From SmartCard to Javacard

Project unfolding

- Installing a development kit
- Getting acustomed with its use
- Developing batch files to convert java code into Javacard compatible format
- Developing some applications

Introduction

- 1974 : Credit cards invented by R.Moreno
  - A card formed of plastic body with an embedded integrated circuit.
- 1994 : Standard ISO 7816
- 1996 : SmartCard ➔Javacard (Schlumberger)
  - Application domains
  - Assembler language ➔Java language
  - A virtual machine onto smart cards=JCVM

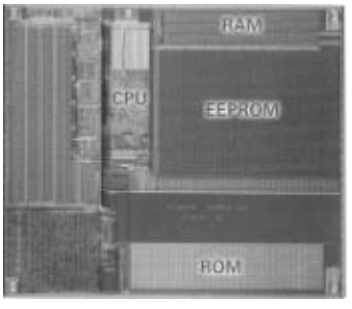
SUMMARY

- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

Smart card composition

- Smart card = passive computer
- Smart card is composed by:
  - A communication interface
  - Memories
  - A microprocessor



Smart card

- Smart cards memories
  - ROM (16kb-24kb)
    - Stores the COS (Card Operating System) and permanent data
    - Fixed by firms
  - RAM (256b-1kb)
    - Stores temporary data (process)
  - EEPROM (16kb-64kb)
    - Electrical Erasable Programmable Read Only Memory
    - Stores permanent data
- Operations made by microprocessor

Bring Javacard technology on a card

- Limited Memory space problem
- Solutions
  - JCVM splitting
  - Verification off-card
  - Javacard=a subset of Java
    - 1% of JDK API necessary

- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

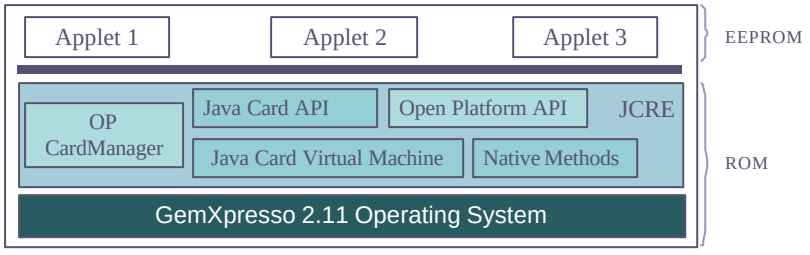
Javacard language

- The Javacard API restrictions
  - » Abstract types supported: boolean, byte, short
  - » Simple dimension arrays
  - » Exceptions supported
  - » No threads
- 4 packages ex: Package javacard.framework
  - Install ()
  - Register ()
  - Process ()
  - Select () and Deselect ()

Use of Java

- Applications easy to build
- Multi-applications capable
- Secure (sandboxes)
- Dynamic
- Portable

Java Card Runtime Environment



- Implementing JCVM and APIs
- Managing temporary data storage
- Saving applets onto a card (AID = Applet Identifier)
- Managing the communication between Card Acceptance Device (CAD) and card

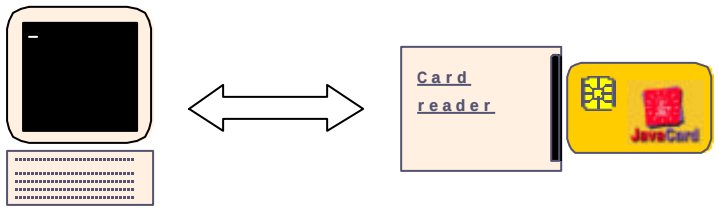
Lifetime of a java card

- Hardware :
  - Masking and initialization/personalization
  - Activation of the JCVM
  - End of the physical card lifetime
- Software :
  - Running of the JCVM
  - Applet=small application using Java technology
  - Creating Persistent objects & Temporary objects (install method)

- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

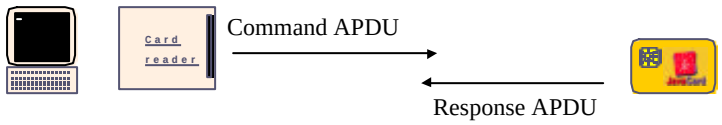
Information exchange

- No integrated input or output devices
- Card Acceptance Device (CAD) = reader



Exchanges (ISO 7816-4)

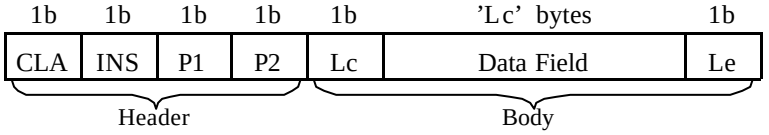
- Server (javacard) ⇔ Client (CAD)



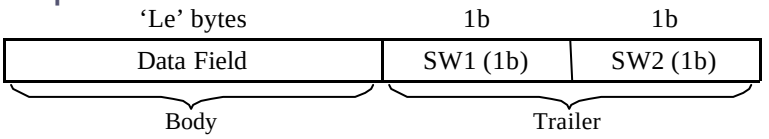
- CAD —APDU→ JCRE
- JCRE —Process(APDU)→ applet
- Instruction executed by the applet

Exchanges (ISO 7816-4)

- Command APDU format



- Response APDU format



- ATR (Answer To Reset) message (>33b) : data relevant to transmission protocol & card

- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

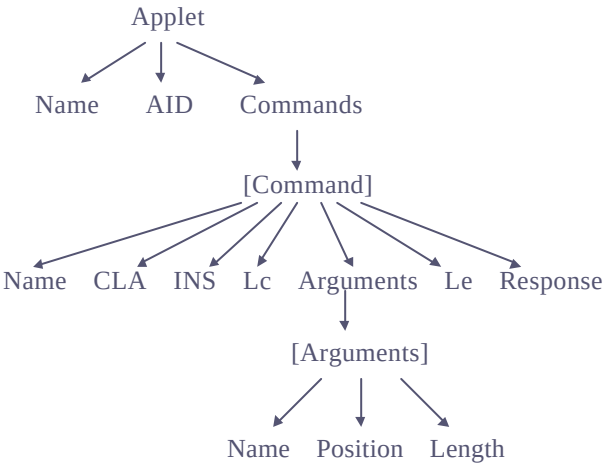
Existing Development kits

- Odyssey_Lab (Bull)
- Cyberflex (Schlumberger)
- Gemxpresso RAD (Gemplus)

Loading a Javacard application

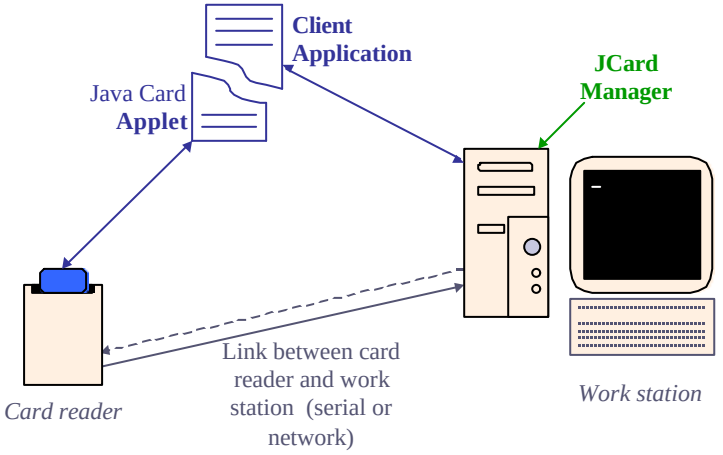
- 1: compilation of file.java by JDK
- 2: conversion of file.class into two files (file.jca and file.exp)
- 3: compression of file.jca into file.jar
- 4: loading with JCManager

Applet structure



- Smart Card Architecture
- Javacard Environment
- Exchanges
- Applets Development
- Illustration

Exchanges

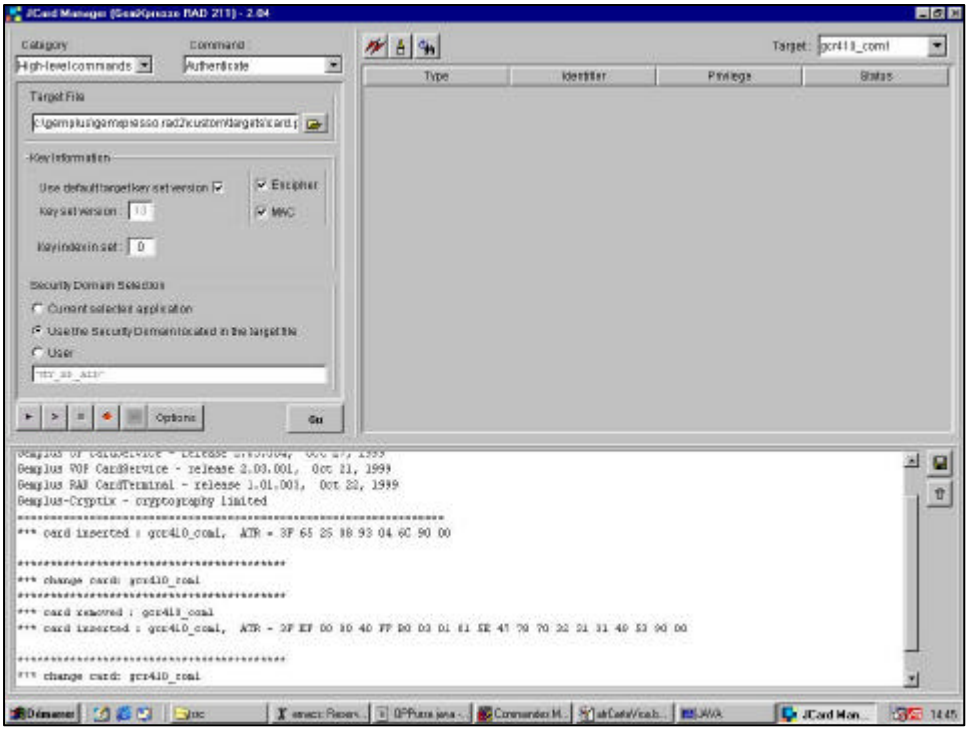


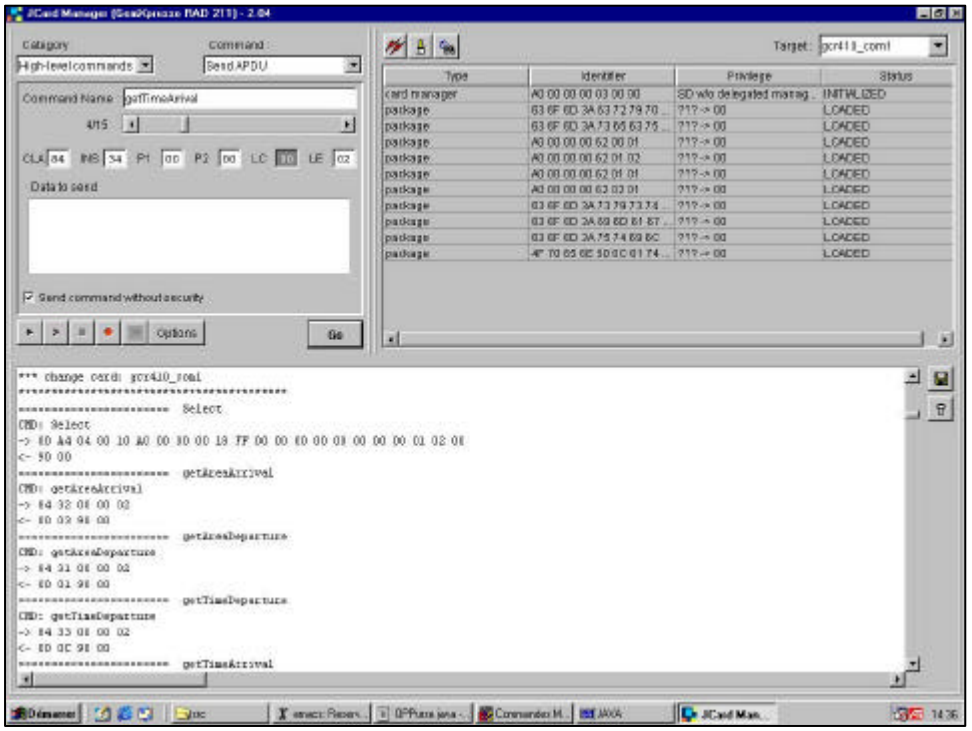
Implementation

- Based on the OPPurse example given by GemXpresso
  - Security declarations in the source code
  - Client/server communication
- Developed Javacard applications
  - “Pointeuse”
  - “Reservation”

Developed applets

- Pointeuse
  - Adapting a smart card application into Javacard
- Reservation
  - Storing and retreiving informations about a flight reservation.
  - ToDo : using some OPPurse features within the applet





Conclusion

- Smart card ➔ Java card
- Javacard = Universal card
- Firms can develop their own applications

- Linked to our formation :
  - A communication protocol
  - Java programming with restrictions

Within 10 years, Javacard is likely to become a systematically used standard ...