Smart Card Concepts

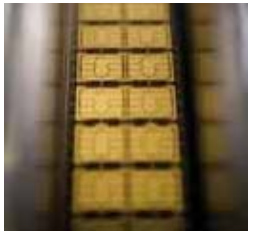
Carte à puce et Java Card

Master SIC

2009-2010

Jean-Louis Lanet

Jean-louis.lanet@unilim.fr



Université de Limoges | FACULTÉ DES SCIENCES ET TECHNIQUES

Agenda

- Card Technology
- Standards
- Manufacturing
- Operating system

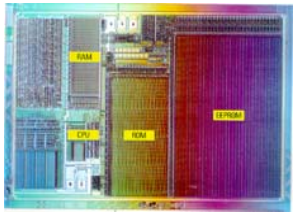
Magnetic-strip cards

- Defined by ISO 7811-2 (properties) -4 (coding) –5 (location of the magnetic stripes)
- Storage capacity 1000 bits

Features	Track 1	Track2	Track 3
Amount of Data	79 char	40 char	107 char
Data Coding	6 bit alpha	4 bit BCD	4 bit BCD
Data density	210 bpi	75 bpi	210 bpi
Writing	Not Allowed	Allowed	Allowed

What is a Smart Card?

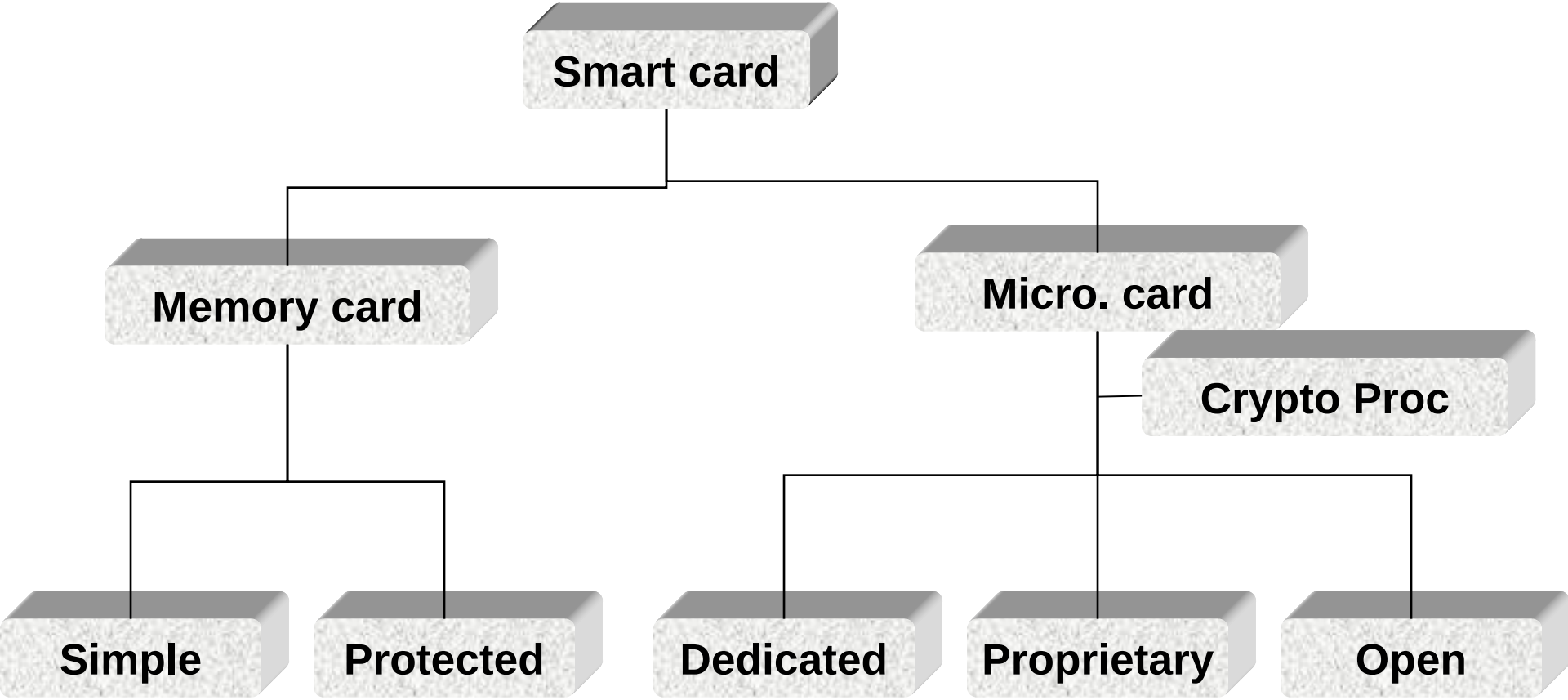
A piece of silicium on a plastic body



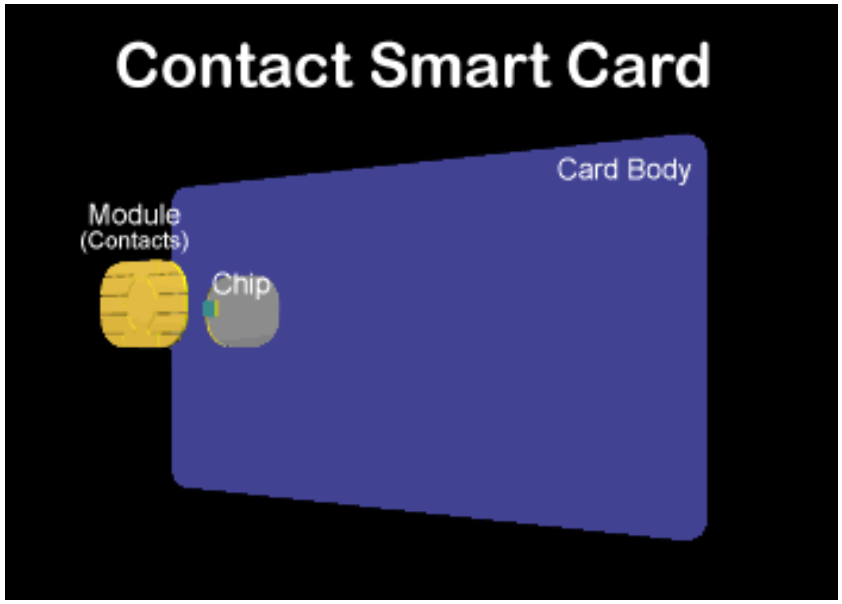
Chip

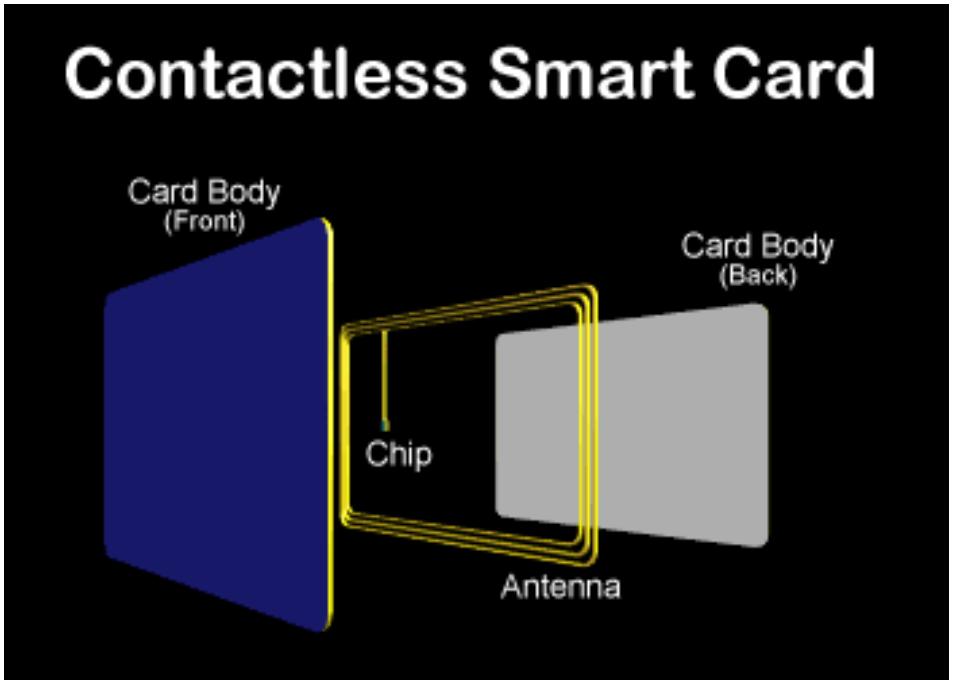
A very secure way of storing a small amount of sensitive data

Smart Cards


Smart card
Memory card
Micro. card
Crypto Proc
Simple
Protected
Dedicated
Proprietary
Open

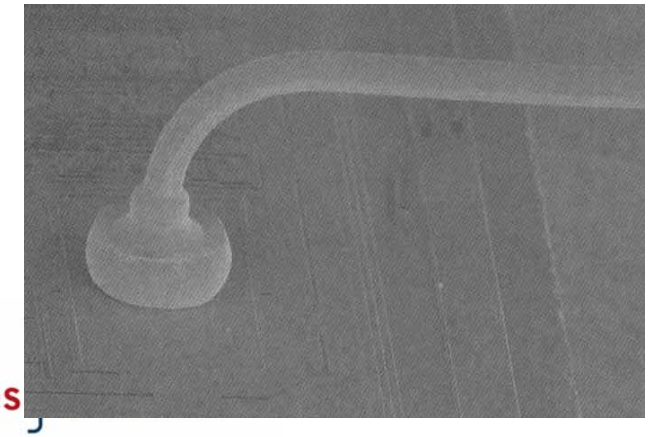
Contact / Contact less

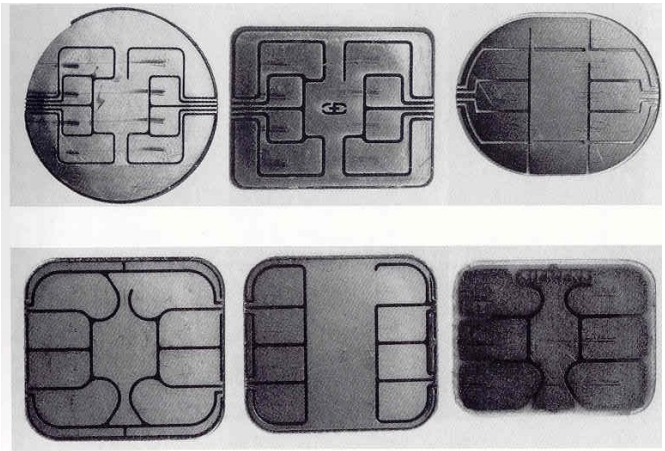




Contact

- Electrical connections between the chip and the module (wire bonding process),
- 8 contacts (C1-C8) but only 6 used (see ISO7816-2),
- C6 used as Vpp while EEPROM where not embedding charge pump,
- Supply voltage 2,7v (SIM) to 5,5v (standard TTL) and clock provided by the reader.





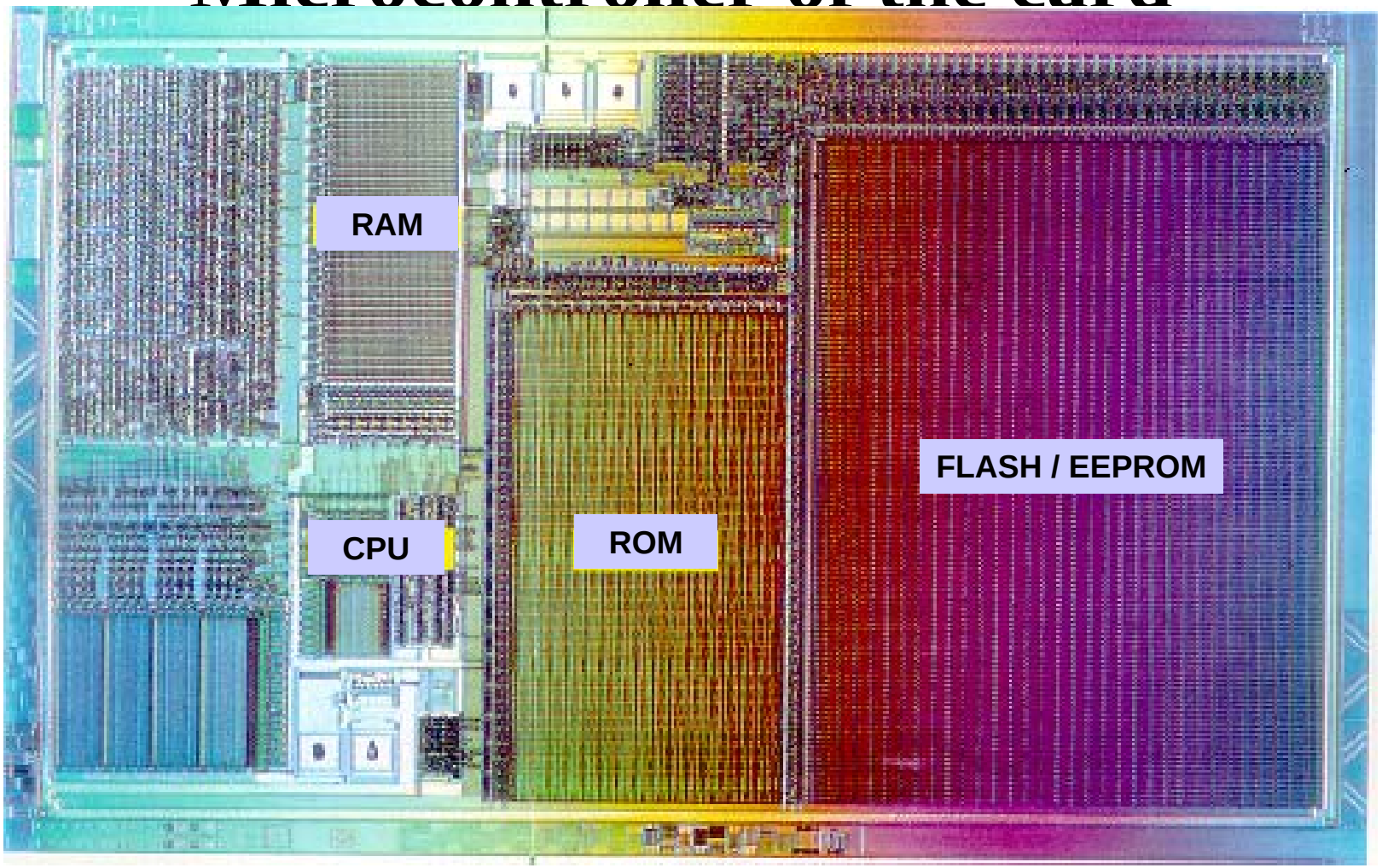
Contact less card (NFC)

- No electrical connection (*cf.* RFID technology) used of inductive coupling to supply power to the chip,
- Need : modulator, demodulator, anti-collision mechanism, voltage regulator, reset generator and an aerial.
- For data transfer all known digital modulation techniques can be used (ASK, FSK and PSK).
- Standards : close coupling ISO/IEC 10536 (3-5Mhz), proximity cards ISO/IEC 14443 (13,56Mhz) and Hand Free Cards ISO/IEC 15693,
- Used for public transportation, ski pass, access control, payment with GSM…

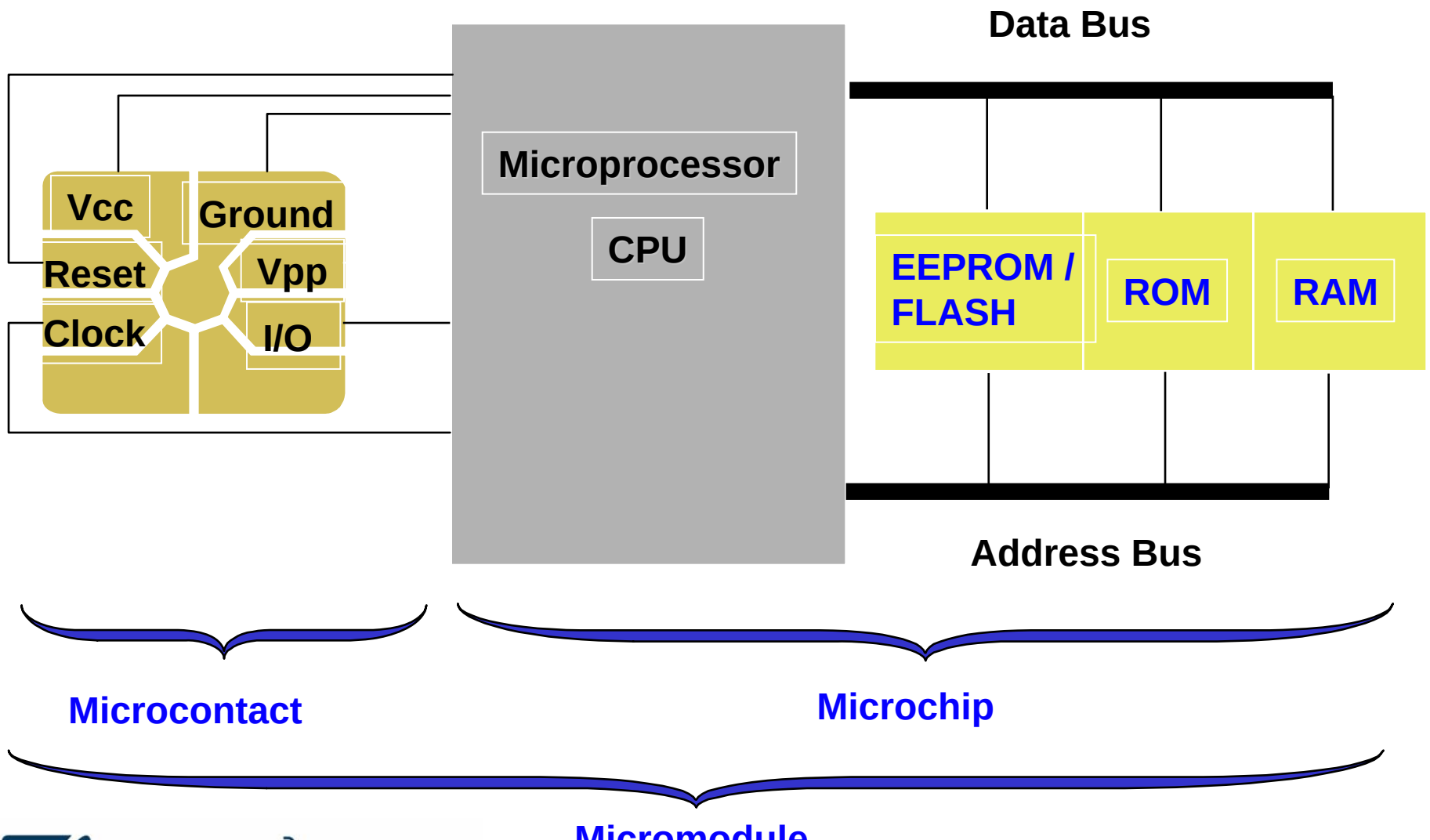
Form Factor

- With contact:
  - ISO 7810, 7816-1,1816-2
  - USB
- Contactless : several standards
- Hybrids
- Buttons
  - iButton (1-wire)
  - JavaRing
- Dongle (serial, parallel, USB, mmc)

Microcontroller of the card



Contact card



Smart Card memories

	RAM	EEPROM	FlashRAM	FéRAM
Persistency	No	Yes	Yes	Yes
Read acc.	0.1µs	0.15µs	0.15µs	0.15µs
Write	0.1µs	10µs	10µs	0.4µs
Erase	-	5ms	100ms	-
Granularity	-	4bytes	64bytes	-
Cycles	Unlimited	10^6	10^5	10^{10}

Comparing Smart Card vs. PC

	Smart Card	PC	Ratio
RAM	1kbyte	128Mbyte	130 000
Storage	64kbyte	6Gbytes	100 000
Baud rate	192 kbits	100Mbits	500
CPU Speed	20 Mips	500Mips	25

Future…

- Screen Keyboard
  - Protection again false terminal
- On board clock
  - Avoid timing attack
- A battery
  - Volatile memory is hardier to observe
  - New functionalities
- Server, http, iPV4, iPV6…

Source Alien Technology http://www.alientechnology.com

Agenda

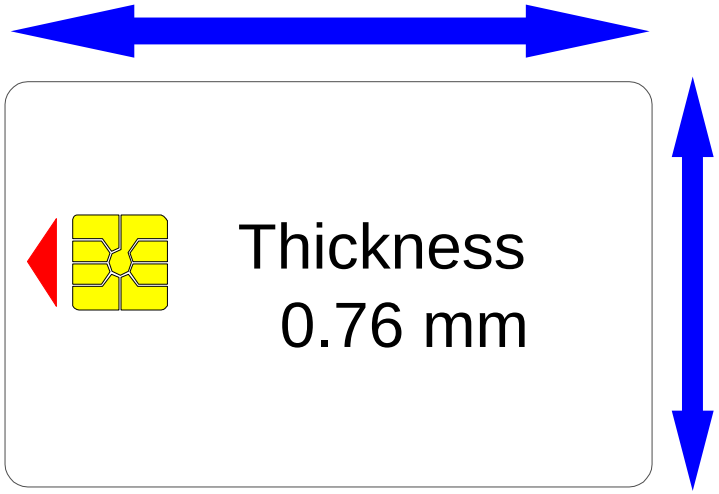
- Card Technology
- Standards
- Manufacturing
- Operating system

ISO/IEC 7816
Integrated circuits cards with contacts

- ISO/IEC 7816-1 : Physical characteristics.
- ISO/IEC 7816-2 : Dimension & location of contacts.
- ISO/IEC 7816-3 : Electronic signals & transmission protocols.
- ISO/IEC 7816-4 : Inter-industry commands and file system.
- ISO/IEC 7816-5 : Registration system for applications in IC card.
- ISO/IEC 7816-6 : Inter-industry data elements.
- ISO/IEC 7816-7 : Inter-industry commands for Structured Card Query Language (SCQL).
- ISO/IEC 7816-8 : Security architecture and related inter-industry commands.

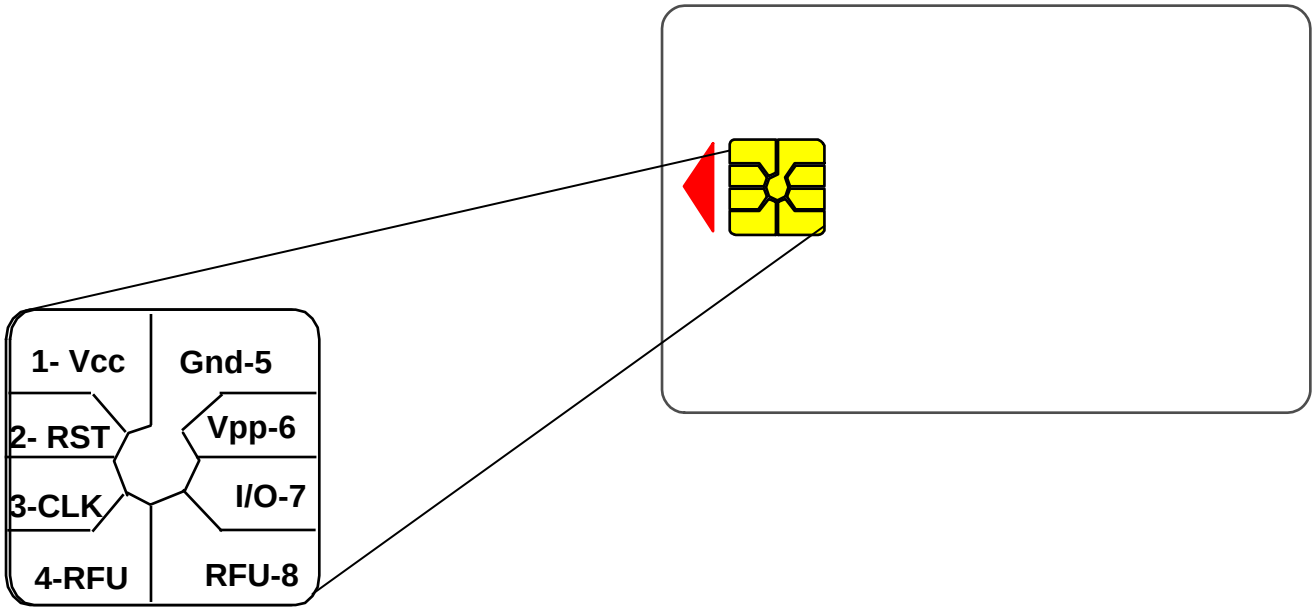
ISO/IEC 7816-1 (7810)

- Governs the physical characteristics of a smart card :



ISO/IEC 7816-2

- Governs the dimension and location of the chip contact :



ISO/IEC 7816-3

- Electrical characteristics :
  - clock frequency [1 MHz, 5 MHz],
  - communication speed.
- Transmission protocols :
  - T=0, T=1, T=CL defined,
  - T=14 reserved for proprietary protocols.
- Answer to reset (ATR)
- Protocol type selection (PTS) :
  - If several protocols supported or if parameters need to be adjust
  - Negotiable mode and specific mode

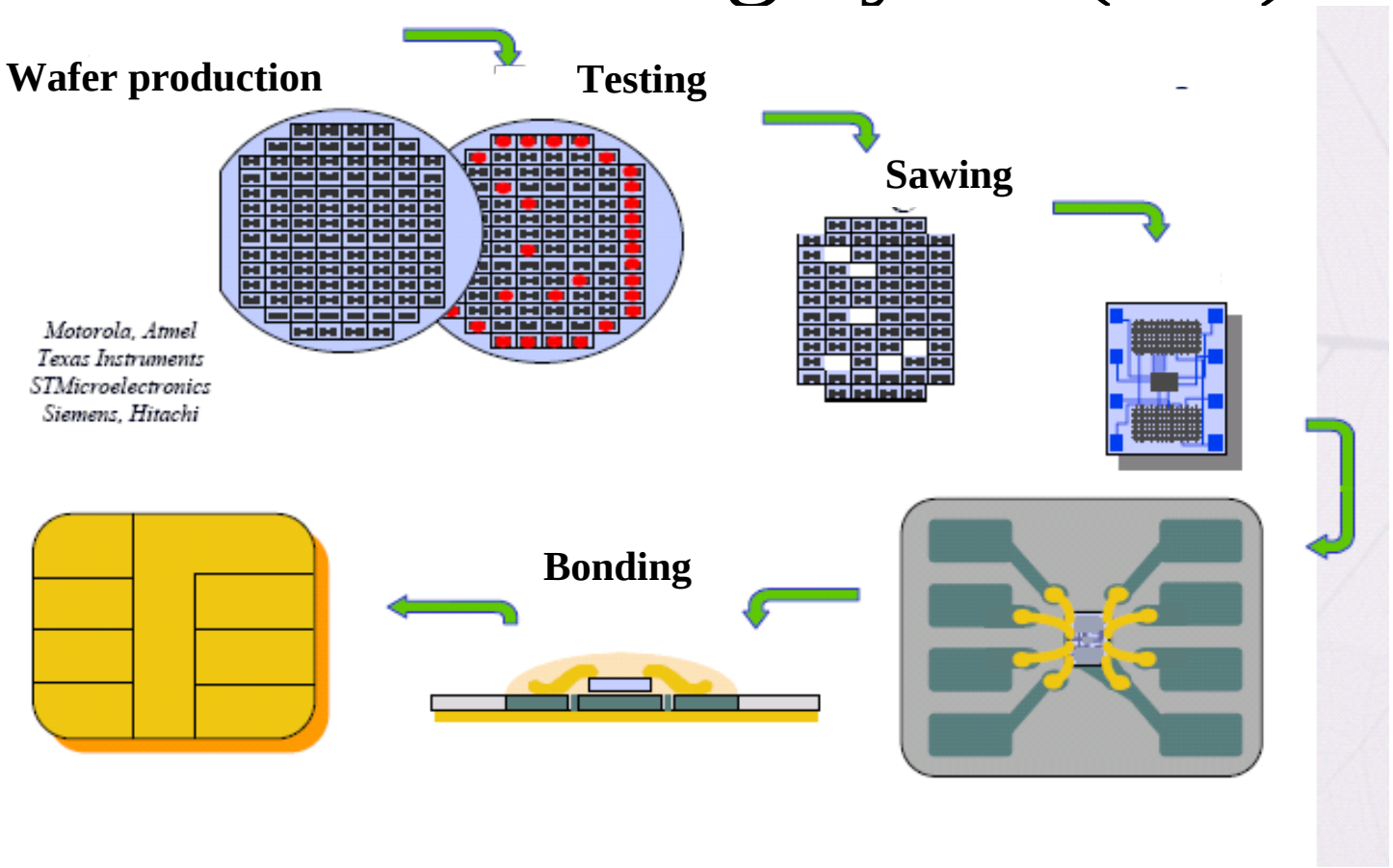
ISO/IEC 7816-4

- There are no user programs, no memory management and no parallelism.
- It just defines the file system
  - Specifies contents of messages (commands, responses).
  - Structure of files and data.
- and the security architecture
  - Access methods to files and data.
  - Methods for secure messaging.
- But also the filter mechanism.

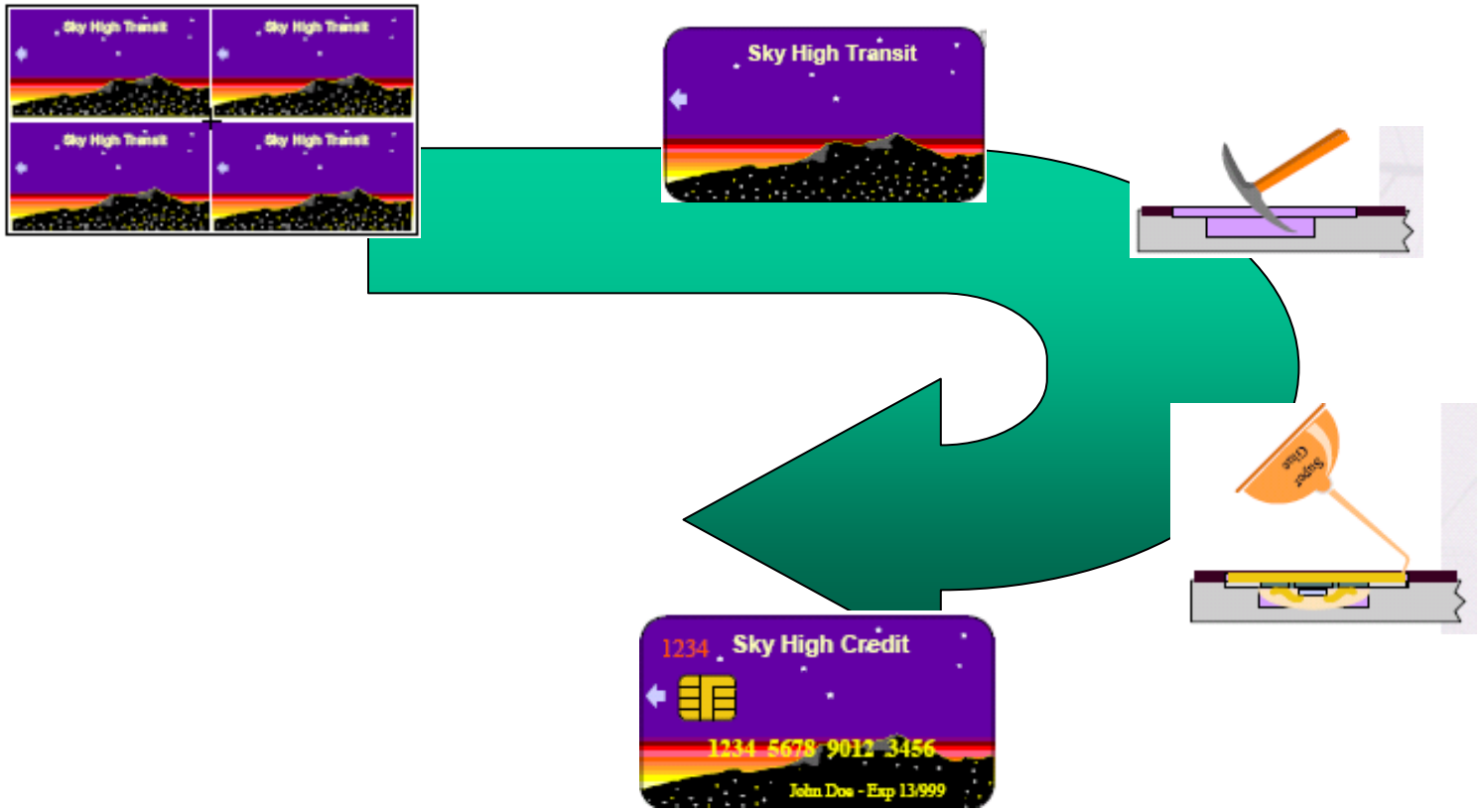
Agenda

- Card Technology
- Standards
- Manufacturing
- Operating system

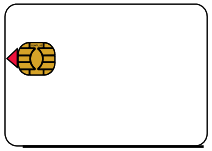
Manufacturing cycle (1/3)



Manufacturing cycle (2/3)



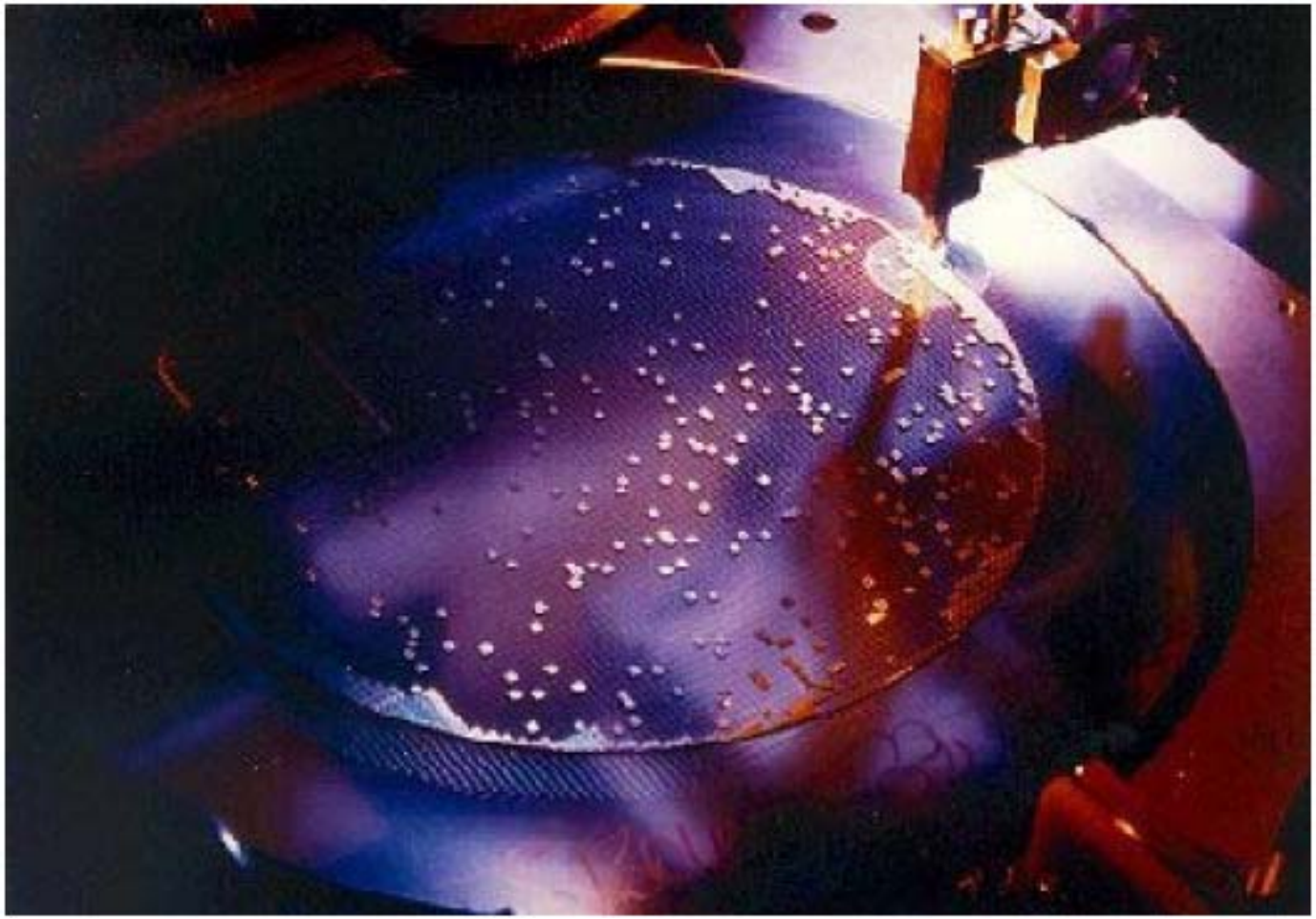
Manufacturing Cycle (3/3)



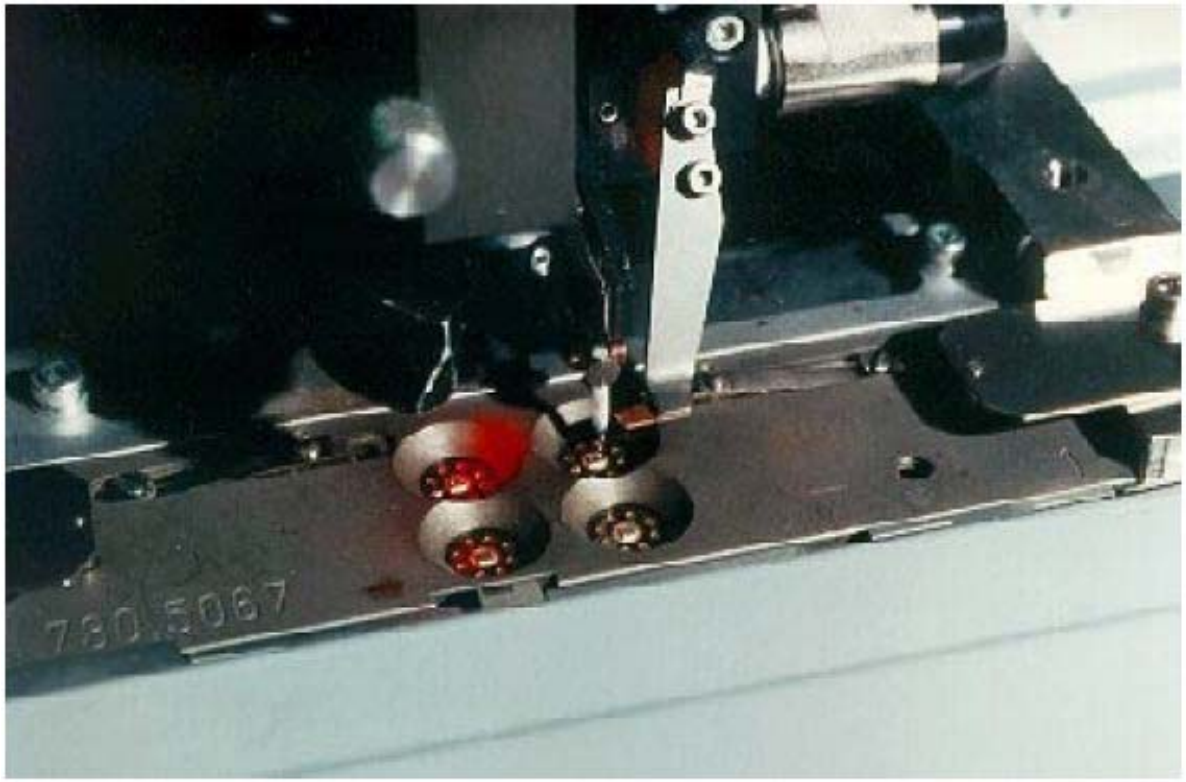
Manufacturing : Sawing

Cutting silicon wafer into individual chips. During the previous step, electrical test, defective chips are marked with an ink drop.

Manufacturing : Testing

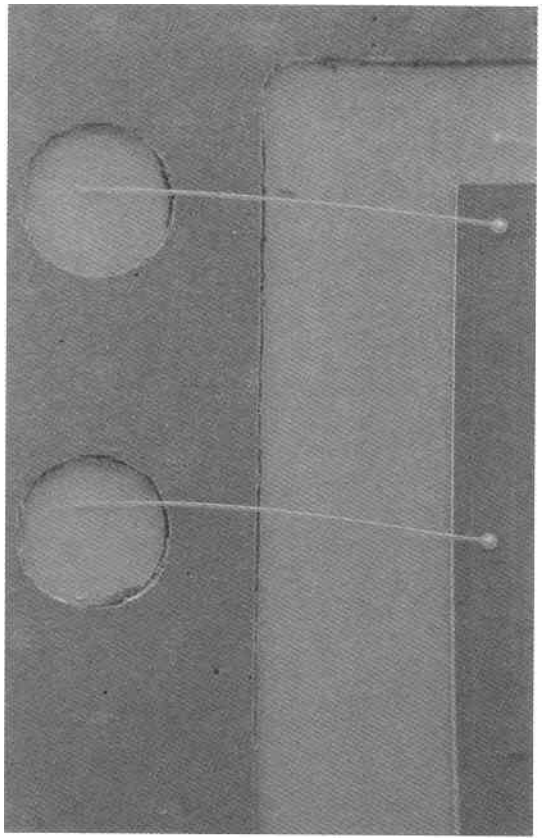


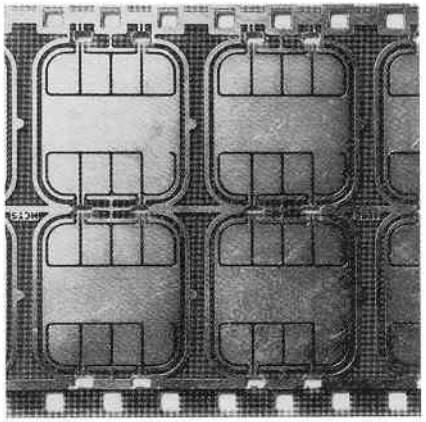
Manufacturing : Bonding

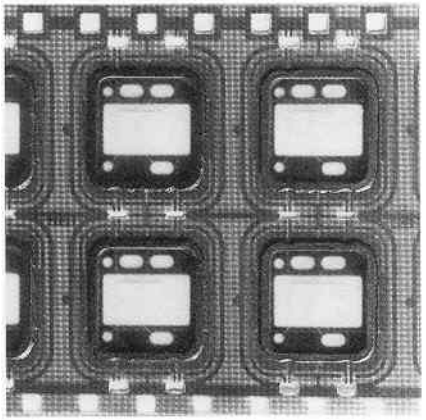


Electrically connecting the chip's bonding pads and the contacts on the micro module using gold wires.

Bonding



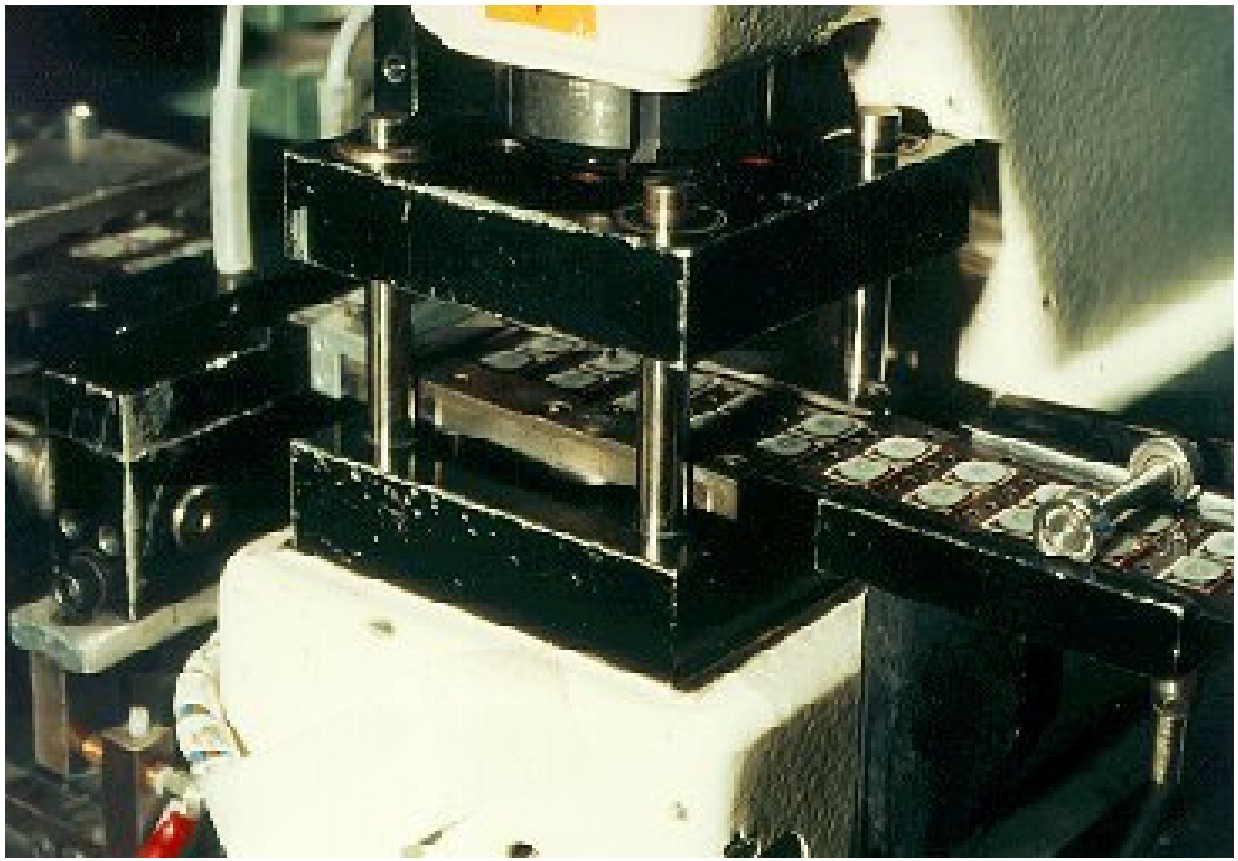




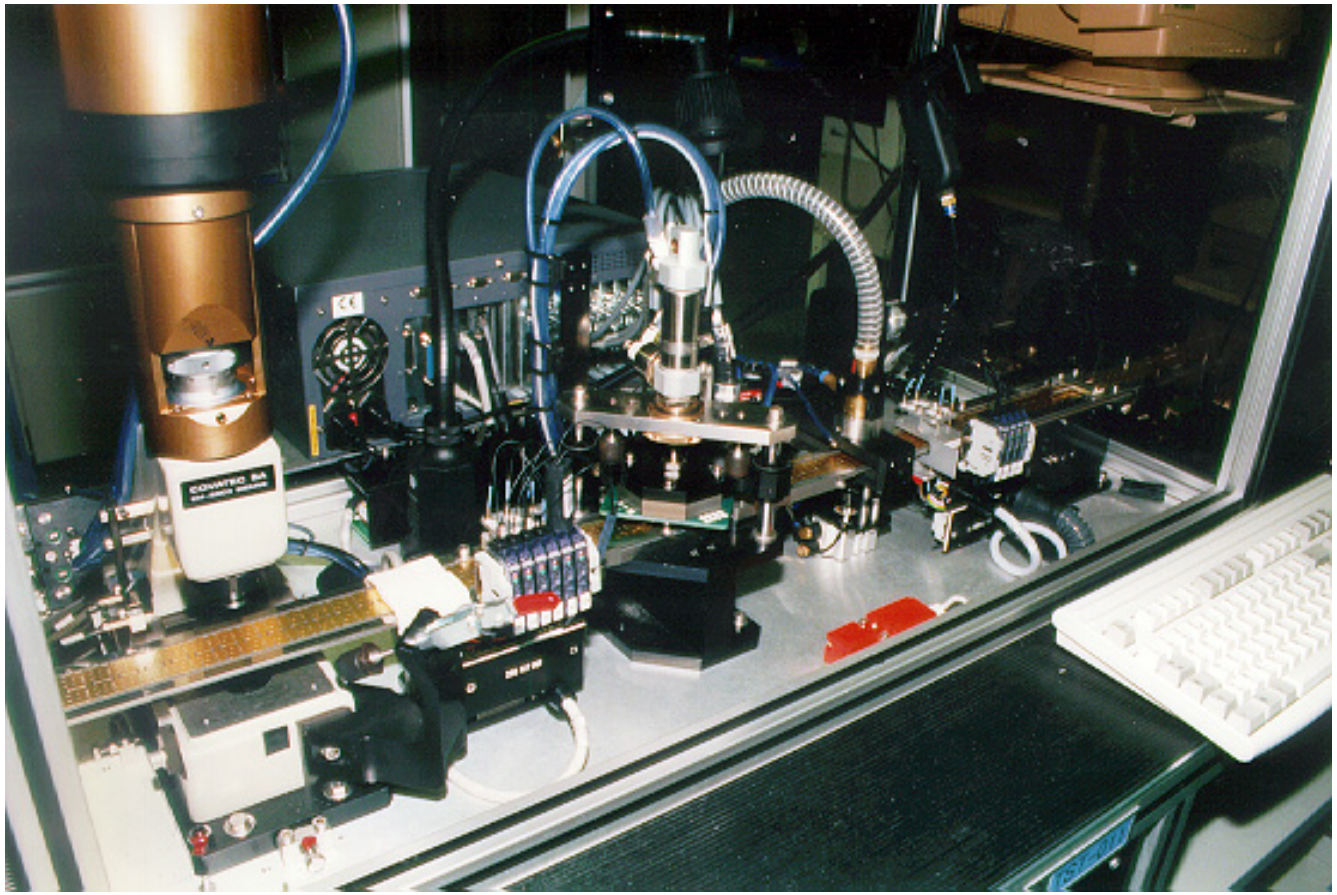
Manufacturing : potting

Protecting the chip and wires with a drop of epoxy resin, ensuring the physical durability of the micro module

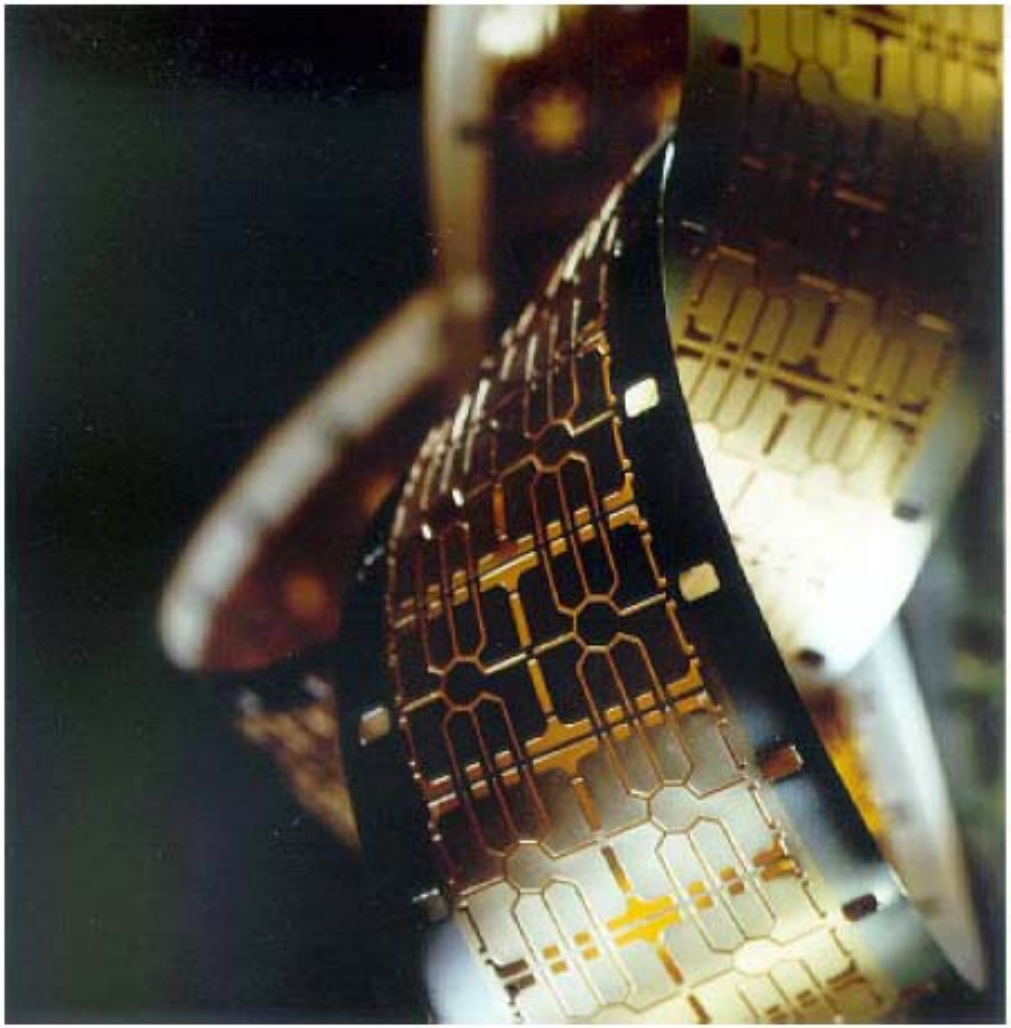
Grinding



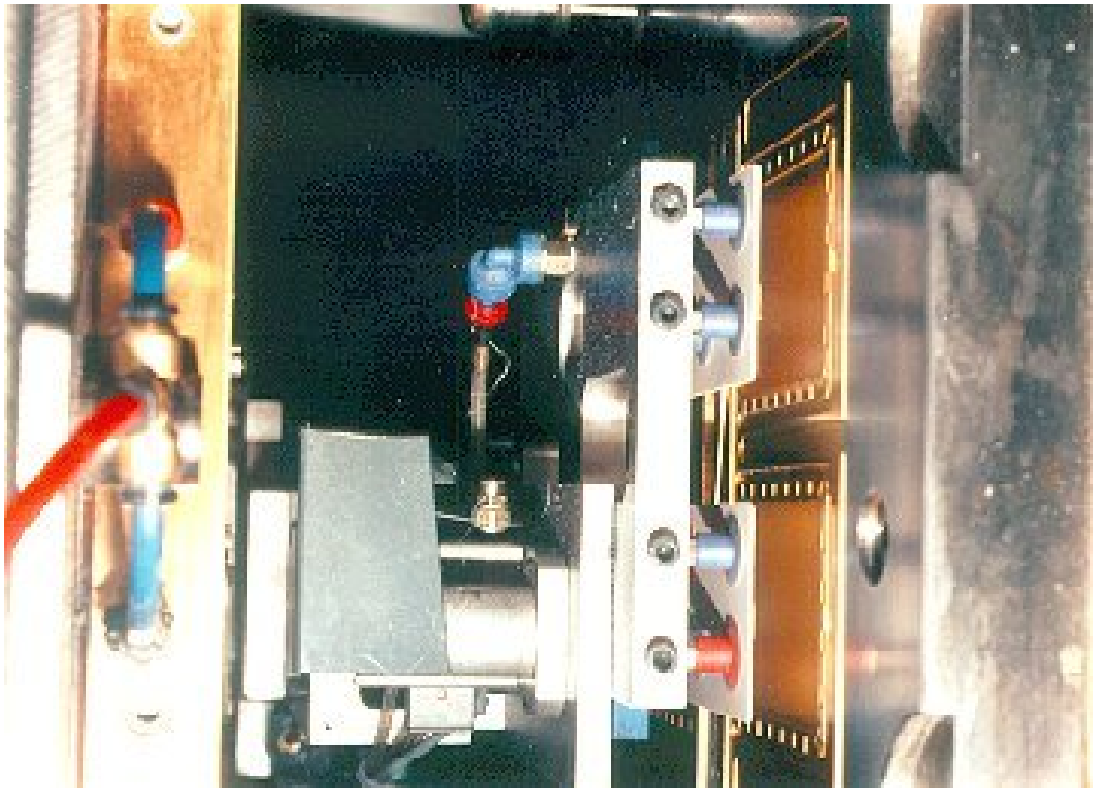
Electrical Testing



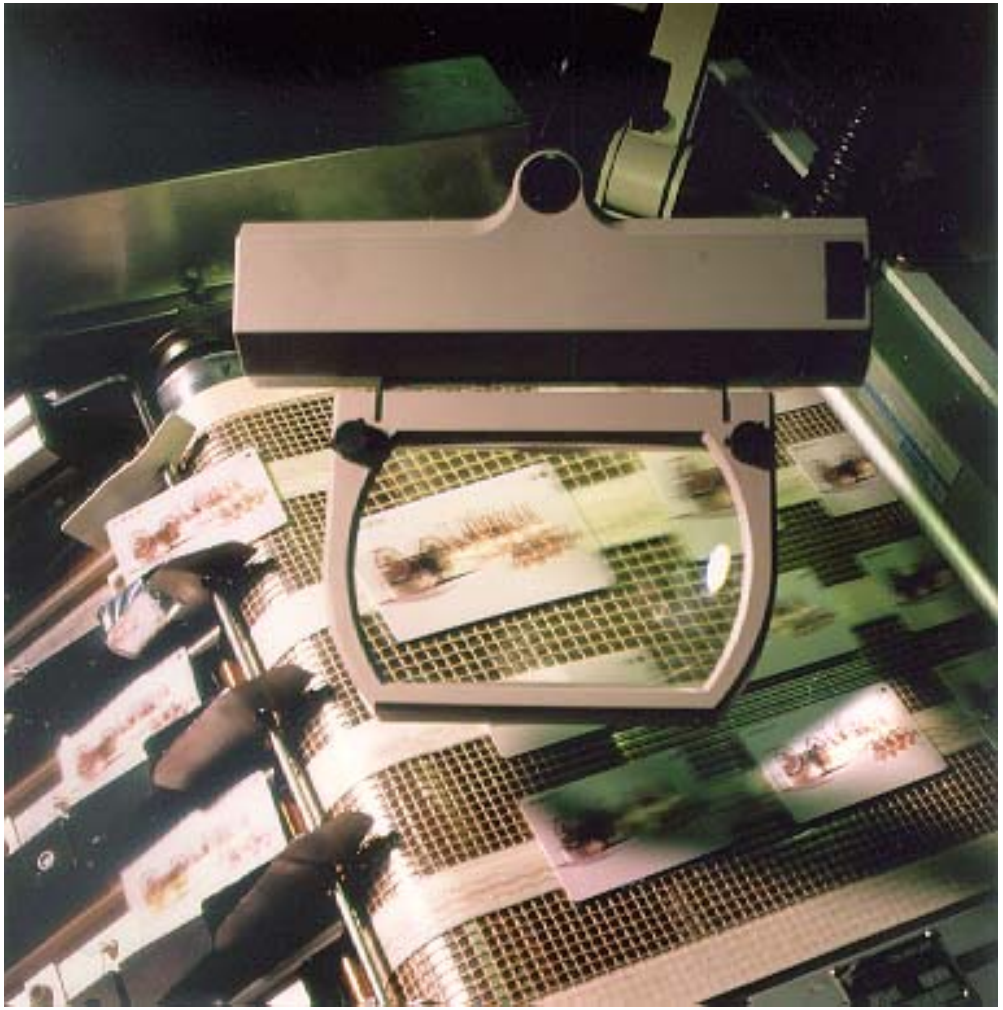
Manufacturing: finished modules



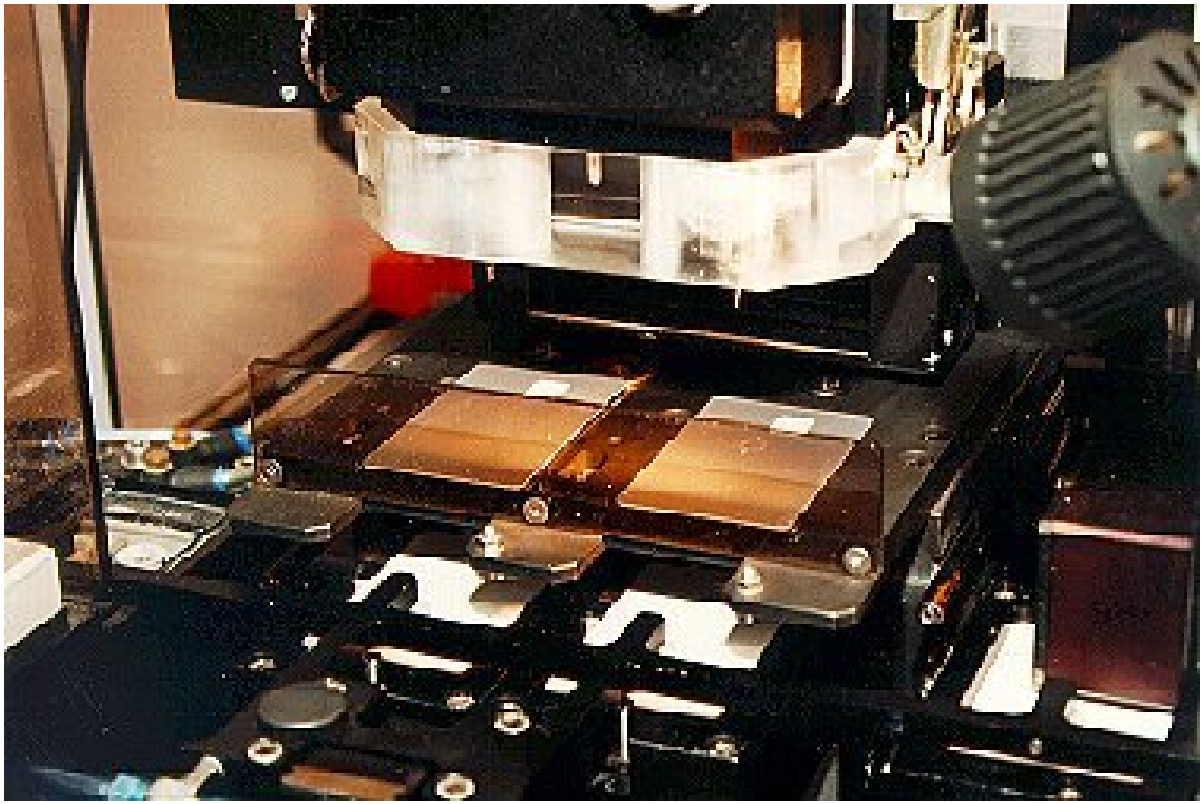
Card Moulding



Offset Printing



Grinding

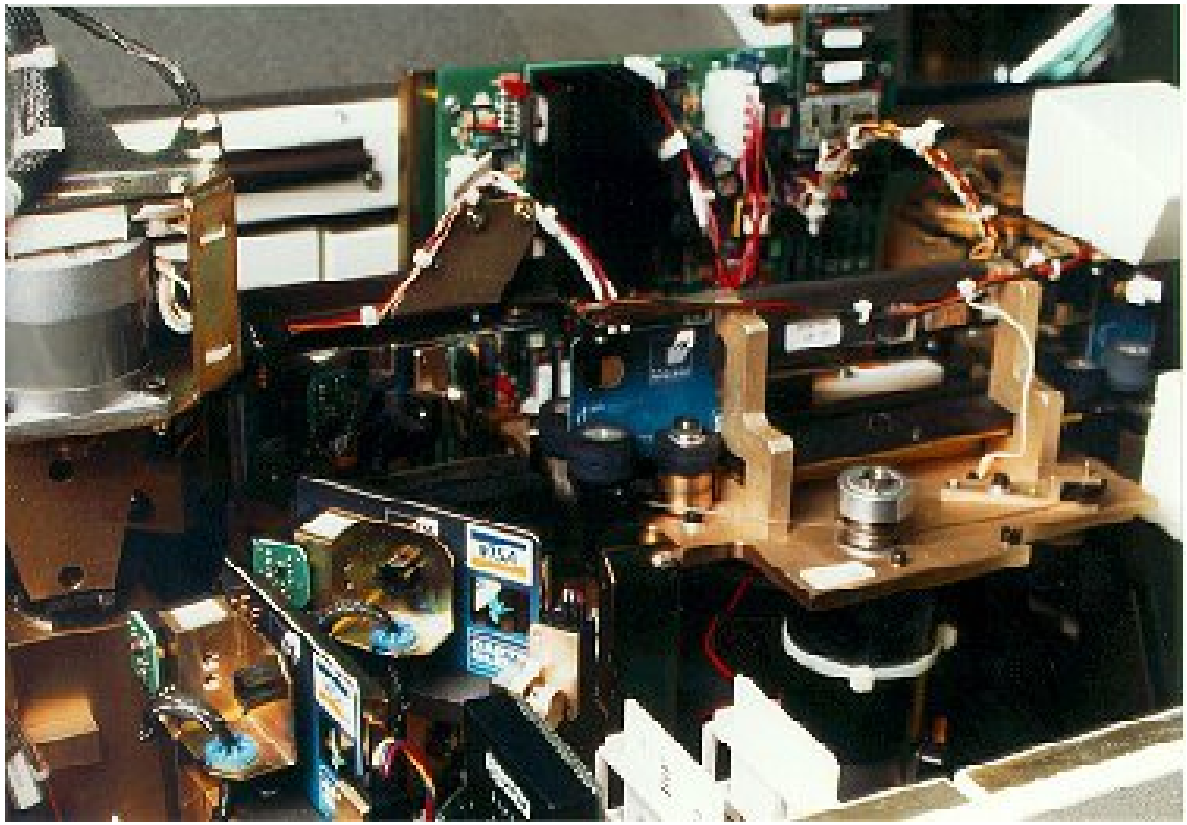


Manufacturing : Embedding & Test

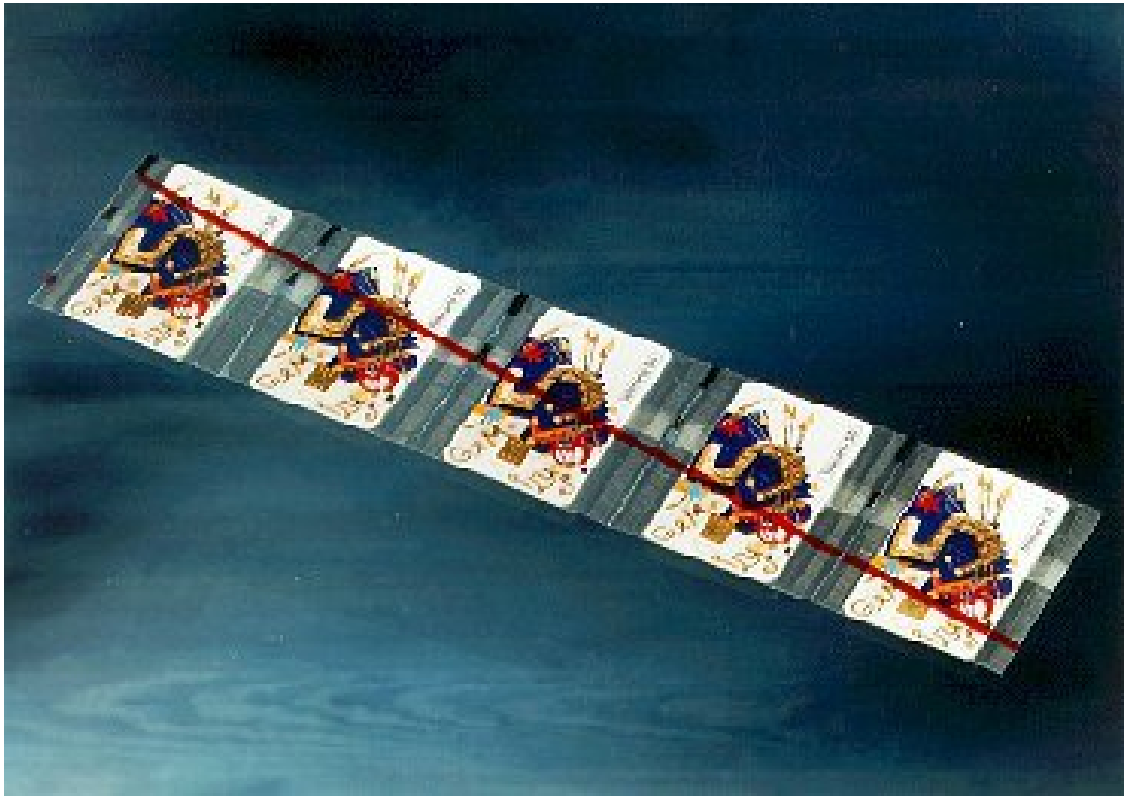
Plug-In

Cutting to plug-in format for mobile phones.

Personalization



Packing



Agenda

- Card Technology
- Standards
- Manufacturing
- Operating system

Introduction

- At the beginning no real OS only stand alone applications
- Mask your own code
  - Pros: small footprint, complete control
  - Cons : development in C and target assembly language, use emulators, Mask lead time 2 months, bug fixes.

Development 7816-4

- Use proprietary cards
  - What you get
    - File system
    - Fixed set of APDU commands: read/write files, cryptographic primitives
  - Pros: off the shelf product, cheaper
  - Cons not extensible, bug fixes.

Fundamentals

- Functions :
  - Transfer data to and from the card
  - Controlling the execution of the commands
  - Managing the files, controlling the access to the files,
  - Management of the life cycle
  - Managing and executing cryptographic algorithms
- No user interface or access to external memory,
- Program written in ROM code (no self modifying techniques allowed),
- No change are possible once the chip is manufactured => quick and dirty programming IS NOT AN OPTION !!!
- Smart Card OS is reliable and robust,
- Design consideration :
  - Persistence…
  - Closely coupled with the hardware

Smart Card Reader exchange

- The card NEVER initiates a communication with the reader

Smart Card introduction

Response to the ATR

Protocol negotiation PTS

Negotiation answer PTS

Command APDU

Answer APDU

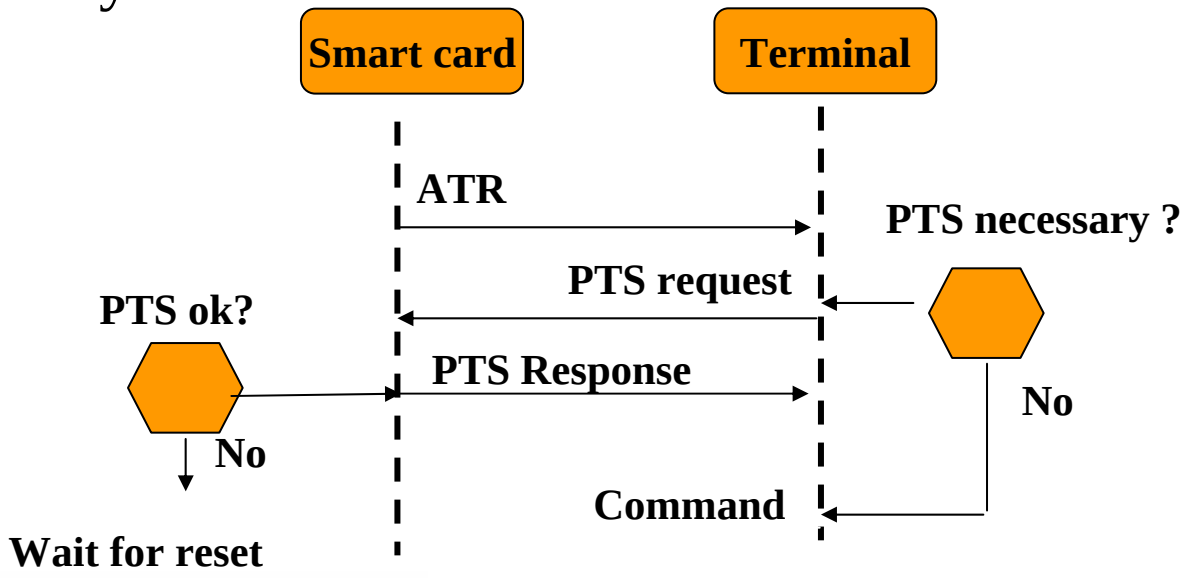
End of session

Answer To Reset

- Starts the smart card program,
- Data elements TS-T0-Tabcd-T1…k-TCK
  - TS: Byte coding convention (3B direct, 3F inverse)
  - T0 : Format characters
  - Ta,b,c,d : Interface characters,
  - T1..T_k: Historical characters to identify OS, version number of the ROM mask, can be omitted.
  - TCK: XOR checksum from T0 to the last byte before TCK.

Protocol Type Selection

- Needed only if the terminal wants to modify parameters,
- If the card agrees, it sends the PTS back to the terminal
- Otherwise the terminal execute a reset (warm => protocol change),
- Only one PTS after the ATR.



Transmission protocols

- T=0 most widely used (1989), T=1 block oriented
- T=14 Japan and Germany

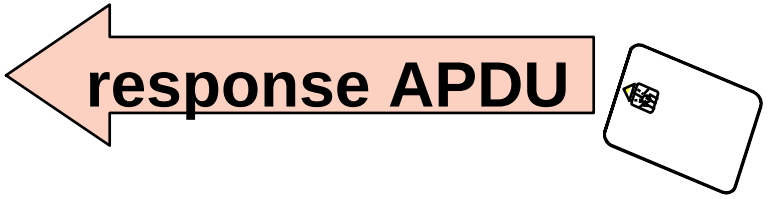
Transmission protocol	Meaning	ISO
T=0	Asynchronous, half duplex, byte oriented	7816-3
T=1	Asynchronous, half duplex, block oriented	7816-3
T=2	Asynchronous, full duplex, block oriented, tbs	10536-4
T=14	National functions	No ISO

Transport protocols

- T=0
  - Byte oriented, Serial transmission (1 start bit, 8 bits data, 1 parity bit, 2 stop bits)
  - Transmission error (parity only) 2 etu mute (“0”)
- T=1
  - Block oriented, Header : NAD, PCB, LEN; data : INF, CRC.
  - NAD 3 bits destination address, 3 bits source address
  - PCB define the kind of block
    - I (#block, more) numbered mod 2, more = 1, another block follow
    - R(#block, error) numbered mod 2, next expected bloc,
    - S specific command (RESYNC, IFS, ABORT, WTX)

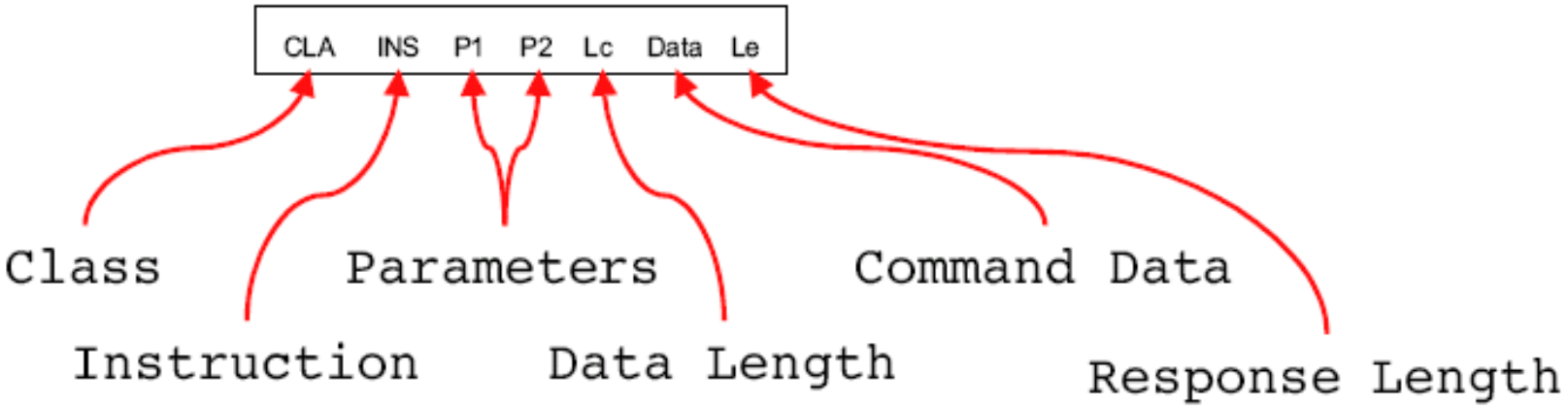
The Application Protocol Data Unit

- Independence of application versus low layers
- An APDU contains either :
  - a command message,
  - a response message.



APDU Syntax

APDU Command



APDU Response

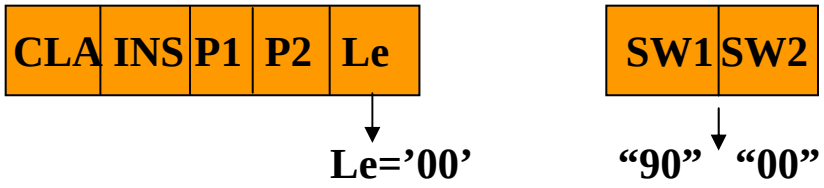
CLA Class byte

b7 to b4	b3	b2	b1	b0	Meaning
			X	X	Logical channel number
	0	0			No secure messaging
	1	0			Secure messaging header not authentic
	1	1			Secure messaging header authentic
‘0’					Structure and coding compliant with 7816-4
‘8’,’9’					User specific codes
‘A’					Structure and code defined in additional document GSM11.11

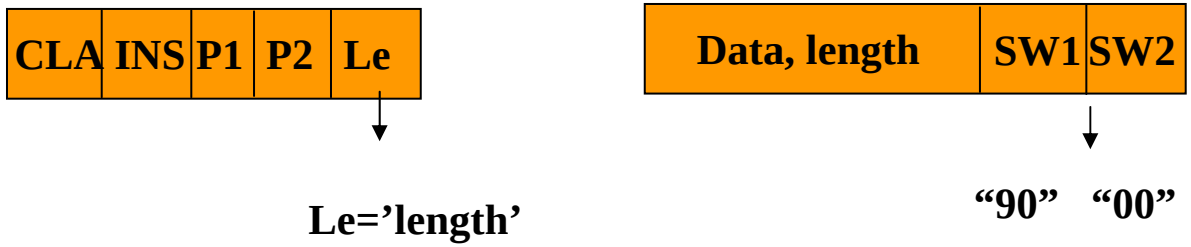
Class	Application
’80’	Electronic purse compliant with EN 1546-3
‘8x’	Credit card compliant with EMV-2
‘A0’	GSM compliant with prETS 300 608/GSM 11.11

APDU Command-Response

- Different configurations
  - Command without data, response without data

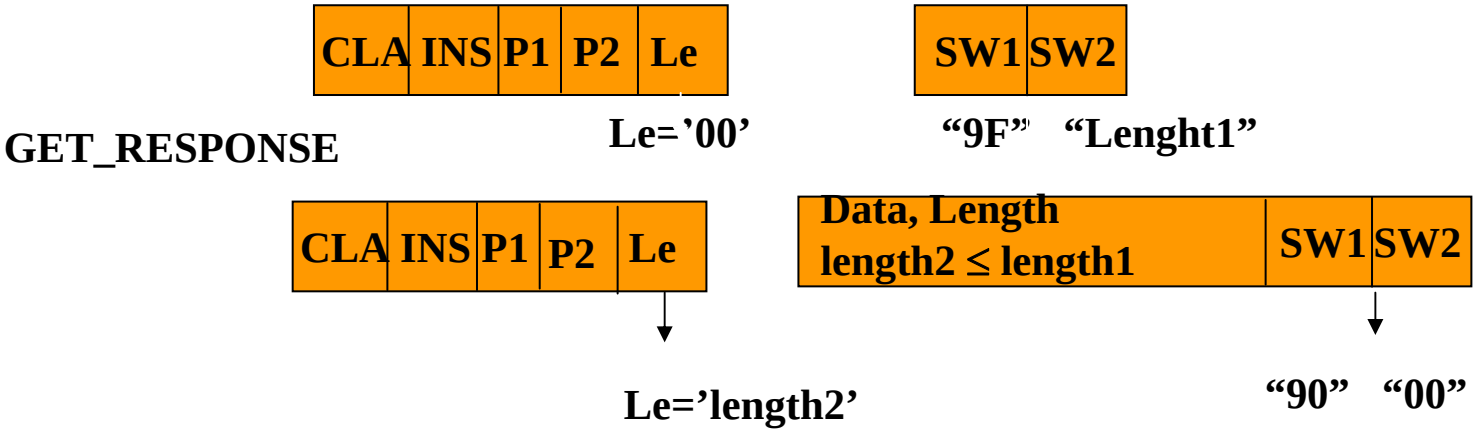


  - Command without data, response with data length known

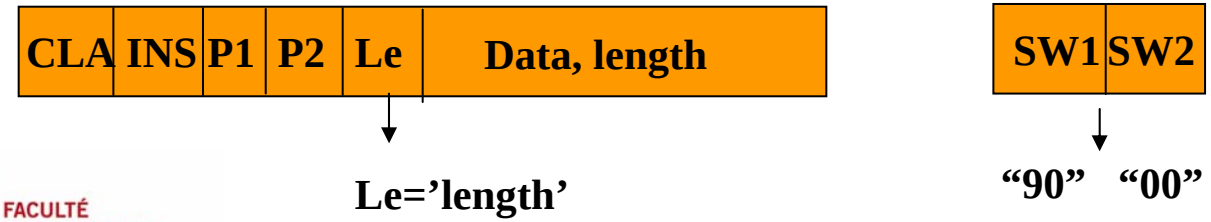


APDU Command-Response

– Command without data, response with data, length unknown

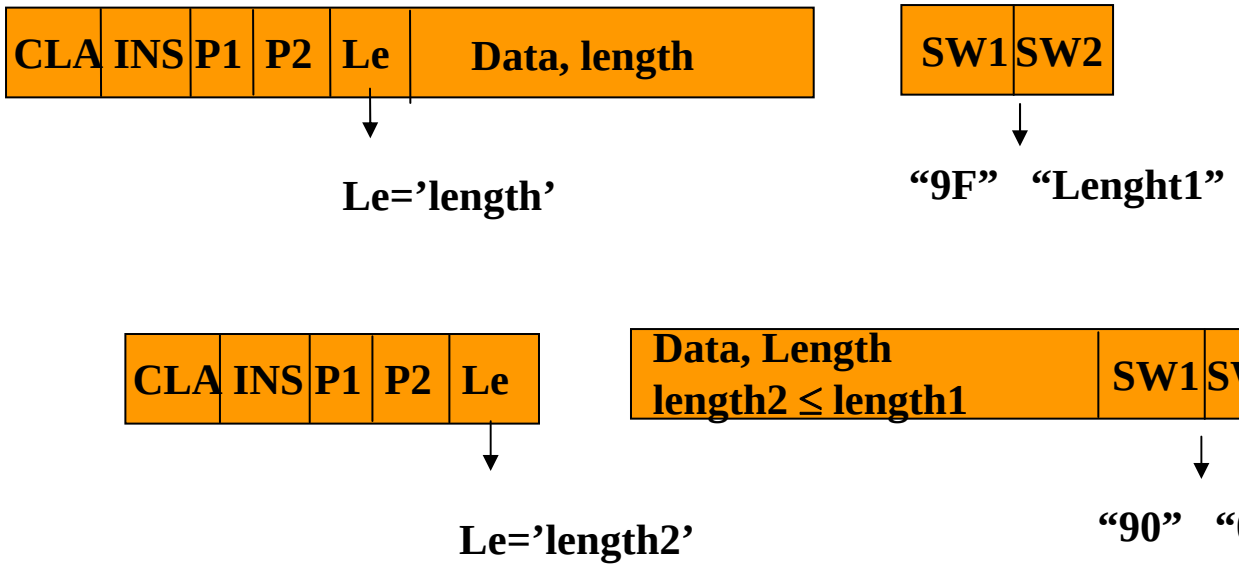


– Command with data, response without data



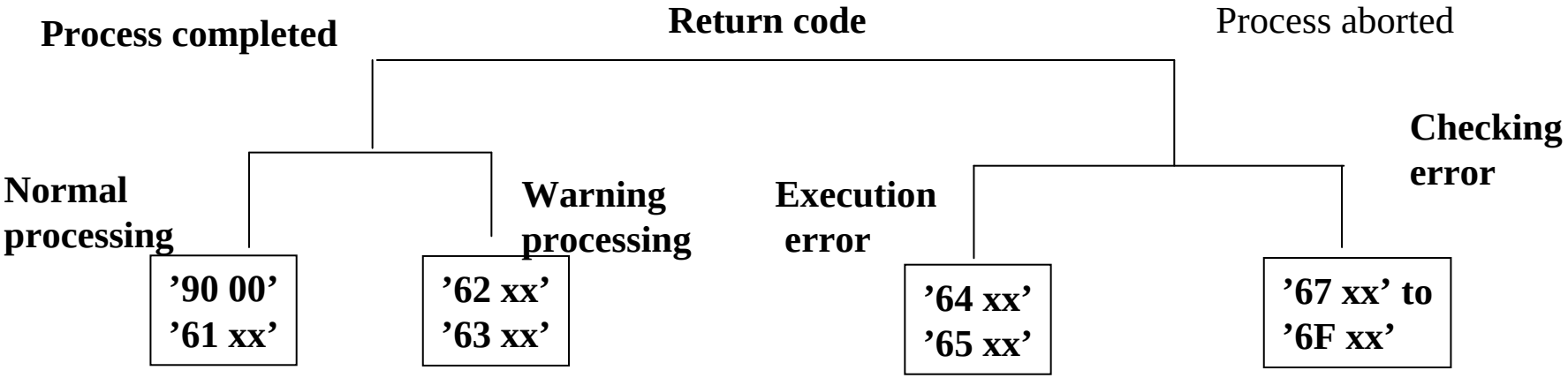
APDU Command-Response

– Command with data, response with data length known or unknown

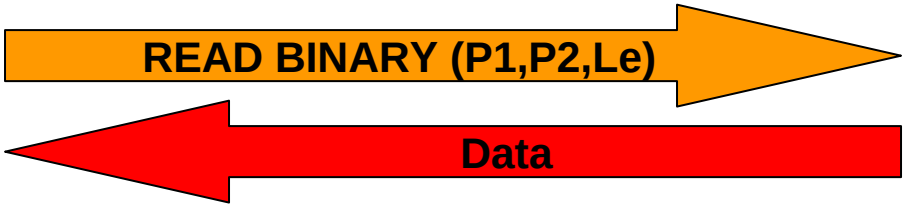


Return Codes

- SW1, SW2 = ’90 00’ command successful, ’63xx’ or ’65xx’ means EEprom has been modified,
- More than 50 different return codes defined by standard,
- Often not respected…



Example



- P1=Offset High,
- P2=Offset low.

Syntax :

CLA	INS	P1	P2	Le
A0	B0	xx	yy	Le

P1, P2 : specify the data to be retrieved
Le : length of data to retrieve

Soft Masks

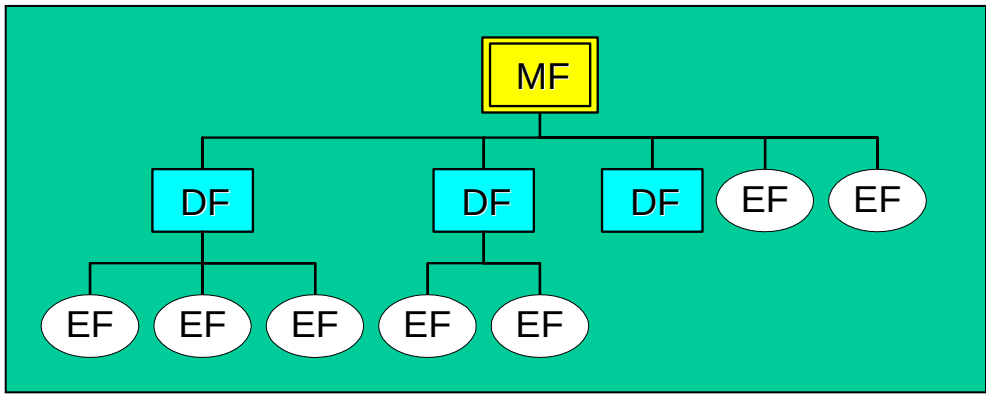
- It is an extension of the hard mask
- Often written in C, compiled and linked to the libraries,
- Can be download in Eeprom if the card is not blocked,
- Need ?
  - Bug fixes,
  - Adding new functionalities,
  - A customer needs a rewriting of a command…

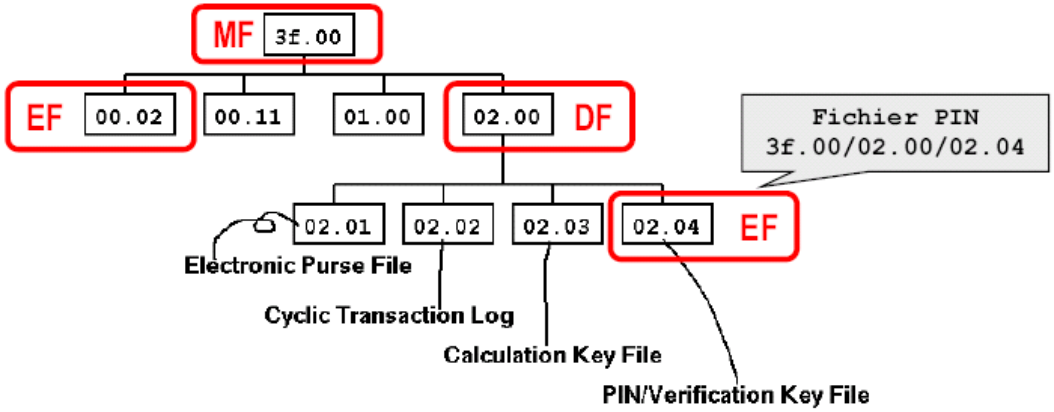
7816-4 based OS

- Data are stored in files structures in Eeprom,
- A file must be selected before any action,
- Made of a header and a body,
  - The header stores the access conditions and the structure of the file.
  - For security reasons header and body are stored on different eeprom pages

File Organization

- MF - *Master File* :
  - root of the file structure,
  - can be seen as a main directory.
- DF - *Dedicated File* :
  - contains other files,
  - can be seen as a directory.
  - each DF will behave like independent card.
- EF - Elementary File :
  - contains data.
  - working or internal EF





File name

- File IDentification (FID) 2 bytes
  - All EFs within a single directory must have different FID
  - Nested directories (DF) may not have the same FID
  - AN EF within a directory (MF or DF) may not have the same FID as the next lower or higher directory.
- For historical reasons the MF is “3F 00” and GSM DF have a value of “7F” for the first byte.
- In addition to its FID a DF has a DF name often used together with AID defined in 7816-5.
- AID is made of :
  - RID registered aid mandatory 5 bytes
  - PIX proprietary application identifier extension (0..11 bytes).

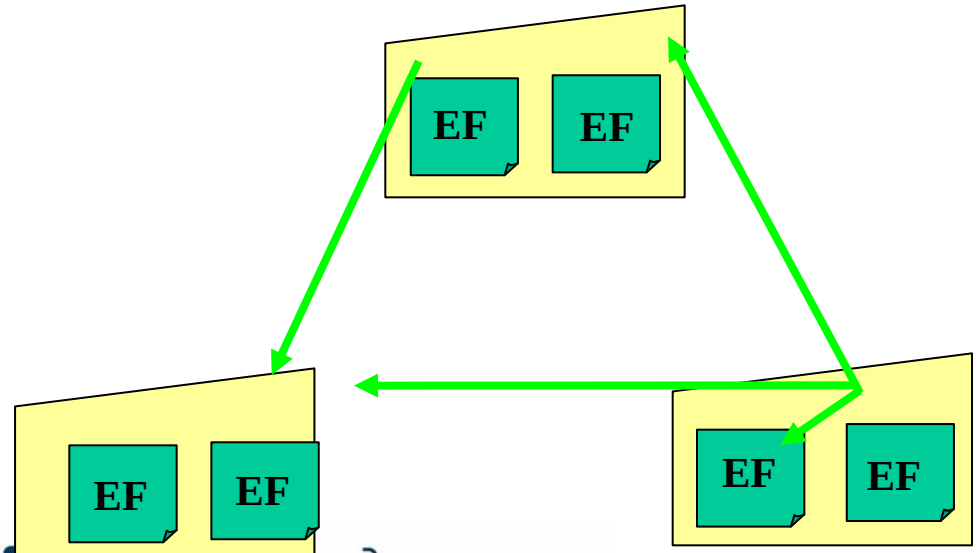
AID

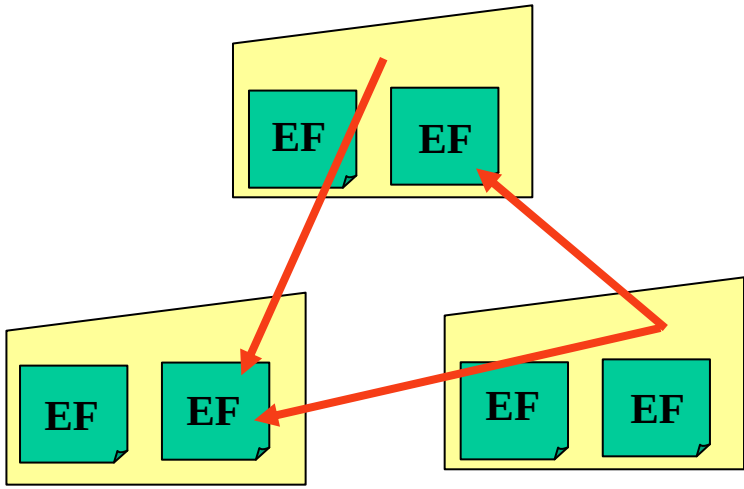
- RID is assigned by a national or international authority
- RID is assigned only one is used to identify applications

RID			Meaning
D1	D2-D4	D5-D10	
X	-	-	Registration category (A international, D national)
	X		Country code according ISO 3166
		X	Application vendor number assigned by national or international body

File selection

- Only one file is selected at a time,
- The MF is always selectable and is implicitly selected after a reset
- FID are not unique => restriction in selection





EF File structures

- Four data structure
  - Binary (transparent) files (data accessible through an address)
  - Sequential record fixed size or variable size
  - Cyclic buffer
- Transparent file
  - No internal structure
  - Accessed for reading or writing in bytes or blocks with an offset value
  - Often used with a small amount of data,
  - Commands READ BINARY, WRITE BINARY and UPDATE BINARY

EF File structures

- Linear fixed file structure
  - Linking fixed length records,
  - The smallest unit is a record,
  - Commands: READ RECORD, WRITE RECORD and UPDATE RECORD
  - From 1..254
- Linear variable file structure
  - Same commands,
  - Need additional info concerning the length of each records
  - Optimise the memory usage.

EF File structures

- Cyclic file structure
  - Based on the linear fixed file structure,
  - The EF contains a pointer on the last written record numbered 1, the previous 2, etc…
  - Can be accessed by addressing the first, the last, the next or previous record.

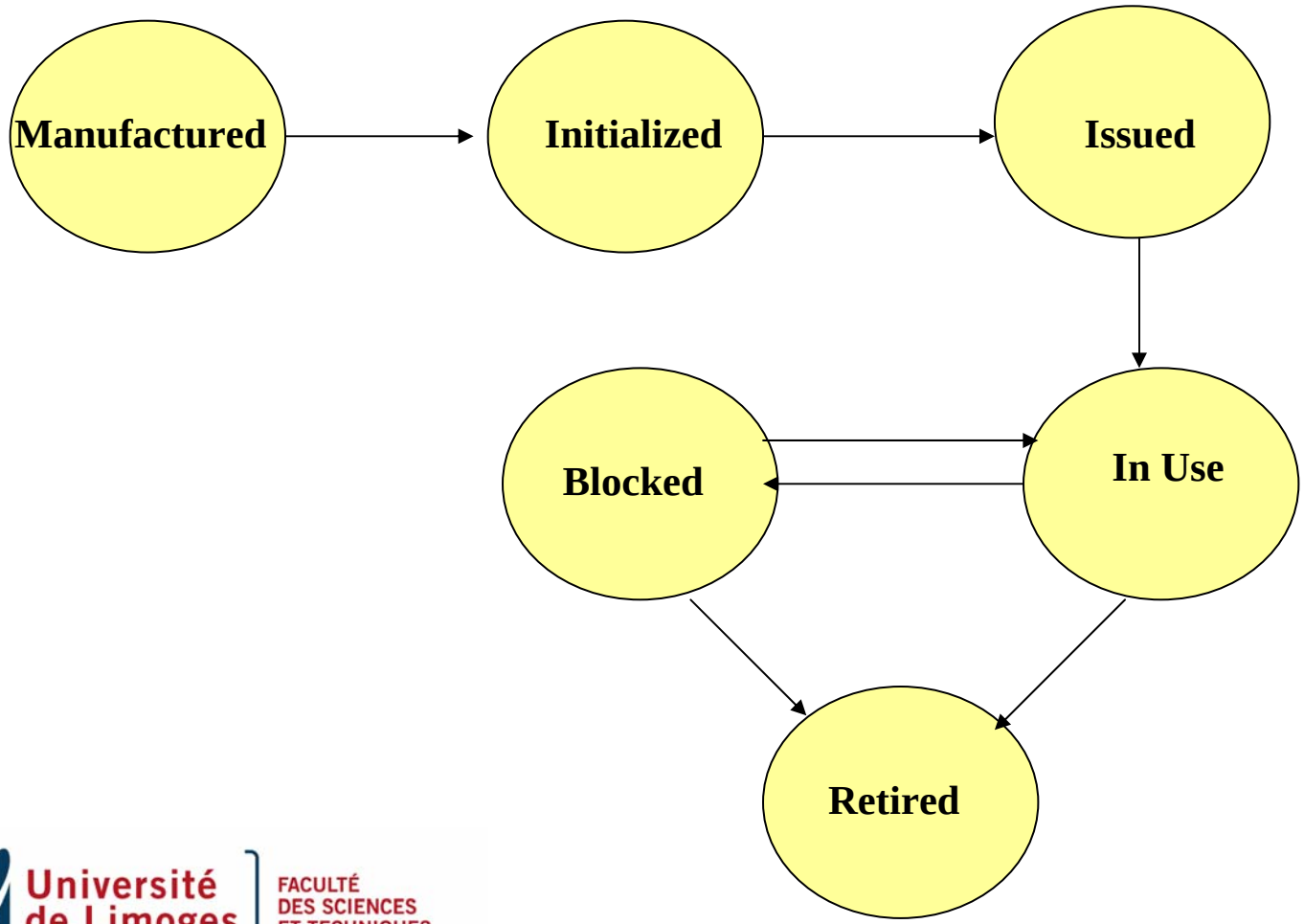
File Access Conditions

- Security is based on file access privileges,
- Access information coded in the header, defined when a file is created and usually cannot be changed later.
- For MF and DF
  - no information stored for data access (read and write)
  - But for creation and deletion of files.
- The PINs are stored in separate elementary files, EF_{CHV1} and EF_{CHV2} for example

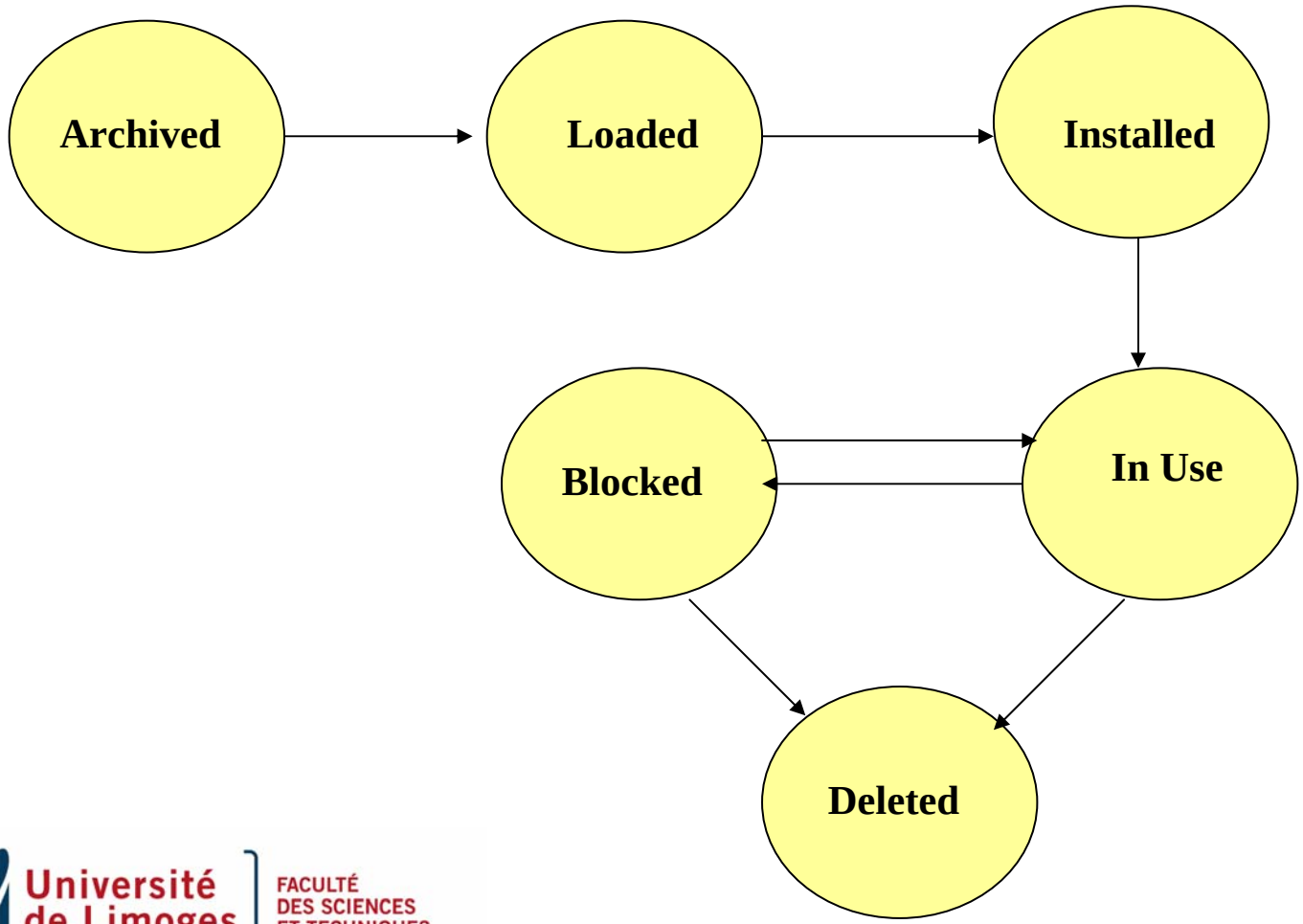
File attributes

- Five kinds of EF files
  - **Always (ALW):** Access of the file can be performed without any restriction.
  - **Card holder verification 1 (CHV1):** Access can only be possible when a valid CHV1 value is presented
  - **Card holder verification 2 (CHV2):** Access can only be possible when a valid CHV2 value is presented
  - **Administrative (ADM):** Allocation of these levels and the respective requirements for their fulfilment are the responsibility of the appropriate administrative authority
  - **Never (NEV):** Access of the file is forbidden

Smart Card live cycle



Application live cycle



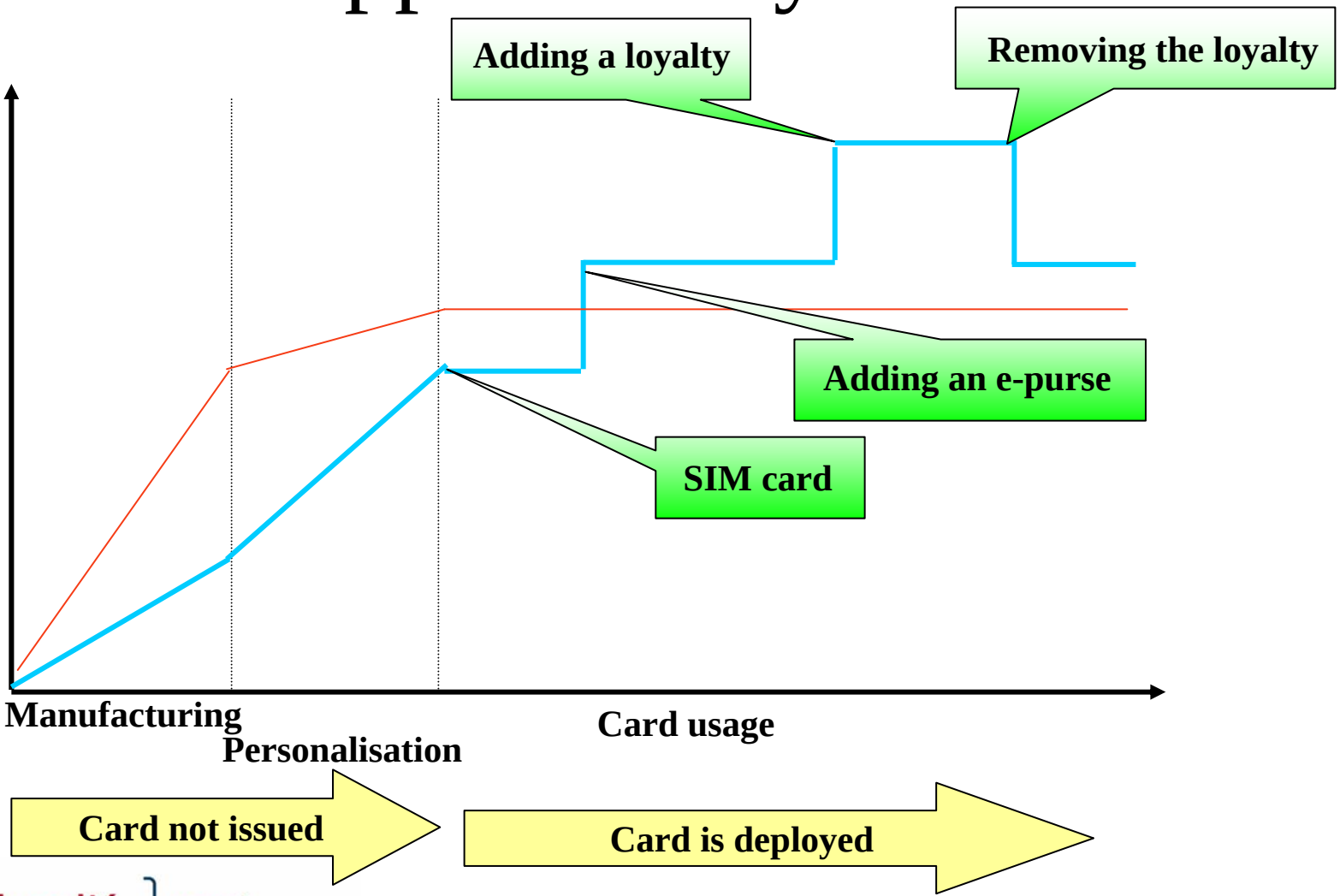
OS based on 7816-4

- Pro
  - Cheap, easy to use
  - Possible to insert new commands
- Cons
  - Unable to execute code
  - Frozen after personalisation phase,
  - Data oriented.

Time to market

- Time between decision and product launch
- Could take as one year if mask need to be redevelop
- Not really adapt to current market :
  - Mobile phone is a highly competitive market,
  - Interoperability is needed,
  - Development cost are too important,
- Smart card manufacturers developed generic smart card: open cards
  - With real operating system
  - Able to download application during their life cycle

Applet life cycle



Open cards

- From a developer point of view:
  - Until now, writing an application required a specific knowledge,
  - No need of smart card specialists,
  - Solution : use general purpose programming language (C, Java, Visual Basic…)
  - Much more easier to integrate applications,
  - More tools to test applications,
- From an end-user point of view
  - Several application on a single card,
  - Possibility to load/unload application when needed.

Smartcards of the present days

- Java Card
  - Embedded virtual machine,
  - Open standard (Java Card 2.2),
  - Wide support of the industry
    - IBM, Visa,…
  - Reduction of development time.

Applet development

- Write code in Java
- Compile it
- Debug it (simulator)
- Verify and Convert it (specific byte code)
- Load it
  - Personalization center
  - Point of sale
  - Over the Internet

MULTOS

- Based on the MEL (Multos Executable Language) interpreter.
  - Operating system and memory firewalls
  - Virtual Machine layer to provide abstraction
  - Application Programming Interface (API)
  - Application management including secure loading and deleting methods.
- See http://www.multos.com

Basic Card

- Based on the Basic language
  - DOS like file system,
  - P-Code byte code interpreter
- PRO
  - Fit well for a small amount of cards
- CONS
  - Not supported by major smart card manufacturers
  - Proprietary code (http://www.zeitcontrol.de)

Next step ?

- Internet Card (2004): portable web server,
- Access to the secret stored in the card through your browser,
- Use the USB port, TCP/IP protocol, data security through SSL, multi-threading, full garbage collection… *id est* JC3.0

Any question ?