Root-of-Trust and Trusted Execution Environment

Crosscon and (secure) friends

28th June 2024

Macarena C. Martínez-Rodríguez [1], Nicola Tuveri [2]

[1] Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC-US, macarena@imse-cnm.csic.es

[2] Tampere University, nicola.tuveri@tuni.fi

IMSE -cnm Instituto de Microelectrónica de Sevilla

Tampere University

Goal and Outline

Goal

Implementation of a hardware **Root-of-Trust** (RoT) for secure identity and cryptographic operations coupled with an open-source **Trusted Execution Environment (**TEE) to protect trusted applications.

Outline

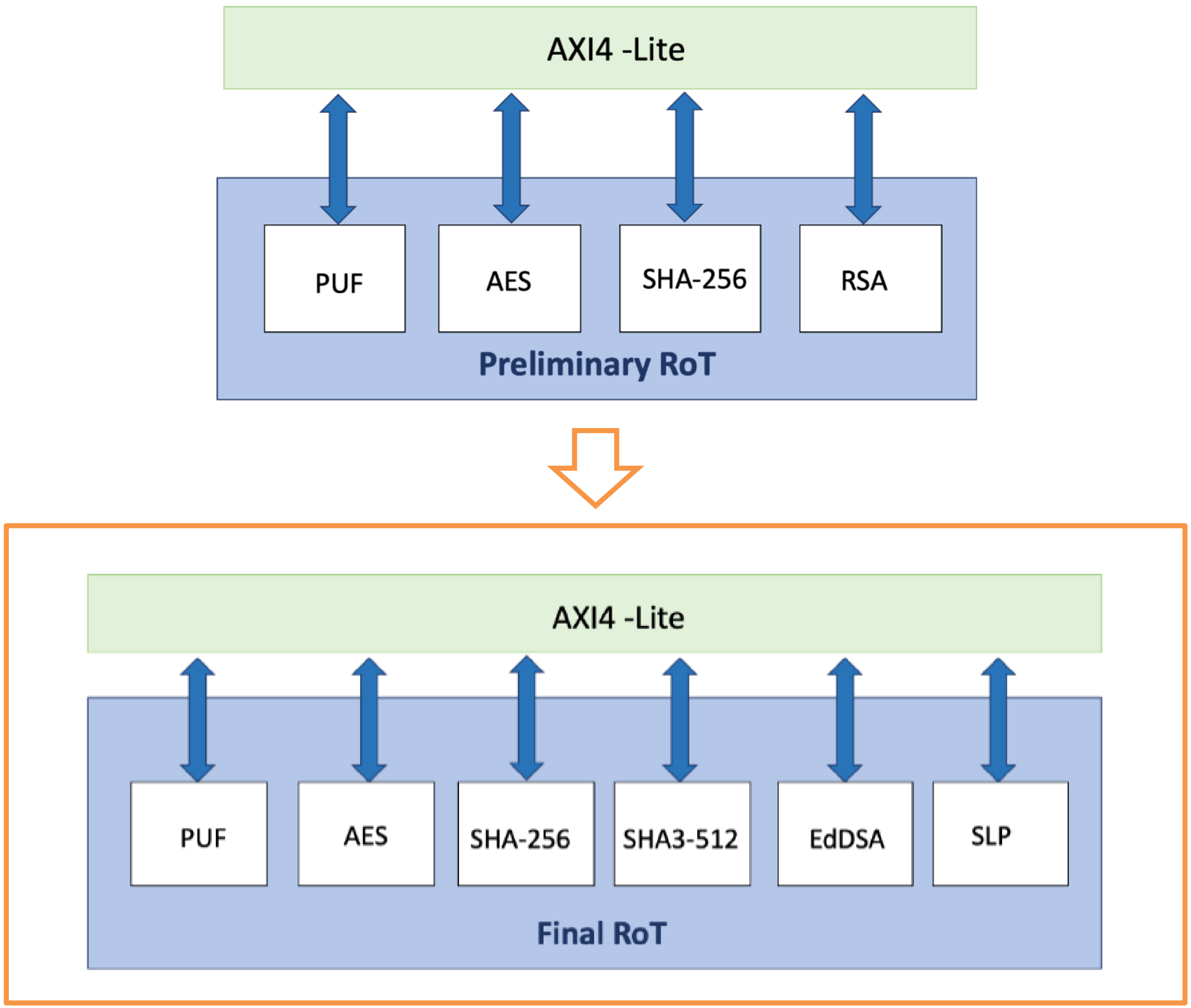
- HW Root-of-Trust
- Components of the HW RoT
- SPIRS platform using programmable logic
- Integration with the TEE
- HW RoT Protection
- Software integration within TEE

Hardware Root-of-Trust

- Implementation of a hardware **Root-of-Trust** (RoT) for secure identity and cryptographic operations.
- RTL description of each module is tecnnologically indepent.
- Each IP module is provided with AXI standard interface.
- All of them are compliant with the NIST test vector

[1] Rojas-Muñoz, L.F. *et al.* (2024). Cryptographic Security Through a Hardware Root of Trust. In: Skliarova, I., Brox Jiménez, P., Véstias, M., Diniz, P.C. (eds) Applied Reconfigurable Computing. Architectures, Tools, and Applications. ARC 2024. Lecture Notes in Computer Science, vol 14553. Springer, Cham. https://doi.org/10.1007/978-3-031-55673-9_8

Preliminary and final RoT

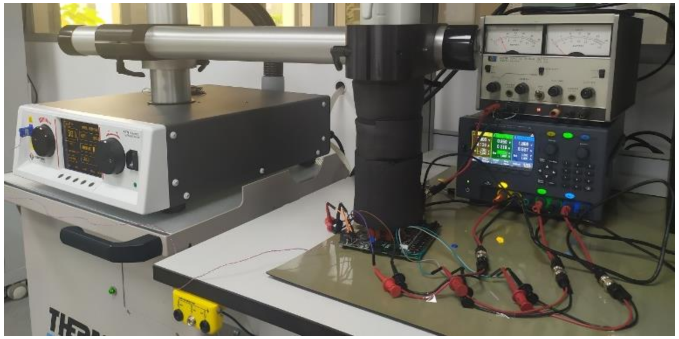


Physical Unclonable Functions (PUF)

IMSE-cnm Instituto de Microelectrónica de Sevilla · Tampere University · SPIRS

- Constrained-resource architecture based on Ring oscillators.
- Functionality:
  - ID

HDintra	HDinter
0.03	47.5-48.5



  - TRNG

NIST Test 800-22 ✓	NIST Tests 800-90b ✓	AIS 31 Test ✓

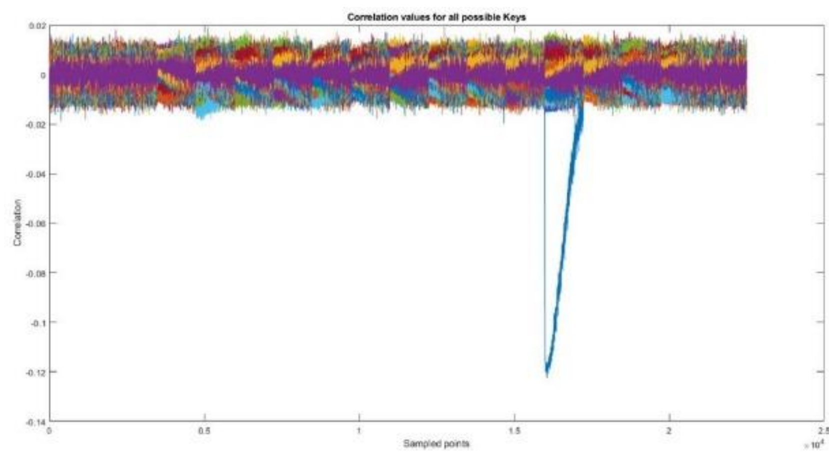
- More compact.
- It includes **countermeasures** against Electromagnetics attacks.
- Performance evaluated under variations voltaje supply and temperature.

[2] Martínez-Rodríguez, M.C.; Rojas-Muñoz, L.F.; Camacho-Ruiz, E.; Sánchez-Solano, S.; Brox, P. Efficient RO-PUF for Generation of Identifiers and Keys in Resource-Constrained Embedded Systems. Cryptography **2022**, 6, 51. https://doi.org/10.3390/cryptography6040051

[3] Rojas-Muñoz, L.F.; Sánchez-Solano, S.; Martínez-Rodríguez, M.C.; Brox, P. True Random Number Generation Capability of a Ring Oscillator PUF for Reconfigurable Devices. *Electronics* **2022**, *11*, 4028. https://doi.org/10.3390/electronics11234028

- Functionality:

  Symmetric cipher for data encryption and decryption.
- Architecture that implements 128/256 bits AES -ECB.
- AESAVS NIST ✓
- Includes **countermeasures** integrating a signature generator for fault injection attacks (FIA) and Leakage-Resilient Masking Scheme for side-channel attacks.



[4] V. Zúñiga-González, E. Tena-Sanchez and A. J. Acosta, "A Security Comparison between AES-128 and AES-256 FPGA implementations against DPA attacks," *2023 38th Conference on Design of Circuits and Integrated Systems (DCIS)*, Málaga, Spain, 2023, pp. 1-6, doi: 10.1109/DCIS58620.2023.10336003.

SHA

- Functionality:

  hashing function

- **SHA-2**:
  - Architecture that implement all hash functions within the SHA-2 family.
  - CAVP NIST ✓
  - Enhanced arquitecture.
- **SHA-3**:
  - Architecture for the Keccak function intended for use in the hash functions of the SHA-3 family.
  - CAVP NIST ✓

[5] E. Camacho-Ruiz, S. Sánchez-Solano, M. C. Martínez-Rodríguez and P. Brox, "A complete SHA-3 hardware library based on a high efficiency Keccak design," 2023 IEEE Nordic Circuits and Systems Conference (NorCAS), Aalborg, Denmark, 2023, pp. 1-7, doi: 10.1109/NorCAS58970.2023.10305448.

Digital Signature accelerator

- Functionality:

  HW acceleration of generation and validation of Digital signatures

- RSA accelerator:
  - architecture based on a Karatsuba modular multiplier.
  - NIST Test vector for RSA DS ✓
- EdDSA25519 accelerator:
  - architecture based on a 4-level Karatsuba modular multiplier.
  - Test Vectors provided by the RFC 8032 ✓

System Level protector (SLP)

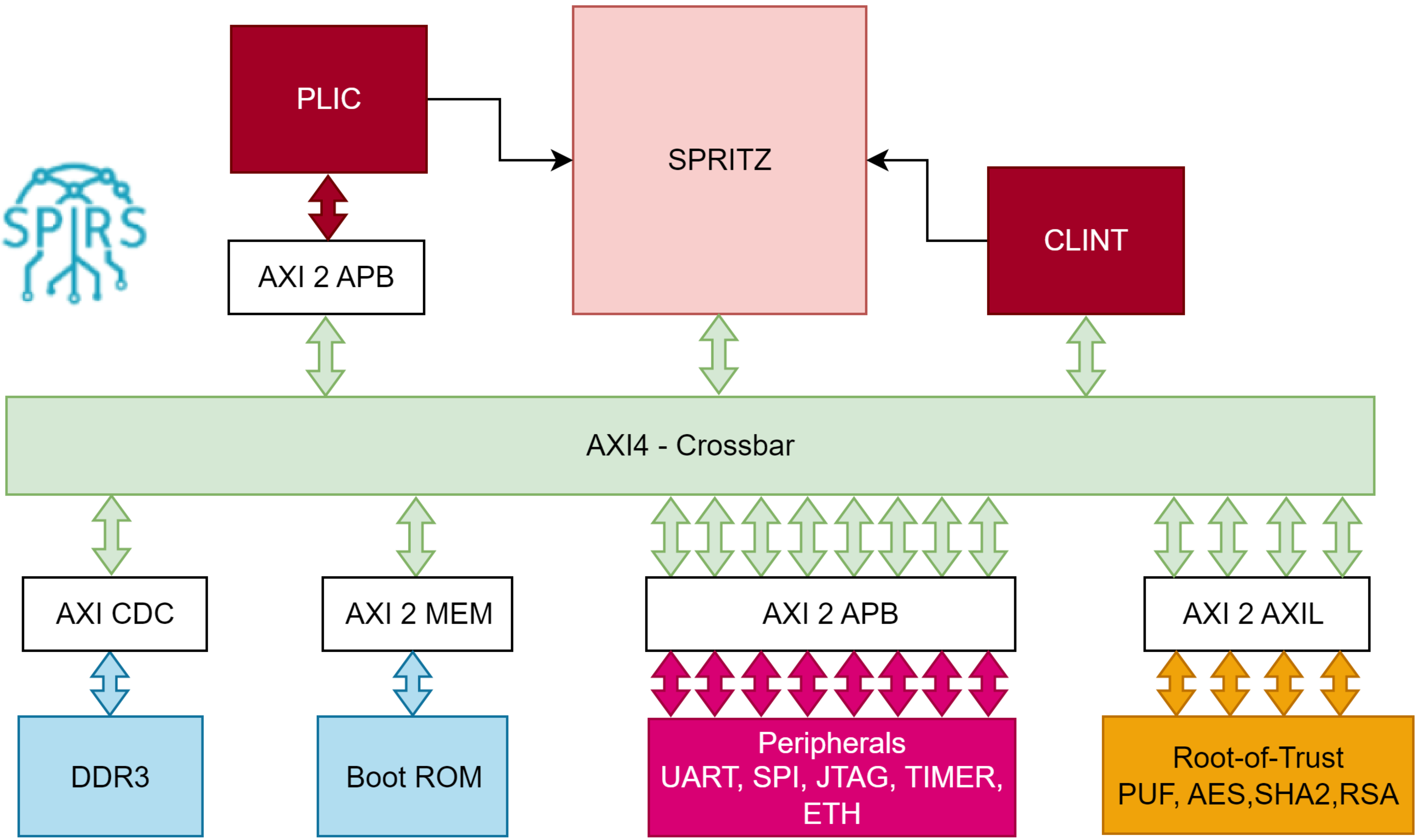
- Functionality:

  prevent FIA attacks across the entire system

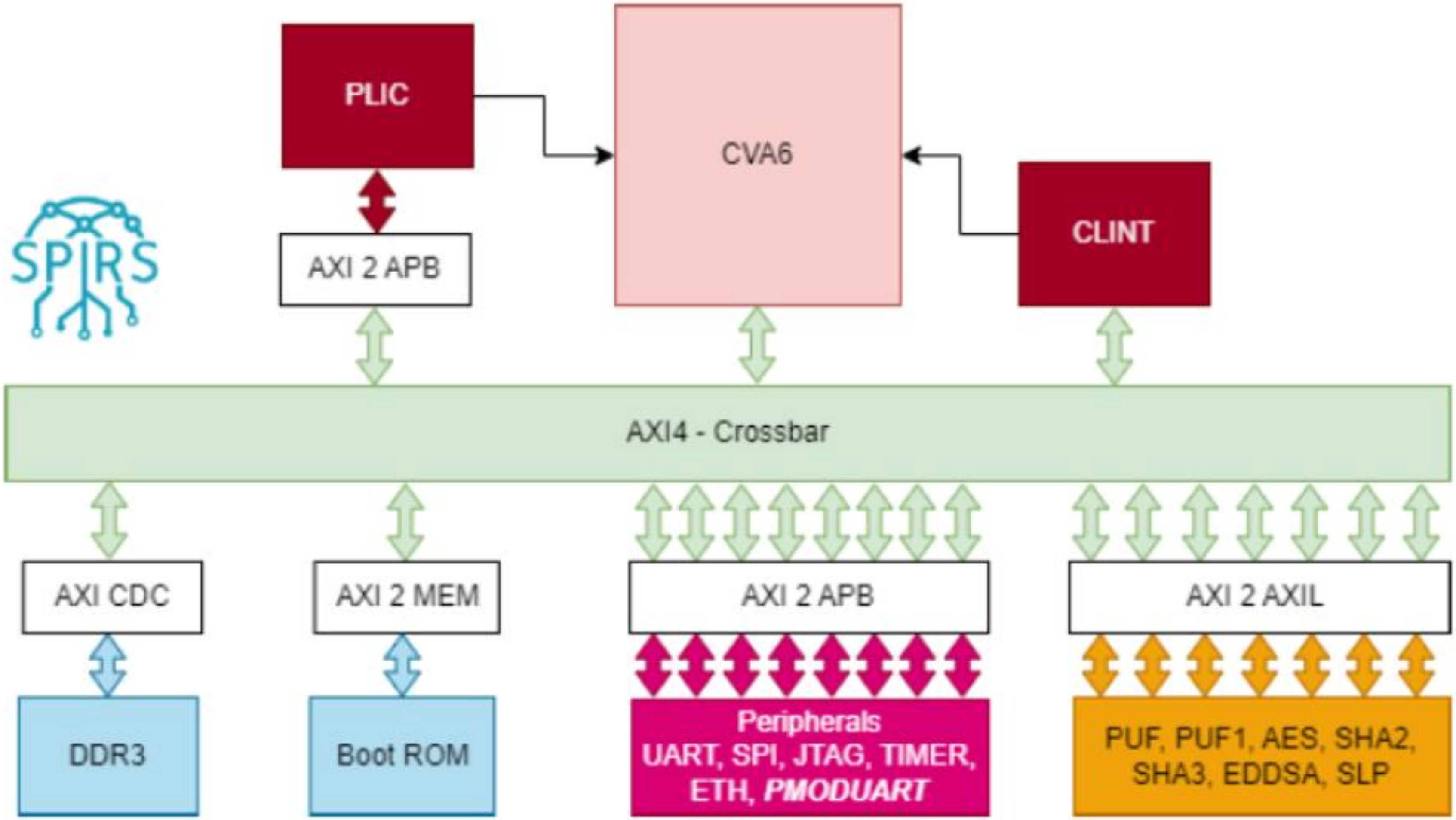
- Architecture to detect
  - signal glitches
  - temperature and voltage out of range

[6] Potestad-Ordóñez, F.E.; Casado-Galán, A.; Tena-Sánchez, E. Protecting FPGA-Based Cryptohardware Implementations from Fault Attacks Using ADCs. *Sensors* **2024**, *24*, 1598. https://doi.org/10.3390/s24051598

Preliminary SPIRS platform

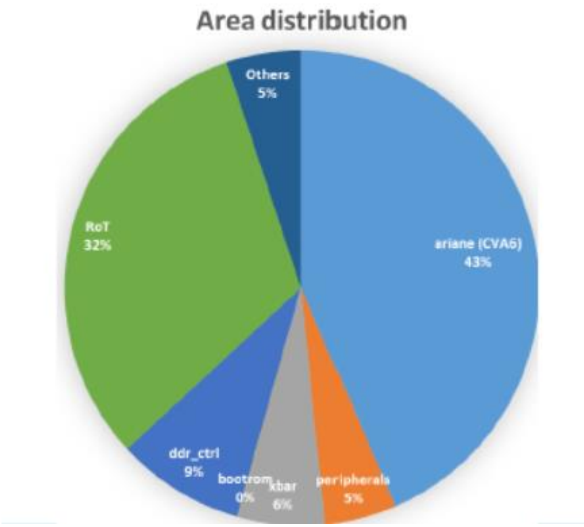


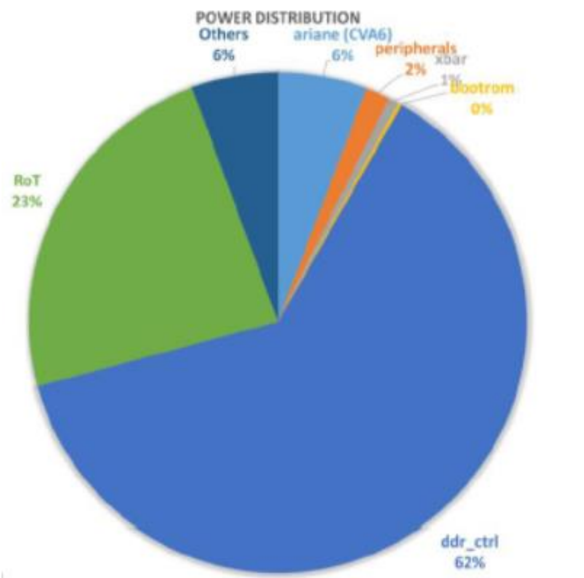
Final SPIRS platform (I)



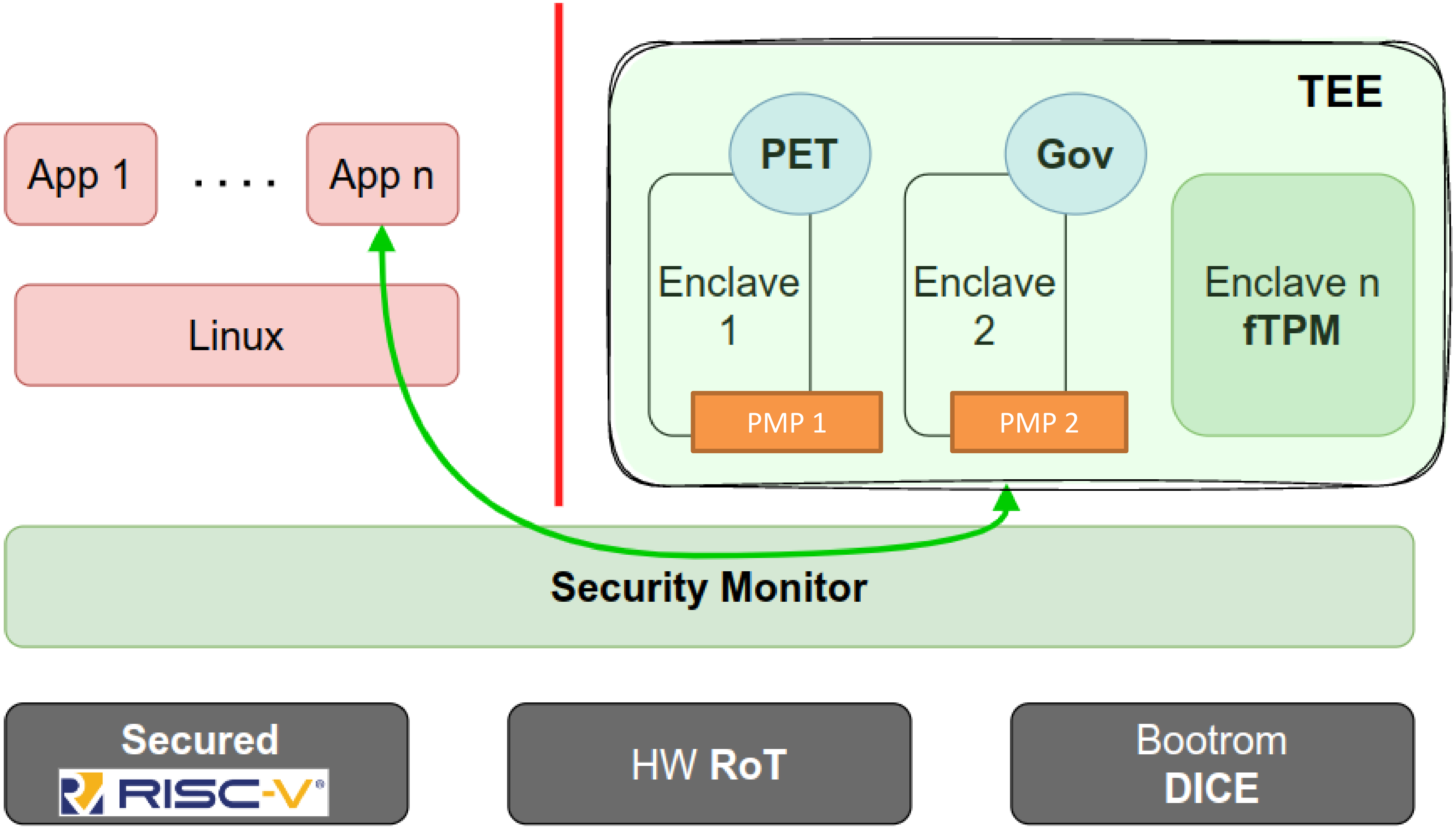
Final SPIRS platform (II)

Area	#LUTs	118630
	#FFs	77173
	#RAM36	81
	#DSP	135
Genesys 2 occupation	LUTs (%)	58.21
	FFs (%)	18.93
	RAM36 (%)	18.20
	DSP (%)	16.07
Power (W)		2.578
Frequency (MHz)		50



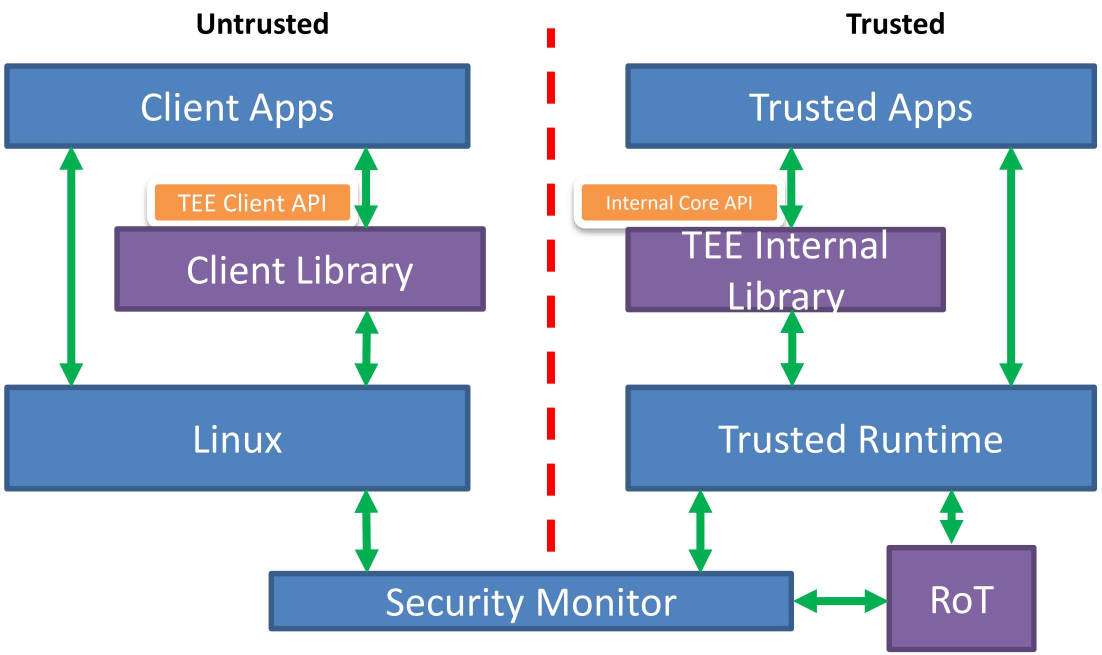


Integration within the TEE



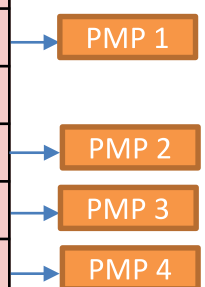
SPIRS TEE Design

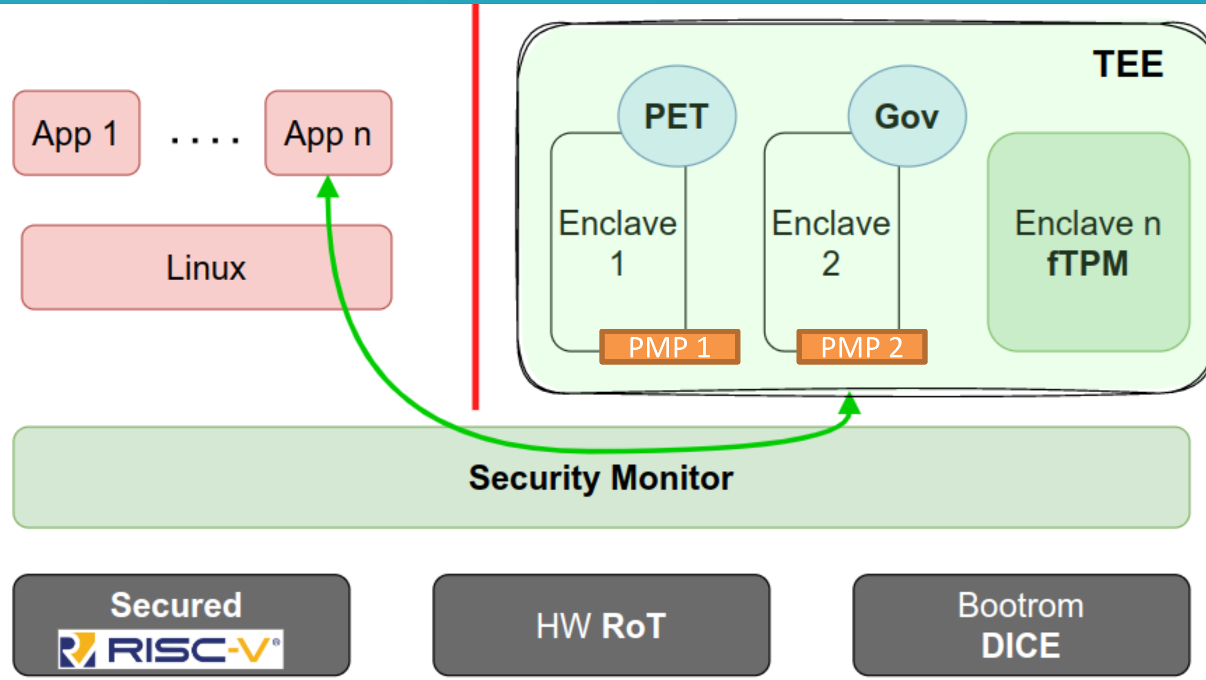
- Based on the open-source project: **Keystone**
- **GlobalPlatform** TEE API
  - TEE Client API (v1.0)
  - TEE Internal Core API (subset of, v 1.1.2)
  - Reducing the gap between **RISC-V** and **ARM** ecosystems



HW RoT Protection

Base	Description
0x0000_0000	Debug Module
0x0001_0000	ROM
...	
0x1000_0000	UART
0x1800_0000	TIMER
0x3000_0000	ETHERNET
0x1000_2000	UART2
0x4100_0000	**AES**
0x4200_0000	PUF1/TRNG
0x4300_0000	**SHA2**
0x4400_0000	**SHA3**
0x4500_0000	**EdDSA**
0x4700_0000	SLP
0x4800_0000	PUF2
...	
0x8000_0000	DRAM





- Memory mapped devices
- Only selected blocks are reachable
  - No PUF / TRNG, SM-exclusive
- Isolation achieved through PMP

Software integration within TEE (I)

- **Leverages the *RoT Library for SPIRS Platform (v4.3)***
  1. **Developed in WP2**
  2. **Available at: gitlab.com/hwsec/lib_rot_spirs**
  3. **API for accessing HW RoT cores**
     - **Linux Userspace apps**
     - **SPIRS TEE Trusted Applications**
  4. **Includes a demo for:**
     - **AES**
     - **SHA2**
     - **SHA3**
     - **EdDSA accelerator**
     - **PUF/TRNG**

Software integration within TEE (II)

- **Leverages the *RoT Library for SPIRS Platform (v4.3)***
- **Allow TAs to access non-critical HW RoT cores**
  - **AES, SHA2, SHA3, EdDSA**
- **Maintain high level API**
  - **Same source used by a Linux application can be used from a TA**
  - **Low-level API changes only**

```
MMIO_WINDOW win_sha3_512;

createMMIOWindow(&win_sha3_512, BASEADDR_SHA3_512,  MS2XL_LENGTH);

sha3_512(buf_in_sha3_512, buf_out_sha3_512, length_sha3_512, win_sha3_512, 0);

closeMMIOWindow(&win_sha3_512);
```

- **Published in SPIRS Keystone repository v1.6.0**
  - **Based on PR#418@Keystone (upstream)**
  - **Demo included in SPIRS TEE SDK repository**

Questions & Answers