

# CONCEPTION ET DEVELOPPEMENT D'APPLICATIONS INTERNET

## CDAI 1 - BIS - JAVAEE

Université Paris Dauphine

Master M2 MIAGE

Année 2014-2015

Bekhouché Abdesslem

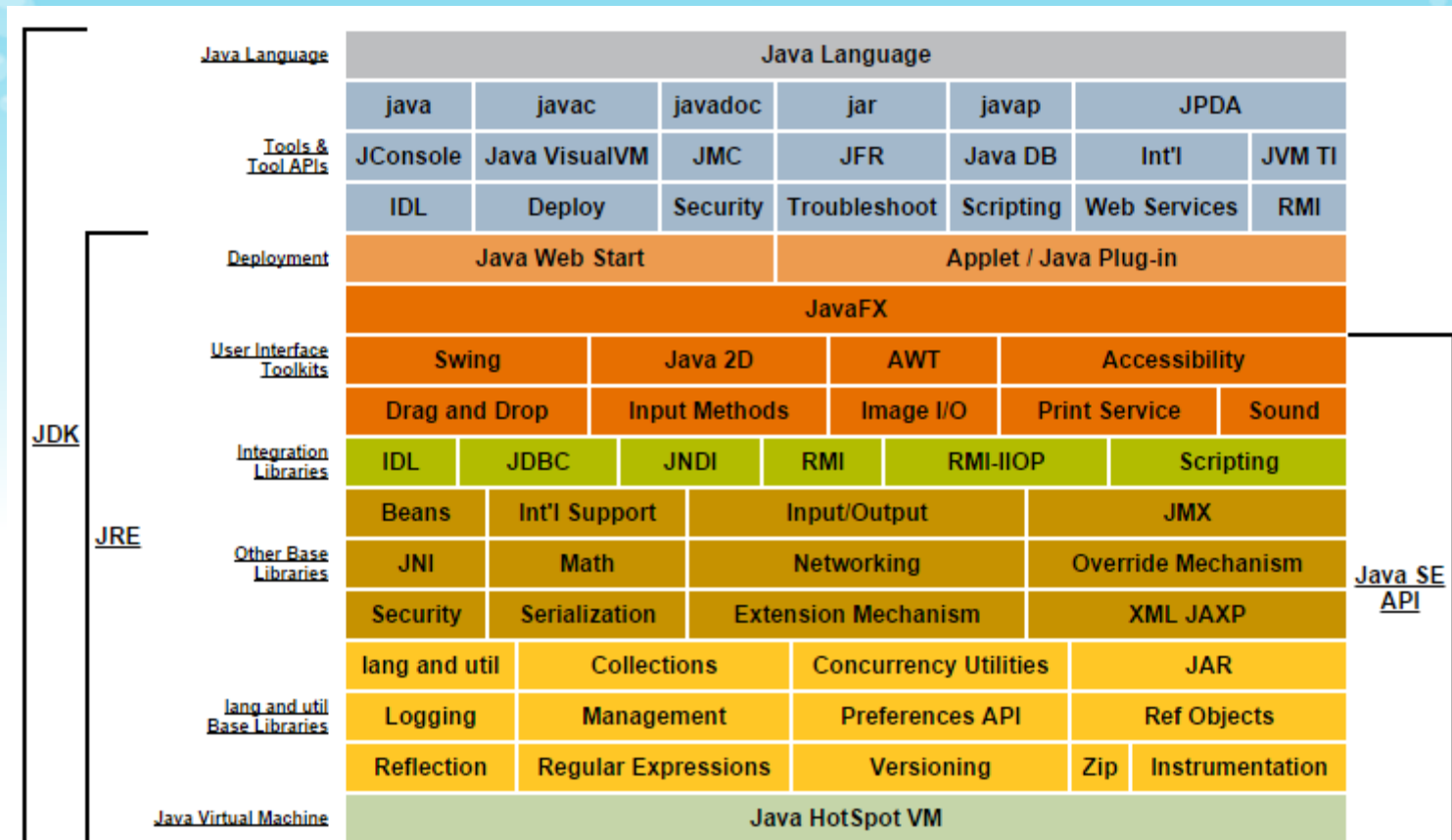
Sobral Diogo



# PLAN

1. INTRODUCTION/DÉFINITIONS
2. PLATEFORMES EXISTANTES
3. TECHNOLOGIES MIDDLEWARE (API)
4. EVOLUTION
5. ARCHITECTURE N-TIERS
6. SERVEUR D'APPLICATIONS
7. AVANTAGES COMPOSANTS
8. WEBOGRAPHIE

# 1. DEFINITIONS : JRE, JDK, SDK



<http://docs.oracle.com/javase/7/docs/index.html>

# 1. INTRODUCTION

## JAVA EE: JAVA PLATFORM ENTREPRISE EDITION

Ensemble de concepts pour le développement d'applications réparties:

- défini par Sun Microsystems (racheté par Oracle en 2009);
- basé sur le langage Java;
- composé par un ensemble de spécifications:
  - JSR: Java Specifications Requests
  - liste <http://www.oracle.com/technetwork/java/javae/ttech/index.html>
- en évolution "permanente" depuis 1996/97
  - J2EE 1.0 (servlet + EJB + JDBC), ... , 1.3, 1.4,
  - Java EE 5 2006
  - Java EE 6 (JSR 318) 2009
  - Java EE 7 (JSR 342) 2013
- domaines applicatifs visés
  - E commerce (B2B & B2C)
  - systèmes d'informations
  - sites web
  - plates-formes de service (Audio-visuel, telco, ...)

## 2. PLATEFORMES EXISTANTES

- Implémentation de référence : Java EE 7 SDK GlassFish v4.1
- Commerciales:
  - WebSphere (IBM), WebLogic (Oracle), NEC, SAP, Fujitsu Interstage Application Server, TmaxSoft, ...
  - VOIR <http://www.oracle.com/technetwork/java/javaee/overview/compatibility-jsp-136984.html>  
et <http://www.oracle.com/technetwork/java/javaee/licensees-138280.html>
- Open source:
  - JBoss, JOnAS, Geronimo, OpenEJB, JFox, ...
- Processus de certification mis en place par Sun:
  - TCK (Test Compatibility Kit);
  - payant sauf pour plates-formes open-source;
  - assez lourd (~20 000 tests) à mettre en oeuvre.

# 3. TECHNOLOGIES MIDDLEWARE 1/2

- Un ensemble de technologies middleware pour la construction d applications réparties
- Communications distantes:
  - RMI IIOP: requête/réponse (TCP + IIOP + sérialisation Java)
  - JAX: requête/réponse Web Service (SOAP, REST)
  - JMS: MOM (message oriented middleware) : message + boîte à lettres
- Services systèmes:
  - JNDI: annuaire
  - JTA: gestion de transactions
  - JPA: gestion de la persistance des données
  - JASS: sécurité, contrôle d'accès (rôles, droits)
  - JMX: administration de la plate-forme

# 3. TECHNOLOGIES MIDDLEWARE 2/2

- Page Web dynamiques : JSP, servlet, JSF
- Connecteurs (JCA) pour interactions avec serveurs tiers:
  - JDBC: accès client/serveur aux SGBD
  - JavaMail : envoi/réception de mail

# 4. EVOLUTION 1/3

Augmentation permanente du nombre d'API:

## Java EE Past, Present, & Future





## 4. EVOLUTION 2/3

Conséquence : de plus plus lourd

Solutions:

- élagage (pruning)
  - certaines spécifications deviennent optionnelles (CMP 2.x, JAX-RPC, JSR 88, etc.)
  - pourraient être retirées des versions futures;
- profile:
  - version allégée de la plate-forme avec un sous-ensemble de spécifications
  - 2 (pour l'instant) : Web Profile et Full Profile

# 4. EVOLUTION 3/3

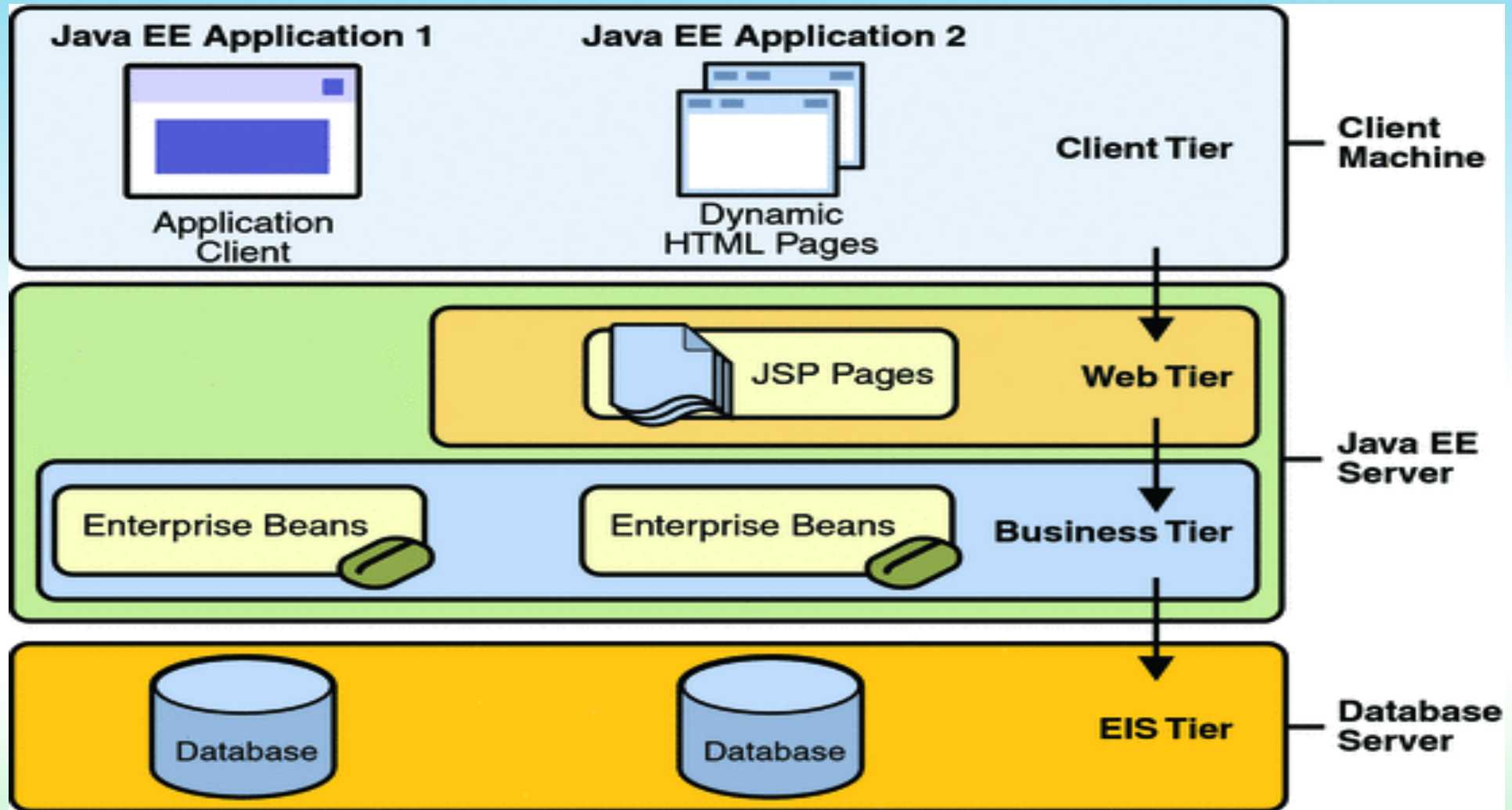
- Web Profile

- JSF, servlet, JSP, EL, JSTL, Managed Beans, Interceptors, JTA, JPA, etc.
- EJB Lite
  - local session bean, injection, CMT/BMT, security
  - non présent : MDB, EJB WS Endpoint, RMI-IIOP, EJB 2.x, Timer bean, CMP/BMP

# 5. ARCHITECTURE N-TIERS

- Client:
  - riche: application java
  - léger: navigateur web
- Serveur d'applications
  - hébergent des applications à base de composants
  - composants EJB: classes Java conformes au modèle EJB
  - composants Web: servlet ou JSP
- SGBD
  - fournit un support de stockage pour les données de l'application:
    - 80% : SGBDR (Oracle, SQL Server, PostgreSQL, ...)
    - 20% : autres applications de stockage (ex : NoSQL)

# 5. ARCHITECTURE N-TIERS : Schéma



# 6. SERVEUR D'APPLICATIONS

- Application JEE =
  - 0, 1 ou +sieurs composants EJB
  - 0, 1 ou +sieurs composants Web
- Plusieurs rôles dans le développement:
  - développeur de composants Web
  - développeur de composants EJB
  - assembleur d applications
  - déployeur et gestionnaire d'applications
- 4 services fournis par le serveur au conteneur EJB
  - persistance (JPA)
  - transaction (JTA)
  - nommage (JNDI)
  - sécurité (JASS)
- Middleware style CORBA: ces services sont intégrés dès le départ à la plate-forme

# 7. AVANTAGES COMPOSANTS

3 notions importantes des intergiciels basés composant:

- séparation des préoccupations
  - développer le métier indépendamment des préoccupations techniques
  - composants plus facilement réutilisables
- inversion du contrôle
  - conteneur prenant en charge l'exécution du code métier (composant)
  - conteneur assure lien avec la partie technique
  - configurer plutôt que programmer
- injection de dépendances
  - vers d'autres composants, vers des services techniques
  - retirer du code métier la gestion des liens vers les autres composants métiers
  - faire gérer l'architecture applicative par le conteneur

# 8. WEBOGRAPHIE

<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

<http://www2.lifl.fr/~seinturi/middleware/javaee.pdf>