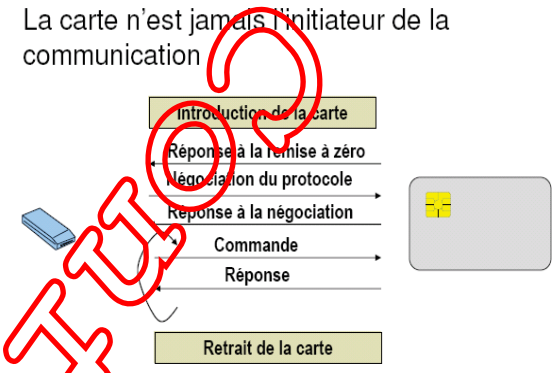
ISO 7816-3

- Cette partie normalise :
  - les protocoles de transmission (TPDU : Transmission Protocol Data Unit) :
    - T=0, protocole orienté octet
    - T=1, protocole orienté paquet
    - T=14, réservé pour les protocoles propriétaires
  - la sélection du type de protocole (PTS : Protocol Type Selection)
  - la négociation des paramètres du protocole (PPS : Protocol Parameter Selection)

  - la réponse au reset (ATR : Answer To Reset) qui correspond à la mise en route du prog ROM de la carte
    - Min. 2 et Max. 33 caractères et 5 champs
    - Permet de fixer :
      - Les conventions de codage des octets
      - Le temps de transmission d 'un bit
      - La valeur de la tension de programmation
      - Le protocole de communication
      - Un historique qui s'affichera à la mise sous tension de la carte (ex: version de l 'OS)



  - les caractéristiques électriques comme :
    - la fréquence d'horloge (entre 1MHz et 5MHz)
    - la vitesse des communications (jusqu'à 115200 bauds)

Exemples d'ATR

- Carte Santé Vitale

ATR = 3F 65 25 00 2C 09 69 90 00

- Carte Bancaire CB

ATR = 3F 65 25 08 36 04 6C 90 00

- Carte Verte Monéo

ATR= 3B E6 00 FF 81 31 42 45 19 16 01 01 27 B1 37

- Carte Cinema (perimé)

ATR = 3B 23 00 35 13 96

- Carte GSM Itineris

ATR=3F 2F 00 30 AF 59 02 01 02 80 00 17 0A 0E 83 1E 9F 16

- GXP211_PKIS

ATR = 3F 6D 00 00 80 31 80 65 B0 05 01 02 5E 83 00 90 00

- GemSafe

ATR= 3B A7 00 40 18 80 65 A2 08 01 01 52

- Schlumberger Palmera

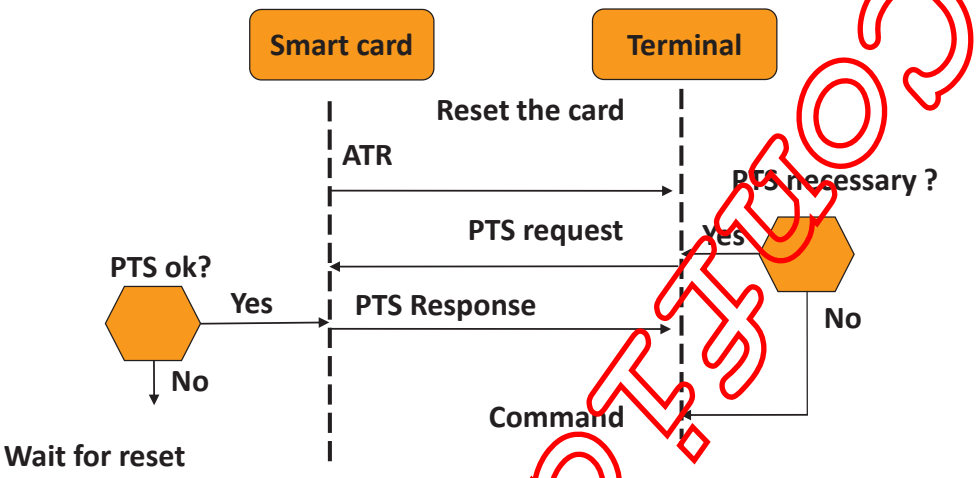
ATR= 3B 65 00 00 9C 02 02 06 01

- Cyberflex Access e-gate 32K

ATR=3B 75 94 00 00 62 02 02 00 80

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.



Comportements de la carte et du lecteur lors d'un Reset

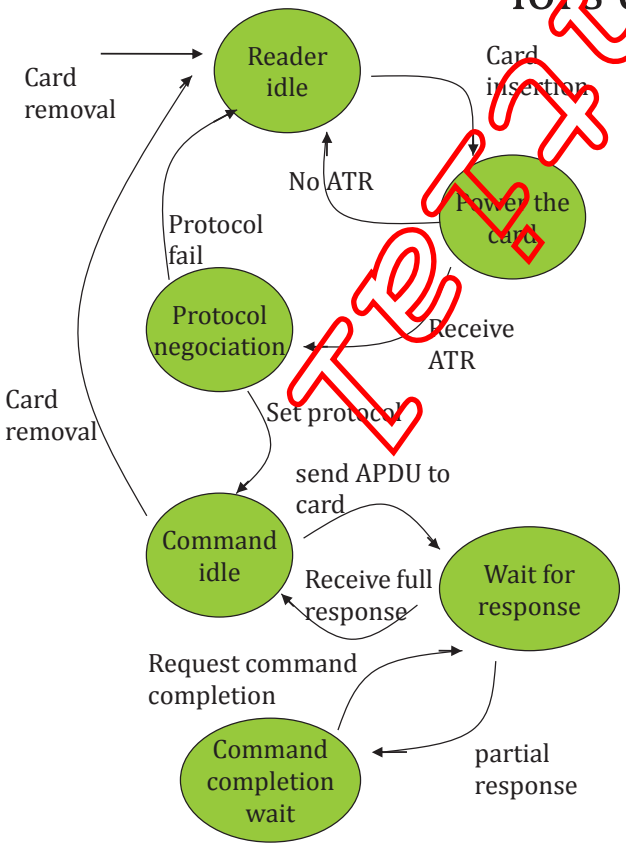


Diagramme d'état du lecteur

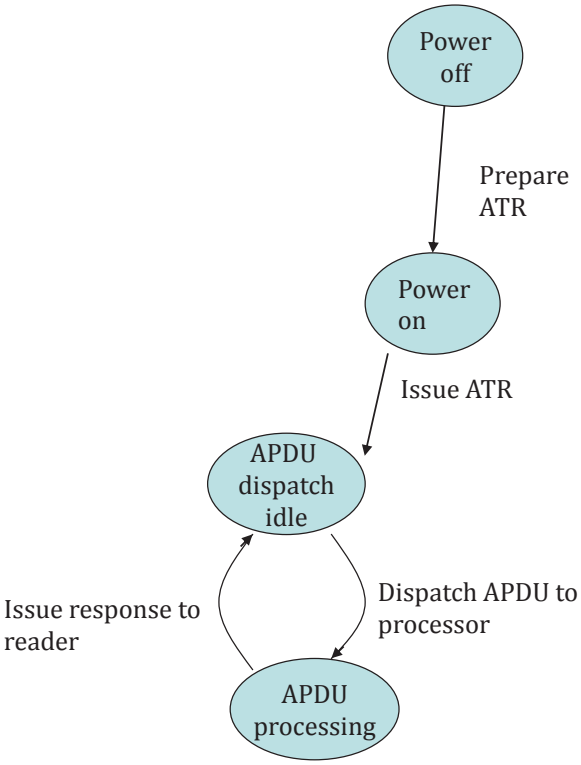


Diagramme d'état de la carte

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.

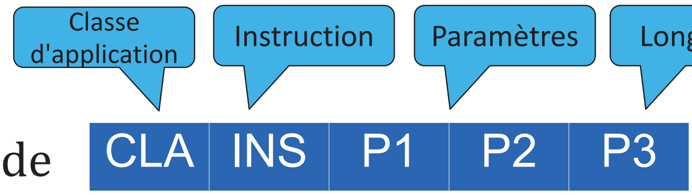
NAD 3 bits destination address, 3 bits source address

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)

T=0 Structure d'une commande/réponse

- Commande



CLA est normalisé (par exemple FF est utilisé pour le PPS)

INS ne doit pas être 9x ou 6x et **devait** être paire avant l'édition 3 de l'ISO7816-3 (2006)

- Réponse

Les valeurs sont normalisées

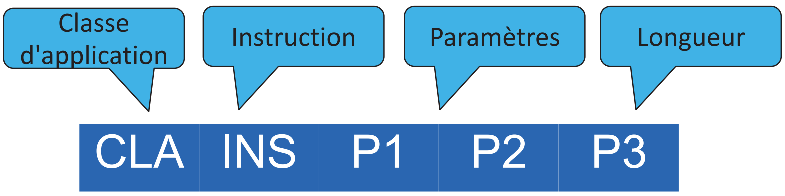
- 90 00 : succès
- 6E xx : classe inconnue
- 6D xx : instruction inconnue
- ...

Protocole T=0 (Transmission semi-duplex de caractères asynchrones)

Commande entrante (envoie des données à la carte)

Commande sortante (récupère des données de la carte)

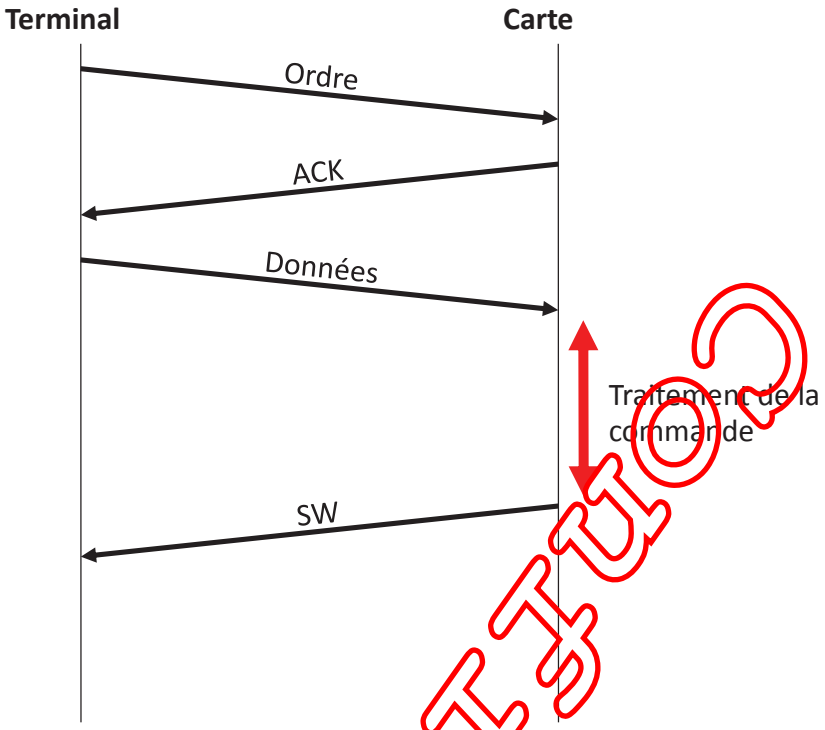
Structure de l 'ordre



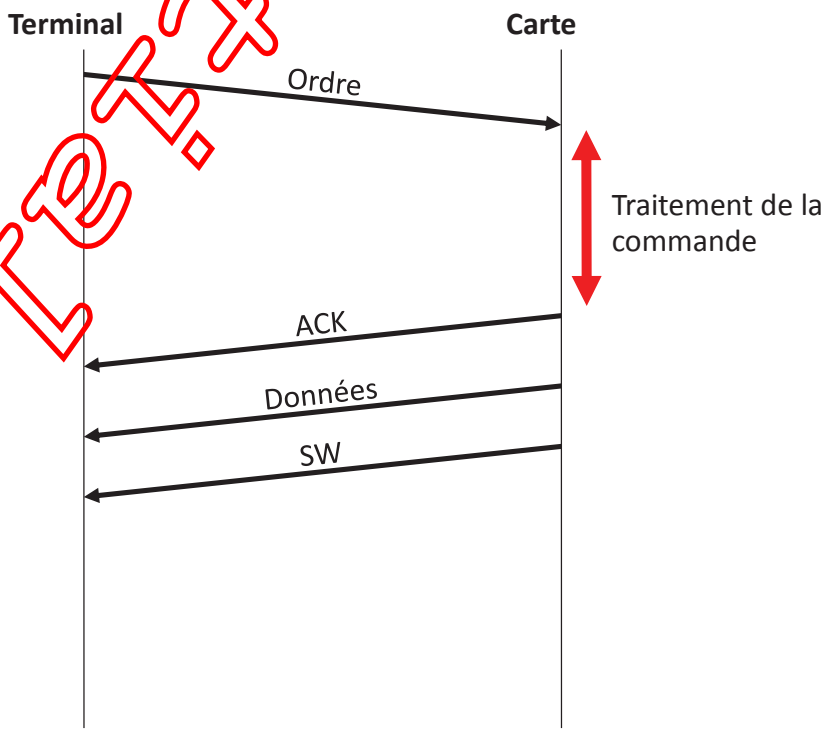
Octets de procédure

Byte	Value	Action on data transfer	Then reception of
NULL	'60'	No action	A procedure byte
SW1	'6X' (≠'60'), '9X'	No action	A SW2 byte
ACK	INS	All remaining data bytes (if any bytes remain)	A procedure byte
	INS ⊕ 'FF'	The next data byte (if it exists)	A procedure byte
The first two editions of ISO/IEC 7816-3 specified the use of two values of ACK (namely, the exclusive-or of the value of INS with '01' and 'FE') to control the deprecated use of contact C6 (see 5.1.1). These two values are deprecated.			

T=0 : Commande entrante



T=0 : Commande sortante



ISO 7816-4

- ISO 7816-4 vise à assurer une interopérabilité.

But : indépendance des applications par rapport aux couches physique et liaison

- Il spécifie :
  - le contenu des messages entre la carte et le lecteur
    - les commandes
    - les réponses
  - les structures des fichiers et de données :
    - l'accès à ces données
    - l'architecture de sécurité
    - la sécurisation des communications
- Le protocole APDU
  - Protocole de niveau application.
    - La commande APDU (C-APDU) : émise par le CAD vers la carte

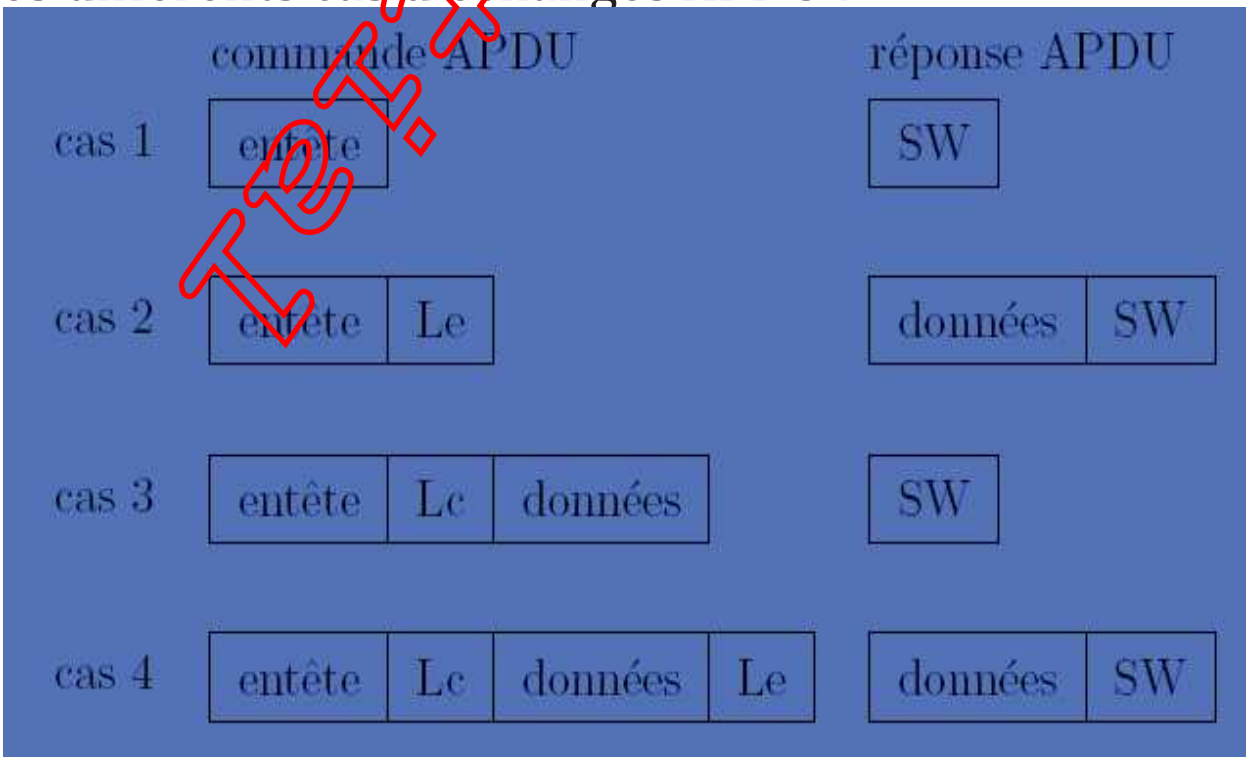
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc *(1-3 octets)*	Champs de Données *(Nc octets)*	Le *(1-3 octets)*

    - La réponse APDU (R-APDU) : transite de la carte vers le CAD

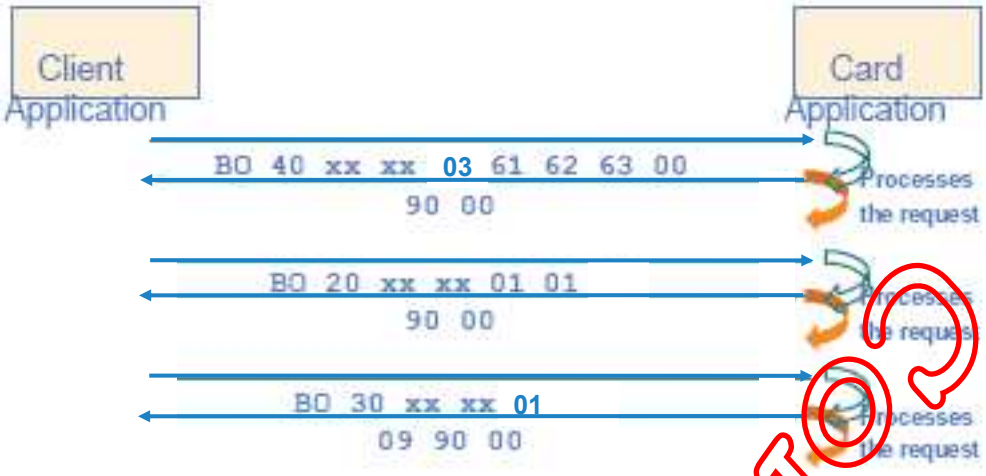
Corps optionnel	État obligatoire	
Champs de Données *(Nr octets <= Ne)*	SW1	SW2

ISO 7816-4

- Les différents cas d'échanges APDU :



Exemple de communication APDU



- La transmission d'APDU en T=0 ou T=1 est détaillée dans les annexes de l'ISO7816-3

Complexité

- Afin de pouvoir rendre transparent la communication APDU vis à vis des différents protocoles sous-jacents, on verra en Java Card qu'il sera nécessaire d'appeler les méthodes de communication dans un certain ordre.
- En effet, T=0 par exemple ne permet pas d'avoir des commandes entrantes **et** sortantes ...

CLA Class byte

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	x	-	-	-	-	**Command chaining control**
0	0	0	0	-	-	-		-- The command is the last or only command of a chain
0	0	0	1	-	-	-	-	-- The command is not the last command of a chain
0	0	0	-	x	x	-	-	**Secure messaging indication**
0	0	0	-	0	0	-	-	-- No SM or no indication
0	0	0	-	0	1	-	-	-- Proprietary SM format
0	0	0	-	1	0	-	-	-- SM with command header not processed
0	0	0	-	1	1	-	-	-- SM with command header authenticated
0	0	0	-	-	-	x	x	**Logical channel number from zero to three**
0	0	1	x	x	x	x	x	Reserved for future use by ISO/IEC JTC 1/SC 17
0	1	x	-	-	-	-	-	**Secure messaging indication**
0	1	0	-	-	-	-	-	-- No SM or no indication
0	1	1	-	-	-	-	-	-- SM with command header not processed
0	1	-	x	-	-	-	-	**Command chaining control**
0	1	-	0	-	-	-	-	-- The command is the last or only command of a chain
0	1	-	1	-	-	-	-	-- The command is not the last command of a chain
0	1	-	-	x	x	x	x	**Logical channel number from four to nineteen**
1	x	x	x	x	x	x	x	Proprietary class. The application context defines the other bits.
1	1	1	1	1	1	1	1	Invalid (used by PPS in ISO 7816-3)

interindustry class

CLA Class byte

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

INS Instruction byte

Table 4.1 — Commands in the alphabetic order

Command name	INS	See
ACTIVATE FILE	'44'	Part 9
APPEND RECORD	'E2'	7.3.7
CHANGE REFERENCE DATA	'24'	7.5.7
CREATE FILE	'E0'	Part 9
DEACTIVATE FILE	'04'	Part 9
DELETE FILE	'E4'	Part 9
DISABLE VERIFICATION REQUIREMENT	'26'	7.5.9
ENABLE VERIFICATION REQUIREMENT	'28'	7.5.8
ENVELOPE	'C2', 'C3'	7.6.2
ERASE BINARY	'0E', '0F'	7.2.7
ERASE RECORD (S)	'0C'	7.3.8
EXTERNAL (/ MUTUAL) AUTHENTICATE	'82'	7.5.4
GENERAL AUTHENTICATE	'86', '87'	7.5.5
GENERATE ASYMMETRIC KEY PAIR	'46'	Part 8
GET CHALLENGE	'84'	7.5.3
GET DATA	'CA', 'CB'	7.4.2
GET RESPONSE	'C0'	7.6.1
INTERNAL AUTHENTICATE	'88'	7.5.2
MANAGE CHANNEL	'70'	7.1.2
MANAGE SECURITY ENVIRONMENT	'22'	7.5.11
PERFORM SCQL OPERATION	'10'	Part 7
PERFORM SECURITY OPERATION	'2A'	Part 8
PERFORM TRANSACTION OPERATION	'12'	Part 7
PERFORM USER OPERATION	'14'	Part 7
PUT DATA	'DA', 'DB'	7.4.3
READ BINARY	'B0', 'B1'	7.2.3
READ RECORD (S)	'B2', 'B3'	7.3.3
RESET RETRY COUNTER	'2C'	7.5.10
SEARCH BINARY	'A0', 'A1'	7.2.6
SEARCH RECORD	'A2'	7.3.7
SELECT	'A4'	7.1.1
TERMINATE CARD USAGE	'FE'	Part 9
TERMINATE DF	'E6'	Part 9
TERMINATE EF	'E8'	Part 9
UPDATE BINARY	'D6', 'D7'	7.2.5
UPDATE RECORD	'DC', 'DD'	7.3.5
VERIFY	'20', '21'	7.5.6
WRITE BINARY	'D0', 'D1'	7.2.4
WRITE RECORD	'D2'	7.3.4

— In the interindustry class, any valid INS code not defined in ISO/IEC 7816 is reserved for future use by ISO/IEC JTC 1/SC 17.

Table 4.2 — Commands in the numeric order

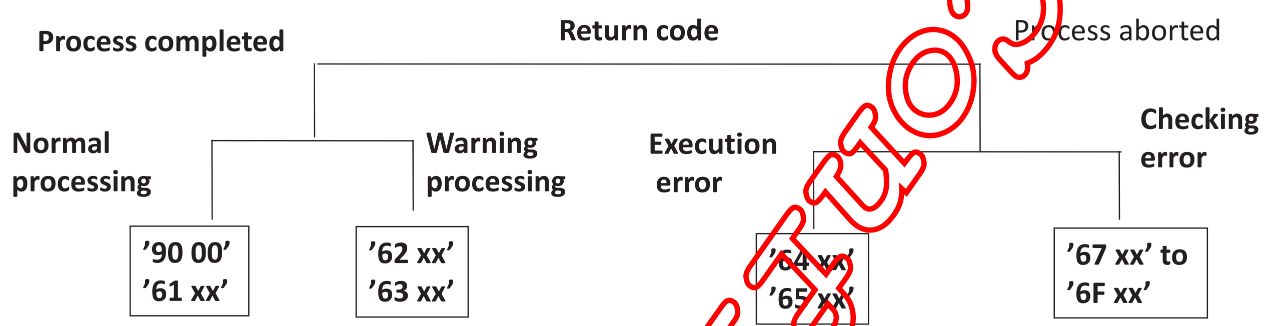
INS	Command name	See
'04'	DEACTIVATE FILE	Part 9
'0C'	ERASE RECORD (S)	7.3.8
'0E', '0F'	ERASE BINARY	7.2.7
'10'	PERFORM SCQL OPERATION	Part 7
'12'	PERFORM TRANSACTION OPERATION	Part 7
'14'	PERFORM USER OPERATION	Part 7
'20', '21'	VERIFY	7.5.6
'22'	MANAGE SECURITY ENVIRONMENT	7.5.11
'24'	CHANGE REFERENCE DATA	7.5.7
'26'	DISABLE VERIFICATION REQUIREMENT	7.5.9
'28'	ENABLE VERIFICATION REQUIREMENT	7.5.8
'2A'	PERFORM SECURITY OPERATION	Part 8
'2C'	RESET RETRY COUNTER	7.5.10
'44'	ACTIVATE FILE	Part 9
'46'	GENERATE ASYMMETRIC KEY PAIR	Part 8
'70'	MANAGE CHANNEL	7.1.2
'82'	EXTERNAL (/ MUTUAL) AUTHENTICATE	7.5.4
'84'	GET CHALLENGE	7.5.3
'86', '87'	GENERAL AUTHENTICATE	7.5.5
'88'	INTERNAL AUTHENTICATE	7.5.2
'A0', 'A1'	SEARCH BINARY	7.2.6
'A2'	SEARCH RECORD	7.3.7
'A4'	SELECT	7.1.1
'B0', 'B1'	READ BINARY	7.2.3
'B2', 'B3'	READ RECORD (S)	7.3.3
'C0'	GET RESPONSE	7.6.1
'C2', 'C3'	ENVELOPE	7.6.2
'CA', 'CB'	GET DATA	7.4.2
'D0', 'D1'	WRITE BINARY	7.2.6
'D2'	WRITE RECORD	7.3.4
'D6', 'D7'	UPDATE BINARY	7.2.5
'DA', 'DB'	PUT DATA	7.4.3
'DC', 'DD'	UPDATE RECORD	7.3.5
'E0'	CREATE FILE	Part 9
'E2'	APPEND RECORD	7.3.6
'E4'	DELETE FILE	Part 9
'E6'	TERMINATE DF	Part 9
'E8'	TERMINATE EF	Part 9
'FE'	TERMINATE CARD USAGE	Part 9

Exemples de commandes APDU

Champ de la commande APDU	Valeurs
CLA	80 = cartes de crédit françaises, cartes vitales françaises, A0 = cartes SIM (téléphonie) 00 = cartes Monéo (porte-monnaie en France), Mastercard, Visa
INS	20 = vérification du PIN, B0 = Lecture B2 = Lecture de record D0 = Écriture DC = Écriture de record A4 = Sélection du répertoire (directory) C0 = Demander une réponse (get response)
P1, P2	paramètres contenant des adresses à lire
LEN	longueur prévue pour la réponse ou bien longueur de l'argument de l'instruction
ARG	contient LEN octets (octets à écrire, PIN à vérifier, etc.)

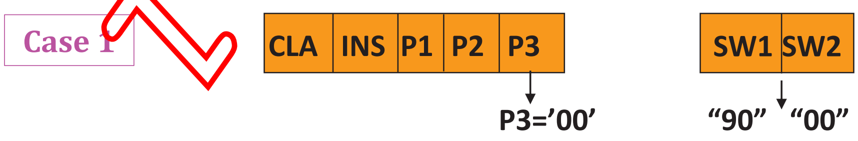
Return Codes

- 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…



Les différents cas en T=0

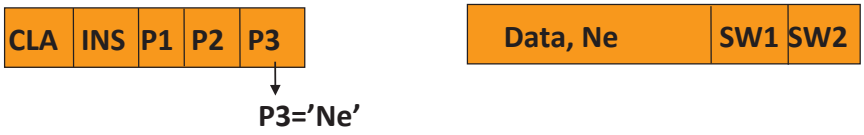
- Case 1
  - ❑ Command without data, response without data



Les différents cas en T=0

- Case 2 S (Le with 256 ≥ Ne ≥ 1 Ne =00 means 256)
  - ❑ Process completed; Ne accepted

Case 2S.1

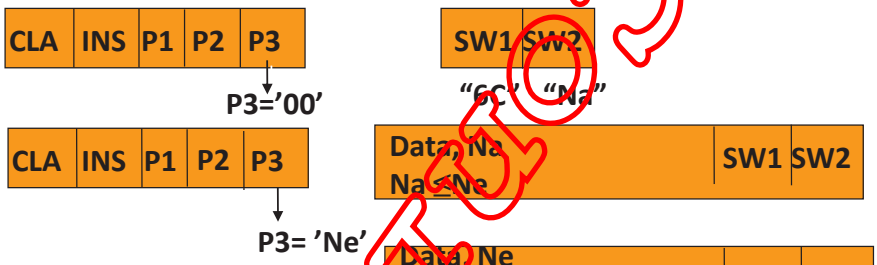


  - ❑ Process aborted; Ne definitely not accepted

Case 2S.2

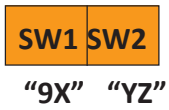
  - ❑ Process aborted; Ne not accepted, Na indicated

Case 2S.3



  - ❑ SW1 SW2 = '9XYZ', except for '9000'

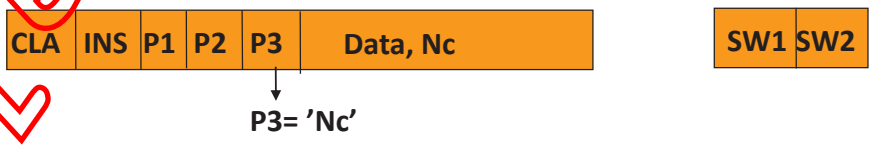
Case 2S.4



Les différents cas en T=0

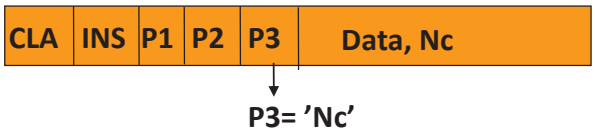
- Case 3S (Lc with 255 ≥ Nc ≥ 1)
  - ❑ Command with data, response without data

Case 3S



Les différents cas en T=0

- Case 4S (Lc with 255 ≥ Nc ≥ 1 and Le with 256 ≥ Ne ≥ 1 Ne =00 means 256)



P3= 'Nc'

- ❑ Process aborted

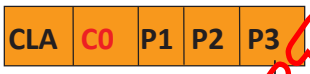
Case 4S.1

"6X" with 6X ≠ 61, 62 and 63

- ❑ Process completed

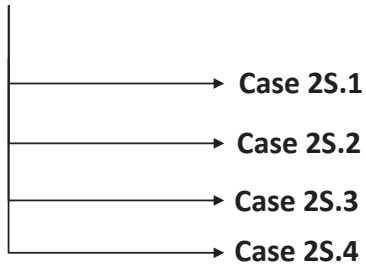
"90" "00"

Case 4S.2



GET RESPONSE

P3='Ne'



Les différents cas en T=0

- ❑ Process completed with information added

Case 4S.3

"61" "Nx"

GET RESPONSE

P3=min(Ne,Nx)

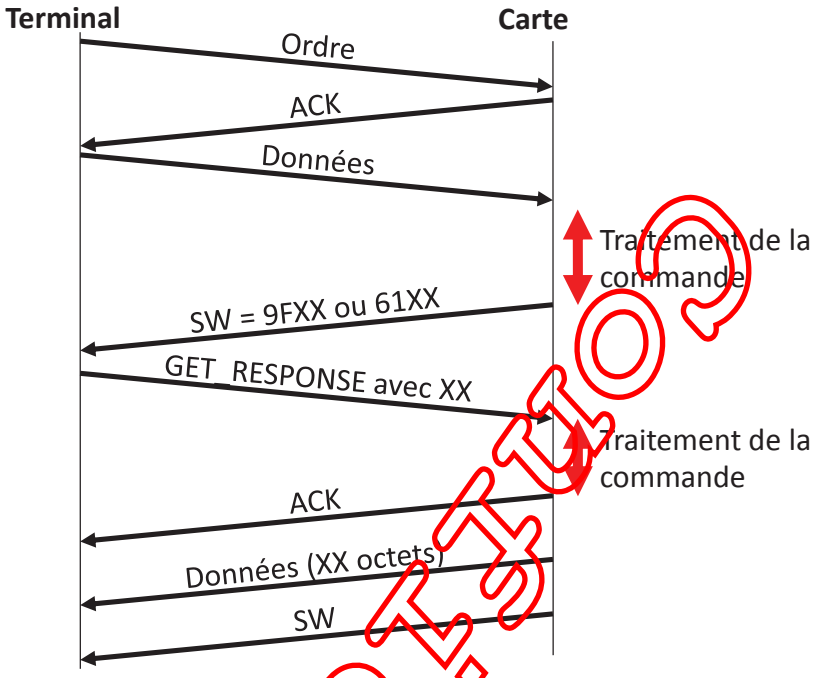
- ❑ SW1 SW2 = either '62XY' or '63XY' or '9XYZ', except for '9000'

Case 4S.4

"9X" "YZ"
"62" "XY"
"63" "XY"

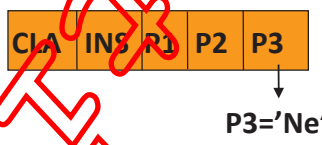
Comment est traité le cas 4 APDU (données entrantes et sortantes) ?

Utilisation de la commande spéciale GET_RESPONSE

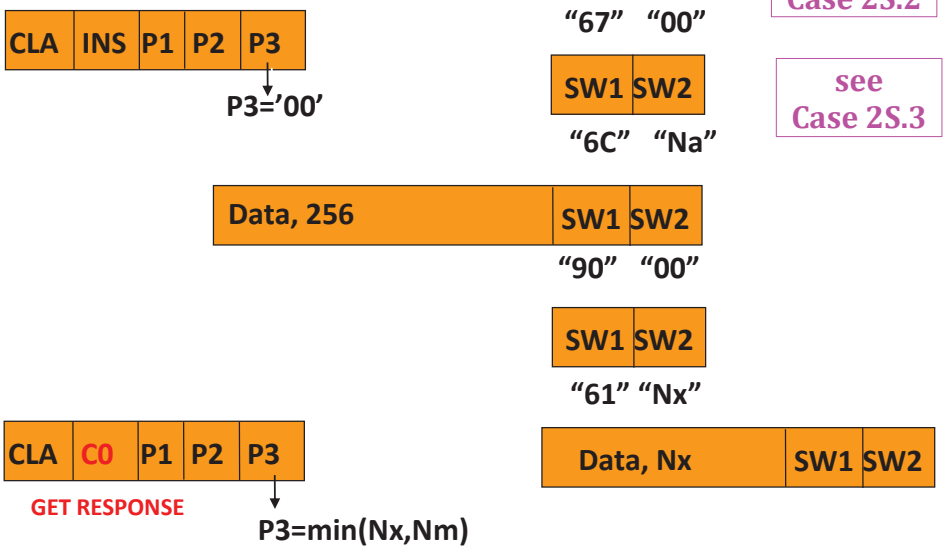


Les différents cas en T=0

- Case 2E (Le with 65536 ≥ Ne ≥ 1 Ne =0000 means 65536)
  - Case 2E.1 00xxxx with Ne = xxxx and Ne ≤ 256 i.e. xxx from 0001 to 0100
    - Similar to Case 2S



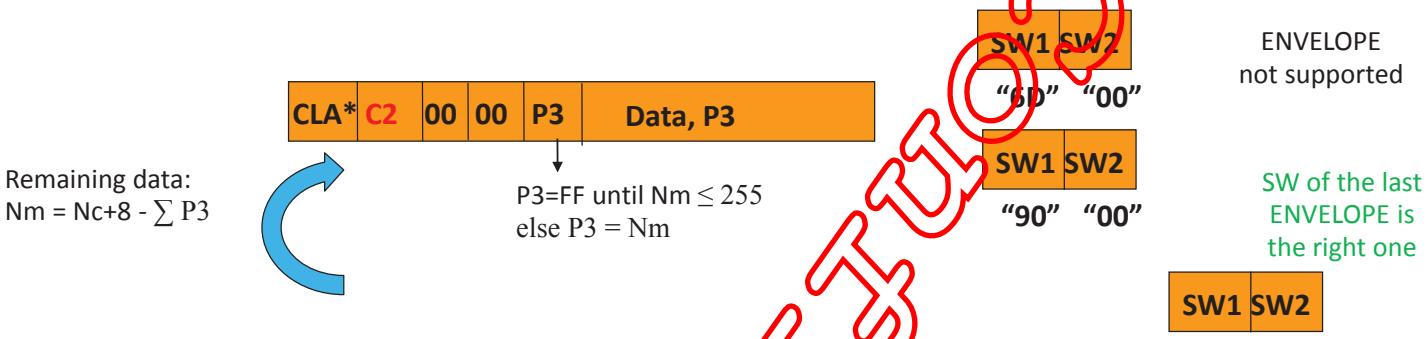
  - Case 2E.2 00xxxx with Ne = xxxx and Ne > 256 i.e. xxx either 0000 or from 0101 to FFFF
    - Since Ne>256 shall be mapped to Case 2S.3



Les différents cas en T=0

- Case 3E (Lc with 65535 ≥ Nc ≥ 1)
  - ❑ Case 3E.1 00xxxx with Nc = xxxx and 1 ≤ Nc ≤ 255 i.e. xxx from 0001 to 00FF
    - ➢ Similar to Case 3S

  - ❑ Case 3E.2 00xxxx with Nc = xxxx and Nc > 255 i.e. xxx from 0100 to FFFF
    - ➢ Since Nc>255 shall be split into consecutive segment of less than 256 bytes) using **ENVELOPE (C2/C3)**



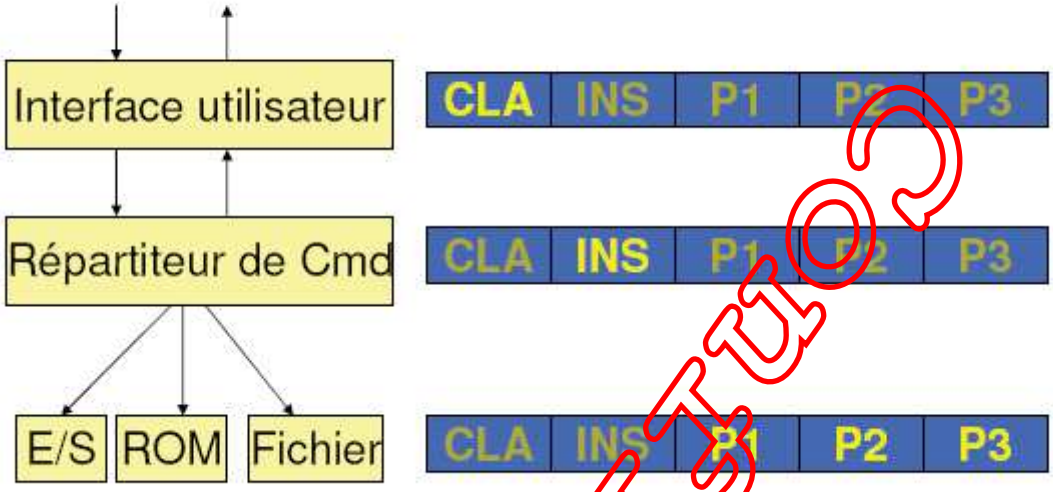
With CLA* = interindustry (000xxxxx or 01xxxxxx)

Data are initial APDU, i.e. CLA||INS||P1||P2||00||xx||xx||DATA[0]… DATA[y]… DATA[Nc-1]

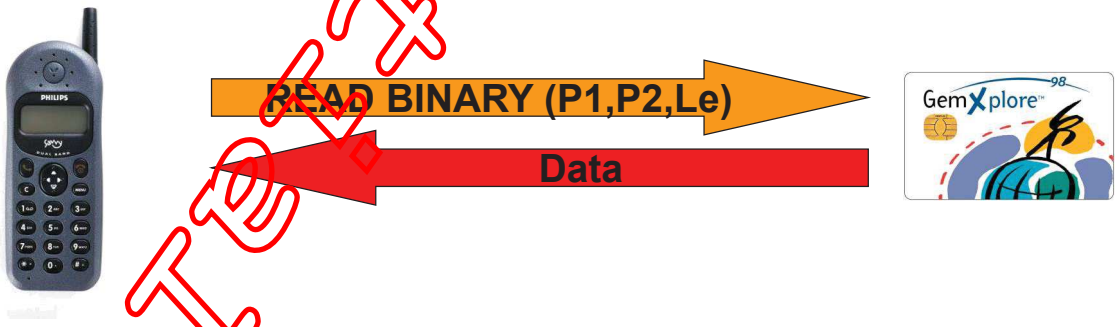
Les différents cas en T=0

- Case 4E
  - ❑ De simples combinaisons des cas précédents !

Un dispatcheur



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