

Hauptseminar

Die Einführung in die Java Card Technologie

Nachweis von Sicherheitseigenschaften für Java Card
durch Approximative Programmauswertung

Veranstalter Pr. T. Nipkow
Dr. M. Strecker

Autor Tao Zhuang

07,01,2002

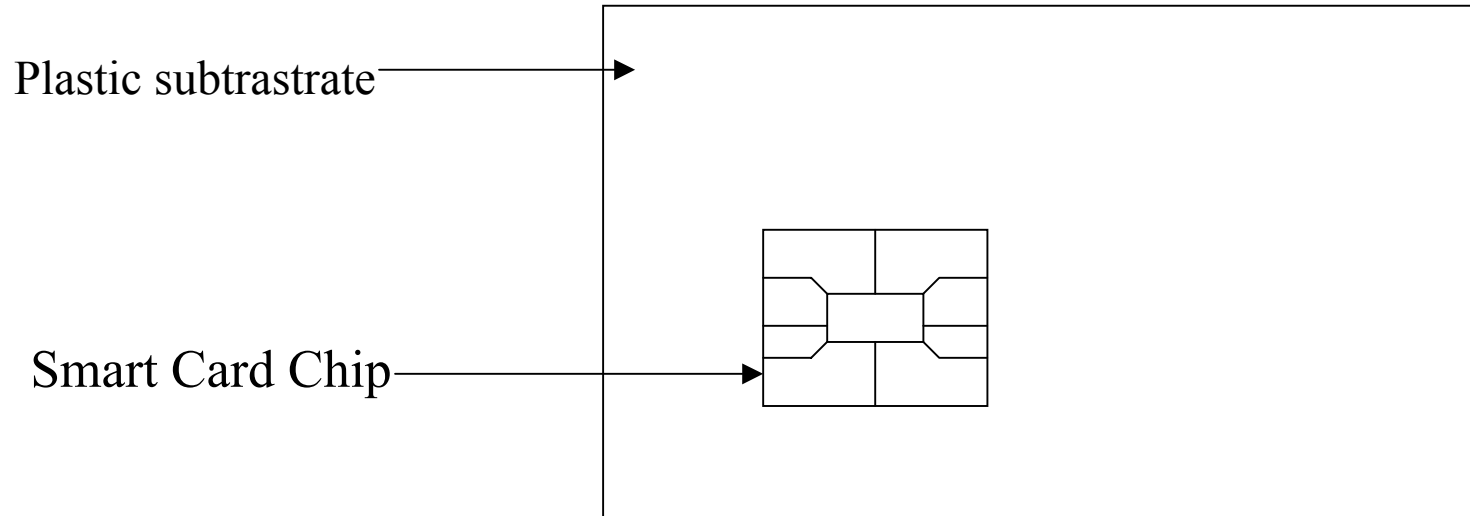
Überblick

- Einleitung
- Smart Card Basis
- Überblick der Java Card Technologie
- Die Objekte der Java Card
- Atomicity und Transaktion

Einleitung

- Die Smart Card
- Die Vorteile von Smart Cards
- Die Vorteile von Java Cards
Technologie

Smart Card



Die Vorteile Von Smart Cards

- Sicherheit
- Tragbarkeit
- Die Card leicht zu nutzen

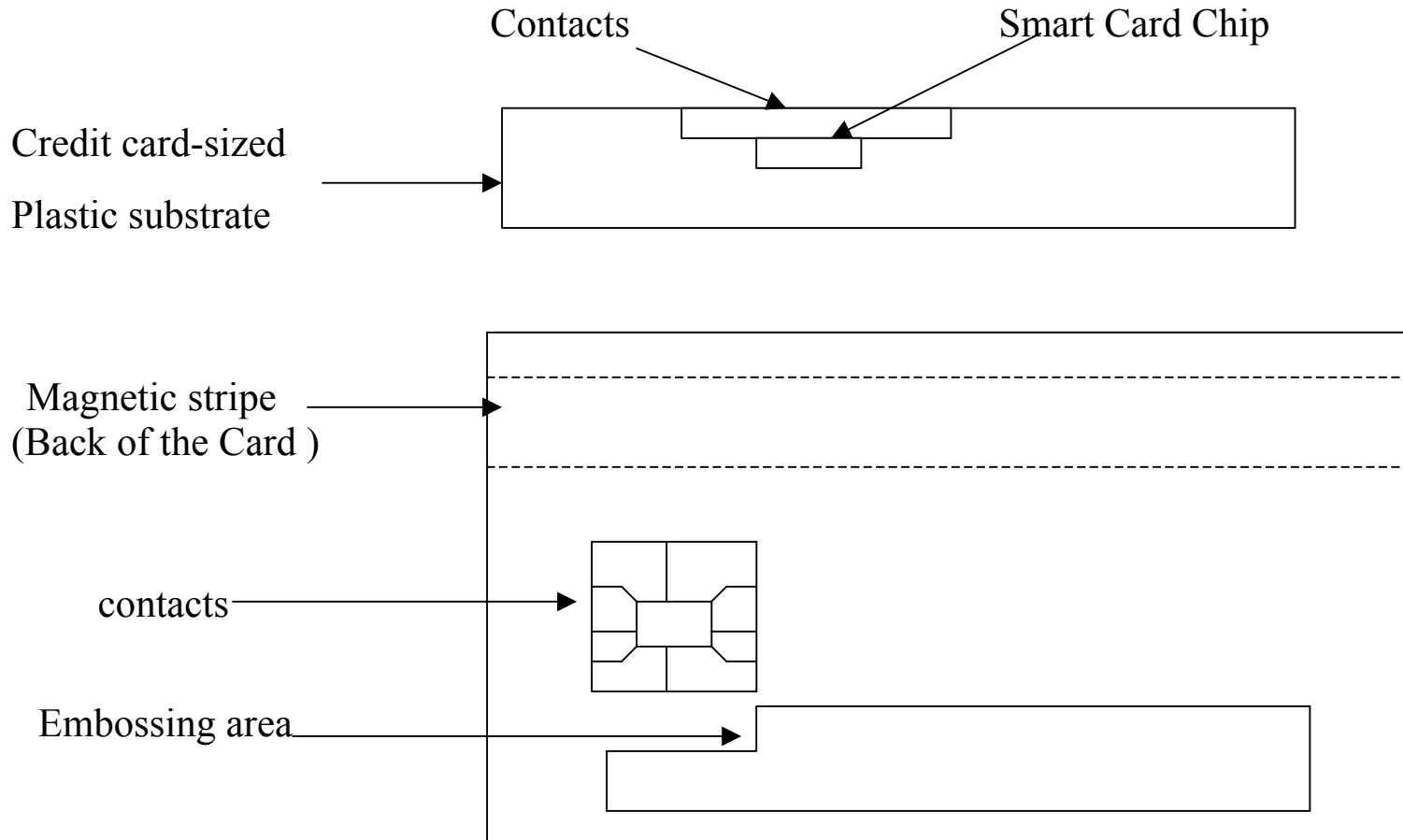
Die Vorteile von Java Card Technologie

- Die Applikation leicht zu entwickeln
- Sicherheit
- Hardware unabhängig
- Die multiplen Applikationen zu speichern und verwalten

Smart Card Basis

- Überblick
- Basisarte der Cards
- Smart Card Hardware
- Smart Card Kommunikation
- Smart Card Betriebssystem
- Smart Card System

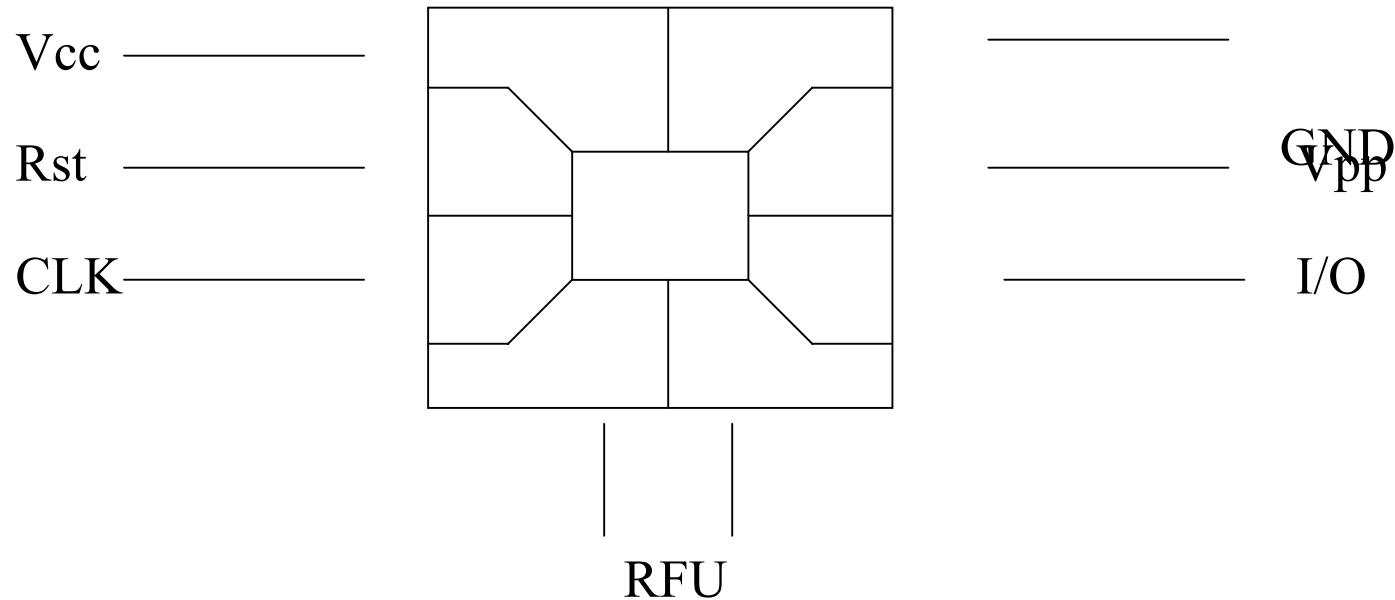
Die Physischerscheinung der Java Card



Smart Card Hardware

- Smart Card Kontakte Punkte
- Smart Card CPU(8 bit , 16 bit , 32 bit)
- Smart Card Koprozessor
- Smart Card Speichersystem

Acht Kontaktpunkte



- Vcc :bietet dem Chip den Betriebsstrom an.
- RST:sendet ein Signal , um den Mikroprozessor wieder zu starten (Warm reset).
- CLK:bietet das external clock Signal .
- GND:Referenzvolt .
- Vpp :benutzt von der Alten Card .
- I/O :transportiert die Daten und Befehle zwischen Smart Card und CAD.

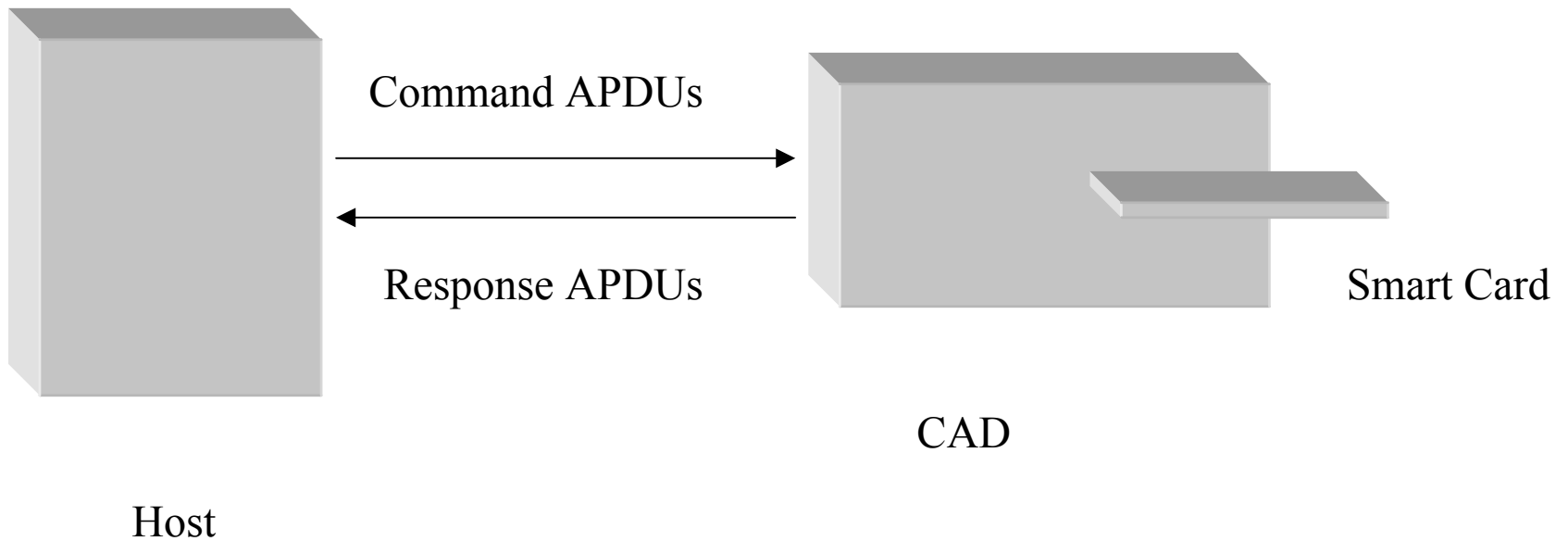
Smart Card Speichersystem

- ROM (read only memory)
- EPPROM (electrical erasable programmable read only memory) :100,000 mal zugreifbar die Daten bleiben 10 Jahr und die Geschwindigkeit der Schreibens ist 1000 mal langsamer als RAM .
- RAM (random access memory)

Smart Card Kommunikation

- CAD und Host Applikationen
- Smart Card Kommunikationsmodel
- Smart Card Kommunikation
- APDU Protokoll
- TPDU Protokoll

Smart Card Kommunikationsmodell



APDU Struktur

Command der Struktur von APDU

Mandatory header				Optional Body		
CLA	INS	P1	P2	Lc	Data Field	Le

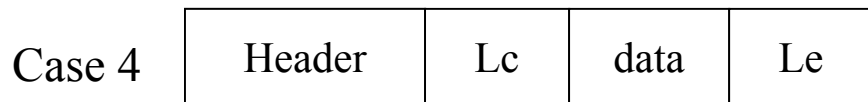
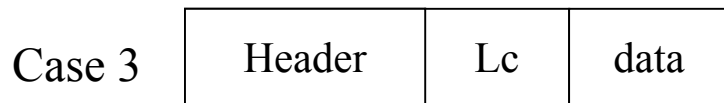
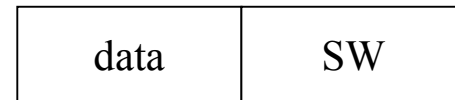
Response APDU Struktur

Optional Body	Mandatory Trailer	
Data Field	SW1	SW2

APDU Struktur

Command APDU

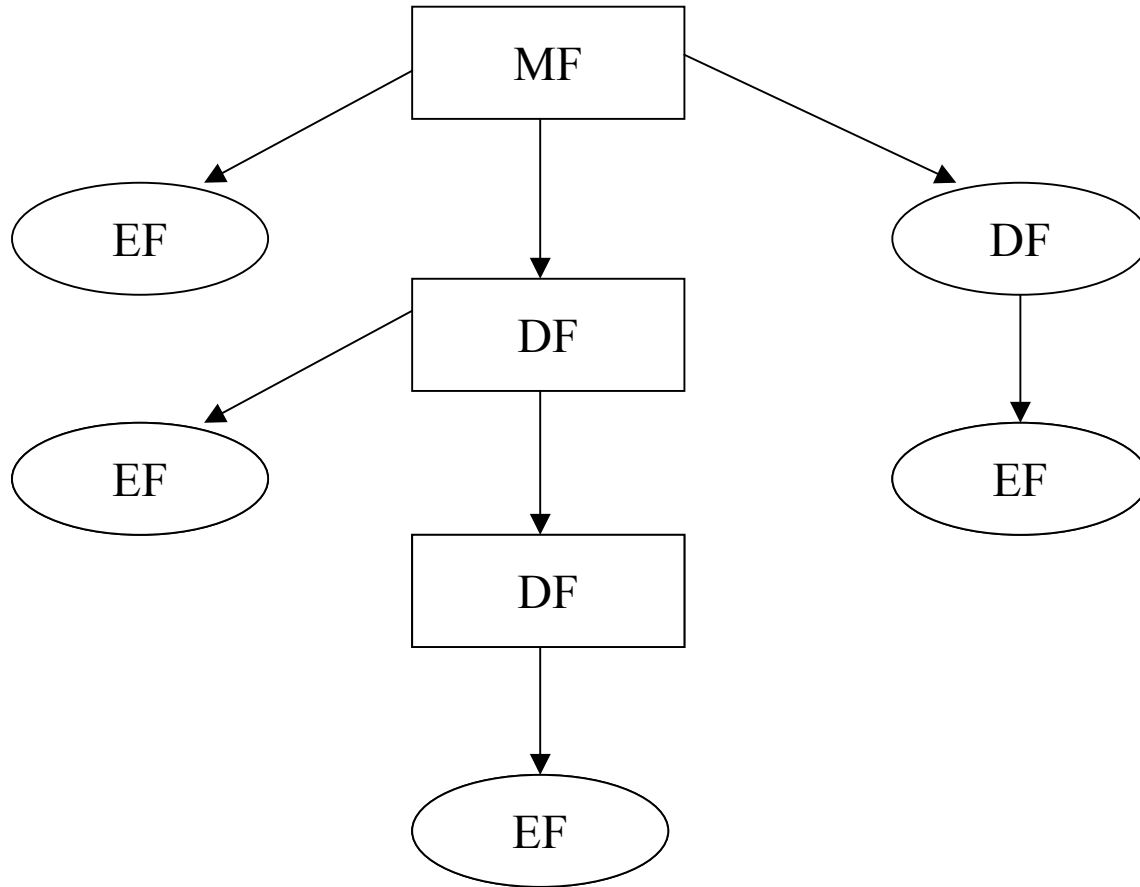
Response APDU



Smart Card Betriebssystem

- Smart Card File System
- Master File
- Dediziertes File
- Elementares File

ISO 7816-4 File System Structure



Smart Card System

- Host System

- Bearbeitet die Kommunikation zwischen der Userapplikation und der Card.

- Bietet Unterstützung zur Smart Card Infrastruktur. (z.B. Card Management Sicherheit , Key Management) .

- Normalerweise in Java ,C,C++ geschrieben .

Smart Card System

- Card System

- Bearbeitet die I/O Kommunikation mit dem Host .
- Verstelt die Integration und Sicherheit der Daten sicher .
- Unterstützt die ISO Filesystem.

Überblick der Java Card Technologie

- Überblick der Architektur
- Submenge der java Card Sprache
- JCVM
- Java Card Installer und Off-Card Installationsprogramm
- JCRE
- Java Card APIs
- Applet Entwicklungsprozess
- Applet Installation

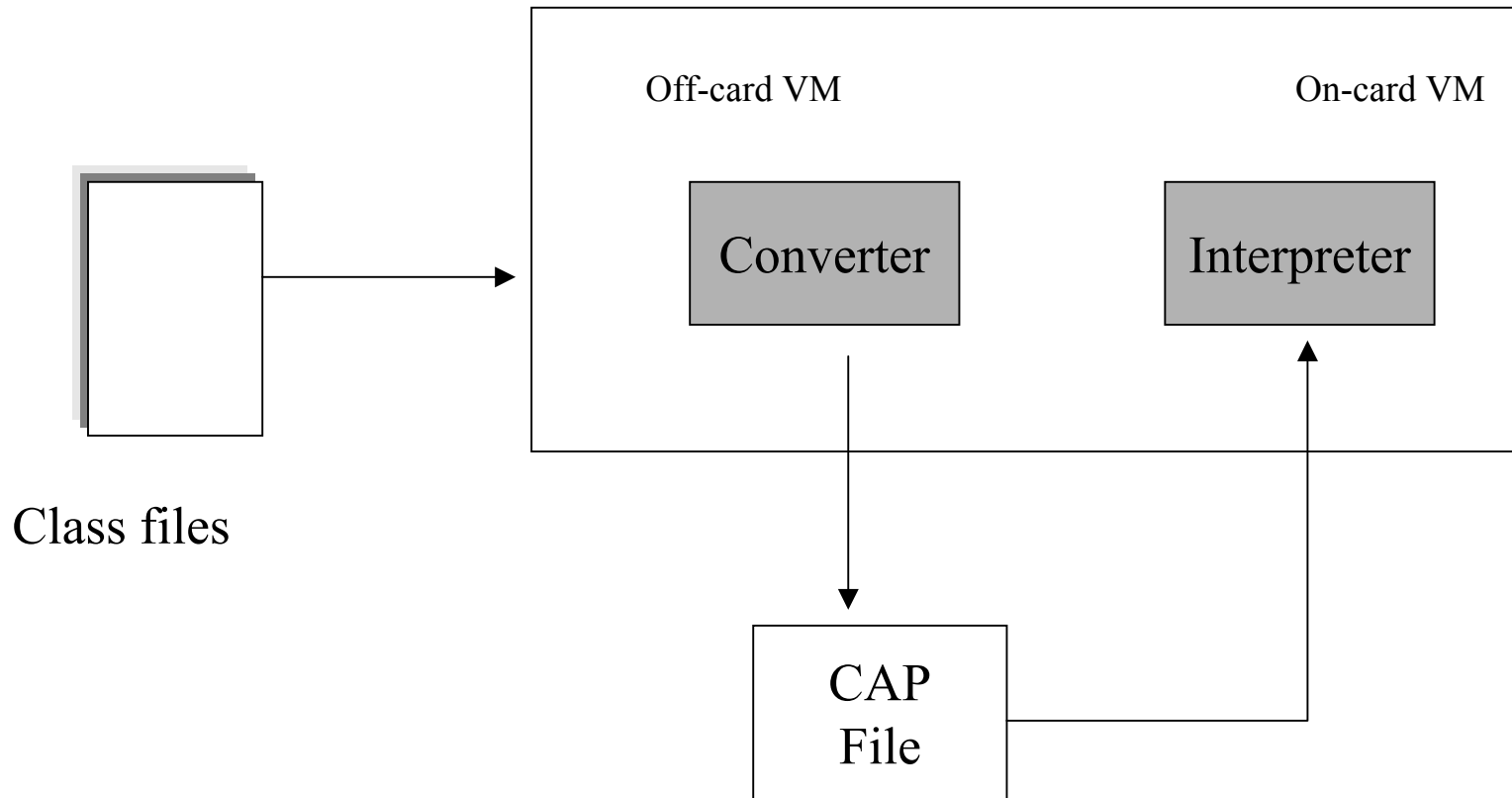
Ueberblick der Architektur

- Konfiguration der Smart Card
- JCVM (Java Card visual Maschine):
 - Off-Card System
 - On-Card System
- JCRE
- Platform
 - JCVM 2.1
 - JCRE 2.1
 - Java Card API 2.1

Submenge der Java Card Sprache

Supported Java Features	Unsupported Java Features
<ul style="list-style-type: none">• Small primitive data type: boolean ,byte,short• One-dimensional arrays• Java packages , classes ,interfaces , and exceptions• Java object-oriented features:inheritance ,virtual methods ,overloading and dynamic object creation ,access scope ,and binding rules• The int keyword and 32-bit integer data type support are optional .	<ul style="list-style-type: none">• Large primitive data types :long ,double , float• Characters and strings• Multidimensional array• Dynamic class loading• Security manager• Garbage collection and finalization• Threads• Object serialization• Object cloning

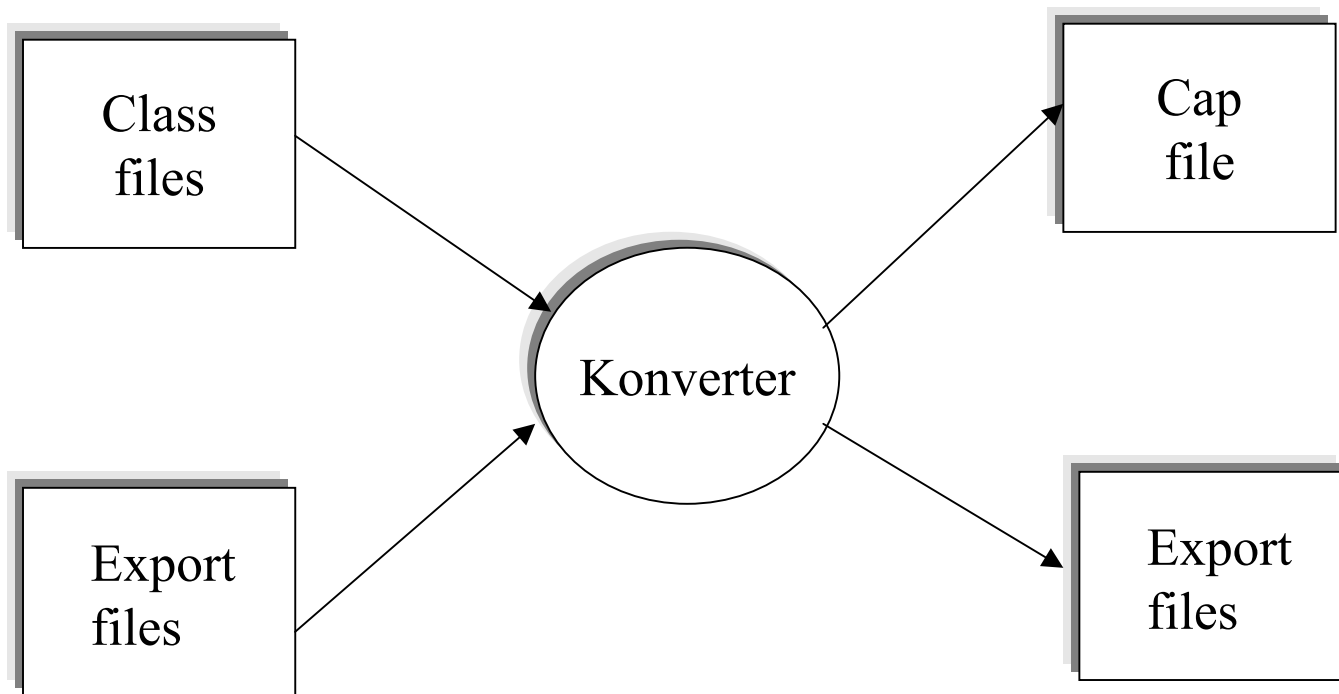
Java Card Visuale Masching



Java Card Virtuelle Masching

- CAP Files und Export File
- Java Card konverter
- Java Card Interpreter

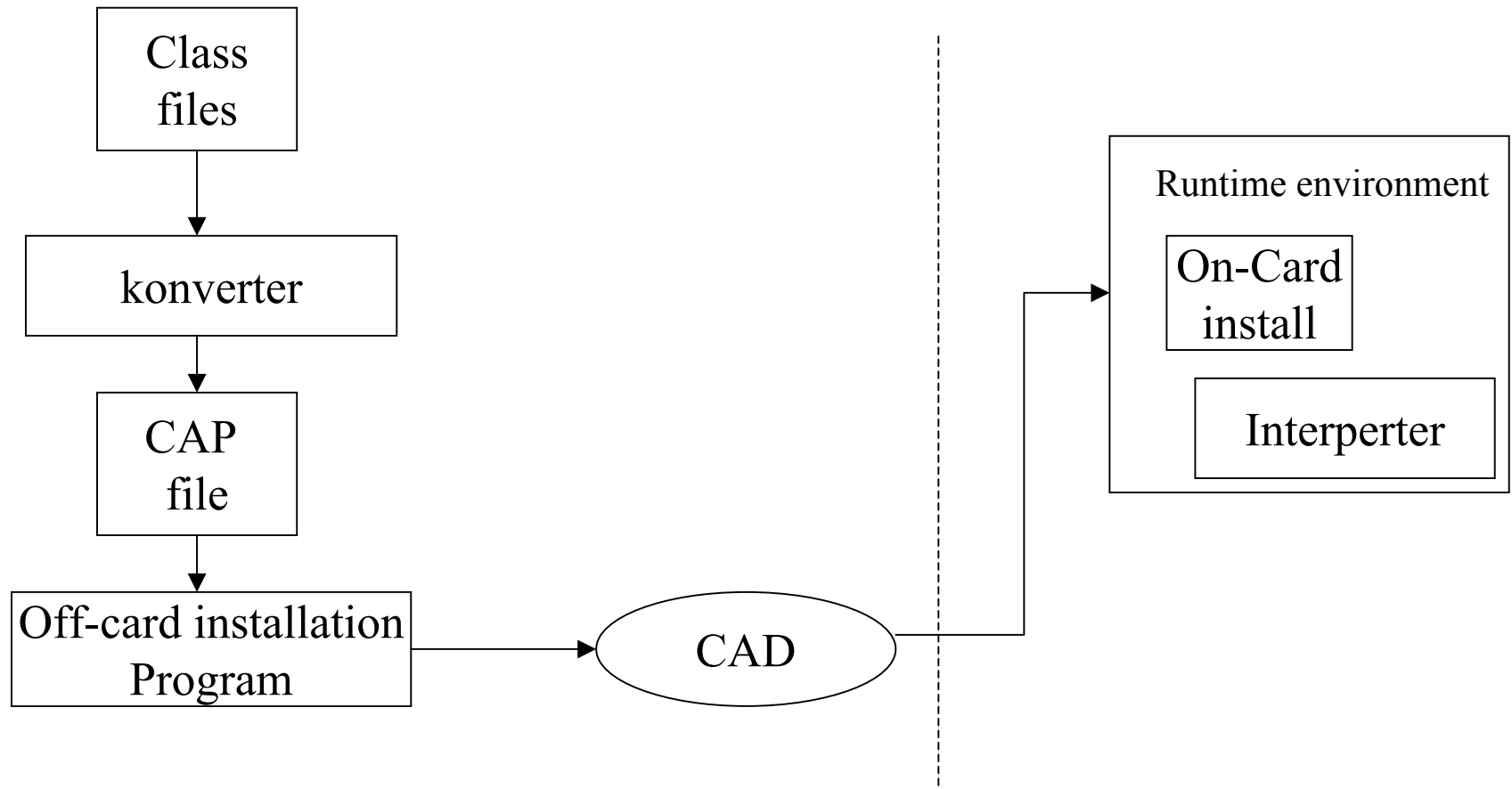
Konvertierung einer Package



Java Card Interpreter

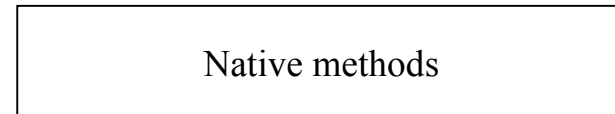
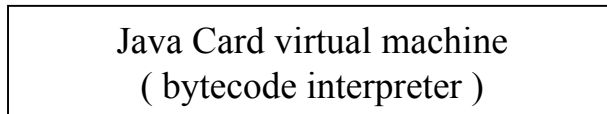
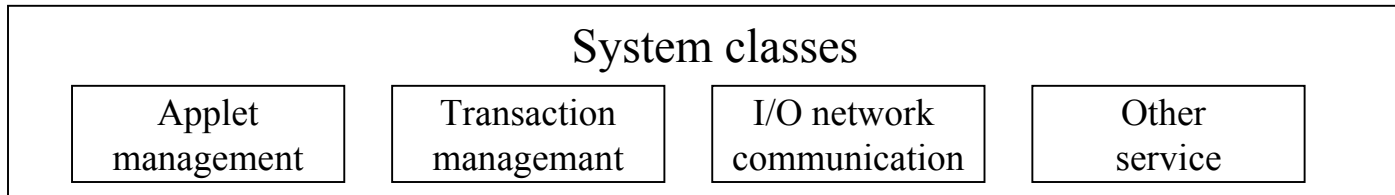
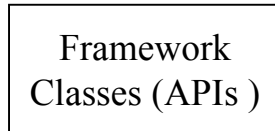
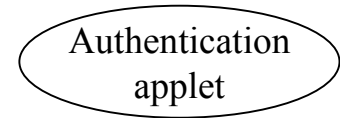
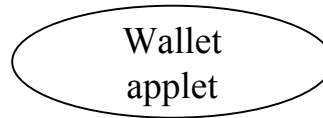
- Führen die Bytecode aus .
- Kontrollieren die Speicherverteilung und die Objekterzeugung

Java Card Installer und Off-Card Installationsprogramm



Java Card Runtime Environment

Applets



On-card System Architecture

Java Card Runtime Environment

- JCRE Lebensdauer
- JCRE Operation in einer CAD Session
- Java Card Runtime Eigenschaften
 - Persistente und transiente Objekte
 - Atomare Operationen und Transaktionen
 - Applet Firewall und der sharing Mechanismus

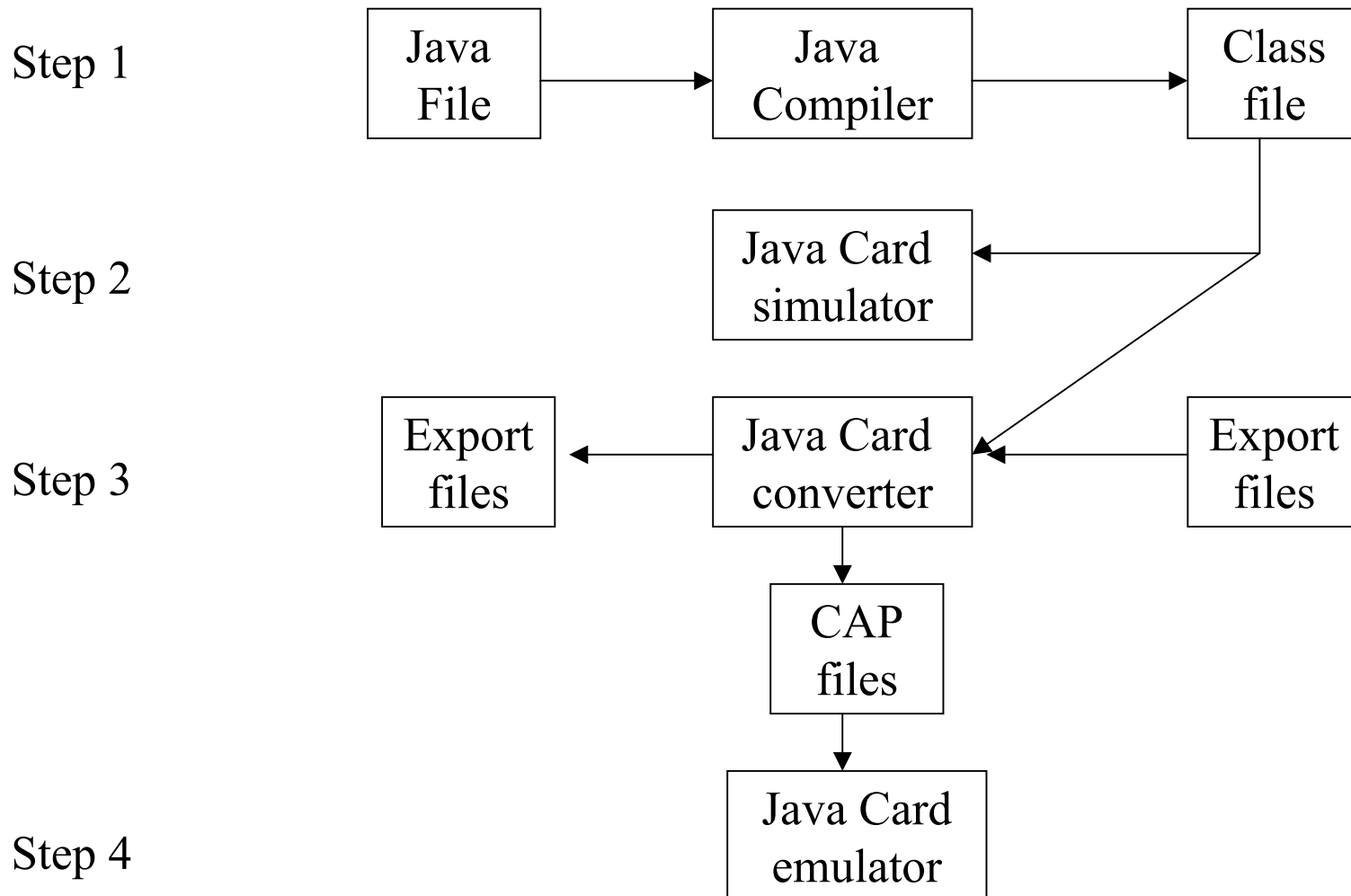
Java Card APIs

- Java.lang Package
- Javacard.framework Package
- Javacard.security Package
- Javacardx.crypto Package

Java.lang Package

Object	Throwable	Exception
RuntimeException	ArithmeticException	ArrayIndexOutOfBoundsException
ArrayStoreException	ClassCastException	IndexOutOfBoundsException
NullPointerException	SecurityException	NegativeArraySizeException

Applet Entwicklungsprozess



Applet Installation

- Rom Applets
- Preissuance oder Postissuance Applets
- Postissuance Applet Installation
- Die Grenze der Installation

Java Card Objekte

- Java Card Speichermodel
- Persistente Objekte
- Transiente Objekte

Java Card Speichermodel

- ROM
- EEPROM (16 K)
- RAM (1 K)

Persistente Objekte

- Ein persistenter Objekt wird von new Operator erzeugt .
- Ein persistenter Objekt bleibt seinen Status und Werte durch die CAD session
- Irgendein Update in einem einzelnen Feld in einem Persistenten Objekt ist atomic .
- Ein persistenter Objekt kann von einem transienten Objekt refiziert werden .
- Ein persistenter Objekt kann einen transienter Objekt refizieren .
- Falls ein persistenter Objekt nicht mehr refiziert werden kann , wird er von Garbage weggenommen .

Transienter Objekt

- Eigenschaften der transienten Objekte
- Die Arte der transienten Objekte
- Die Erzeugung der transienten Objekte
- Die Erzeugung und Zerlegung der Objekte

Die Eigenschaften der transienten Objekte

- Ein transienter Objekt wird vom aufruf der Java Card APIs erzeugt .
- Ein transienter Objekt bleibt seine Status und Werte nicht durch die CAD Session .Die Werte werden wie vereinbarte als (zero ,false,null) definiert.
- Irgendein Update in einem transienten Objekt ist nicht atomic .
- Ein transienter Objekt kann von einem persistenten Objekt referenziert werden .
- Ein transienter Objekt kann einem persistenten Objekt referenzieren .
- Falls ein transienter Objekt nicht mehr von anderen Objekten referenziert werden kann , wird er von Garbage genommen.

Die Arte der transienten Objekte

- `CLEAR_ON_RESET` : speichert durch die Selektion der Applet nicht durch des Card Resets.
- `CLEAR_ON_DESELECT`: speichert so lange die der Applet selektiert wird und nicht durch die Selektion der Applet und des Card Resets .

Die Erzeugung der transienten Objekte

Methods	Result a transient boolean array
Public static boolean [] makeTransientBooleanArray(short length , byte event)	Create a transient boolean array
Public static byte [] makeTransientByteArray(short length , byte event)	Create a transient byte array
Public static short[] makeTransientShortArray(short length, byte event)	Create a transient short array
Public static Object[] makeTransientShortArray(short length , byte event)	Create a transient object array

Atomocity und Transaktionen

- Atomicity
- Block Data Update in einem Array
- Transaktionen

Block Data Update in einem Array

```
Public static short arrayCopy (byte[] src ,short srcOff ,  
                                byte[] dest ,short desOff ,short length )
```

```
Public static short arrayCopyNonAtomic (byte  
src ,short srcOff ,byte[] dest ,short desOff ,short length )
```

```
Public static short arrayFillAtomic (byte [] bArray ,  
short bOff , shoet bLen ,byte bValue )
```

•Transaktionen

- Commit Transaktion
- Stoppen Transaktion
- Vernetzte Transaktion
- Commitkapazität
- TransaktionException

Commit Transaktion

//begin a transaction

Jcsystem.beginTransaction()

//allmodificacions in a set of updates of persistent data

//are temporary until the transaction is committed

...

Commit a transaction

JCSystem.commitTransaction();

TransaktionException

- IN_PROGRESS
- NOT_IN_PROGRESS
- BUFFER_FULL
- INTERNAL_FAILURE

Zusammenfassung

- Smart Card Hardware
- Smart Card Kommunikation
- Java Card Applet Installation
- Java Card Memory Model
- Transiente und persistente Objekte
- Atomicity und Transaktion

Vielen Dank für Ihre
Anwesenheit!