La Carte à Microprocesseur

Un système embarqué en plein essor

Tegawendé F. Bissyandé

tegawende.bissyande@fasolabs.org

Cours préparé pour

L'Université Ouaga I Pr. Joseph Ki-Zerbo(UFR SEA)

April 6, 2017

FasoLabs.org

La carte à puce dans nos vies...

Familles de cartes d'un point de vue programmation :

Cartes spécifiques

- Réservées aux grands groupes industriels *(Cartes bancaires, cartes SIM, cartes de TV cryptées)*
- En général pour production de masse
- Lourde *(en terme de mise en oeuvre)* mais souple *(dans l'installation de l'OS, les fichiers, les instructions)*
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse $\Rightarrow$ s'addresser à un encarteur tel Gemalto

Cartes personnalisables

- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications *(commandes de débit/crédit/calcul_de_solde pour paiement électronique)*

Cartes à OS ouverts

- Carte à puce "non terminée" – programer son propre interpréteur de commandes
- Le microcontrolleur est programmé dans un langage évolué (tel Java) qui est précompilé

Familles de cartes d'un point de vue programmation :

Cartes spécifiques

- Réservées aux grands groupes industriels *(Cartes bancaires, cartes SIM, cartes de TV cryptées)*
- En général pour production de masse
- Lourde *(en terme de mise en oeuvre)* mais souple *(dans l'installation de l'OS, les fichiers, les instructions)*
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse $\Rightarrow$ s'addresser à un encarteur tel Gemalto

Cartes personnalisables

- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications *(commandes de débit/crédit/calcul_de_solde pour paiement électronique)*

Cartes à OS ouverts

- Carte à puce "non terminée" – programer son propre interpréteur de commandes
- Le microcontrolleur est programmé dans un langage évolué (tel Java) qui est précompilé

Familles de cartes d'un point de vue programmation :

Cartes spécifiques

- Réservées aux grands groupes industriels *(Cartes bancaires, cartes SIM, cartes de TV cryptées)*
- En général pour production de masse
- Lourde *(en terme de mise en oeuvre)* mais souple *(dans l'installation de l'OS, les fichiers, les instructions)*
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse $\Rightarrow$ s'addresser à un encarteur tel Gemalto

Cartes personnalisables

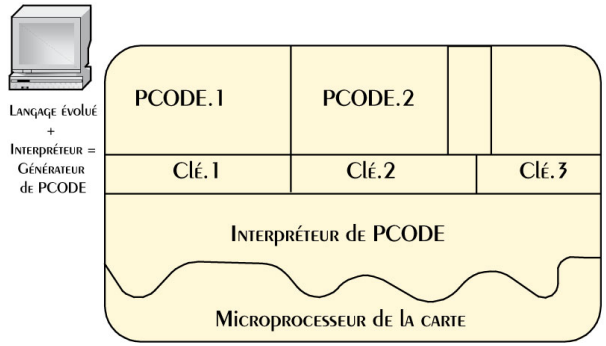
- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications *(commandes de débit/crédit/calcul_de_solde pour paiement électronique)*

Cartes à OS ouverts

- Carte à puce "non terminée" – programer son propre interpréteur de commandes
- Le microcontrolleur est programmé dans un langage évolué (tel Java) qui est précompilé

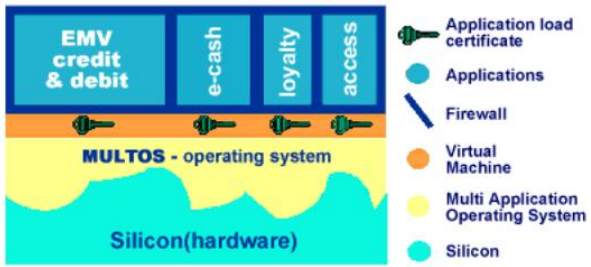
Cartes à OS ouvert (1)

- Précompilation vers un code intermédiaire : le `P code`
- Microcontrolleur est programmé avec un interpréteur de P code
- + Carte réellement programmable
- + Carte programmée en langage évolué (C, Basic, Java)
- + Pas de contrainte de production en masse
- + Application développée est portable



Systeme Multos

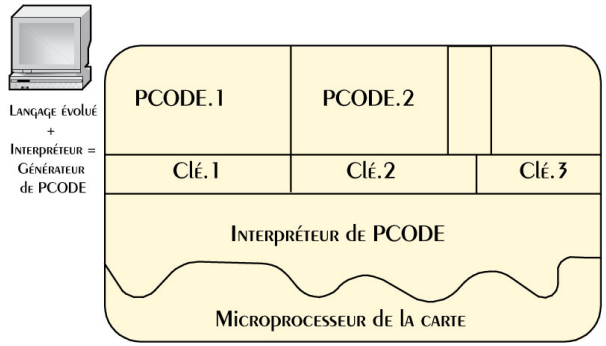
- L'ancetre des cartes à OS ouverts
- OS multi-application proposé initialement par Mondex et Master Card
- Chaque application dispose d'un pare-feu
- Ajout et "Mise à jour" possible d'applications sans le fournisseur



FasoLabs.org

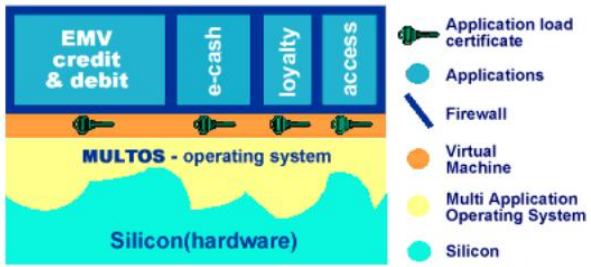
Cartes à OS ouvert (1)

- Précompilation vers un code intermédiaire : le P code
- Microcontrolleur est programmé avec un interpréteur de P code
- + Carte réellement programmable
- + Carte programmée en langage évolué (C, Basic, Java)
- + Pas de contrainte de production en masse
- + Application développée est portable



Systeme Multos

- L'ancetre des cartes à OS ouverts
- OS multi-application proposé initialement par Mondex et Master Card
- Chaque application dispose d'un pare-feu
- Ajout et "Mise à jour" possible d'applications sans le fournisseur



FasoLabs.org

La Basic Card

- Commercialisée par une petite société allemande (Zeit Control)
- Fonctionne comme les autres modèles à OS ouverts
- originalité: le P code provient de la compilation d'un programme en Basic
- Meme langage utilisé pour la carte et pour le terminal

La Java Card

- Se programme bien sûr en Java
- La carte contient un interpréteur de *bytecode*
- En developpement, la machine virtuelle Java est donc scindée en deux
- En fait, un sous-ensemble du langage java est supporté par la Java Card

Intérêt

1. La Java Card est vraiment normalisée
2. Sûreté des applications assurée par le langage
3. La Java Card autorise la coexistence de plusieurs applications
4. La Java Card supporte la modification dynamique des applications (cycle de vie)

Cartes à OS ouvert (2)

La Basic Card

- Commercialisée par une petite société allemande (Zeit Control)
- Fonctionne comme les autres modèles à OS ouverts
- originalité: le P code provient de la compilation d'un programme en Basic
- Meme langage utilisé pour la carte et pour le terminal

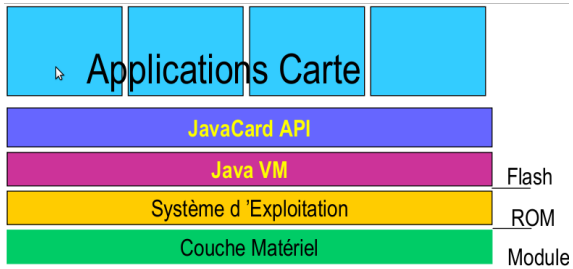
La Java Card

- Se programme bien sûr en Java
- La carte contient un interpréteur de *bytecode*
- En developpement, la machine virtuelle Java est donc scindée en deux
- En fait, un sous-ensemble du langage java est supporté par la Java Card

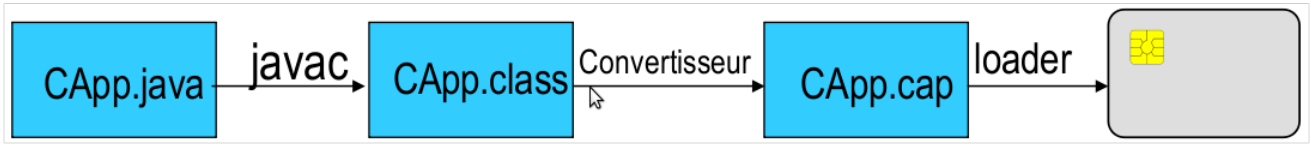
Intérêt

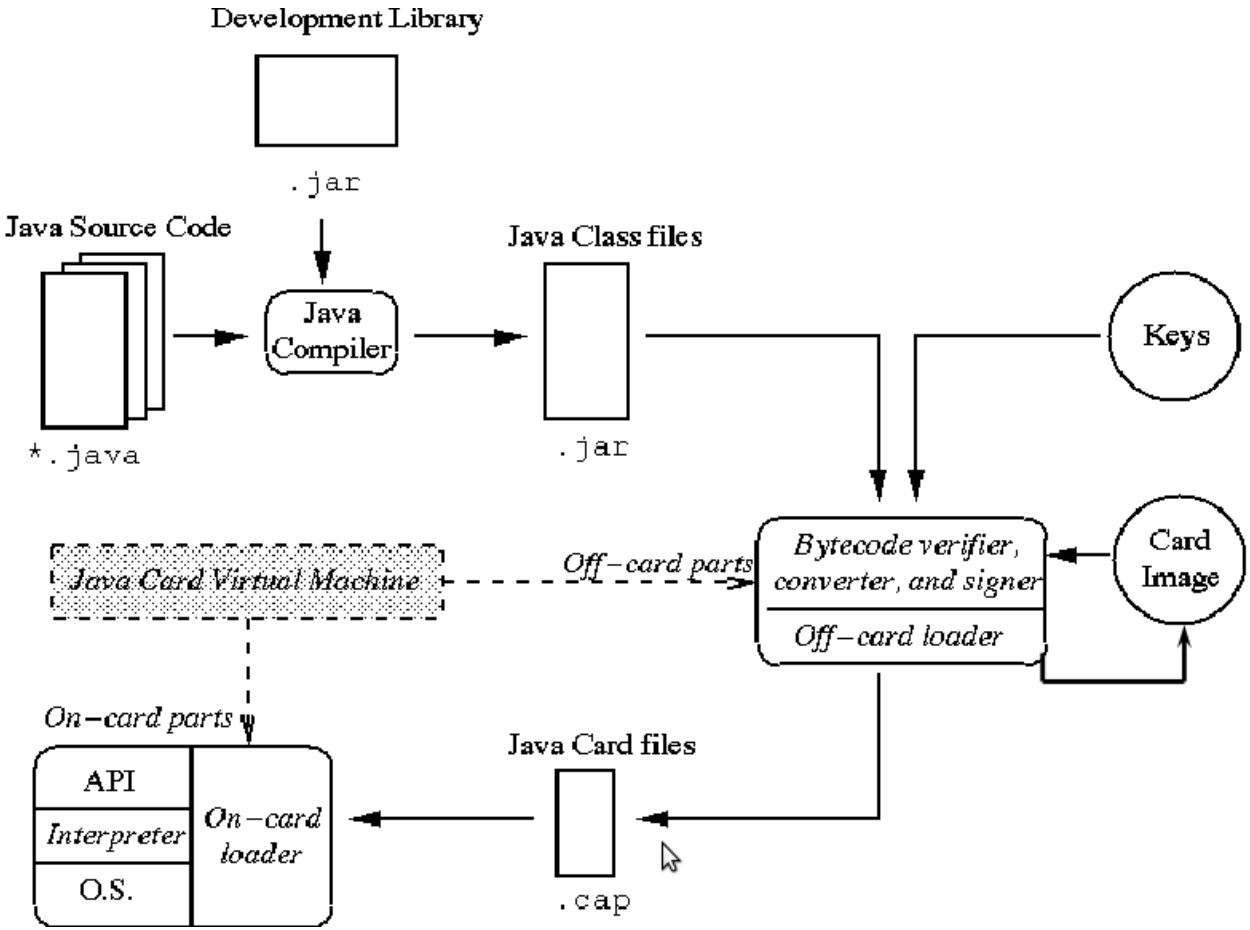
1. La Java Card est vraiment normalisée
2. Sûreté des applications assurée par le langage
3. La Java Card autorise la coexistence de plusieurs applications
4. La Java Card supporte la modification dynamique des applications (cycle de vie)

FasoLabs.org



- Pas de chargement dynamique de classe
- Allocation dynamique d'objets supportée mais
  - pas de garbage collection
  - pas de désallocation explicite non plus :(
  - ...
- Quelques types de base (`byte`, `int`, `boolean`) – pas de `char` (ni de classe `String`), `double`, `float`, `long`
- Objets supportés / Mécanisme d'héritage supporté / Pas de threads
- Securité : Notion de paquetages & modifieurs `public`, `private`, `protected`


CApp.java
javac
CApp.class
Convertisseur
CApp.cap
loader


Development Library
.jar
Java Source Code
Java Compiler
Java Class files
Keys
*.java
.jar
Java Card Virtual Machine
Off–card parts
Bytecode verifier, converter, and signer
Off–card loader
Card Image
On–card parts
API
Interpreter
O.S.
On–card loader
Java Card files
.cap