NFC Technologies for the Internet Of Things

Pr Pascal Urien

Telecom Paristech

Co-Founder of the EtherTrust Company

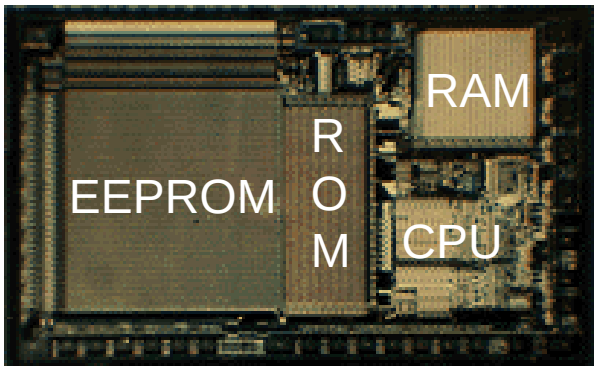
Agenda

- Introduction to NFC technologies
- About NFC standards
- NFC in mobile phones
- Identity For NFC
- Use Case 1: NFC Keys for the Internet of Things
- Use Case 2: The Emergence of the Cloud Of Secure Elements (CoSE)
- Use Case 3: Security for the NFC, LLCPS

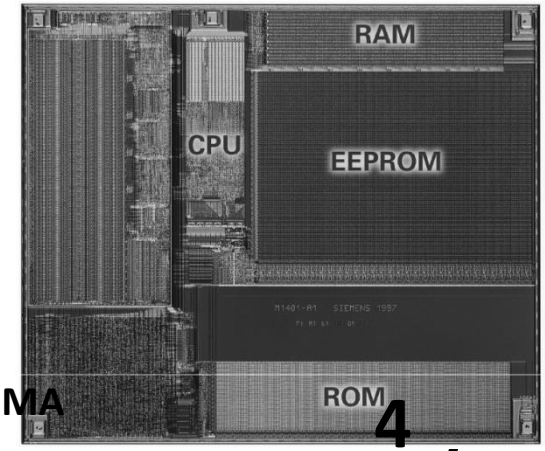
Introduction to NFC Technologies

Smartcard Genesis

- 1980, First BO' French bank card, from CP8
- 1988, SIM card specification
- 1990, First ISO7816 standards
- 1991, First SIM devices
- 1995, First EMV standards
- 1997, First Javacard
  - The javacard is a subset of the java language
  - Patent US 6,308,317
- 1998, JCOP (IBM JC/OP)
- 1999, Global Platform (GP)
- 2002, First USIM cards



1988, the 21 (BO') chip



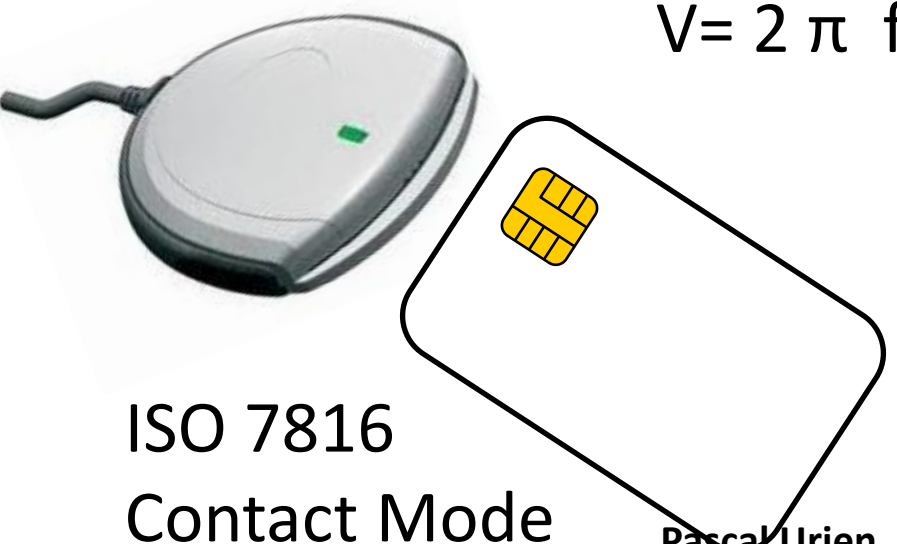
Siemens (SIM) chip, 1997

NFC Genesis

- 1994, Mifare 1K
  - In 2011 Mifare chips represent 70% of the transport market.
- 2001, ISO 14443 Standards (13,56 Mhz)
  - Type A (Mifare)
  - Type B
  - Type F (Felica)
- 2004, NFC Forum
  - Mifare (NXP), ISO14443A, ISO14443B, Felica (Sony)
  - Three functional modes :
    - Reader/Writer, Card Emulation, Peer to Peer
- NFC controllers realize NFC modes

From ISO 7816 to ISO 14443

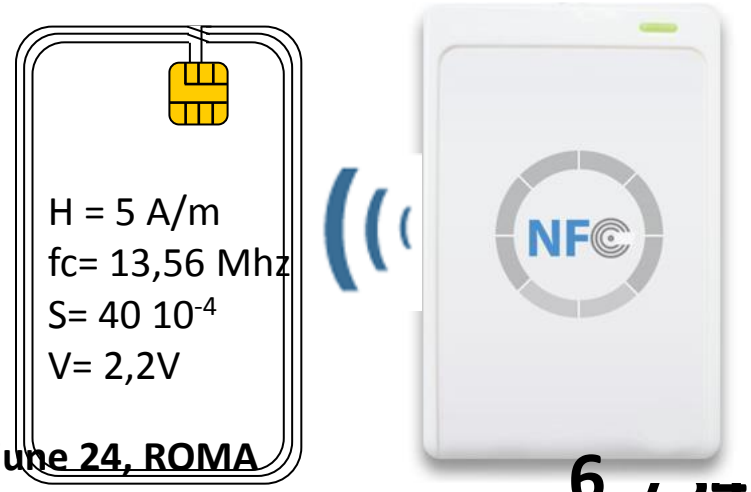
- The basic idea of Wi-Fi design was Wireless Ethernet.
- The basic idea of ISO 14443 design was Wireless (ISO 7816) Smartcard.
  - **Contrary to IEEE 802.11 there is no security features at the radio frame level.**



ISO 7816
Contact Mode

$$V = 2\pi f_c S \mu_o H$$

ISO 14443
Contactless Mode

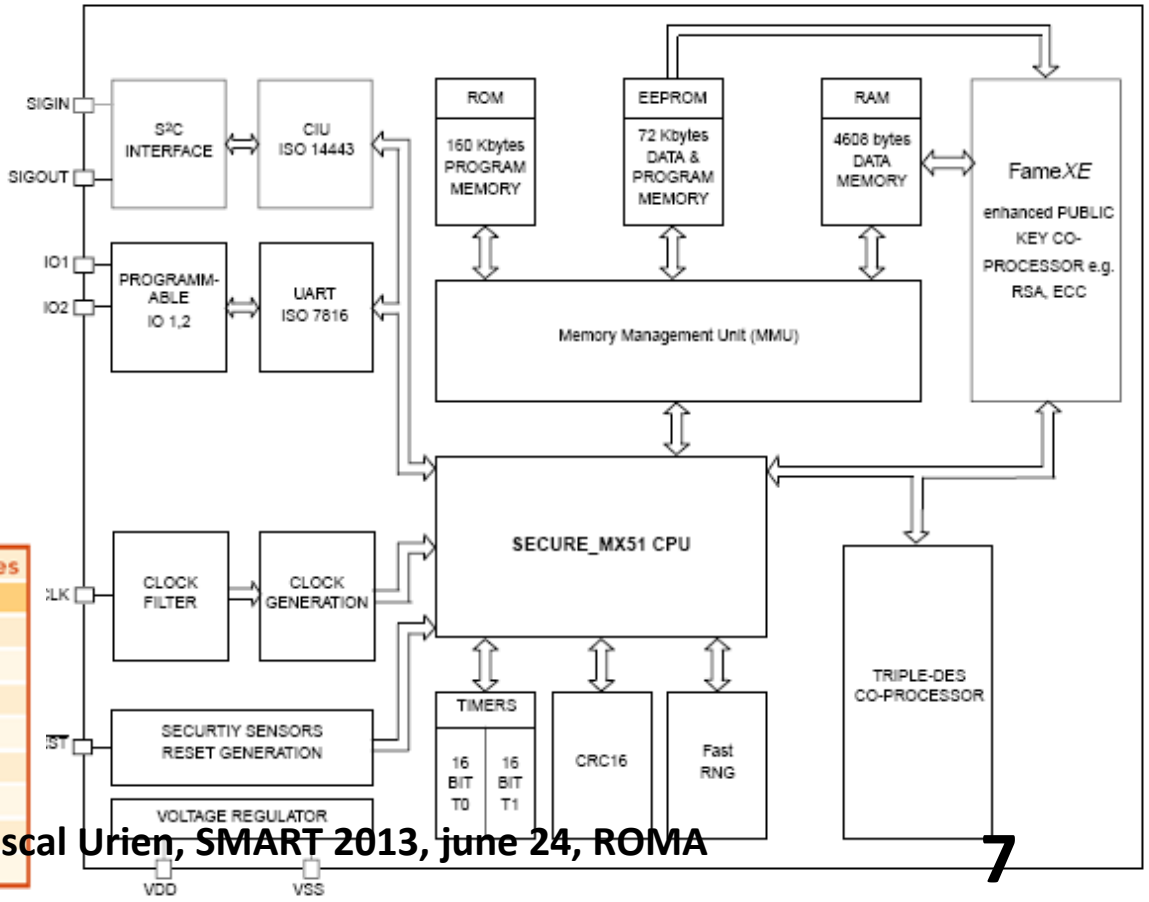


What is a Secure Element ?

A Secure Element (SE) is a Secure Microcontroller, equipped with host interfaces such as ISO7816, SPI or I^2C .

EXAMPLE: NXP PN532

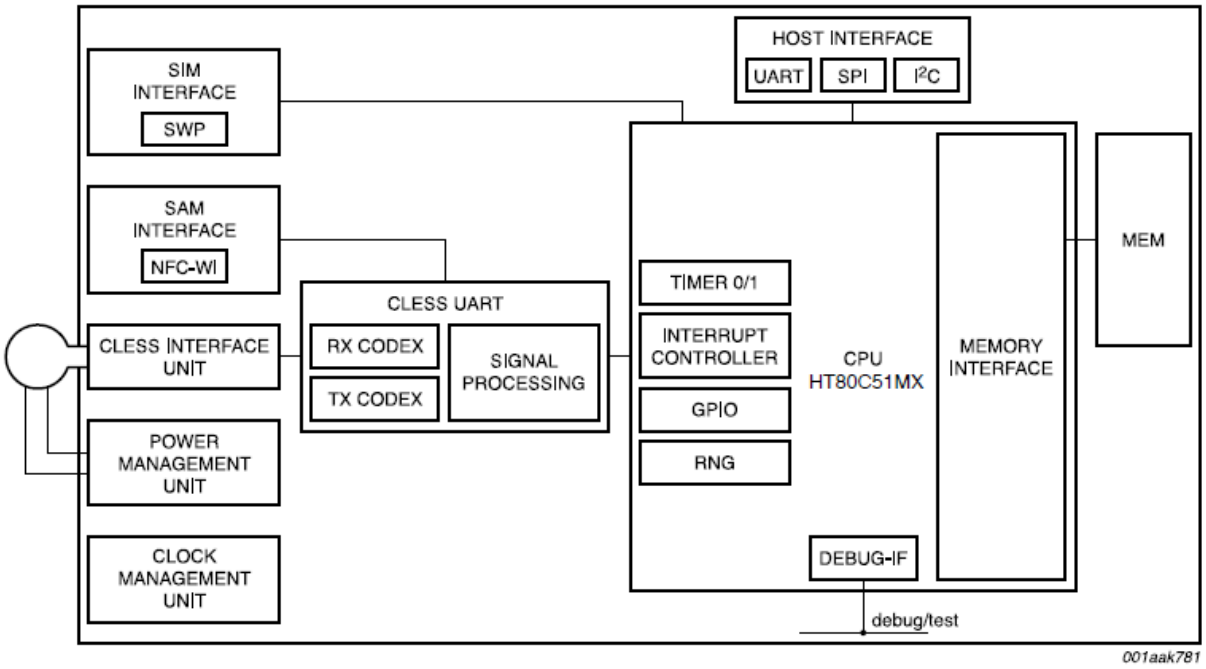
OS JAVACARD JCOP
GP (Global Platform)
ROM 160 KB
EEPROM 72 KB
RAM 4KB
Crypto-processor
3xDES, AES, RSA, ECC
Certification CC EAL5+
Security Certificates EMVCo

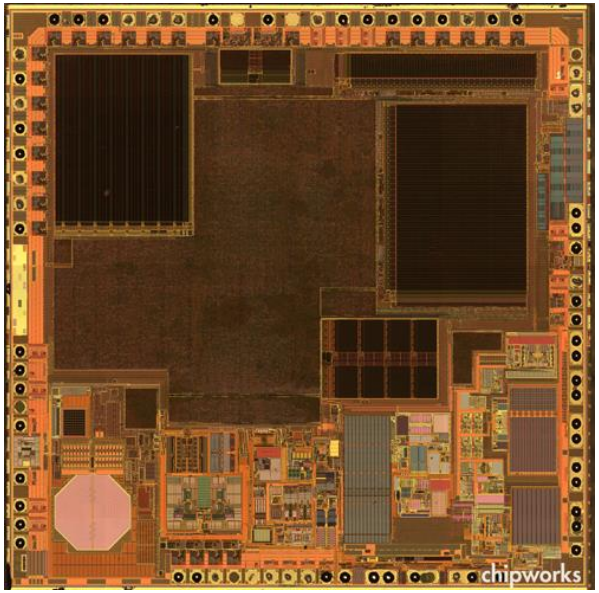


	NFC secure modules
Product features	**PN65L**
Embedded NFC IC	PN532
Available host interfaces	serial, SPI, I²C
Embedded Secure IC	P5CN072
OS for secure device	JCOP or 3rd party
Stacked passive component IC	yes
Package thickness	1.2 mm
Package size	7x7 mm²
Package type	HLQFN48

NFC and Secure Elements

- Some NFC Controllers embed a Secure Element
  - In that case the card emulation mode may be managed by the embedded secure element
  - This is the Google Secure Element Android Model





www.chipworks.com

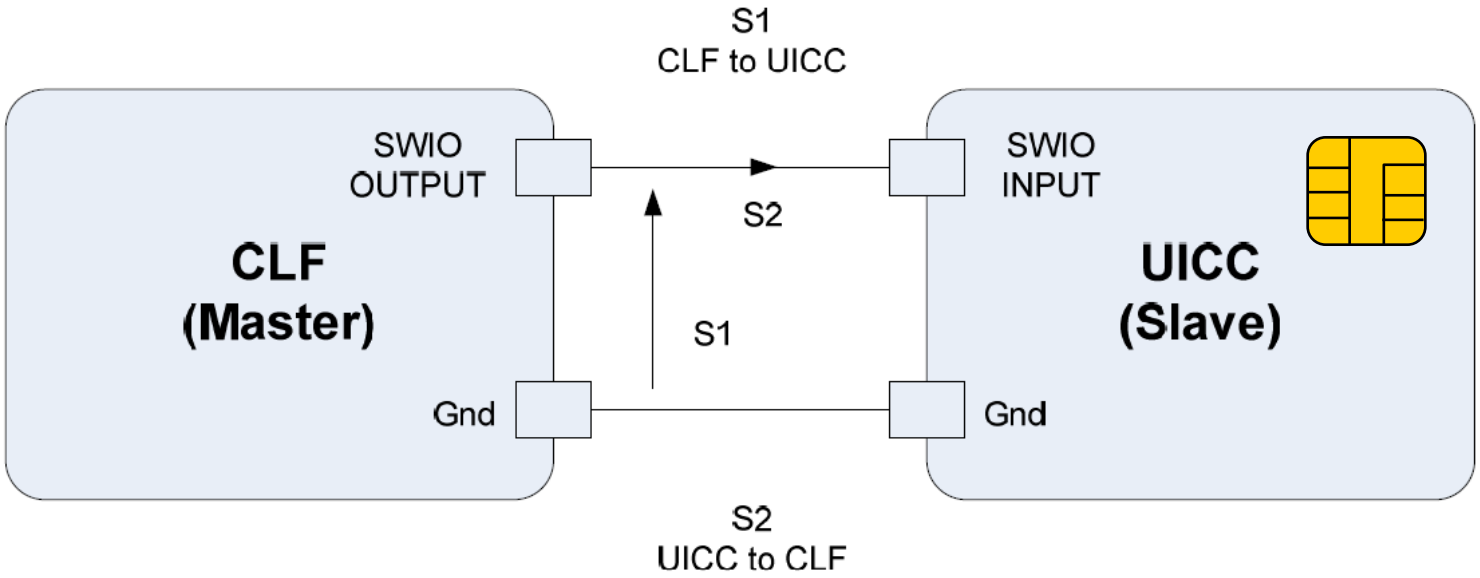
Reader/writer ISO 14443 –A-B, MIFARE, FeliCa®, NFC Forum tags, ISO 15693

Card Emulation ISO 14443 –A-B-B', MIFARE, FeliCa RF , SWP

RAM 5Ko, ROM 128 Ko, EEPROM 52 Ko

The SIM card becomes an NFC device: the Contactless Front-end (CLF)

The ETSI TS 102 613 Standard

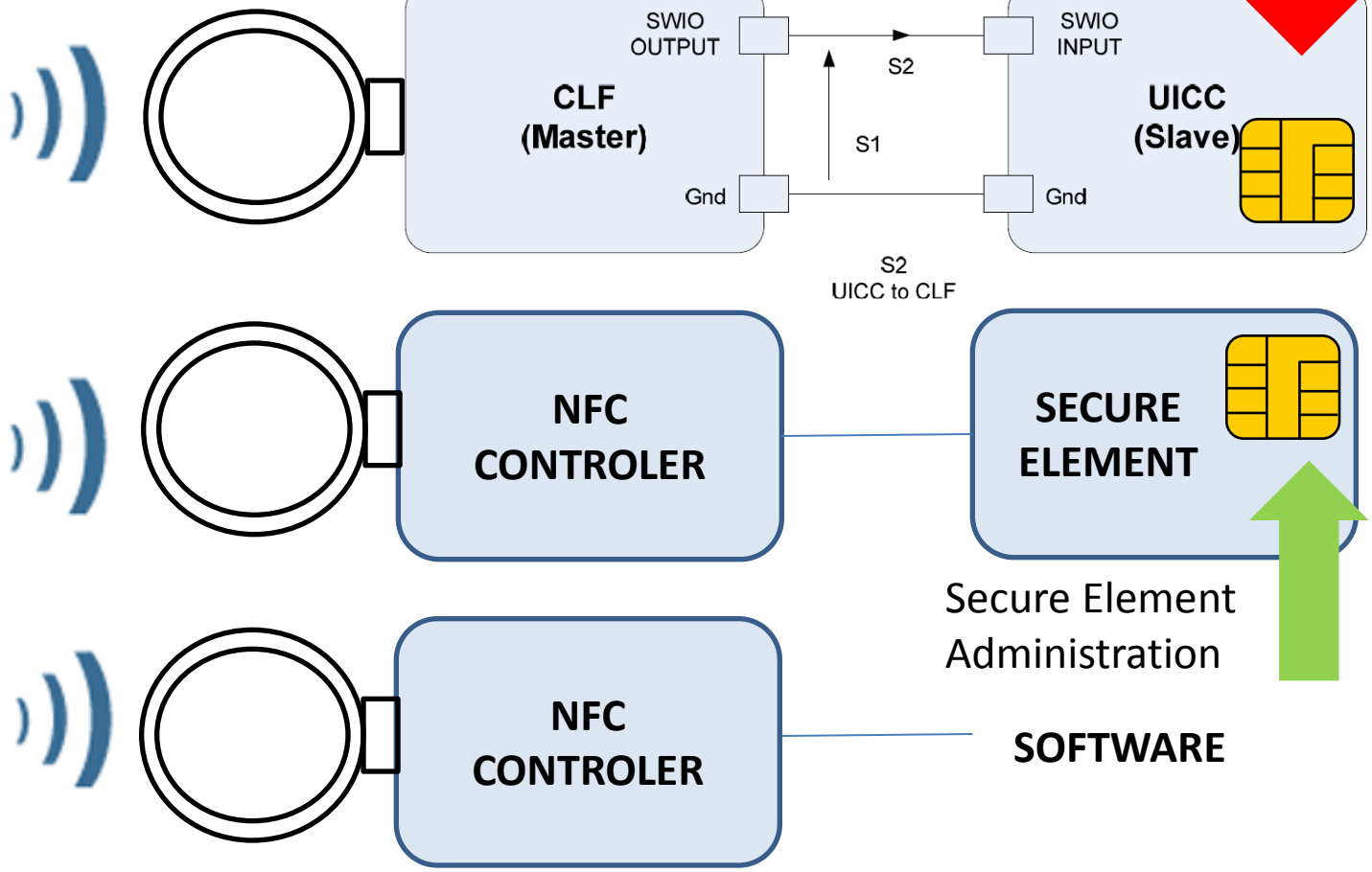


A simplified HDLC protocol: SHDLC

A physical Link: Single Wire Protocol (SWP)

NFC Reader/Writer & Card Emulation

OTA(Over The Air)
Administration
SMS, GSM 3.48



Reader/Writer

Card Emulation

NFC P2P Mode Illustration

- Android NDEF Push Protocol Specification
  - Version 1, 2011-02-22
  - Proprietary protocol, Android 2.3
  - Replace by SNEP for Android 4.x

"The NDEF Push Protocol (NPP) is a simple protocol built on top of LLCP which is designed to push an NDEF message from one device to another."

Initiator Target

Using the Google NDEF Push Protocol (NPP)

NFC Initiator

NFC Target

ATR_REQ, NFC-MAGIC VERSION WKS (Well-Known Service) LTO (Link-Timeout)

ATR_RES, NFC-MAGIC VERSION WKS (Well-Known Service) LTO (Link-Timeout)

LLCP-SYMM [0000]

CONNECT [0521 060F 636F6D2E616E64726F69642E6E7070]
DSAP=1, SSAP=33, Service="com.android.npp"

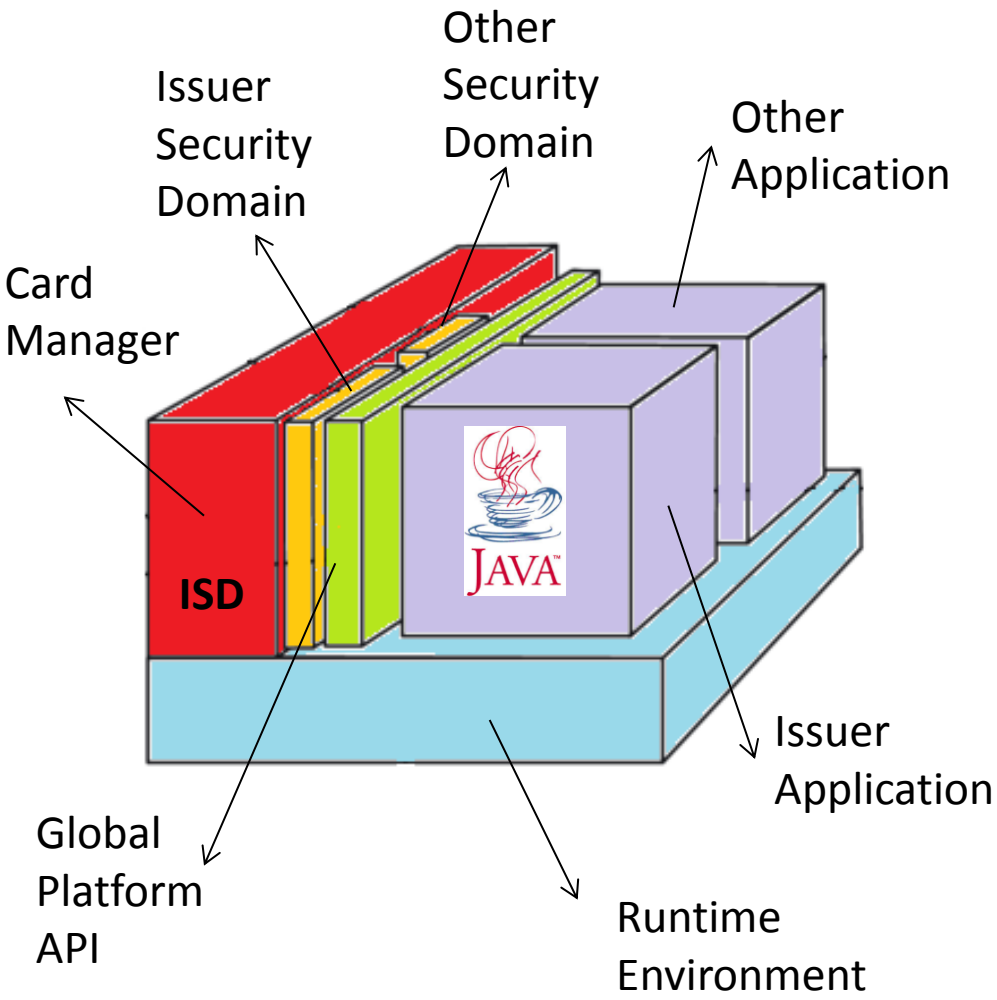
CC (Connection Complete) [859002020078]
DSAP=33, SSAP=16, MUI (Maximum Information Unit)

Information
[**432100 0100000001010000000F D1010B5402656E 6B657976616C7565**]
DSAP=16, SSAP=33, N(S)=0, N(R)=0, NPP HEADER, NDEF RECORD, **keyvalue**

RR(1) [855001], SSAP=16, DSAP=33

DISCONNECT [4161] DSAP=16, SSAP=33

Smartcard Administration: GP



The VISA Model*

KMC-ID (6B)

CSN (Chip Serial Number, 4B)

KMC (DES Master Key for Personalization Session Keys)

K_{ENC} K_{MAC} K_{DEK}

Select ISD

Mutual Authentication

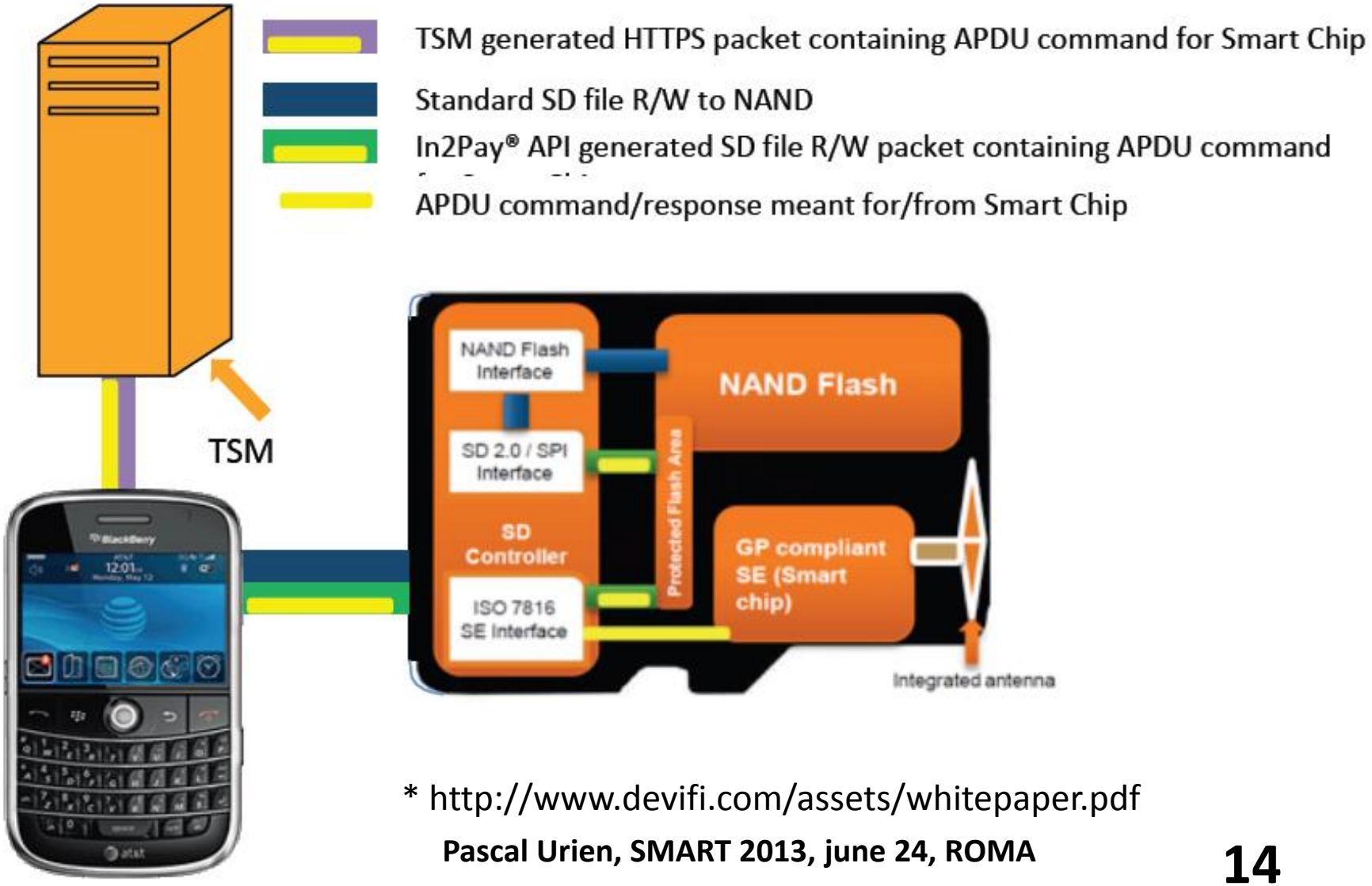
SKU_{ENC} SKU_{MAC} SKU_{DEK}

Secure Channel

Application Management downloading - deletion

*EMV Card Personalization Specification Version 1.1 July 2007

The In2Pay Administration Model*



* http://www.devifi.com/assets/whitepaper.pdf

The Google Platform

Reader/Writer

Card Emulation

Peer to Peer

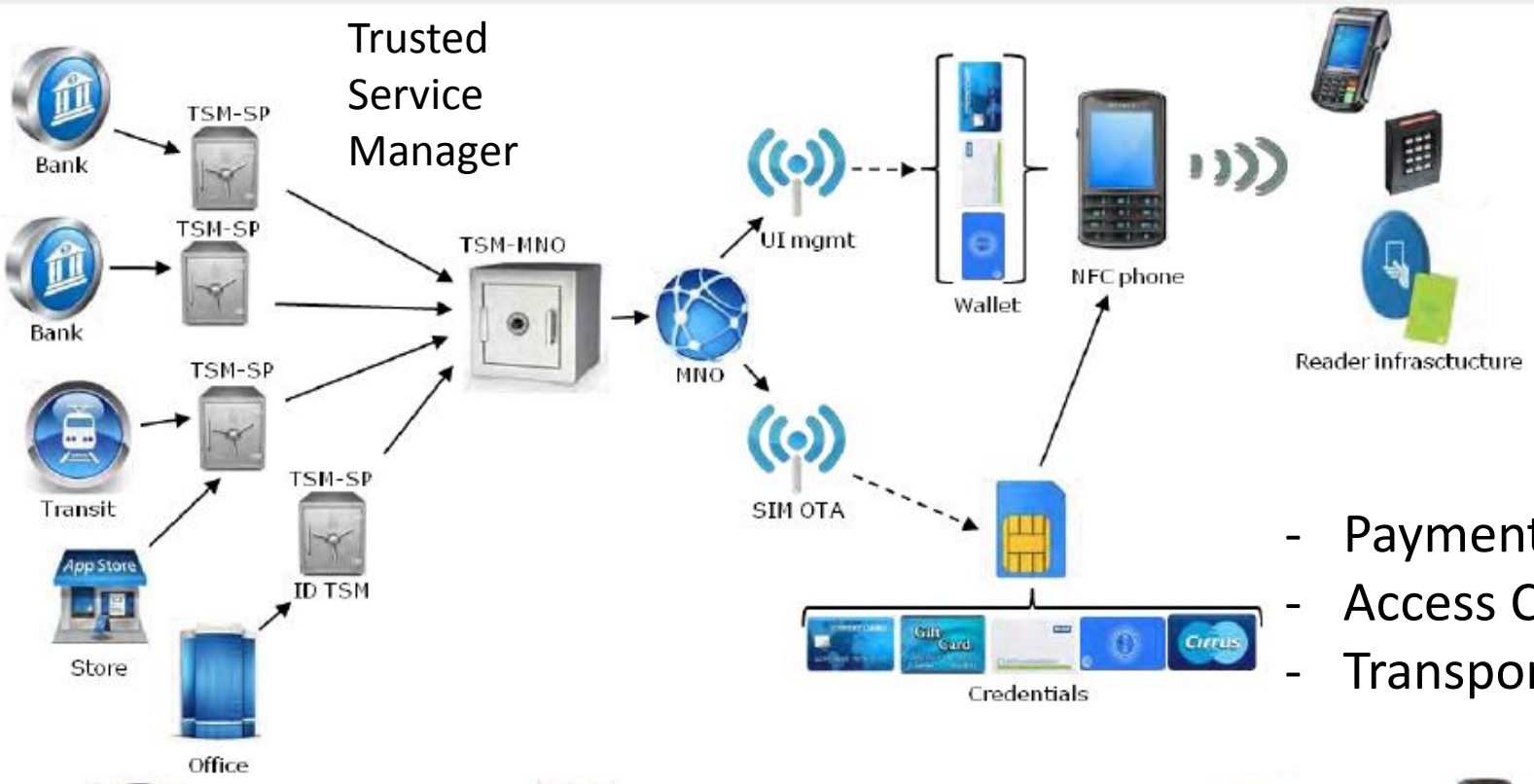
Android Beams

NFC Tags

- EMV Magnetic Stripe Profile
- Cloud Storage

SNEP

HID NFC White Paper: SIM centric Services



- Payment
- Access Control
- Transport

NFC ecosystem with the Secure Element in the SIM and one MNO

About NFC Specifications

In the NFC Jungle

NFC Standards Overview

Activity	Technology / Device Platform						
Listen, RF Collision Avoidance, Technology Detection, Collision Resolution	NFC-A ISO 14443-2A ISO 14443-3A				NFC-B 14443 -2B 14443 -3B	NFC-F ISO 14443-2A ISO 14443-3A FELICA	
Device Activation		Type 1 Tag Platform	Type 2 Tag Platform	Type 4A Tag Platform	Type 4B Tag Platform	Type 3 Tag Platform	
Data Exchange	NFC-DEP Protocol	Type 1, 2, and 3 Tag Half-duplex Protocol		ISO-DEP Protocol		Type 1, 2, and 3 Tag Half-duplex Protocols	NFC-DEP Protocol
Device Deactivation	NFCIP-1			ISO 14443-4			NFCIP-1

NDEF

SNEP

LLCP

Passive Mode
Active Mode

NFCIP-1

*ISO/IEC_18092 standard and NFCIP-1 standards are similar

DEP: Data Exchange Protocol (Supports Read/Write Operations for Tags)

NFC Radio

ISO 14443
106 kbps
212 kbps
424 kbps
848 kbps

Standard	PCD to ICCC Reader to Card	PICC to PCD Card to Reader
ISO 14443-2A NFC-A	ASK 100% Modified Miller	Subcarrier fc/16 OOK Manchester
ISO 14443-2B NFC-B	ASK 10%, NRZ-L	Subcarrier fc/16 BPSK, NRZ-L

NFCIP-1
Passive
Mode

Bit Rate	Initiator	Target
106 kbps	ASK 100% Modified Miller	Subcarrier fc/16 OOK Manchester
212-424 kbps	ASK 8-30% OOK Manchester	ASK 8-30% OOK Manchester

NFCIP-1
Active
Mode

Bit Rate	Initiator	Target
106 kbps	ASK 100% Modified Miller	ASK 100%, Modified Miller
212-424 kbps	ASK 8-30 % OOK Manchester	ASK 8-30%, OOK Manchester

NFC TAGs

NDEF Format for passive TAG

- **Type 1**
  - Based on ISO 14443-A
  - Innovision Topaz, Broadcom BCM20203
- **Type 2**
  - Similar to Type1
  - Based on ISO 144413-A
  - Compatible with NXP MIFARE Ultralight.
- **Type 3**
  - Similar to Type1
  - Based on the Japanese Industrial Standard (JIS) X 6319-4.
  - Compatible with Sony Felica
- **Type 4**
  - Similar to Type1
  - Based on ISO 14443-A
  - Compatible with standard ISO 14433-4 Smartcards
- **NXP-specific type tag**
  - Mifare Classic

LLCP NDEF services

- SNEP: Simple NDEF Exchange Protocol
- SNEP Requests and SNEP Responses
- LLC service access point address 4
- Service Name “urn:nfc:sn:snep”

LLCP: a Bridge to LAN Technologies

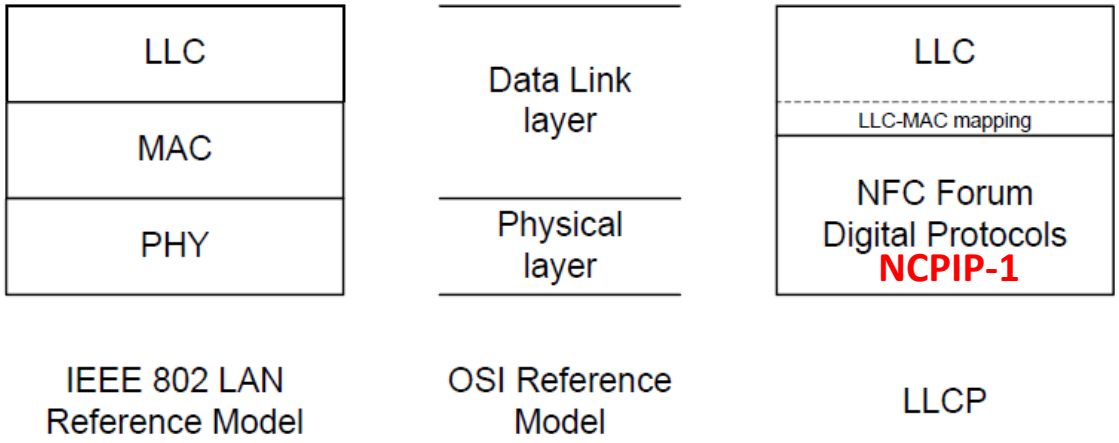
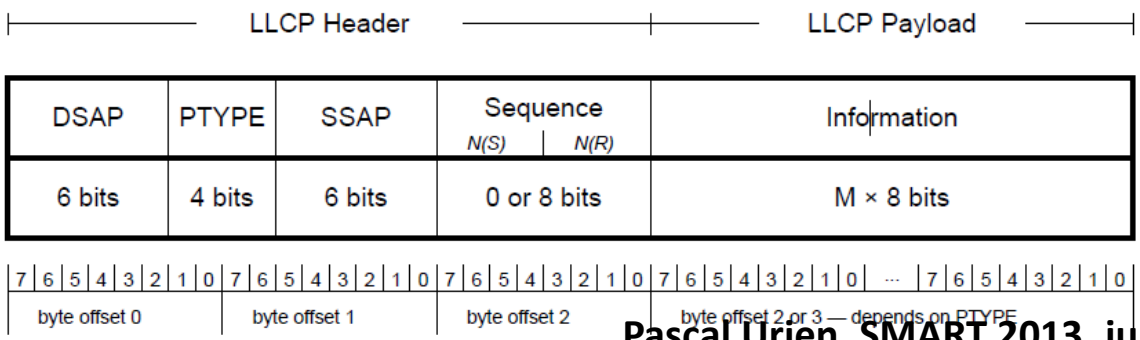


Figure 1: Relationship to OSI Reference Model

Table 3: PDU Type Values

PDU Type	PTYPE	Link Service Class
SYMM	0000	1, 2, 3
PAX	0001	1, 2, 3
AGF	0010	1, 2, 3
UI	0011	1, 3

PDU Type	PTYPE	Link Service Class
CONNECT	0100	2, 3
DISC	0101	1, 2, 3
CC	0110	2, 3
DM	0111	1, 2, 3
FRMR	1000	2, 3
SNL	1001	1, 2, 3
reserved	1010	
reserved	1011	
I	1100	2, 3
RR	1101	2, 3
RNR	1110	2, 3
reserved	1111	



NDEF: NFC Data Exchange Format

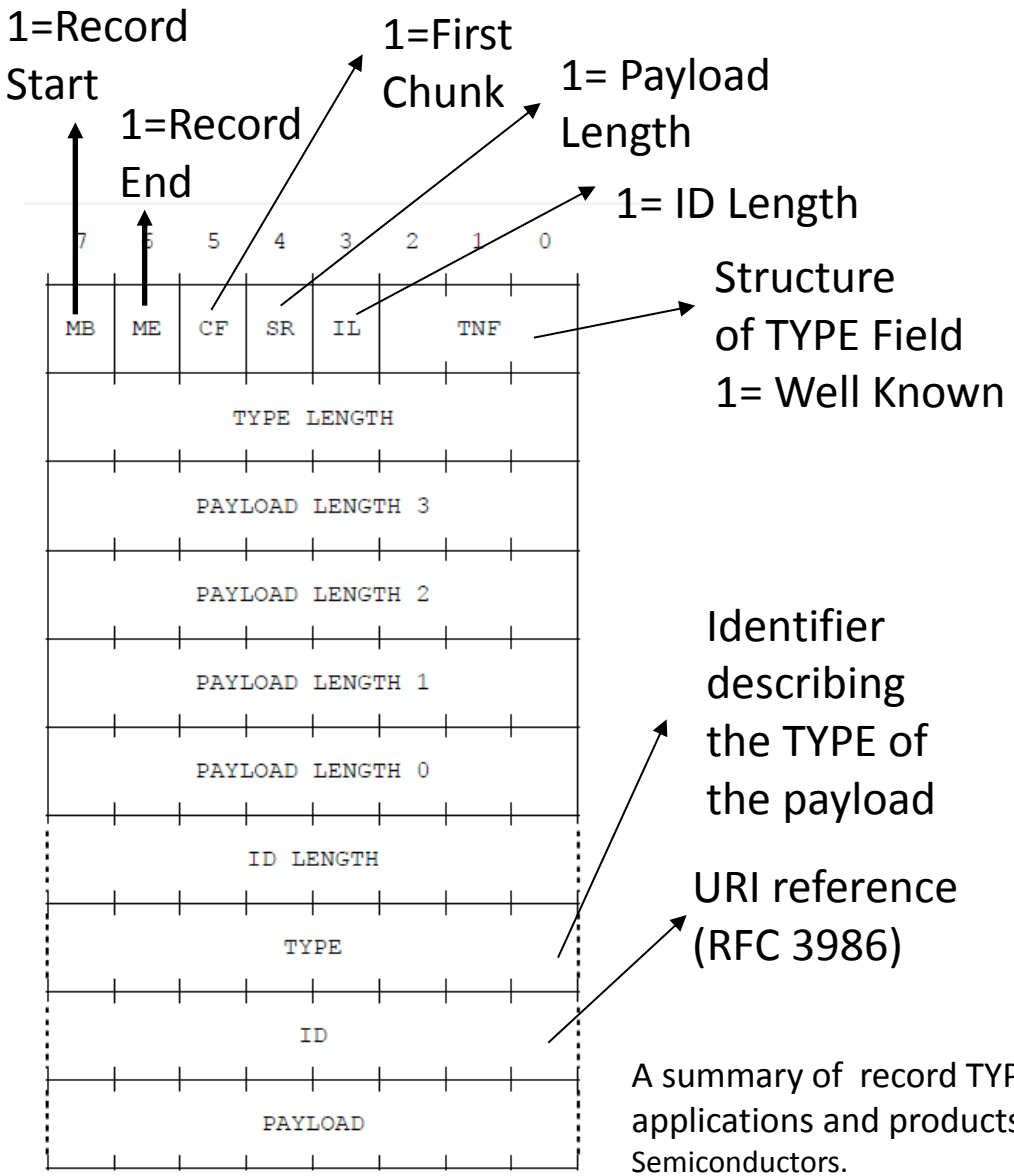


Figure 3. NDEF Record Layout

NDEF Record Example:
(NFC Text Record Type Definition)

D1: 1 1 0 1 0 001
01: Type Length
0A: Payload Length
54: Type= 'T', Text
02: ID= UTF8
65 6E: "EN"
53 61 6D 70 6C 65 20: "Sample "

A summary of record TYPE may be found in "NFC Tags A technical introduction, applications and products Rev. 1.3 - 1 December 2011 White paper", NXP Semiconductors.

Example of Type2 Tag with Mifare

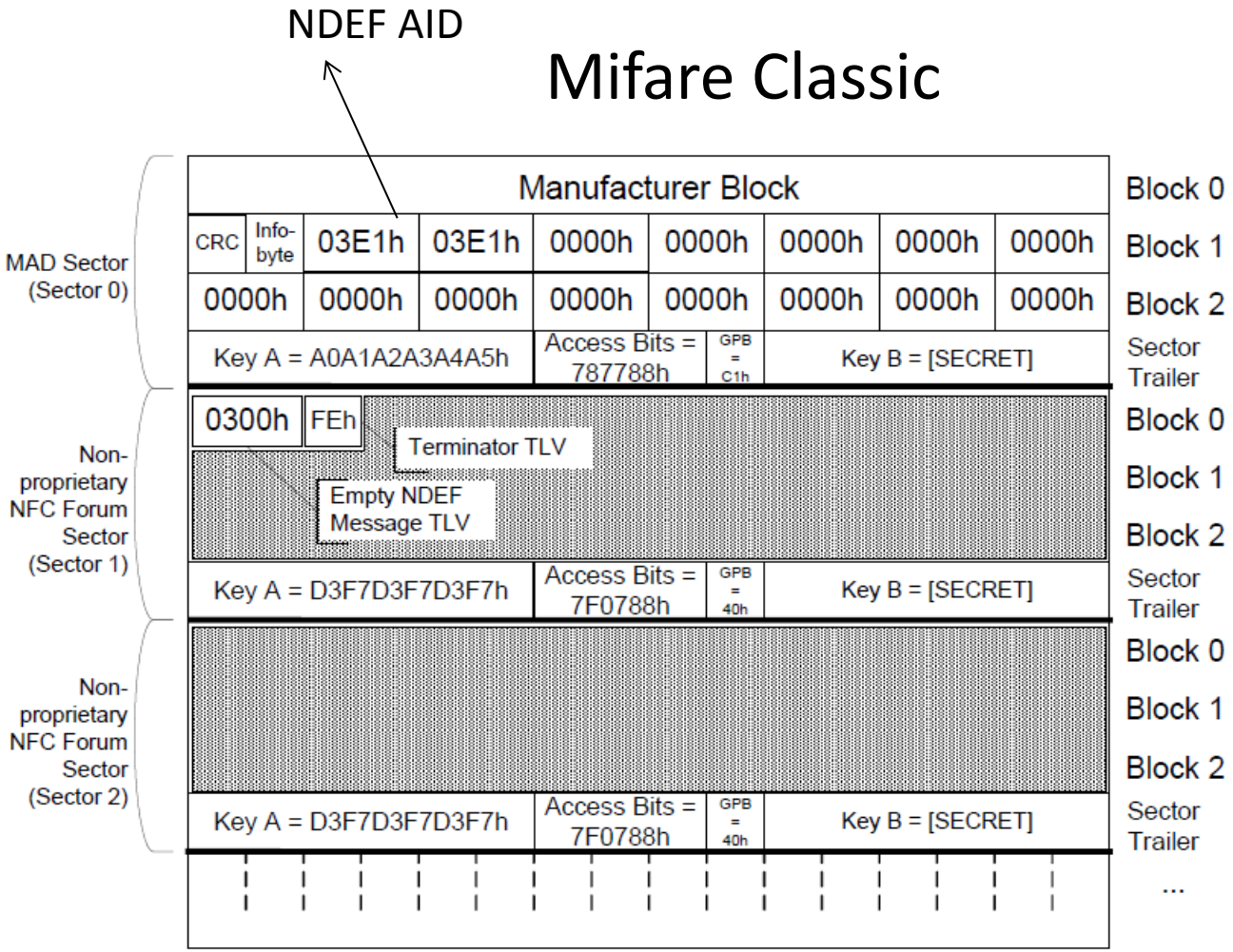
Mifare Ultralight

Block	Hex		ASCII
0	04D3	E0BF	.Óà¿
1	0277	1E80	.w.€
2	EB48	0000	ëH..
3	E110	0600	á...
4	030E	D101	..Ñ.
5	0B54	0265	.T.e
6	6E53	616D	nSam
7	706C	6520	ple
8	FE00	0000	þ...
9	0000	0000	....
10	0000	0000	....
11	0000	0000	....
12	0000	0000	....
13	0000	0000	....
14	0000	0000	....
15	0000	0000	....

Type2 Tag

Size 48 bytes

Mifare Classic



Mifare Application Directory, MAD

SNEP, Android 4.x

NFC Initiator

NFC Target

ATR_REQ, NFC-MAGIC VERSION WKS LTO →

← **ATR_RES**, NFC-MAGIC VERSION WKS LTO

LLCP-SYMM [0000] →

LLCP-SYMM [0000] ←

LLCP-SYMM [0000] →

CONNECT [1120], DSAP=4, SSAP=32 ←

CC [818402020078], DSAP=32, SSAP=4, MUI →

Information , SNEP PUT
[132000 10020000000F D1010B5402656E **6B657976616C7565**]
DSAP=4, SSAP=32, NS=0, NR=0, SNEP HEADER, NDEF RECORD**, keyvalue** ←

RR(1) [834401], SSAP=4, DSAP=32 →

Information, SNEP Success [830401 10810000000] ,
SSAP=4, DSAP=32, NS=0, NR=1 →

RR(1) [13600], DSAP=4, SSAP=32 ←

DISCONNECT [1160] DSAP=4, SSAP=32 ←

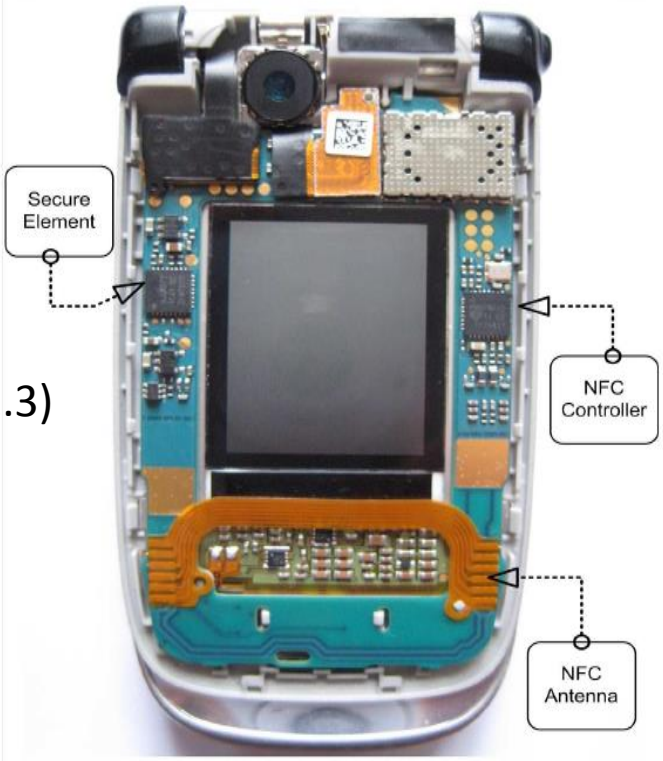
DM [C400], DSAP=16, SSAP=32 →

LLCP-SYMM [0000] ←

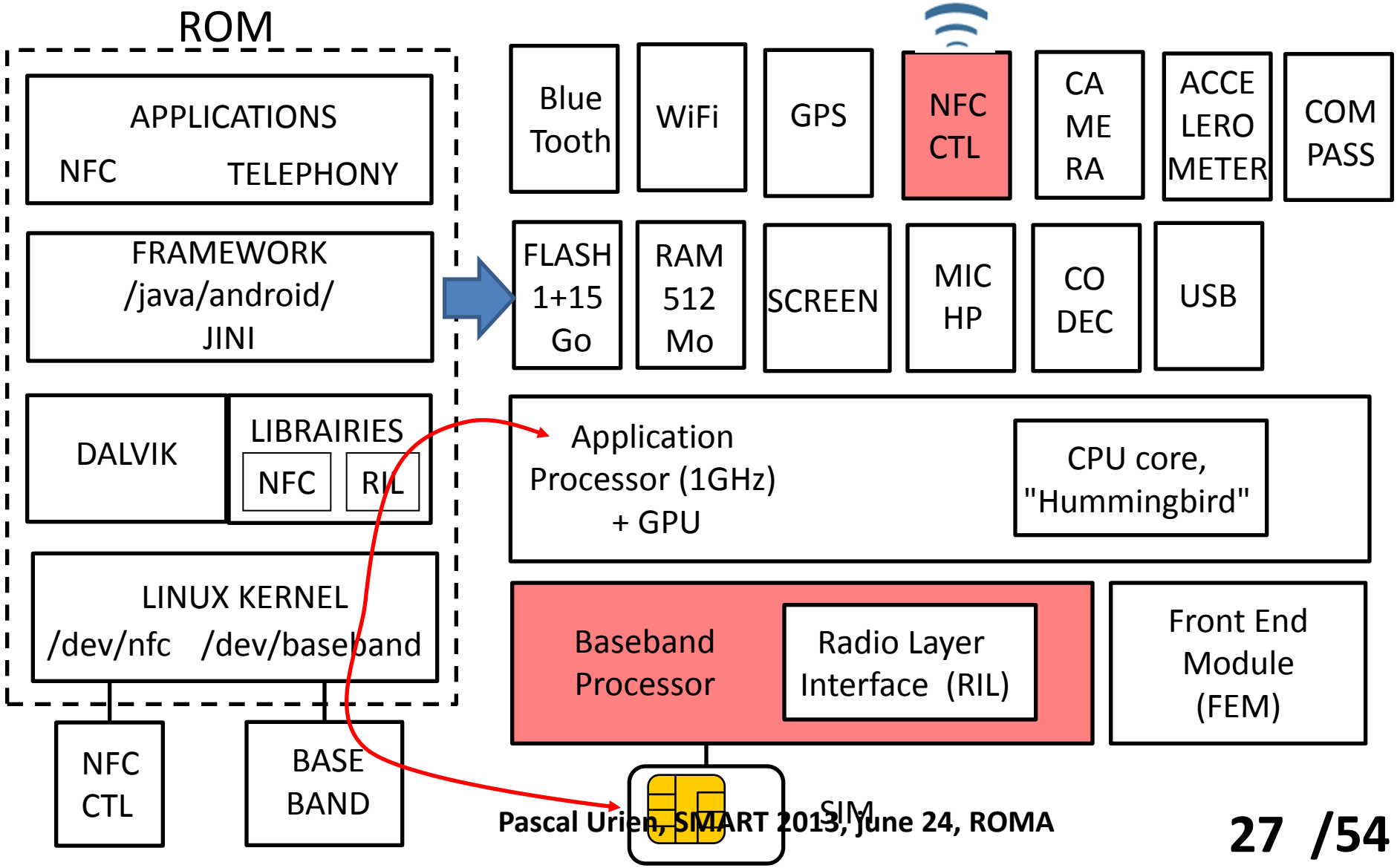
NFC In Mobile Phones

NFC and Smartphones

- Nokia
  - Card Emulation and SWP
  - NOKIA 6131
- Android 2.3 (Gingerbread), Android 4.0
  - Reader/Writer and P2P
  - Nexus S (v2.3) , Galaxy Nexus (v4.0), Galaxy S2(v2.3)
  - NXP NFC Controller PN65N
- RIM JDE 7.0.0, Blackberry 10
  - Reader/Writter and Card Emulation
  - JSR 177 (SIM Access)
  - Blackberry Bold 9900, 9930
  - INSIDE SecureRead NFC Controller
- IPHONE
  - External NFC Reader
  - Rumors for the NFC support



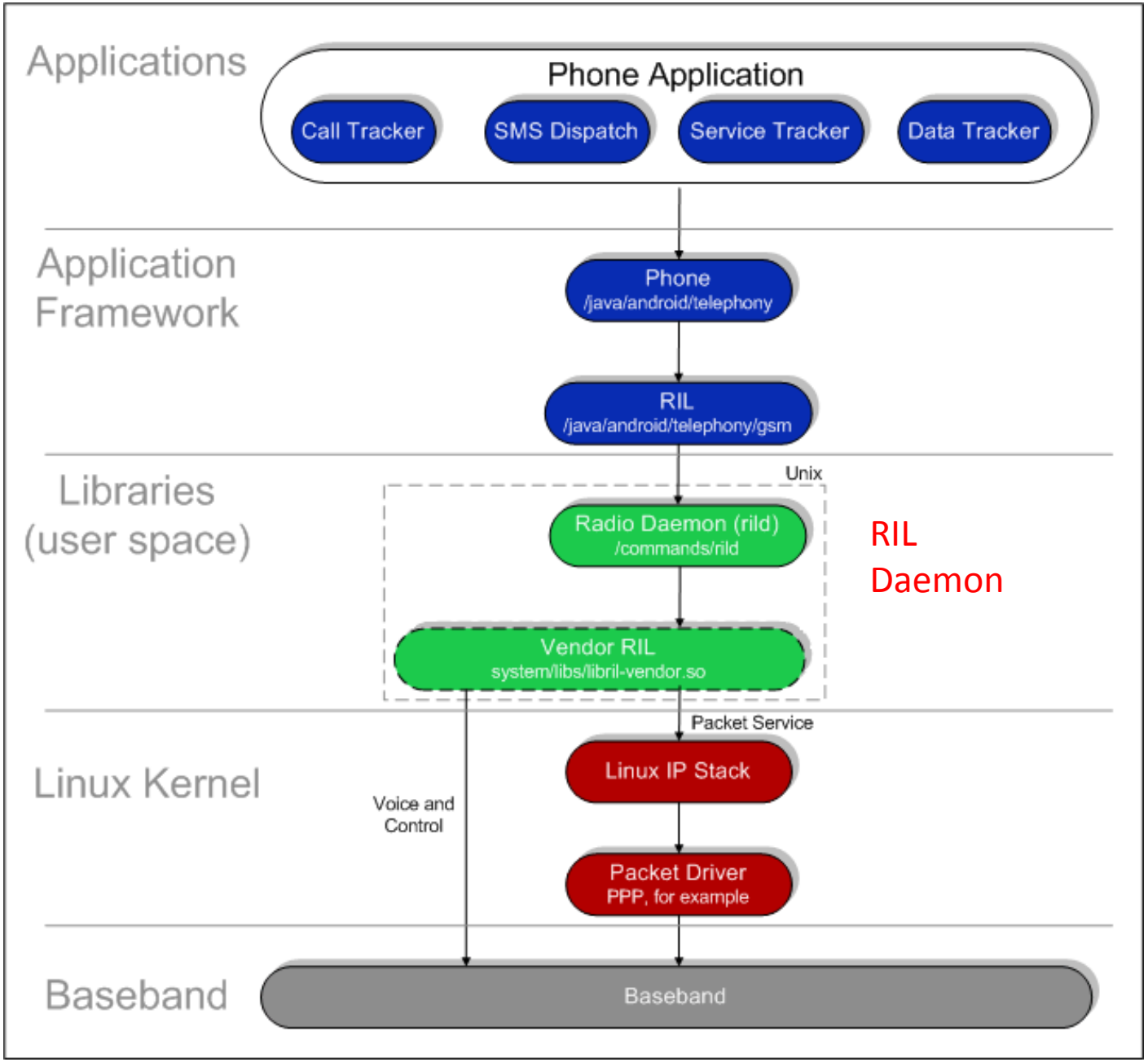
Hardware and Software Architecture of the Nexus S Android Phone



Proprietary Libraries

Hardware Component	Company
Orientation sensor	AKM
Wi-Fi, Bluetooth, GPS	Broadcom
Graphics	Imagination
NFC	NXP
GSM	Samsung

RIL Details

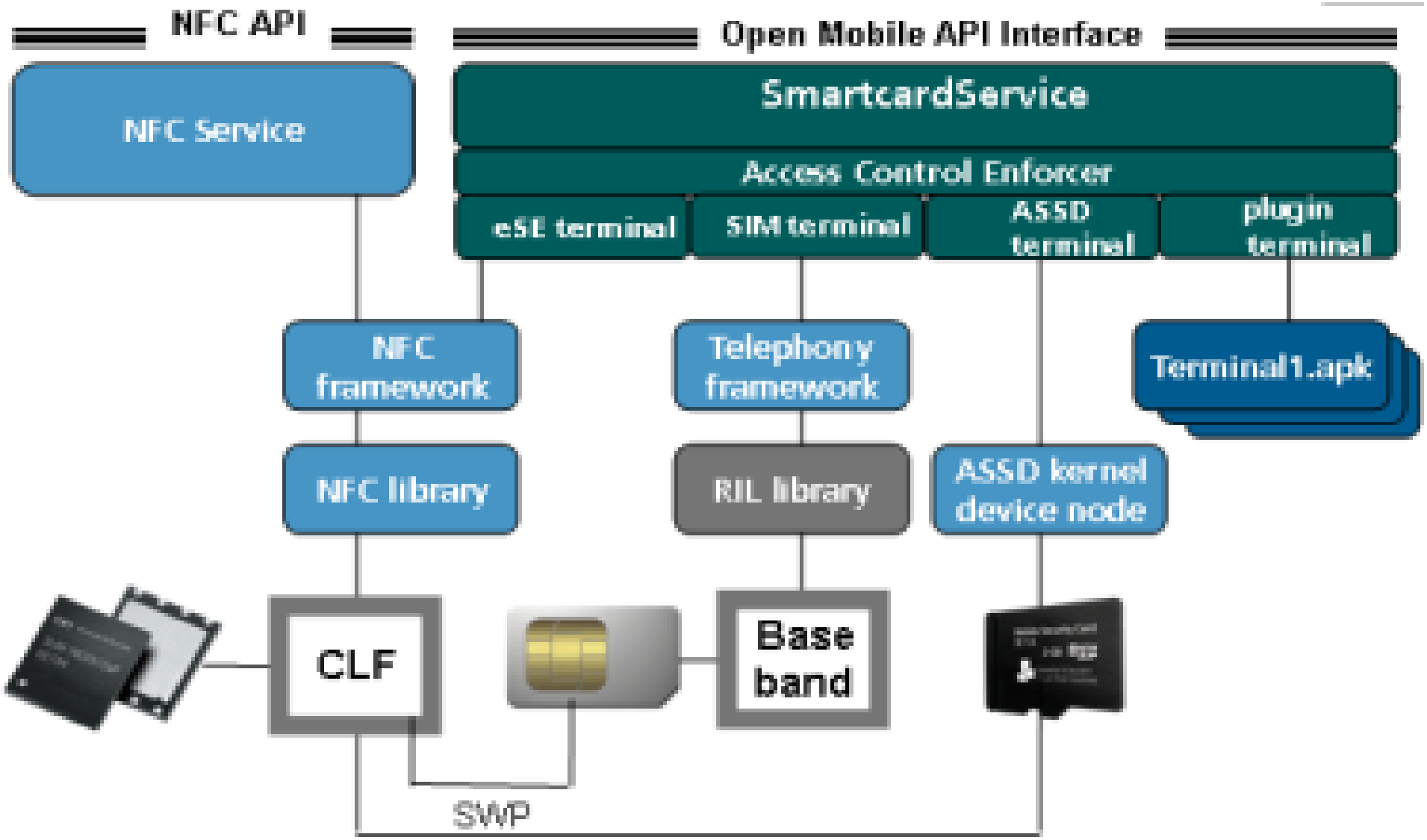


RIL Daemon

Telephony services (ingoing call, outgoing calls are managed through RIL packets

http://www.kandroid.org/online-pdk/guide/telephony.html

2011, Open Mobile API



Open Mobile API & Security Policy

- The API defines a generic framework for the access to Secure Elements in a mobile environment. It is based on four main objects.
  - The ***SEService*** is the abstract representation of all SEs that are available for applications running in the mobile phone.
  - The ***Reader*** is the logical interface with a Secure Element. It is an abstraction from electronics devices which are needed for contact (ISO 7816) and contactless (ISO 14443) smartcards.
  - The ***Session*** is opened and closed with a Reader. It establishes the logical path with the SE managed by the Reader.
  - The ***Channel*** is associated with an application running in the SE and identified by an ID (the AID= Application IDentifier)

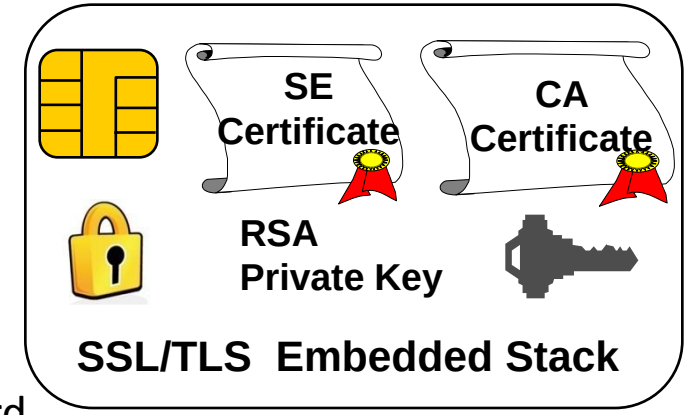
- In order to protect the USIM from a non-authorized Android application, an access control (AC) mechanism based on the PKCS#15 standard is used.
  - The PKCS#15 repertory (hosted by the SIM) contains three files defined for the access rules
  - The Access Control Main File (EF-ACMF) gives a reference to the Access Control Rules File (EF-ACRF)
  - The Access Control Rules File (EF-ACRF) stores a list of Access Control Conditions File (EF-ACCF), each of them being associated to a particular AID.
  - **Each Access Control Conditions File (EF-ACCF), contains a SHA1 digest of the mobile application whose access to embedded software running in the Secure Element (and identified by its AID) is authorized.**

Identity For NFC

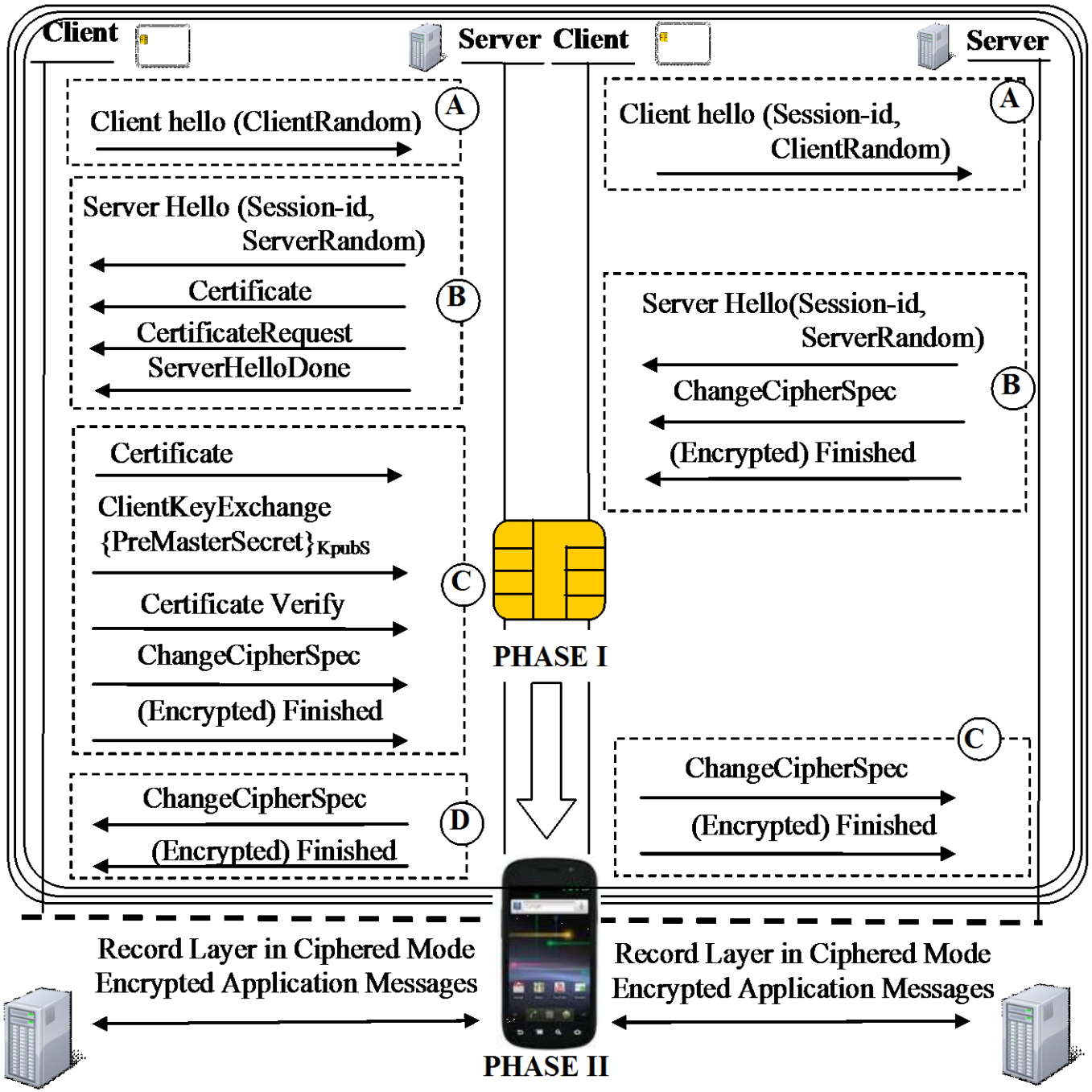
"On the Internet, nobody knows you're a dog."

A NFC Identity Model (NFC-ID)

- SSL/TLS is THE Internet security standard
- A tiny SSL/TLS STACK embedded in a Secure Element
  - Javacard 2.x
  - WORE! Write Once, Run Everywhere !
  - A small memory footprint
    - 20 KB for Client only mode
    - 25 KB for Client/Server mode
- A transport free from TCP/IP flavors
  - Datagram like transport
  - Specified by an IETF draft, draft-urien-eap-smartcard
- Each client has a Certificate
  - Each Secure Element has an Identity (its X509 Certificate)
  - **Strong mutual authentication, between SE and remote server**
  - Establishment of Secure Channels
  - Optional transfer of SSL/TLS session from SE to terminals (i.e. mobile)



Urien, P., "An OpenID Provider based on SSL Smart Cards", IEEE CCNC 2010.
Urien, P., "Convergent Identity: Seamless OPENID services for 3G dongles using SSL enabled USIM smart cards", IEEE CCNC 2011
Urien, P. et All, "A breakthrough for prepaid payment: end to end token exchange and management using secure SSL channels created by EAP-TLS smart cards", IEEE CTS 2011
Urien, P. et All, "A new keying system for RFID lock based on SSL dual interface NFC chips and android mobiles", IEEE CCNC 2012



NFC SSL/TLS Exchange Full Mode and Resume Mode

The NFC-ID Lifecycle

Container: Information protected (i.e. ciphered) by the SE public key, and signed by a trusted entity

Secure Element Certificate

End of Life

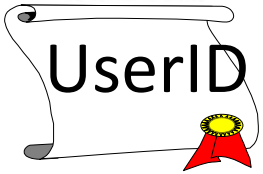
User Agent

Application Server*

SSL/TLS

SSL/TLS

IDentity Provider (IdP)



Application Certificate

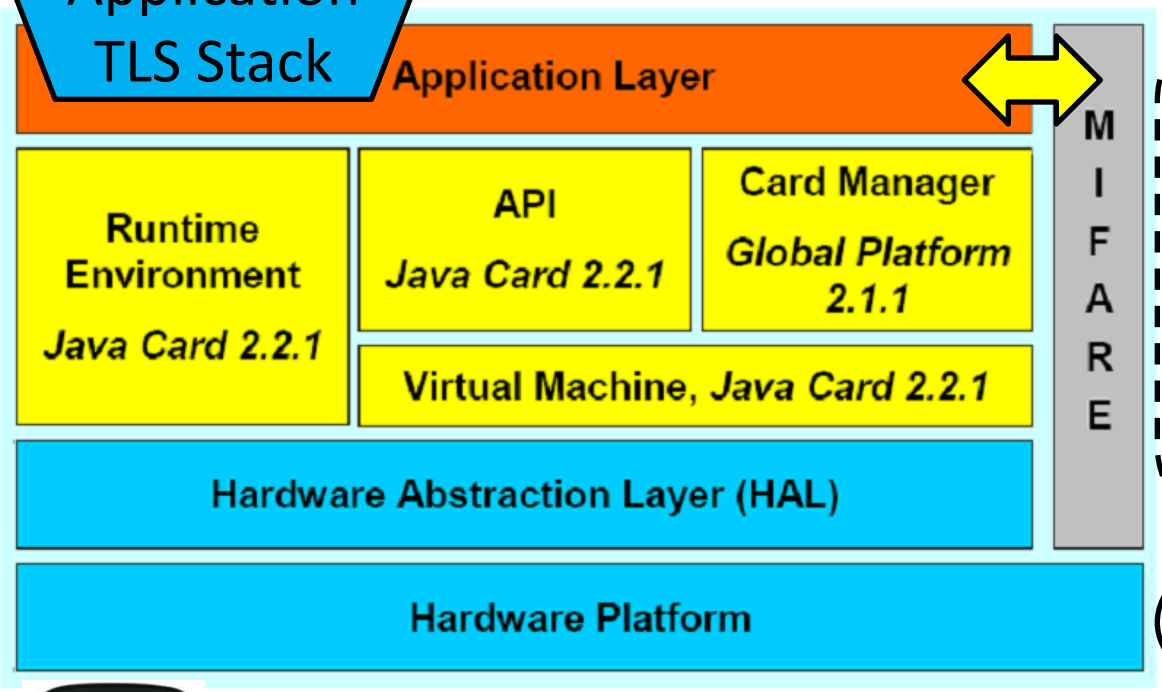
*http://www.morpho.com/IMG/pdf/morpho_telecoms_simply_me_2p_gb.pdf

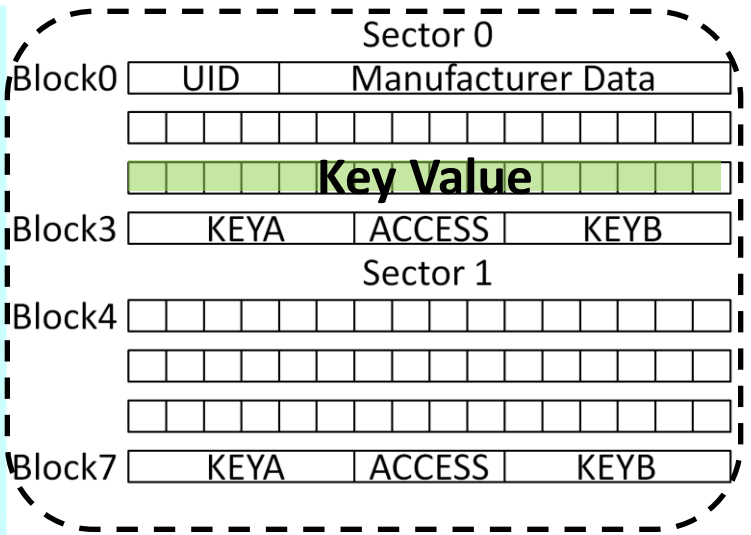
Use Case 1 : NFC Keys for the Internet of Things

Dual Interface NFC Device : 3 components

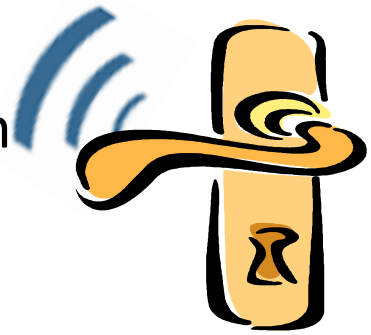
JavaCard Key Application TLS Stack

3 MIFARE API





2 Mifare Emulation



1 Secure Microcontroller with NFC resources equipped with a Java Virtual Machine

NFC Mobile

Legacy Mifare Lock

The CES 2012 Demonstration (30s)

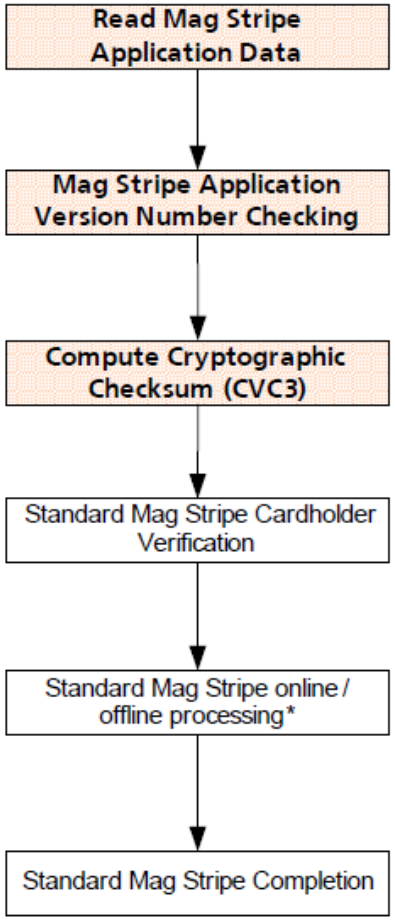
http://www.facebook.com/photo.php?v=3030751173533

CES 2012, Las Vegas

Use Case 2
The Emergence of the Cloud Of Secure Elements (CoSE)

About NFC Payments

- Some NFC payments are based on the MasterCard PayPass specification
- There is two modes
  - Mag Stripe, a four digits CVC3 (*Card Verification Value*) is computed from a 3xDES and various parameters (PAN, ATC counter,…)
  - Contactless EMV
- The Secure Element securely performs calculations or runs the EMV application
- Contactless payments introduce a new paradigm, the virtualization of the bank card.
- The merchant terminal doesn't known where is running the payment application on the mobile side.



Where is running the SE application ?

* MasterCard® *PayPass™, M/Chip, Acquirer Implementation Requirements,* v.1-A4 6/06

Some Details with EMV Mag. Stripe*

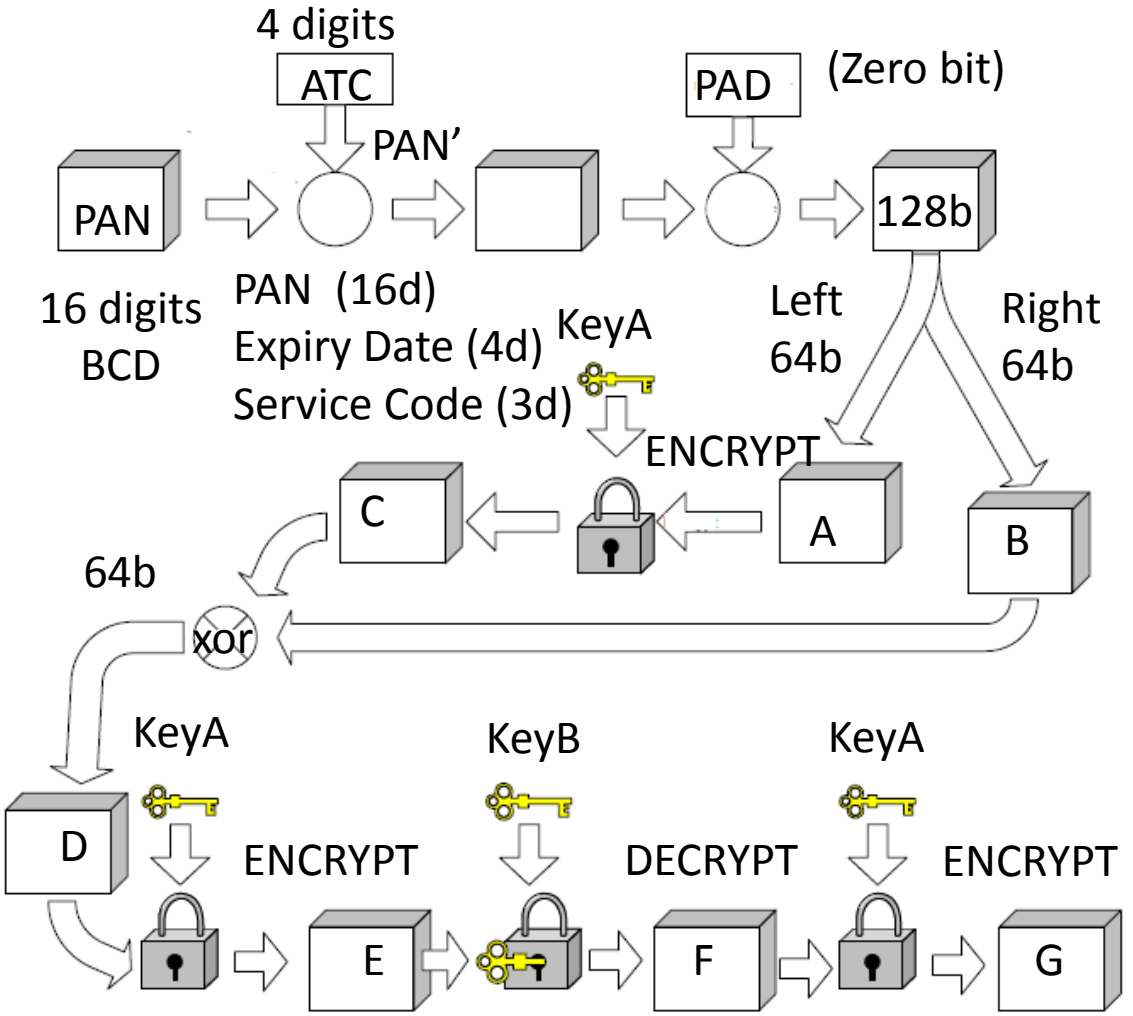
SELECT 2PAY.SYS.DDF01

SELECT MasterCard Google Prepaid Card

GET PROCESSING OPTIONS

READ first record

COMPUTE Cryptographic Checksum

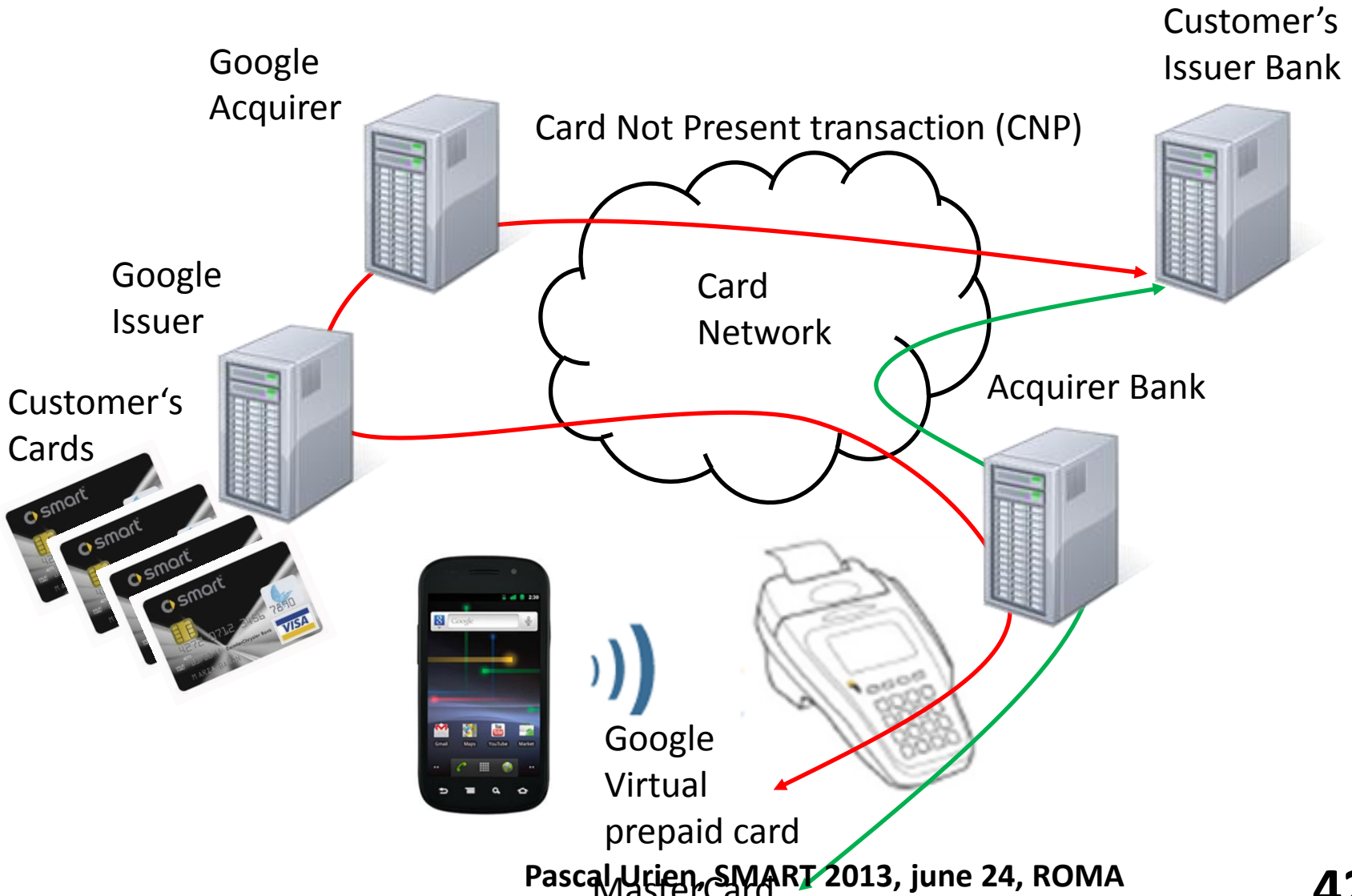


G=A23FB35FC89AE3A9
23358939AFBFCAEA
23358939**05**152040

CVC3=233

*Visa Contactless Payment Specification Version 2.0.2 July 2006

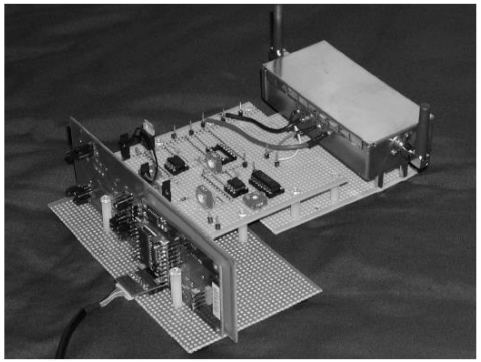
Google Wallet 2



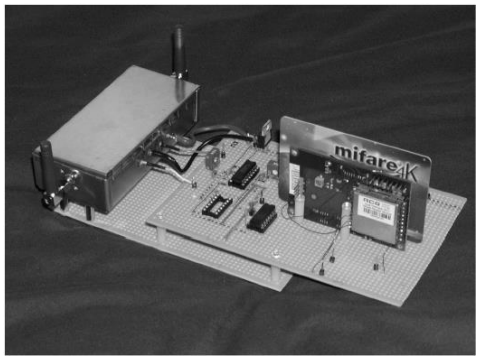
About Relay Attack

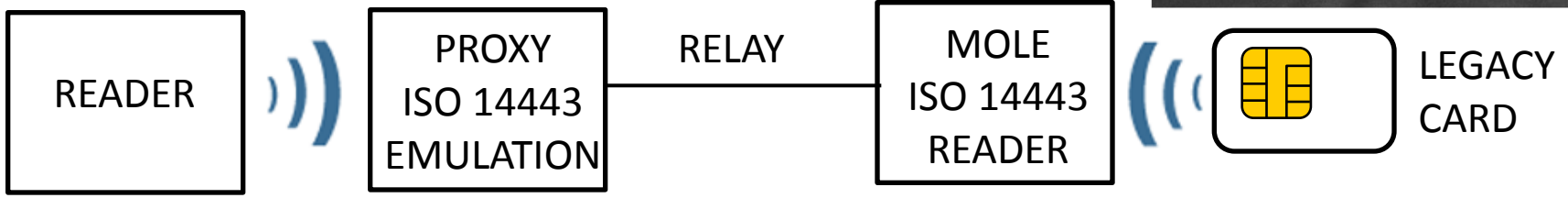
- In 2005, G.Hancke introduced the concept of the “Relay Attack”
- The basic idea is that a reader working with ISO14443 device, reads a fake card (the proxy) which is connected via radio to an other device (the mole) working with a legacy card .
- As a result the reader manages a session with a remote device

proxy



mole





“A Practical Relay Attack on ISO 14443 Proximity Cards” Gerhard Hancke, 2005

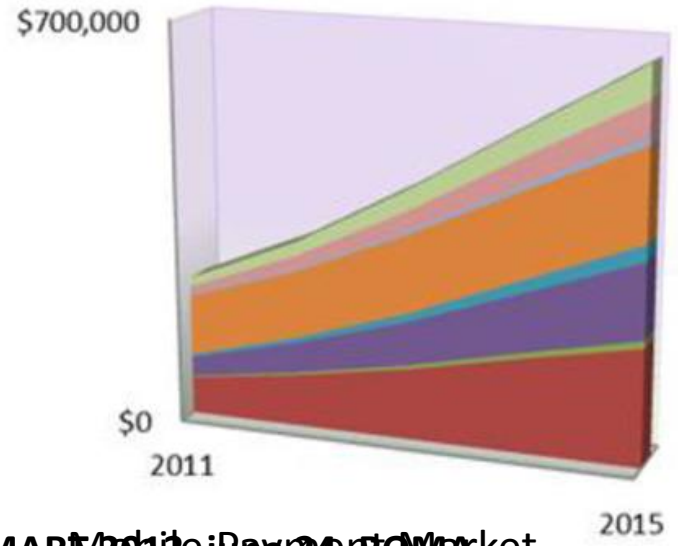
“Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks”, Saar Drimer and Steven J. Murdoch, 2007

Where is located a bank card ?

- In the SIM/USIM
  - Use SWP for NFC communication
  - MNO model
- In a NFC SecureSD
  - Tyfone, DeviceFidelity
- In a NFC Controller
  - The Google Model
- Somewhere in the Cloud
  - GoogleWallet2
  - SimplyTapp
  - EtherTrust

2012 Google fees	
Less than $3,000	2.9% + $0.30
$3,000 - $9,999.99	2.5% + $0.30
$10,000 - $99,999.99	2.2% + $0.30
$100,000 or more	1.9% + $0.30

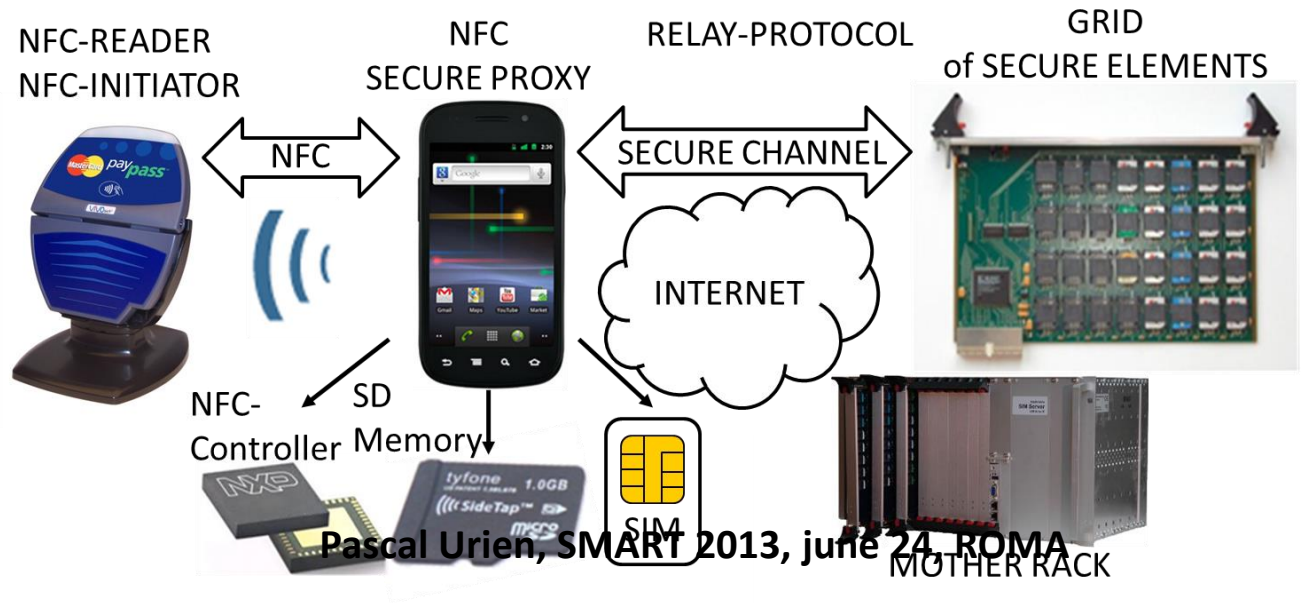
2017 Forecasts
-Mobile Payments 1300 billions $
-NFC payments 200 billions $



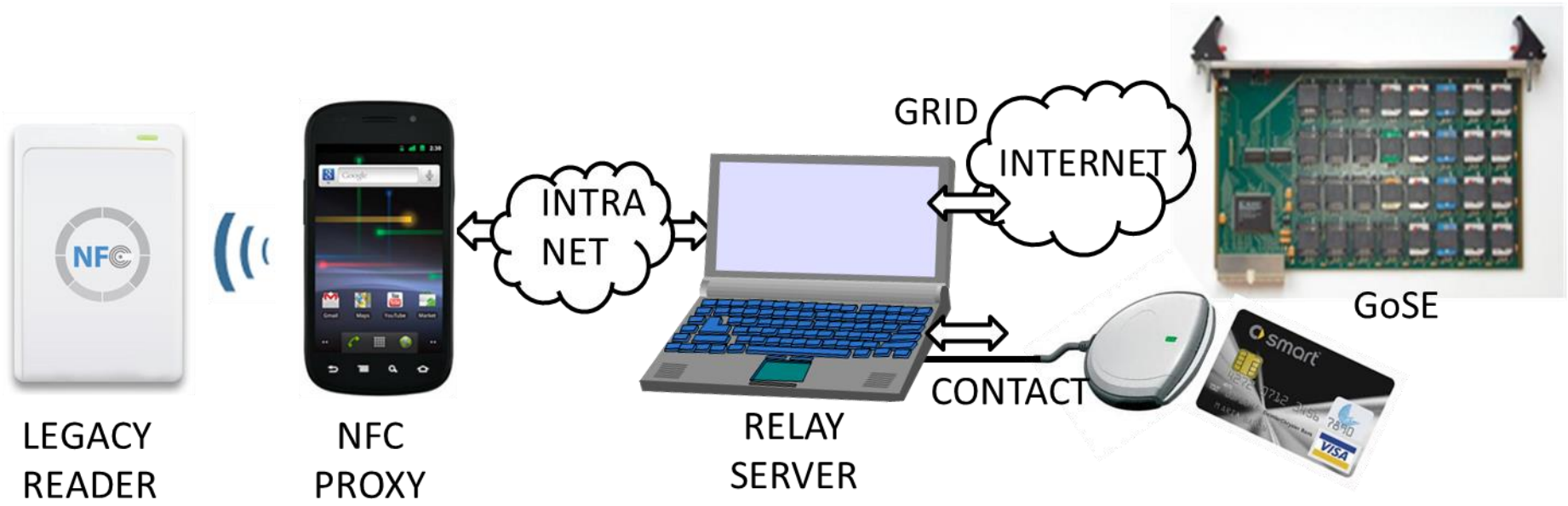
Mobile Payment Market

Cloud Of Secure Elements

- A cloud of secure elements (CSE) comprises the following five elements
  - Applications (typically written in javacard) stored in secure elements.
  - Grids of secure elements (GoSE). Secure Elements embed Issuer Security Domain, which manage the lifecycle of applications. Applications may move from a grid to another.
  - A Relay-Protocol (RP) enforces security between the GoSE and the NFC proxy, thanks to a secure channel, such as TLS.
  - The NFC Secure Proxy (NSP) controls the session with the NFC reader (or initiator) and the dialog with the GoSE according to the relay protocol. This software entity should manage a SE located in the smartphone.
  - A NFC reader (or NFC initiator) is used by legacy applications (payment, transport,…); however future services could work with the P2P mode.



Experimental Platform



Request	Data Size	Con-tact	Re-lay	Re-play	Relay -Replay -Contact
1 Select AID	7	47	221	67	60
2 Read Result	41	47			
3 Read Record 1,File 1	7	31	150	20	36
4 Read record 1, File 1	32	63			
5 Read Record 2, File 1	7	47	190	20	45
6 Read Record 2, File 1	36	78			
7 Read Record 3, File 1	7	47	121	20	54
8 Select AID	7	47	210	40	45
9 Read result	60	78			
10 Get Processing Options	7	187	330	40	72
11 Read Result	23	31			
12 Read Record 1, File 1	7	16	120	40	2
13 Read Record 1, File 1	30	62			
14 Read Record 2, File 1	7	31	220	70	25
15 Read Record 2, File 1	65	94			
16 Read Record 1, File 2	7	31	368	50	85
17 Read Record 1, File 2	157	202			
18 Read Record 2, File 2	7	31	240	30	54
19 Read Record 2, File 2	93	125			
20 Read Record 3, File 2	7	32	411	80	80
21 Read Record 3, File 2	157	219			
22 Read Record 4, File 2	7	31	370	80	41
23 Read Record 4, File 2	158	218			
24 Read Record 1, File 3	7	15	240	50	35
25 Read Record 1, File 3	95	140			
26 Read Record 2, File 3	7	31	210	20	66
27 Read Record 2, File 3	60	93			
28 Read Record 3, File 3	7	31	130	20	32
29 Read Record 3, File 3	17	47			
30 DDA	7	437	820	60	151
31 Read Result	138	172			
32 Generate AC	7	421	560	62	31
33 Read Result	39	46			
Total	1320	3228	4911	769	Mean 51

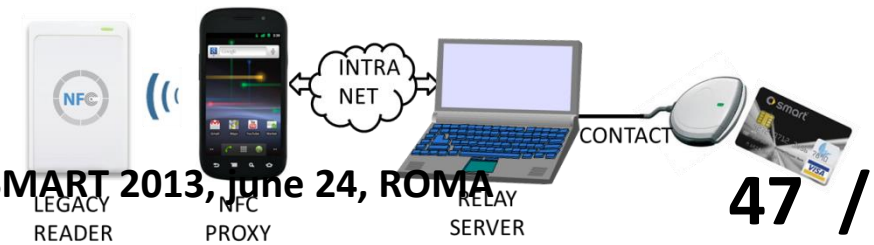
EMV transaction timings, Intranet

Network Cost

<Tcost = Trelay – Tcontact – Treplay> = 51ms

Cache Operation

769 – 60 – 62 + 820 + 560 = 2027 ms

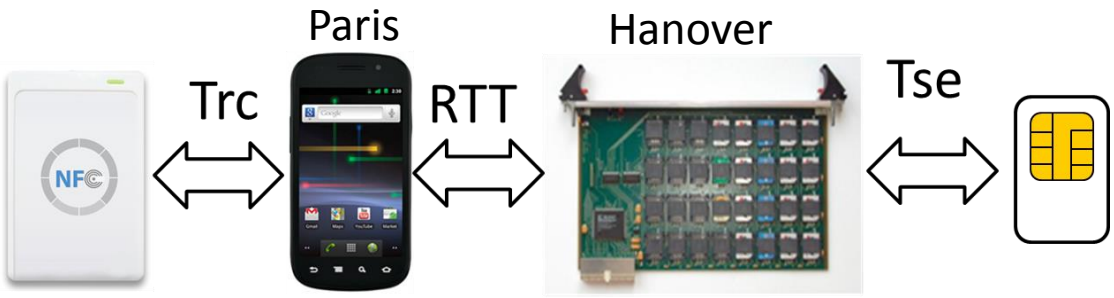


Test with EMV magstripe profile

One TCP packet per ISO7816 command (APDU),
RTT 70 ms (Paris – Hanover)

Network Cost (Internet)
<Tcost = Trelay – Tgrid – Treplay>= 83 ms

Request	Data Size	Con-tact	Grid	Relay Grid	Re-play	Grid -Replay -Relay
1 Select AID	21	16	94	219	62	63
2 Read Result	53	0	141	265	47	77
3 Select AID	23	15	109	219	63	47
4 Read Result	41	0	109	296	78	109
5 Read Record 1, File 1	7	16	78	142	47	17
6 Read Record 1, File 1	115	16	203	531	141	187
Total	260	63	734	1672	438	Mean 83



Trc = Tr0 + L x D
where Tr0 is a fix delay (about 20-40 ms), L is the length of exchanged data, D is the NFC throughput (104 Kbits/s in our platform, i.e. D= 0,1ms/byte)

Trr= Trc + RTT + Tse,
Where Trc is the time consumed by the NFC proxy , RTT is the round trip time over internet (ranging between 50ms to 100ms), and Tse is a time consumed by a legacy secure element for the request (such as 440ms for DDA or 420ms for GenerateAC).

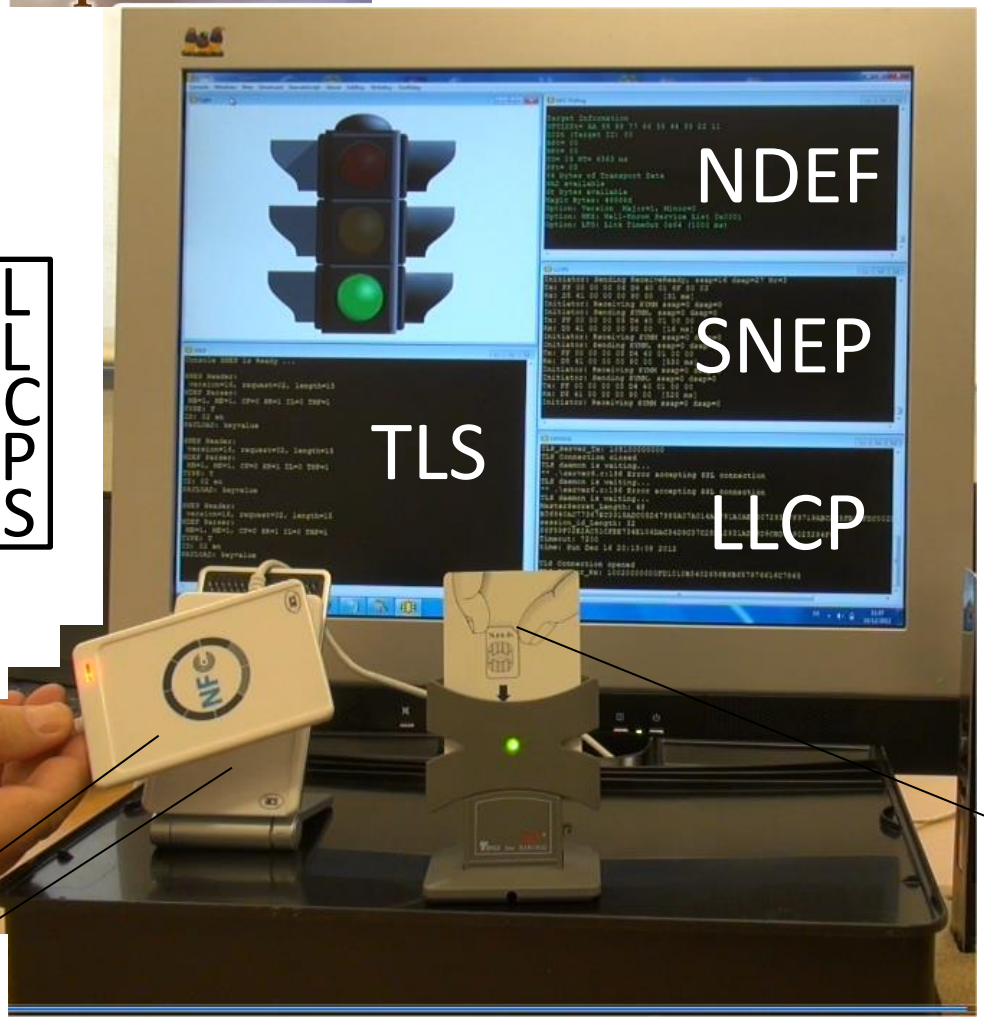
Use Case 3
Security for NFC
LLCPS
urn:nfc:sn:tls:snep

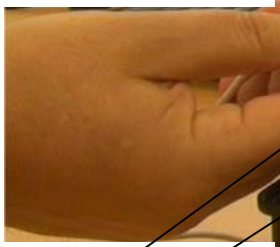
2012 LLCPS Platform

OpenSSL

NDEF
SNEP
TLS
LLCP
NFCIP-1

LLCPS





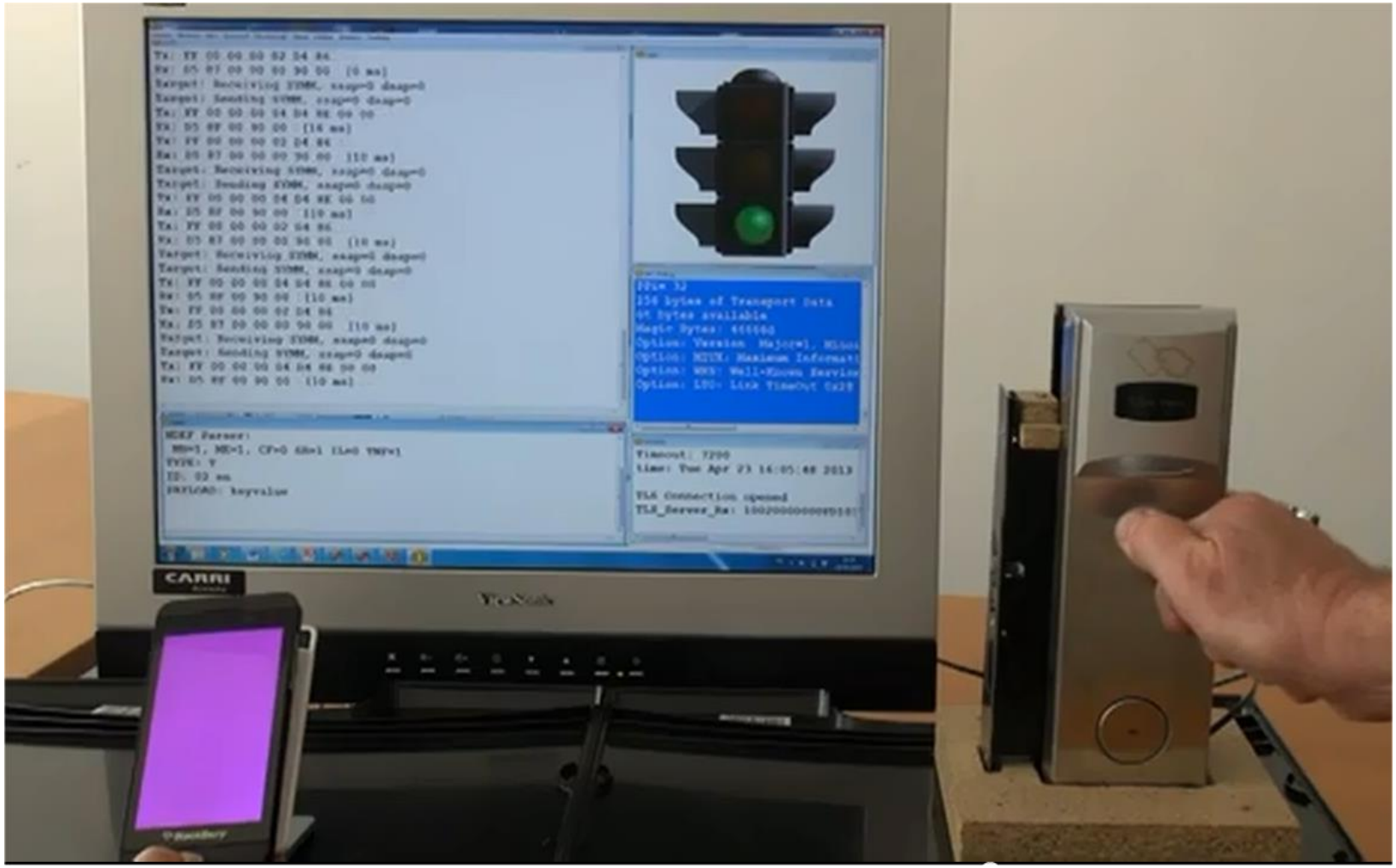
NFC-Controller(s)

TLS

SIM

www.youtube.com/watch?v=CvJvHxOi3eU

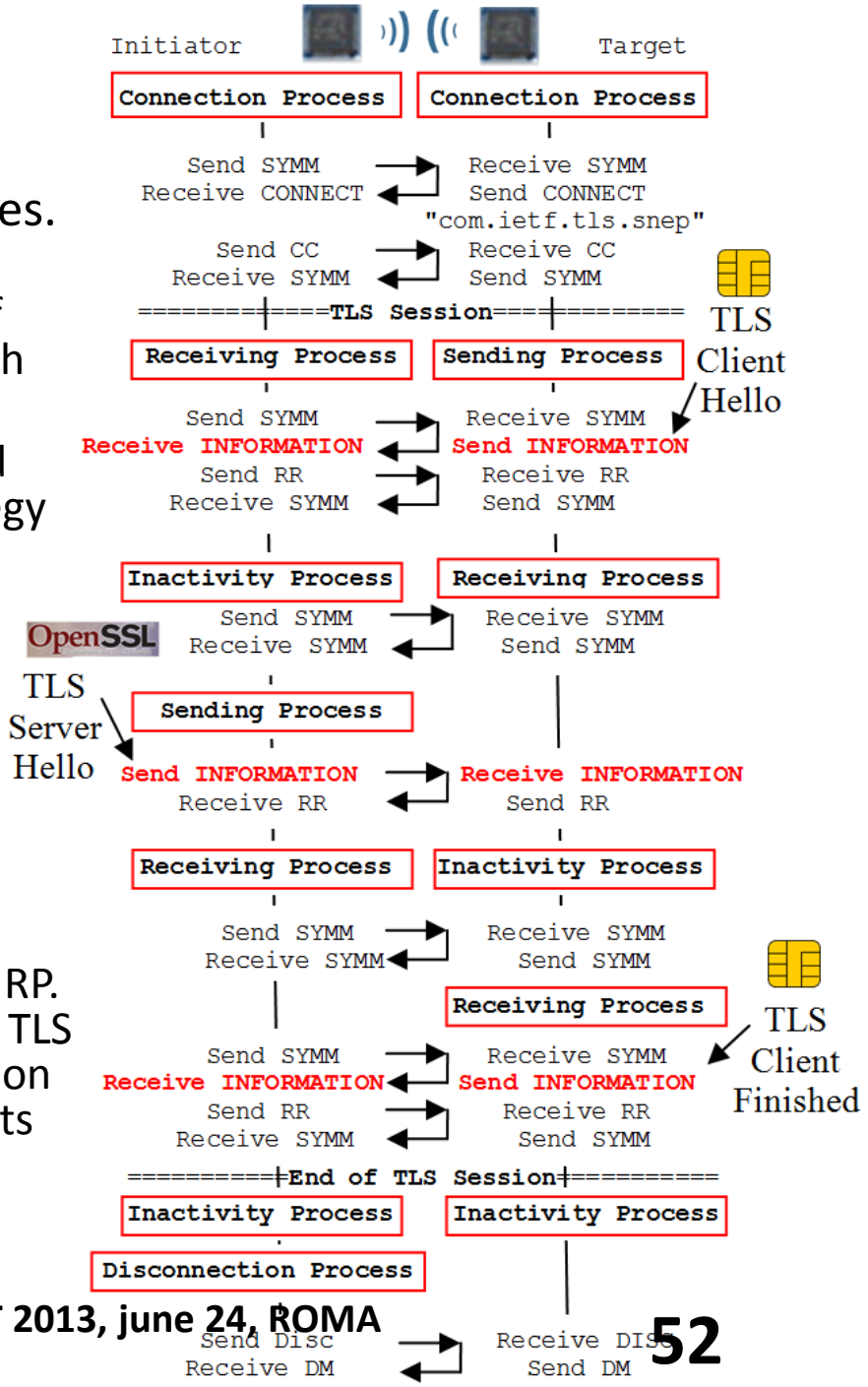
2013, First Tests With BB 10



http://www.youtube.com/watch?v=CW341clZylw

LLCPS

- The LLCPS layer manages five exclusive processes in order to exchange TLS messages.
  - **The connection process (CP)** and **the disconnection process (DP)** are in charge of establishing and releasing LLCP sessions with the "com.ietf.tls.snep" service.
  - **The sending process (SP)** sends a requested amount of data according to a simple strategy that performs segmentation, transmits INFORMATION PDUs and waits for acknowledgments (RR).
  - **The receiving process (RP)** waits for a requested amount of incoming data; the reception of each incoming INFORMATION packet is acknowledged by a RR PDU.
  - **The inactivity process (IP)** periodically generates SYMM symbols which may be echoed by other processes such as IP, SP or RP. SYMM generation is a consequence of slow TLS processing by a secure element, or interaction between the mobile operating system and its user



SNEP - Simple NDEF Exchange Protocol & NDEF - NFC Data Exchange Format

- Once a TLS session is established, SNEP packets are securely exchanged.
- In our demonstration we use only SNEP-Put and SNEP-Success packets.
- A value, i.e. as a key encoded according to the NDEF format, is pushed from the target to the virtual lock

SNEP Put Packet

10 SNEP Version
02 Put
00 00 00 0E Payload Length

NDEF Record :
(NFC Text Record Type Definition)

D1: 1 1 0 1 0 001
01: Type Length
0A: Payload Length
54: Type= 'T', Text
02: ID= UTF8
65 6E: "EN"
53 61 6D 70 6C 65 20: "Sample "

Questions