nascent
Secure Technologies

20 anos

Cryptographic Hardware and Secure Elements

A security architect’s view

www.nascent.com.br

Secure Application Development

Sean Michael Wykes

sean.wykes@nascent.com.br

- **British** born and educated, living in Brazil since 1997
- **Masters Degree (’92) in Information Engineering** from Southampton University
- **20+ years experience** in design and development of systems and secure applications, based on technologies such as smart-cards and secure elements.
- Author of **“Criptografia Essencial - a Jornada do Criptógrafo”** – Elsevier 2016.

Smart Card Technology

Origins and Evolution

Origins

1968 Patent – Plastic cards with microchips

Jürgen Dethloff / Helmut Gröttrup

1974 Patent on chip Cards

Roland Moreno – ‘Father’ of smart-cards

1978 Patent on ”Self Programmable one-chip MPU”

Michel Ugon - Bull

1979 First Production Cards

Bull, Motorola

Note: Bibliography diverges on exact details and attribution

Evolution

~40 years

~1980 **Memory Cards**

Hardwired logic, EPROM, E^2PROM memory

Microprocessor Cards

Secure MCU, Crypto HW, Monolithic OS

Multi-Application Cards

VM + Applets, App-Firewalls, Remote App Management

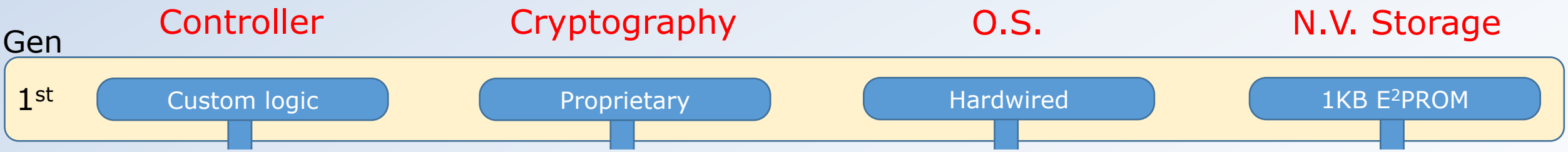
Near Field Communication & Secure Elements

NFC, Mobile (SE) and Cloud (HCE)

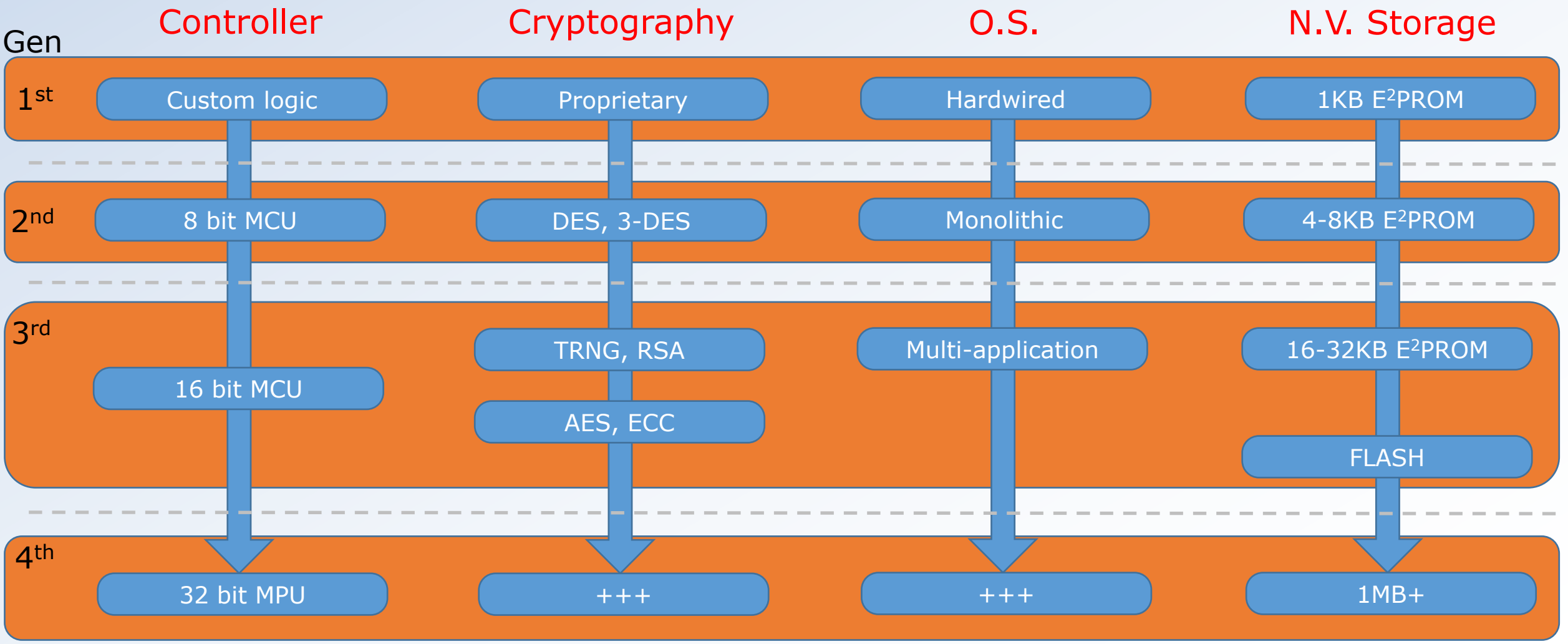
2018+ ***IoT & Consumer***

Embedded SE, integrated, emerging technologies …

Evolving HW and SW Features

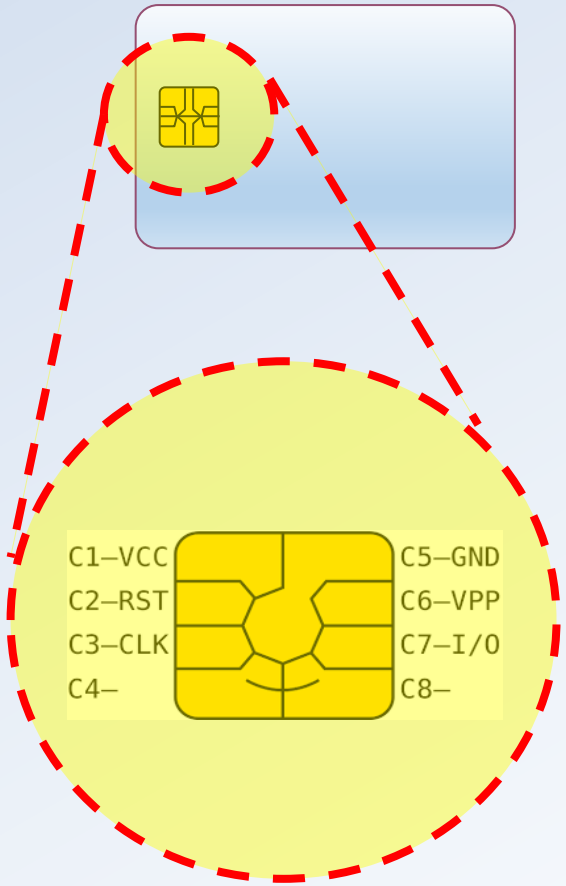


Evolving HW and SW Features

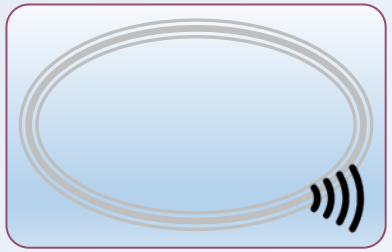


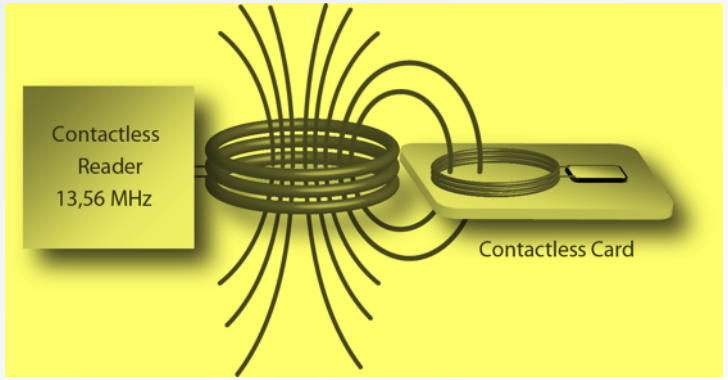
Communication Interfaces

Contact

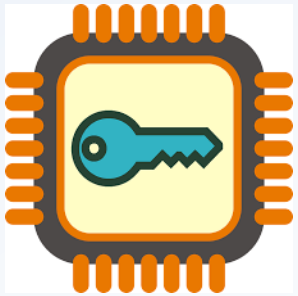


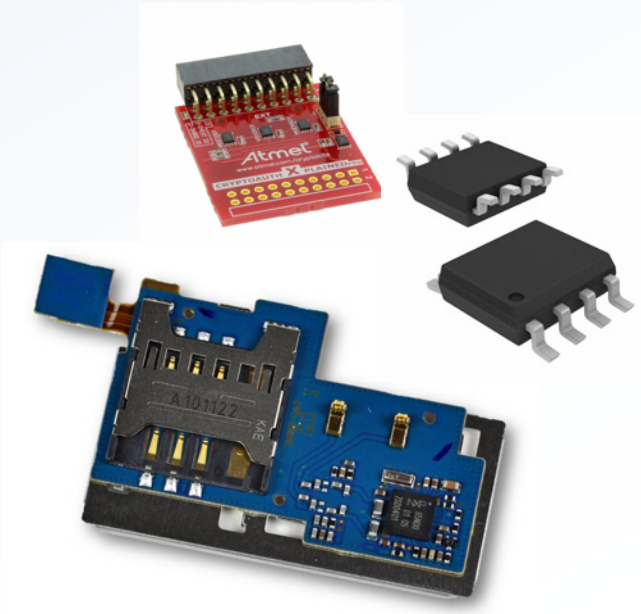
Contactless





Embedded



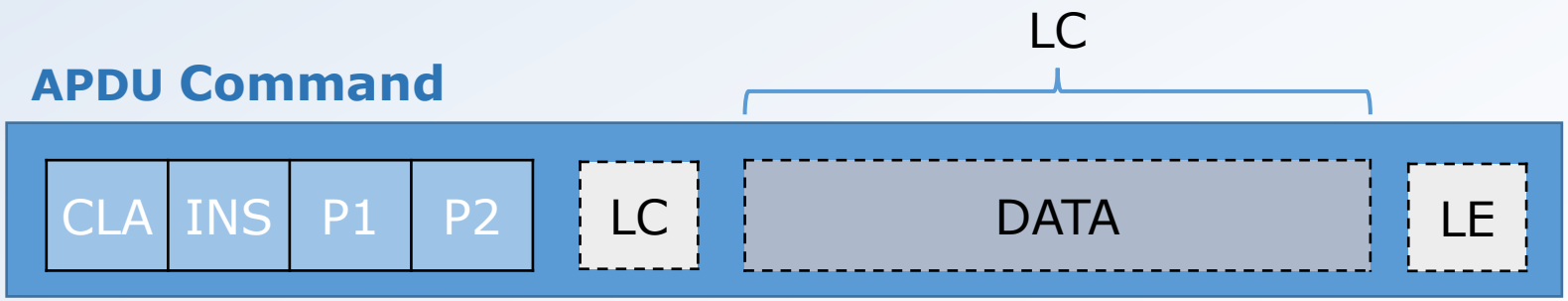


Communication Flow

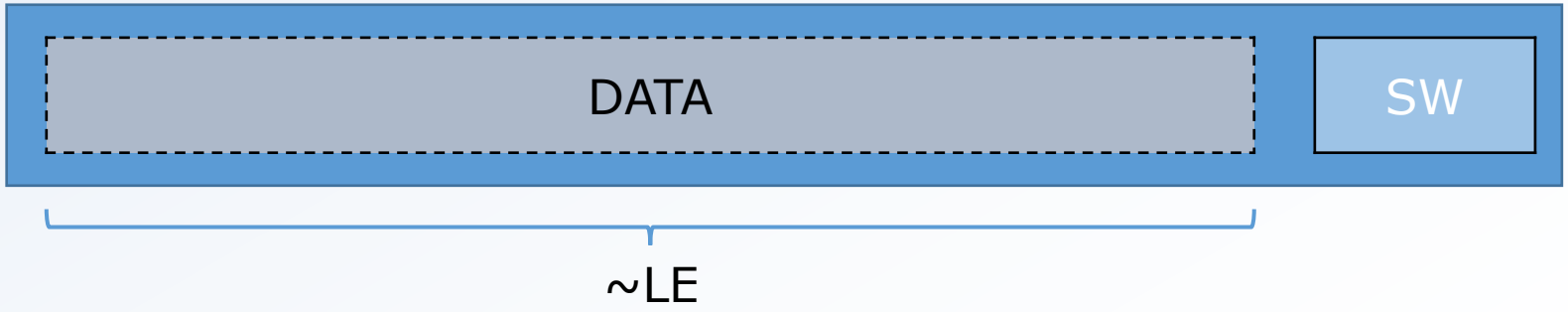
APDU Command

APDU Response

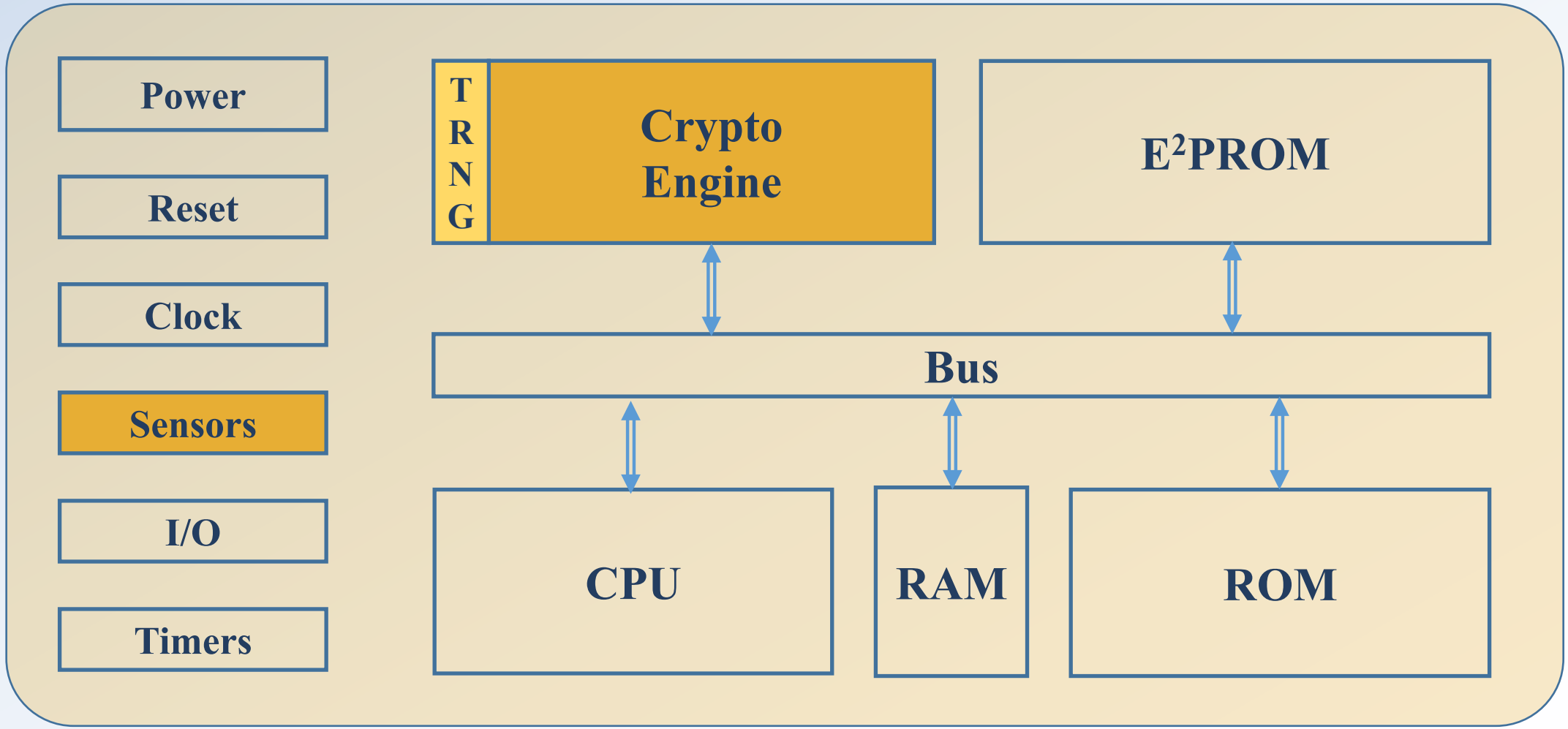
APDU Command



APDU Response



Typical Secure MCU Architecture



Tamper and Side-Channel Resistance

Inbuilt defenses against:

Physical (Invasive) Attacks

De-packaging, Micro-probing, rev-Engineering,
Scanning Electron Microscopy, …

Active and Environmental Attacks

Temperature, Voltage Glitches, Laser,
Clock, Reset, …

Passive Monitoring

Current Consumption,
Electromagnetic Emissions,
Timing

Case 1

Access Control System for Industrial Premises

Circa 1998

Project Brief

Project

Employee / Contractor ID and Access Control System

Location

Steelworks, in Volta Redonda (~6km^2)

Contactless ID Badges

Using MIFARE "Classic" cards (~40,000)

Robust

Hostile Industrial Environment
Fault-tolerant distributed system architecture

System Architecture

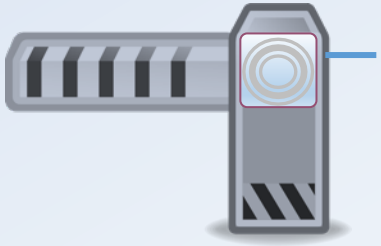
Turnstiles

RS485

Concentrators

TCP/IP

Host / DB



Barriers

Contactless ID Badge

Doors

Reception/HR

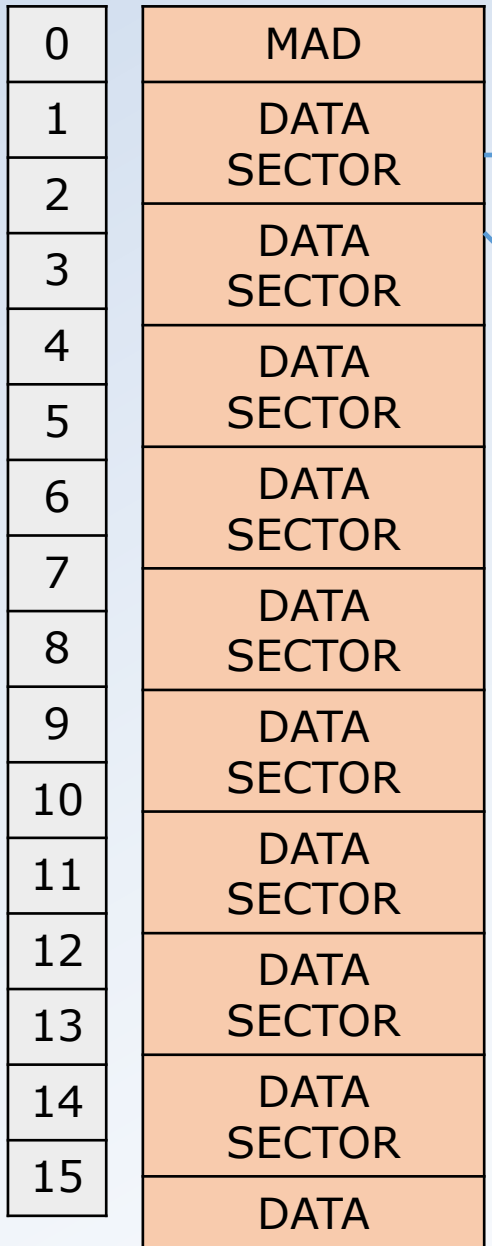
Contactless ID Badges

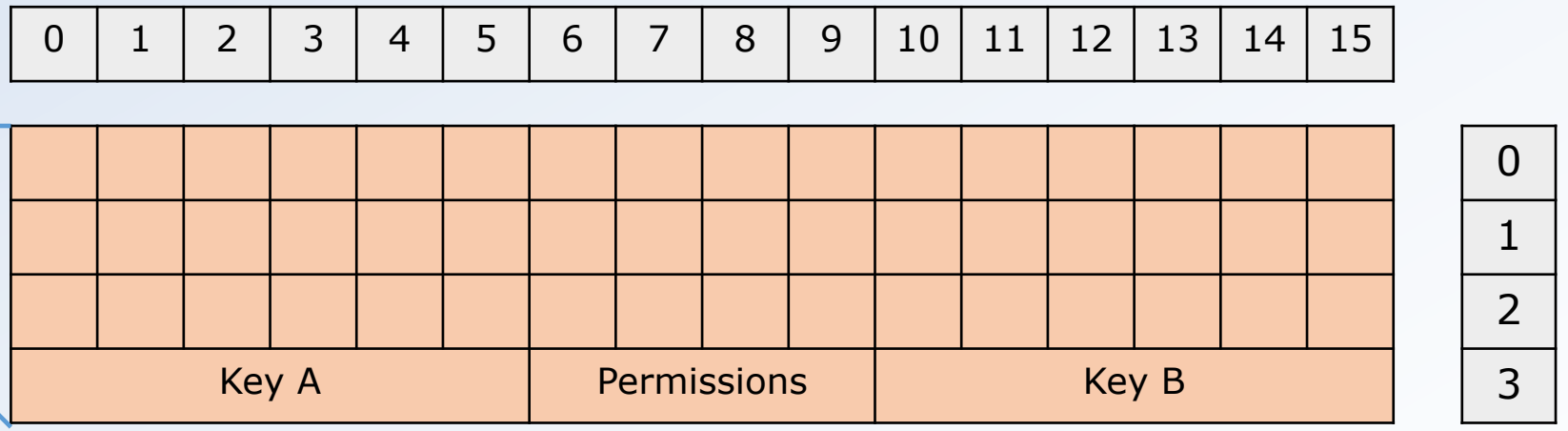
MIFARE Classic Contactless Smartcards

Features:

- ✓ *State of the Art* **in 1998**
- ✓ 1KB Capacity, divided in sectors and blocks
- ✓ Cryptographic authentication, based on proprietary CRYPTO-1 cipher (48 bits)
- ✓ On-chip Unique Identifier

MIFARE Classic Layout





- 1KB Capacity : 16 **sectors** x 64 bytes
- 1 sector = 4 **blocks** of 16-bytes
- Two keys (A/B) per *sector*
- Access permissions, per block

Data Block ⟵ { Read / Decrement, Write / Increment } ⟶ **Value** Block

MIFARE Classic Security Properties

Contactless ID Badge

Contactless Authentication Protocol

Contactless Reader

Cryptographic Keys

Cryptographic Keys

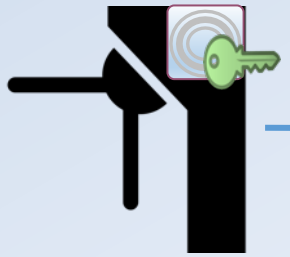
Unique Serial Number

Access Permissions

Confidentiality
Data Integrity
Entity Authentication

TRUST

Key Generation, Storage and Provisioning



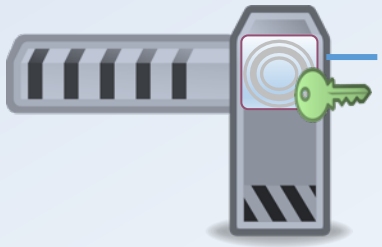
Turnstiles

RS485

Concentrators

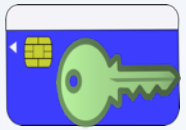
TCP/IP

Host / DB



Barriers

Key Master



Doors

Contactless ID Badges

Reception/HR

In 1998,

This Commercial System's Security level was considered

OPTIMUM

Given the Available Technology

Case 2

Contactless Ticketing System

Circa 2008

Project Brief

Project

Electronic Ticketing System for inter-municipal busses

Localization

Southern Brazil

Contactless Tickets

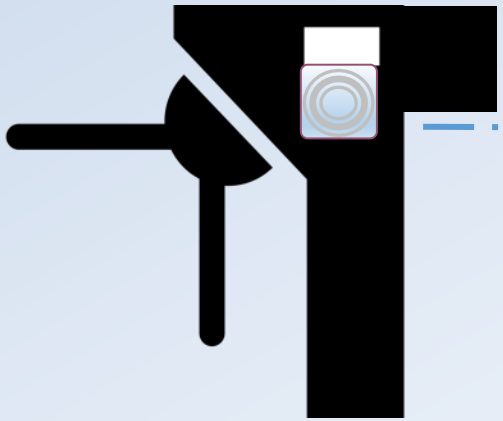
Using same MIFARE ”Classic” cards

Security Requirements

Anti-fraud, anti-cloning protection

Fault-tolerant distributed system architecture

System Architecture



Turnstiles

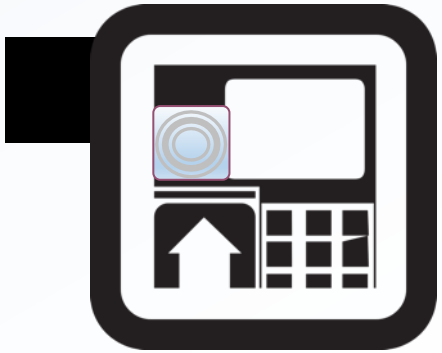
WIFI

Bus Garage

TCP/IP

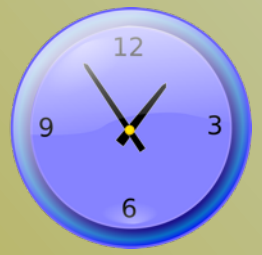
Host / DB

Contactless Ticket



Ticket Machine

In 2008,
Can we use a Security Solution from 1998



SECURITY
WEAKENS over time

We LEARN to build
BETTER SECURITY

Security Evaluation of 1998 Solution

❌ ***SYSTEM-WIDE KEY*** – in all cards and readers

- Card Data/Permissions may be ALTERED

❌ Newer Cards have ***Programmable ID***

- Transaction Logs may be ADULTERATED

❌ *Proprietary **CRYPTO-1** Cipher rev. Engineered*

- Cards may be CLONED or "*RESTORED*"

How can Security be

IMPROVED

Using the Same Low-Cost Cards

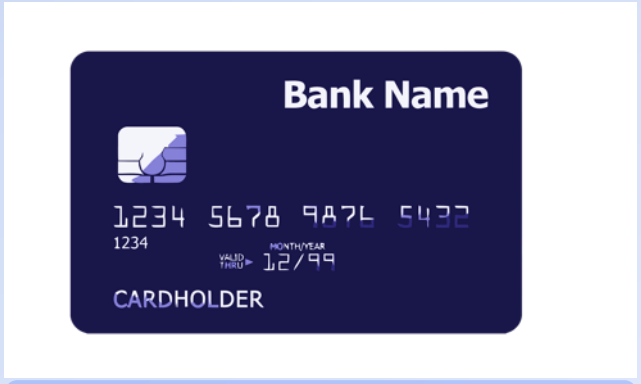
Introducing the Secure Element

Eurosmart:

"A **SECURE ELEMENT** contains a **certified microcontroller** and **embedded software**. It is secure, personal and portable and comes in **multiple form factors** : smart card, USB token, microSD, etc. …*{snip}*… Secure elements have a **strategic role** in **protecting digital identities** and are **vital to ensure digital security and privacy**."

Many different Form Factors

SIM Card

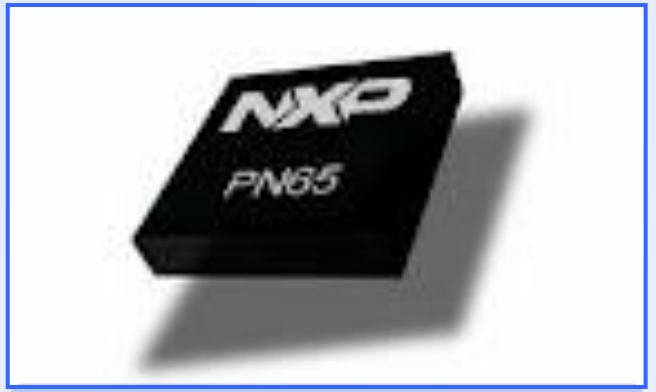


Smart Card

Secure uSD

USB Tokens

Integrated SoM



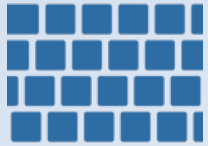
Embedded SE

Another Definition

Global Platform:

"A **Secure Element** (SE) is a **tamper-resistant** platform, typically a one chip **secure microcontroller**, capable of **securely** hosting **applications** and their **confidential and cryptographic data** (e.g. key management) in accordance with the rules and **security requirements** set forth by a set of well-identified **trusted authorities**."

Characteristics of Secure Elements



Isolated Execution Sandboxes

Clearly Defined Access Controls

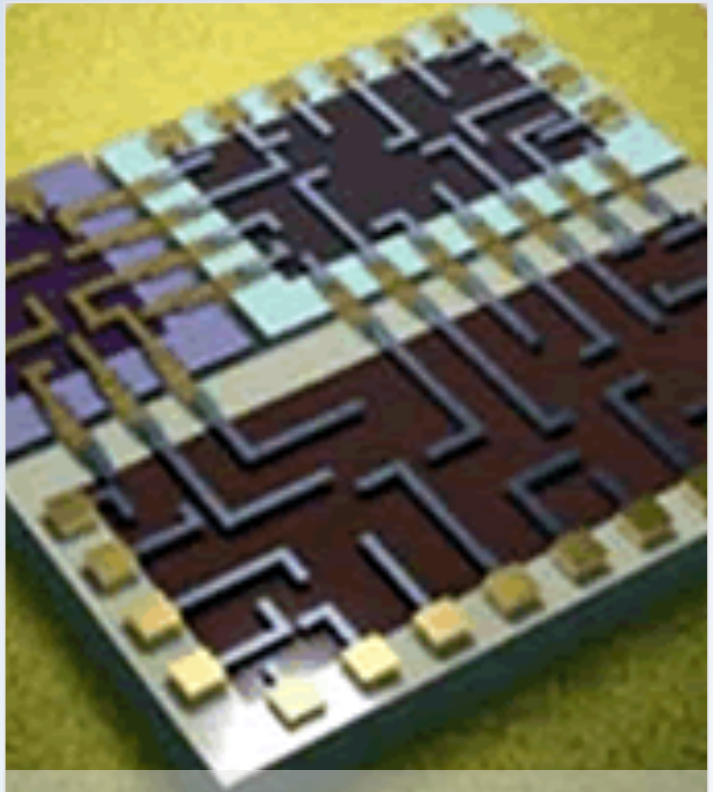
Minimized TCB - Trusted Code Base

Secure Cryptographic Service Stack

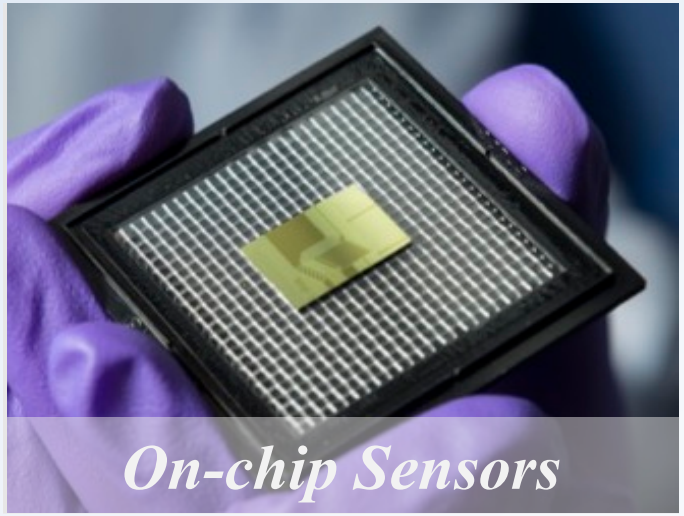
Protected Remote Update Mechanisms

Tamper Resistance

Secure Elements include multiple protection mechanisms against physical, active and passive attacks, including SCA & FI.



Meshes and Encrypted Busses

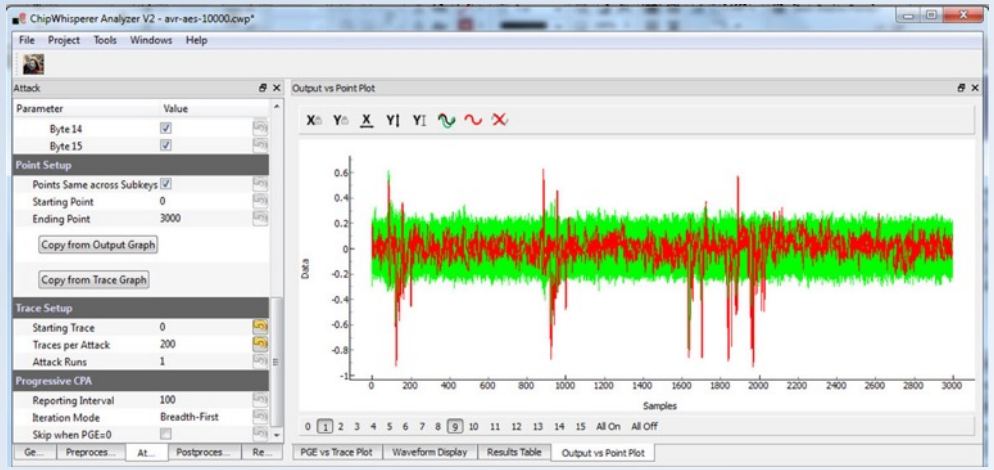


On-chip Sensors

Power / Electromagnetic analysis countermeasures

Proprietary and Patented Techniques

Side Channel Attacks



Side Channels

"A side-channel leaks information as a result of some physical, electrical or other behavioural characteristic of a system, that can be measured"

Execution Speed

Current Consumption

Current Leakage

Electromagnetic Emissions

Fault Injection Attacks

Fault Injection

“Perturbation of an execution environment with the sole objective of provoking a specific failure, in a controlled manner, within software or electronic circuits.”

Voltage

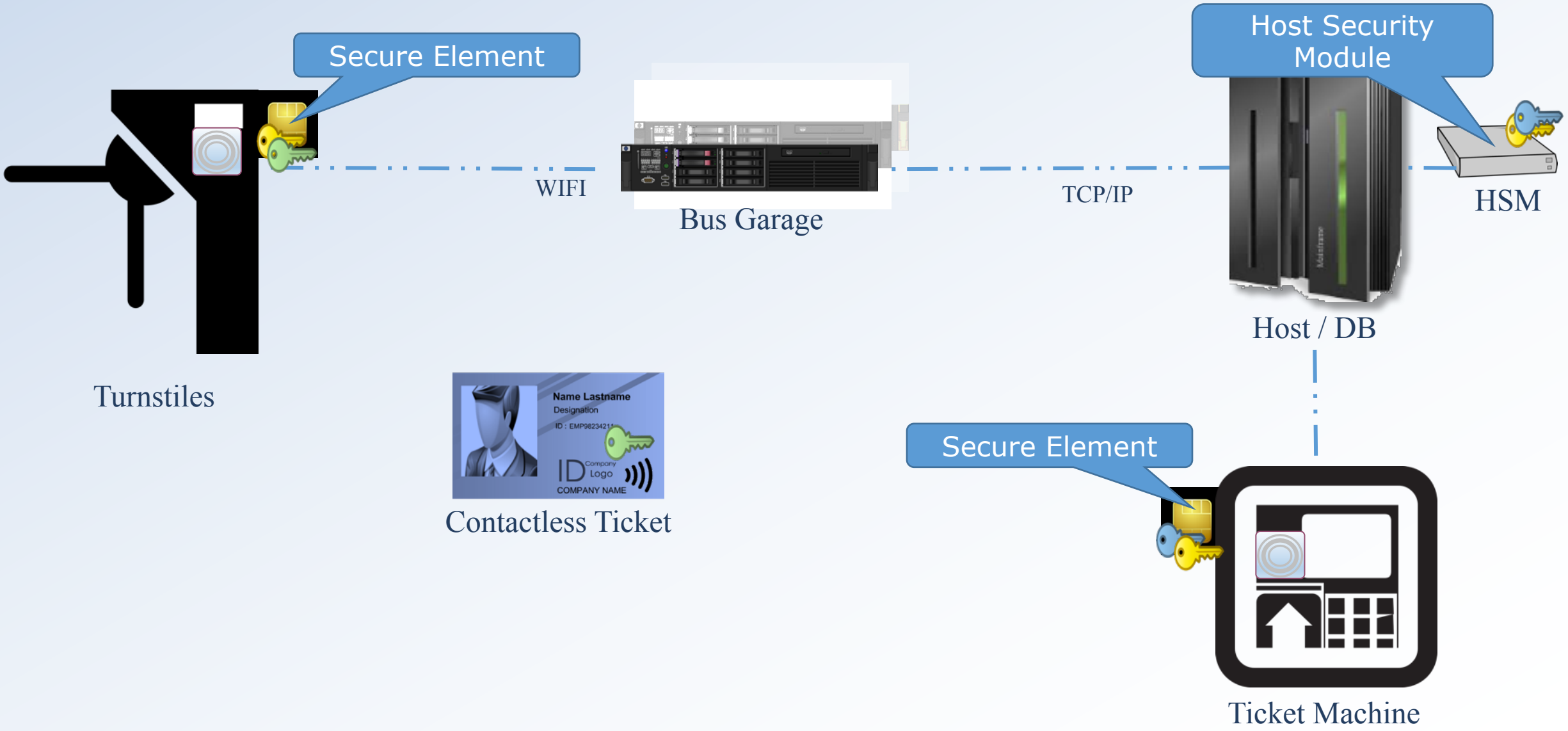
Clock Glitching

Electromagnetic Pulses

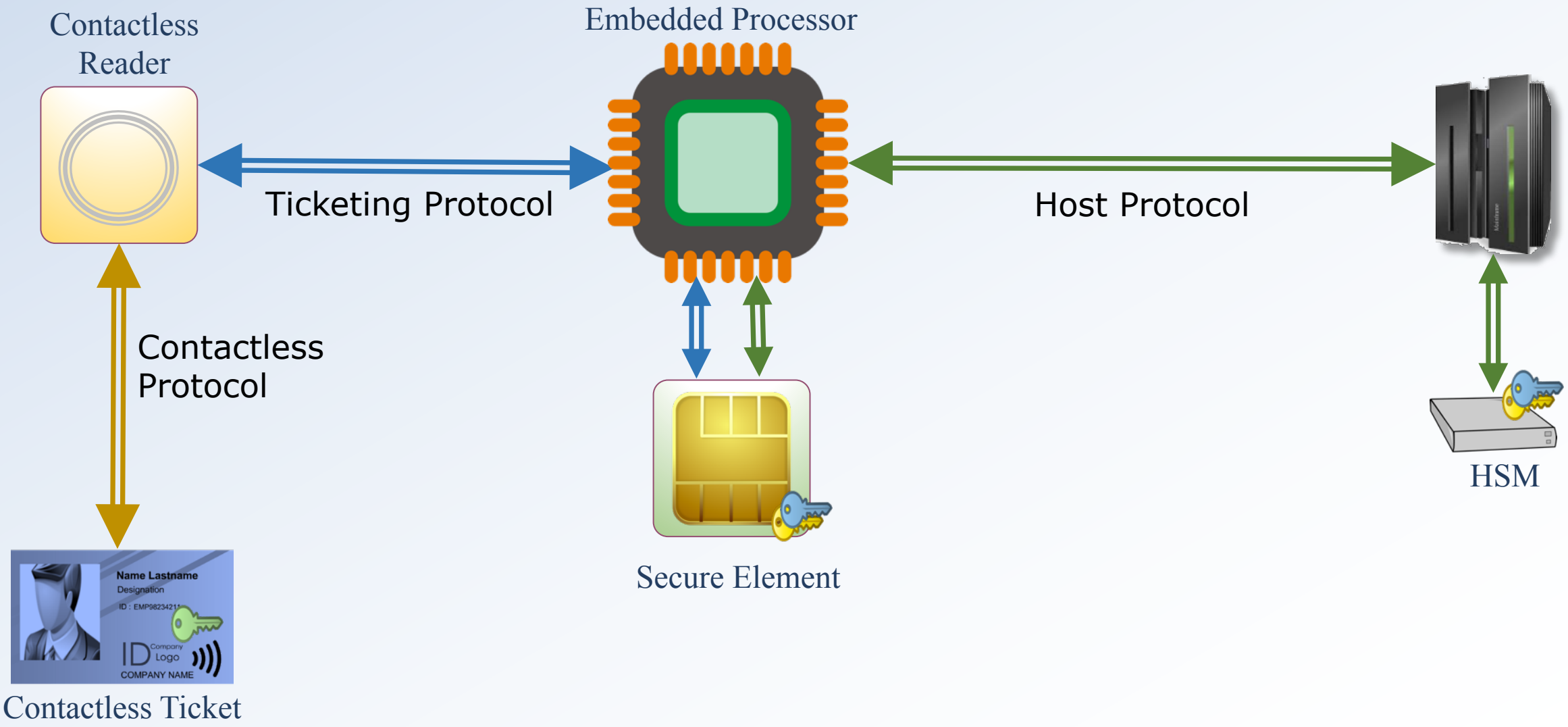
Case 2

Contactless Ticketing System using **Secure Elements**

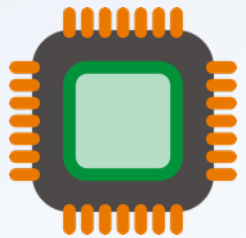
New Architecture



Information Flow



Example of Secure Decrement



Serial Number

Key “A”

Cryptographic MAC

Ticket Block

Valid Ticket

Decrement

Key “B”

Cryptographic MAC

Log Record

New Ticket Block

Cryptographic MAC

Security Evaluation

- ✓ Serial Number and Master Keys used to *derive* set of UNIQUE *per card* keys
- ✓ Cryptographic SIGNATURES protect
  - ✓ Ticket *Values*
  - ✓ *Log Records*
  - ✓ Host *Commands*
- ✓ All *Keys* and *Cryptographic Operations* PROTECTED by Secure Element

Properties of the Secure Elements

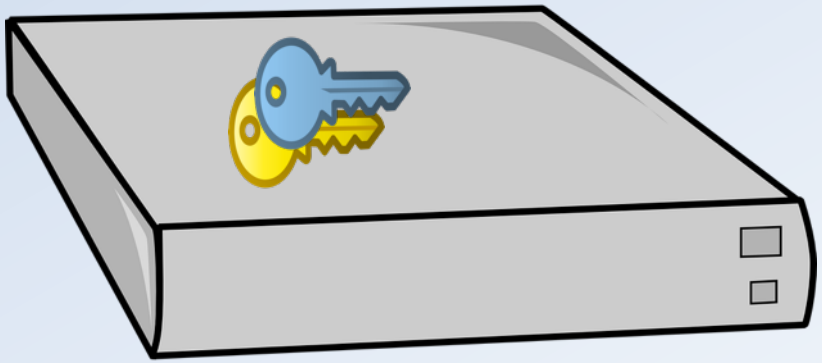
- ✓ Acts as ***Hardware*** **Root of Trust**
- ✓ **Secure Container** for **Keys** and other **Critical Data**
- ✓ **Secure Execution Environment** for stack of "High Level" **Secure Services**
- ✓ **UNIQUE** keys for **Mutual HOST Authentication**

Did Somebody Say

? **H.S.M.** ?

Host/Hardware Security Module

HSM – Hardware Security

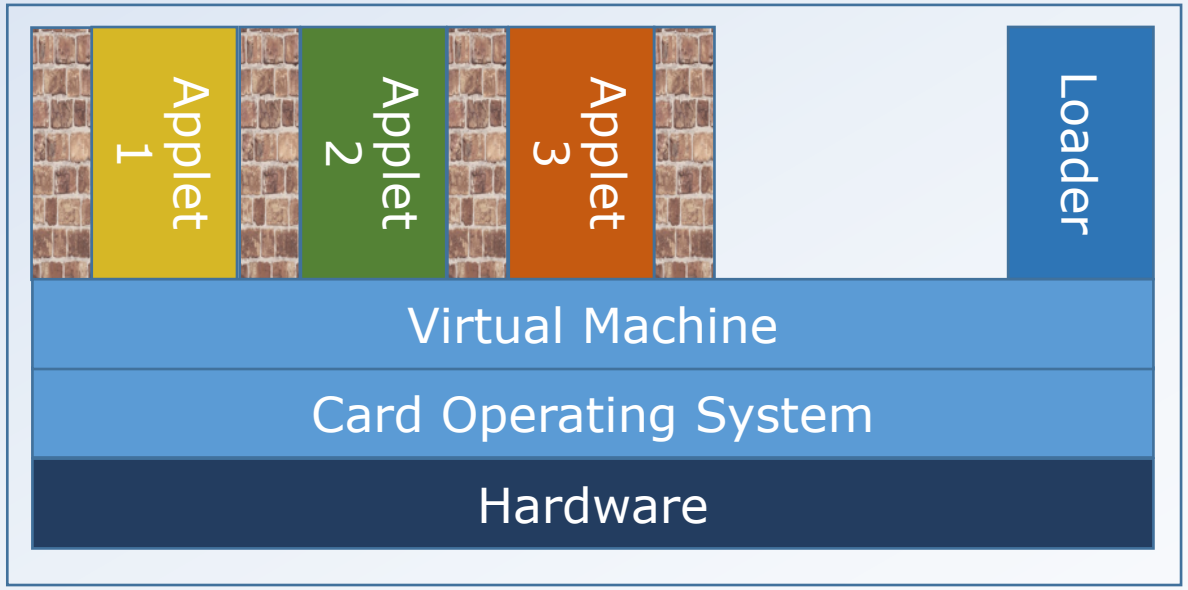


- ✓ High-performance **Cryptographic Hardware for Servers**
- ✓ **Secure Key Storage** – *plaintext* **Keys** never leave HSM
- ✓ Modern HSM's offer a **Secure Execution Environment** for stacks of "High Level" **Secure Services**

Secure Elements

In a Modern World

Multi-Application Paradigm



- ✓ Multiple **independent** card-resident applications (applets)
- ✓ Highly specialized **Virtual Machine** executes applet code
- ✓ **Firewalls** enforce Applet separation
- ✓ **Isolation** between applets and Card Operating System
- ✓ **Well-specified API** for comms, crypto, storage, state management
- ✓ **Remote** Applet Management

Principal Alternatives

MULTOS

- "Open" Consortium – MAOSCO
- Develop in C / ASM
- Markets: Banking, e-ID, IoT
- Centralized KMA using RSA certificates, or Issuer-centric (step-one)

JavaCard

- Sun Microsystems, now Oracle
- Develop in Java (OOP)
- Markets: SIM-Cards, Banking, e-ID
- Issuer-centric management

Development Process

SmartDeck Suite

Write code in C (and/or ASM if desired)

- Reduced libc, no dynamic mem.
- Libraries of *Primitives*
- Global data-spaces (NV, private, public)

Compile, then …

- Generate ALU – Application Load Unit
- Generate/Request ALC – Load Certificate
- Emulate, or Load and test

JavaCard Dev.

Write code in Java

- Lots of missing types and classes
- Use *javacard.** namespaces
- Static objects, *fixed* mem. usage

Compile, then …

- Java byte-card conversion
- Off-card byte-code verifier
- Emulate, or Load and test

Embedded (resource-constrained) *mindset* and **card-specific** functionality

Non-volatile Memory, Communication, Cryptography, Atomic Transactions, …

Case 3

NFC Mobile Payments using *Virtual* Secure Elements

2018

NFC

NFC (Near Field Communication) Forum
created in 2004

Contactless
REBRANDED

3 Operating Modes

Reader / Write Mode

Device can read/write any supported TAG type

Card Emulation Mode

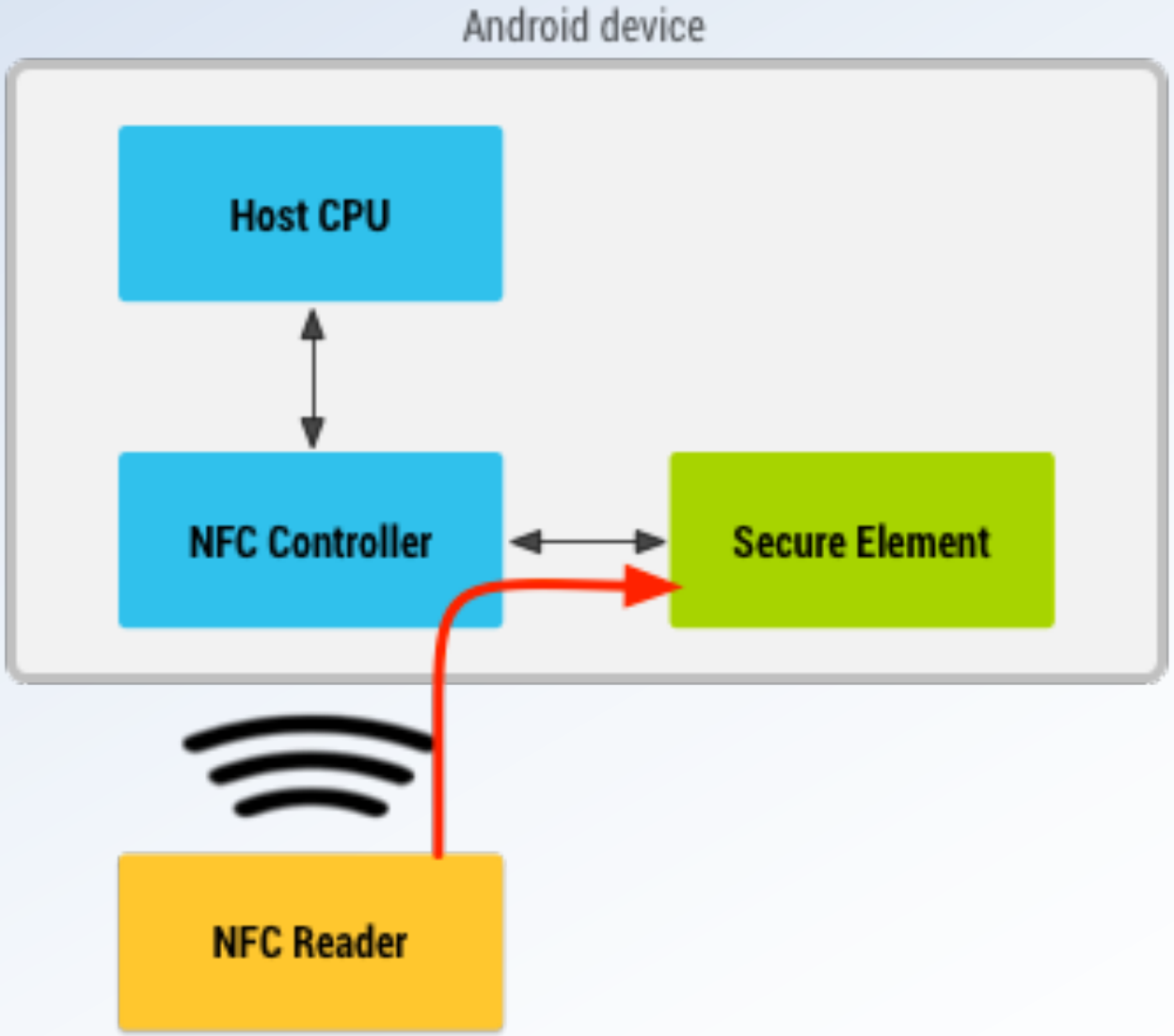
Device acts as a contactless

Peer to Peer Mode

Two NFC devices can exchange data

4 Types of TAG

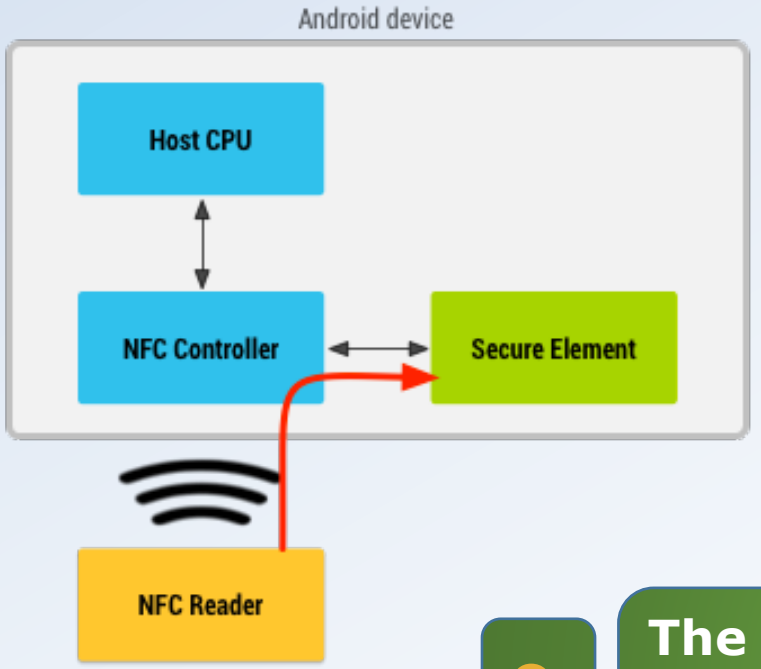
NFC-style Secure Elements



From NFC to HCE, in 3 easy steps …

1

Developer says:
"I want to load my APPLET onto the phone's Secure Element"

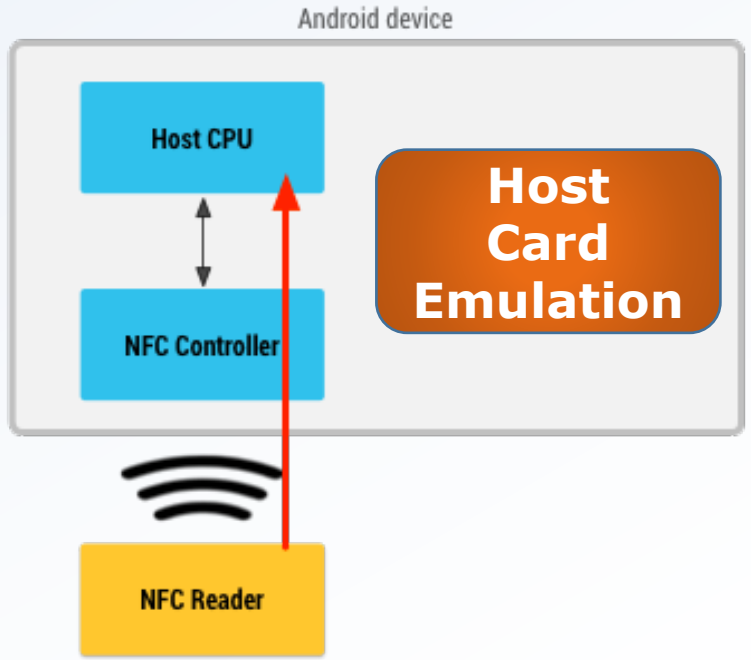


2

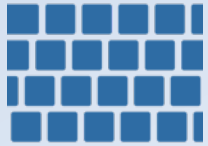
The problem:
"Who *OWNS* the mobile Secure Element" ?

3

Google's solution:
"Forget about the Secure Element, we'll do it in software!"



Characteristics of HCE



Uses Standard Android Sandboxes

APP processes APDU commands

Delegate security to Server?

Use Android HW-Backed Keystore?

Root/Debug/Tools can be a problem

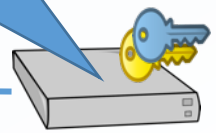
Payment Token Service Provider

Payment Request

Payment Network

Tokens, [Security]

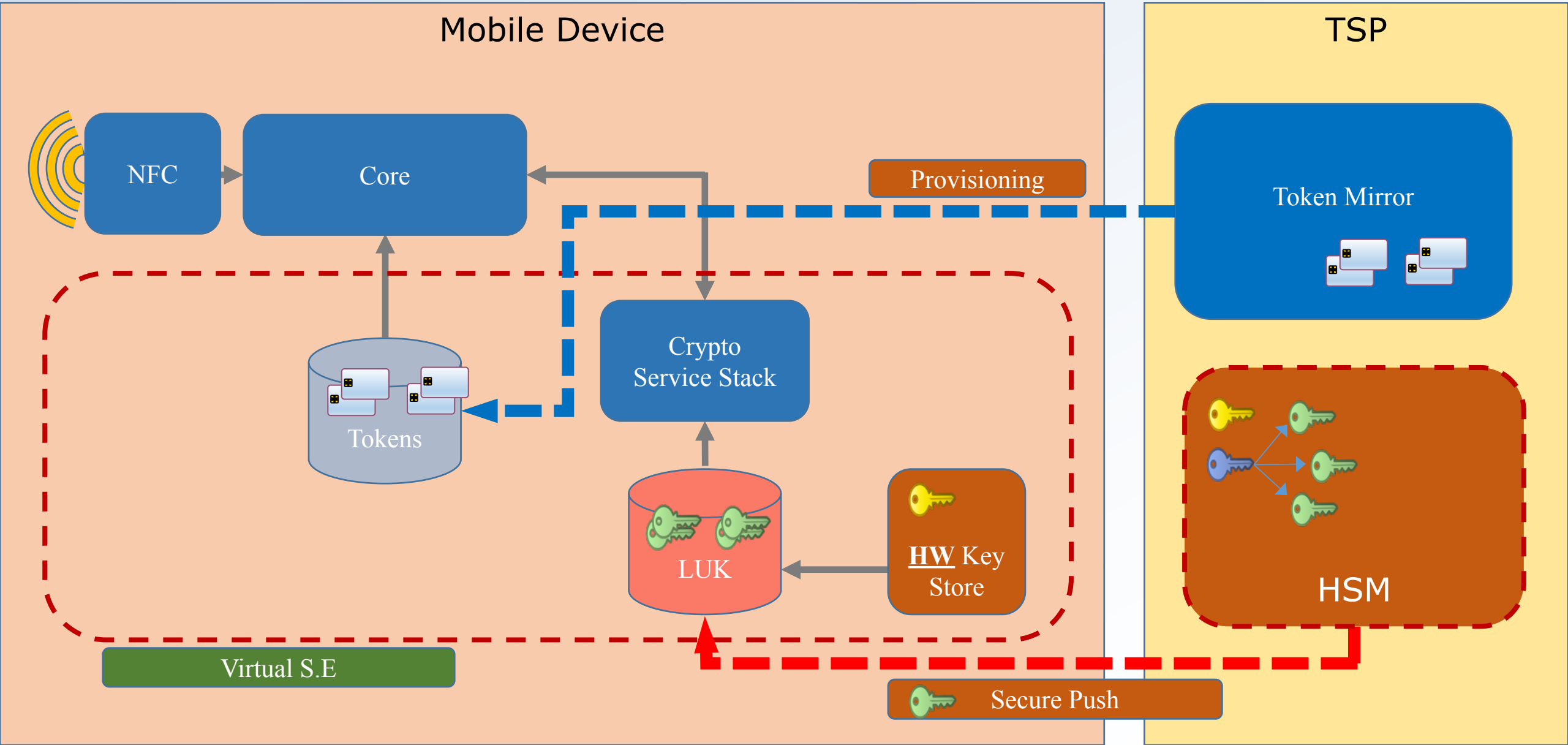
Host Security Module



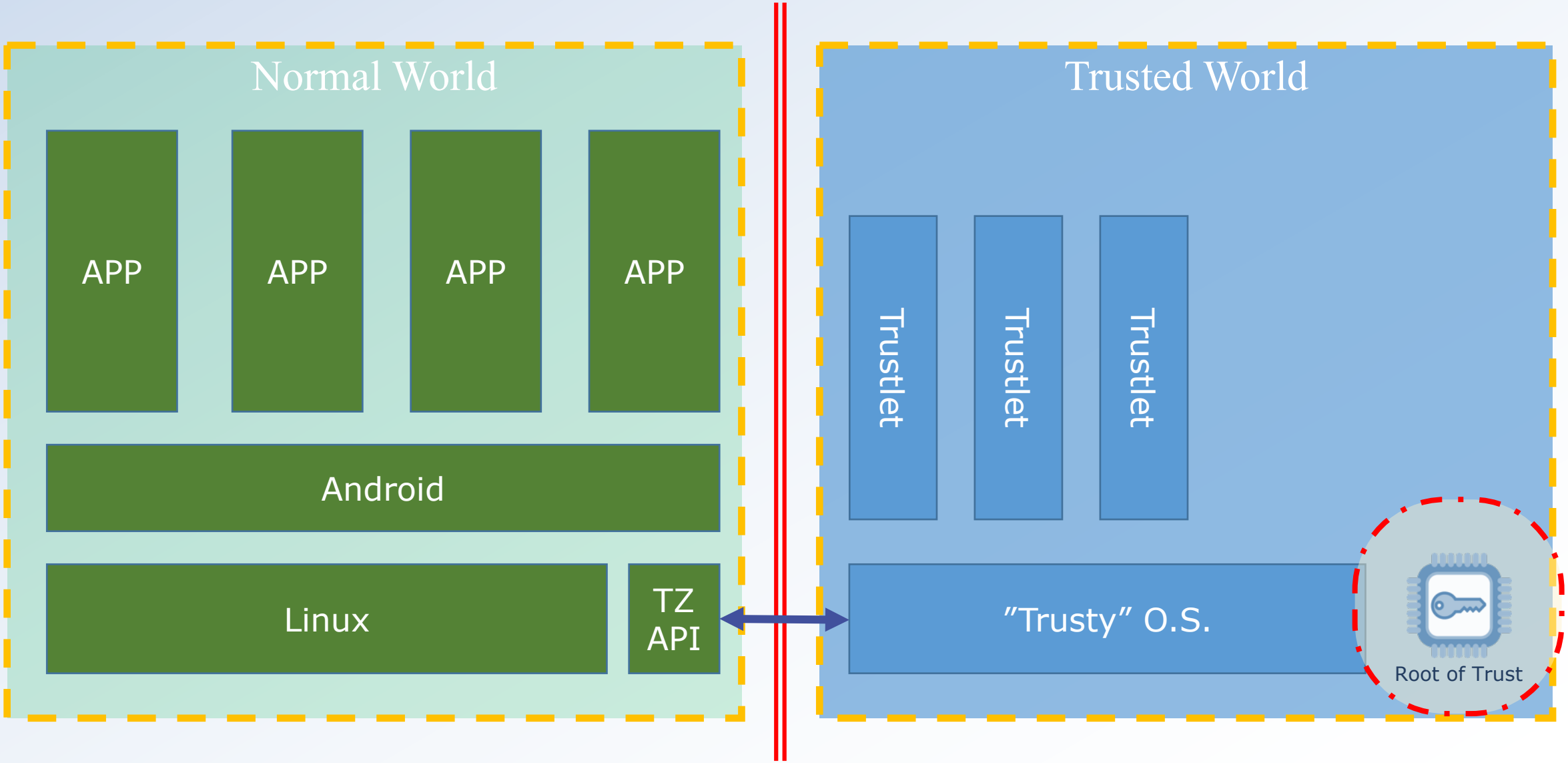
HSM

Token Service Provider

Virtual Secure Elements



Android/TEE Architecture

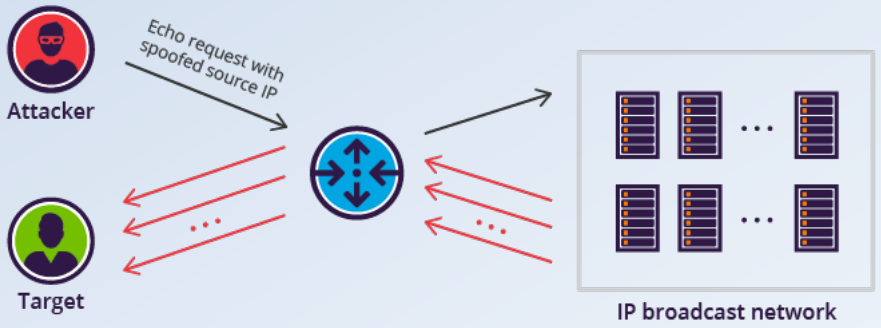


Secure Elements

For the IoT

A Promising Future In the Making ...

IoT : Security Recommendations**

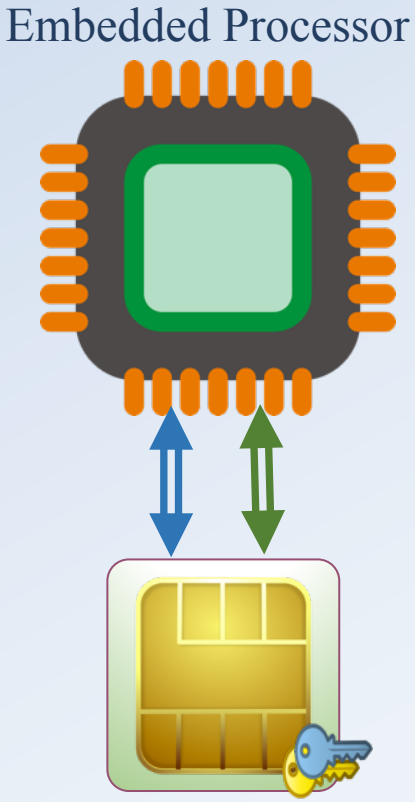


- ✓Secure by Design
- ✓Hardened Configurations
- ✓Secure Updates
- ✓Unique Credentials per Device
- ✓Cryptography

** BITAG Internet of Things Security and Privacy Recommendations NOV/2016

Secure Elements for the IoT

Remaining Problems ?



- ✗ Interception/Spoofing of communications
- ✗ Compromise of Embedded Software
- ✗ Transfer Secure Element to other machine
- ✗ Performance?

→ Have a chat with Bart and Jan Tobias ☺

Hartelijk bedankt !

Obrigado!

sean.wykes@nascent.com.br

CRYPTOGRAPHIX

https:// cryptographix.org/explore

nascent Secure Technologies

/in/seanwykes