElement(name="myName")
    @Size(max = 255)
    @Column(name = "NAME")
    private String name;

    @XmlElement(name="myPosition")
    @Column(name = "POSITION")
    private Integer position;

    @XmlElementWrapper(name = "myProjectItemList") // name of enclosing XML-Tag
    @XmlElement(name="myProjectItem") // name of detail element
    @OneToMany(mappedBy = "parentId")
    private Collection<Projectitem> projectitemCollection;

```