Smart Cards a(s) Safety Critical Systems

Gemplus Labs

Pierre.Paradinas@gemplus.com

GEMPLUS Labs

Agenda

- Smart Card Technologies
  - Java Card™
- Smart Card a specific domain
  - Card Life cycle
  - Our Technical and Business constraints
- FM and safety card development

Historical account

- 1967: First idea on the use of electronic component in credit card (Europe, US, Japan).
- 1974: Roland Morenos patents
- 1979: First Bull CP8 card prototype
- 1982-1984: First experimentation in France
- 1987-1989: ISO standard
- 1990-1999: Applications
  - French “Carte Bleue” for banking
  - European mobile phone with GSM/SIM cards
  - Health insurance, e-purse,…
- 1997: First Java based open card

Smart Cards Standards (1/2)

- ISO 7816-1
  - Physical characteristic, constraints, size
- ISO 7816-2
  - Dimension and location of the contacts
- ISO 7816-3
  - Electric signal and transmission protocols
  - Card Answer to Reset: information about card characteristic
  - T=0; T=1

Smart Cards Standards (2/2)

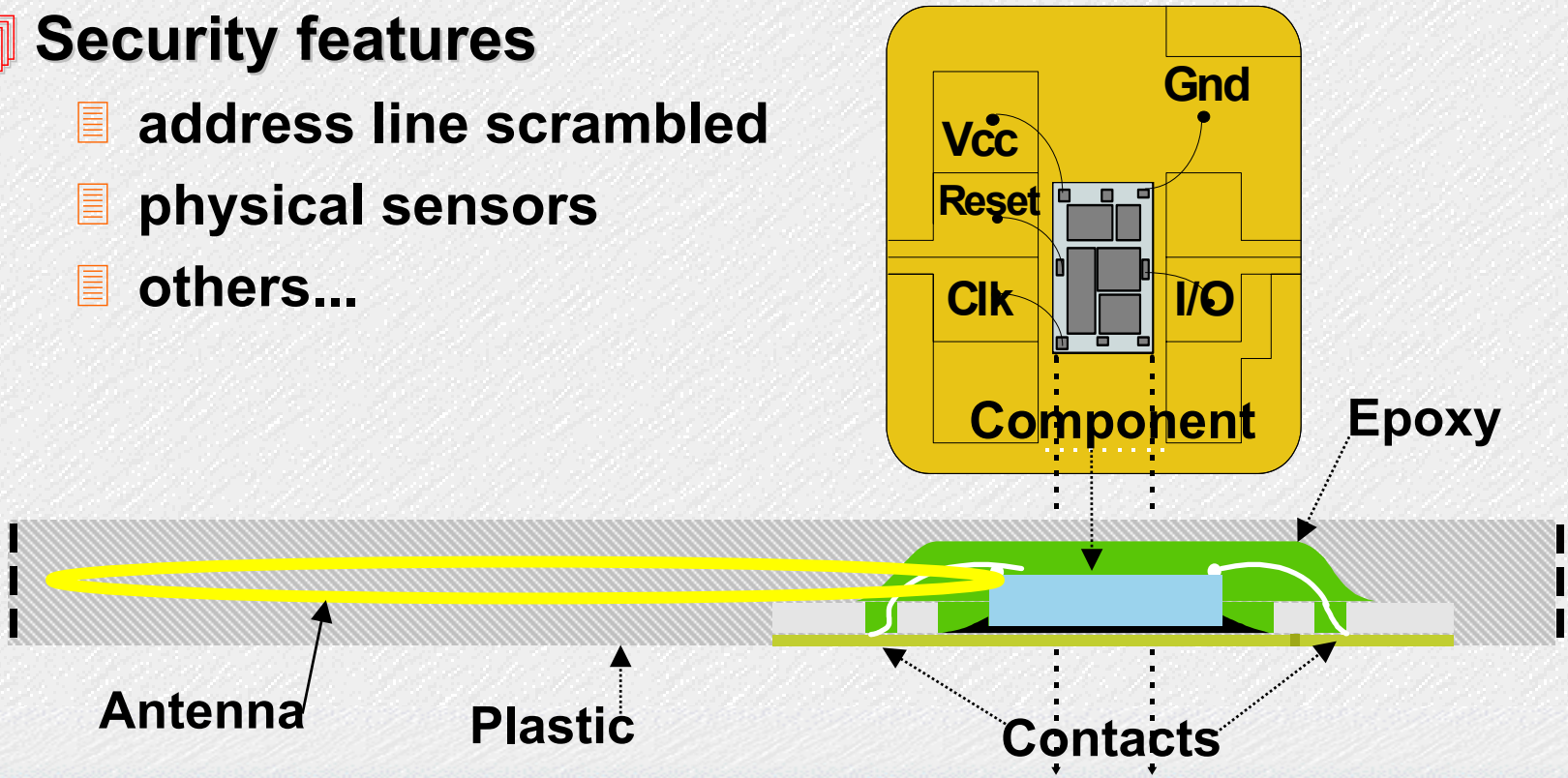
- ISO 7816-4
  - Structure of the exchanged messages of command - response
  - APDU Application Protocol Data Unit.
- ISO 7816-5
  - Application identifiers
- ISO 7816-6
  - Data element of interchange
- ETSI GSM 11.1: Command messages for SIM cards
- EMV: Command messages for payment cards
- JC 2.1...

Different Kind of Cards

- Memory cards
  - A simple memory without a processor
  - Data card contains data burned in read only memory
  - Token card: one bit in memory = one token (phone card)
- Memory cards with logic
  - Token card with electronic control to enhance security
- Microprocessors cards (smart cards)
  - A module includes a processor with RAM, ROM and EEPROM, the COS and the application.

Smart card modules

- Power and clock provided by the reader
- Chip hidden under the contacts into a glue
- Single chip (w/o a cryptographic-processor)
- Security features
  - address line scrambled
  - physical sensors
  - others...



Smart Card Microcontrollers

- Microcontrollers
  - 8 bit for low cost application
  - 16/32 bit will be used
- Limited resources
  - ROM 8 to 64 kb; contain the burned OS
  - RAM 256 to 2 kb; fast and volatile, used as working memory
  - EEPROM 2 to 64 kb; used as memory storage, slow and subject to wear (anti stress mechanism).
- Only one communication line (half duplex)

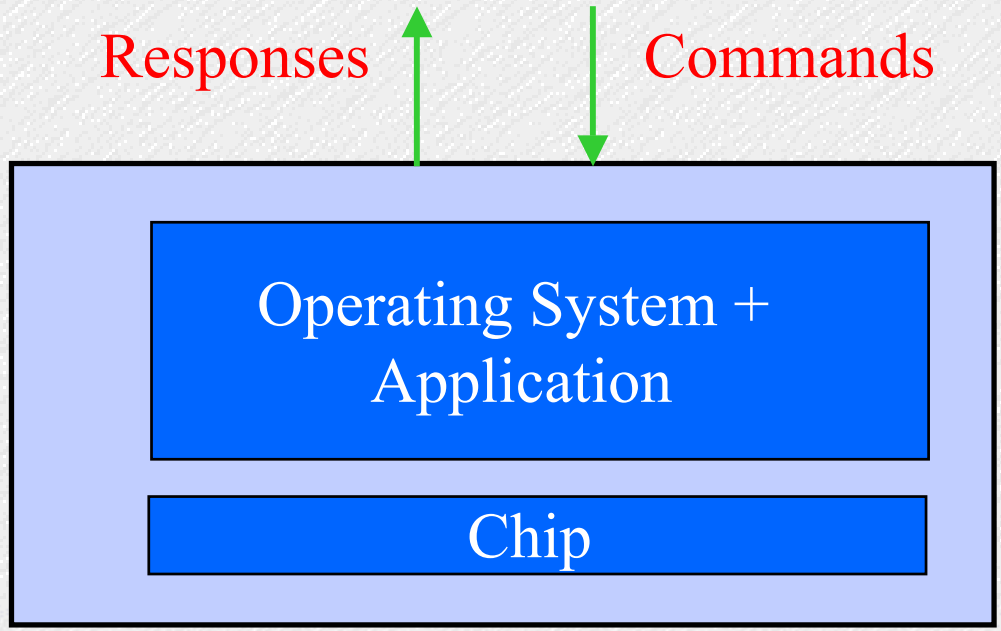
Small Software

- Some thousand lines: tractable with the current tools,
- Only sequential code,
- Limited number of features,
- Public domain specification (Java Card),
- Reactive system with one I/O line,

- Assembly and C are used….

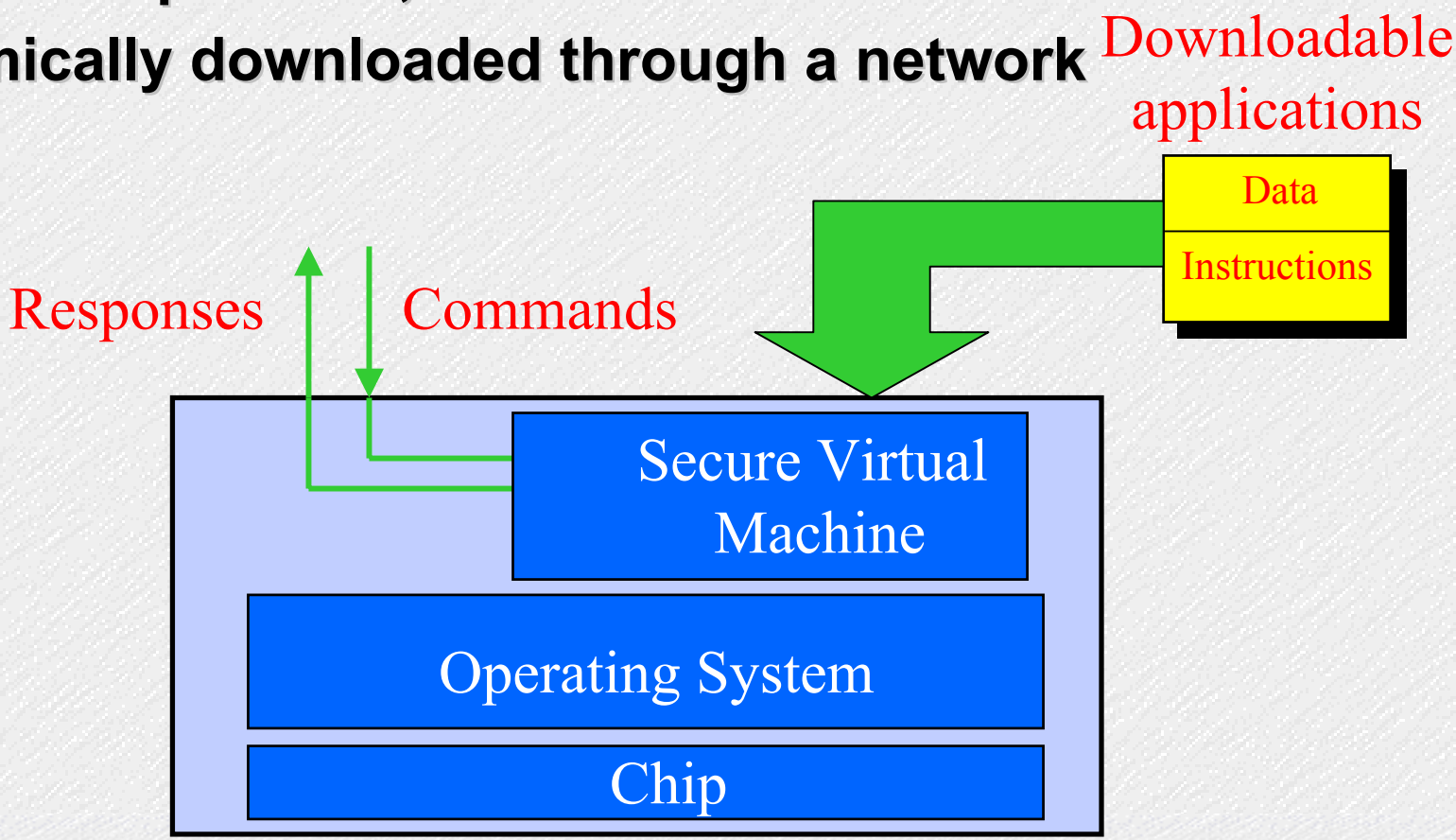
Motivations of Open Card

- Applications are developed by the card provider in a secure environment,
- Drawbacks:
  - time consuming
  - costly
  - poor flexibility
  - time to market



Open Cards...

- Applications developed by the customer or any application provider,
- Dynamically downloaded through a network



Introduction to the Java card

- The Java Card
- The JVM architecture
- The security procedures

What is a Java card ?

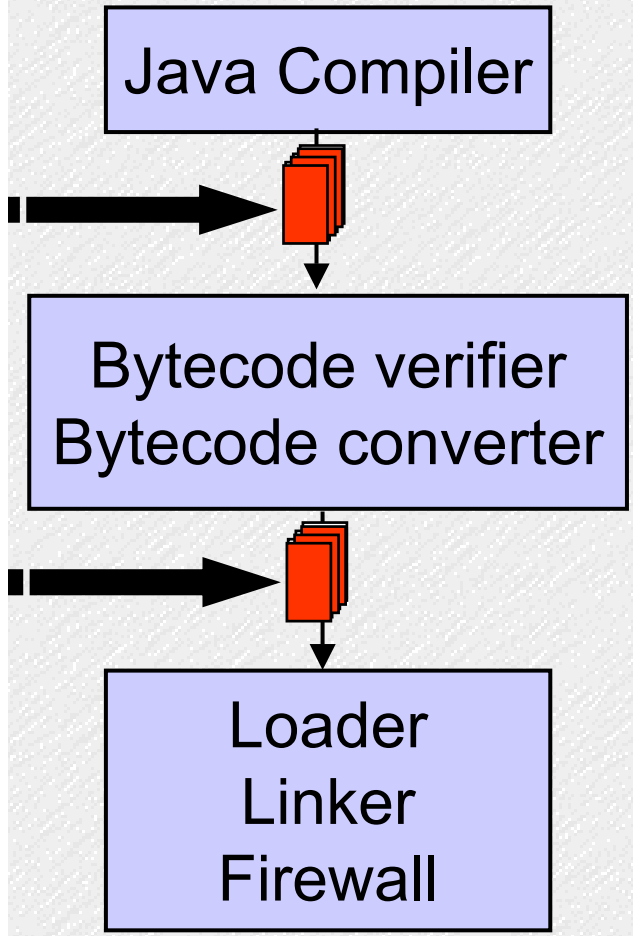
- The Java Card
  - a smart card dedicated to Java applications
  - a platform with highly limited resources
  - a dedicated Java language
  - a multi-application device
  - a specific Java Virtual Machine (JVM) architecture.

A subset of Java

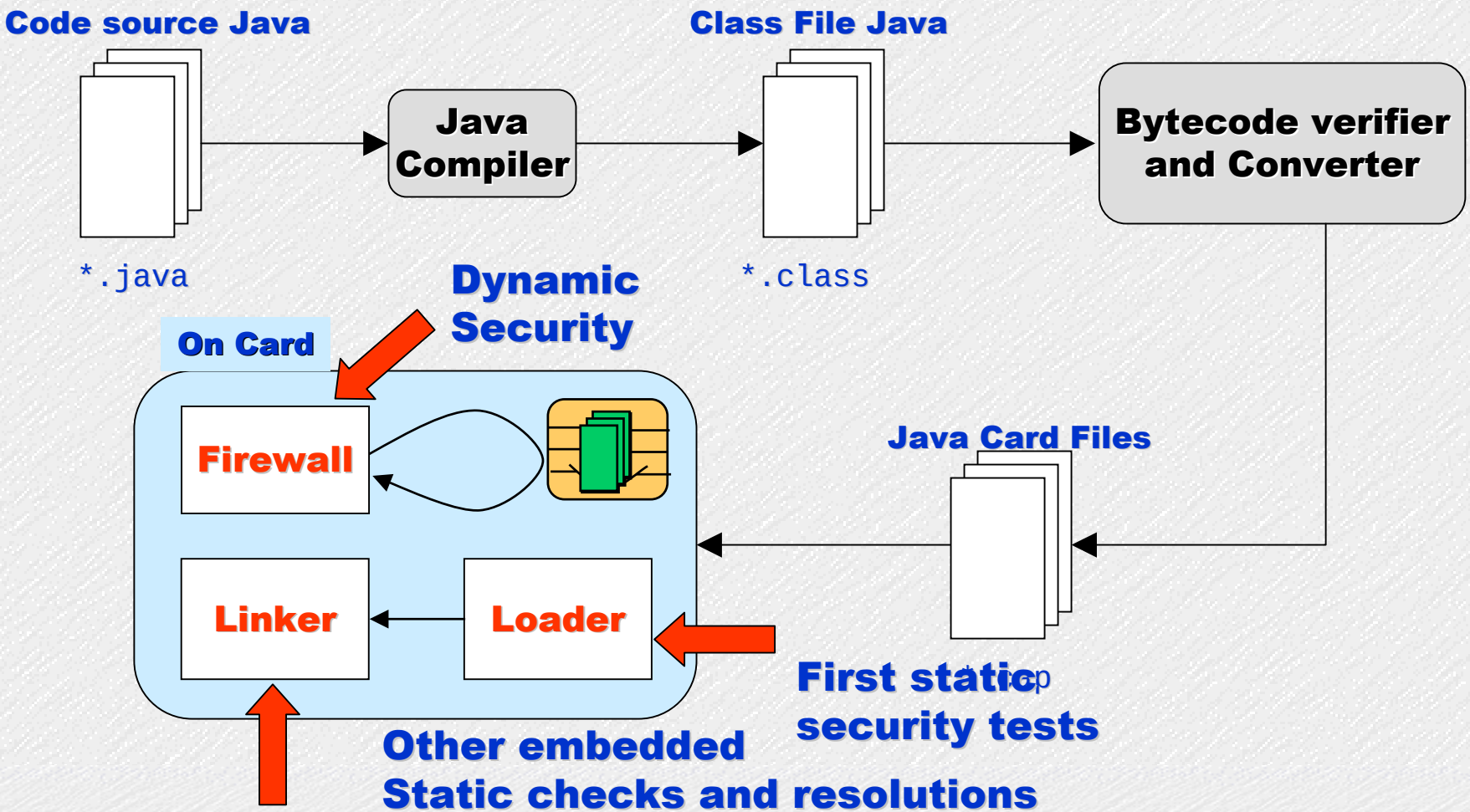
- A single thread virtual machine
- Unsupported features
  - Dynamic class loading
  - String and Thread classes
  - Double, float, char types
  - multiple dimension arrays
  - java.lang.System class
  - Garbage collection
  - Security manager
- The Applet Firewall
- Programming limitations

The JVM architecture

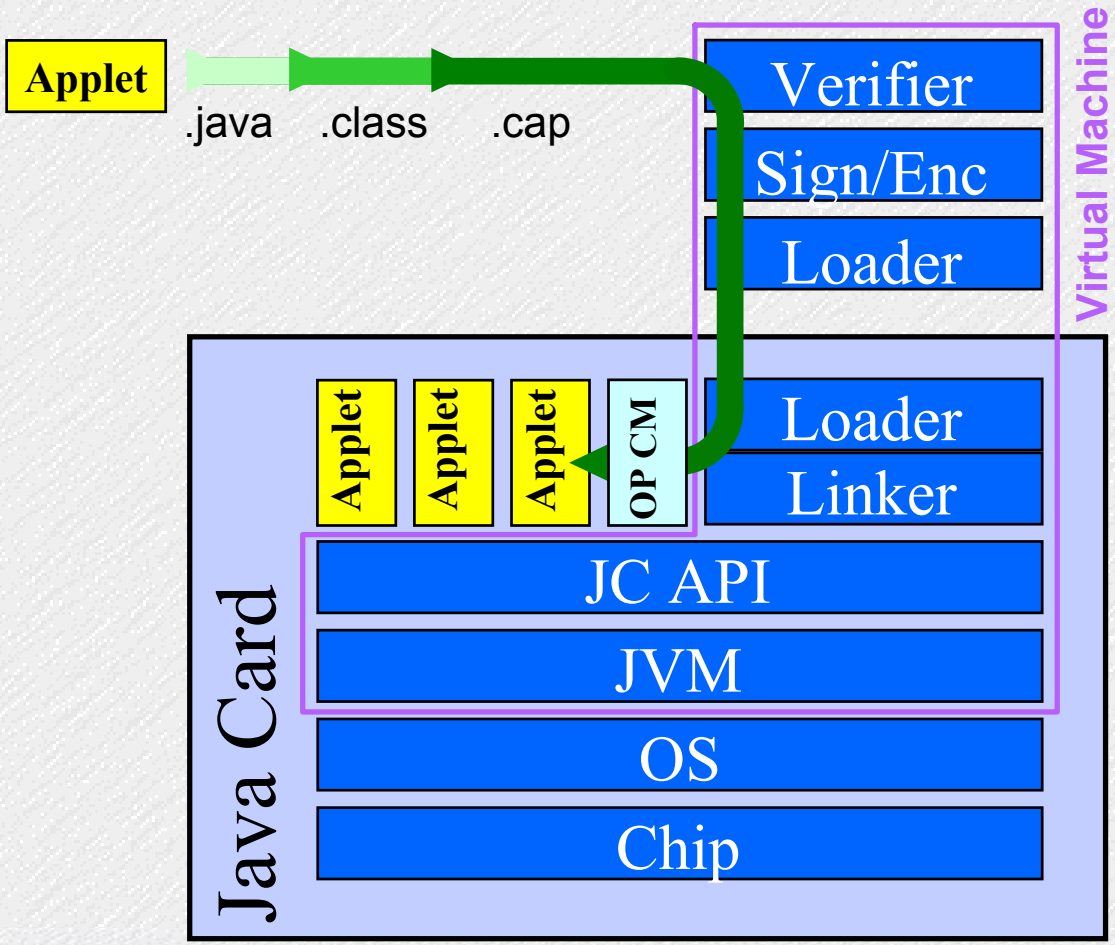
- Developer property
- Applet provider
- Java card features



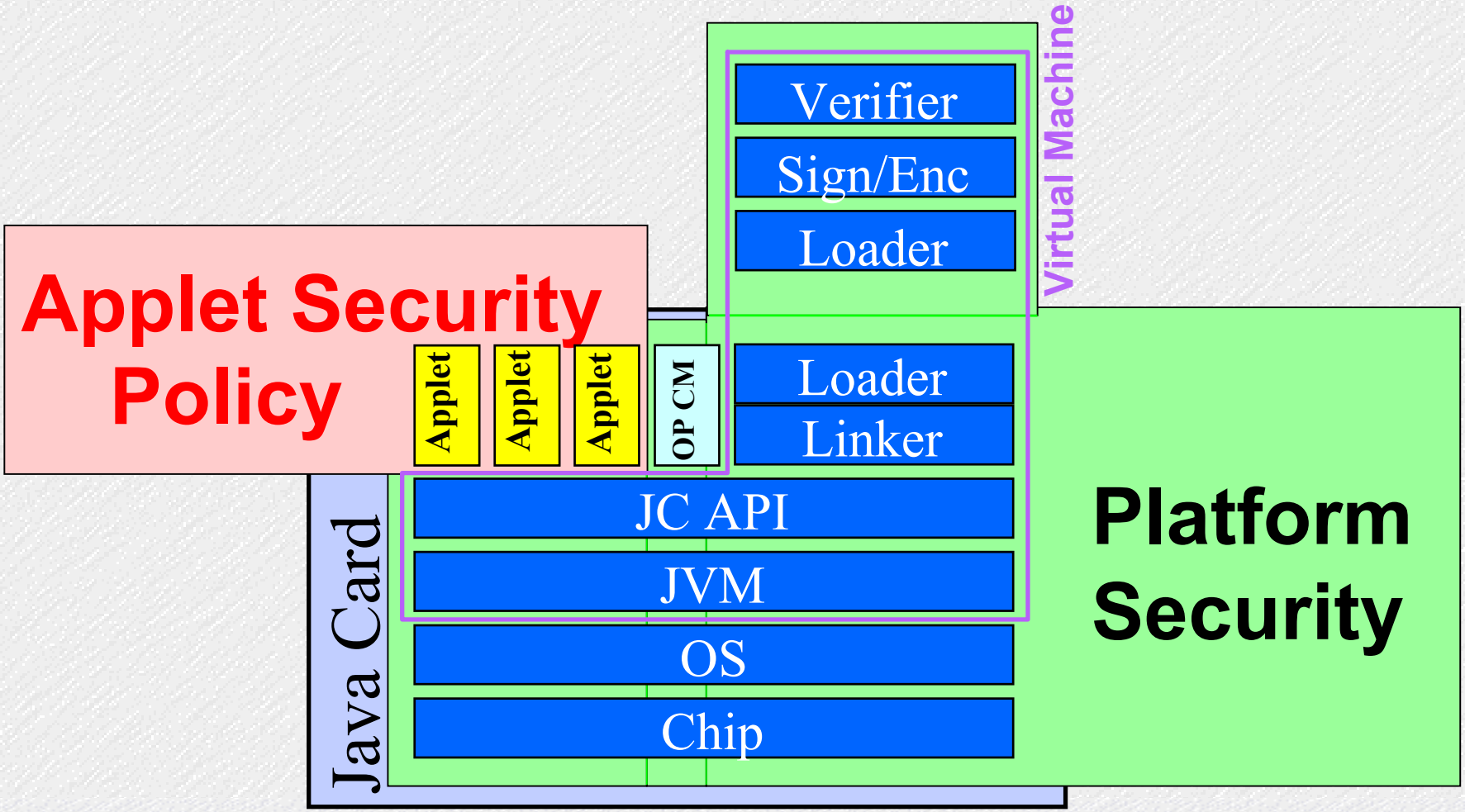
Java Card Environment



Java Card Security Chain

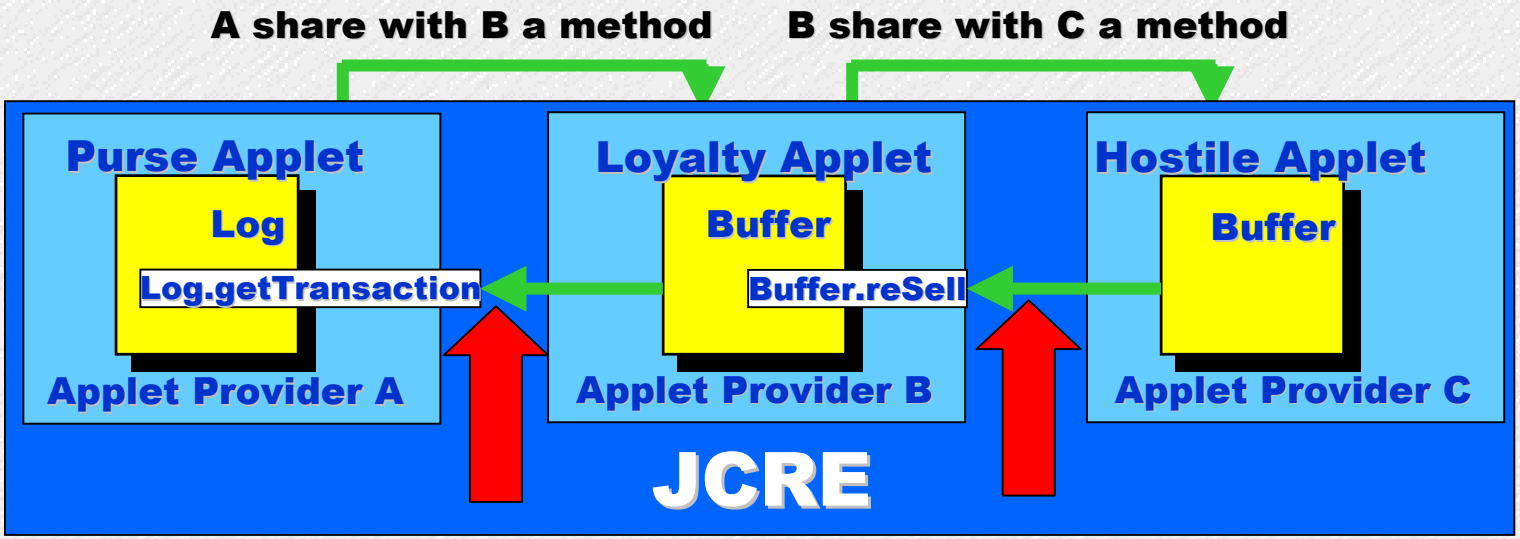


Java Card Security Chain



...and the sharing mechanism

- The Java Card specification provides a mechanism to share data between several applets,
  - For example: a purse and a loyalty applet can share methods and/or objects,
  - Due to the limited resources of the smart cards new services or libraries will be offered.



Two security levels

- Applications are no more developed under card issuer control,
- Platform security
  - Traditional means,
  - Use of formal methods.

  => Models of the platform security modules
- Application security
  - There is a need for a global security policy
  - Flow control (data and/or code sharing)
  - Resources consumption (memory, CPU, method calls...)

  => Static analysis of applet configurations (part of the CMS)

Smart card... a specific domain ?

- Short development cycle
- Short life time
- Mass product, million of smart cards
- A specific life cycle

Smart Card Lifetime (1/2)

- **Manufacturing**
  - **Application masked in the ROM**

    OS libraries and command dispatcher,

    Application routines.
  - **Card serial number and issuer references**
- **Initialisation**
  - **Writing in EEPROM application data**
  - **Secret key and object attributes (r,w,rw,...)**
- **Personalisation**
  - **Writing in EEPROM card holder data**
  - **Graphical (picture, logo, hologram…)**

Smart Card Lifetime (2/2)

- Usage
  - Process APDU command from a reader
  - Send back a response APDU or an error APDU
  - For open card only: application downloading
- End
  - Deactivation (unauthorized action), memory overhead, loss, theft, …

Smart Card domain is different from the traditional formal methods application domain

External constraints

- **Certification need**
  - **National rules (*e.g.* German and Hungarian market)**
  - **Required by customers**
- **Certification requirements**
  - **Common Criteria EAL5, ITSEC E-4,**

    Security policy modeling (SPM), proof of the coherence,

    Semi formal correspondence between SPM and HLD
  - **Higher level EAL 7**

    Proof of the high level and detailed design.

    Proof of the implementation of the security policy by the security functions
- **Specific customer requests**

Internal constraints

- Specifications become more complex
  - Increasing number of functionality related with the memory size increasing,
  - Complexity of the new OS based on virtual machine.
- Gemplus masters traditional OS development
  - But... OS qualification is very costly,
  - And FM can reduce development cycle by automatically generating the test suites.

The killer application for FM ?

It seems that smart card domain is the ideal field where…

- FM can be applied (reasonable size),
- Potential benefit seems to be interesting.

Formal Methods and Smart Cards

- Java Card Verifier using a Model Checker [Posegga],
  - They transform each method of an applet into a state transition system (SMV),
  - They propose an abstraction (type),
  - The state is given by the virtual machine state
  - Security properties as temporal formulae are verified with the model checker

Formal Methods and Smart Cards

- And more recent works... :
  - Proof of a verifier using Coq on the [F&M] subset of the byte code [Bertot], Kestrel Institute [Qian],
  - Modelling of a large subset of the Java Card Byte Code in B [Gemplus], Isabelle [Nipkow] and Coq [Jakubietz],
  - The Loop project at Niemegen University [Poll],

...and others...

... BUT... difficult to put in practice

- Economic constraint: the smart card is a mass product
  - No development overhead is admissible (except the certification process)
- Development process constraints:
  - The software must be provided to the chip manufacturer within a given deadline,
  - Very small life time,
  - Highly optimised software.

What is missing...

- Lack of metrics with formal developments
  - Imprecise industrial reports on formal developments,
  - Difficulties to predict time development,
  - Hard to estimate the cost.
- Methodology
  - Modelling and proving are not common activities,
  - Links between semi formal and formal specification.
- Tool improvements
  - Code generator (C and BC Java)

Gemplus works in progress

- EC funded Projects
  - Verificard: improve the safety of both the JavaCard platform and Java Card applications, using formal methods
  - Matisse: methodology and metrics (MATISSE) to use FM in an industrial context
- French funded Project
  - BOM: provide tools (Code generator)