TrustedIoT

CORNET TRUSTED-IOT

16 / 11 / 2023

VUB VRIJE UNIVERSITEIT BRUSSEL

KU LEUVEN

TECHNISCHE UNIVERSITÄT DRESDEN

Universität Rostock Traditio et Innovatio

GFaI

SUMMARY

VUB	Environmental monitoring	Heterogeneous embedded architectures
KULeuven	Drones	Multi-core RISC-V
BTU/Rostock	Industry 4.0	Coarse grained reconfigurable architectures (CGRAs)
TUD	Mobile robots	Ultra low-powered (FPGAs)
GFAI	Cooperative robots	Heterogeneous system solutions

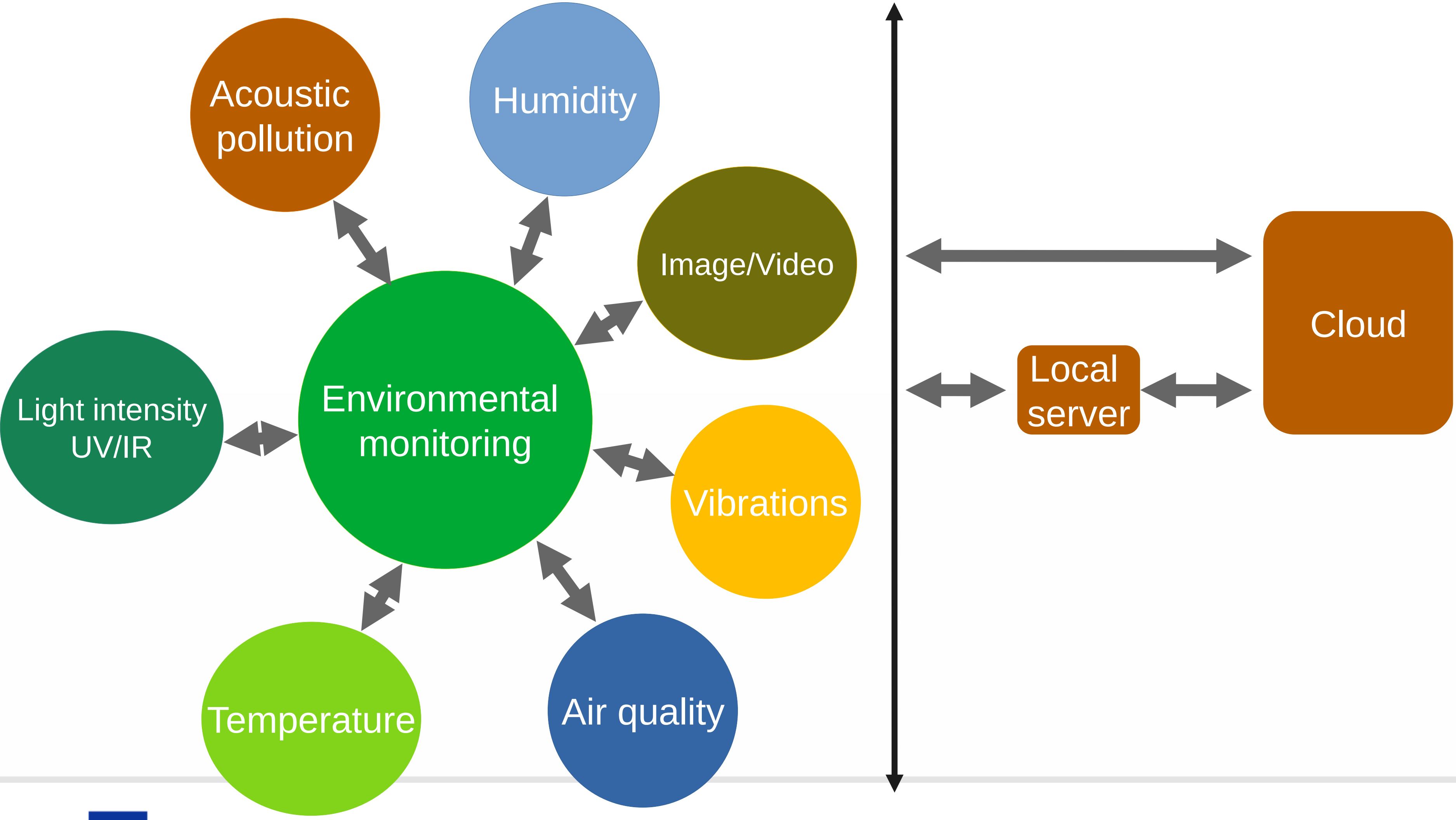
VUB
SECURE EXECUTION FOR EMBEDDED ENVIRONMENTAL MONITORING APPLICATIONS

Laurent Segers
An Braeken
Bruno da Silva
Abdellah Touhafi

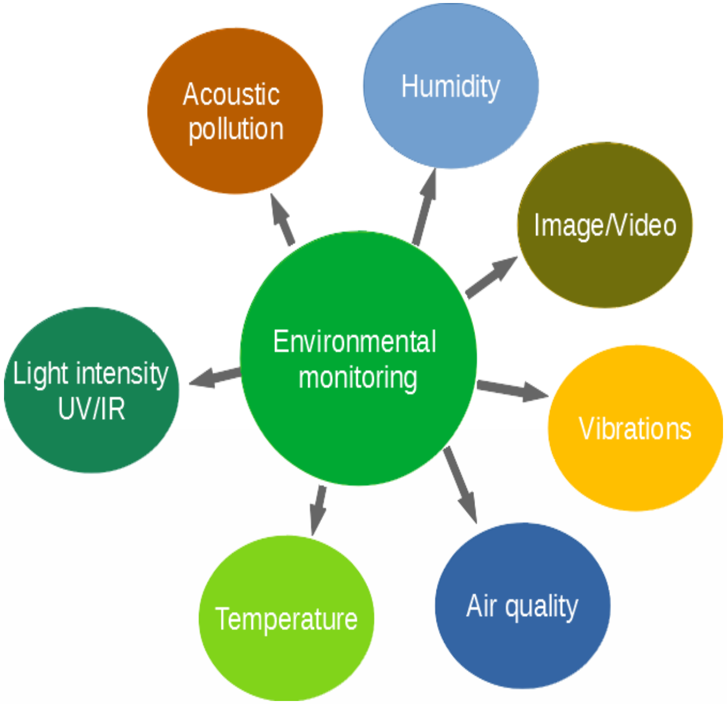
VUB - ENVIRONMENTAL MONITORING

ENVIRONMENTAL MONITORING

TOPOLOGY



ENVIRONMENTAL MONITORING



SECURE LOW-END SENSING

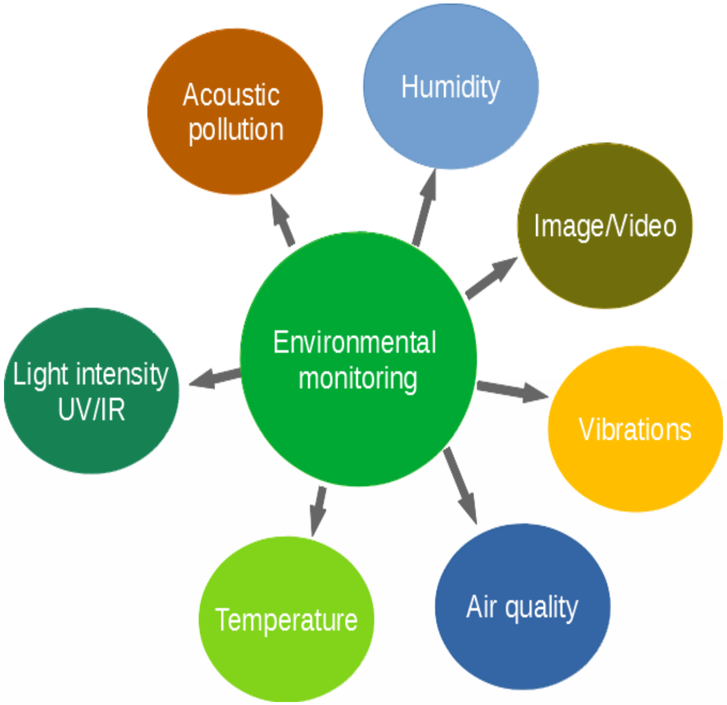
- Limited memory/processing capabilities
- Capable of reading sensors with low update rates (i.e. 1Hz, 10Hz)
- Data integrity & confidentiality of sensor-readouts
- Trusted GPS & RTC

SECURE REMOTE UPDATE

- Lightweight key agreement protocol using PUF
- Secure attestation

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*

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
- Secure bootloaders
- Trusted peripherals (when possible)
- Optimizations for secure storage
- Secure over the air updates

ENVIRONMENTAL MONITORING

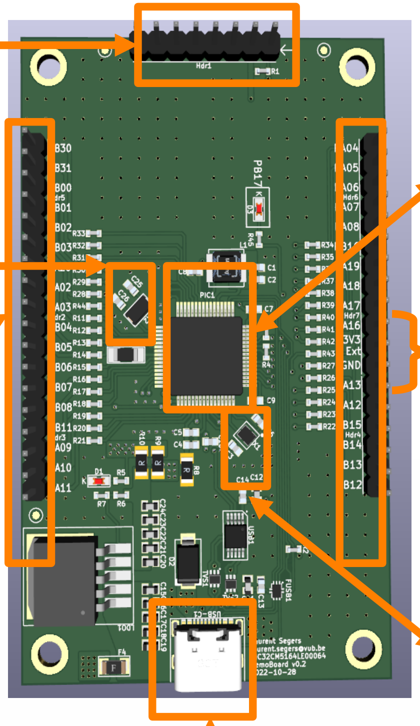
LOW-END DEVICES – TRUSTED EXECUTION ENVIRONMENT

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)

LOW-END SENSING

MICROCHIP PIC32CM5164 BASED ON ARM23

Custom designed board



Programming header

RTC @ 32kHz

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

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

PIC32CM5164L**S**00064 (secure)

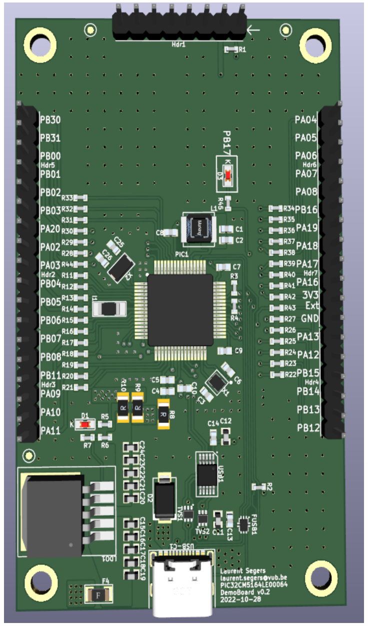
External power

Main crystal @ 32MHz

USB for power over USB + commucation to PC

LOW-END SENSING

MICROCHIP PIC32CM5164 ARM23 LOW-END EMBEDDED PLATFORM



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

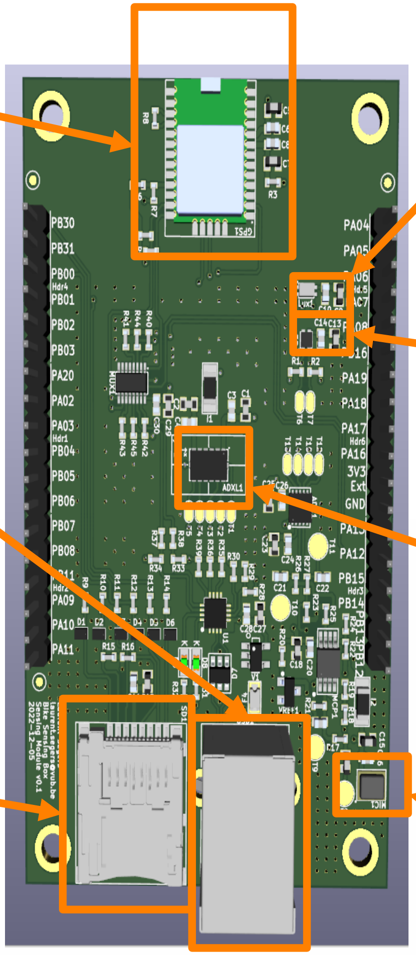
Unique 128-bit serial number

Separate registers for secure and non-secure application

LOW-END SENSING

SENSOR MODULE

Grouping sensors in secure/non-secure peripherals



GPS (L96-M33)

VEML3328
light sensor (RGB+IR)

SHT41 temperature +
humidity sensor

UART to USB
communication to PC

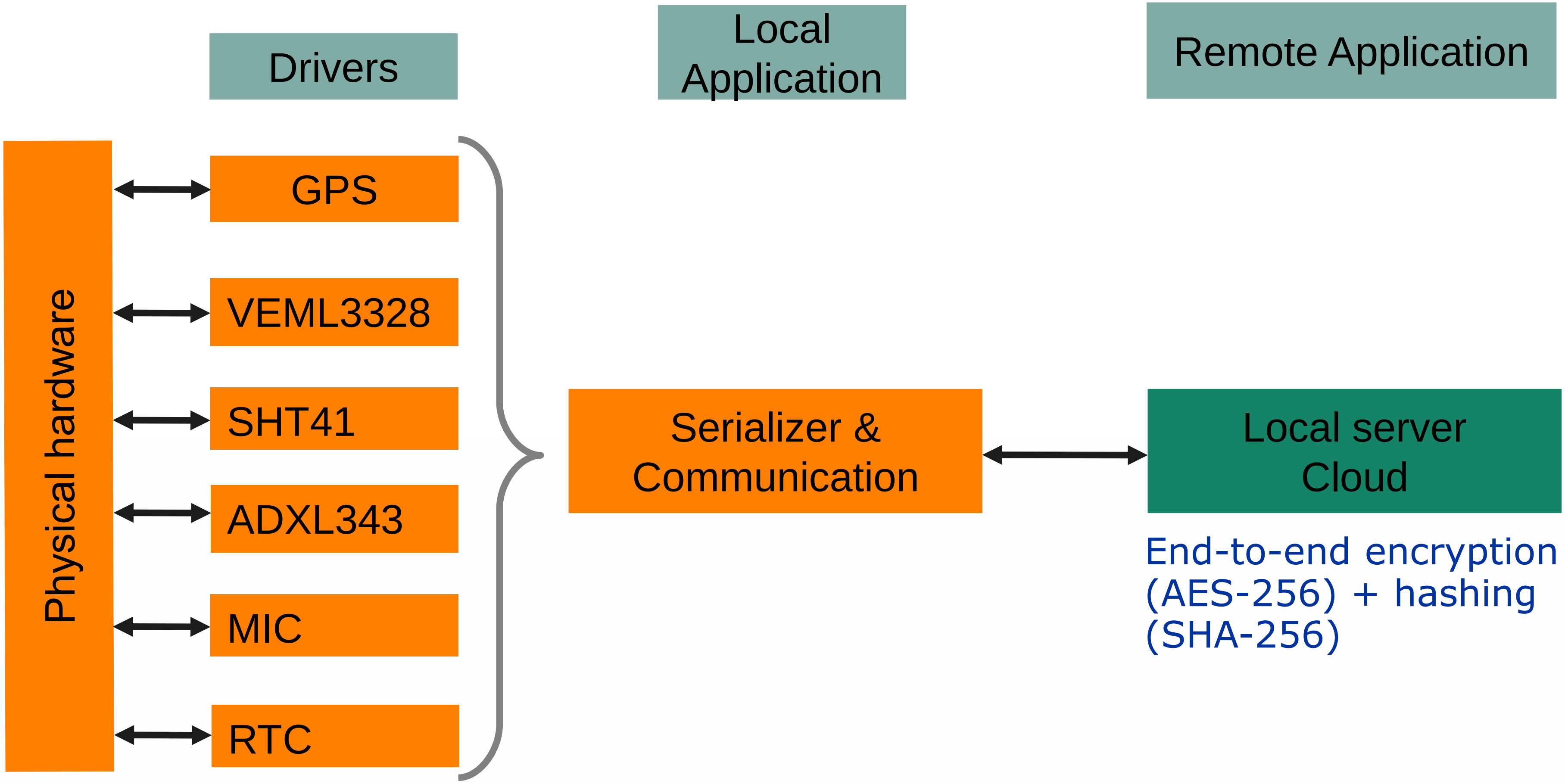
ADXL343 3-axis
accelerometer

SD-card for logging

SPU0410LR5H-QB
analog microphone
+ SPI ADC

LOW-END SENSING

EMBEDDED FIRMWARE (1): MODULAR APPROACH WITHOUT TRUSTZONE



Drivers

Local Application

Remote Application

Physical hardware

GPS

VEML3328

SHT41

ADXL343

MIC

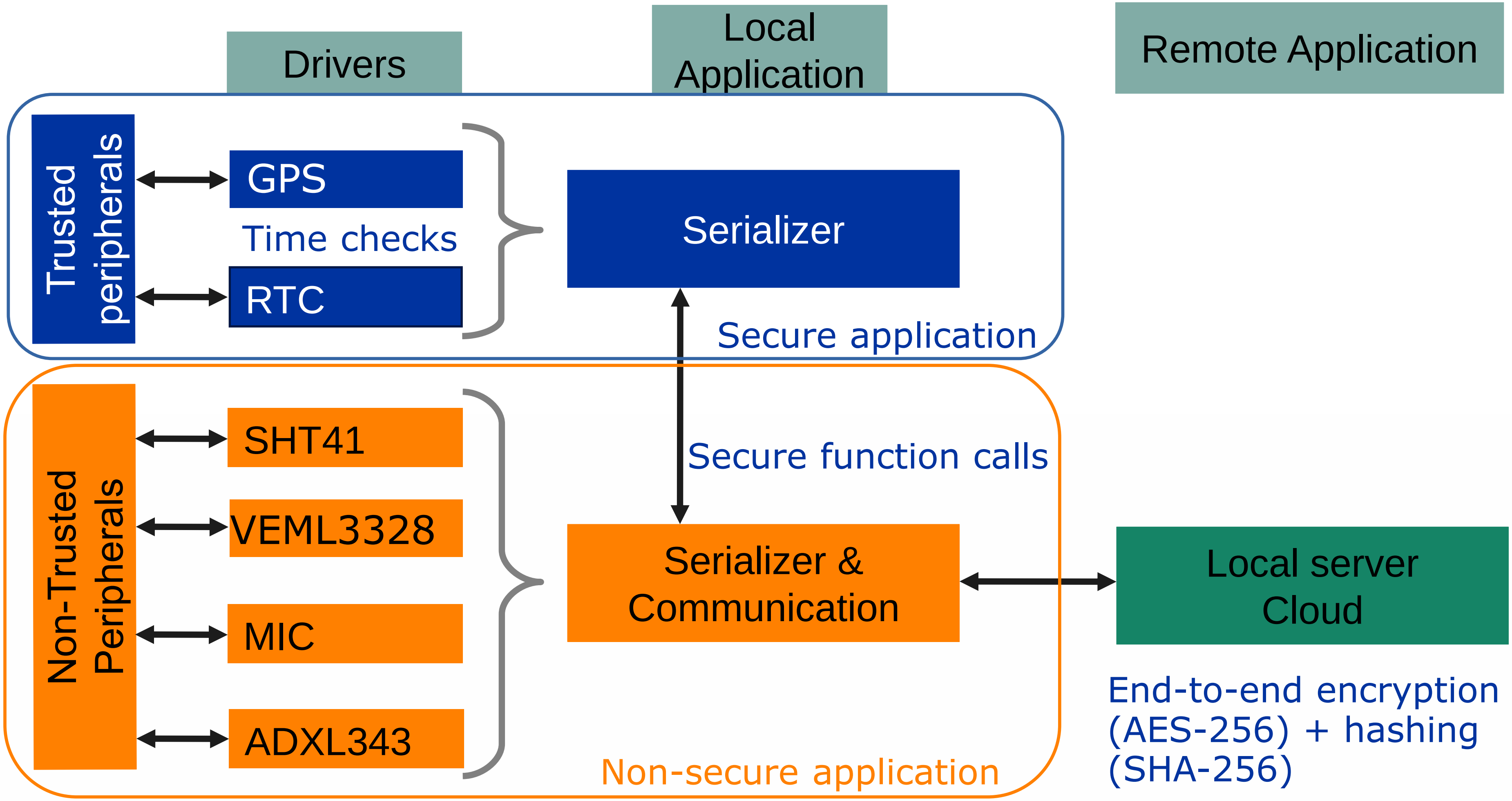
RTC

Serializer & Communication

Local server Cloud

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

EMBEDDED FIRMWARE (2): MODULAR APPROACH WITH TRUSTZONE



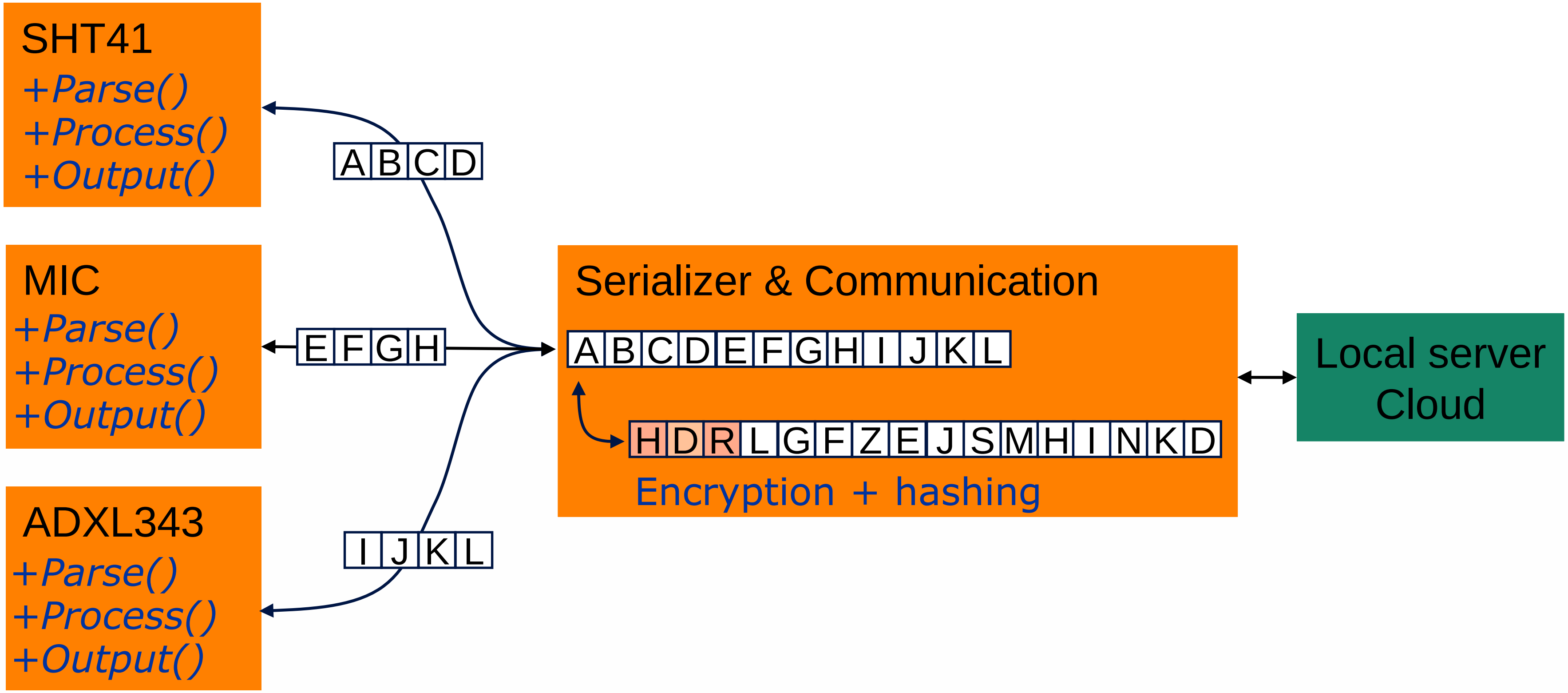
End-to-end encryption (AES-256) + hashing (SHA-256)

EMBEDDED FIRMWARE & TRUSTZONE (3): CONSIDERATIONS

- 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
- 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

LOW-END SENSING

EMBEDDED FIRMWARE (4): CODE-WISE



SHT41
+Parse()
+Process()
+Output()

MIC
+Parse()
+Process()
+Output()

ADXL343
+Parse()
+Process()
+Output()

A B C D

E F G H

I J K L

Serializer & Communication

A B C D E F G H I J K L

H D R L G F Z E J S M H I N K D

Encryption + hashing

Local server
Cloud

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

LOW-END SENSING

ADDITIONAL RESOURCE CONSUMPTION

Code execution time / power overhead TrustZone

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

Program code overhead due to TrustZone

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

Secure data transmission

- → Data sent in “plain readable” format: ~34-80 bytes per packet
  → AES-256 CBC encryption + IV: ~17-32 bytes additional
  → SHA-256 hashing: 32 additional bytes
- → Total overhead: 49-64 bytes => 100% on average

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...

Silent auto-updates
→ project discrepancies
→ compiler flag discrepancies
→ new project then required

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

LOW-END SENSING

SUMMARY & NEXT STEPS

- ✓ Microchip ARM23 based platform selected and programmed
- ✓ TrustZone and secure remote communication
- ✓ Firmware development challenges
- ➔ Fine-grained impact analysis of TrustZone and secure communication
- ➔ Remote (secure) programming of application
- ➔ Lightweight key agreement protocol using PUF
- ℹ Limitations of programming tools & resolution

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