

La carte à microprocesseur

De la carte à mémoire à Java Card

Un exemple de système embarqué

Pierre.Paradinas @ cnam.fr
Cnam/Cedric
Systèmes Enfouis et Embarqués (SEE)

1

Plan Java Card

- La carte à “mémoire” histoire d’une technologie
- La carte à microprocesseur
 - Architecture matériel
 - Normes et standards
 - Domaine d’applications
 - **Java Card**
 - Autre modèle .Net
 - Nouveaux modèles applicatifs
- Enjeux et perspectives de l’industrie

Java Card - Introduction

- Besoin marché vers des systèmes “programmables”
 - Recherche d'un meilleur T2M
 - Recherche de solution “évolutive” (dépasser la ROM)
- Avant d'arriver à cela les applications sont :
 - Longues à développer (cycle de deux ans !)
 - Statiques dans leurs ensembles de fonctions (ROM)
- Des tentatives...
- 1ère version : octobre 1996, démarrage et produit réel en 1998, une réalité industrielle à partir de 2000. En 2004, le nombre de Java Card vendu atteindra le milliard.

Etapes du développement de l'industrie

- La carte à microprocesseur et les grandes étapes de développement de la technologie (marché et technique) :
 - Les pionniers (1975-1985) : de(s) idées aux premiers produits, les bases technologiques sont établies,
 - 1985-1995 : les marchés et les déploiement importants sont là (CB & GSM), la technologie du départ est améliorée et renforcée, des limites apparaissent comme le besoin de flexibilité qui annonce Java Card,
 - 1995-2005 : explosion du marché (EMV, ID), qui s'accompagne d'un nouveau paradigme les cartes évolutives basées sur Java Card.
 - 2005-?? : la carte devient un élément du “réseau”.

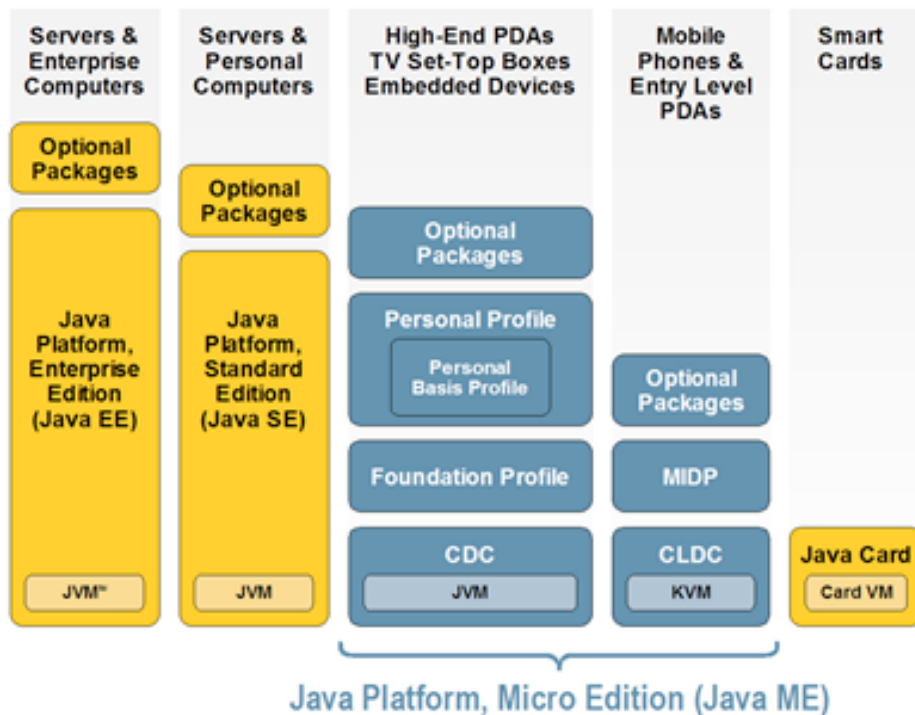
L'arrivée de la Java Card

- Novembre 1996, la première proposition d'utilisation de Java pour les cartes est faite par une équipe de Schlumberger (Austin),
 - Proposition de Java Card API permettant la programmation en Java de/dans la carte
 - C'est la Java Card 1.0
- Bull, Gemplus et Schlumberger créent le Java Card Forum
 - Le JCF discute et propose à Sun des spécifications.
- Novembre 1997, publication de la spécification Java Card 2.0
 - Gemplus démontre en octobre/novembre CASCADE, le premier chip RISC 32 bit (ARM 7) avec de la mémoire Flash, "une/son" implémentation de la Java Card 2.0 et des DMIs (Direct Method Invocation)...

Les évolutions de la 2.x

- La version 2.0 de la Java Card Specification introduit :
 - Un environnement d'exécution
 - La possibilité d'écrire des applets avec une approche OO,
 - même si le format de chargement n'est pas encore spécifié...
- Mars 1999, la version 2.1 qui contient 3 parties est publiée :
 - Java Card API Specification
 - Java Card Runtime Environment Specification
 - Java Card Virtual Machine Specification

Un élément de la technologie Java



P. Paradinas - Cnam - 2005/2006

7

A propos du modèle de licence...

- La spécification est disponible sur :
 - <http://java.sun.com/products/javacard/>
- Vendre des cartes (avec ou sans logo) et afficher une compatibilité avec la technologie implique d'être licencié **Java Card Technology**;
 - ce qui donne accès à :
 - Une implémentation de référence
 - La suite de test de compatibilité
 - Du support spécifique



P. Paradinas - Cnam - 2005/2006

8

A propos du modèle de licence...

JAVA AUTHORIZED LICENSEES OF JAVA CARD TECHNOLOGY

-  The companies listed below have licensed Java Card technology from Sun Microsystems. Only Java Card licensees can ship products that bear the "Java Powered" logo and claim compatibility with the Java Card Platform specification and Java Card TCK.

 ARM, Aspects, Axalto, CCL/ITRI, Fujitsu, Gemplus, Giesecke & Devrient GmbH, GoldKey Technology, HiSmarTech, I'M Technologies Ltd., IBM, incard, KEBTechnology, Logos Smart Card, Microelectrónica Española, Oberthur Card Systems, ORGA Kartensysteme, SAGEM, Sermepa, Setec, Sharp, Smart Card Laboratory Inc., SSP Solutions, Inc., STMicroelectronics, Toppan Printing, Trusted Logic.

- D'après <http://java.sun.com/products/javacard/licensees.html> (le 18/11/05)

Avantages de la technologie

-  Interoperabilité des applets
-  Sécurité (sécurité du langage, évaluation,...)
-  Multi-application
-  Dynamique
-  Ouverte et compatibilité

Le Java Card Forum

- Association regroupant les fabricants de silicium, les encarteurs et des clients
- Promouvoir la solution de la Java Card
- Définir des choix technologiques puis les proposer à Sun qui en fait le “standard”. Processus différent du JCP (Java Community Process : <http://jcp.org>).



[http://
www.javacardforum.org](http://www.javacardforum.org)

Introduction à la technologie Java Card

- Prise en compte des architectures matérielles des cartes dont les tailles sont très réduites : moins de 1Ko RAM, 24-48 Ko de ROM et 8 à 16 Ko NVM (EEPROM).
- Pour “mettre la technologie Java dans une carte”, les choix suivants sont faits :
 - Réduire les fonctionnalités du langage
 - “Distribuer le modèle” la VM entre le “in-card” et le “off-card”
- 3 parties :
 - Java Card API Specification
 - Java Card Runtime Environment Specification
 - Java Card Virtual Machine Specification

Java Card Langage et Java Langage

Caractéristiques supportées	Caractéristiques non supportées
booléen, octet, court	long double, flottant
tableau à une dimension	caractères et chaînes
paquetage Java, classes, interface et exceptions	tableau à plusieurs dimensions
héritage, méthode virtuelle, surcharge et création d'objet	chargement dynamique de classe
mot clé <code>int</code> et entier 32-bit sont optionnel	gestionnaire de sécurité (security manager)
	pas de ramasse miettes
	threads
	sérialisation

P. Paradinas - Cnam - 2005/2006

13

Java Card API 2.1

3 paquetages de référence

-  `java.lang`
-  `javacard.framework`
-  `javacard.security`

Une extension

-  `javacardx.crypto` (liée à des problèmes d'exportation de produit)

P. Paradinas - Cnam - 2005/2006

14

java.lang

- Strict sous ensemble de java.lang
- La classe `Object` est la racine de la hiérarchie de classe Java Card
- Classes supportées : `Object` and `Throwable`
- La classe `Object` n'a qu'un constructeur

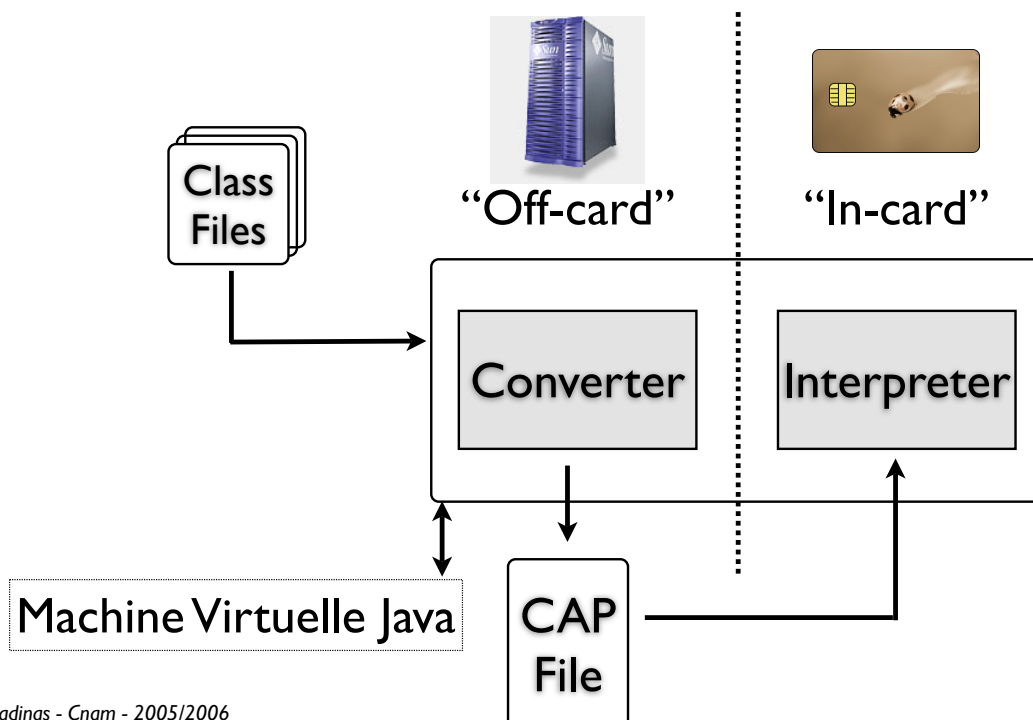
javacard.framework

- Contient les “spécificités” de la carte
- La classe `Applet` :
 - Fournit un cadre à l'exécution et à l'interaction avec le JCRE de l'applet
 - Les applets utilisateurs doivent étendre cette classe
- La classe `APDU`
 - Pour la gestion des échanges avec le monde extérieur
- La classe `PIN`
 - Assure la gestion de code secret

javacard.security

- Basé sur le paquetage java.security
- Permet la gestion des clés et fonction cryptographique
- En plus des algorithmes classiques, ajoute aussi la fonction génération de nombre aléatoire, signature et calcul de fonction de compressions

La modèle machine virtuelle



CAP file et Export File

 Le “CAP File” contient :

 Représentation binaire exécutable de classes

 information sur les classes

 BC (Byte Code) exécutable

 information nécessaire à l'édition de lien

 information pour la vérification,...

 Il est au format JAR (Java Archive), c'est ce qui est reçu par la carte

CAP file et Export File

 Le fichier “Export” est utilisé par le convertisseur.

 Information utilisée pour la vérification et l'édition de lien.

 Contient des informations publiques sur les API :

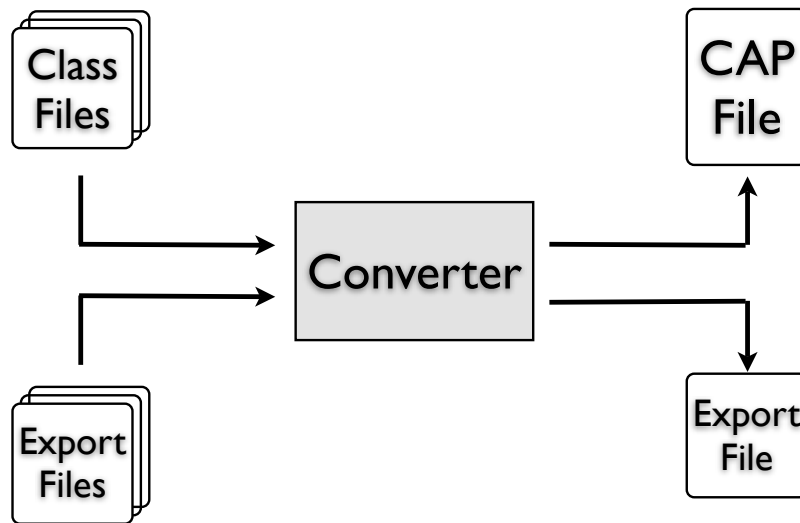
 nom et champs des classes

 signatures des méthodes

 information pour l'édition de lien entre package

 Il ne contient pas de BC, il peut donc être publié avec une applet permettant ainsi à celle-ci d'avoir des objets utilisables (partageables).

Le convertisseur



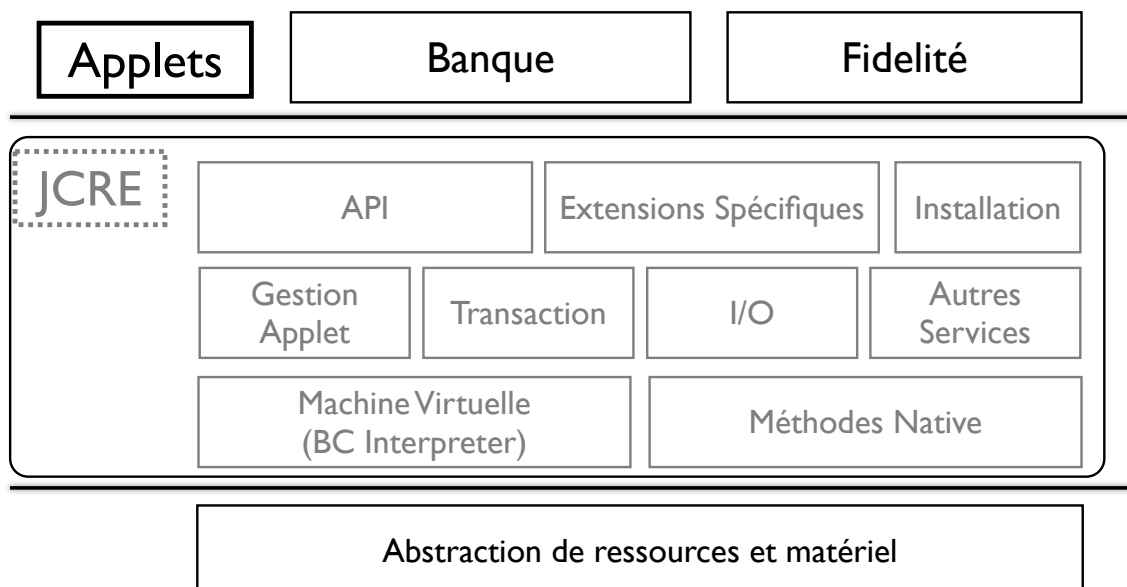
Le convertisseur

- Prend en charge les opérations faites par le class loader de la JVM :
 - Vérification de la conformité du format du Class File
 - Tester la conformité sur les aspects du langage Java
 - Initialisation des variables statiques
 - Résolution des références (classes, méthodes et champs) et mise sous forme compacte pour être plus efficace dans un petit système
 - Optimiser le BC
 - Allocation et création des structures qui représentent les classes dans la VM

L'interpréteur

- Il offre un support d'exécution au BC contenu dans le CAP file. Il permet ainsi aux applets chargées dans une carte de s'exécuter sur n'importe quelle plateforme.
- Il réalise :
 - Exécution du BC
 - Contrôle de l'allocation de la mémoire
 - Assure la sécurité (voir plus loin)
- L'installation d'applet est réalisée par un module en charge de cette action dans la carte mais aussi à l'extérieur pour la gestion du protocole (aspect sécurité voir autre cours)

Architecture d'une Java Card



JCRE : cycle de vie et session carte

- En environnement station de travail, la VM Java est un processus, elle est donc initialisée au lancement puis arrêtée à la fin du processus, les objets en RAM sont donc perdus.
- Pour que les informations soient conservées d'une session à l'autre :
 - Dans le cas de la carte, l'initialisation de la VM est effectuée une seule fois au "début de la vie de la carte", les objets et les données sont conservés en mémoire non volatile (NVM : EEPROM, Flash,...)
 - Lors de chaque session avec la carte :
 - Mise sous tension : le JCRE est "réactivé"
 - La carte reçoit et traite des commandes APDU (Process_APDU)
 - Mise hors tension : le JCRE est "suspendu"

P. Paradinas - Cnam - 2005/2006

25

Caractéristiques du JCRE

- Objets persistant et temporaire
 - Les objets Java Card sont par défaut persistants
 - Pour des raisons d'efficacité (vitesse de lecture/écriture en NVM) et de sécurité (clé, résultat intermédiaire), les applets peuvent créer des objets temporaires
- Opération atomique et transaction
 - La JCVM assure que pour tout les champs d'un objet ou d'une classe les écritures sont atomiques
 - Le JCRE fournit une API aux applets permettant de regrouper plusieurs écritures et d'offrir la cohérence de ces écritures (Begin Transaction, Commit, Roll-Back)

P. Paradinas - Cnam - 2005/2006

26

Applet firewall et mécanisme de partage

- Chaque applet s'exécute dans un espace qui lui est propre.
 - Une applet ne peut avoir d'action sur une autre applet.
 - La JCVM doit assurer cette caractéristique.
- Il existe un mécanisme de partage, qui permet à une applet d'accéder à des services offerts par une applet ou par le JCRE.

Construction d'application JavaCard

- Une application carte
 - Code dans la carte
 - (application serveur = Applet Javacard)
 - Code dans le terminal
 - (application cliente)
- Une application construite en 3 étapes
 - Ecriture de l'application serveur (applet)
 - Installation de l'applet dans la javacard
 - Ecriture de l'application cliente

Écrire applet Java Card

Java Card API 2.1

Etapes du développement d'une applet :

spécifier les fonctions de l'applet :

 spécifier les AID

 écrire le corps de applets

 compiler (.class)

 convertir (.cap)

 charger dans la carte

La Java Card 2.2

 Canaux logiques

 Suppression d'applets et de paquetage

 Mécanisme de suppression d'objet (explicite)

 Java Card Remote Method Invocation

 Support pour l'AES et des courbes elliptiques

 Support pour le sans contact

Java Card - S

- Pour certaines applications la technologie est parfois trop “riche” et le coût de la licence élever par rapport aux prix final. Pour cela une version S a été introduite.
- La "Java Card S" permet aux licenciés de concevoir des cartes dont seule la fonction de chargement dynamique d'applet est absente. Ce qui permet de
 - Récupérer du code,
 - Utiliser des composant moins cher,
 - Diminuer les coûts de certification.

Conclusion

- Java Card donne un langage de programmation aux applications dan sla carte avec de bonnes propriétés
- L'infrastructure de chargement est définie par OP (Open Platform)...
- Une version 3.0 est en cours et devait apporter encore de progrès

Bibliographie

-  <http://java.sun.com/products/javacard/>
-  Technology for Smart Cards:Architecture and Programmer's Guide, Zhiqun Chen,Addison Wesley, September, 2000
-  Understanding Java Card 2.0, Zhiqun Chen & Rinaldo Di Giorgio
-  <http://www.javaworld.com/javaworld/jw-03-1998/jw-03-javadev.html>
-  <http://javacardforum.org>
-  Java Card les produits des fabricants de cartes