VUB
SECURE EXECUTION FOR EMBEDDED ENVIRONMENTAL MONITORING APPLICATIONS

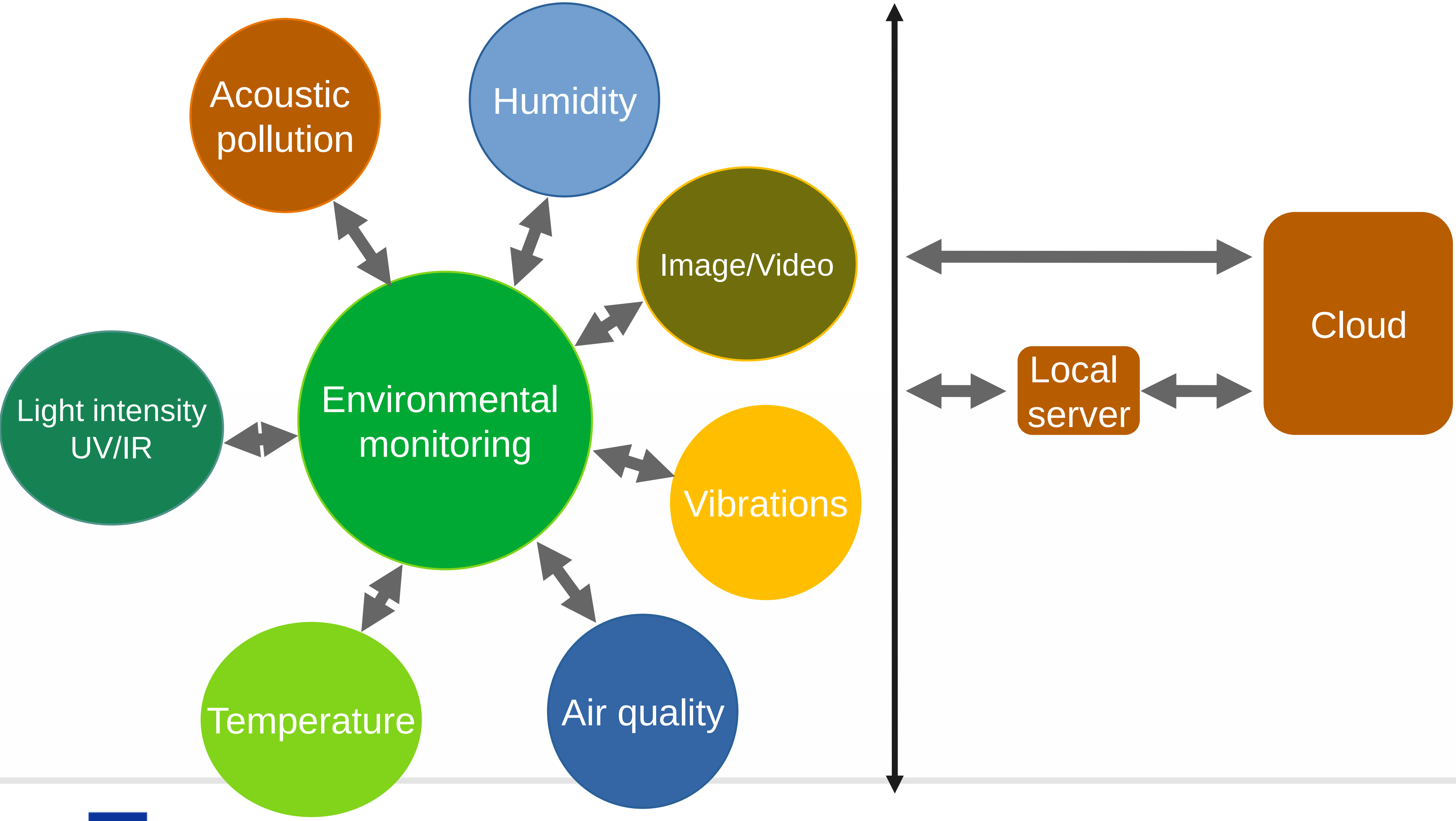
Laurent Segers
An Braeken
Bruno da Silva
Abdellah Touhafi

17 / 04 / 2024

VUB - ENVIRONMENTAL MONITORING

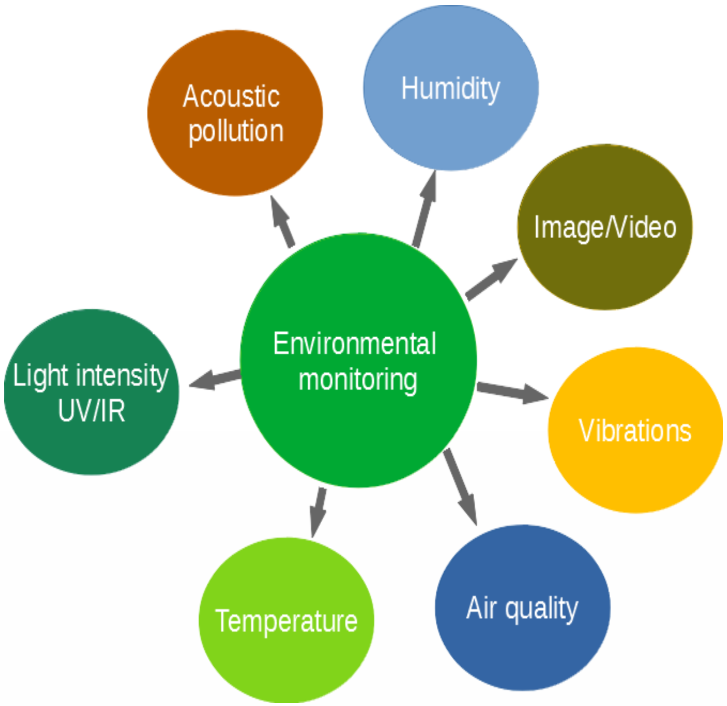
ENVIRONMENTAL MONITORING

TOPOLOGY



VRIJE UNIVERSITEIT BRUSSEL

ENVIRONMENTAL MONITORING



SECURE LOW-END SENSING

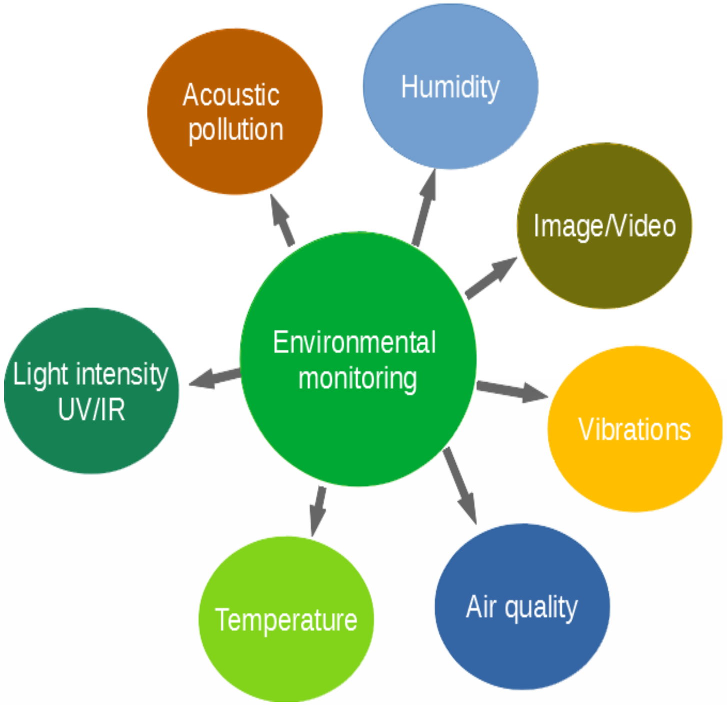
- Limited memory/processing capabilities
- Capable of reading sensors with low update rates (i.e. 1Hz, 10Hz)
- Data integrity & confidentiality of sensor-readouts
- Trusted GPS & RTC

SECURE REMOTE UPDATE

- Lightweight key agreement protocol using PUF
- Secure boot

ENVIRONMENTAL MONITORING

LOW-END SENSING



Risks & mitigation

- Moving device to other location
  Location awareness (GPS) can mitigate security risks

- Wireless communication → spoofing, jamming, read-out of data, data alteration
  → *Store jammed data locally until successful retransmission*
  → *Encryption/integrity protection of transmitted data*

- Modifying/Reading of locally stored data
  Data encryption, data integrity check

- Firmware (mis)configuration
  → *integrity test during attestation*

- Over the air updates compromised with spoofed firmware/configuration
  → *Authentication + encryption of firmware*

VRIJE UNIVERSITEIT BRUSSEL

LOW-END SENSING

SECURITY REQUIREMENTS (HARDWARE – SILICON SUPPORTED)

- Minimal Hardware-based code execution isolation if possible
  → TrustZone
- Basic Root-of-Trust (for some applications)
- Secure boot
- Trusted peripherals (when possible)
- Optimizations for secure storage
- Secure over the air updates

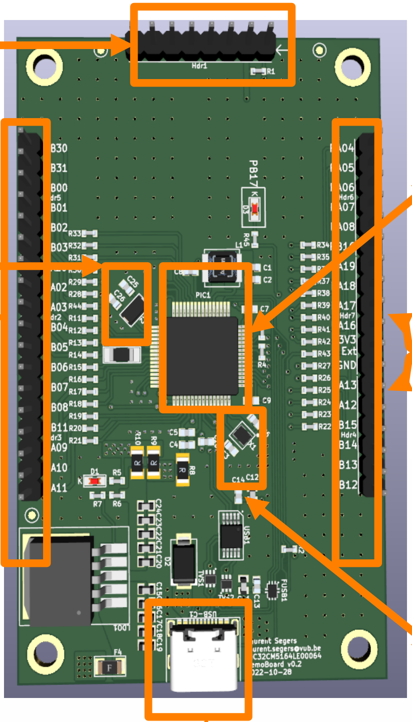
TRUSTED EXECUTION ENVIRONMENT (JUNE 2023)

NXP/Freescale	STMicroelectronics	Microchip
LPC5500-series based on the **ARM-Cortex-M33 MCUs**	STM32 based on **ARM-Cortex-M33** (STM32L5 and STM32U5) ultra-low-power MCUs	PIC32CM5164 LS60/LS00 based on **ARM-Cortex M23**
• TrustZone • Energy efficiency • SRAM PUF-based RoT • Encrypted images • ~ 4.5€/pc (1000pc)	• TrustZone • Ultra low-power • Cryptographic modules integrated • ~7.5€/pc (1000pc)	• TrustZone • Ultra low-power • Cryptographic modules integrated • Exist in secure and non-secure variants • ~4€/pc (1000pc)

MICROCHIP ARM CORTEX-M23

PIC32CM5164-BASED PROTOTYPE

Custom designed board



Programming header

RTC @ 32kHz

IO (+interruptable IO)
Sercom (SPI, I2C, UART)

PIC32CM5164L**E**00064 (non-secure)

PIC32CM5164L**S**00064 (secure)

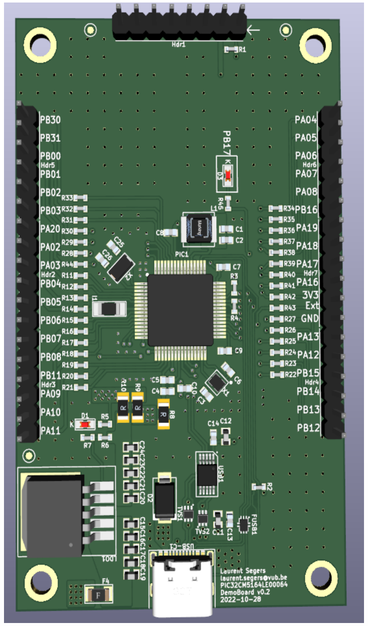
External power

Main crystal @ 32MHz

USB for power over USB + commucation to PC

MICROCHIP ARM CORTEX-M23

PIC32CM5164-BASED PROTOTYPE



Based on ARM23 core platform with 512kB flash, 64kB SRAM, 32kB boot ROM

Offers TrustZone (5 regions in flash, 2 regions in data flash and 2 regions in SRAM)

1 TRNG, AES-256/192/128, multiple SHA methods

Public key validation support, 1 internal sign private key attestation

Secure boot with customizable secure boot public key

Optimized for secure storage + TrustRAM

Up to 8 anti-tamper output IO + secure pin multiplexing to isolate secure communication channels

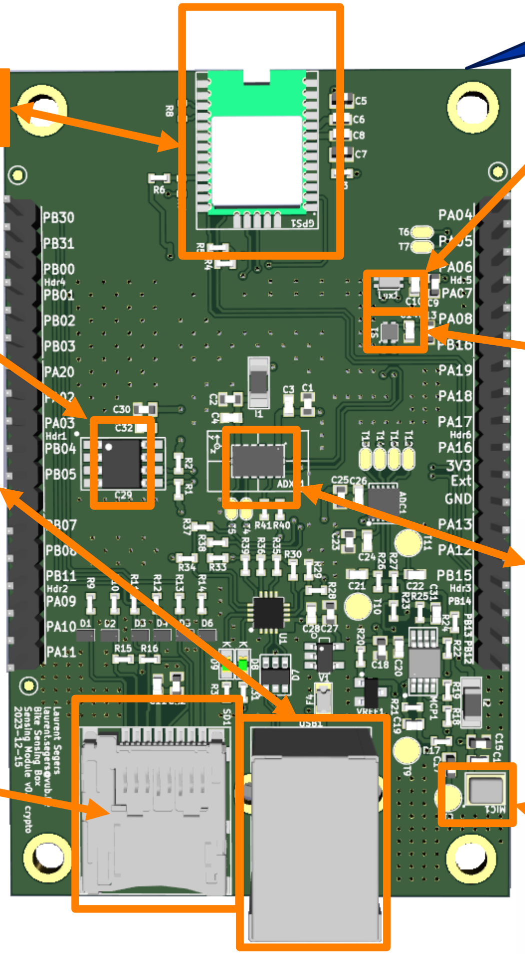
Unique 128-bit serial number

Separate registers for secure and non-secure application

LOW-END SENSING

SENSOR MODULE

Grouping sensors in secure/non-secure peripherals



GPS (L96-M33)

ATSHA204A crypto-authentication module

UART to USB communication to PC

SD-card for logging

VEML3328 light sensor (RGB+IR)

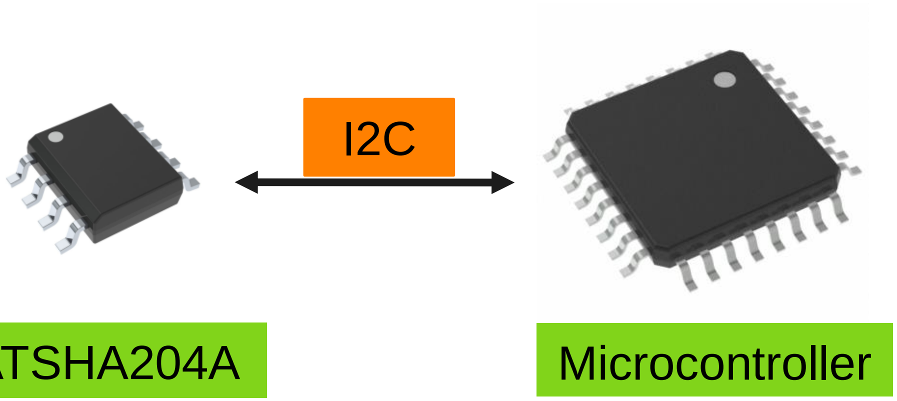
SHT41 temperature + humidity sensor

ADXL343 3-axis accelerometer

SPU0410LR5H-QB analog microphone + SPI ADC

ATSHA204 CRYPTO AUTHENTICATOR CHIP

→ KEY AGREEMENT PROTOCOL



I2C

ATSHA204A

Microcontroller

Crypto element with protected / anti-cloning key storage

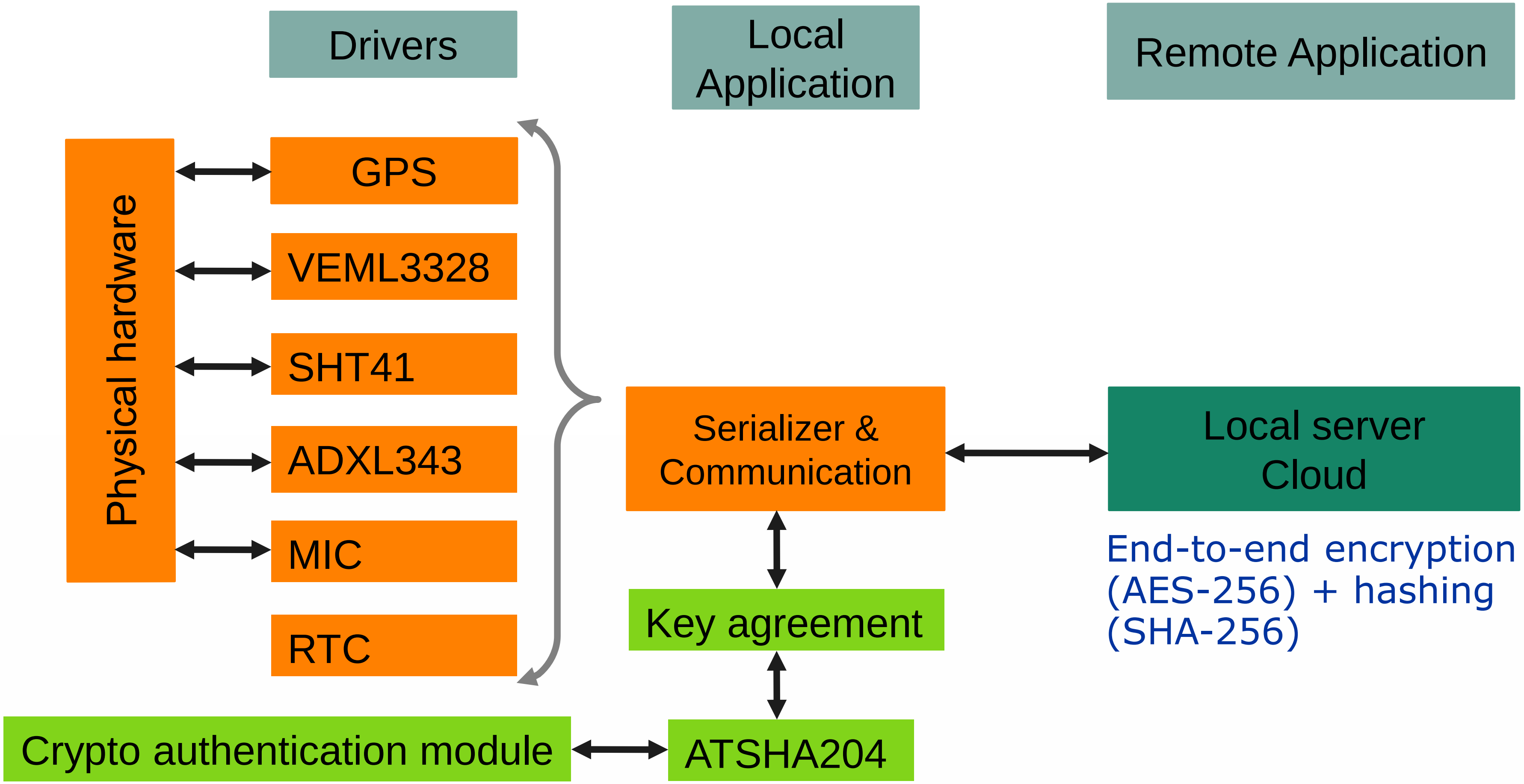
Secure configurable storage for up to 16x256 data segments / keys

Multiple hardware based crypto algorithms
→ (Hash-base) Message Authentication Code (MAC)

3 different memory segments which can be locked
→ can be “disabled” when intrusion detected → unusable

EMBEDDED FIRMWARE (1)

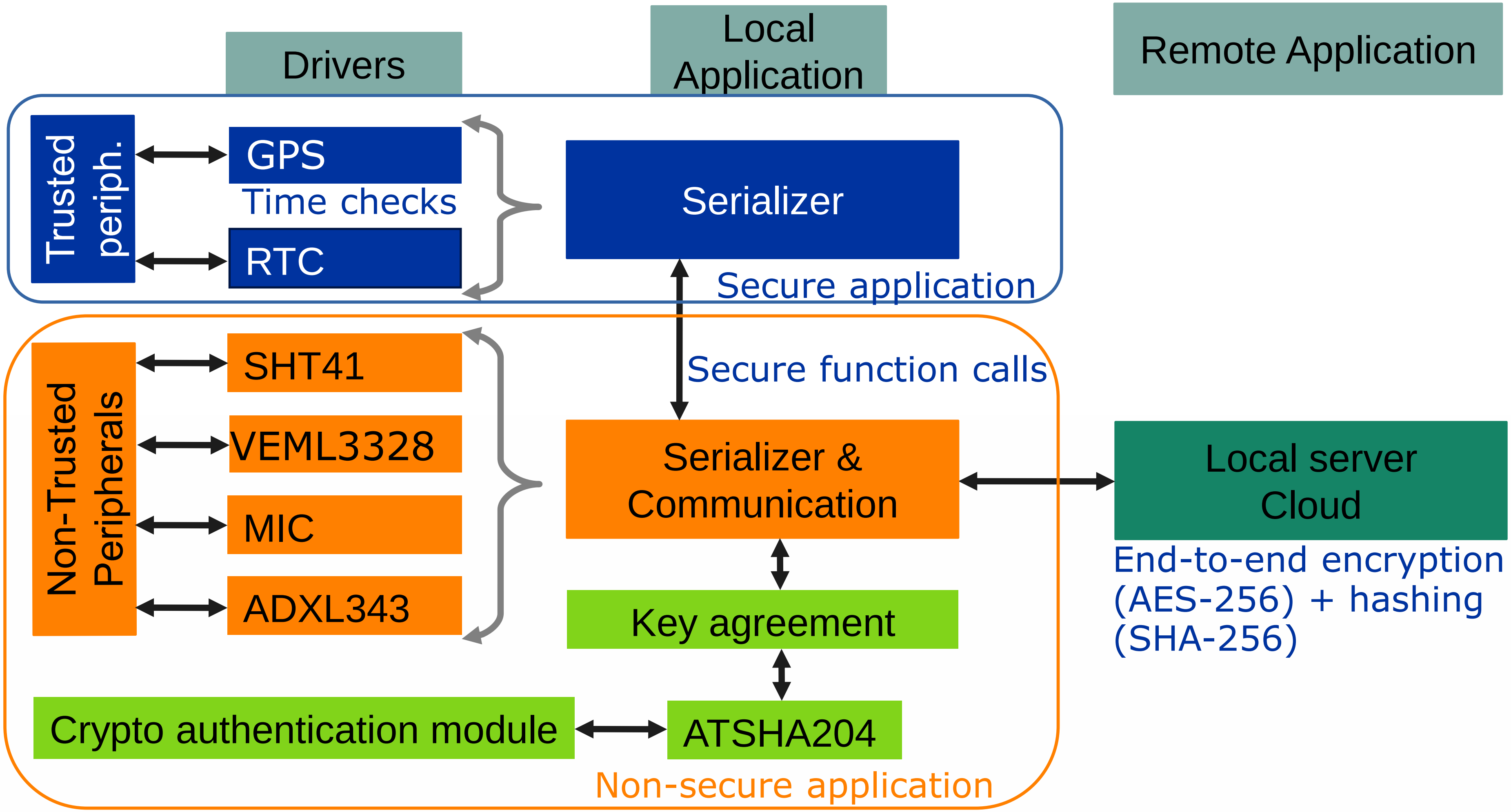
MODULAR APPROACH WITHOUT TRUSTZONE



End-to-end encryption (AES-256) + hashing (SHA-256)

EMBEDDED FIRMWARE (2)

MODULAR APPROACH WITH TRUSTZONE



End-to-end encryption (AES-256) + hashing (SHA-256)

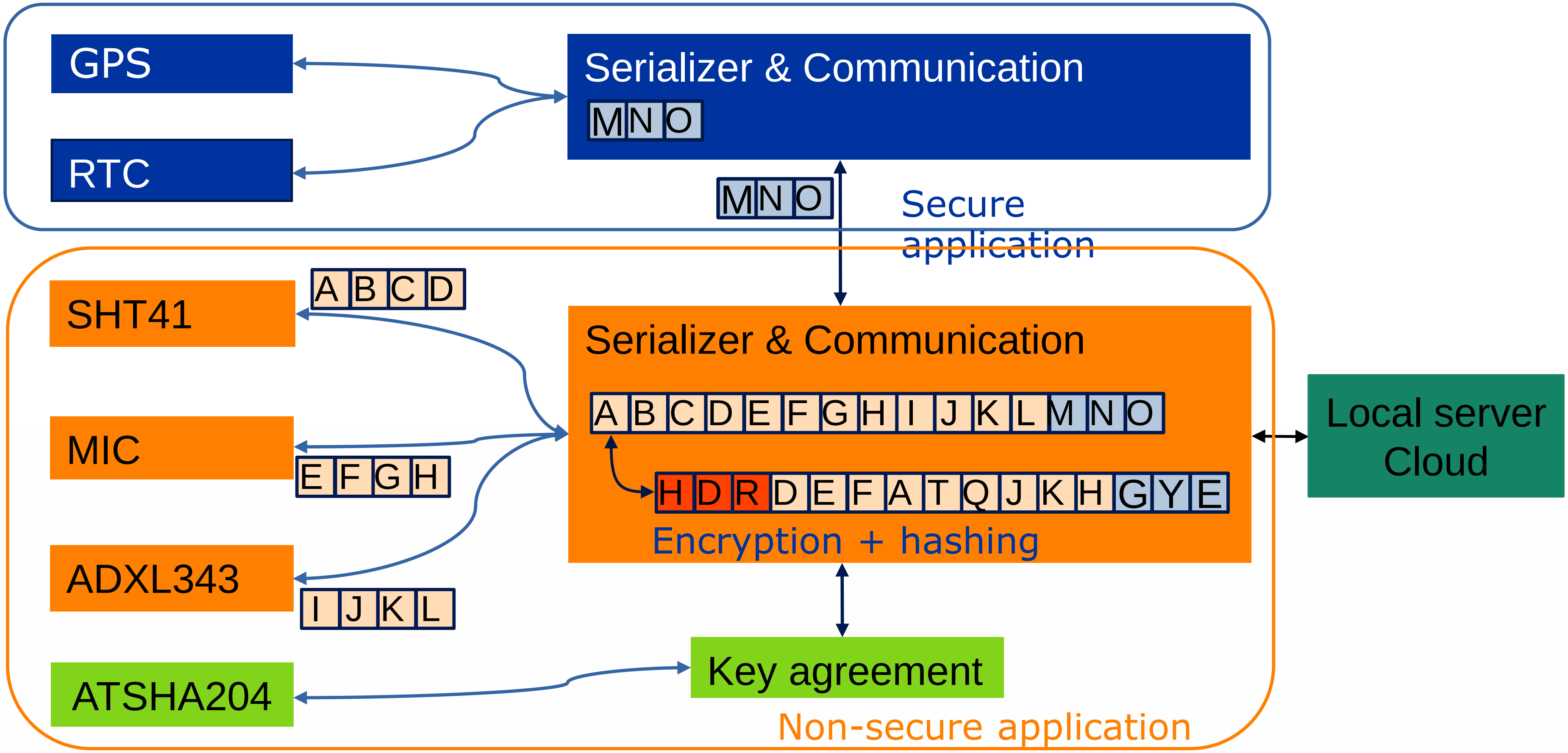
EMBEDDED FIRMWARE (3)

CONSIDERATIONS WITH TRUSTZONE

- One program flow on regular microcontrollers without TrustZone
- TrustZone involves re-thinking application into secure and non-secure code → 2 program flows!
- Special function calls between secure and non-secure code (veneers)
- Limited number of libraries/peripherals can be in TrustZone
- Hardware peripherals (sensors and communication) bound to secure/non-secure code → double set of hardware registers
- Crypto authentication module is part of non-secure application

EMBEDDED FIRMWARE (4)

COMMUNICATION STRATEGY



Communication drivers & serializer derived from OSI model
Local server / Cloud apply opposite operations

COMMUNICATION KEY ISSUES

SECRET COMMUNICATION KEY FRESHNESS

Secret key might get leaked

→ Secret key can be reverse engineered via firmware extraction

→ Sensitive data leading to secret key might get compromised

→ Encryption information can be “learned” based on pattern search

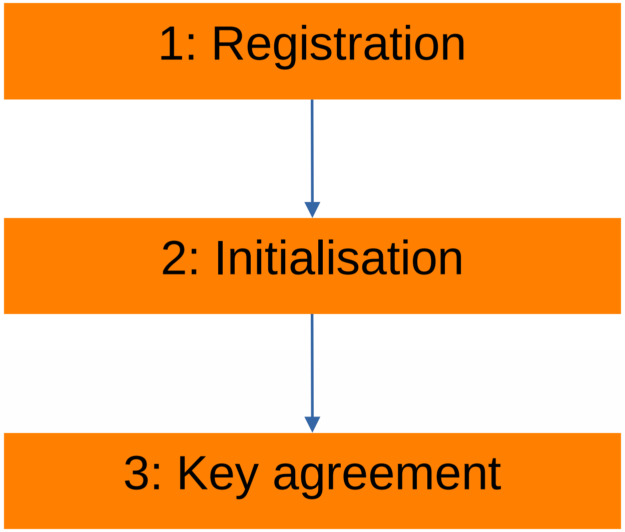
→ Firmware errors might leak sensitive information

Need for secret key refresh – Key agreement protocol

❗ Key agreement protocols generate “predictable” new keys
→ Physical Unclonable Function (PUF)?

KEY AGREEMENT PROTOCOL WITH PUF

OUTLINE



1: Registration

2: Initialisation

3: Key agreement

PUF => P(input)=output
→ silicon variable but repeatable

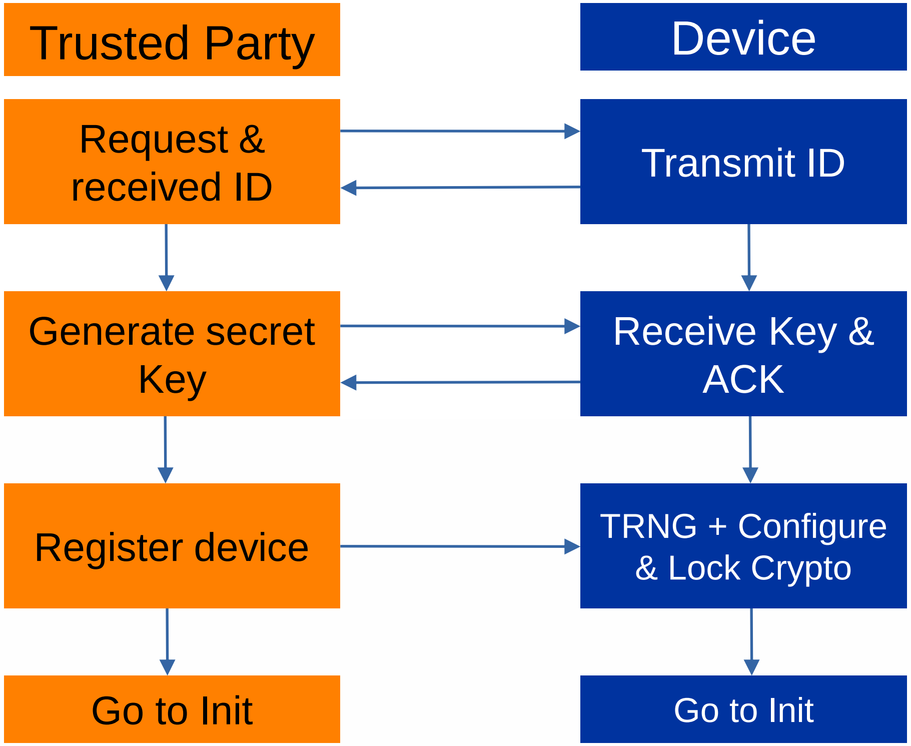
Freescale ARM Cortex M33 has built-in SRAM-based PUF

!!Most MCUs do not have SRAM-based PUF!!

ATSHA204: permanent storage for TRNG ~PUF
→ Proper configuration + locking
→ TRNG overwritten when undesired access
→ TRNG can not be read

KEY AGREEMENT

1: REGISTRATION



Trusted Party

Request & received ID

Generate secret Key

Register device

Go to Init

Device

Transmit ID

Receive Key & ACK

TRNG + Configure & Lock Crypto

Go to Init

Occurrence: once

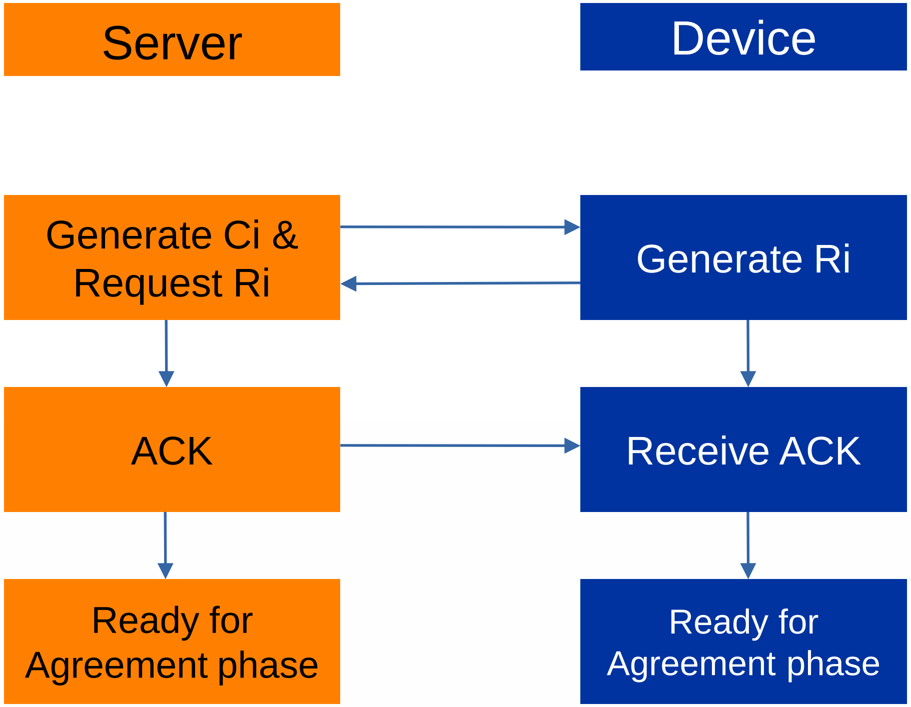
Physical connection or via Trusted Third Party

Device initializes the ATSH204 module with TRNG to mimic PUF

ATSH204 locked after registration finished

KEY AGREEMENT

2: INITIALISATION



Occurrence: after registration, once

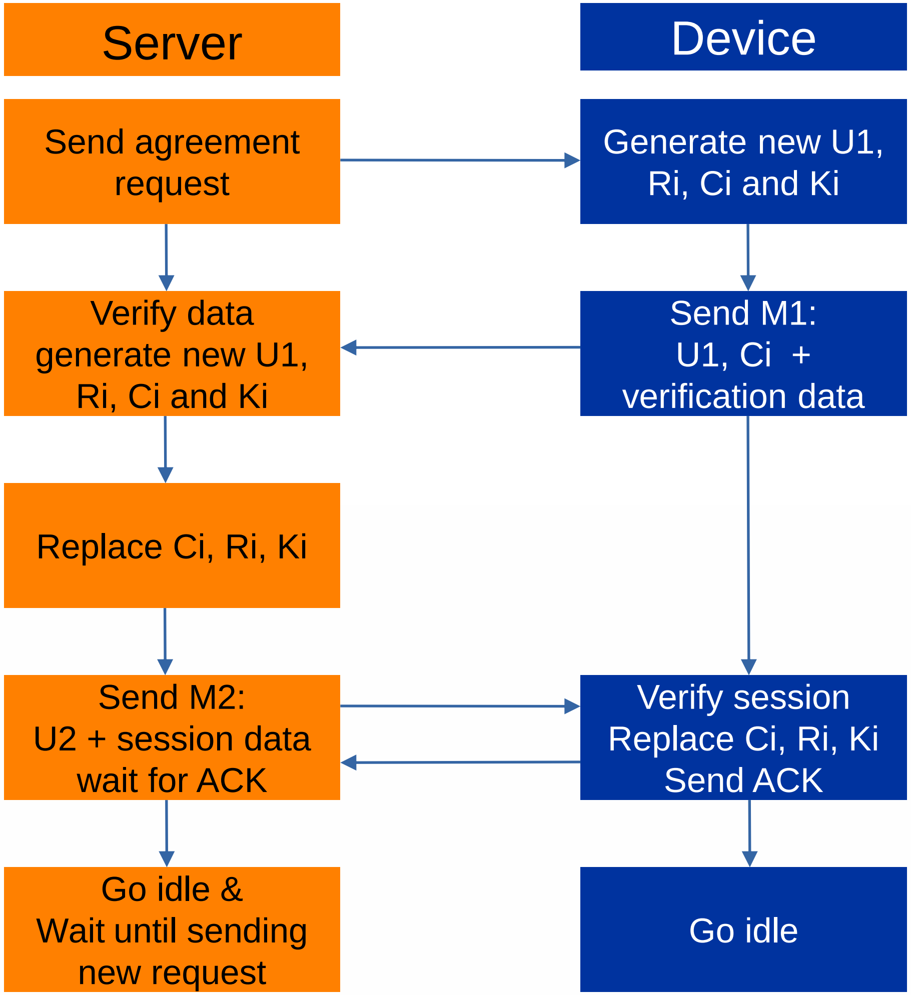
Via secure connection

Device tests ATSH204 with response generation

H(Ki|P(Ci)) = Ri
Generated on ATSHA204

Ri = response
Ci = challenge

3: KEY AGREEMENT/RENEWAL


Server
Device
Send agreement request
Generate new U1, Ri, Ci and Ki
Verify data generate new U1, Ri, Ci and Ki
Send M1: U1, Ci + verification data
Replace Ci, Ri, Ki
Send M2: U2 + session data wait for ACK
Verify session Replace Ci, Ri, Ki Send ACK
Go idle & Wait until sending new request
Go idle

Occurrence: when required, multiple times

Via secure connection

Ri = response
Ci = challenge
Ki = symmetric secret key

H(Ki|P(Ci)) = Ri
Generated on ATSHA204

U1, U2: random values

M1, M2: exchange messages

RESOURCE CONSUMPTION

TRUSTZONE + KEY AGREEMENT

Code execution time / power overhead TrustZone

→ Between 100’s cycles up to 1000’s cycles (1-3%)

Peripherals

→ Harmony does not allow secure access to unsecure peripherals
→ Possible via registers & custom drivers

Program code overhead due to TrustZone

→ TrustZone minimum code size: 15kB
→ Memory provisioning at Harmony design phase (min. 20% TrustZone)

Key agreement

→ Longest data communication streams (i.e. up to +-140 bytes)
→ ATSHA204 requires up to 0.4s to compute H(Ki|P(Ci)) = Ri

RESOURCE CONSUMPTION

AES-256 ENCRYPTION + SHA-256 HASHING

Transmission overhead #bytes
→ Data sent in “plain readable” format: ~38-84 bytes per packet
→ Key agreement: up to 140 bytes per packet

→ AES-256 CBC encryption + IV: + 17 to 32
→ SHA-256 hashing: +32 bytes
Total overhead: 49-64 bytes => +- 100% on average

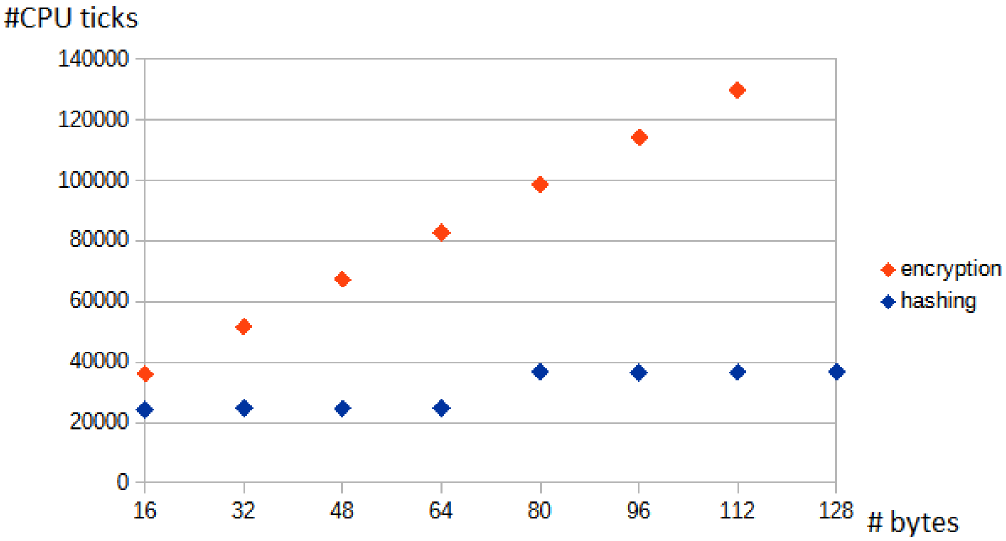
Memory / Flash overhead

Without crypto: 215kB flash / 17.1kB SRAM
With crypto: 207kB flash / 15.2kB SRAM

RESOURCE CONSUMPTION

AES-256 ENCRYPTION + SHA-256 HASHING

Encryption/hashing time overhead



AES-256 (CBC) encryption time increases per block of 16 bytes

SHA-256 hashing increases per block of 64 bytes

Example:
Encryption + hashing of 80 bytes
$T_{required}$ = 2050us + 760us = 2810us

MCU running @48MHz
→ $T_{encryption}$ ~ 425us + 325us * #nblocks of 16 bytes
→ $T_{hashing}$ ~ 260us + 250us * #nblocks of 64 bytes

VRIJE UNIVERSITEIT BRUSSEL

MICROCHIP EMBEDDED TOOL DEVELOPMENT

USER FRIENDLINESS

Device configuration with MPLab X IDE (6.x)
+ Harmony

Code generation of drivers and configuration
→ engineer should focus on applications...

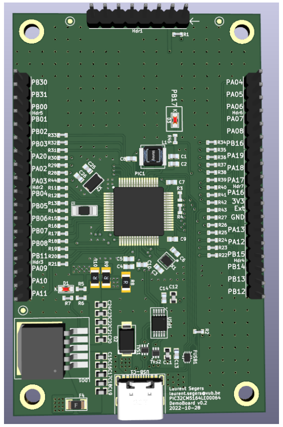
Each new version improves + new features, however...
→ project discrepancies
→ compiler flag discrepancies
→ ~~new project then required~~ → load project dependencies first

Solution/workaround
→ design with harmony/libraries during project creation
→ only update code later on
→ write own drivers on top of CMSIS if possible

CURRENT STATE & IMPLEMENTATION

- ✓ Low-end Microchip ARM23 (ARMv8 architecture) based platform selected and programmed
- ✓ TrustZone and secure remote communication
- ✓ Firmware development challenges
- ✓ Fine-grained impact analysis of TrustZone and secure communication
- ✓ Lightweight key agreement protocol using PUF
- ℹ All Sercoms (I2C, SPI, UART) are used + some methods hit MCU processing boundaries
- ℹ Limitations of programming tools & resolution

CURRENT NEXT STEPS



+

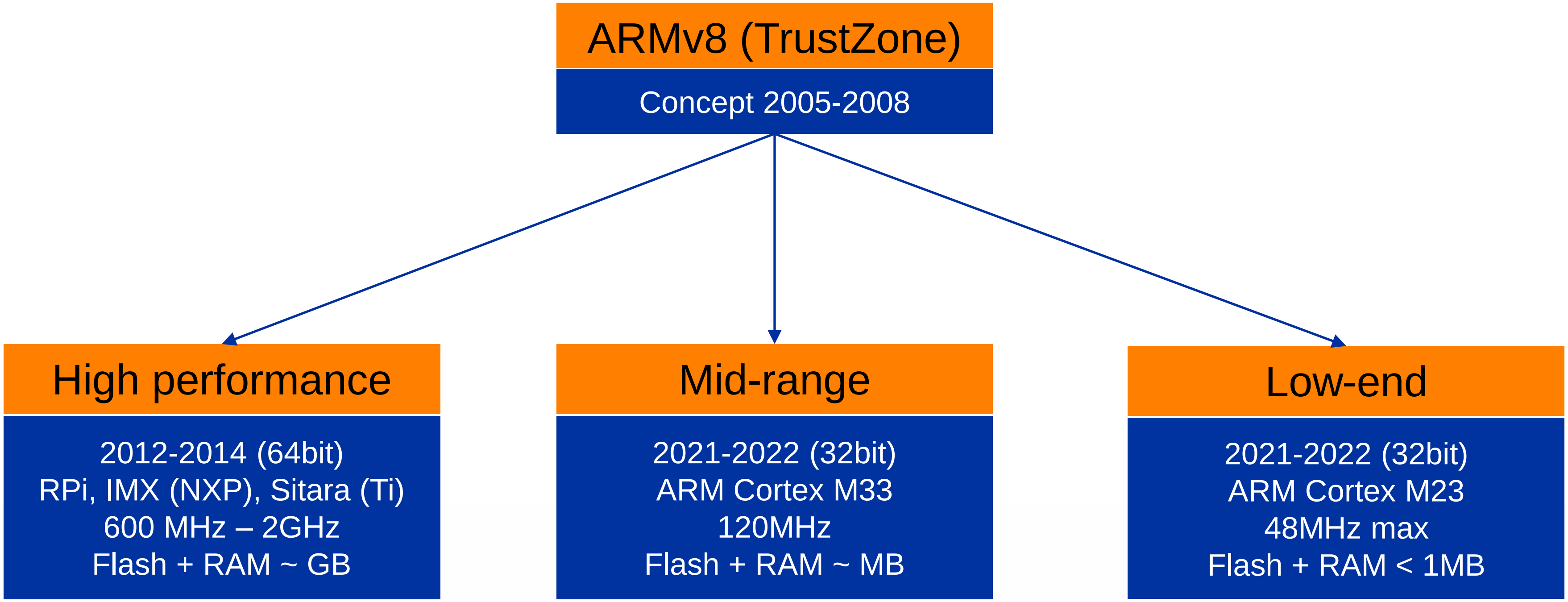
1: Bluetooth wireless connectivity

2: Secure boot

3: Secure boot + Root of Trust at expense of peripherals for ATECC608B module?

3: Remote firmware update? Secure/non-secure peripherals?

ARMV8 TRENDS



ARMv8 (TrustZone)

Concept 2005-2008

High performance

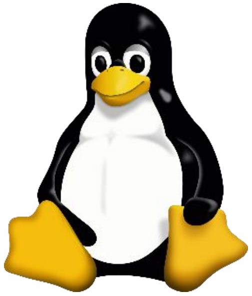
2012-2014 (64bit)
RPi, IMX (NXP), Sitara (Ti)
600 MHz – 2GHz
Flash + RAM ~ GB

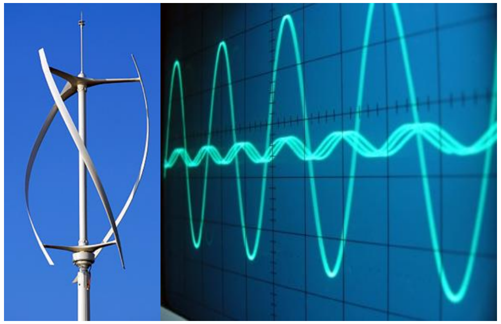
Mid-range

2021-2022 (32bit)
ARM Cortex M33
120MHz
Flash + RAM ~ MB

Low-end

2021-2022 (32bit)
ARM Cortex M23
48MHz max
Flash + RAM < 1MB





ARMV8 TRENDS

Microchip ARM Cortex M23 (2021-2022)
→ PIC32CM LS00 & PIC32CM LS60 series @48MHz
→ New devices are added

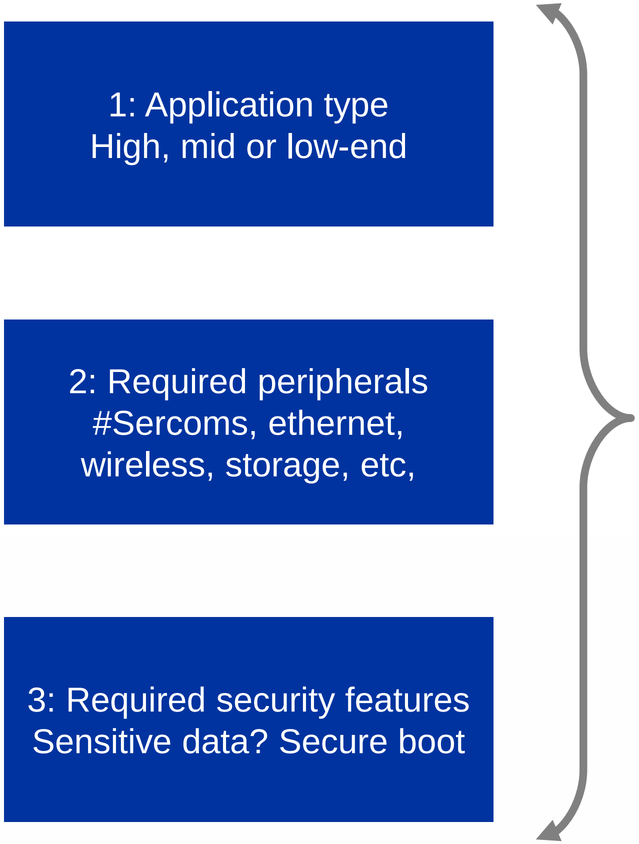
NEW Microchip ARM Cortex M33 (March 2024)
→ PIC32CK SG01/SG00/GC01/GC00 series @120MHz
→ High performance cryptographic accelerators (NDA, NIST FIPS197)
→ Automotive (CAN) + up to 8 sercoms + ethernet
→ ~7.5€/pc (1000pc)

Microchip MPLAB X support for crypto-authenticator modules
→ ATECC608: secure boot
→ ECC204/6: elliptic curve
→ (AT)SHA204/5/6
→ TA100 and TA101: support for TLS
→ Improved support for secure boot, crypto, TLS, etc.

NEW Renesas ARM Cortex M23 (December 2023) @48MHz

NEW GigaDevice (China) ARM Cortex M33:
→ GD32F5 series @200MHz, Embedded World Nuremberg April 2024

PLATFORM SELECTION



1: Application type
High, mid or low-end

2: Required peripherals
#Sercoms, ethernet, wireless, storage, etc,

3: Required security features
Sensitive data? Secure boot

Performance selection shall be similar as before ARMv8

Required peripherals, IO speed, etc. is still very important

MCU selection vs. selection of programming tools

MCU part of family catalog -> upgradability

Secure mechanisms by design!
→ encryption, hashing, secure boot, RoT, etc.

Thank you for your attention

Laurent Segers - laurent.segers@vub.be
An Braeken - an.braeken@vub.be
Bruno da Silva - bruno.da.silva@vub.be
Abdellah Touhafi - abdellah.touhafi@vub.be