

Derrière toi !



Une machine virtuelle !

Yann-Gaël Guéhéneuc

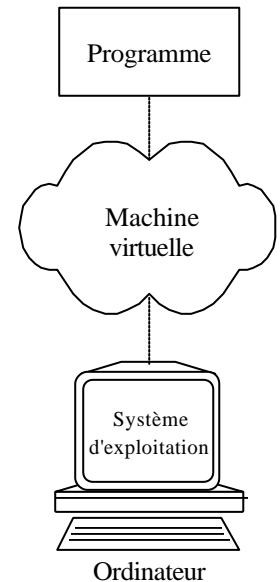
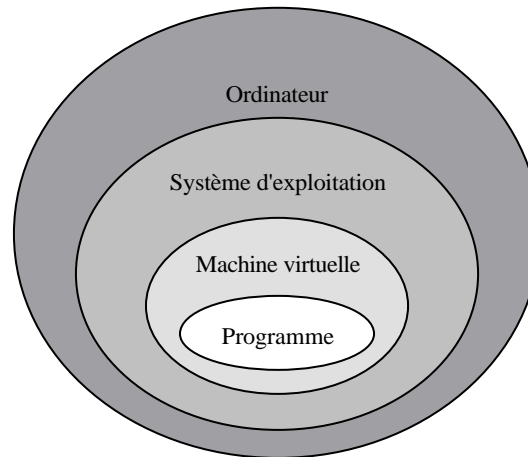
Qu'est-ce que c'est ?

■ Une machine virtuelle c'est :

- Un programme qui s'interpose entre un programme et un système d'exploitation
- Un interpréteur qui interprète un programme pour un système d'exploitation
- Un compilateur qui compile à la volée un programme pour un système d'exploitation

En bref ?

■ Une machine virtuelle c'est :



■ Une machine virtuelle ce n'est pas :

- Inutile



Pourquoi ?

n langages + m systèmes d'exploitation =
 $n * m$ compilateurs

n langages + 1 machine virtuelle + m
systèmes d'exploitation =
 n compilateurs

(+ m machines virtuelles gratuites...)



Comment ?

■ Turing

- Un ensemble fini d'états, d'instructions, et un alphabet fini

■ Machine à registre

- Des registres, un chemin de donnée, un contrôleur

■ Machine à pile

- Une pile, des opérations, un contrôleur (?)

Et aujourd'hui ?

■ La machine virtuelle de Java :

– Par méthode :

- Un cadre

– Par classe :

- Un paquet

– Par *threads* :

- Une pile
- Un registre

– Pour tous les *threads* :

- Un tas
- Une zone « méthodes »
- Une pile « méthodes natives »



Et ce n'est pas tout ?

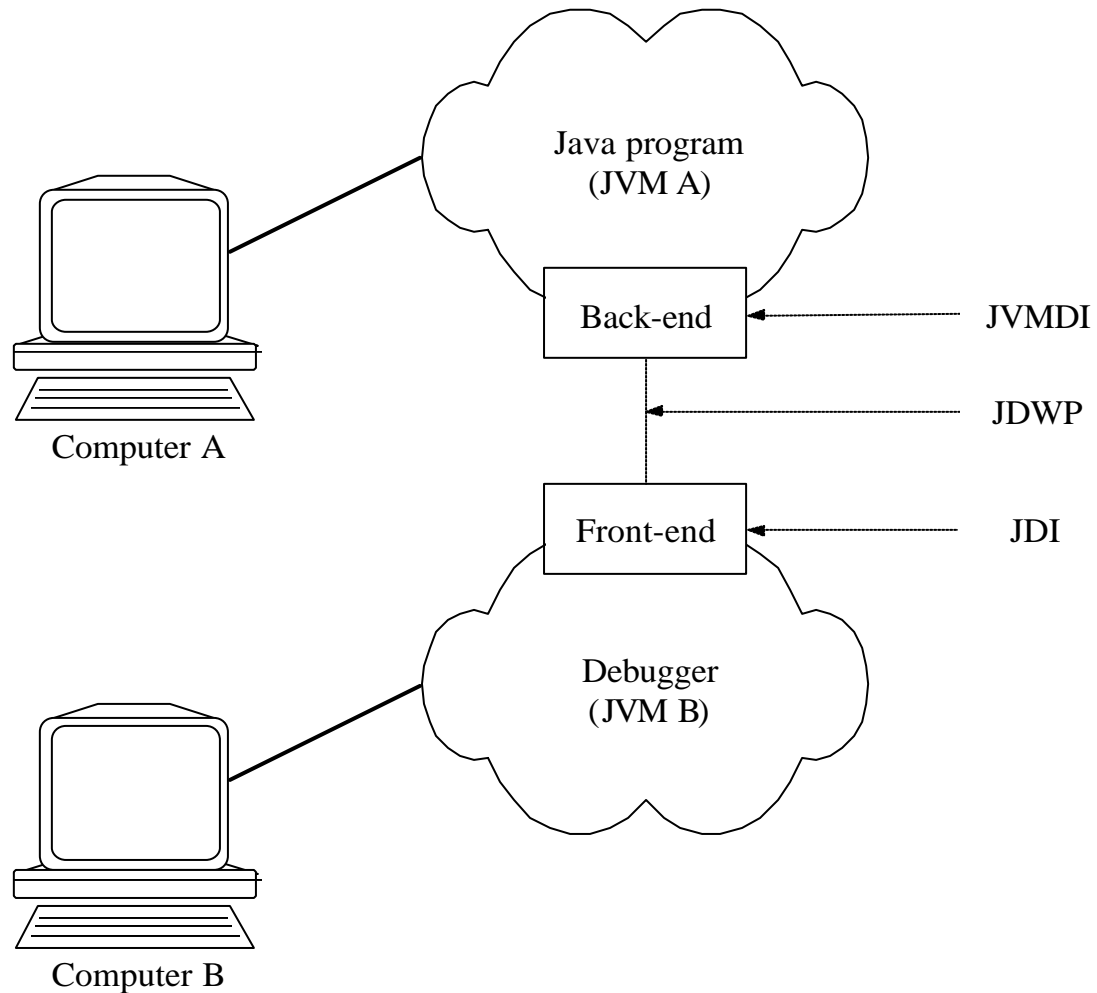
- La machine virtuelle de Java :
 - Java Native Interface
 - Garbage collector
 - Java Virtual Machine Profiler Interface
 - Java Platform Debug Architecture



JPDA ?

- The Java Platform Debug Architecture
 - Tout pour déboguer / contrôler un programme qui marche dans une machine virtuelle Java
 - Trois niveaux :
 - Java Debug Interface
 - Java Debug Wire Protocol
 - Java Virtual Machine Debug Interface

JPDA ?





JPDA ?

■ Expérimentations :

- Environnements de développement :
 - IBM Eclipse
 - Borland JBuilder
 - Sun Forte
- Pas encore testé personnellement !

■ Espoirs :

- Espionner la vie et la mort des instances