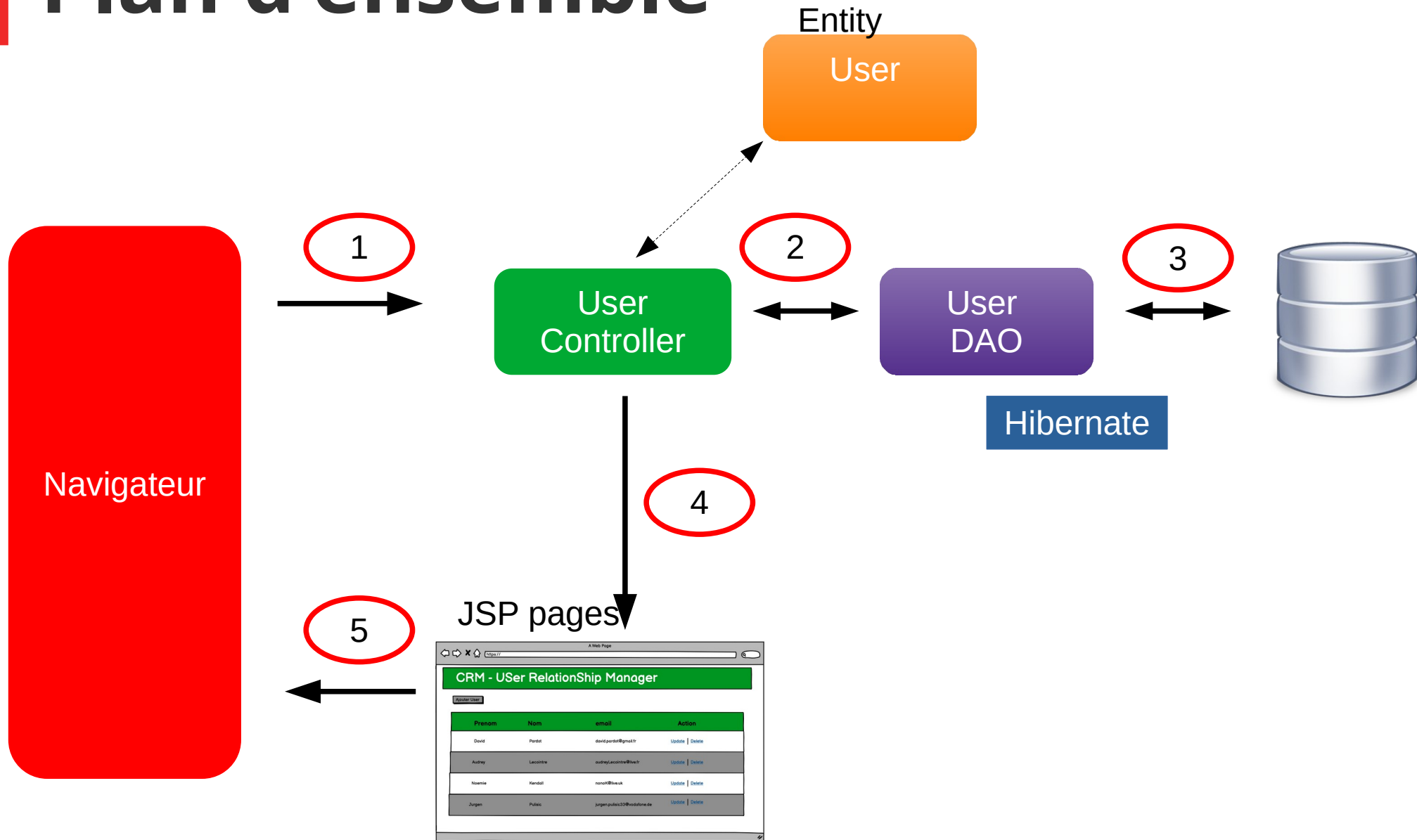


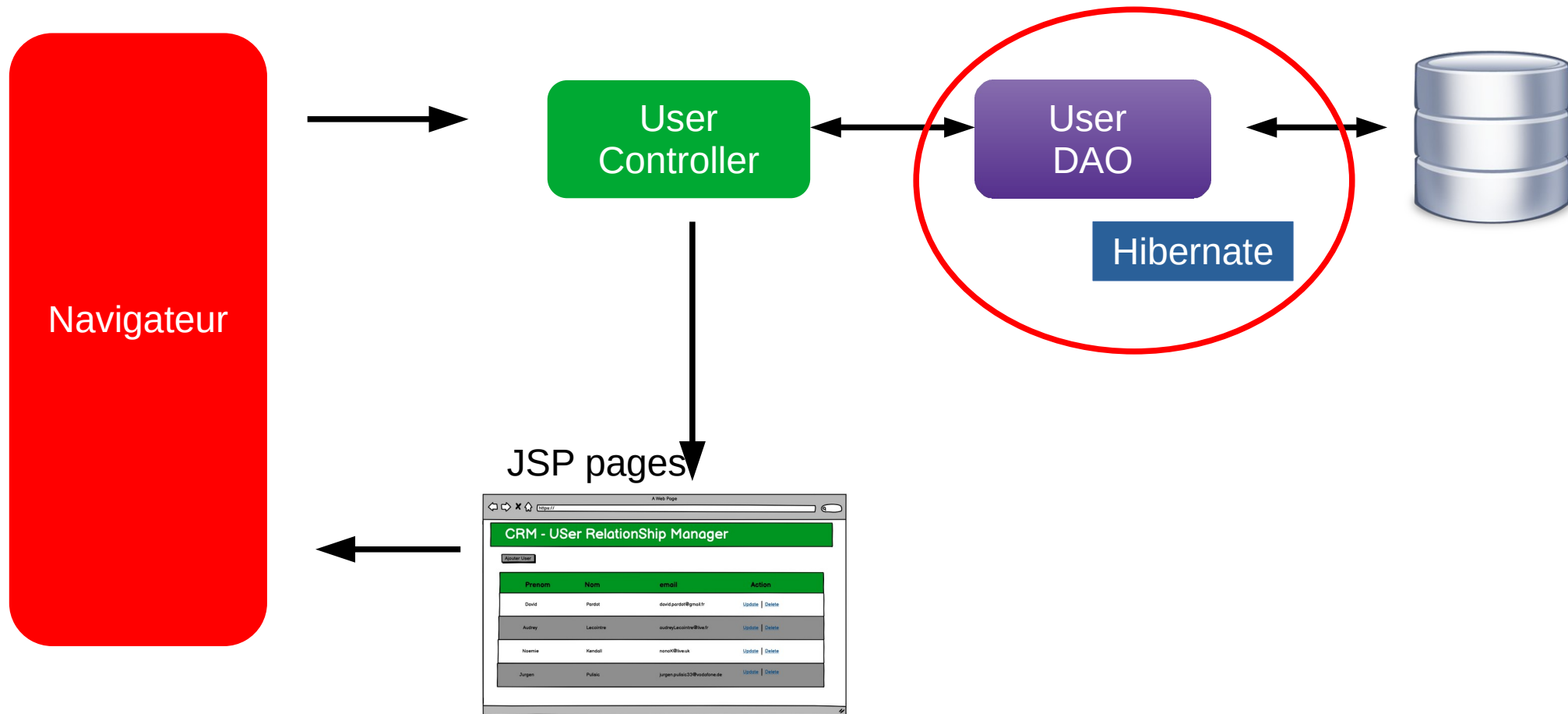


DAO

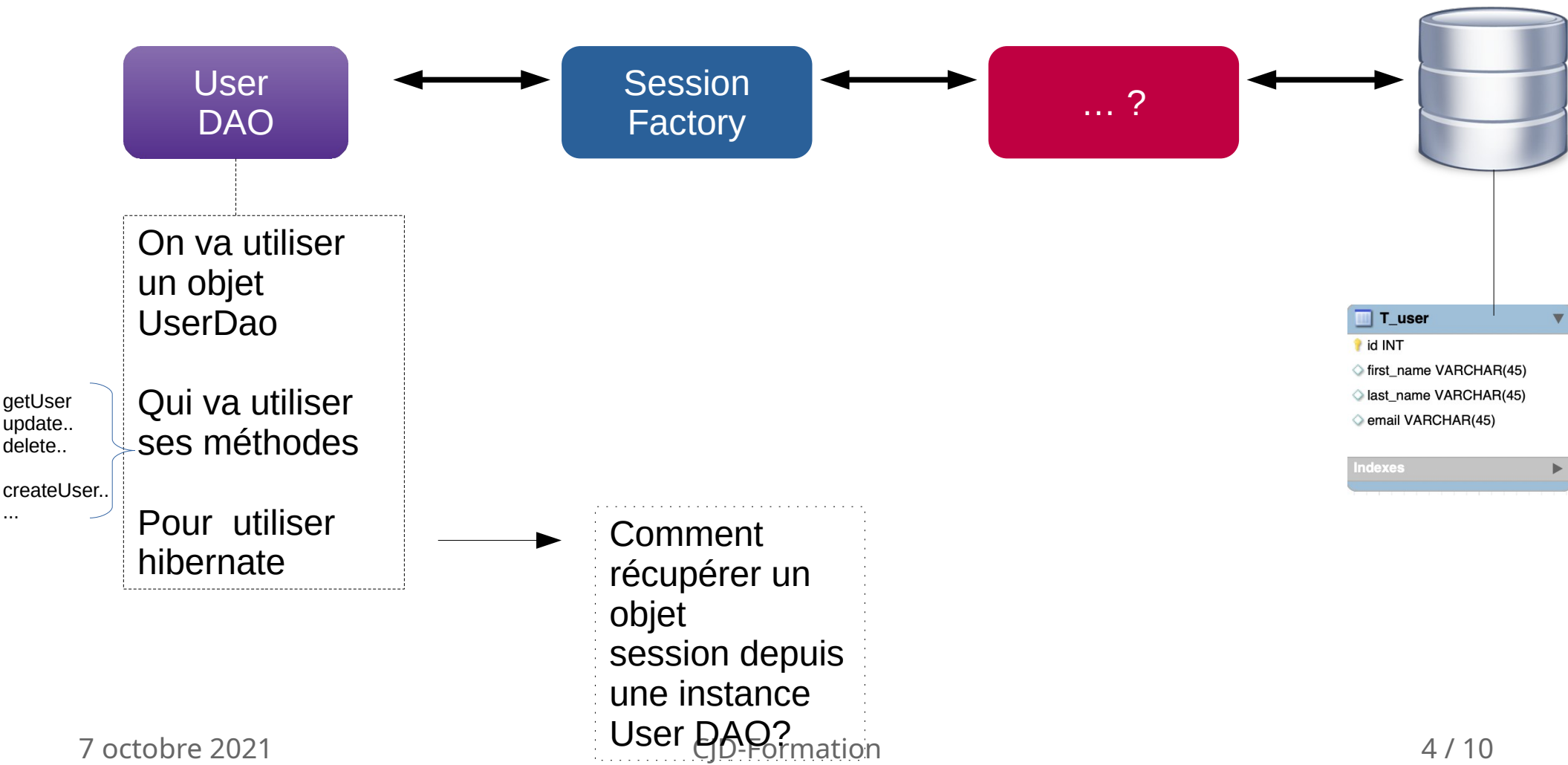
Plan d'ensemble



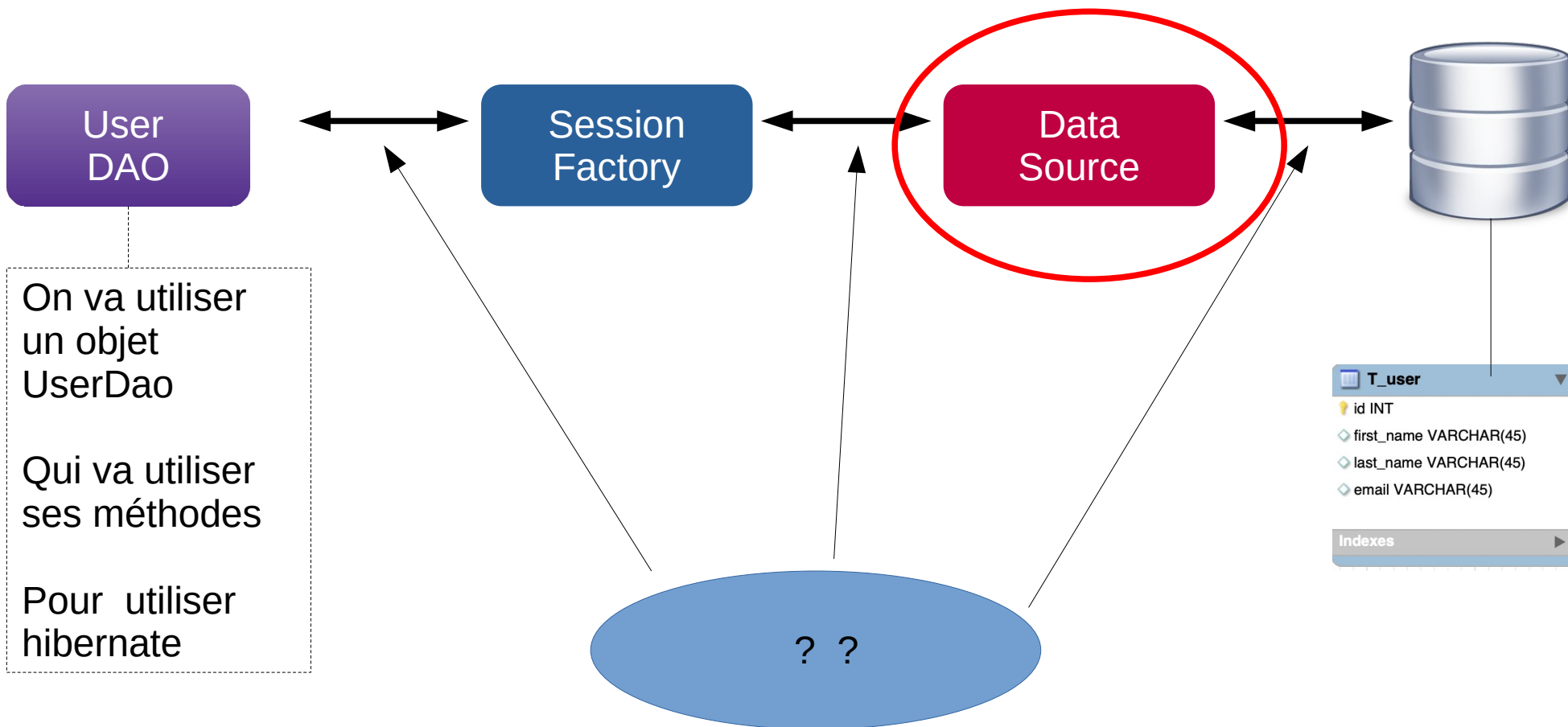
Plan d'ensemble



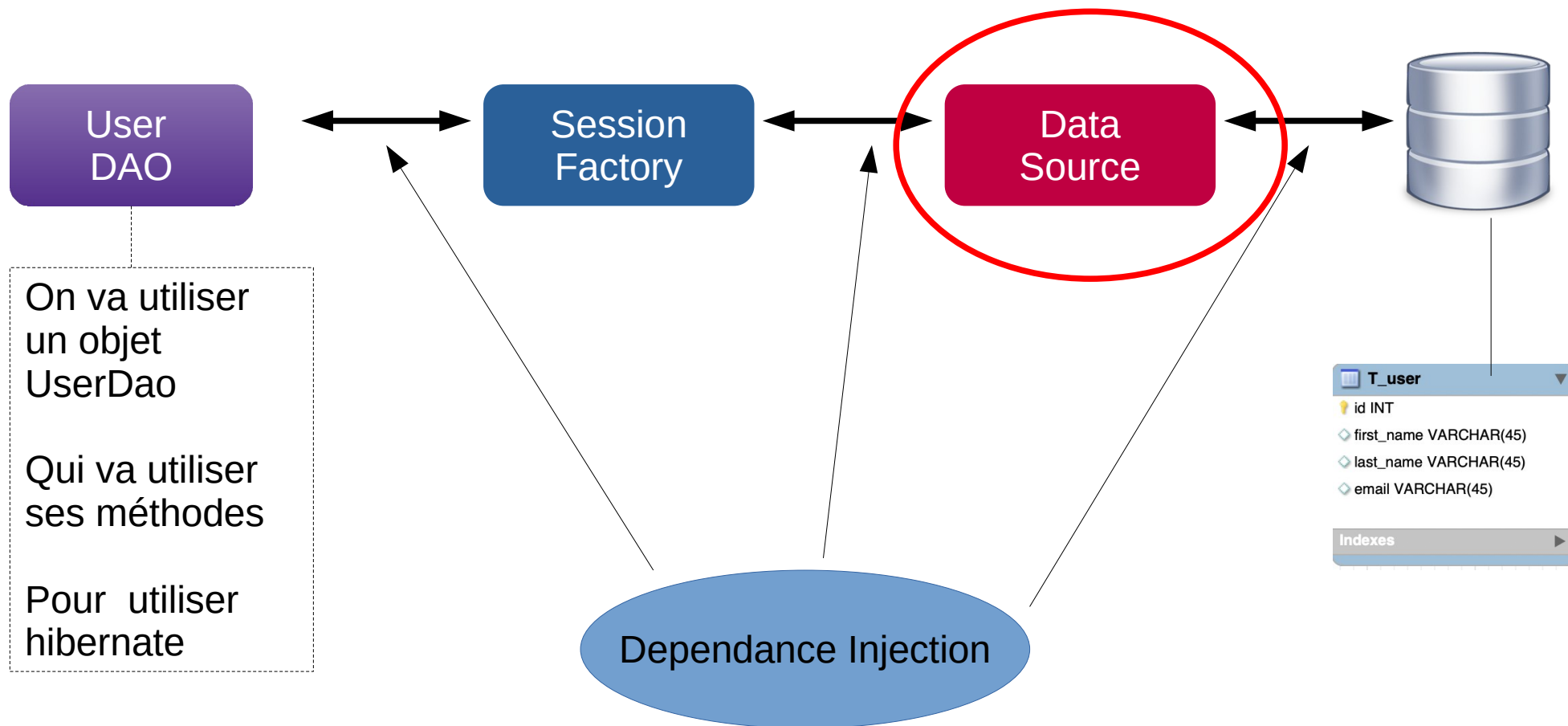
User Data access Object



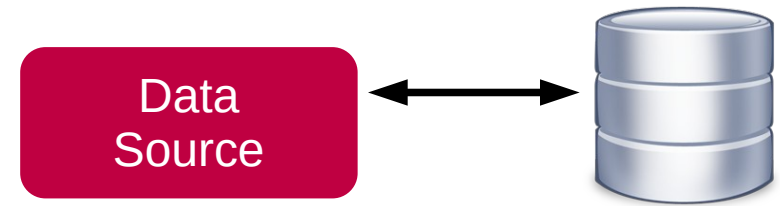
User Data access Object



User Data access Object



Data Source



- Le data source contient les informations pour se connecter à la bdd

```
<!-- Step 1: Define Database DataSource / connection pool -->
<bean id="myDataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">

    <property name="driverClass" value="com.mysql.cj.jdbc.Driver" />
    <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/web_user_manager?
useSSL=false&serverTimezone=UTC" />
    <property name="user" value="my-app-account" />
    <property name="password" value="my-app-pwd" />

    <!-- these are connection pool properties for C3P0 -->
    <property name="minPoolSize" value="3" />
    <property name="maxPoolSize" value="20" />
    <property name="maxIdleTime" value="30000" />
</bean>
```

Session Factory



- La session Factory dépend du data source

```
<!-- Step 2: Setup Hibernate session factory -->
<bean id="sessionFactory"
      class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">

    <property name="dataSource" ref="myDataSource" />

    . . .

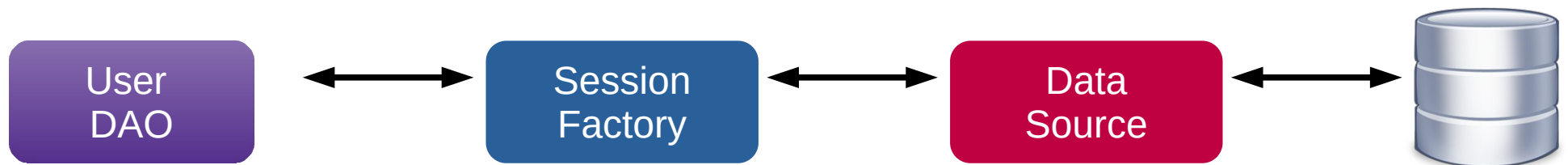
</bean>
```

DI avec xml

User DAO

Définir une Interface DAO

Définir une implémentation DAO
Injecter la session factory



Définir une interface DAO

```
package com.tld.dao;

import java.util.List;
import com.tld.entity.User;

public interface UserDao {

    public List<User> getUsers();

}
```

Définir une implémentation de cette interface

@Autowired pour
l'injection de dépendances

```
public class UserDaoImpl implements UserDao{

    @Autowired
    private SessionFactory sessionFactory;

    @Override
    public List<User> getUsers() {
        ...
    }

}
```