HIBERNATE

ORM JAVA

INTRODUCTION

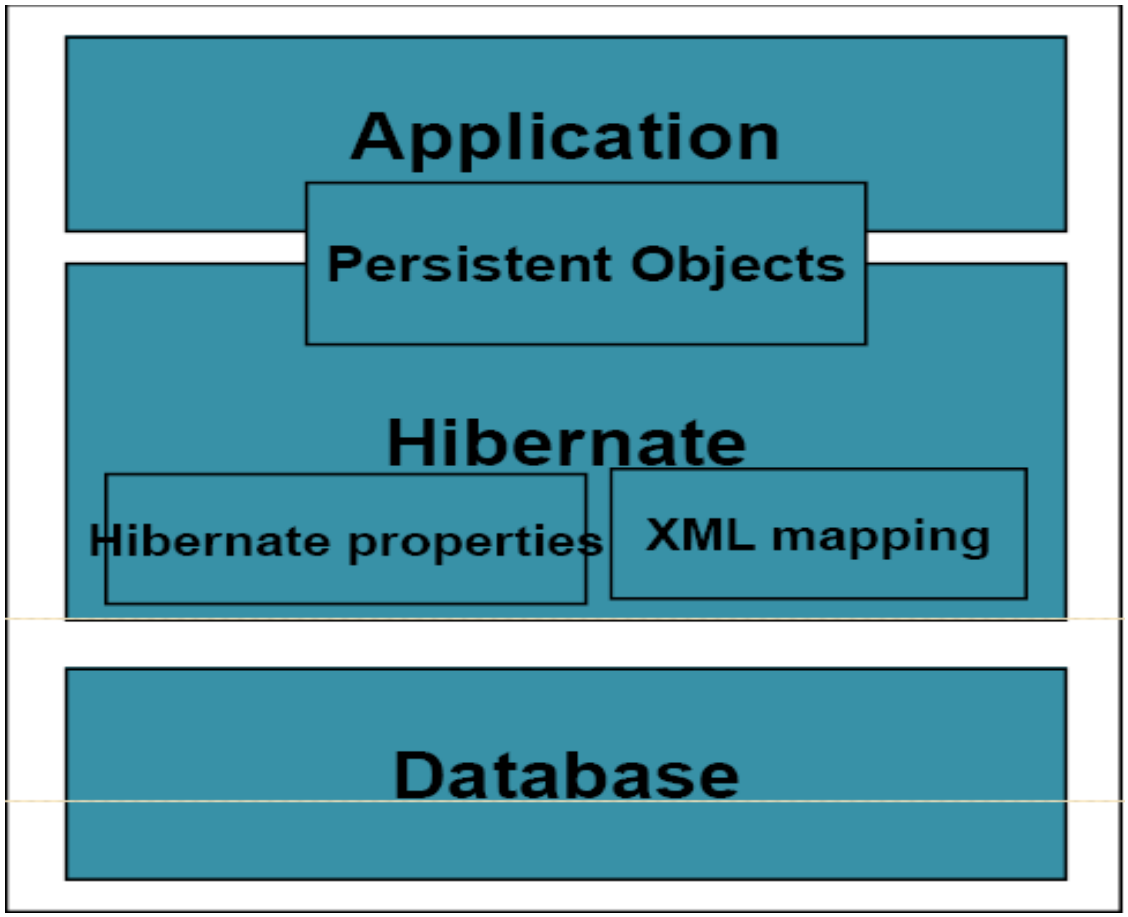
HIBERNATE est un projet open source visant à proposer un outil de **MAPPING** entre les objets et des données stockées dans une base de données relationnelle.

Ce projet ne repose sur aucun standard mais il est très populaire notamment à cause de ses bonnes performances et de son ouverture avec de nombreuses bases de données

Les bases de données supportées sont les principale du marché: **DB2**, **Oracle**, **MYSQL**, **PostgeSQL**, **Sybase**, **SQL Server**

.....

ARCHITECTURE

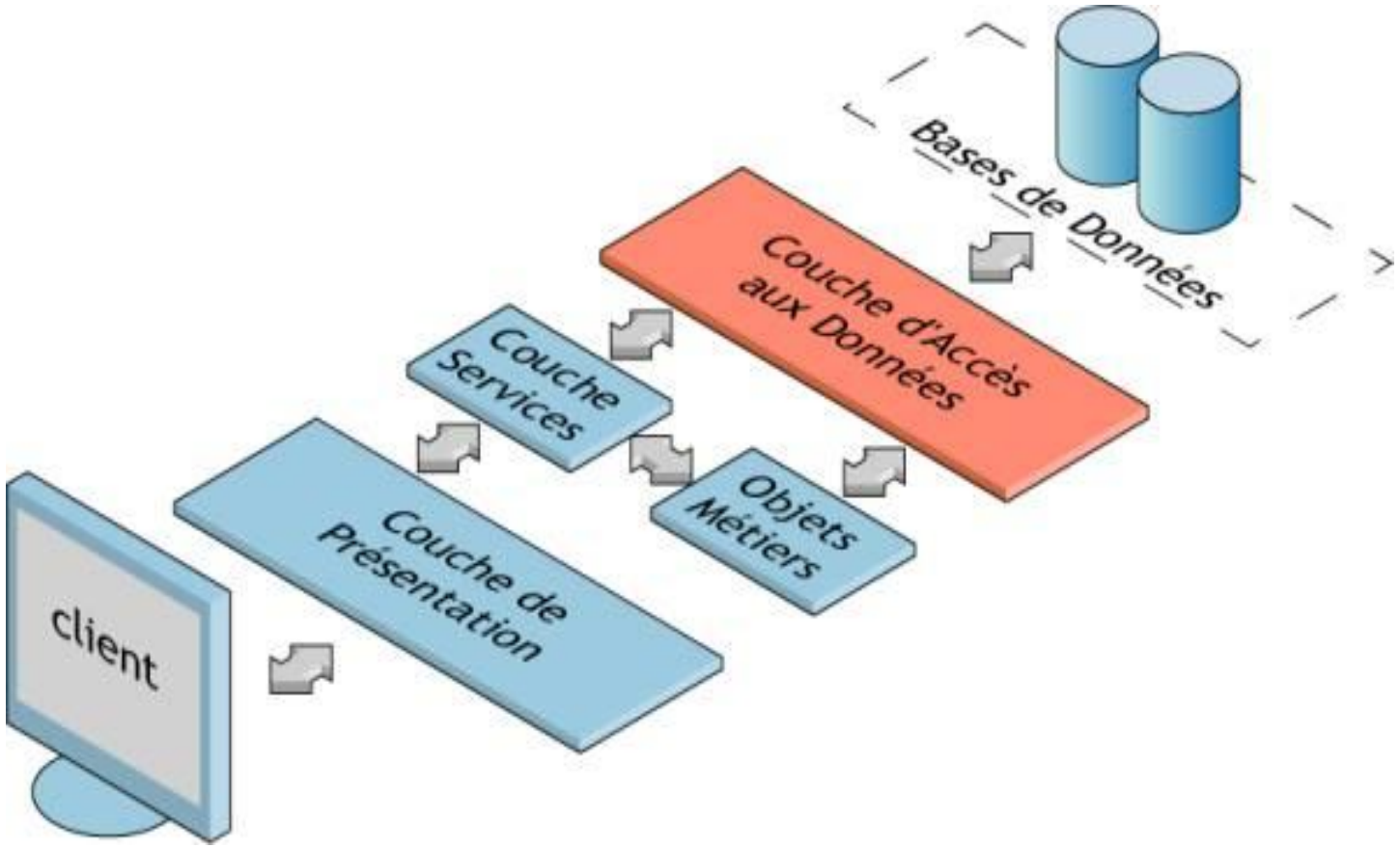


ELEMENTS HEBERNATE

Hibernate a besoin de plusieurs éléments pour fonctionner:

- Une classe de type JAVABEAN qui encapsule les données d'une occurrence d'une table.
- un fichier de correspondance qui configure la correspondance entre la classe et la table ou des annotations.
- Des propriétés de configuration notamment des informations concernant la connexion à la base

CONTEXTE : CRÉER UNE COUCHE D'ACCÈS DONNÉES (DAO)



DÉFINITION & HISTORIQUE

- Outil Mapping Objet / Relationnel en Java (ORM)
- Né suite à complexité EJB entity (EJB1.x, EJB2.x)
- Juillet 2000 - Gavin King
- JSE et JEE
- **www.hibernate.org**
- Inclus dans JBOSS (Red Hat)
- Standardisé par spécifications JPA / EJB3 Entity

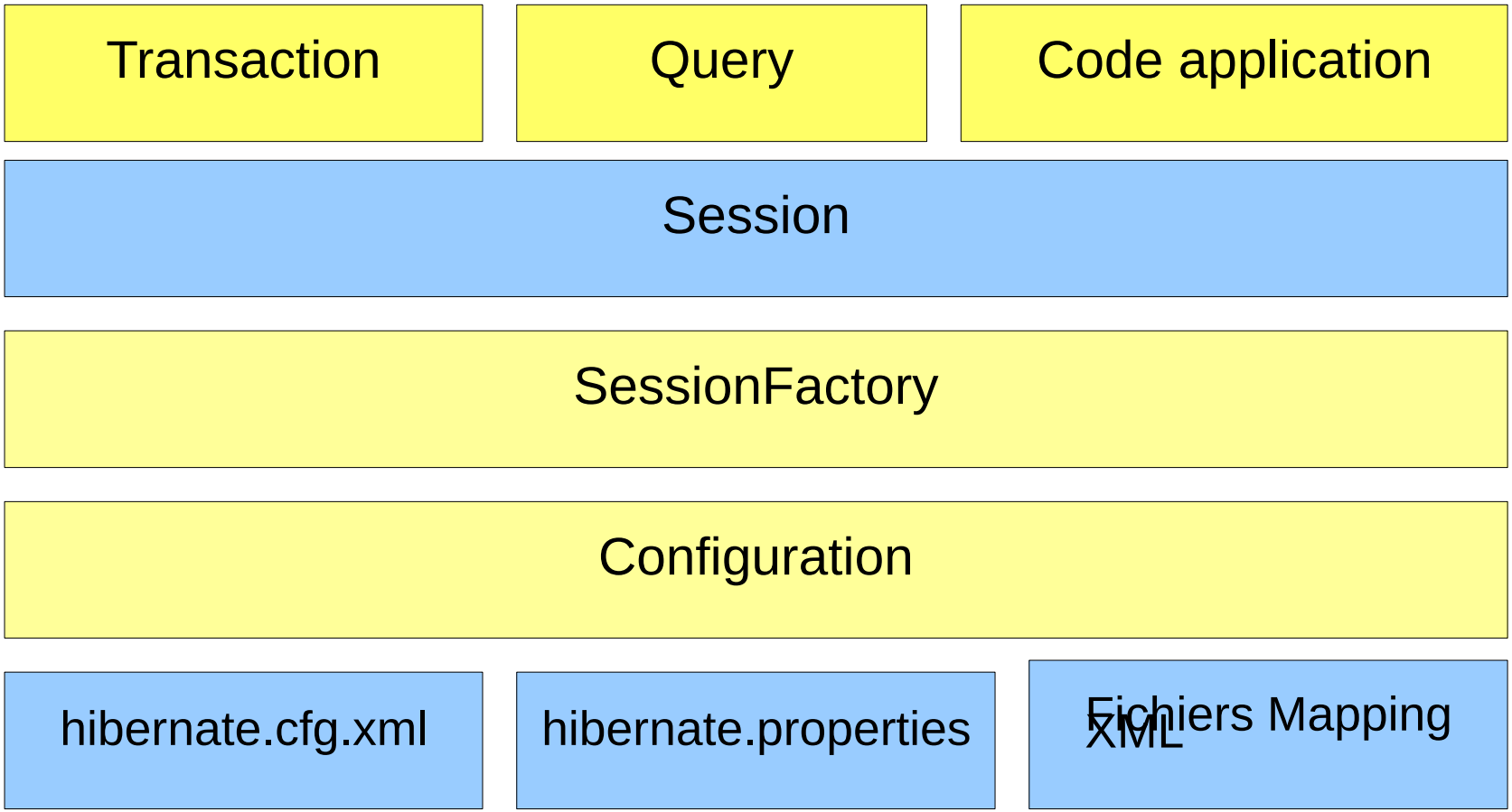
OUTIL DE MAPPING OBJET / RELATIONNEL

- **Service**
  - **Synchronisation** propriété objet / champ table
    - propriété 'theme' de l'objet Formation
    - champ 'theme' table FORMATIONS
- **Avantages**
  - **Améliore portabilité du** code si changement DB
  - **Gain de 30 à 40 %** nb lignes de certains projets
  - **Le développeur pense en termes d'objet**

FICHIER DE MAPPING

- Fichier XML Décrit comment se fera la persistance des objets d'une classe en DB
- Format XML (ou annotations si jdk1.5 +)
- Se place dans le même répertoire que la classe et se nomme **Classe.hbm.xml** si la classe s'appelle Classe.

ARCHITECTURE HIBERNATE : ENVIRONNEMENT JAVA

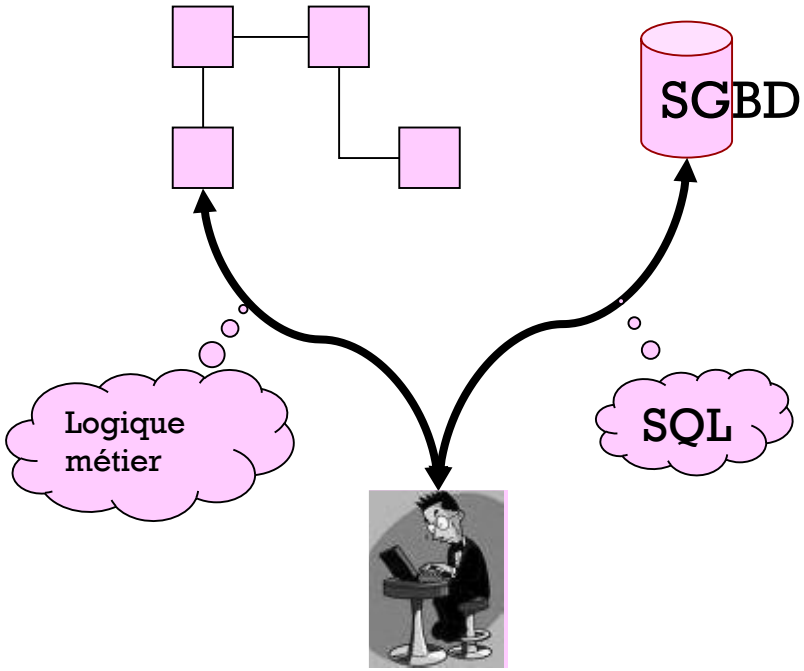


CONCURRENTS

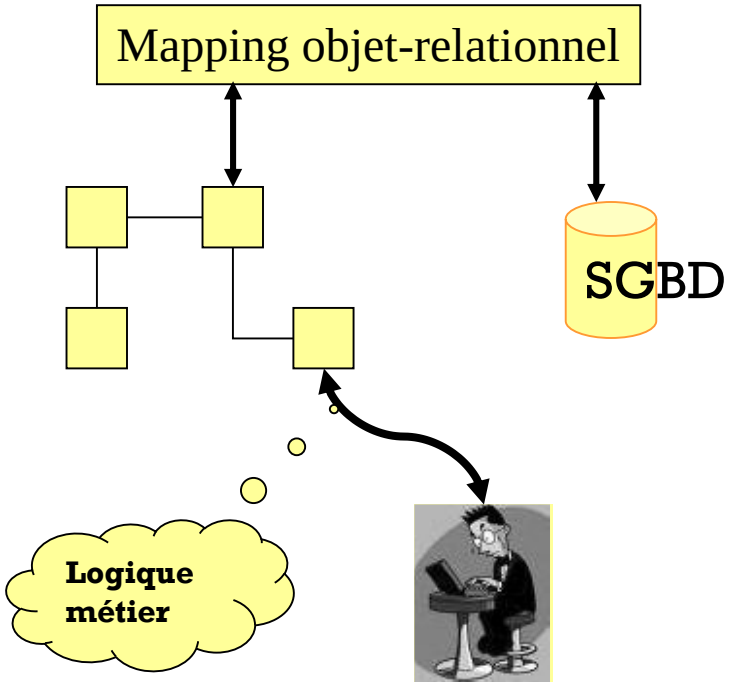
- **JDBC**
- **EJB Entity**
- **TopLink**
- **OpenJPA (http://openjpa.apache.org/)**
- **Ibatis (http://ibatis.apache.org/)**
- **Castor/JDO**
- **...**

TRANSPARENCE DE LA PERSISTENCE

Sans Hibernate



Avec Hibernate



L'ÉQUATION HIBERNATE

JavaBeans

\+ SGBDR

\+ Données de mapping et de configuration

= Persistence de données

FICHE TECHNIQUE HIBERNATE

Informations

Développé par	Red Hat
Première version	23 mai 2001
Dernière version	5.4.4 (30 juillet 2019)
Version avancée	6.0.0.Alpha2 (4 avril 2019)
Dépôt	github.com/hibernate/hibernate-orm
Écrit en	Java
Environnement	Multiplate-forme (JVM)
Langues	anglais
Type	Mapping objet-relationnel
Licence	Licence publique générale limitée GNU
Site web	hibernate.org

LES TRANSACTION

- Une transaction est un ensemble d'opérations qui doivent être exécutées en tant qu'unité.
- Ces opérations peuvent être synchrone ou asynchrone, et peuvent impliquer la persistance de données, l'envoi de messages, la validation des cartes de crédit, etc
- Un exemple classique est l'opération de transfert de compte à compte bancaire : on enlève sur un compte,on dépose sur l'autre…
  - Si l'une des deux opérations échoue, perte de cohérence !
  - On veut que soit les deux opérations réussissent, mais si une d'elles échoue, on annule le tout et on remet les comptes dans l'état initial !
  - On dit que les deux opérations forment une seule et même *transaction* !

ATELIER N° 1

- Sous un projet java standard ou Maven.
- À l'aide de fichier de mapping.
- Test: create – validate - update

ATELIER N° 2

- Les fichiers de configuration

ATELIER N° 3

- Générer les beans à partir des tables.