Java Card™

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

valeur C : ISTR

Plan

- Hervé Lebougre & Joel Routaboul
- La carte à microprocesseur
  - Architecture matériel
  - Normes et standards
  - Domaine d'applications
  - Java Card

Carte mémoire

- fonction logique de stockage de données, pas de CPU
  - notion de zone
  - algorithme d'authentification
  - exemple : Télécarte, T2

Carte à microprocesseur

- 1978 brevet SPOM, Bull, Michel Ugon
  - Self Programming Once Memory
  - CPU
- 1 ère implémentation de la “smart card” (CP8)
  - RAM : 36 o
  - EPROM : 1 ko
  - ROM : 1,6 ko

Architecture logicielle

- Gestion protocole d'E/S
- Gestion de la mémoire
- Fonction cryptographique
  - "Telepass" algorithme secret

Architecture

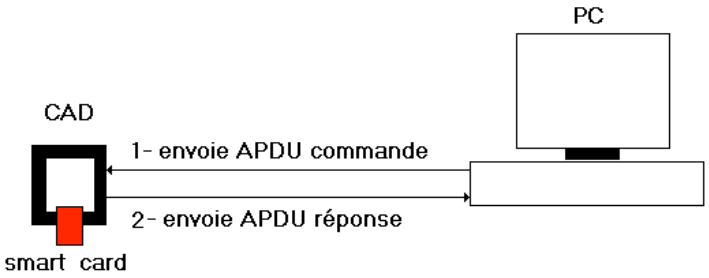
- CPU : 8, 16 & 32 bits
- Mémoires
  - RAM : 512 ko
  - EEPROM/Flash : 128/256 ko
  - ROM : 256/512
- Cellule cryptographique et générateur de nombre (aléatoire)
- Détecteur de sécurité

Les standards ISO 7816

- Part 1
- Part 2
- Part 3
- Part 4
- Part 5
- Part 6
- Part 7

La carte et son environnement

- Protocole de communication normalisé dans l'ISSO 7816-3 et 4
- La carte est un serveur



Le protocole 7816

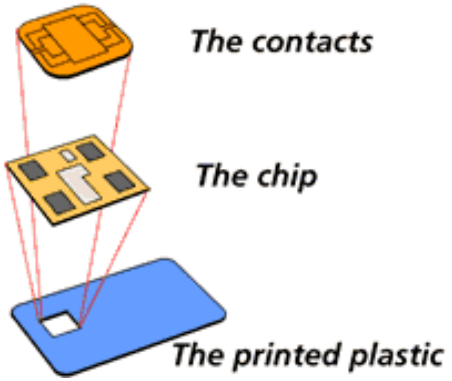
- APDU command

entête				corps du message		
CLA	INS	P1	P2	Lc	données	Le

- APDU response

Données	SW1	SW2

Cycle de vie d'une carte



- Fabrication
- Emission :
  - avec personnalisation par rapport :
    - émetteur et ses services
    - porteur et ses services
  - distribution
- Vie de la carte (perte)

Les applications

- GSM - Standards ETSI
- Paiement - Standards EMV

- Les nouveaux besoins :
  - plus de souplesse, adapatbilté, programmabilité,...

Java Card - Introduction

- besoin marché vers des systèmes programmables
- 1ère version : octobre 1996 mais démarrage réel en 1998, une réalité depuis 2/3 ans

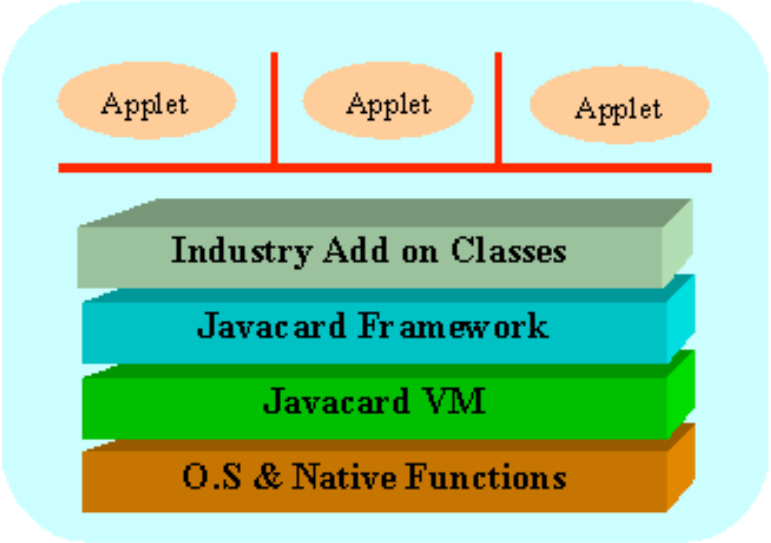
Sun et Java Card Forum

- Schlumberger et Gemplus créent le JCF en 1997
  - Association regroupant les fabricants de silicium, les encarteurs et des clients
    - Promouvoir la solution de la javacard
    - Définir des choix technologiques puis les proposer à Sun qui en fait le “standard”

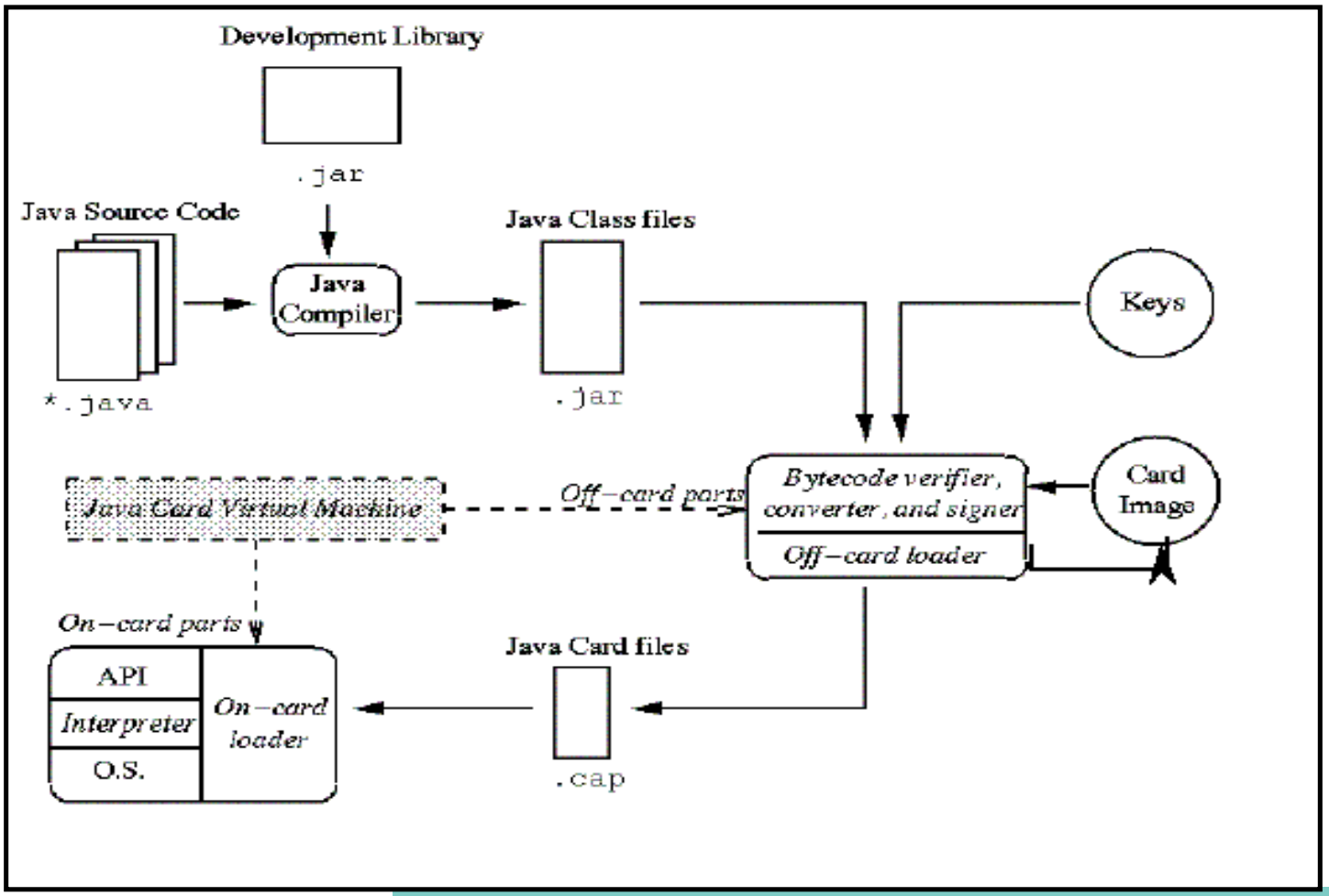
http://www.javacardforum.org

Architecture d'une Java Card

- Carte basée sur un interpréteur de bytecode java
- Nécessite de pré-compiler les applications



Détails


Development Library
.jar
Java Source Code
*.java
Java Compiler
Java Class files
.jar
Keys
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

Javacard par rapport à Java

- Pas de chargement dynamique de classes
- Objets : Allocation dynamique d'objets supportée (new) MAIS
  - Pas de ramasse miette
  - Pas de désallocation explicite
- la mémoire allouée ne peut pas être récupérée
- Pas de méthode finalize()

Javacard par rapport à Java

- Supportés :
  - Booléan,Byte, Short,Int
  - Object
  - Tableaux à 1 dimension
  - Allocations dynamiques
  - Packages
  - Exceptions
  - Interfaces
  - Méthodes natives

Javacard par rapport à Java

- Non supportés
  - Float, double,
  - Char, String
  - Tableaux à n dimensions
  - Class et ClassLoader
  - Ramasse-Miettes
  - SecurityManager
  - Threads

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 (API 2.2 est plus récente)
- Etapes du développement d'une applet :
  - spécifier les fonctions de l'applet :
  - spécifier les AID
  - écrire les applets
  - compiler (.class)
  - convertir (.cap)
  - charger dans la carte

Conclusion

- Java Card donne un langage de programmation aux applications avec de bonnes propriétés
  - les applications ont la même structure
  - néanmoins RMI dans la JC2.2
- L'infrastructure de chargement est définie par OP (Open Platform)

Bibliographie

- http://java.sun.com/products/javacard/
- Technology for Smart Cards: Architecture and Programmer's Guide, Zhiqun Chen September 18, 2000
- Understanding Java Card 2.0, Zhiqun Chen & Rinaldo Di Giorgio