

# Java EE 6

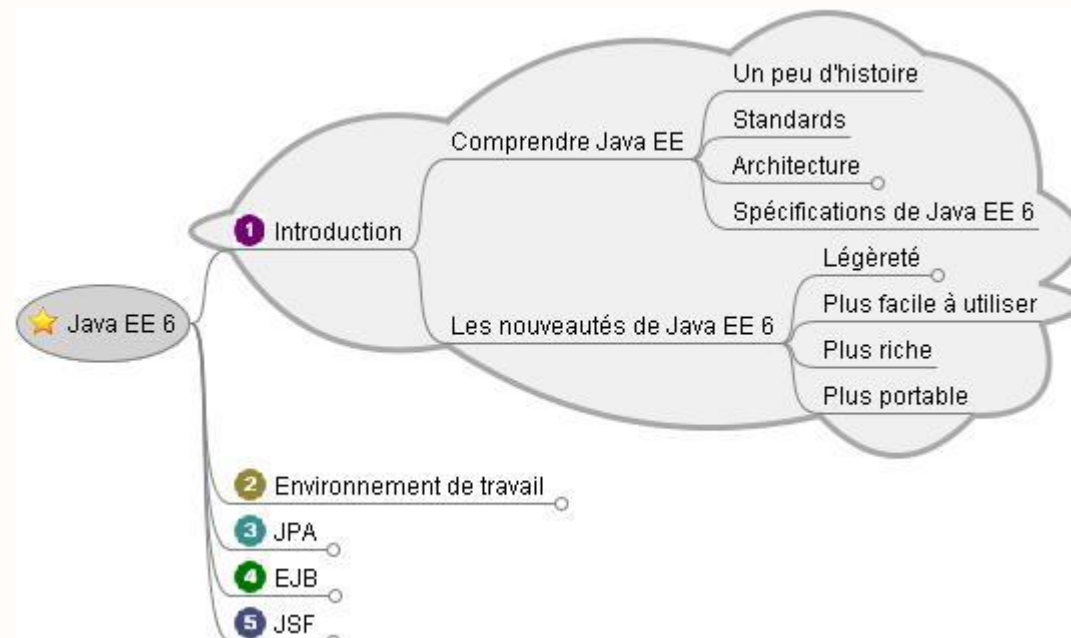
Développer des applications d'entreprise

- **Chapitre 1**
  - **Introduction**



# Agenda

Planning de la formation



# Introduction

## Pourquoi Java EE

- Les entreprises sont en compétition au niveau mondial,
- Elles doivent être disponibles 24h/24 via Internet et gérer différentes langues, devises et fuseaux horaires,
- Les temps de réponses doivent être rapides,
- Il faut garantir la sécurité et la sauvegarde des données, la réplication éventuellement,

# Introduction

## Pourquoi Java EE

- Il faut présenter les données via différents types d'interfaces – navigateur internet, téléphone, ...
- Il faut proposer des interfaces de saisie conviviales (clients riches),
- Il faut faire communiquer son système d'information avec celui de ses partenaires,
- Il faut faire communiquer toutes les technologies de son propre système d'information

# Introduction

## Pourquoi Java EE

- Il faut pouvoir communiquer dans différents protocoles,
- Il faut pouvoir monter en charge en fonction de la progression de la demande

C'est pour ça qu'a été créé  
**Java Enterprise Edition**

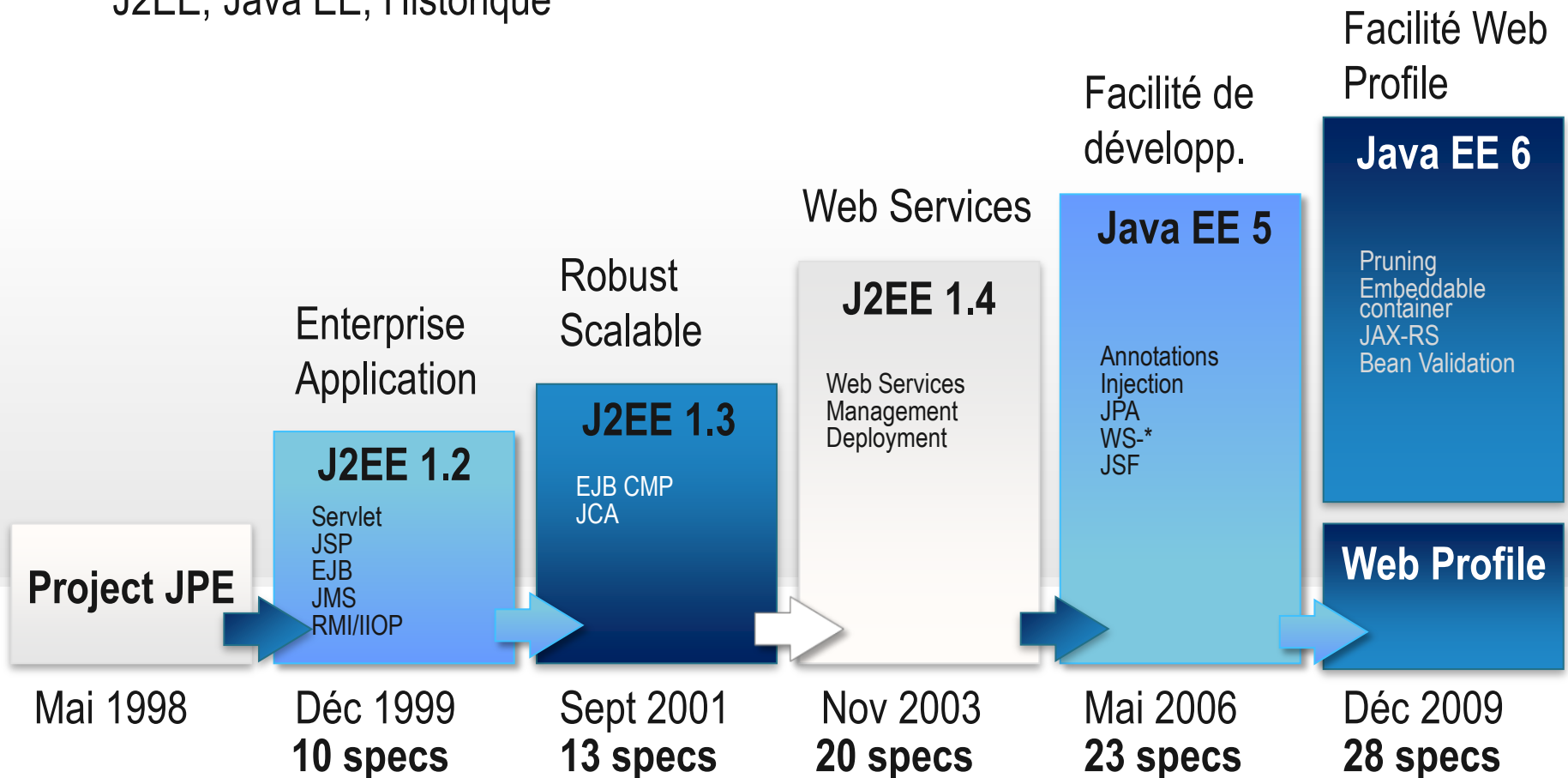
# Introduction

## Comprendre Java EE



# Introduction

J2EE, Java EE, Historique



# Introduction

## Etat de la plateforme

- Java EE 6 est une release importante qui marque le 10ème anniversaire de Java EE
- Elle s'appuie sur le passé et continue de simplifier le développement des applications d'entreprise, notamment grâce à l'expérience accumulée pendant ces 10 ans, mais aussi grâce au dynamisme de la communauté Open Source dont elle s'inspire et à la rigueur du processus JCP (Java Community Process)



# Introduction

## Etat de la plateforme

- Aujourd'hui :
  - Javav EE est une plateforme bien documentée, avec des développeurs expérimentés, une large communauté et de nombreuses applications déployées sur les serveurs d'entreprises,
  - Java EE est une suite d'API pour construire des applications multi-niveaux distribuées à base de composants standards,
  - Ces composants standards peuvent être déployés dans différents conteneurs offrant un certain nombre de services.

# Introduction

## Caractéristiques de Java EE 6

- Une continuité dans la voie de la simplicité de développement en étendant les concepts d'annotations, POJO et configuration par l'exception à la couche présentation (dont les web services),
- Une amélioration des API comme EJB 3.1, JPA 2.0 ou le Timer Service,
- Une amélioration de la portabilité des applications par la standardisation des nommages JNDI,
- Une dépréciation de spécifications (concept de pruning),
- La création de sous-ensembles de la plateforme à travers le concept de profils

# Introduction

## Standards

- Java EE est basé sur des standards décrits dans les JSR (Java Specification Requests), contrairement à d'autres frameworks,
- Grâce à cela, votre application est déployable dans n'importe quel serveur d'application (open source ou commercial) qui implémente ces standards, avec très peu de changements
- Les serveurs d'applications les plus connus sont : WebLogic, WebSphere, GlassFish, JBoss, Jonas, Hibernate, Open JPA, Jersey, ...

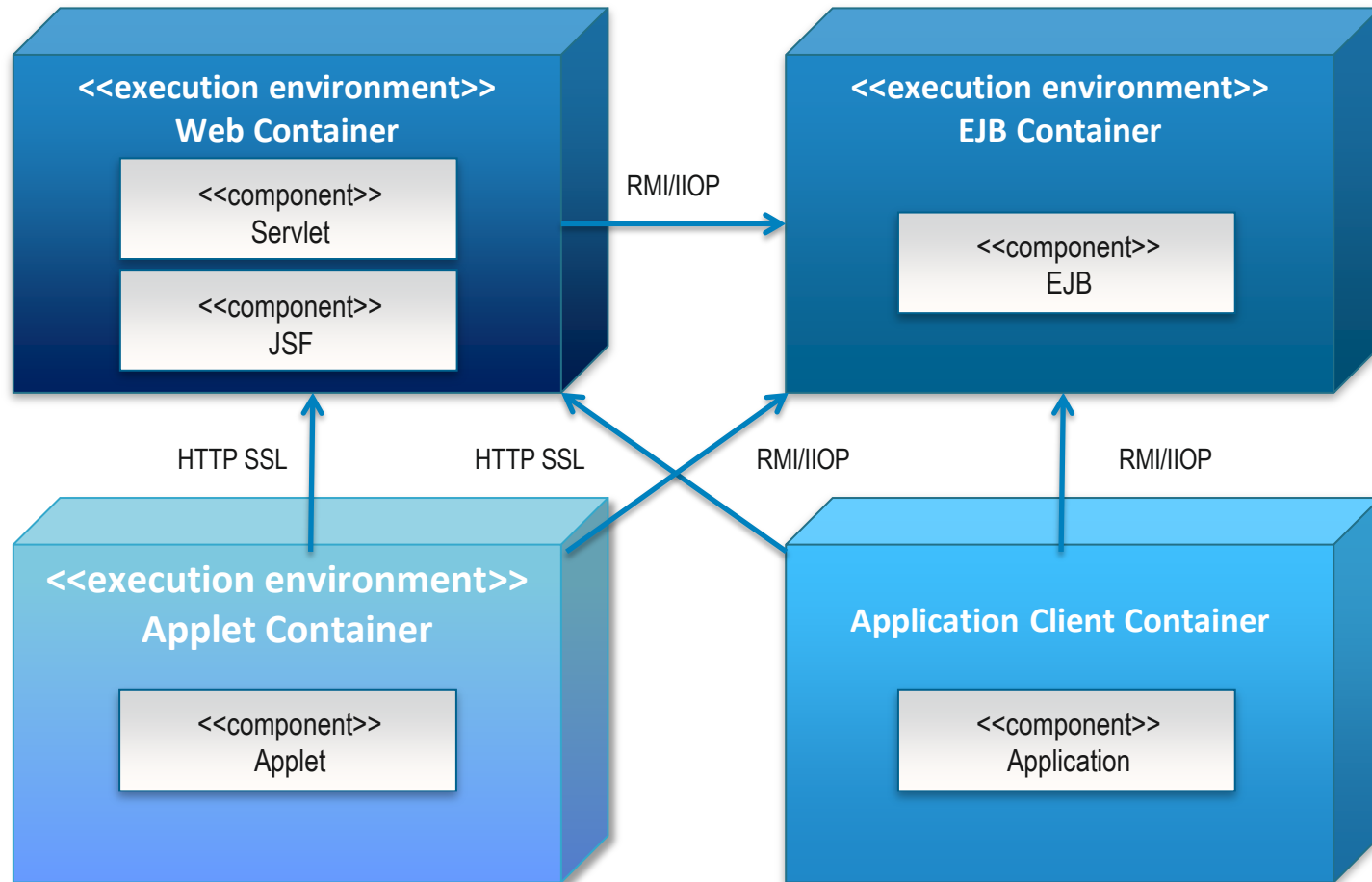
# Comprendre Java EE

## Architecture

- Java EE est un ensemble de spécifications implémentées par différents conteneurs,
- Un conteneur est un environnement d'exécution qui fournit un certain nombre de services aux composants qu'il porte (cycle de vie, injection de dépendances, transactions, sécurité, ...),
- Les composants respectent des contrats prédéfinis, ils sont notamment packagés de façon standard,
- Toutes les API de Java SE restent utilisables par les composants Java EE

# Comprendre Java EE

## Composants et conteneurs



# Comprendre Java EE

## Composants

- L'environnement d'exécution de Java EE définit 4 types de composants :
  - Applets : exécutées dans un navigateur web
  - Applications : programmes exécutés sur le poste client, avec ou sans interface, ayant accès aux services Java EE
  - Applications web : exécutées dans un conteneur web et répondant à des requêtes HTTP
  - Enterprise Java Beans : composants contenant la logique métier, accédés de façon locale ou distante (RMI, HTTP, SOAP, REST)

# Comprendre Java EE

## Conteneurs

- L'infrastructure de Java EE est divisée en domaines logiques appelés conteneurs,
- Chaque conteneur joue un rôle spécifique, supporte un jeu d'API et offre des services aux composants (sécurité, persistance, transactions, recherche, injection, ...)
- Le conteneur cache la complexité technique et facilite la portabilité,
- Suivant le type d'application à construire, il faut comprendre les contraintes de fonctionnalités de chaque conteneur,

# Comprendre Java EE

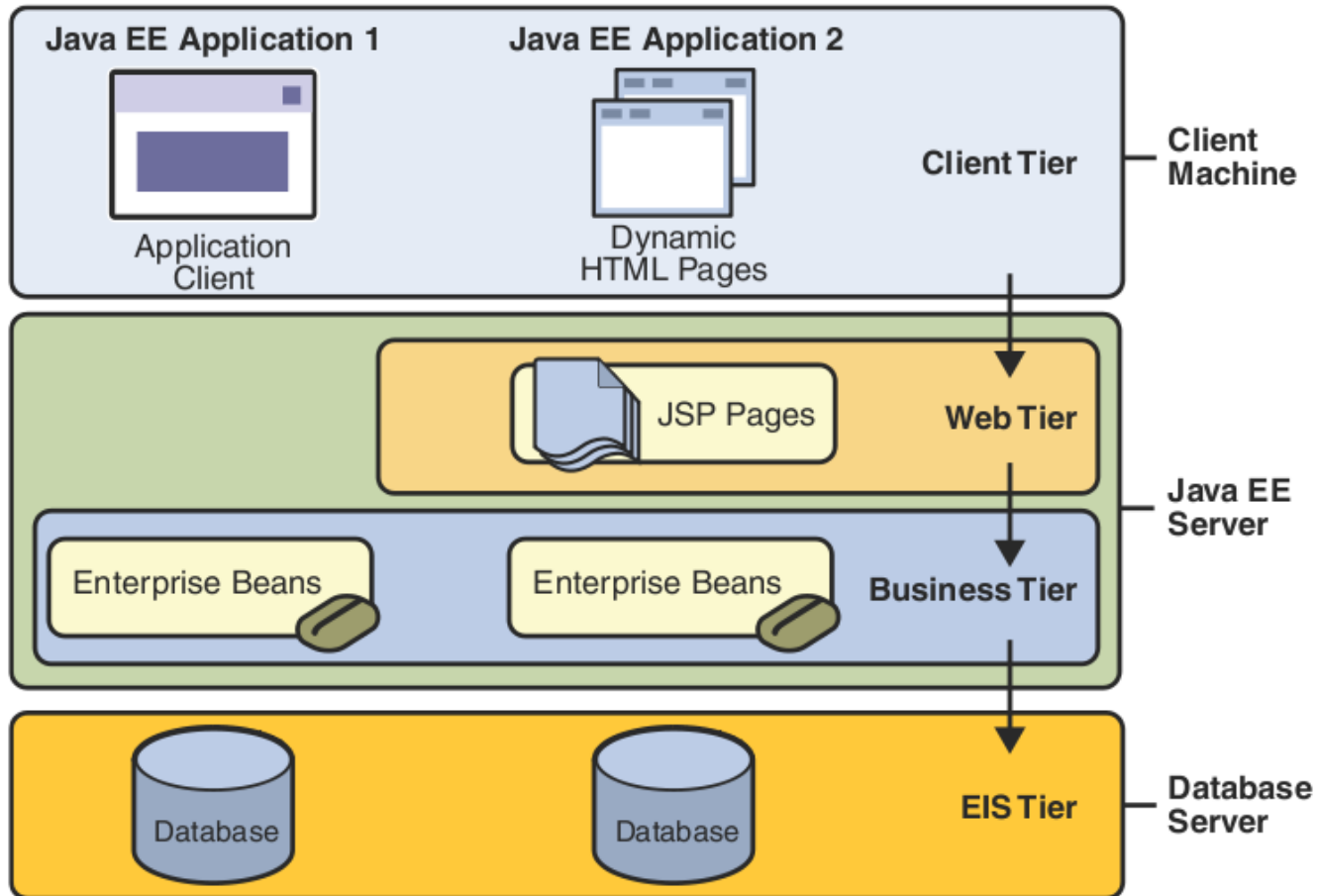
## Services

- Les conteneurs fournissent les services aux composants déployés.
- En tant que développeurs, vous pouvez vous concentrer sur l'implémentation de la logique métier plutôt que sur ces aspects techniques communs aux applications d'entreprise.

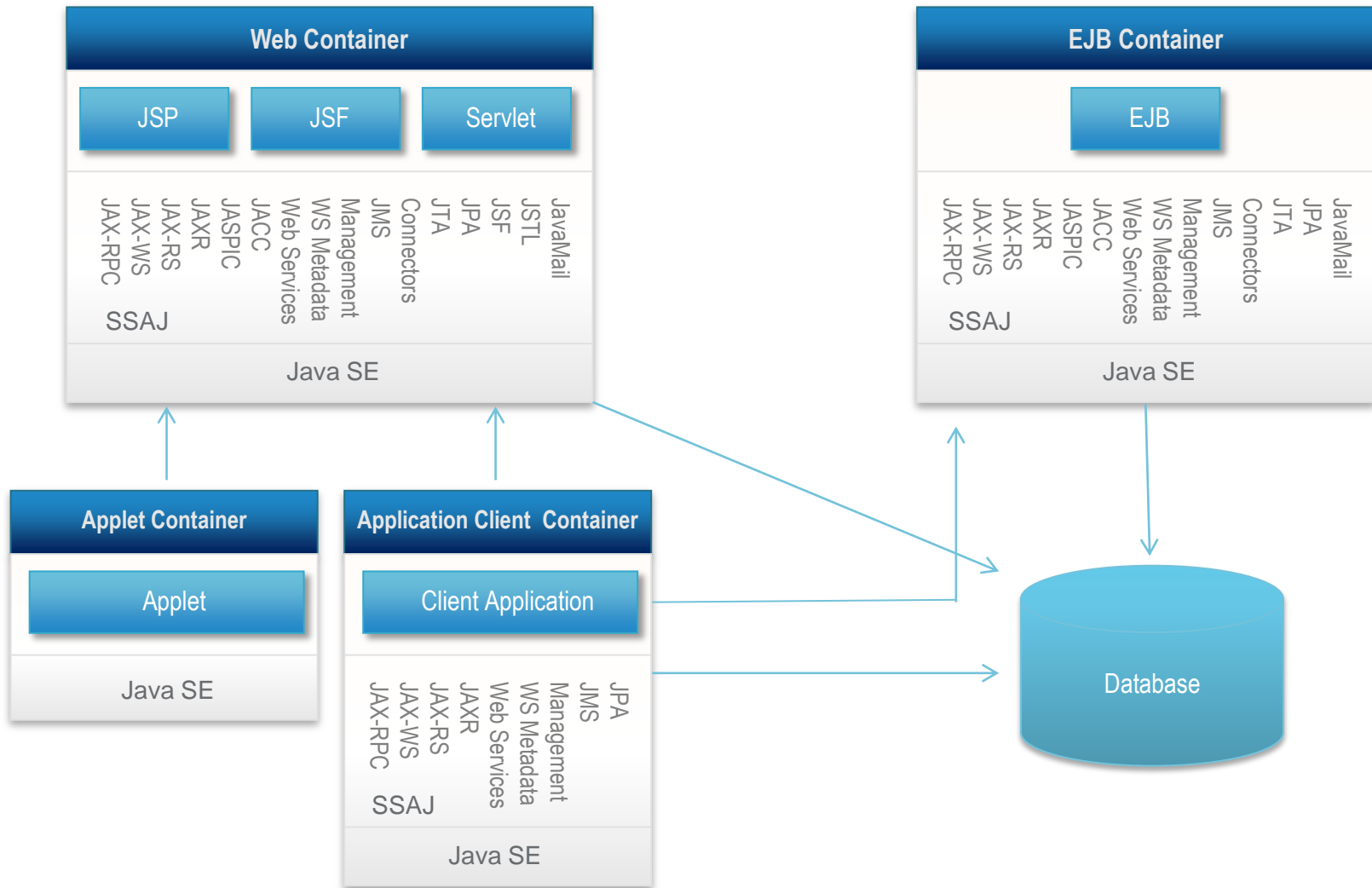


# Comprendre Java EE

Architecture en couches



# Comprendre Java EE



# Comprendre Java EE

## Architecture

- Services
  - JTA : Java Transaction API : gestion des transactions utilisée par le conteneur et l'application
  - JPA : Java Persistence API : service de transformation objet-relationnel (ORM)
  - JMS : Java Messaging Service : service de communication asynchrone par messages entre composants. Point à point ou Publish-Subscribe
  - JNDI : Java Naming and Directory Interface : accès à des services de nommage et de recherche

# Comprendre Java EE

## Architecture

- JavaMail : envoi de mails
- JavaBeans Activation Framework (JAF) : framework de gestion des données dans différents types MIME. Utilisé par JavaMail
- XML processing : Java API for XML Processing (JAXP) pour manipuler les documents XML
- JCA : Connecteurs java vers des systèmes externes, comme mainframes, bases, ERP, ...
- Security services : Java Authentication and Authorization Service (JAAS) contrôle les accès

# Comprendre Java EE

## Architecture

- Web Services : supporte SOAP (JAX-WS) et les services RESTful (JAX-RS)
- Management : composants pour contrôler conteneurs et serveurs. Java Management Extensions (JMX)
- Deployment : Java EE Deployment Specification définit un contrat entre outils de déploiement et serveurs JEE

# Comprendre Java EE

## Architecture

- Protocoles réseau
  - Les composants déployés dans les conteneurs sont accessibles par différents protocoles. JEE supporte les protocoles suivants :
    - HTTP : protocole du web
    - RMI – IIOP : Remote Method Invocation permet d'invoquer des objets distants indépendamment du protocoles de transport. Le protocole de transport par défaut de Java SE est JRMP (Java Remote Method Protocol). RMI-IIOP est une extension de RMI utilisée pour s'intégrer avec CORBA. Les composants peuvent invoquer des objets CORBA.

# Comprendre Java EE

## Architecture

- Packaging : pour être déployés dans un conteneur, les composants doivent d'abord être packagés de façon standard. Java SE définit les Archive Java (jar) contenant fichiers source et ressources. Java EE définit d'autres types d'archives basées sur le format jar : war, ear, EJB jar, AAC jar.

# Comprendre Java EE

## Spécifications de Java EE 6

- Java EE 6 est une spécification définie par la JSR 316 et regroupe un ensemble de 28 autres spécifications.
- Un serveur d'application compatible Java EE 6 doit implémenter toutes ces spécifications



# Comprendre Java EE

## Spécifications de Java EE 6

**Table 1-1.** *Java Enterprise Edition Specification*

Specification	Version	JSR	URL
Java EE	6.0	316	<a href="http://jcp.org/en/jsr/detail?id=316">http://jcp.org/en/jsr/detail?id=316</a>

**Table 1-2.** *Web Services Specifications*

Specification	Version	JSR	URL	Pruned
JAX-RPC	1.1	101	<a href="http://jcp.org/en/jsr/detail?id=101">http://jcp.org/en/jsr/detail?id=101</a>	X
JAX-WS	2.2	224	<a href="http://jcp.org/en/jsr/detail?id=224">http://jcp.org/en/jsr/detail?id=224</a>	
JAXB	2.2	222	<a href="http://jcp.org/en/jsr/detail?id=222">http://jcp.org/en/jsr/detail?id=222</a>	
JAXM	1.0	67	<a href="http://jcp.org/en/jsr/detail?id=67">http://jcp.org/en/jsr/detail?id=67</a>	
StAX	1.0	173	<a href="http://jcp.org/en/jsr/detail?id=173">http://jcp.org/en/jsr/detail?id=173</a>	
Web Services	1.2	109	<a href="http://jcp.org/en/jsr/detail?id=109">http://jcp.org/en/jsr/detail?id=109</a>	
Web Services Metadata	1.1	181	<a href="http://jcp.org/en/jsr/detail?id=181">http://jcp.org/en/jsr/detail?id=181</a>	
JAX-RS	1.0	311	<a href="http://jcp.org/en/jsr/detail?id=311">http://jcp.org/en/jsr/detail?id=311</a>	
JAXR	1.1	93	<a href="http://jcp.org/en/jsr/detail?id=93">http://jcp.org/en/jsr/detail?id=93</a>	X

# Comprendre Java EE

## Spécifications de Java EE 6

**Table 1-3.** *Web Specifications*

Specification	Version	JSR	URL	Pruned
JSF	2.0	314	<a href="http://jcp.org/en/jsr/detail?id=314">http://jcp.org/en/jsr/detail?id=314</a>	
JSP	2.2	245	<a href="http://jcp.org/en/jsr/detail?id=245">http://jcp.org/en/jsr/detail?id=245</a>	
JSTL (JavaServer Pages Standard Tag Library)	1.2	52	<a href="http://jcp.org/en/jsr/detail?id=52">http://jcp.org/en/jsr/detail?id=52</a>	
Servlet	3.0	315	<a href="http://jcp.org/en/jsr/detail?id=315">http://jcp.org/en/jsr/detail?id=315</a>	
Expression Language	1.2	245	<a href="http://jcp.org/en/jsr/detail?id=245">http://jcp.org/en/jsr/detail?id=245</a>	

# Comprendre Java EE

## Spécifications de Java EE 6

**Table 1-4.** *Enterprise Specifications*

Specification	Version	JSR	URL	Pruned
EJB	3.1	318	<a href="http://jcp.org/en/jsr/detail?id=318">http://jcp.org/en/jsr/detail?id=318</a>	
JAF	1.1	925	<a href="http://jcp.org/en/jsr/detail?id=925">http://jcp.org/en/jsr/detail?id=925</a>	
JavaMail	1.4	919	<a href="http://jcp.org/en/jsr/detail?id=919">http://jcp.org/en/jsr/detail?id=919</a>	
JCA	1.6	322	<a href="http://jcp.org/en/jsr/detail?id=322">http://jcp.org/en/jsr/detail?id=322</a>	
JMS	1.1	914	<a href="http://jcp.org/en/jsr/detail?id=914">http://jcp.org/en/jsr/detail?id=914</a>	
JPA	2.0	317	<a href="http://jcp.org/en/jsr/detail?id=317">http://jcp.org/en/jsr/detail?id=317</a>	
JTA	1.1	907	<a href="http://jcp.org/en/jsr/detail?id=907">http://jcp.org/en/jsr/detail?id=907</a>	

# Comprendre Java EE

## Spécifications de Java EE 6

**Table 1-5.** *Management, Security, and Other Specifications*

Specification	Version	JSR	URL	Pruned
JACC	1.1	115	<a href="http://jcp.org/en/jsr/detail?id=115">http://jcp.org/en/jsr/detail?id=115</a>	
Bean Validation	1.0	303	<a href="http://jcp.org/en/jsr/detail?id=303">http://jcp.org/en/jsr/detail?id=303</a>	
Common Annotations	1.0	250	<a href="http://jcp.org/en/jsr/detail?id=250">http://jcp.org/en/jsr/detail?id=250</a>	
Java EE Application Deployment	1.2	88	<a href="http://jcp.org/en/jsr/detail?id=88">http://jcp.org/en/jsr/detail?id=88</a>	X
Java EE Management	1.1	77	<a href="http://jcp.org/en/jsr/detail?id=77">http://jcp.org/en/jsr/detail?id=77</a>	X
Java Authentication Service Provider Interface for Containers	1.0	196	<a href="http://jcp.org/en/jsr/detail?id=196">http://jcp.org/en/jsr/detail?id=196</a>	
Debugging Support for Other Languages	1.0	45	<a href="http://jcp.org/en/jsr/detail?id=45">http://jcp.org/en/jsr/detail?id=45</a>	

# Les nouveautés de Java EE 6

## Objectifs

- Objectif premier de Java EE 6 : continuer de simplifier le développement en continuant le travail initié avec Java EE 5 :
  - Dans Java EE 5, les EJBs, entités persistantes, et web services ont été revus pour suivre une approche plus orientée objet (classes java implémentant une interface) et pour utiliser les annotations comme façon de définir les méta-données (les fichiers de déploiement XML devenant optionnels),

# Les nouveautés de Java EE 6

## Objectifs

- Java EE 6 suit cette voie et emploie les mêmes principes à la couche web. Maintenant, un Managed Bean JSF est une simple classe Java avec annotation et un descripteur XML optionnel.
- Java EE 6 se concentre aussi sur la simplicité d'utilisation de la plateforme en introduisant les profils et en marquant "pruned" (déprécié) des vieilles technologies.
- Java EE 6 améliore aussi certaines spécifications et en ajoute de nouvelles.

# Les nouveautés de Java EE 6

## Allègement

- Le groupe d'expert Java EE 6 a dû répondre au problème suivant : comment rendre la plateforme plus légère tout en ajoutant de nouvelles spécifications ?
- La réponse des experts fut :
  - Pruning,
  - Profils,
  - EJB lite

# Les nouveautés de Java EE 6

## Allègement - Pruning

- Le processus de "Pruning" consiste à proposer des fonctionnalités à supprimer des prochaines versions de Java EE.
- Certaines fonctionnalités seront remplacées par d'autres plus récentes (ex Entity Beans remplacés par JPA), et d'autres quitteront le groupe JEE pour redevenir des JSR indépendantes.



# Les nouveautés de Java EE 6

## Allègement - Pruning

- Les fonctionnalités suivantes sont marquées "pruned" :
  - EJB 2.x Entity Beans CMP (partie de JSR 318) : modèle de persistance → remplacé par JPA
  - JAX-RPC (JSR 101) → remplacé par JAX-WS
  - JAXR (JSR 93) : communication avec les annuaires UDDI → quittera JEE
  - JEE Application Deployment (JSR 88) : outils de déploiement → quittera JEE
  - JEE Management (JSR 77) : outils de management → quittera JEE

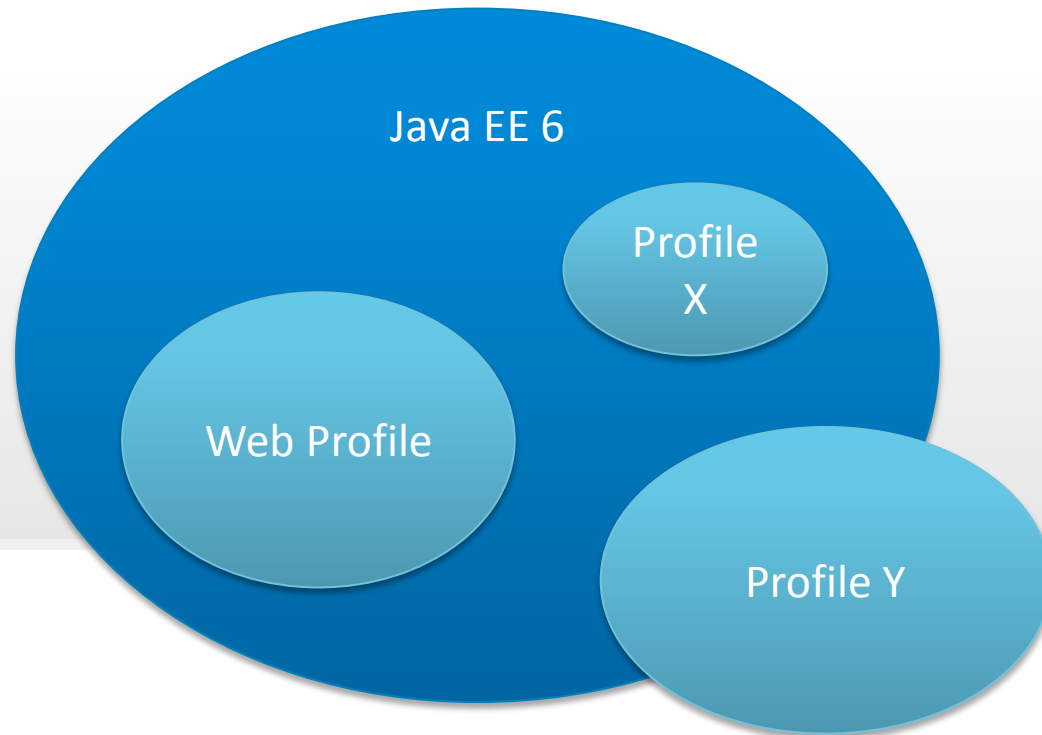
# Les nouveautés de Java EE 6

## Allègement - Profils

- C'est une nouveauté majeure de la plateforme. Le but est de réduire la taille de la plateforme de déploiement pour suivre les besoins des développeurs : une critique majeure de JEE est en effet que quelle que soient les besoins de l'application à déployer, la plateforme implémente toujours toutes les spécifications de la version de JEE.
- Les profils sont des sous-ensembles ou sur-ensembles de la plateforme.

# Les nouveautés de Java EE 6

Allègement - Profils



# Les nouveautés de Java EE 6

## Allègement - Profils

- La spécification Java EE 6 définit explicitement un seul profil appelé "Web Profile"
- Web Profile permet de créer des applications web
- Web Profile 1.0 est défini dans une spécification séparée
- D'autres profils pourront être créés dans le futur
- Du coup, des serveurs d'applications conformes Web Profile 1.0 plutôt que Java EE 6 vont apparaître

# Les nouveautés de Java EE 6

## Allègement - Profils

**Table 1-6.** *Web Profile 1.0 Specifications*

Specification	Version	JSR	URL
JSF	2.0	314	<a href="http://jcp.org/en/jsr/detail?id=314">http://jcp.org/en/jsr/detail?id=314</a>
JSP	2.2	245	<a href="http://jcp.org/en/jsr/detail?id=245">http://jcp.org/en/jsr/detail?id=245</a>
JSTL	1.2	52	<a href="http://jcp.org/en/jsr/detail?id=52">http://jcp.org/en/jsr/detail?id=52</a>
Servlet	3.0	315	<a href="http://jcp.org/en/jsr/detail?id=315">http://jcp.org/en/jsr/detail?id=315</a>
Expression Language	1.2		
EJB Lite	3.1	318	<a href="http://jcp.org/en/jsr/detail?id=318">http://jcp.org/en/jsr/detail?id=318</a>
JPA	2.0	317	<a href="http://jcp.org/en/jsr/detail?id=317">http://jcp.org/en/jsr/detail?id=317</a>
JTA	1.1	907	<a href="http://jcp.org/en/jsr/detail?id=907">http://jcp.org/en/jsr/detail?id=907</a>
Common Annotations	1.0	250	<a href="http://jcp.org/en/jsr/detail?id=250">http://jcp.org/en/jsr/detail?id=250</a>

# Les nouveautés de Java EE 6

## Facilité d'utilisation

- En plus d'être une plateforme plus légère, Java EE 6 cherche aussi à être plus facile d'utilisation
- Java EE 6 simplifie la couche web :
  - servlets, managed beans JSF, validators, renderers deviennent des classes java annotées avec un fichier de déploiement XML optionnel
  - Dans la plupart des cas, le fichier faces-config.xml devient optionnel aussi pour JSF

# Les nouveautés de Java EE 6

## Facilité d'utilisation

### BookController.java

```
@ManagedBean
public class BookController {
    @EJB
    private BookEJB bookEJB;

    private Book book = new Book();
    private List<Book> bookList = new ArrayList<Book>();

    public String doCreateBook() {
        book = bookEJB.createBook(book);
        bookList = bookEJB.findBooks();
        return "listBooks.xhtml";
    }
    // Getters, setters
}
```

# Les nouveautés de Java EE 6

## Facilité d'utilisation

- Les EJBs sont également devenus plus faciles à développer :
  - S'ils ne doivent être accédés que localement, une simple classe java annotée sans interface est suffisante,
  - Ils peuvent être déployés directement dans un fichier war sans être packagés dans un fichier jar.



# Les nouveautés de Java EE 6

## Facilité d'utilisation

BookEJB.java

```
@Stateless
public class BookEJB {
    @PersistenceContext(unitName = "chapter01PU")
    private EntityManager em;

    public Book findBookById(Long id) {
        return em.find(Book.class, id);
    }

    public Book createBook(Book book) {
        em.persist(book);
        return book;
    }
}
```

# Les nouveautés de Java EE 6

Plus riche

- Java EE 6 devient également plus riche en améliorant des spécifications et en ajoutant de nouvelles :
  - JAX-RS permet de développer des web services RESTful
  - JPA 2.0 ajoute des collections d'objets simples (String, Integer, ...), le verrouillage pessimiste, une syntaxe JPQL plus riche, une nouvelle API Query Definition et le support du cache
  - Appels asynchrones d'EJBs, Timer service plus évolué
  - Session bean de type singleton

# Les nouveautés de Java EE 6

## Facilité d'utilisation

### BookResource.java

```
@Path("books")
public class BookResource {
    @PersistenceContext(unitName = "chapter01PU")
    private EntityManager em;

    @GET
    @Produces({"application/xml", "application/json"})
    public List<Book> getAllBooks() {
        Query query = em.createNamedQuery("findAllBooks");
        List<Book> books = query.getResultList();
        return books;
    }
}
```

# Les nouveautés de Java EE 6

## Facilité d'utilisation

### CacheEJB.java

```
@Singleton
public class CacheEJB {
    private Map<Long, Object> cache = new HashMap<Long, Object>();
    public void addToCache(Long id, Object object) {
        if (!cache.containsKey(id))
            cache.put(id, object);
    }
    public Object getFromCache(Long id) {
        if (cache.containsKey(id))
            return cache.get(id);
        else
            return null;
    }
}
```

# Les nouveautés de Java EE 6

Plus portable

- L'objectif de Java EE a toujours été de pouvoir déployer les applications sur n'importe quel serveur d'application. Cependant, comme les spécifications ne couvrent pas tous les détails, certaines implémentations ne sont pas portables (fichiers de configuration, noms JNDI, ...)
- Java EE 6 fixe maintenant la syntaxe des noms JNDI

# Les nouveautés de Java EE 6

Plus portable

- Une autre des difficultés de JEE a toujours été le test et l'utilisation des EJB dans un environnement Java SE. Certains serveurs d'applications (ex Jboss) ont des implémentations spécifiques pour le faire
- EJB 3.1 définit un "embedded container" c'est à dire une API standard pour exécuter les EJBs dans un environnement Java SE

# Conclusion

Ce qu'il faut retenir

- Java EE 6 permet de construire des applications d'entreprise
- Java EE 6 regroupe un ensemble de spécifications
- Java EE 6 va dans le sens de la facilité de développement et apporte un lot de nouveautés

