
Présentation de Java

Stéphane Frénot – Frédérique Laforest – Frédéric Le-Mouël
2006

INSA lyon

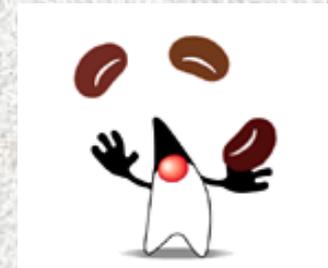
Département Télécommunication Services & Usages

Langage de programmation

Write once, run anywhere

Java ?

- Printemps 90 : Naughton, Gosling et Sheridan :
 - *"Le consommateur est le centre du projet, il faut construire un environnement de petite taille avec une petite équipe et intégrer cet environnement dans une nouvelle génération de machines : des ordinateurs simples pour des gens normaux."*
- Printemps 91 : Microprocesseur grand public.
 - La "Green Team" prototype une machine de pilotage de l'électroménager
- Août 91 : Gosling développe Oak
- Août 92 : -----duke----->



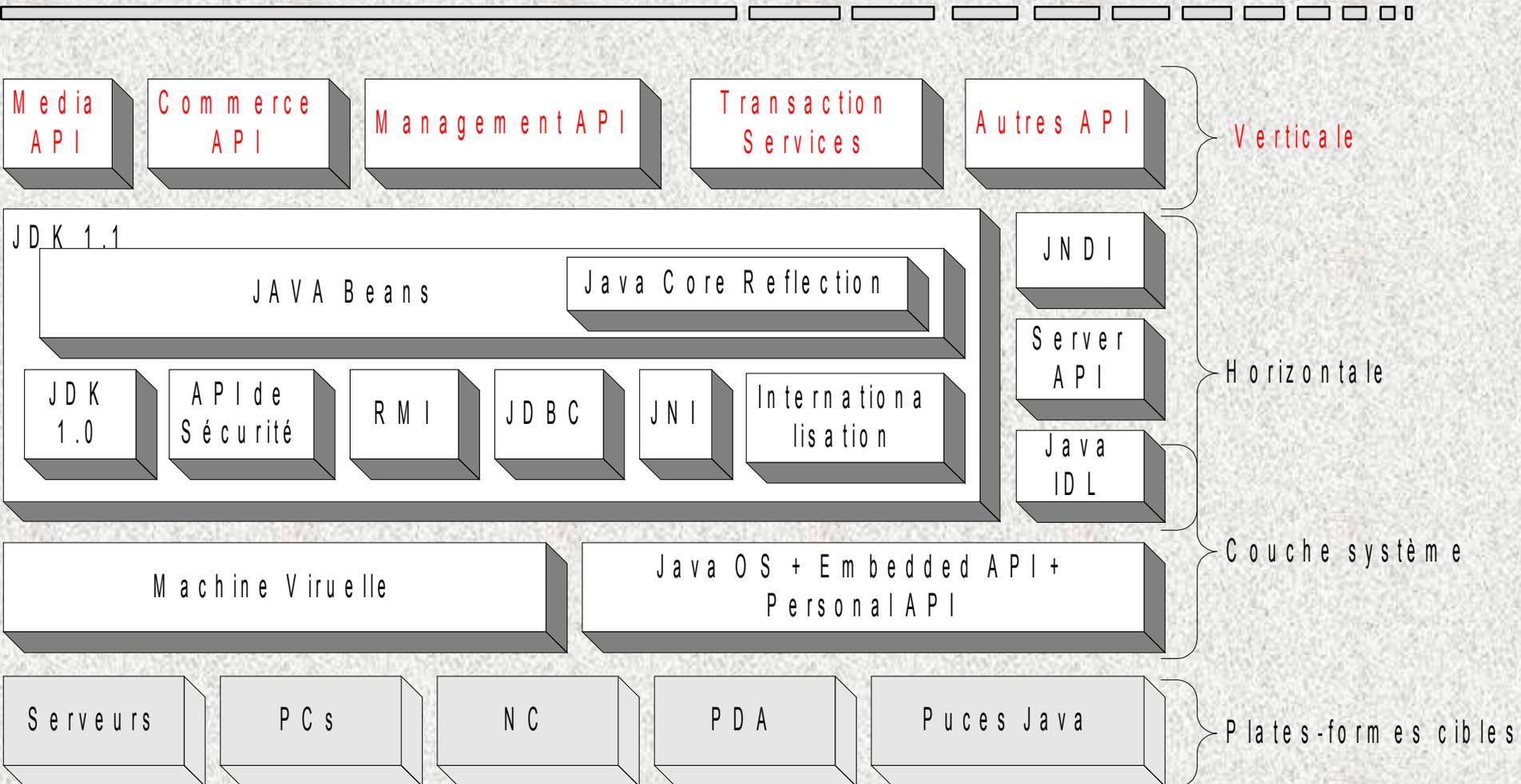
Java ?

- Eté 1993 : Sté. "First Person" est en train de couler
 - Eric Schmidt (Sun) demande une adaptation au Net
 - Gosling : travaille sur le code
 - Naughton : cherche une application stratégique
- Janvier 1995
 - Oak ==> Java, HotJava
- Août 1995 - Première licence sur Netscape
- Janvier 1996 - JDK 1.0.1
- Fév. 97 - JDK 1.1
- Jan 99 - JDK 2.0 (aka 1.2)
- 2003 J2se 1.4, J2ee, J2me (Standard, Entreprise, Micro)

Ce que c'est !

- Une architecture technique
- Un langage OO
- Une bibliothèque de 5000 éléments

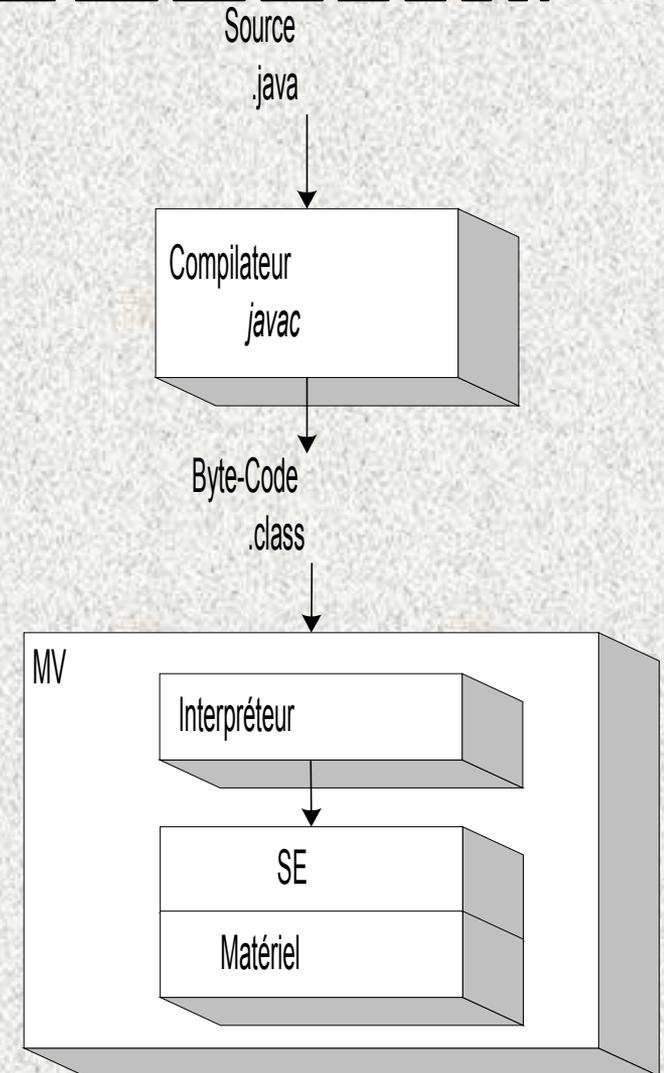
Plateforme Java : architecture



Java Client/Serveur : Nicolas, Avare, Najman - Eyrolles

Machine Virtuelle : Emulateur

- Byte-Code/P-Code/J-Code
 - opcode : 1 octet pour l'instruction
 - 0,n opérandes
- MicroProcesseur logiciel
 - Jeu d'instructions
 - Registres (pc, optop, frame, vars)
 - Pile, Heap
 - Ramasse-miettes
 - Espace de stockage des méthodes
 - Tas de constantes
- \Rightarrow Compilé ou Interprété



Machine Virtuelle

- Système d'exploitation
- Isolation
- Autonomie

====> Avantages ?

Java est portable

- Le compilateur Java génère du *byte code*.
- La *Java Virtual Machine* (JVM) est présente sur Unix, Win32, Mac, OS/2, Netscape, IE, ...
- Le langage a une sémantique très précise.
- La taille des types primitifs est indépendante de la plate-forme.
- Java supporte un code source écrit en Unicode.
- Java est accompagné d'une librairie standard.

Java est distribué

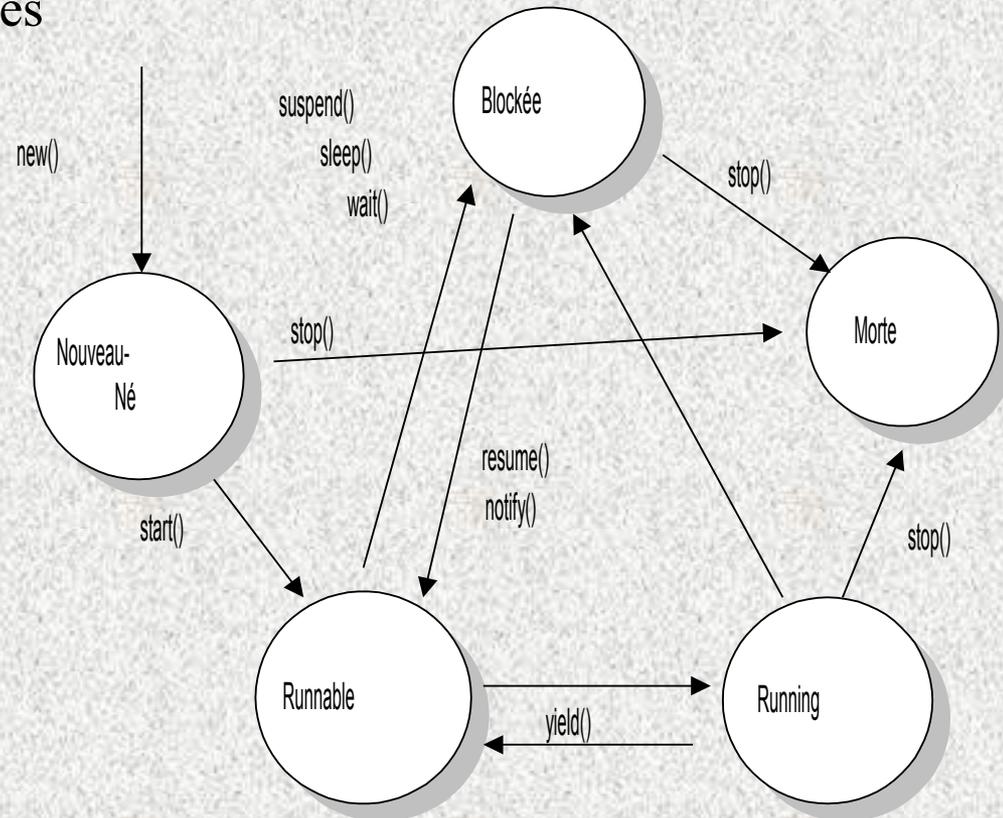
- API réseau (java.net.Socket, java.net.URL, ...).
- Chargement / génération de code dynamique.
- Applet.
- Servlet.
- Remote Method Invocation : RMI
- Interconnexion sur CORBA

Java est robuste

- A l'origine, c'est un langage pour les applications embarquées
- Gestion de la mémoire par un garbage collector (ramasse miettes)
- Pas d'accès direct à la mémoire
- Mécanisme d'exception
- compilateur contraignant (erreur si exception non gérée, si utilisation d'une variable non affectée, ...)

Java est multi-threadé

- Exécution de tâches en //
- Mémoire, Code et Ressources partagés
- Economie de ressources
- Un thread \sim méthode qui rend immédiatement la main
- Exemple événements (IHM, gc)
- + priorités
- + synchronisation
 - (moniteur, synchronized)
- Implantation dépendante du SE



Java est sécurisé

- Indispensable avec le code mobile.
- Pris en charge dans l'interpréteur.
- Trois couches de sécurité :
 - *Verifier* : vérifie le *byte code*.
 - *Class Loader* : responsable du chargement des classes.
 - *Security Manager* : accès aux ressources.
- Code certifié par une clé.

Java est semi-réflexif

- Le langage s'autodécrit
- Les éléments du langage sont pilotable de l'extérieur
- Possibilité de faire de la meta-programmation
- La machine virtuelle est pilotable de l'extérieur

API du JDK (Paquetages)

- java.lang : classes de bases (+reflect)
- java.io : entrées/sorties
- java.util : utilitaires (structures, dates, events) (+zip)
- java.net : réseau
- java.applet : gestion des applets
- java.awt : interface graphique (image, +datatransfert, +event)
- java.beans : définition de composants réutilisables
- java.math : entier de taille variable
- java.rmi : invocation distante (+dgc, +registry, +server)
- java.security : (+acl, +interfaces)
- java.sql : jdbc ...
- java.text : traduction, chaine=f(langue)

Architecture technique : Conclusion

- Notion de bac à sable
 - Isolation
 - Robustesse
 - En pleine évolution
- Utilise les ressources du système d'exploitation sous-jacent
 - Accès aux ressources (réseau, disque, mémoire...)
 - Mais ...

Le kit java

- JDK vs JRE
 - Java Development Kit / Java Runtime Environment
- Versions du jdk
 - 1.1 -> version de démonstration
 - 1.2 -> 1ère version utilisable (interfaces graphiques)
 - 1.3 -> 1ère version industrielle
 - 1.4 -> Version courante (compilation JIT)
 - 1.5 -> Version avancée (grande taille)
 - 40 Mo de bibliothèques standards compressées
 - 13000 classes
 - 100k la machine virtuelle
- N'est pas open-source

La famille java

- **Java 2 SE**
 - java 2 standard edition (à partir du jdk1.4)
 - Kit et run-time standard java
- **Java 2 ME**
 - Java 2 Micro-Edition pour environnements contraints
 - CLDC / CDC
 - Connected Limited Device Configuration
 - Connected Device Configuration
 - Midp 2.0 : couche de spécialisation constructeur
- **Java 2 EE**
 - Java 2 Enterprise Edition
 - Principalement api EJB

L'Iceberg Java

