jahia

Présentation J2EE

Stéphane Croisier, Directeur
Serge Huber, Directeur Technique

13 Juin 2002

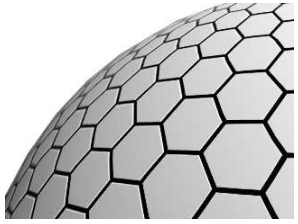


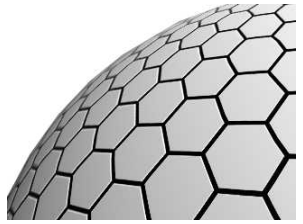
Table des Matières

☑ Qui sommes-nous?

☑ Introduction J2EE

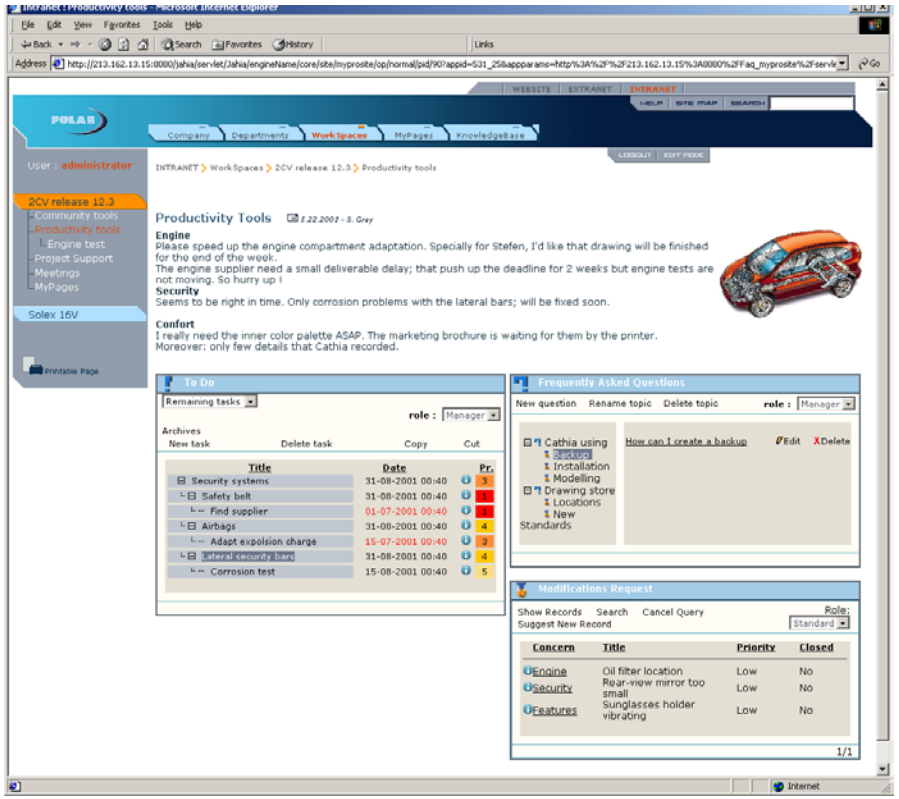
☑ Architecture J2EE

☑ Avantages / Désavantages



Le Projet Jahia

But: Développer un logiciel de Gestion de Contenu (CMS) et de Portail d'Entreprise (EIP)



Historique:

- Version 1 basée en C++ multi-plateformes

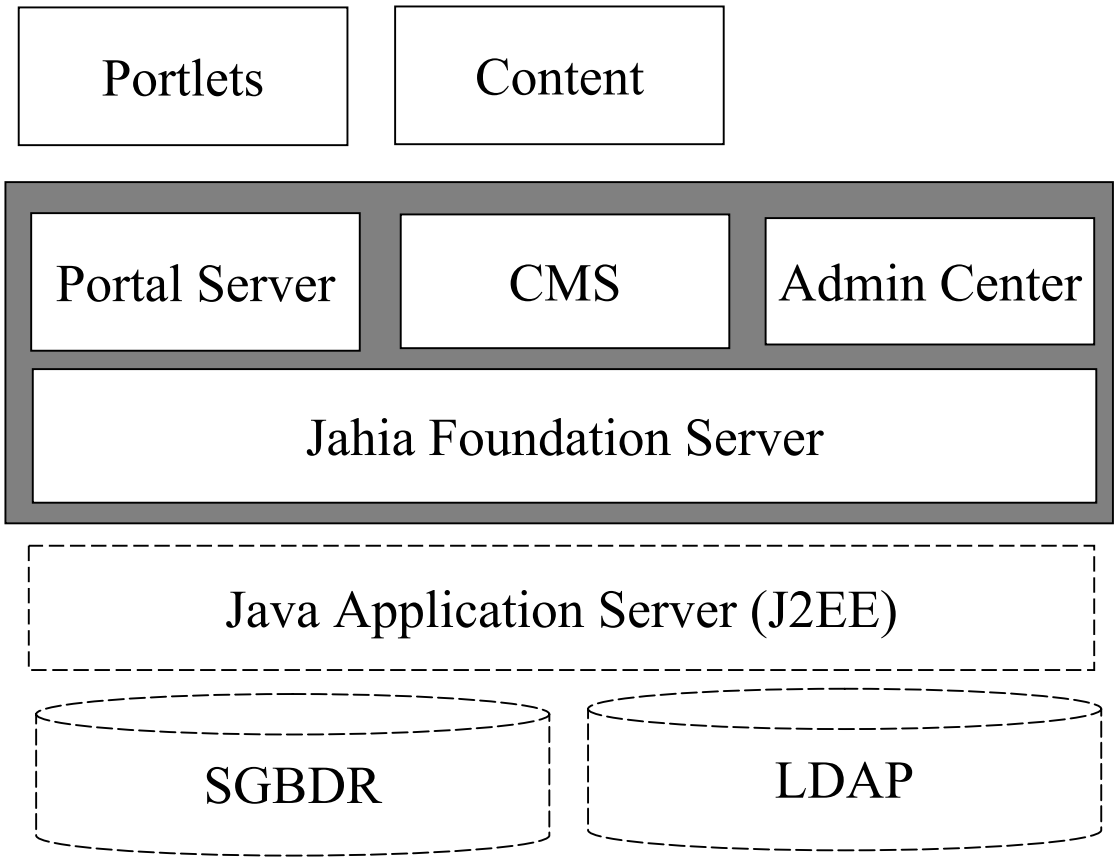
Résultat: 300'000 lignes de code pour moins de fonctionnalités que prévues

- Version 2 - Migration sur plateforme J2EE

Même taille de code pour beaucoup plus de fonctionnalités

Critères: Portabilité (Unix, Linux; NT), Stabilité, Performance, Standardisation, Parts de marché potentiels les plus grandes, Réduction de la complexité, Concurrence…

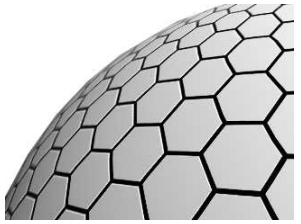
Jahia: Architecture logiciel



Pourquoi développer sur un Framework?

Les systèmes d'information deviennent:

- ☑ De plus en plus complexes (transactionnel; SGBD/LDAP; XML…)
- ☑ Hétérogénité des systèmes propriétaires ou dédiés (Legacy Systems)
- ☑ Demandes fortes d'interconnexions (passage d'une architecture « par silo » à une architecture par « couches de services interopérables »)
- ☑ Changements technologiques permanents
- ☑ Souci de rationalisation économique
  - Investissements et équipes de développement de plus en plus lourds
  - Arriver à tout recoder soit-même est devenu impossible



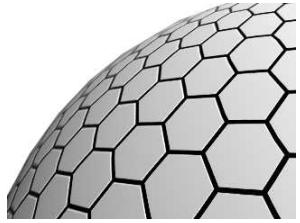
Pourquoi développer sur un Framework?

Conséquences:

Similaire à des incubateurs pour les start-ups, il y a des avantages et des désavantages

- ☑ S'appuyer sur une tuyauterie et des composants génériques existants et stables
- ☑ Volonté de réduire la complexité du projet
- ☑ Souci de se baser sur des standards afin d'assurer la pérennité des développements effectués
- ☑ Recentrage sur la logique métier et la valeur ajoutée fonctionnelle et métier
- ☑ Stabilité du système/ Performance / Portabilité laissé autant que possible à des tiers.
- ☑ Culture globale homogène qui facilite le transfert des compétences

Le choix d'un framework est devenu incontournable pour tout projet de développement conséquents et donc, de facto, pour l'ensemble du système d'information.

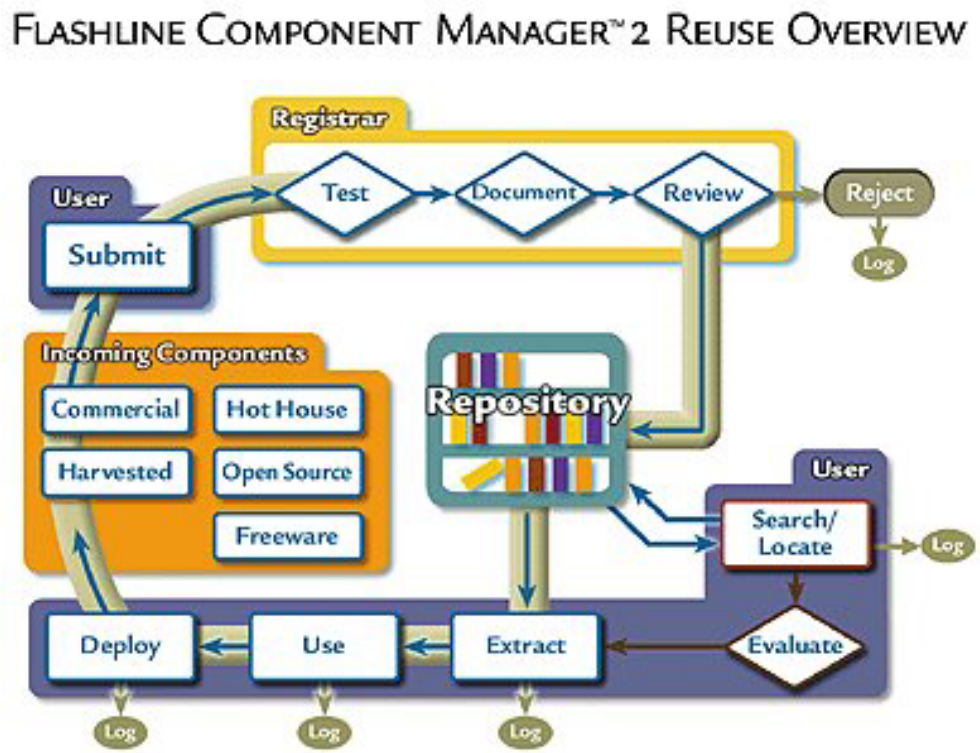


Approche orientée composants

Volonté de:

1) Ne pas réinventer la roue
2) D'utiliser des composants tiers réutilisable, stables et standardisés
3) Créer sa propre bibliothèque de composants réutilisables afin de réduire les coûts à moyen et long terme.

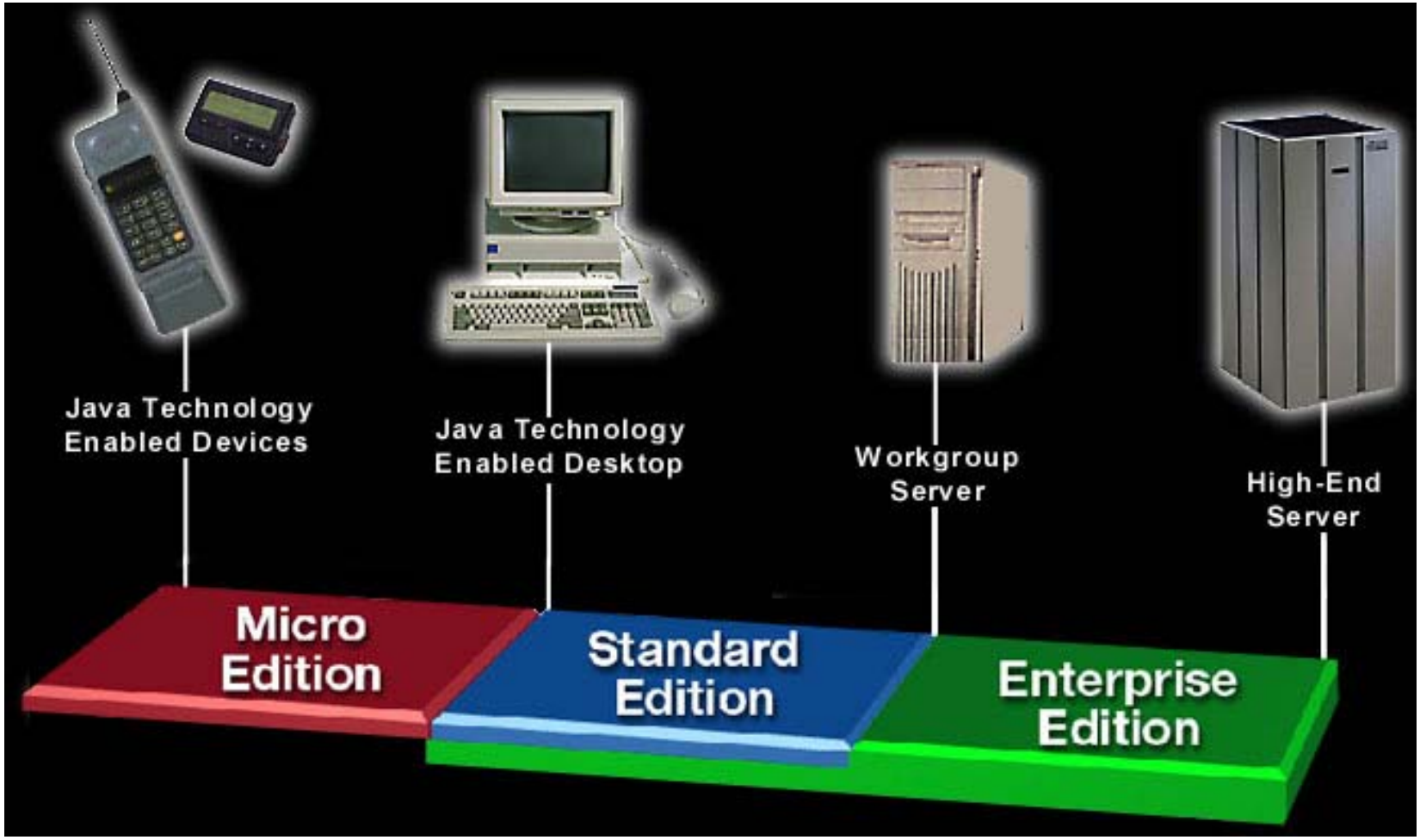
« La réutilisation logiciel peut réduire les coûts de développements de 15 à 75%. La réutilisation d'un composant coûte 30 à 75% moins chère que son re-développement. En 2003, au moins 70 pourcent de toutes les applications seront construites à partir de composants. » Gartner, 2001



J2EE Intro

Introduction à J2EE

J2EE



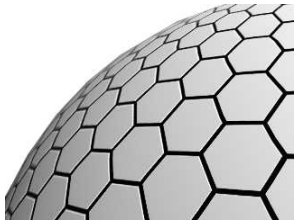
Produit versus Spécification

J2EE n'a pas la même signification que .NET

1) J2EE est une spécification avec de multipes implémentations
2) .NET est un mélange de produits et de spécifications

Avant de détailler les deux solutions, il s'agit donc de comparer ce qui est comparable.

(.NET à considérer comme architecture de développement et comme langage(s)/Runtime et non pas comme couches logiciels supplémentaires (base de données; progiciel de gestion de contenu; de portails d'entreprise, d'EAI…).



J2EE: un standard

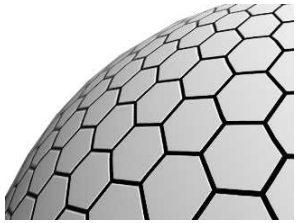
J2EE = Ensemble de spécifications issues de la collaboration entre différents éditeurs logiciels, développeurs et vendeurs hardware (JCP - www.jcp.org)

- Java Community Process (environ 200 JSR en cours sur quasiment tous les thèmes possibles)

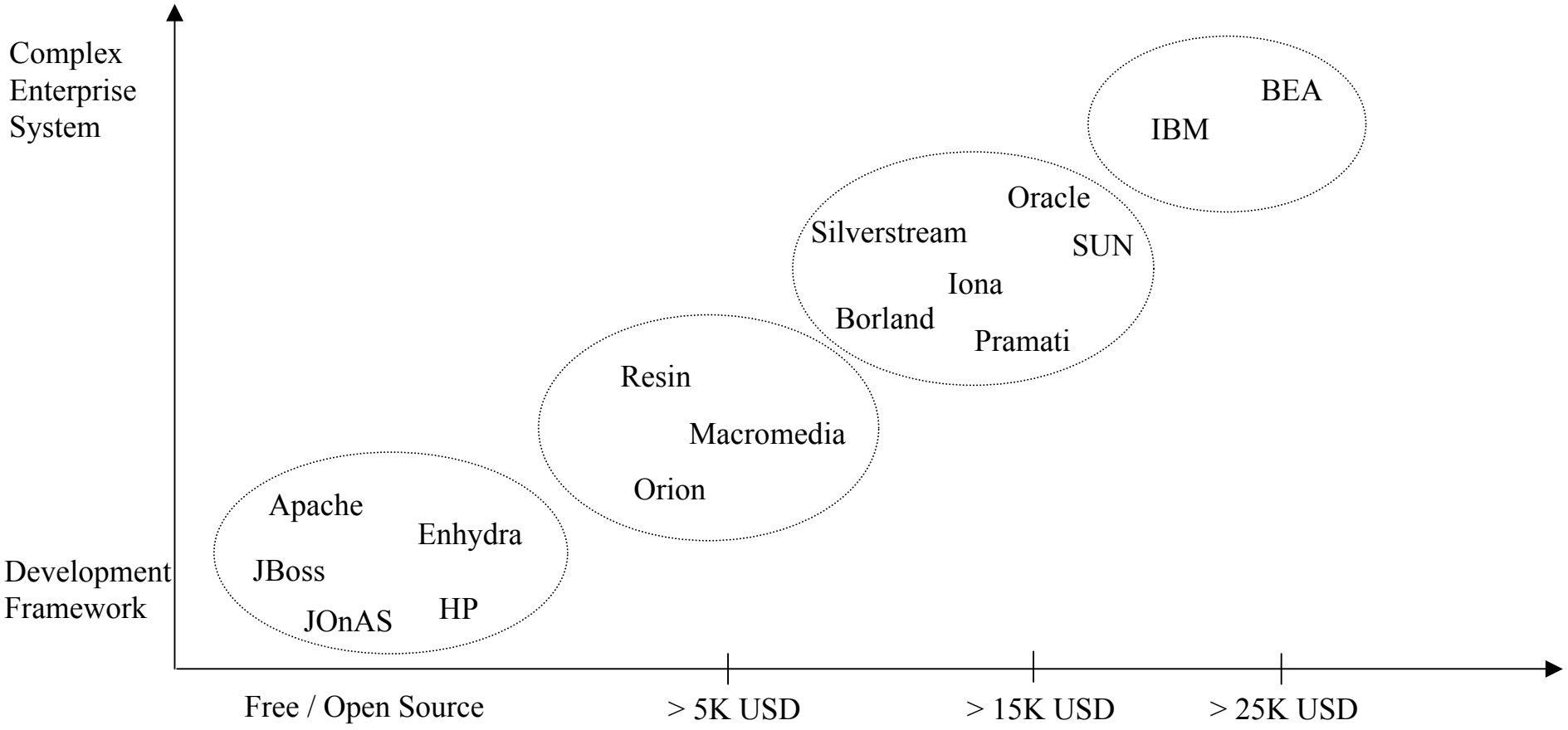
~40 vendeurs officiels implémentent tout ou partie de la spécification (app servers + outils de dév.)

- No single -vendor lock-in

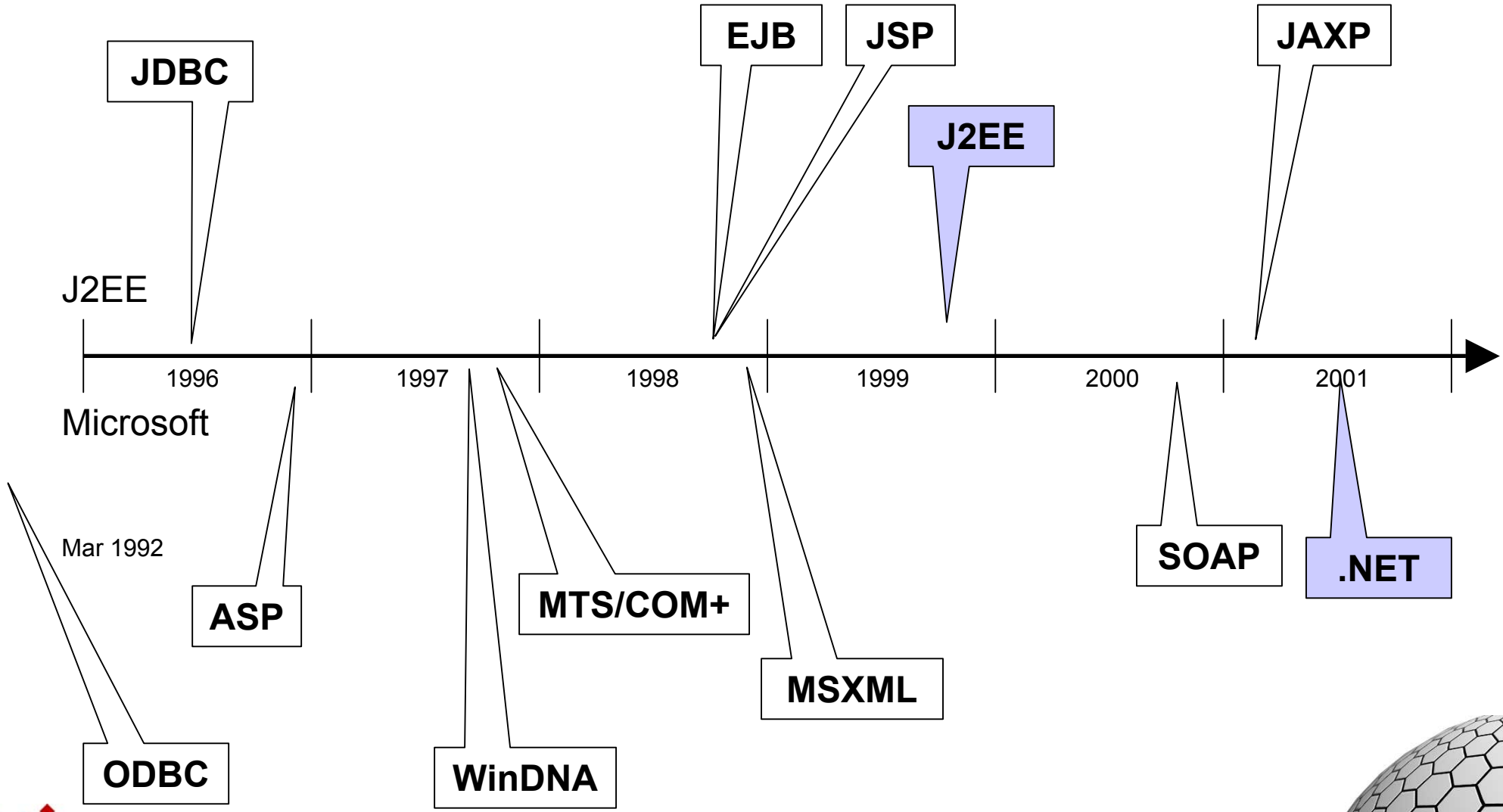
- ☑ Basé sur le language Java - JRE interprète du bytecode
- ☑ Compatibility Test Suite
- ☑ Une implementation de référence de chaque JCP est fournie
- ☑ J2EE Blueprints Design Guidelines sont également fournis



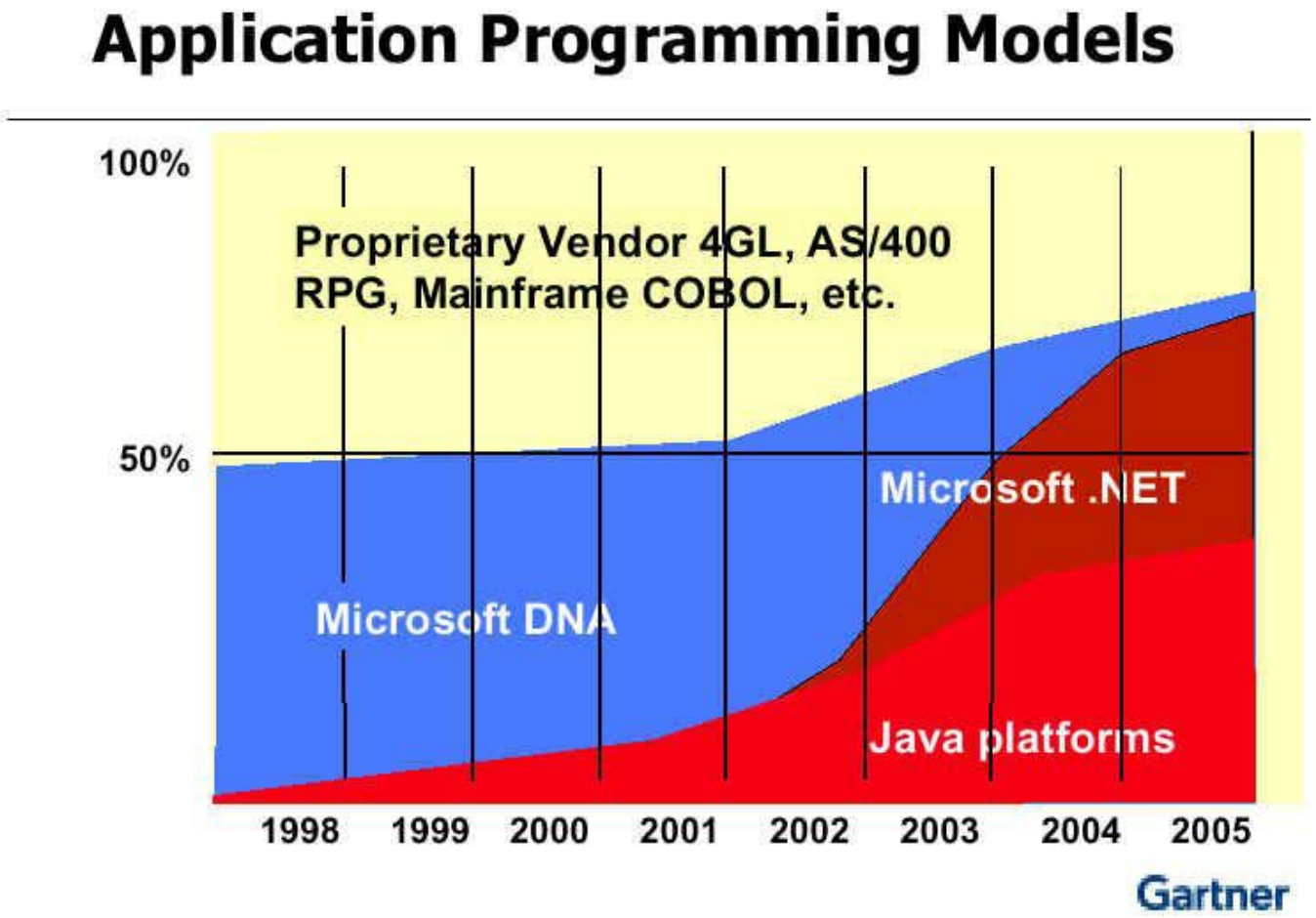
J2EE/Tools Players (~40)



Evolution MS/J2EE



Parts de marché



Source: Gartner Research

J2EE = une commodité?

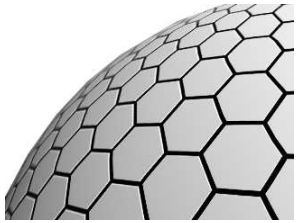
Constat: J2EE devient une commodité.

Conséquence:

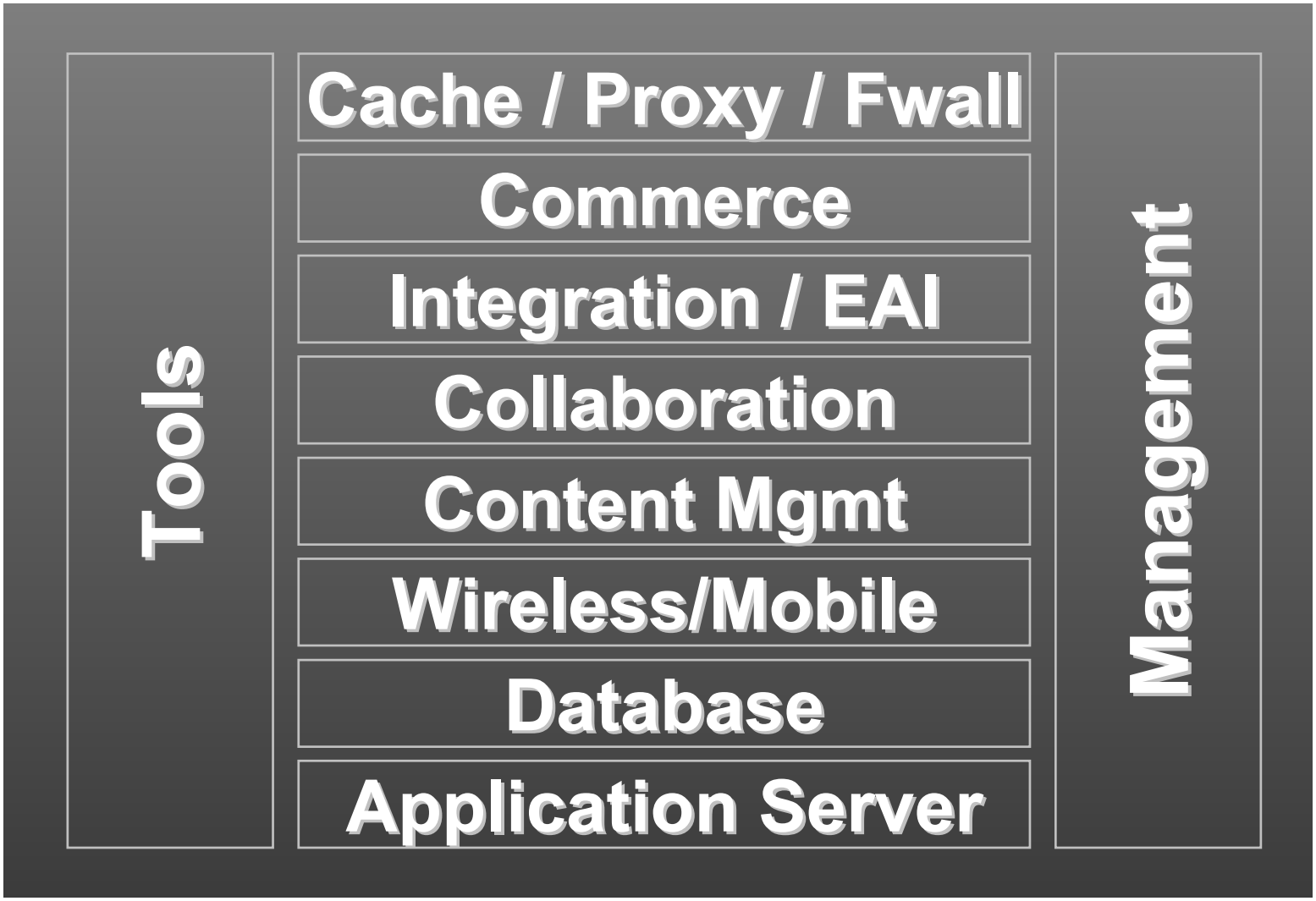
Les vendeurs s'attaquent aux couches logiciels supérieures.

Entre autres:

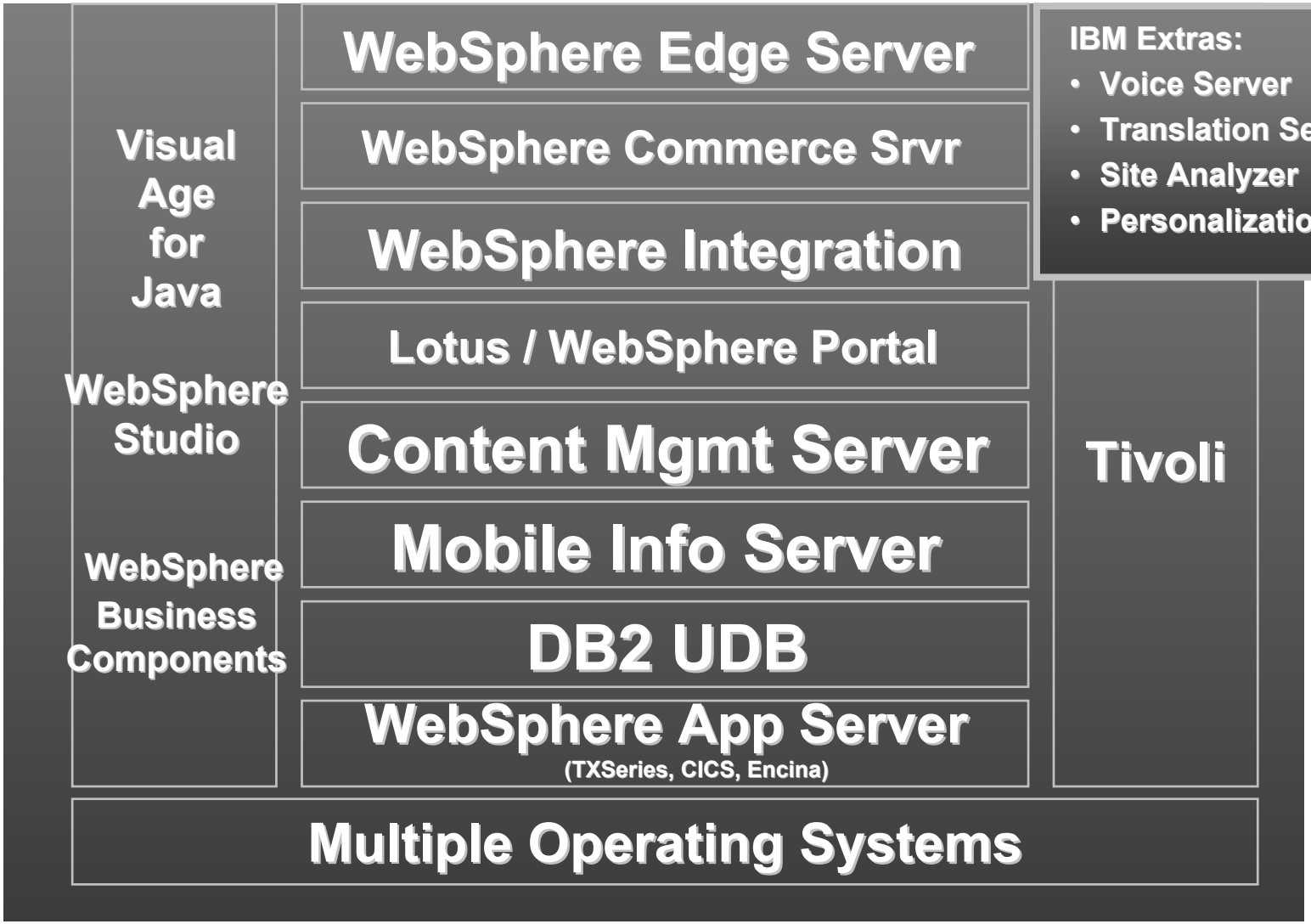
- ☑ Outils de développements
- ☑ SysAdmin/Load Balancing/…
- ☑ CMS/Portails d'entreprise/Serveur de Personalisation
- ☑ Calendar/Emails/Collaboration
- ☑ Ecommerce
- ☑ EAI / Intégration



Couches logiciels



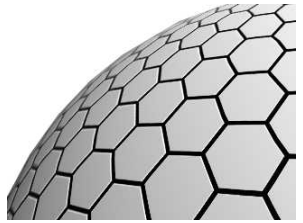
Couches Logiciels



Stack Complet J2EE/Tools (IDE) GRATUIT

Un serveur J2EE complet avec un environnement de développement performant peuvent désormais être installé gratuitement. En outre, la totalité du code source est disponible (open source)

IDE	SUN NetBeans / IBM Eclipse
Framework/Components	Apache / Open Source
Application Server	Tomcat / JBoss
Database	MySQL / PostgreSQL / Interbase
Operating System	Linux



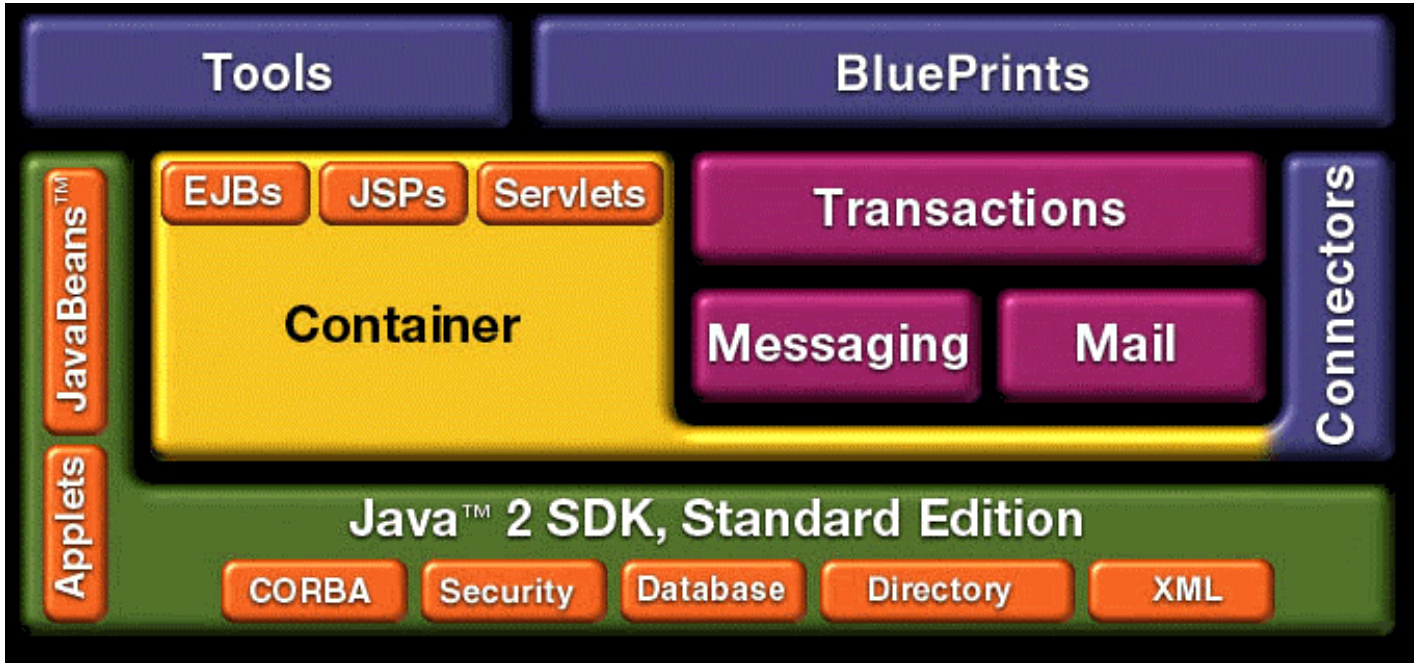
J2EE Architecture

Architecture J2EE

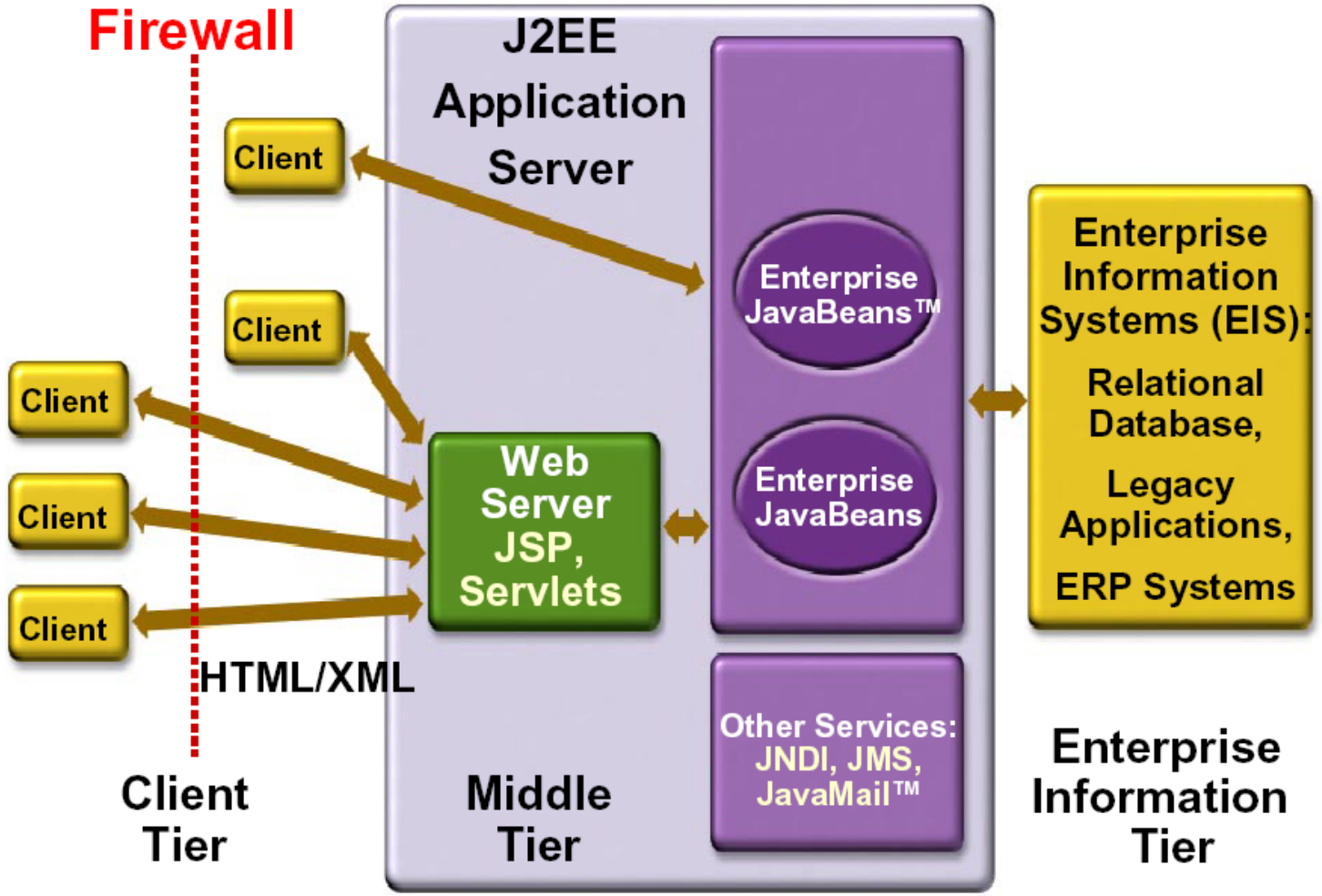
J2EE Containers

« The shared vision between J2EE and .NET is that there is an incredible amount of « plumbing » that goes into building web services, such as XML interoperability, load balancing and transactions.Rather than writing all that plumbing yourself, you can write an application that runs within a container that provides thus tricky services for you. »

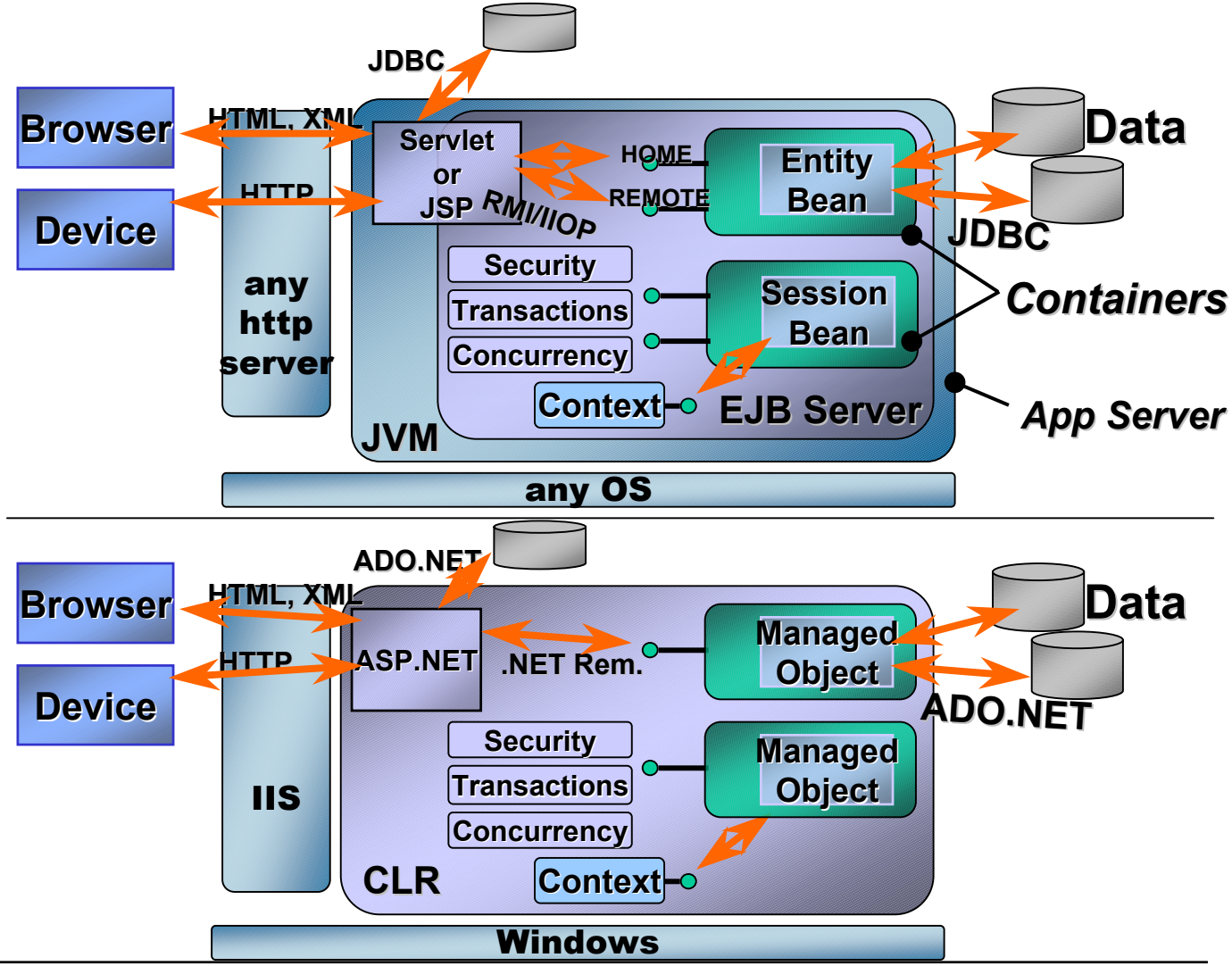
The Middleware Company, June 2001



Architecture trois tiers



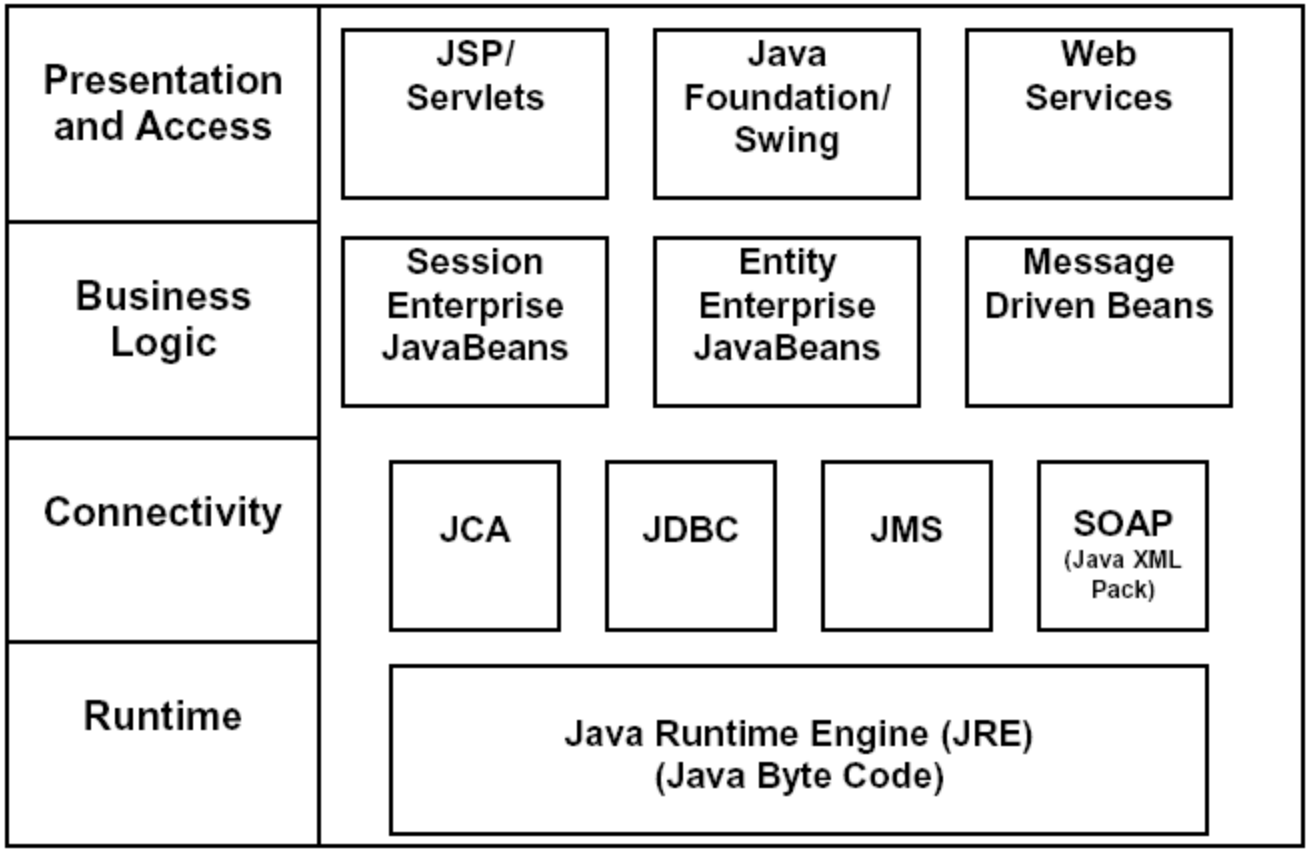
J2EE et .NET: Même Concept



Architecture trois tiers - J2EE

Le tiers intermédiaire permet d'assurer la disponibilité, la montée en charge et la stabilité.

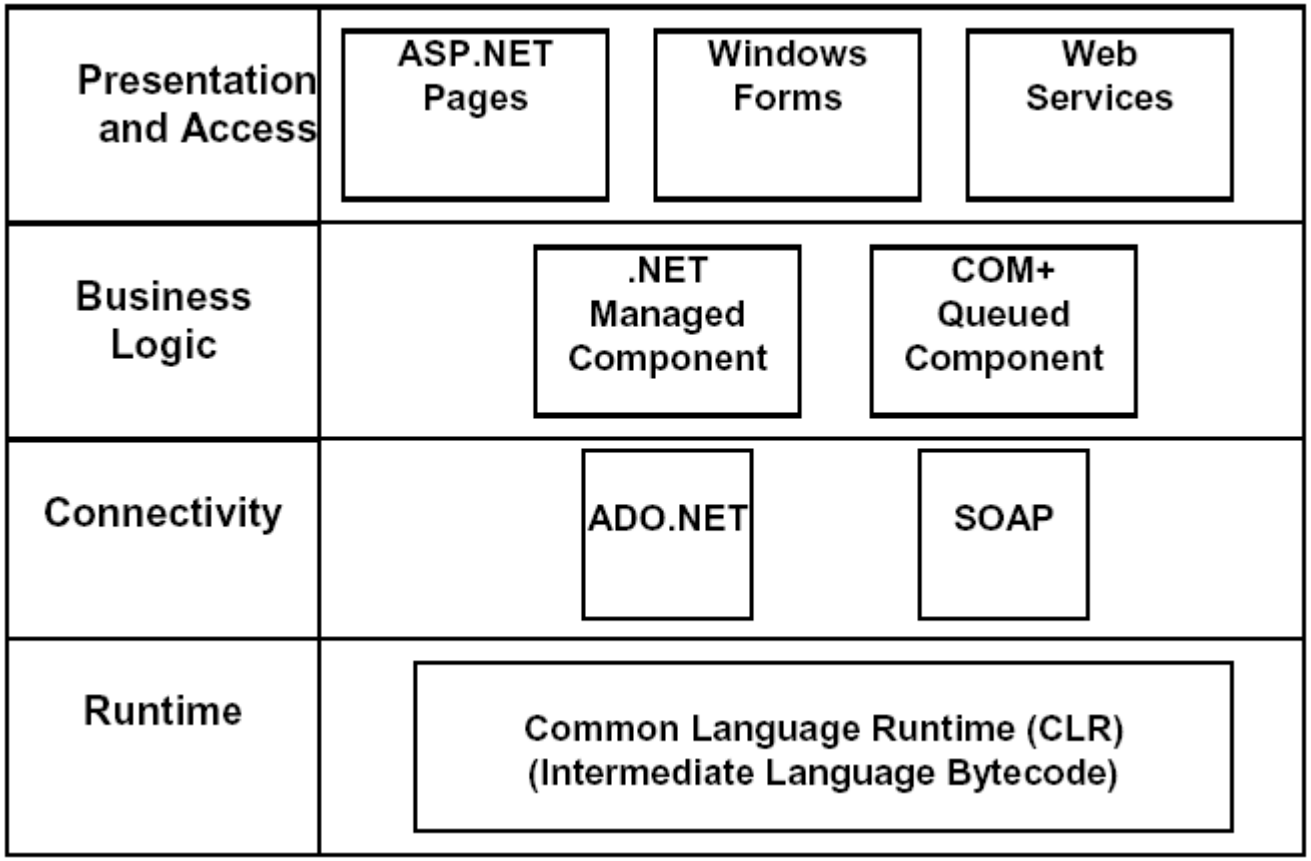
Les containers offrent des services de sécurité, de transactionnel ou de connectivité.



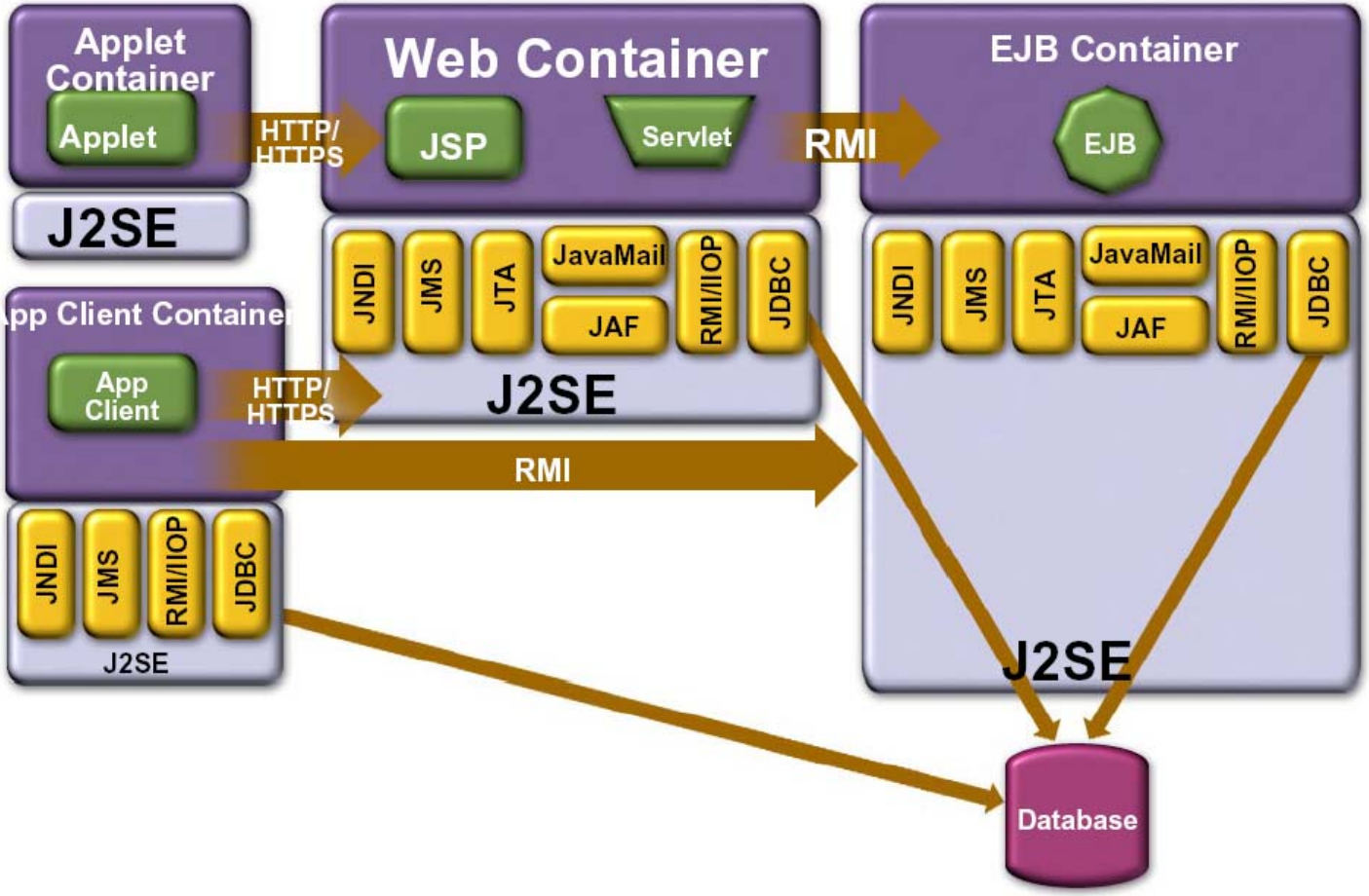
Architecture trois tiers - Microsoft

Offre Microsoft:

Très similaire du moins dans le design architectural.



J2EE Containers



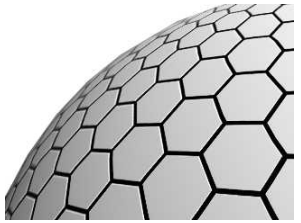
Containers and Components

Containers Handle

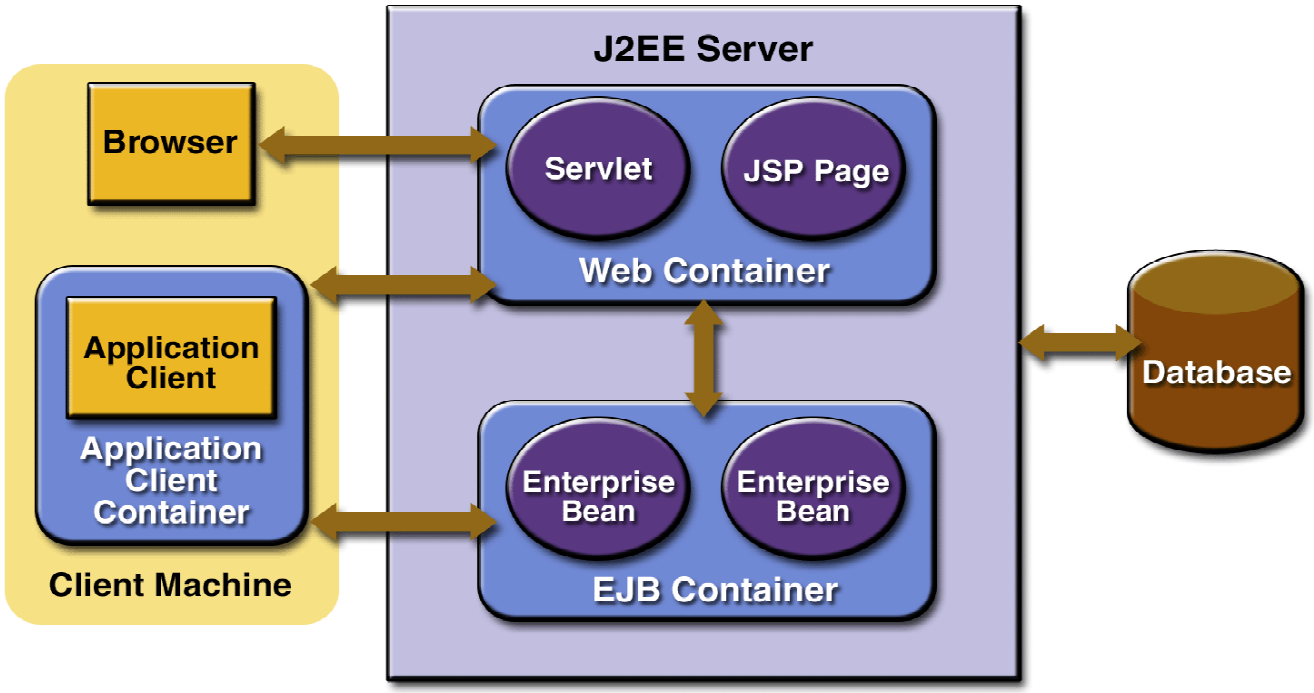
- Concurrency
- Security
- Availability
- Scalability
- Persistence
- Transaction
- Lifecycle management
- Management

Components Handle

- Presentation
- Business Logic

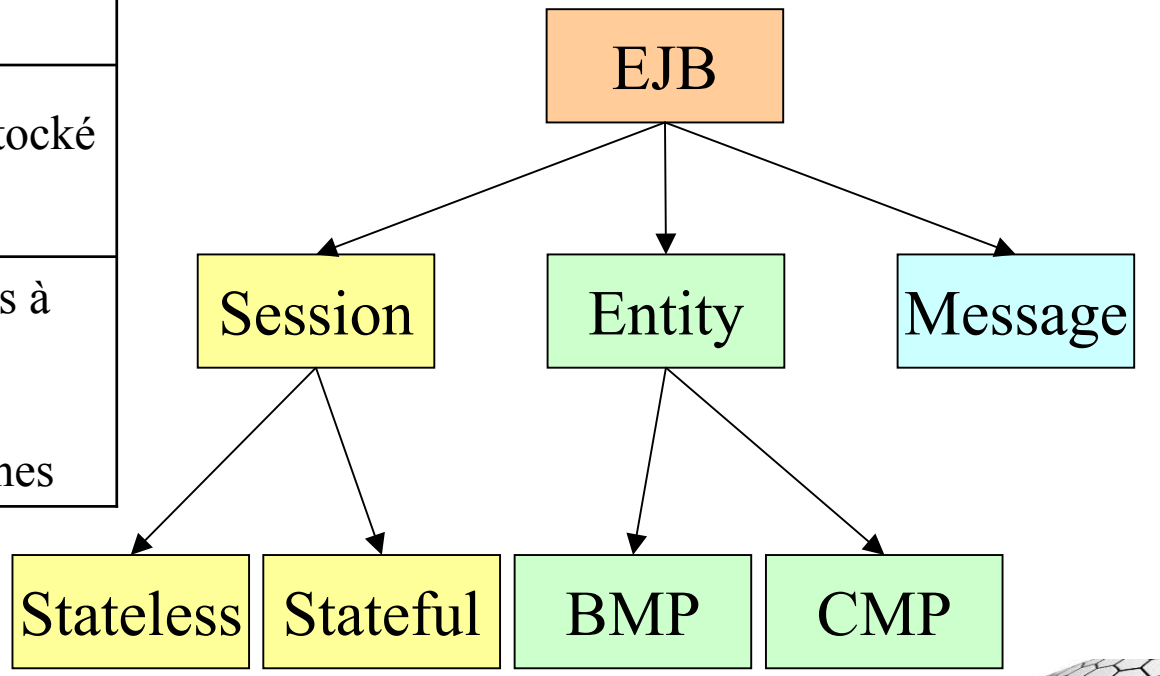


EJB



Types d'EJB

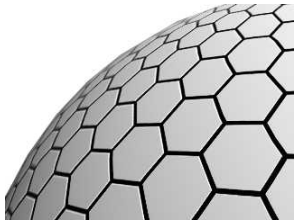
Type d'EJB	Utilité
Session	Effectue une tâche pour un processus client
Entity	Représente un objet métier stocké dans une base de données
Message	Ecoute des messages envoyés à travers l'interface JMS (Java Messaging Service), en les traitant de manière asynchrones



XML

XML/Web Services et J2EE

- ☑ Java XML Pack Summer 02 Release
  - Java API for XML Messaging (JAXM) v1.1
  - Java API for XML Processing (JAXP) v1.2
  - Java API for XML Registries (JAXR) v1.0_01
  - Java API for XML-based RPC (JAX-RPC) v1.0
  - SOAP with Attachments API for Java (SAAJ) v1.1
- ☑ Java API for XML Binding (JAXB)
- ☑ Implementing Enterprise Web Services
- ☑ Java APIs for WSDL
- ☑ Java Services Framework
- ☑ Web Services Security Assertions
- ☑ XML Transactioning API for Java (JAXTX)
- ☑ Java Process Component API (JPC)



Stockage

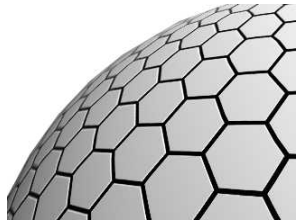
JDBC

- « Low-level » database access in Java
- Drivers avaible for all major database vendors, including MS SQL Server, Oracle,
- Connection Pooling
- Result set navigation, updating
- Batch updates
- Savepoints

JDO

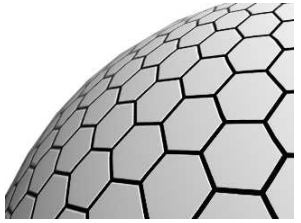
☑ What is JavaTM Data Objects (JDO)?

- JDO (Java Data Objects) is an API for transparent database access. The programmer can write code in the Java programming language that transparently accesses the underlying data store, without using database-specific code.



La fondation Apache (1)

- ☑ Offre d'excellents outils de développement open source en Java
- ☑ Quelques exemples :
  - **Ant**, outils de build, similaire aux Makefile mais en XML
  - **Tomcat**, container de servlet à la norme 2.3, utilisé dans l'implémentation de référence de J2EE
  - **Turbine**, framework de développement d'applications web
  - **Log4J**, librairie et outils pour afficher et gérer des messages de logs
  - **Lucene**, moteur de recherche en Java
  - **Struts**, implémentation MVC pour le développement rapide d'application web
- ☑ Certains de ces outils sont utilisés dans des environments de productions, voire même livrés avec des produits commerciaux (ie Borland Enterprise Server, JBuilder,…)



La fondation Apache (2)

- Projets XML de la fondation Apache
  - **AXIS -** An implementation of the SOAP ("Simple Object Access Protocol") submission to W3C.
  - **Batik -** A Java based toolkit for Scalable Vector Graphics (SVG).
  - **Cocoon -** XML-based web publishing, in Java.
  - **FOP -** XSL formatting objects, in Java.
  - **SOAP -** Simple Object Access Protocol.
  - **Xalan -** XSLT stylesheet processors, in Java and C++.
  - **Xerces -** XML parsers in Java, C++ (with Perl and COMbindings).
  - **Xindice -** A native XML database.
  - **XML-RPC -** A Java implementation of XML-RPC, a popular protocol that uses XML over HTTP to implement remote procedure calls.
  - **XML Security -** Create and verify arbitrary forms of XML Signatures.

Autres standards J2EE

JMS - Java Message Service

JNDI - Java Naming and Directory Interface

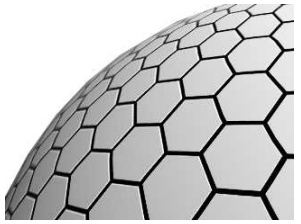
JTA - Java Transaction API

JavaMail API

JCA - J2EE Connector Architecture

JAAS - Java Authentification and Autorisation Service

Et encore beaucoup d'autres « J* »



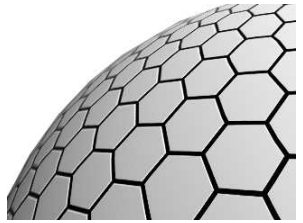
What's coming next in J2EE?

Peer to Peer – Projet JXTA

Java Server Faces

Portlet API

Content API



J2EE Moins et Plus

Avantages et Désavantages de J2EE

Moins et Plus de J2EE

Moins:

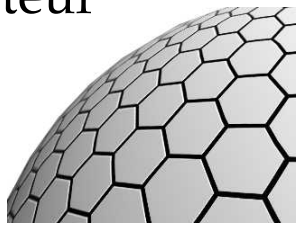
- ☑ Indépendance par rapport au vendeur

  - Vraiment? Interprétations différentes des standards et lock-in par rapport aux couches logiciels supplémentaires

- ☑ Encore des manquements pour certaines fonctionnalités clés (ex: ACL; batch jobs;…)
- ☑ Problématique de formation et d'utilisation (valable pour tous les frameworks)

  La plateforme est immensément riche mais qui peut se vanter de la connaître à fond. Pas assez de « guidance » et d'outils WYSIWYG

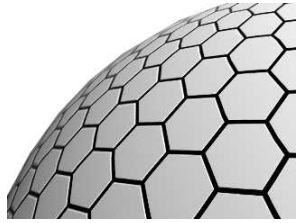
- ☑ Manquement actuel au niveau du front-end (particulièrement Web)
- ☑ Pris par surprise par .NET
- ☑ Java Community Process multi-vendeurs sont parfois trop lents et pas assez ambitieux (politique du compromis et du plus petit dénominateur commun)



Moins et Plus de J2EE

Plus:

- ☑ Politique du choix (choix du vendeur, du framework, des outils, des solutions)
- ☑ Idéal pour des environnements hétérogènes
- ☑ Mature et éprouvé (scalable, available, reliable)
- ☑ Evolution du framework issue du travail d'un consortium ouvert (JCP) où tous le monde peut participer
- ☑ Forte culture Java existante - Forte pénétration chez les développeurs seniors et les architectes logiciels
- ☑ Force de Java dans le domaine éducatif (formation des futurs ingénieurs sur plate-forme J2EE)
- ☑ Bonne image de Java dans la presse (côté anti-monopolistique, anti-grand méchant loup)
- ☑ Plus longue expérience côté Serveur, Transactionnel et Web
- ☑ Devant en matière de « Wireless » et de « Peer-to-Peer »



Conclusion Intro Général

1) J2EE n'est plus l'unique solution pour le développement en entreprise

   a) « Competition is good » bien qu'il y ait toujours eu une vive compétition au sein de Java entre les différents vendeurs.

   b) Regret pour les développeurs qu'il y ait deux plateformes très, très similaires, qui pour des raisons politiques, vont entraver les initiatives d'intégration en entreprise.

2) Une architecture orientée composants réutilisables est devenu une base minimum de développement

   a) Rôle de plus en plus primordial de l'Architecte (choix des standards et assemblage des meilleurs composants)

   b) Le vaste choix de solutions offert en Java est excellent mais demande plus d'analyse et une meilleure compréhension des problématiques techniques

3) Choix d'une plate-forme= facteur stratégique déterminant

   Dans notre cas, si .Net prend la moitié des parts de marché = moitié de clients potentiels en moins pour notre progiciel. Idem du côté de l'acheteur (deux équipes)