
# Java EE 6
# Le novità e uso pratico

# Parte 1

# Licenza per uso e distribuzione

Questo materiale è disponibile per uso non commerciale e può essere alterato e condiviso purché sia utilizzata una licenza equivalente


Attribuzione-Non commerciale-Condividi allo stesso modo 2.5

http://creativecommons.org/licenses/by-nc-sa/2.5/it/deed

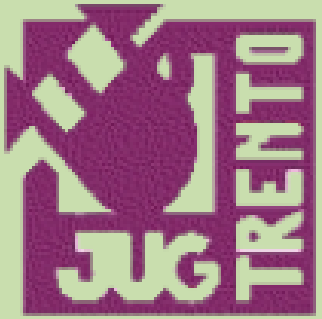
Tu sei libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera e di modificare quest'opera alle seguenti condizioni: (a) devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera; (b) non puoi usare quest'opera per fini commerciali; (c) se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.

# L'autore – Vítor Souza

- Brasiliano (allora scusate gli errori grammaticali);
- Laureato in Informatica:
  - Università Federale dello Espírito Santo, Brasile;
  - Tesi nell'area di Ingegneria del Software;
- Dottorando in informatica dell'Università di Trento;
- Sviluppatore Java dal 1999;
- Co-fondatore del ESJUG, in Brasile;
- Contatto:
  - http://www.disi.unitn.it/~vitorsouza/
  - vitorsouza@gmail.com

# JUG Trento / JUG Bolzano

- Sito Web:
  - http://www.jugtrento.org/
  - http://www.jugbz.org/
- Mailing list:
  http://groups.google.com/group/jugtaa
- Se vi interessa Java, iscrivetevi e partecipate!




JUGTRENTO.ORG

Java User Group

# Agenda

- Cos'è Java EE?
- Le novità di Java EE:
  - Uno sguardo generale;
  - Strumenti per lo sviluppo (IDE, server);
  - Creazione di un'applicazione con NetBeans;
  - Classi di dominio con JPA 2.0;
  - Validazione con Bean Validation;
  - Iniezione di dipendenza e contesti con CDI.

# Informazioni generali

- Java, Enterprise Edition:
  - Prima dalla versione 5 si chiamava J2EE;
  - Piattaforma per lo sviluppo di applicazioni aziendali (scalability, sicurezza, accessibilità, ecc.);
  - Servlets, EJBs, componenti di container con ciclo vita definito, infrastruttura condivisa;
  - Tecnologie incluse: JSP, JDBC, JPA, JSF, ecc.
- Prima versione: 1999;
- Versione 6: dicembre 2009 – JSR 316.

# Obiettivi principali

- Flessibilità:
  - Profili (*profiles*). Quello Web già definito;
  - Taglio (*pruning*): tecnologie opzionali;
- Estensibilità:
  - Punti di estensibilità: registrazione automatica di frameworks non-standard;
- Facilità di sviluppo:
  - Cominciato nella versione 5 (POJOs vs. Entity Beans);
  - Maggiormente migliorata nella versione 6.

# Tecnologie Java EE

| Tecnologia | Java EE 5 | Java EE 6 |
|---|---|---|
| Bean Validation | – | 1.0 |
| Common Annotations for the Java Platform | 1.0 | 1.1 |
| Contexts and Dependency Injection for the Java EE Platform | – | 1.0 |
| EJB (Enterprise Java Beans) | 3.0 | 3.1 CT* |
| EL (Expression Language) | 2.1 | 2.2 |
| Interceptors | – | 1.1 |
| JACC (Java Authorization Service Provider Contract for Containers) | 1.1 | 1.4 |
| JASPIC (Java Authentication Service Provider Interface for Containers) | – | 1.0 |
| Java EE Deployment API | 1.2 | 1.2 CT |
| Java EE Management API | 1.1 | 1.1 |
| JavaMail | 1.4 | 1.4 |
| JAX-RPC (Java API for XML-based RPC) | 1.1 | 1.1 CT |

**CT = Candidato al taglio / * = soltanto gli EntityBeans**

# Tecnologie Java EE

| Tecnologia | Java EE 5 | Java EE 6 |
|---|---|---|
| JAX-RS (Java API for RESTful Web Services) | – | 1.1 |
| JAX-WS (Java API for XML Web Services) | 2.0 | 2.2 |
| JAXB (Java Architecture for XML Binding) | 2.0 | 2.2 |
| JAXR (Java API for XML Registries) | 1.0 | 1.0 CT |
| JCA (Java EE Connector Architecture) | 1.5 | 1.6 |
| JMS (Java Messaging Service) | 1.1 | 1.1 |
| JPA (Java Persistence API) | 1.0 | 2.0 |
| JSF (JavaServer Faces) | 1.2 | 2.0 |
| JSP (JavaServer Pages) | 2.1 | 2.2 |
| JSTL (Standard Tag Library for JavaServer Pages) | 1.2 | 1.2 |
| JTA (Java Transaction API) | 1.1 | 1.1 |
| Managed Beans | – | 1.0 |
| Servlet | 2.5 | 3.0 |
| Web Services Metadata for the Java Platform | 2.0 | 2.1 |

# Strumenti

- Application Server:
  - GlassFish Enterprise Server v3;
  - TMAX JEUS 7 (commerciale, fine 2010);
  - JBoss 6 (M3, ancora incompleto, non certificato);
- Ambiente di sviluppo (IDE):
  - NetBeans 6.8;
  - Eclipse con GlassFish Plugins.

# NetBeans 6.8 + GlassFish v3

## NetBeans IDE 6.9 Download

6.9 | Development | Archive

Email address (optional): 

IDE Language: English

Platform: Linux (x86/x64)

Subscribe to newsletters: ☑ Monthly ☐ Weekly

☑ NetBeans can contact me at this address

Note: Greyed out technologies are not supported for this platform.

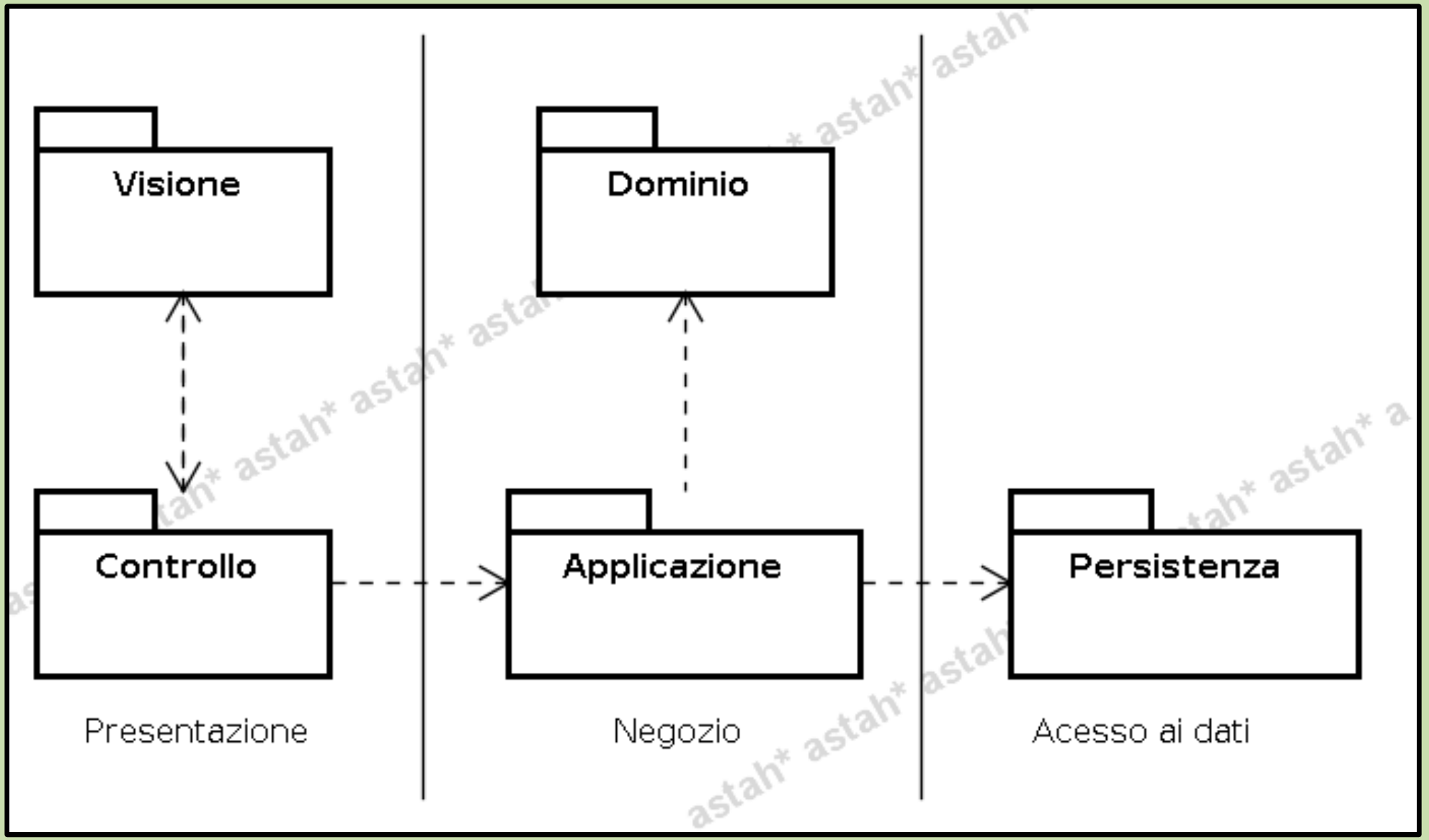
**NetBeans IDE Download Bundles**

| Supported technologies * | Java SE | JavaFX | Java | Ruby | C/C++ | PHP | All |
|---|---|---|---|---|---|---|---|
| NetBeans Platform SDK | ● | ● | ● | | | | ● |
| Java SE | ● | ● | ● | | | | ● |
| JavaFX | | ● | | | | | ● |
| Java Web and EE | | | ● | | | | ● |
| Java ME | | | ● | | | | ● |
| Java Card™ 3 Connected | | | — | | | | — |
| Ruby | | | | ● | | | ● |
| C/C++ | | | | | ● | | ● |
| Groovy | | | ● | | | | ● |
| PHP | | | | | | ● | ● |
| Bundled servers | | | | | | | |
| GlassFish Server Open Source Edition 3.0.1 | | | ● | ● | | | ● |
| Apache Tomcat 6.0.26 | | | ● | | | | ● |
| | Download | Download | Download | Download | Download | Download | Download |
| | Free, 54 MB | Free, 91 MB | Free, 173 MB | Free, 86 MB | Free, 36 MB | Free, 31 MB | Free, 239 MB |

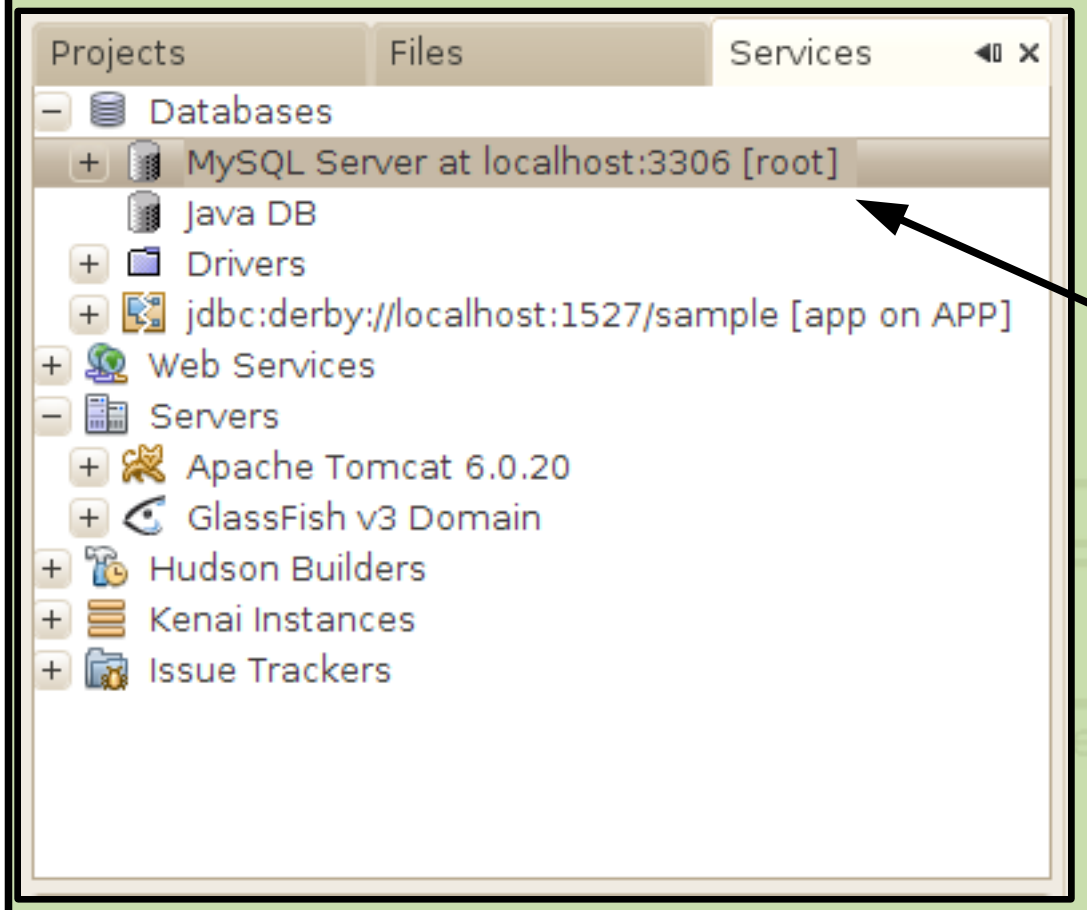
# Dimostrazione – una app completa

- Sistema per gestione di ambulanze (SGA):
  - Cittadini chiamano 118 in caso di emergenza;
  - Operatore registra la chiamata. Deve identificare quelle duplicate e filtrare situazioni banali;
  - Responsabile spedizione cerca l'ambulanza libera più vicina dal luogo dell'emergenza e ne richiede l'intervento;
  - Autista riceve l'ordine di intervento e deve guidare l'ambulanza fino all'emergenza. Posizione e status dell'ambulanza devono essere aggiornati nel sistema;
  - Funzionalità di infrastruttura (CRUD, login, ecc.).

# Architettura



# Base dati



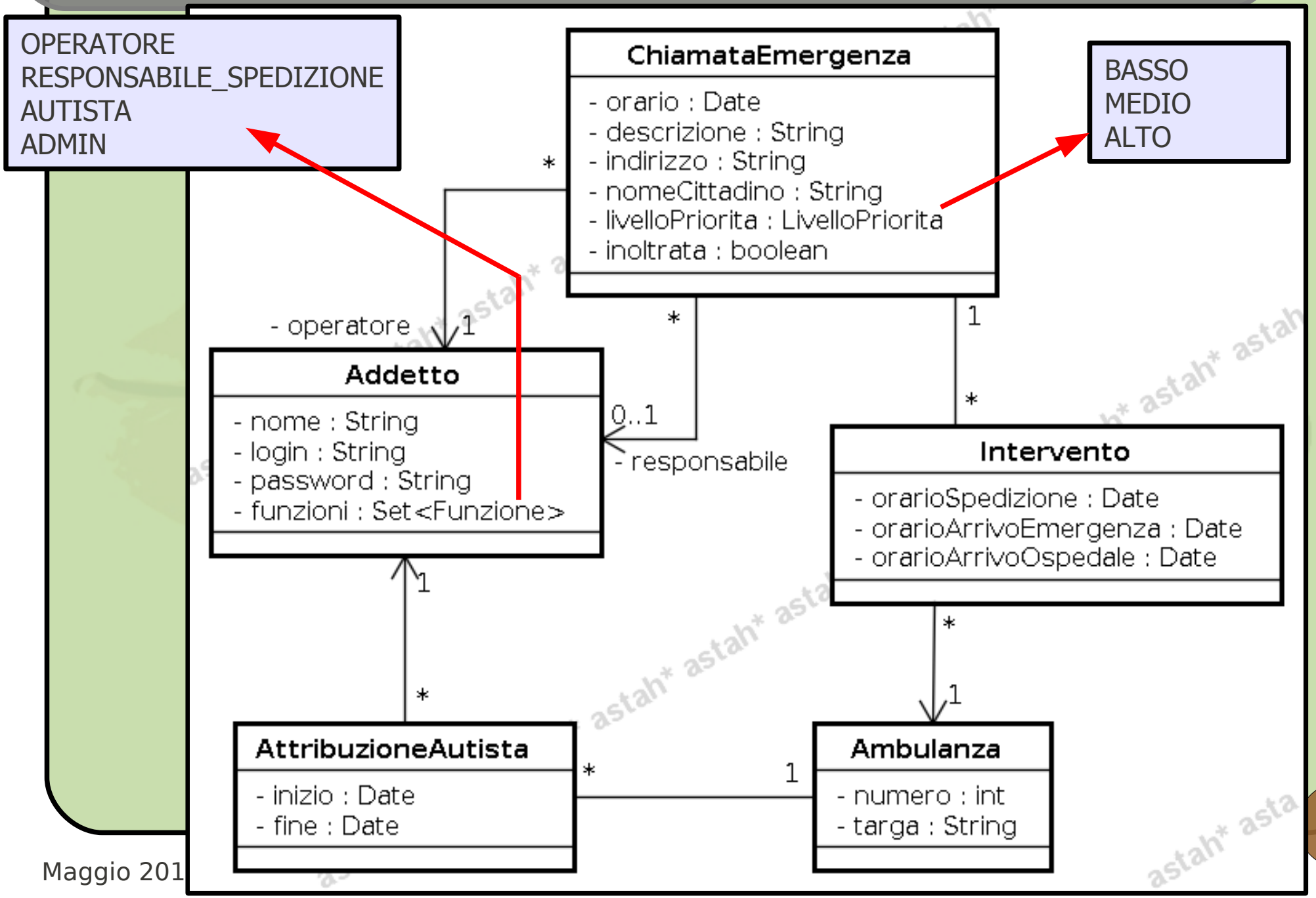

1 - Tasto destro > Start

2 - Tasto destro > Create Database...

Le tabelle saranno create da JPA.

Useremo MySQL nella dimostrazione.

# Modello di dominio



# Piccolo *framework* di persistenza (1)

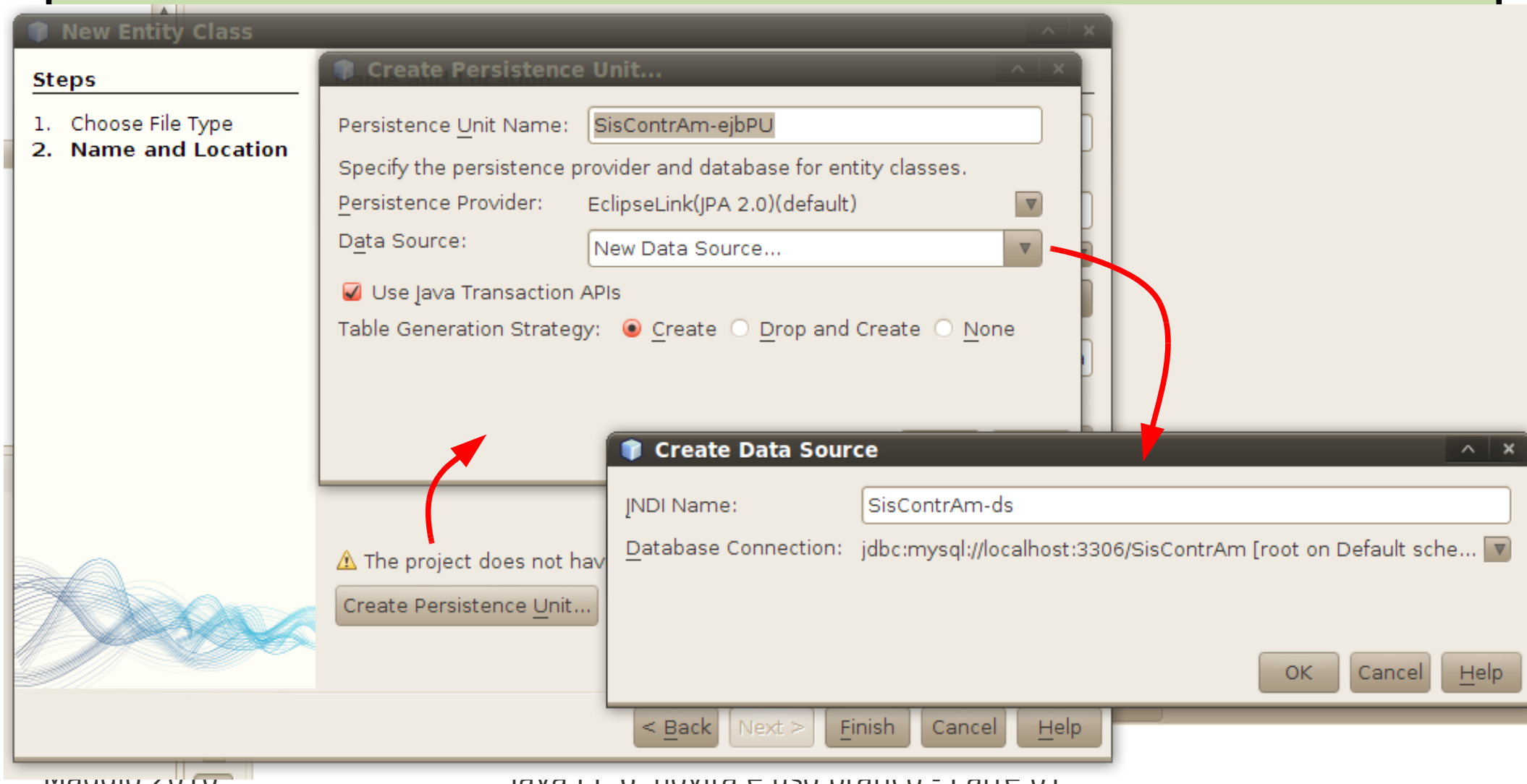
```
@MappedSuperclass
public abstract class OggetoPersistenteImpl
            implements OggetoPersistente, Serializable {
   private static final long serialVersionUID = 1L;

   @Basic @Column(nullable = false, length = 40)
   protected String uuid;

   @Id @GeneratedValue(strategy = GenerationType.AUTO)
   private Long id;

   @Version @Column(nullable = false)
   private Long version;

   public OggetoPersistenteImpl() {
      uuid = UUID.randomUUID().toString();
   }

   /* Metodi get/set, equals(), hashCode(), toString() */
}
```

# Dominio: POJOs e JPA

- Hanno sostituito Entity Beans in Java EE 5;
- Oggetti semplici + annotazioni;
- Novità Java EE 6:
  - Mappatura di collezioni di oggetti *non-entity*;
  - Nuovi operatori JPQL: **`NULLIF`**, **`COALESCE`**, **`INDEX`**, **`TYPE`**, **`KEY`**, **`VALUE`**, **`ENTRY`**;
  - Criteria API;
  - Supporto a *pessimistic locking*;
  - Integrazione con Bean Validation.

# Implementando la 1ª classe dominio

- Clicca col tasto destro nel progetto EJB > *New* > *Entity Class...*



# Dimostrazione

Implementare le classi **`Addetto`** e **`ChiamataEmergenza`** utilizzando JPA. Le novità: **`orphanRemoval`** per l'associazione con **`Spedizione`** e **`@ElementCollection`** per l'attributo **`funzioni`**.

# Validazione – *Bean Validation*

- Validazione trasversale: dal *form* Web alla persistenza (base dati);
- Accentramento nello strato di dominio, ma senza perdere di vista il suo obiettivo: annotazioni;
- Basato su Hibernate Validator;

```
public class Ambulanza extends OggetoPersistenteImpl {
  @NotNull
  private int numero;

  @NotNull
  @Size(min = 8, max = 8)
  private String targa;

  /* ... */
}
```

# Alcune annotazioni di validazione

- `@AssertFalse`, `@AssertTrue` (per *boolean*);
- `@DecimalMax`, `@DecimalMin` (per numeri reali, ma funziona soltanto con `BigDecimal`);
- `@Max`, `@Min` (per numeri interi);
- `@Digits` (solo cifre, string OK, può specificare min/max di cifre delle parti intera e decimale);
- `@Future`, `@Past` (per date);
- `@Pattern` (espressione regolare).

# Regole personalizzate (1)

```
import javax.validation.*;

public class TargaValidator implements
                        ConstraintValidator<Targa, String> {
  public void initialize(Targa constraintAnnotation) { }

  public boolean isValid(String value,
                        ConstraintValidatorContext context) {
    if (value.length() != 8) return false;
    boolean ascendente = true;
    int precedente = Character.digit(value.charAt(4), 10);
    for (int i = 5; ascendente && i < 8; i++) {
      int corrente = Character.digit(value.charAt(i), 10);
      ascendente = corrente > precedente;
      precedente = corrente;
    }
    return ascendente;
  }
}
```

# Regole personalizzate (2)

```
import static java.lang.annotation.ElementType.*;
import java.lang.annotation.*;
import javax.validation.*;
import javax.validation.constraints.*;

@NotNull
@Pattern(regexp = "^[A-Z]{3} [0-9]{4}$")
@Constraint(validatedBy = TargaValidator.class)
@Documented
@Target({ANNOTATION_TYPE, METHOD, FIELD})
@Retention(RetentionPolicy.RUNTIME)
public @interface Targa {
  String message() default "Targa non valida";
  String[] groups() default {};
  Class<? extends Payload>[] payload() default {};
}
```

```
public class Ambulanza extends OggetoPersistenteImpl {
  @Targa
  private String targa;

  /* ... */
```

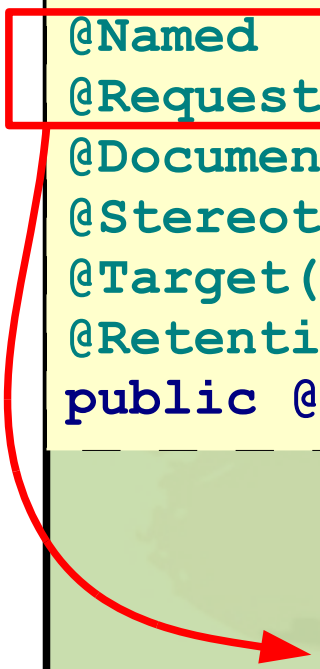
# Contesti e iniezione di dipendenze

- *Contexts and Dependency Injection for the Java EE Platform* (CDI) – JSR 299;
- Annotazioni per definire il contesto:
  - `@ApplicationScoped`, `@ConversationScoped`, `@SessionScoped`, `@RequestScoped`, `@Dependent`;
- Annotazioni per iniettare componenti:
  - `@PersistenceContext`, `@EJB`, `@Resource`, `@Inject`, etc.
- Per accederli nelle pagine JSF: `@Named`;
- Il *container* gestisce tutto (anche senza *setter*)!

# Stereotipi

- Combinazione di annotazioni multiple:

```
@Named
@RequestScoped
@Documented
@Stereotype
@Target(value={TYPE,METHOD,FIELD})
@Retention(value=RUNTIME)
public @interface Model
```

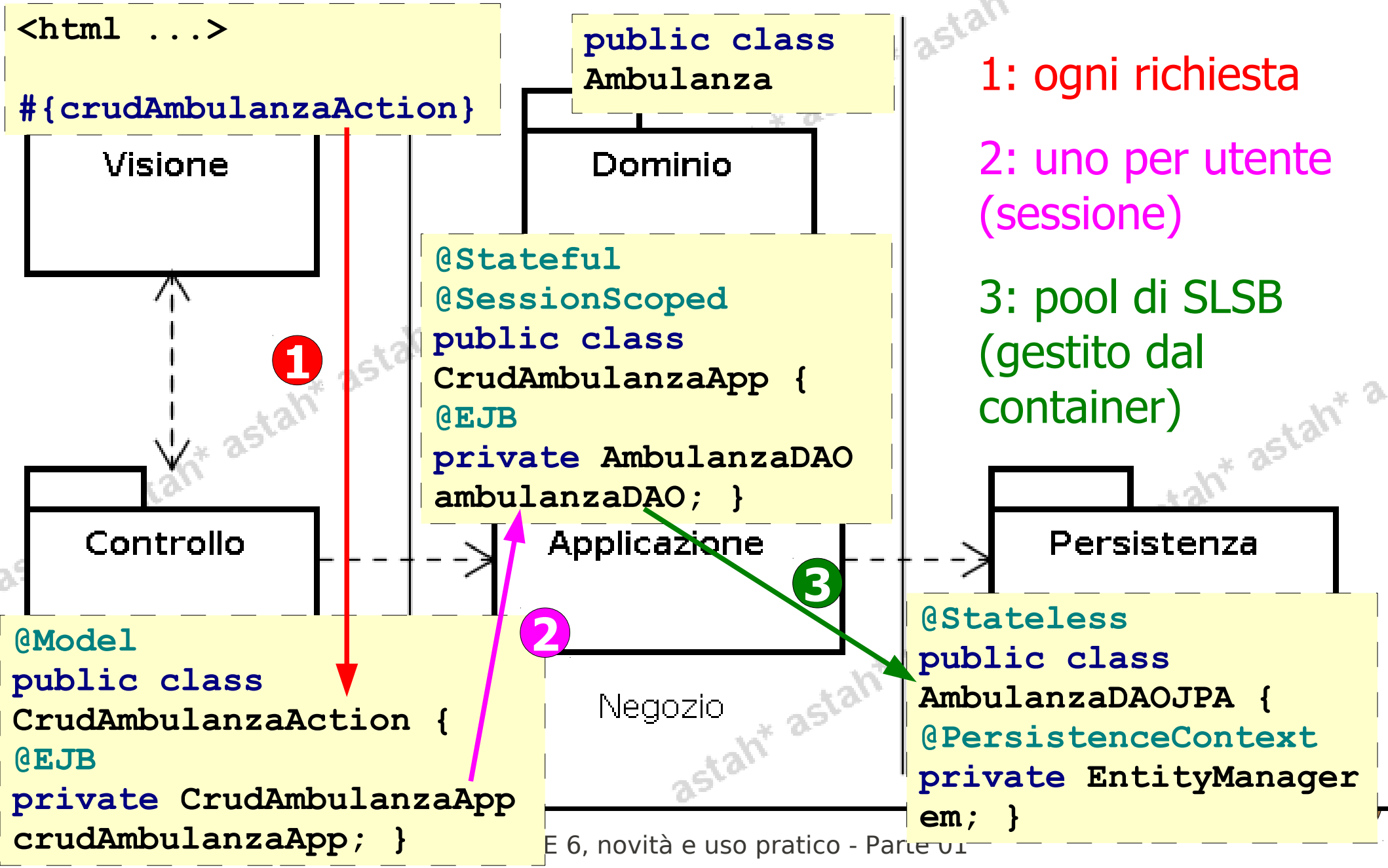


@Model = @Named + @RequestScoped

# JSF *Managed Beans*

- Prima c'era la necessità di dichiararli nel `faces-config.xml`;
- Adesso, basta l'annotazione `@Named`;
- Annotazioni ambigue di JSF:
  - `@ManagedBean`, `@RequestScoped`, `@...Scoped`;
  - JSF accetta anche quelle di CDI, quindi le usiamo ovunque;
- È possibile anche accedere ad un EJB direttamente dalla pagina JSF (meglio?).

# Esempio: un CRUD di Ambulanza




1: ogni richiesta

2: uno per utente (sessione)

3: pool di SLSB (gestito dal container)

# Piccolo *framework* di persistenza (2)

- Praticamente generato da NetBeans:
  - *File > New File... > Java EE > Session Beans for Entity Classes*;

```
public abstract class DAOBaseJPA2<T extends
    OggetoPersistente> implements DAOBase<T>, Serializable {

  protected abstract EntityManager getEntityManager();
  protected abstract Class<T> getClasseDominio();

  public long recuperareConteggio() { /* ... */ }
  public List<T> recuperareTutti() { /* ... */ }
  public List<T> recuperareAlcuni(int[] raggio) {/* ... */}
  public T recuperarePerId(Long id) { /* ... */ }
  public void salvare(T objeto) { /* ... */ }
  public void cancellare(T objeto) { /* ... */ }
}
```

# Dimostrazione

Implementare il CRUD di ambulanze, facendo vedere l'iniezione di dipendenza e contesti con CDI, oltre che la validazione della targa dell'ambulanza.

# Conclusioni

- Fine parte 1. Abbiamo visto:
  - Sviluppare con Java EE 6 è più semplice delle versioni precedenti: meno XML, più annotazioni;
  - *Bean Validation* garantisce l'integrità dei dati;
  - JPA 2 migliorata, ma non abbiamo visto tutto;
- Nella parte 2:
  - Facelets sostituisce JSP come standard per JSF;
  - L'API di Criteri (JPA 2) come alternativa alla JPQL;
  - Gestione delle conversazioni;
  - Supporto AJAX.


# Java EE 6
# Le novità e uso pratico

# Parte 1