Technology
Innovation
Institute

seL4 on RISC-V: Building a Trusted Execution Environment

Everton de Matos

Introduction

- With the **exponential growth of connected devices** and the constant threat of cyber-attacks, there's never been a more crucial time to **ensure that our computational environments are trustworthy**

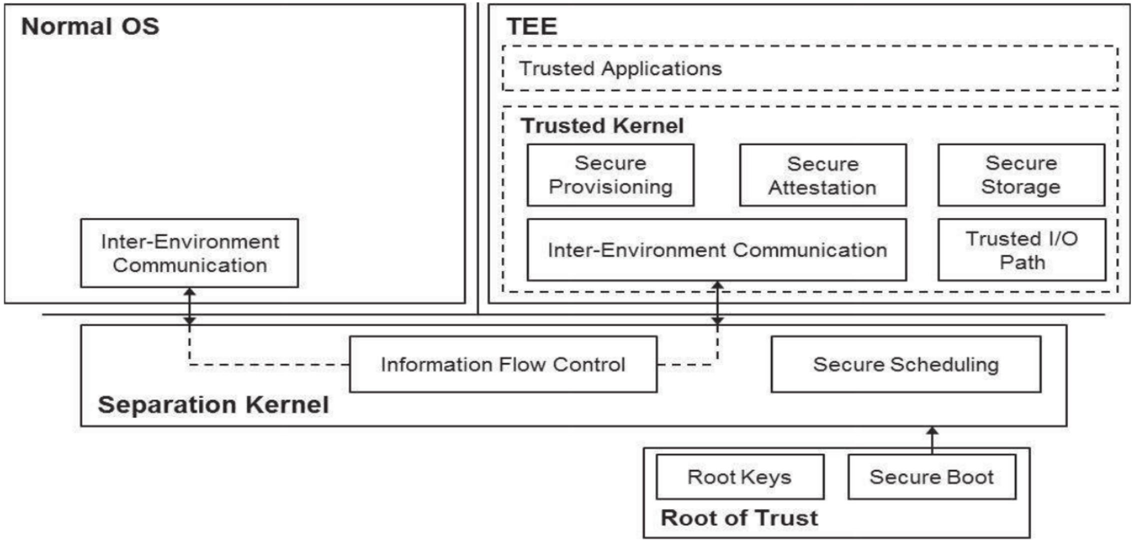
- While hardware security mechanisms and traditional software barriers have their roles, there are **gaps that need to be addressed** to ensure absolute trust in our digital environments

- The necessity for **robust security** solutions is more pronounced than ever. **TEEs** stand as **one of the possible solutions** to meet intricate security needs

- In this landscape, **seL4** presents itself as a strong candidate to anchor a **secure operating system within a TEE**, offering a robust foundation to build trusted digital environments

Trusted Execution Environments

- **Trusted Execution Environments (TEEs)** provide a **secure execution environment** for sensitive applications and data, ensuring that they are protected from attacks and unauthorized access
- TEEs are typically implemented as a separate execution environment within a system, with their own hardware and software resources that are **isolated from the rest of the system**



[1] Sabt, Mohamed, Mohammed Achemlal, and Abdelmadjid Bouabdallah. "Trusted execution environment: w hat it is, and w hat it is not." 2015 IEEE Trustcom/BigDataSE/Ispa. Vol. 1. IEEE, 2015.

Trusted Execution Environments – Use cases

- **Secure Storage**
  - Sensitive data can be stored safely, isolated from the main operating system

- **Secure Execution of Code**
  - Code can be executed in a protected and isolated environment, ensuring the integrity of the operations

- **Cryptographic Operations**
  - Encryption, decryption, and digital signing

- **Remote Attestation**
  - Remote verification of software's integrity and authenticity

- **Secure Multi-party Computation**
  - Parties can jointly compute a function over their inputs while keeping these inputs private

Trusted Execution Environments

- **Hardware TEE and Software TEE**

- ***ARM TrustZone***
  - **Hardware extensions** for ARM processors
  - Allows the device to run in **two different** processor modes, called **normal world** mode and **secure world** mode
  - Involves **software components** to manage and utilize the hardware features effectively

- ***Intel SGX*** *(Software Guard Extensions) – x86*
  - **Hardware-based memory encryption** that isolates specific application code/data in memory
  - While the functioning and management of **private memory regions** are controlled by **software instructions**, the secure and isolated environment is facilitated by underlying hardware features of the CPU

seL4 TEE – Related Work

- TEEVseL4: Trusted Execution Environment for Virtualized seL4-based Systems [2]

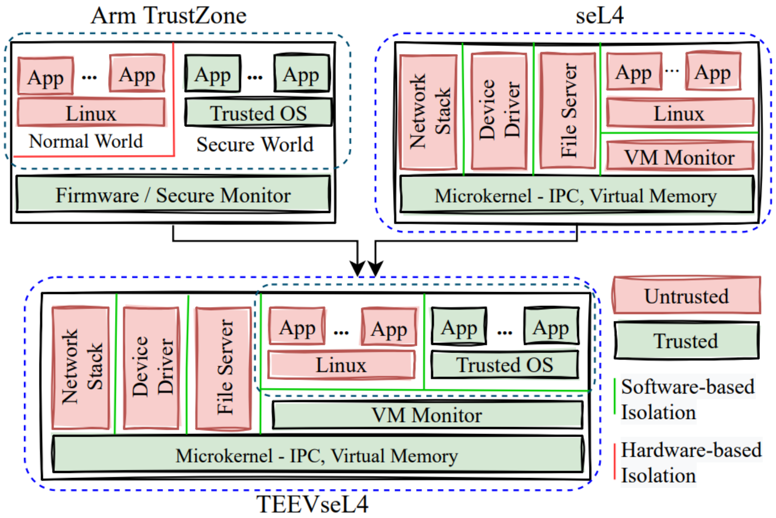


Fig. 1: The TEEVseL4 system architecture, leveraging microkernel (seL4) and Arm TrustZone-compatible software solutions, provides a trustworthy virtualization system with a TrustZone-compatible TEE for secure isolation of security-critical functions.

- MicroTEE: Designing TEE OS Based on the Microkernel Architecture [3]

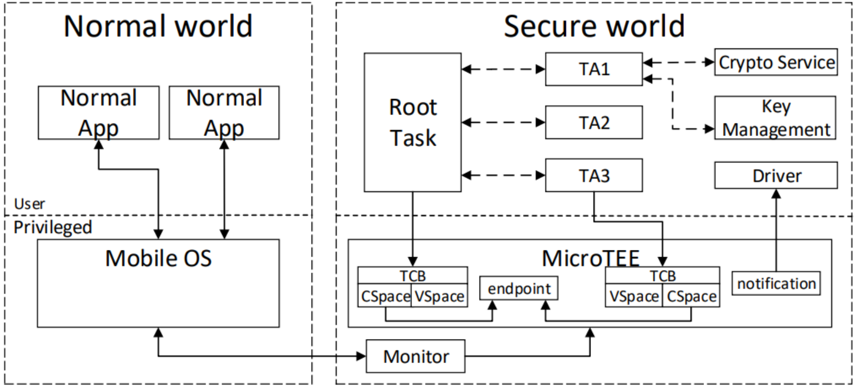


Fig. 2. The Architecture of MicroTEE

[2] Blazevic, B., Peter, M., Hamad, M., & Steinhorst, S.. "TEEVseL4: Trusted Execution Environment for Virtualized seL4-based Systems." 2023 IEEE RTCSA 23.

[3] Ji, D., Zhang, Q., Zhao, S., Shi, Z., & Guan, Y. (2019, August). Microtee: designing tee os based on the microkernel architecture. In 2019 18th IEEE TrustCom (pp. 26-33).

TEE on RISC-V

- **HEX-Five Multizone** [4]
    - Provides hardware-enforced, software-defined separation of multiple security domains, thus enabling isolation in separate "zones"
- **Penglai** [5]
    - Enclave framework, providing a mechanism to run trusted applications in an isolated environment
    - Designed to leverage the hardware isolation features provided by the RISC-V architecture, such as PMP
- **Keystone** [6]
    - Provides customizable TEEs
    - Provided example scenarios:
        - seL4 being used in S mode inside an enclave
        - seL4 being used in M mode as Security Monitor

[4] HEX-Five. https://hex-five.com/multizone-security-tee-riscv/

[5] Feng, E., Lu, X., Du, D., Yang, B., Jiang, X., Xia, Y., ... & Chen, H. (2021). Scalable Memory Protection in the {PENGLAI} Enclave. In 15th USENIX OSDI (pp. 275-294).

[6] Lee, D., Kohlbrenner, D., Shinde, S., Asanović, K., & Song, D. (2020, April). Keystone: An open framework for architecting trusted execution environments. In 15th EuroSys (pp. 1-16).

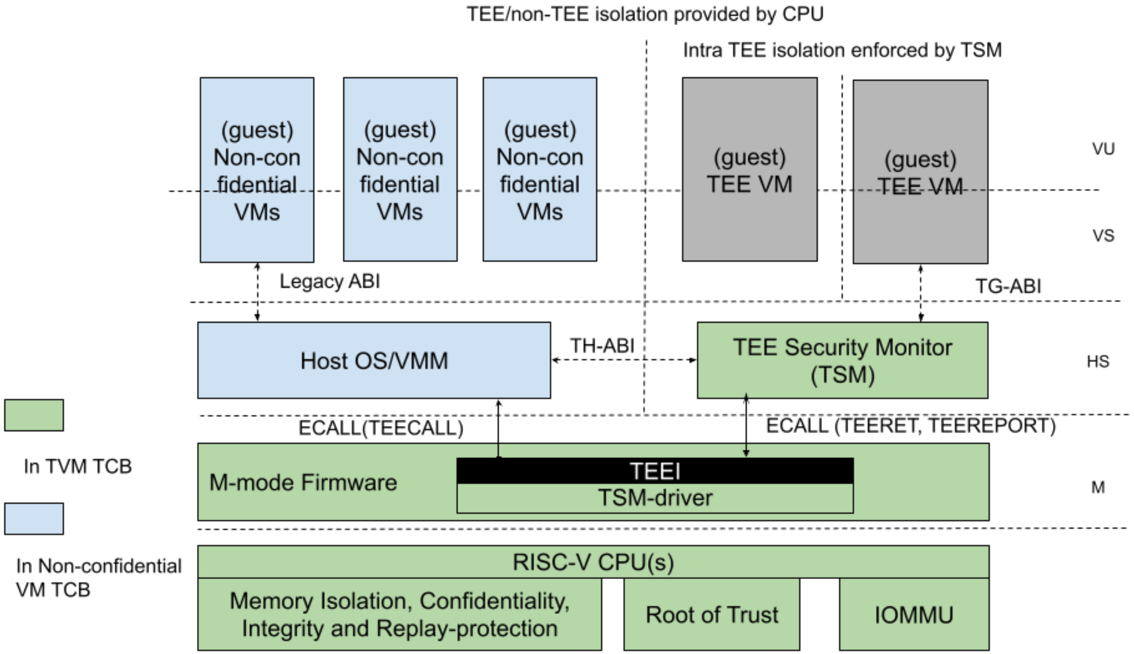
TEE on RISC-V

- **RISC-V Application-Processor TEE (AP-TEE) Task Group Specifications** https://github.com/riscv-non-isa/riscv-ap-tee

  - The AP-TEE extension supports TEE Virtual machines (**TVM**)

  - **New class of hardware-attested trusted execution environment**

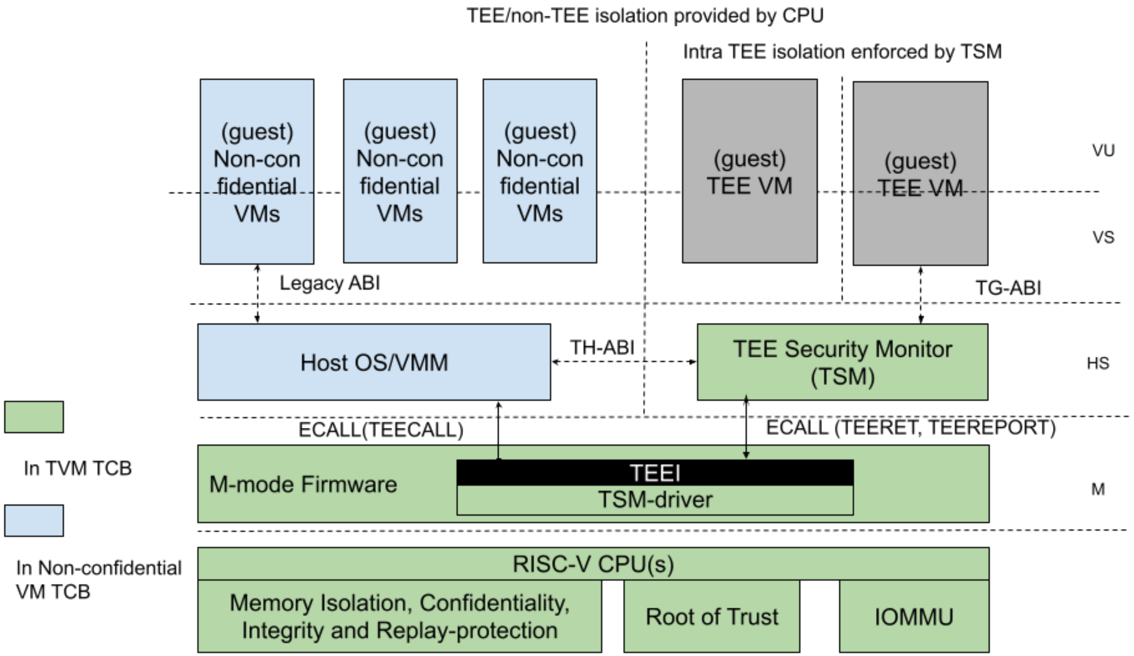
  - Enables the OS or VMM to maintain the role of resource manager (memory, CPU, I/O resources) even for the TVMs


TEE/non-TEE isolation provided by CPU
Intra TEE isolation enforced by TSM
(guest) Non-con fidential VMs
(guest) Non-con fidential VMs
(guest) Non-con fidential VMs
(guest) TEE VM
(guest) TEE VM
VU
VS
Legacy ABI
TG-ABI
Host OS/VMM
TH-ABI
TEE Security Monitor (TSM)
HS
ECALL(TEECALL)
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TEEI
TSM-driver
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU
In TVM TCB
In Non-confidential VM TCB

TEE on RISC-V

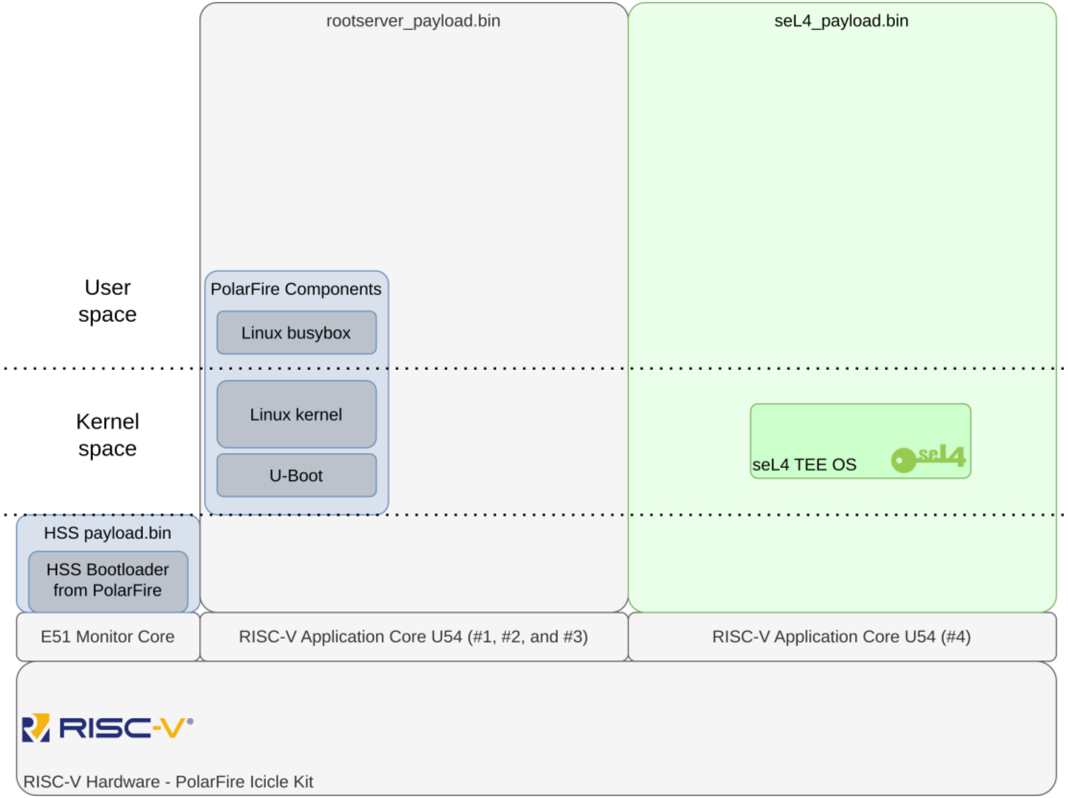
- **RISC-V Application-Processor TEE (AP-TEE) Task Group Specifications** https://github.com/riscv-non-isa/riscv-ap-tee

    - "TEE Security Monitor" (**TSM**) **HS-mode** software module
    - Acts as the trusted intermediary between the VMM and the TVM
    - The **TSM implements a set of TEE "flows"** that are accessed via a Trusted Execution Environment Interface (**TEEI**) ABI hosted by a Trusted Security Manager Driver (**TSM Driver**)


🟩 In TVM TCB
🟦 In Non-confidential VM TCB
TEE/non-TEE isolation provided by CPU
Intra TEE isolation enforced by TSM
(guest) Non-confidential VMs
(guest) Non-confidential VMs
(guest) Non-confidential VMs
(guest) TEE VM
(guest) TEE VM
VU
VS
Legacy ABI
TG-ABI
Host OS/VMM
TH-ABI
TEE Security Monitor (TSM)
HS
ECALL(TEECALL)
ECALL (TEERET, TEEREPORT)
M-mode Firmware
TEEI
TSM-driver
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU

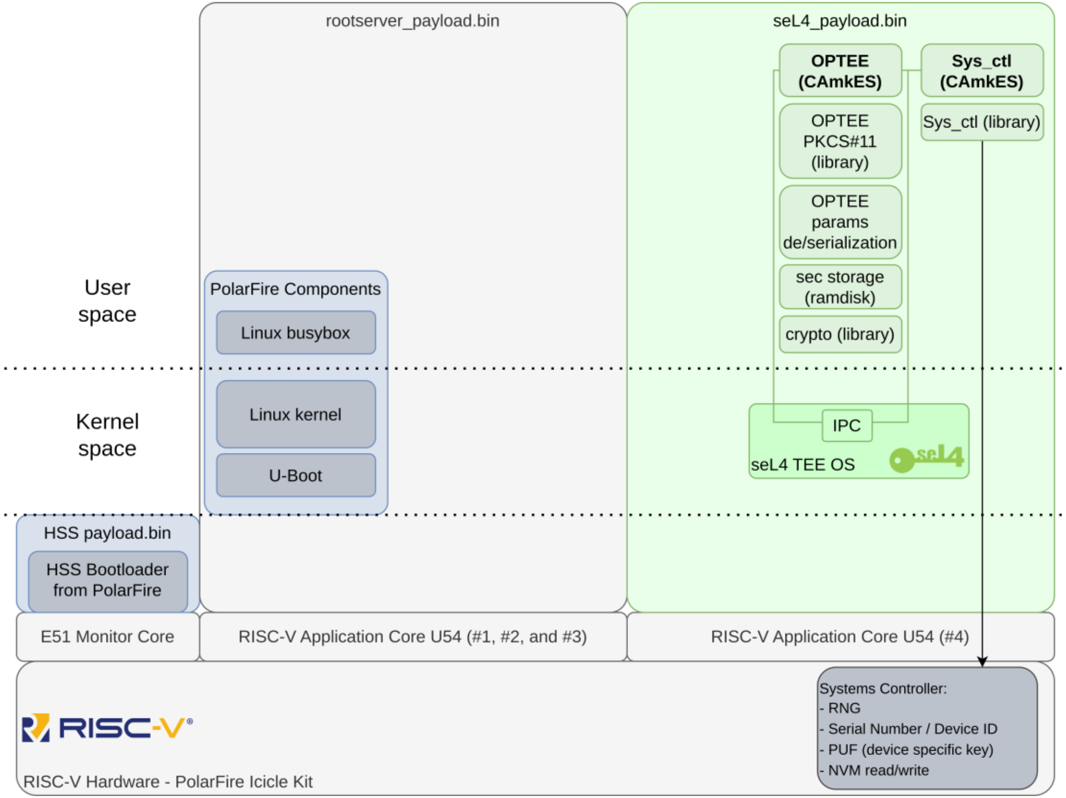
seL4 TEE on RISC-V – Our approach

- **PolarFire SoC FPGA Icicle Kit**
  - 4x U54 Application cores
    - RV64GC
  - 1x E51 Monitor core
    - RV64IMAC
- AMP – PMP configuration
  - 3x U54 – Linux
  - 1x U54 – seL4
- HSS Bootloader
  - E51 Monitor Core



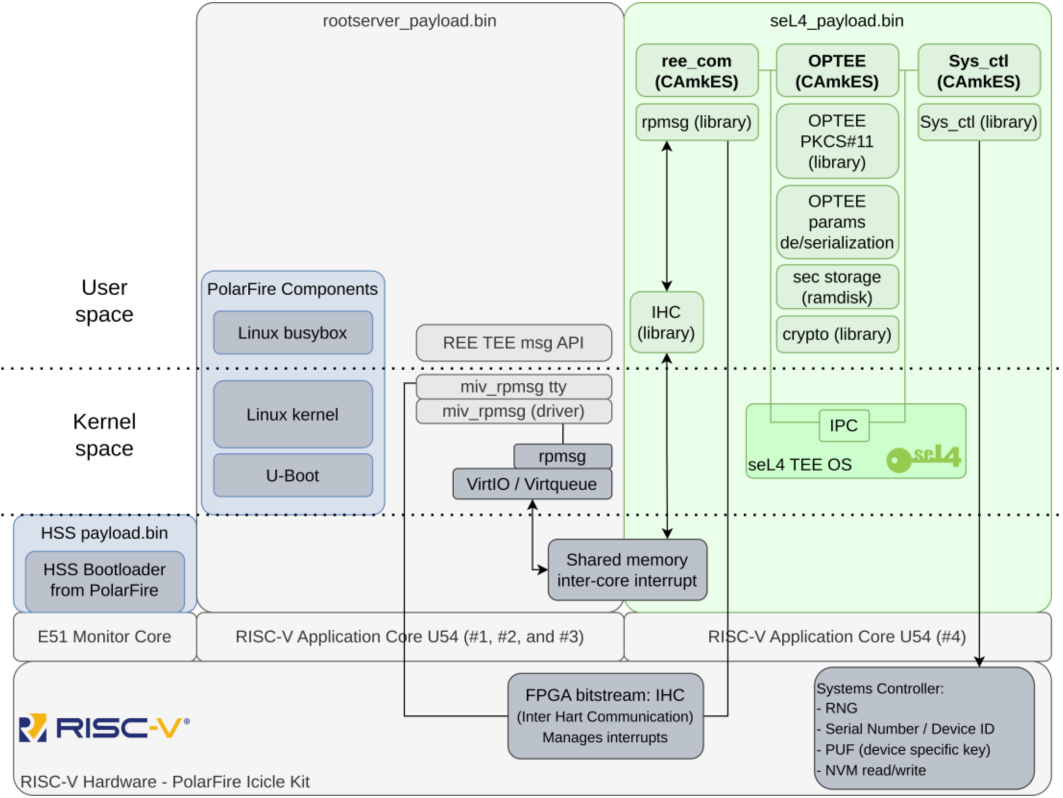
seL4 TEE on RISC-V – Our approach

- **CAmkES applications**
  - TEE services
  - IPC for communication from/to other CAmkES components
- **OPTEE**
  - Following the GlobalPlatform TEE specifications
  - Crypto / sNVM / PKCS#11
- **Sys_ctl**
  - Uses Polarfire system controller services
  - Support for device ID read, sNVM read/write



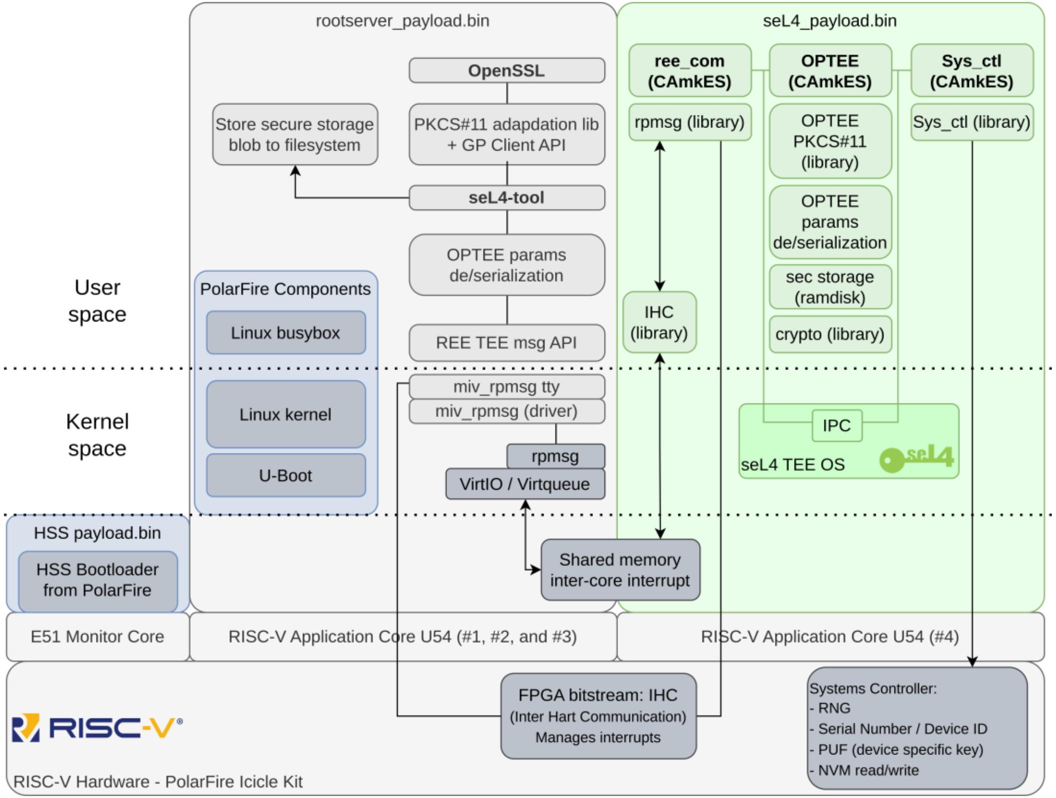
seL4 TEE on RISC-V – Our approach

- **ree_com**
  - CAmkES application
  - IPC for communication from/to other CAmkES components
  - Manages communication messages between Linux ↔ seL4
  - Shared memory
  - Inter Hart Communication handler
  - RPMsg
    - PolarFire SoC RPMsg Linux Support



seL4 TEE on RISC-V – Our approach

- **seL4-tool**
  - Test tool for seL4 TEE
  - Used for initial demonstration and testing seL4 TEE services
  - Running on Linux (REE)
  - Uses the *seL4_TTY_rpmsg* (TEE) driver for communicating between Linux (REE) and seL4 (TEE)
  - Examples:
    - Random number from seL4 TEE
    - Write/Read sNVM
    - Generate keys
    - PUF test



Next steps

- **Short term – within this year:**
    - Open source
    - Performance evaluation
        - Crypto, sNVM
    - Scientific Paper – in progress
- **Mid/Long term:**
    - Monitor RISC-V AP-TEE TG Specifications
    - Enable additional PolarFire Icicle hardware security features
        - Secure Boot, Device-Level Anti-Tamper Features, etc.
    - Transition from CAmkES to Microkit
    - **TEE at HS mode on our RISC-V SoC**

Technology
Innovation
Institute

Thank you!

seL4 on RISC-V: Building a Trusted Execution Environment

Everton de Matos