Laboratoire Bordelais de
Recherche en Informatique

Équipe : Systèmes et
Objets Distribués

La technologie Java Card™

Damien Sauveron
sauveron@labri.u-bordeaux.fr
http://dept-info.labri.u-bordeaux.fr/~sauveron

SERMA TECHNOLOGIES

8 novembre 2002

Plan

Qu'est ce que Java Card ?

Technologie permettant de faire fonctionner des applications écrites en langage Java pour :

- ☞ carte à puce
- ☞ autres périphériques à mémoire limitée

La technologie Java Card définit une plate-forme sécurisée pour cartes à puce, portable et multi-application qui incorpore beaucoup des avantages du langage Java.

Historique

- En Novembre 1996, un groupe d'ingénieurs de Schlumberger cherche à simplifier la programmation des cartes à puce tout en préservant la sécurité.
  $\Longrightarrow$ la spécification Java Card 1.0
- En Février 1997, Bull et Gemplus se joignent à Schlumberger pour cofonder le *" Java Card Forum "*.
- En Novembre 1997, Sun présente les spécifications Java Card 2.0.

Historique

➤ En Mars 1999 sort la version 2.1 des spécifications Java Card. Elles consistent en trois spécifications :

- The Java Card 2.1 API Specification.
- The Java Card 2.1 Runtime Environment Specification.
- The Java Card 2.1 Virtual Machine Specification.

Contribution la plus significative :

- Définition explicite de la machine virtuelle de la Java Card.
- Le format de chargement des applets.

Historique

- En Mai 2000, sort une petite correction $\Longrightarrow$ version 2.1.1
- En Octobre 2000, plus de 40 entreprises ont acquis la licence d'exploitation de la technologie Java Card.
- En juin 2002, spécifications Java Card 2.2.

Les avantages de la technologie Java Card

La facilité de développement des applications grâce :

- à la programmation orientée objet offerte par Java
- à l'utilisation des environnements de développement existants pour Java.
- à une plate-forme ouverte qui définit des APIs et un environnement d'exécution standard.
- à l'encapsulation de la complexité fondamentale du système des cartes à puce.

Les avantages de la technologie Java Card

La sécurité grâce :

- à plusieurs niveaux de contrôle d'accès aux méthodes et aux variables (`public`, `protected`, `private`).
- à un langage fortement typé.
- à l'impossibilité de construire des pointeurs.
- à un " firewall "

Les avantages de la technologie Java Card

L'indépendance au hardware réalisée grâce au langage Java
$\Longrightarrow$ " Write Once, Run Anywhere "

La capacité de stockage et de gestion de multiples applications.
$\Longrightarrow$ possiblité de mises à jour des applications de la Java Card sans avoir besoin de changer de cartes.

La ***compatibilité*** avec les standards existants sur les cartes à puces.

Présentation de son architecture

Contraintes mémoires : 1Ko de RAM, 16Ko d'EEPROM et de 24Ko de ROM

$\Longrightarrow$ **Problèmes** pour construire une carte Java

$\Longrightarrow$ **Solutions** :

- ☞ ***supporter seulement un sous-ensemble des caractéristiques du langage Java***
- ☞ ***découper la machine virtuelle Java*** (JCVM : Java Card Virtual Machine) ***en deux parties***
  - une partie en dehors de la carte
  - une partie sur la carte

Présentation de son architecture

$\Longrightarrow$ **Problème :**

Beaucoup de tâches ne sont plus vérifiées à l'exécution.

$\Longrightarrow$ **Solution :**

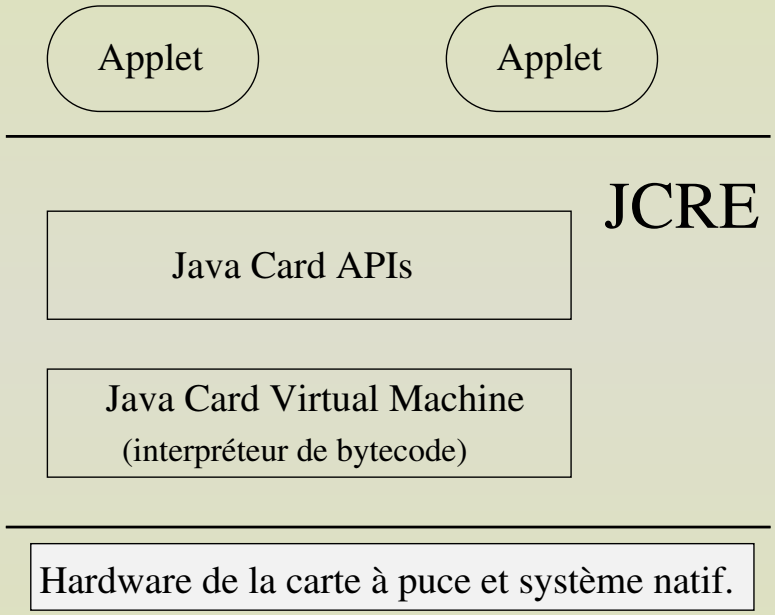
Un environnement d'exécution, ***le JCRE*** (Java Card Runtime Environment) ***chargé de fournir des mécanismes sécuritaires***.

Présentation de son architecture

- ***Le JCRE encapsule la complexité fondamentale*** du système des cartes à puce.
- A cause du découpage de la JCVM, ***la plate-forme est distribuée*** entre la carte à puce et la machine de développement ***dans le temps et dans l'espace***.

Présentation de la Java Card

Présentation de son architecture



Cette architecture de la technologie Java Card est définie par :

- The Java Card 2.1 Virtual Machine Specification.
- The Java Card 2.1 Runtime Environment Specification.
- The Java Card 2.1 API Specification.

Présentation de son architecture

Caractéristiques Java supportées

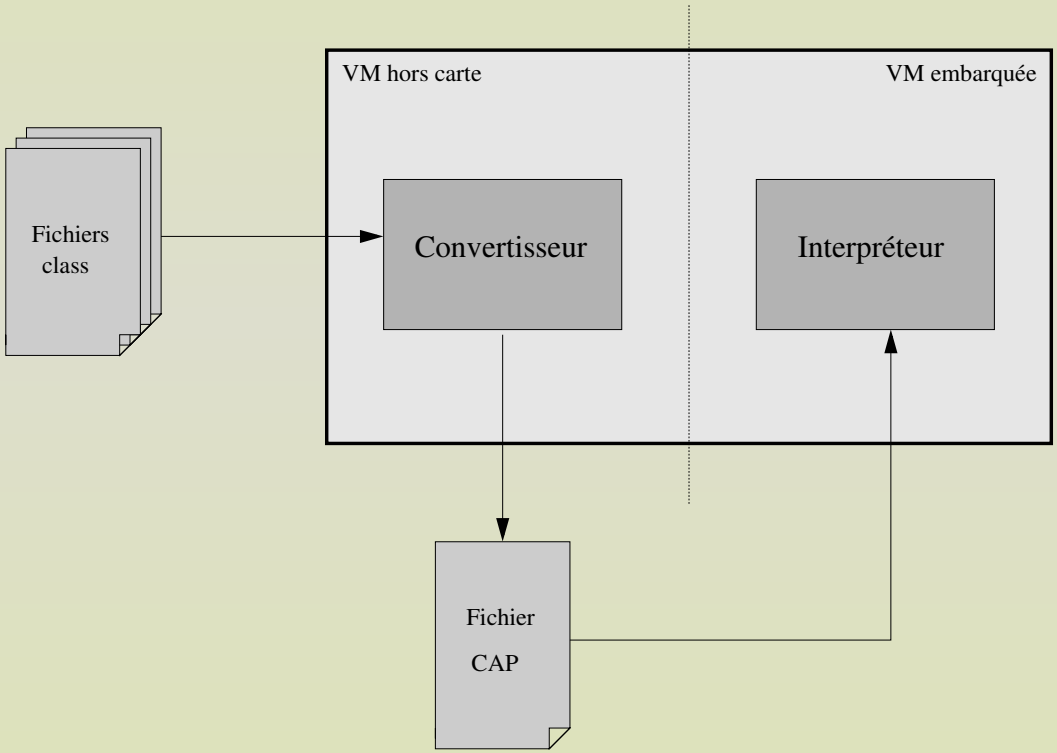
- ✓ Type simple de donnée de petite taille : `boolean`, `byte`, `short`
- ✓ Tableau à une dimension
- ✓ Paquetage Java, classes, interfaces et exceptions
- ✓ Caractéristiques orientées objet : héritage, méthodes virtuelles, surcharge et création dynamique d'objet, contrôle d'accès
- ✓ Le mot clé `int` et le support des entiers sur 32 bits sont optionnels

Présentation de son architecture

Caractéristiques Java non supportées

- ✗ Type simple de donnée de grosse taille : `long`, `double`, `float`
- ✗ Tableau plusieurs dimensions
- ✗ Caractères et chaines
- ✗ Chargement dynamique des classes
- ✗ Security Manager
- ✗ Ramasse-miettes et finalisation
- ✗ Threads
- ✗ Serialisation d'objet
- ✗ Clonage d'objet

La machine virtuelle de la Java Card



Les deux parties implémentent toutes les fonctions d'une machine virtuelle.

La machine virtuelle de la Java Card

Fichier CAP

Le fichier CAP est le format standard de fichier pour la compatibilité binaire de la plate-forme Java Card.

Un fichier CAP contient une représentation binaire exécutable des classes d'un paquetage Java.
Un fichier CAP est un fichier JAR qui contient un ensemble de composants.
Chaque composant décrit un aspect du contenu d'un fichier CAP comme les informations sur les classes, les bytecodes exécutables, les informations de " linkage ", les informations de vérifications, etc.

La machine virtuelle de la Java Card

Fichier `export`

Un fichier `export` contient les informations publiques sur les API pour un paquetage de classes complet.

La machine virtuelle de la Java Card

Convertisseur Java Card

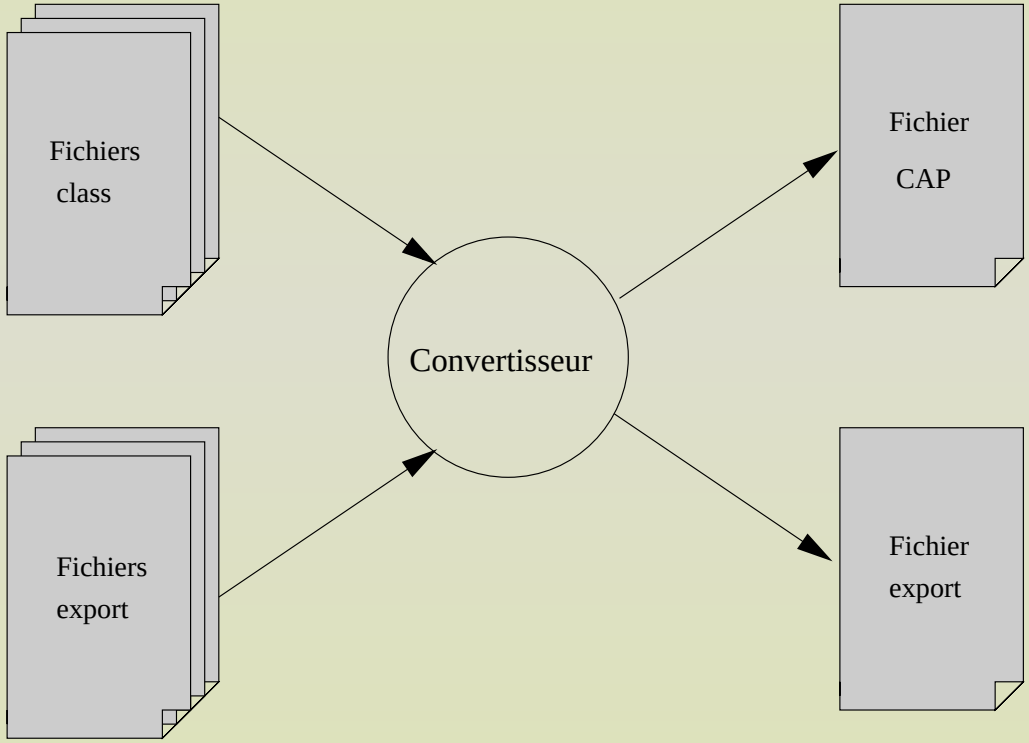
Le convertisseur réalise les taches que la machine virtuelle Java sur une station de travail doit réaliser au chargement des classes :

- ***Vérifie*** que le chargement des ***images des classes Java*** sont bien formées.
- ***Contrôle des violations*** du langage Java Card.
- ***Réalise des initialisations*** de variables `static`.

La machine virtuelle de la Java Card

- ***Résout les références symboliques*** aux classes, méthodes et champs dans la forme la plus compact qui peut être traitée efficacement sur la carte.
- ***Optimise le bytecode*** en tirant avantage des informations obtenu au chargement des classes et au “ linkage ”.
- ***Alloue l'espace et créer les structures de données de la machine virtuelle*** pour représenter les classes.

La machine virtuelle de la Java Card



La sortie produite par le convertisseur est un fichier CAP et un fichier `export` pour le paquetage converti.

La machine virtuelle de la Java Card

Interpréteur Java Card

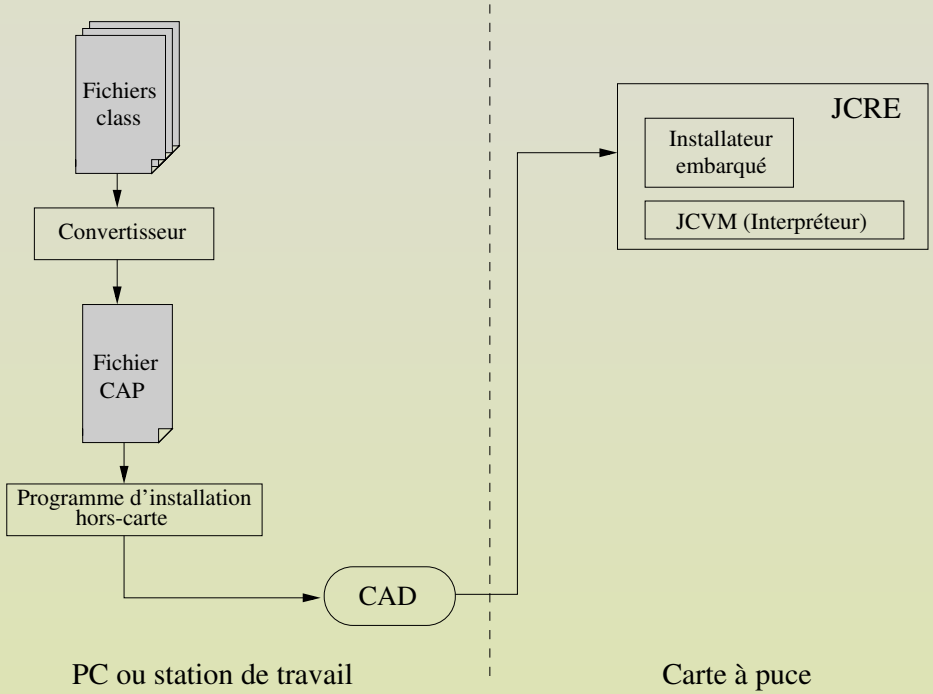
L'interpréteur Java Card fournit le ***support d'exécution*** du modèle du langage Java qui ***autorise une indépendance au hardware*** du code de l'applet.

L'interpréteur réalise les taches suivantes :

- ***Exécute les instructions du bytecode*** et ultimement les exécutions des applets.
- ***Contrôle les allocations de mémoire et les créations d'objets.***
- Joue un rôle crucial pour s'assurer de la ***sécurité*** lors ***de l'exécution.***

La machine virtuelle de la Java Card

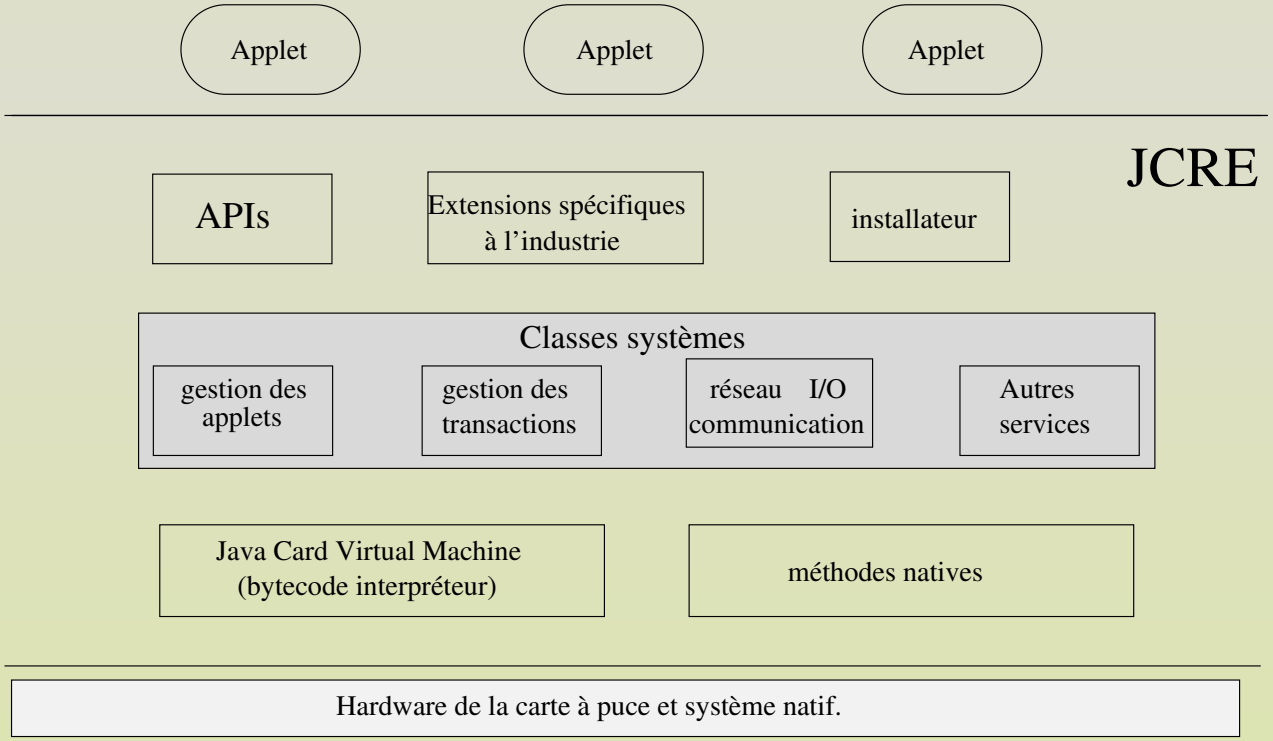
Installateur Java Card et programme d'installation hors carte



Présentation de la Java Card

L'environnement d'exécution de la JC

Le JCRE est responsable de la gestion des ressources de la carte, communication réseau, exécution des applets, du système de la carte et de la sécurité des applets.



L'environnement d'exécution de la JC

Le cycle de vie du JCRE

Le JCRE est initialisé à la phase d'initialisation de la carte (seulement une fois dans le cycle de vie de la carte).

Le cycle de vie du JCRE est le même que le cycle de vie de la carte.

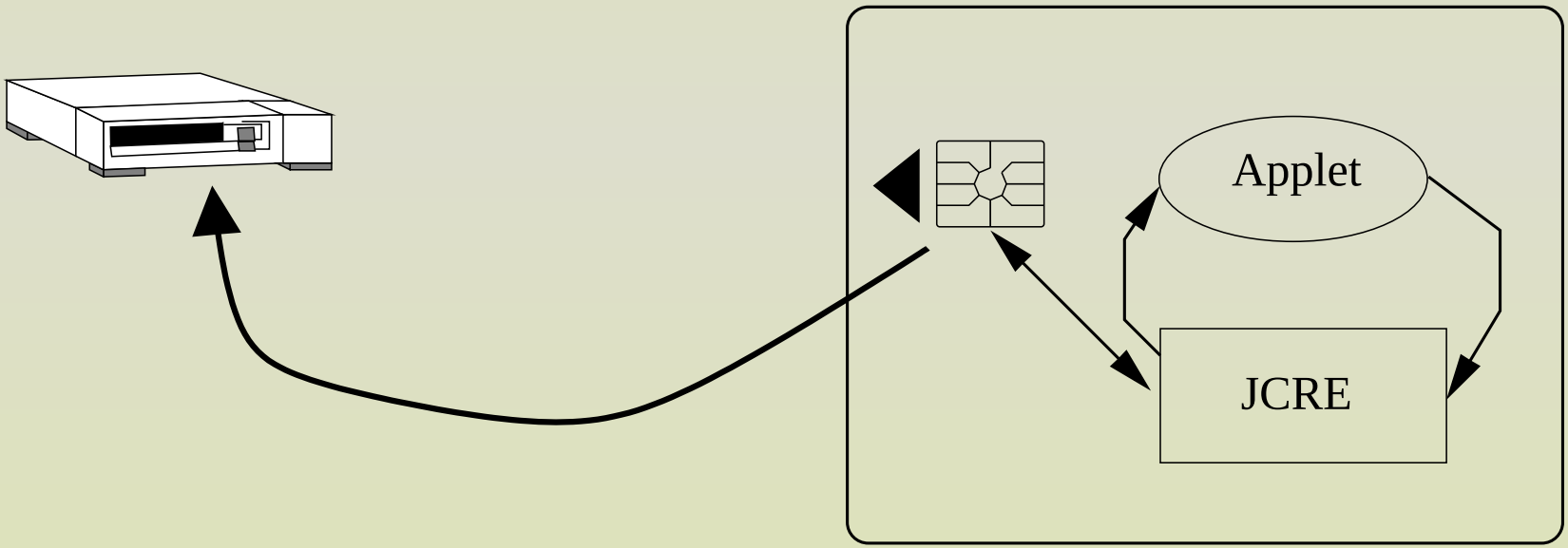
L'environnement d'exécution de la JC

- Quand la ***tension est enlevée***, la ***machine virtuelle*** est seulement ***suspendue***.
  L'état du JCRE et des objets créés sur la carte est préservée.
- A la prochaine ***remise sous tension***, ***le JCRE relance l'exécution de la machine virtuelle*** en chargeant les données depuis la mémoire persistante.
- Durant le reset, si une transaction n'a pas été terminée, le JCRE réalise tous les nettoyages nécessaires pour remettre le JCRE dans un état cohérent.

L'environnement d'exécution de la JC

Comment le JCRE opère durant les sessions CAD ?

le JCRE opère comme une carte à puce typique.



L'environnement d'exécution de la JC

Les caractéristiques du JCRE

- ***Objets persistants et temporaires***
  Objets persitants existent à travers les sessions CAD.
  Objets temporaires contiennent des données temporaires qui ne persistent pas au travers des sessions CAD.
- ***Atomicité et transactions***
  Chaque opération d'écriture de la JCVM est ***atomique***.
  Le champ mis à jour prend soit la nouvelle valeur soit il est restauré à la valeur précédente.
  Une transaction est un bloc d'opérations atomiques.

L'environnement d'exécution de la JC

☞ ***" Firewall "des applets et les mécanismes de partage***

Le " firewall " isole les applets à l'intérieur de leur espace (contexte). Si les applets ont besoin d'accéder à des données ou à des services du JCRE, la machine virtuelle autorise de telles possibilités à travers des ***mécanismes sécurisés de partage.***

Les APIs Java Card

Les APIs Java Card sont un ensemble de classes optimisées pour la programmation des cartes à puce en accord avec le modèle ISO7816.

Le paquetage `java.lang`

Le paquetage Java Card java.lang est un sous ensemble strict de son équivalent le paquetage java.lang sur la plate-forme Java.

Le paquetage `javacard.framework`

Ce paquetage fournit la structure des classes et des interfaces pour le noyau fonctionnel des applets Java Card.

Les APIs Java Card

Le paquetage `javacard.security`

Le paquetage javacard.security fournit une architecture aux fonctions cryptographiques supportées sur la plate-forme Java Card.

Le paquetage `javacardx.crypto`

Le paquetage javacardx.crypto est un paquetage d'extension.

Conclusion et perspectives

- La technologie Java Card révolutionne le monde de la carte en banalisant sa programmation.
- Elle offre la possibilité de réduire les coûts de développement et d'évaluation.
- Problème : la carte multi-applicative fait peur aux industriels et aux utilisateurs (verrou psychologique).
- Solution :
  - utilisation du standard Open Platform pour maintenir un niveau de sécurité supplémentaire afin de rassurer les industriels.
  - utilisation de Java Card comme plate-forme mono-applicative afin de ne pas inquiéter les consommateurs.