Mifare classic 1k
&
NFC-Tools Api

Aghiles ADJAZ
&
Samia BOUZEFRANE
(samia.bouzefrane@cnam.fr)

le cnam

Table of contents

- Mifare Classic 1k
- NFC-Tools Api
- Demo
- Exercises

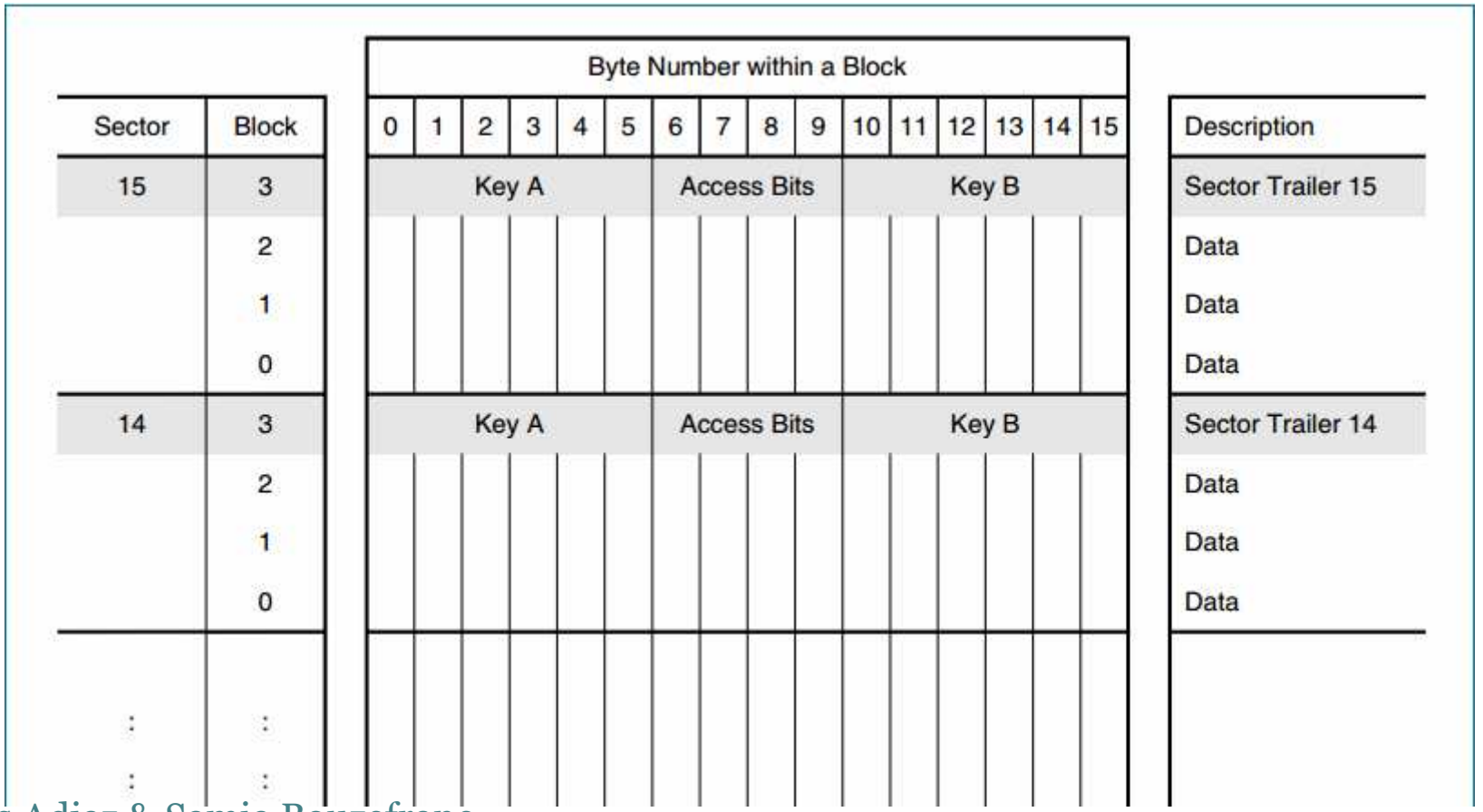
le cnam

Mifare Classic 1k description

- 16 sectors, each sector contains 4 blocks,
- 3 user blocks (block 0 to 2) and one key block (block 3)
- Sector 0 block 0 cannot be used (contains manufactory data)
- Each block contains 16 bytes.
  - 16 * 4 * 16 = 1024 bytes = 1k

le cnam

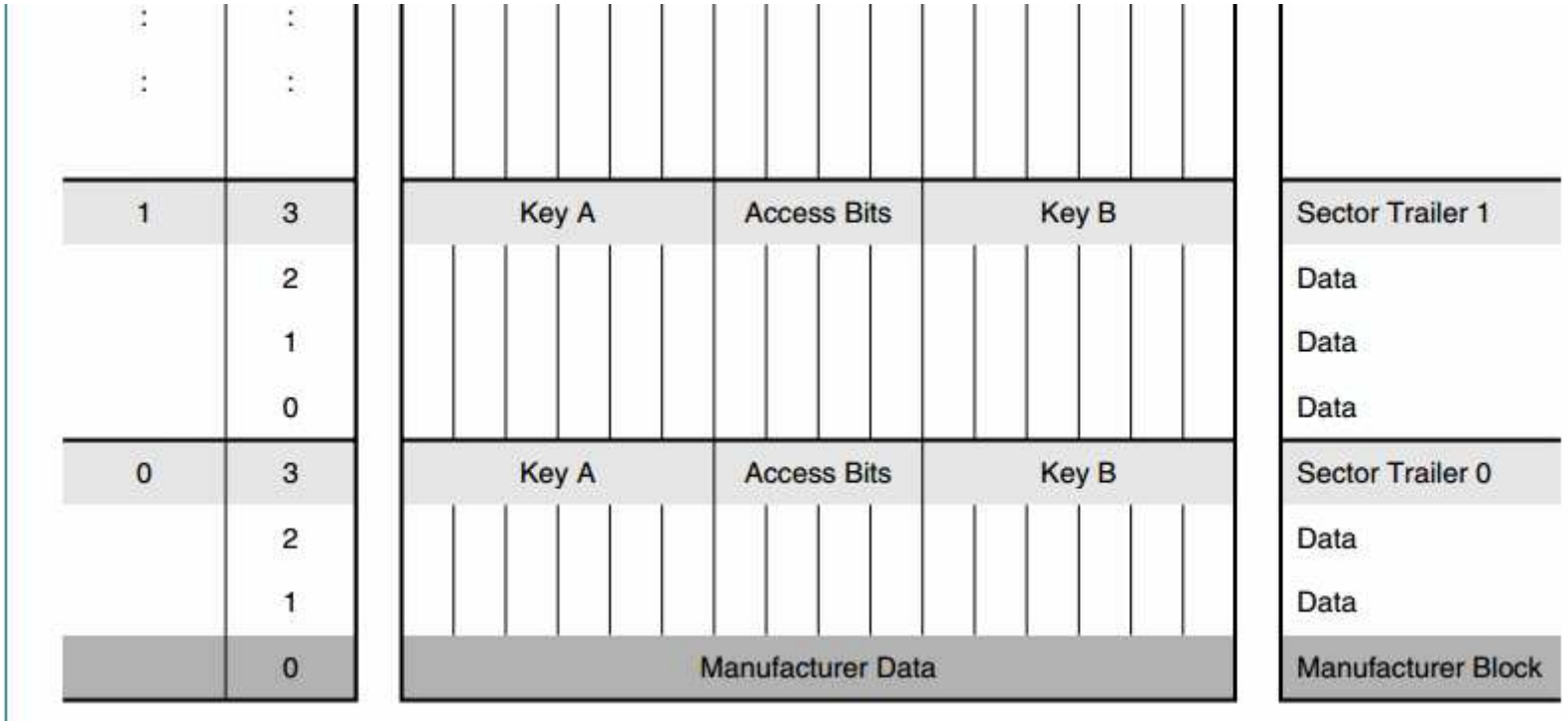
Memory organization



Sector	Block	Byte Number within a Block																Description
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
15	3	Key A						Access Bits				Key B						Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data
14	3	Key A						Access Bits				Key B						Sector Trailer 14
	2																	Data
	1																	Data
	0																	Data
:	:																	
:	:																	

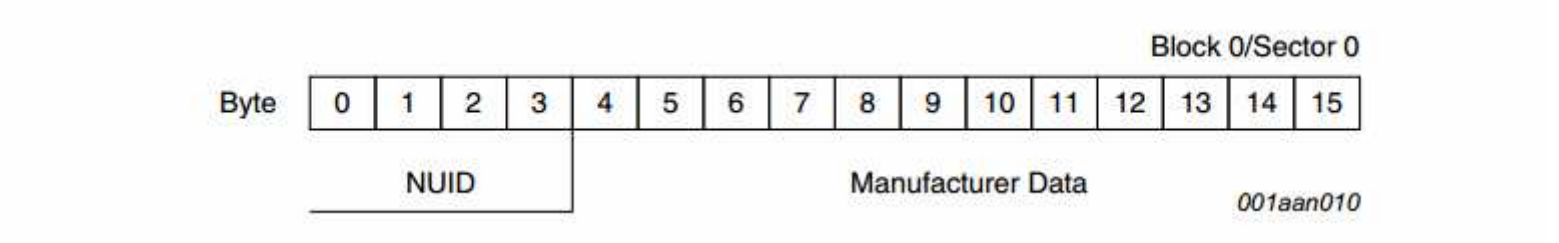
le cnam

Memory organization



le cnam

Manufacturer block

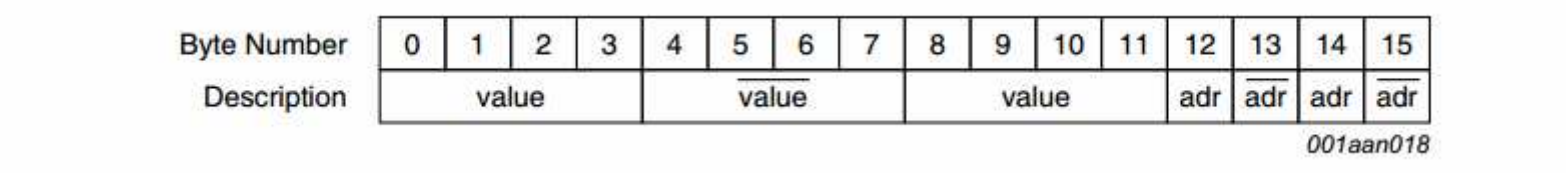


- Programmed and write protected during production
- Non unique Identifier

le cnam

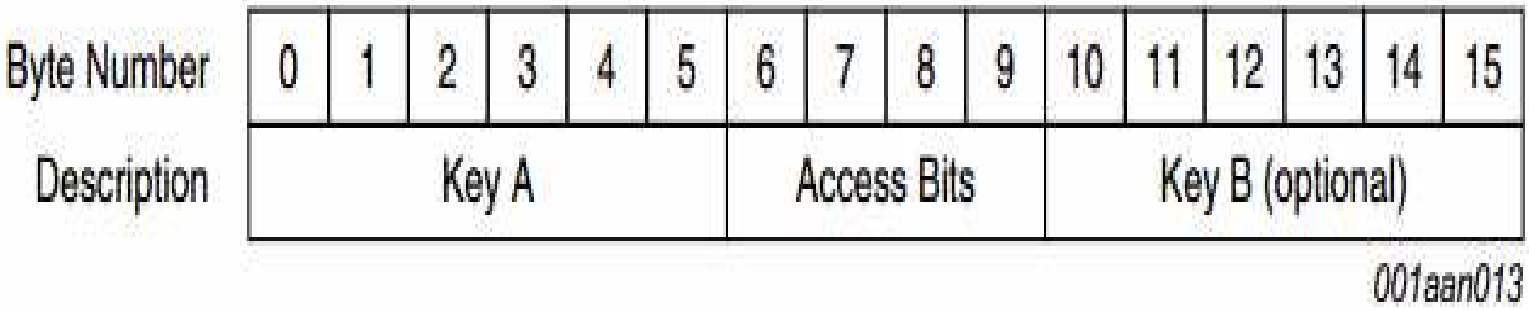
User's blocks

- Two types of user's blocks (Data block and Value Block)
- Data block has no memory structure (read/write block)
- Value Block has a memory structure with a value and an address



le cnam

Sector Trailer



- At chip delivery keys are set to FFFFFFFFFFFFh

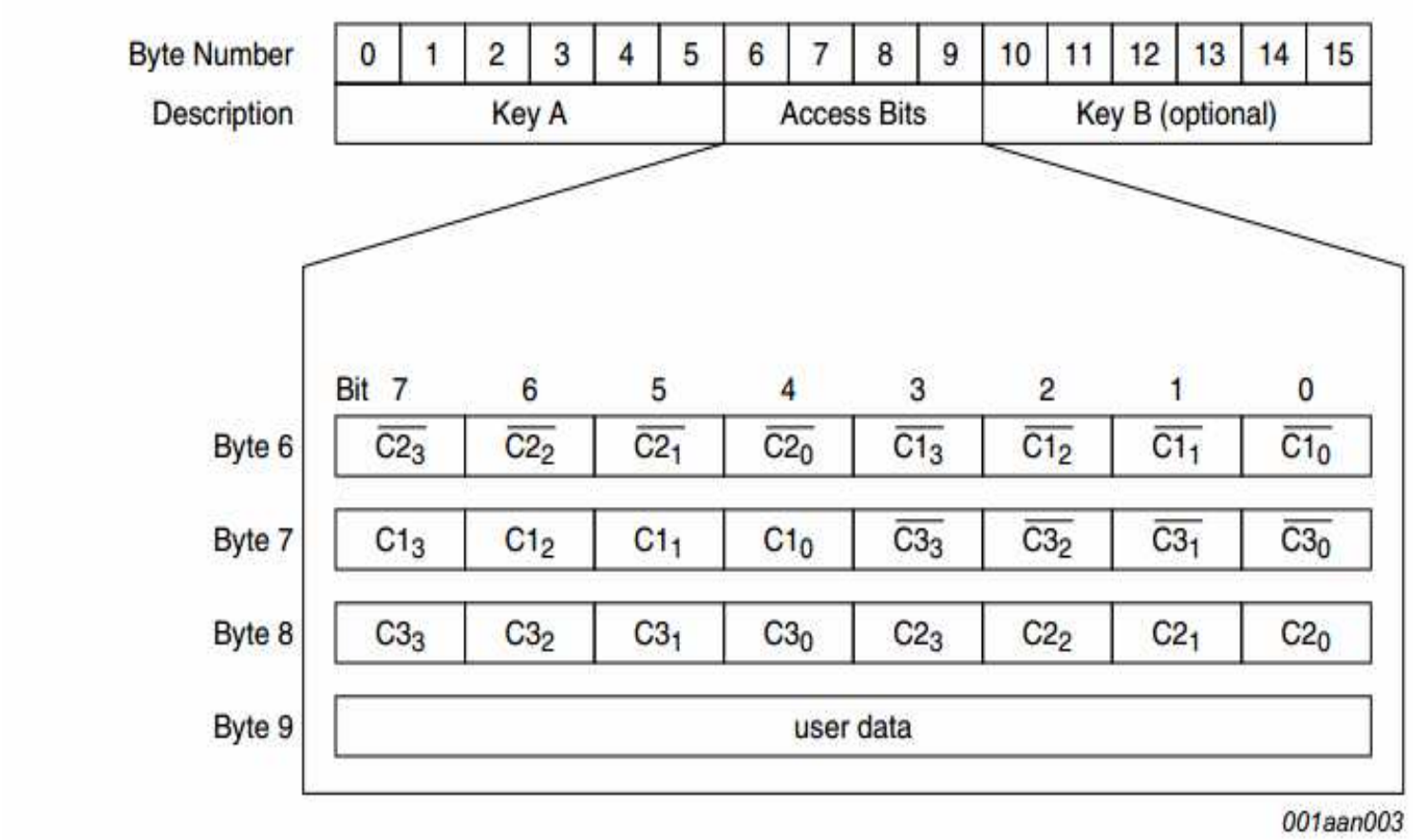
le cnam

Memory Access

Operation	Description	Valid for Block Type
Read	reads one memory block	read/write, value and sector trailer
Write	writes one memory block	read/write, value and sector trailer
Increment	increments the contents of a block and stores the result in the internal data register	value
Decrement	decrements the contents of a block and stores the result in the internal data register	value
Transfer	writes the contents of the internal data register to a block	value
Restore	reads the contents of a block into the internal data register	value

le cnam

Access conditions



le cnam

Access conditions for sector trailer

Access bits			Access condition for						Remark
			KEYA		Access bits		KEYB		
C1	C2	C3	read	write	read	write	read	write	
0	0	0	never	key A	key A	never	key A	key A	Key B may be read[1]
0	1	0	never	never	key A	never	key A	never	Key B may be read[1]
1	0	0	never	key B	key A|B	never	never	key B	
1	1	0	never	never	key A|B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration[1]
0	1	1	never	key B	key A|B	key B	never	key B	
1	0	1	never	never	key A|B	key B	never	never	
1	1	1	never	never	key A|B	never	never	never	

[1] for this access condition key B is readable and may be used for data

le cnam

Access conditions for data blocks

Access bits			Access condition for				Application
C1	C2	C3	read	write	increment	decrement, transfer, restore	
0	0	0	key A|B[1]	key A|B1	key A|B1	key A|B1	transport configuration
0	1	0	key A|B[1]	never	never	never	read/write block
1	0	0	key A|B[1]	key B[1]	never	never	read/write block
1	1	0	key A|B[1]	key B[1]	key B[1]	key A|B[1]	value block
0	0	1	key A|B[1]	never	never	key A|B[1]	value block
0	1	1	key B[1]	key B[1]	never	never	read/write block
1	0	1	key B[1]	never	never	never	read/write block
1	1	1	never	never	never	never	read/write block

[1] if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in previous table). As a consequences, if the reader authenticates any block of a sector which uses the grey marked access conditions and using key B, the card will refuse any subsequent memory access after authentication.

le cnam

NFC-Tools Api

- Class TerminalHandler
- Methods
  - void addTerminal(Terminal)
  - Terminal getAvailableTerminal(String)

Example:

```
TerminalHandler handler = new TerminalHandler();
AcsTerminal terminal = new AcsTerminal();
handler.addTerminal(terminal);
handler.getAvailableTerminal("ACS ACR1281 1S Dual Reader 00 01");
```

le cnam

NFC-Tools Api

- Class NfcAdapter
- Constructor NfcAdapter(Terminal, TerminalMode)
- Methods
  - void registerTagListener (NfcTagListener)
  - void startListening ()
  - void stopListening ()

le cnam

NFC-Tools Api for Mifare

- Class AbstractCardTool which implements NfcTagListener (generally we inherit from this class and redefine the method “doWithReaderWriter”)
- Class MfClassicAccess

  Constructor MfClassicAccess(KeyValue, int sector, int block, int blocksToRead)
- Class MfBlock
- Class DataBlock -> DataBlock(byte[] data)
- Class ValueBlock -> ValueBlock(int value, byte address)
- Class TrailerBlock -> TrailerBlock(byte[] data)

le cnam

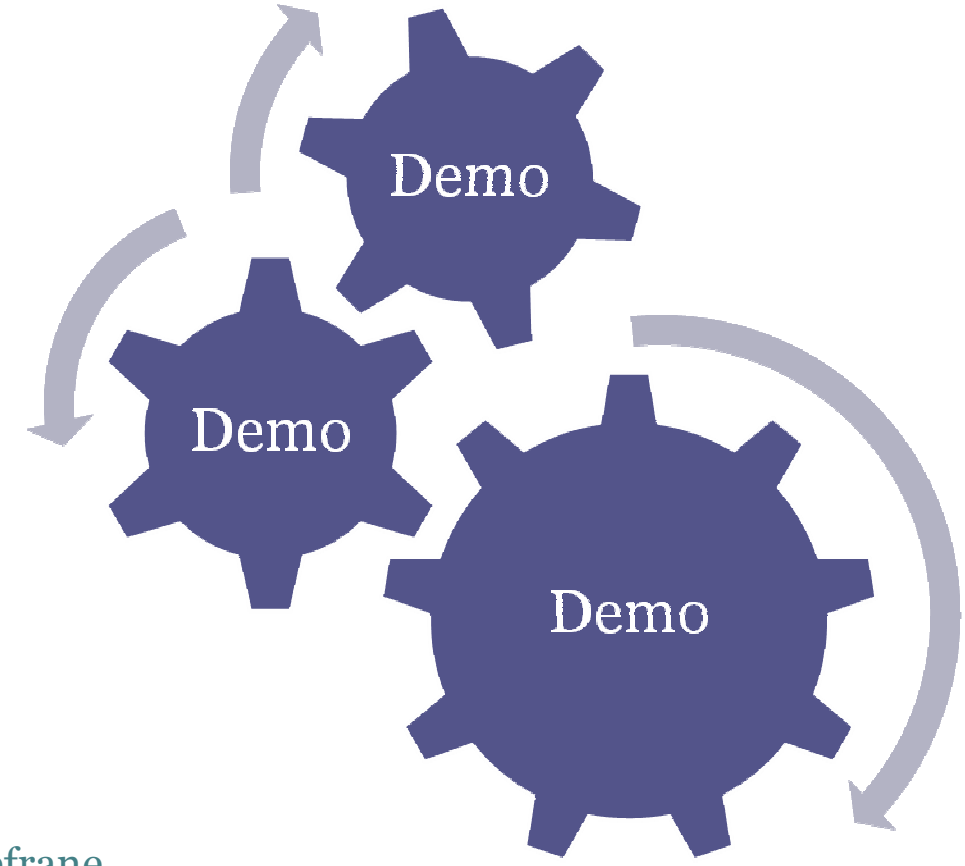
NFC-Tools Api

- Class MfClassicReaderWriter

- Methods
  - void writeBlock (MfClassicAccess , MfBlock)
  - MfBlock[] readBlock(MfClassicAccess)

le cnam

NFC-Tools Api



le cnam

Exercise 1

- Create an application which writes the value 2014 hex to the sector 4 block 0
- Modify the application that we saw in the demo to read only the sector 4 block 0
- The API (jar files) is accessible here:
  - http://cedric.cnam.fr/~bouzefra/cours/Api.zip
- The program that reads a Mifare tag is accessible here:
  - http://cedric.cnam.fr/~bouzefra/cours/Tag_Lecture.zip

le cnam

Exercise 2

- Modify the application of writing to write the trailer sector of sector 4 (change the value of key A to FFFFFFFFFFFE hex) DO NOT CHANGE THE ACCESS BYTES
- Run the application of reading (what do you see?)
- Modify the application of reading in order to read the sector 4 block 0
- Restore the default key A of the sector 4 (FFFFFFFFFFFF hex)

le cnam

References

- [MF1] MF1S503x Manual
- [GRU] https://github.com/grundid/nfctools
- [ACR] ACR122 and ACR128 Manual
- http://cedric.cnam.fr/~bouzefra/cours/Api.zip
- http://cedric.cnam.fr/~bouzefra/cours/Tag_Lecture.zip
- http://cedric.cnam.fr/~bouzefra/cours_smos.html