SiFive

Porting OP-TEE on RISC-V: A Practical Guide and Introduction

Peter Lin | RISC-V Summit China 2025

RISC-V Software Ecosystem (RISE) Project

The development of OP-TEE for RISC-V is currently managed by the RISE **Security Software Working Group**, GitLab repositories https://gitlab.com/riseproject/riscv-optee hosting key components of the RISC-V software stack.

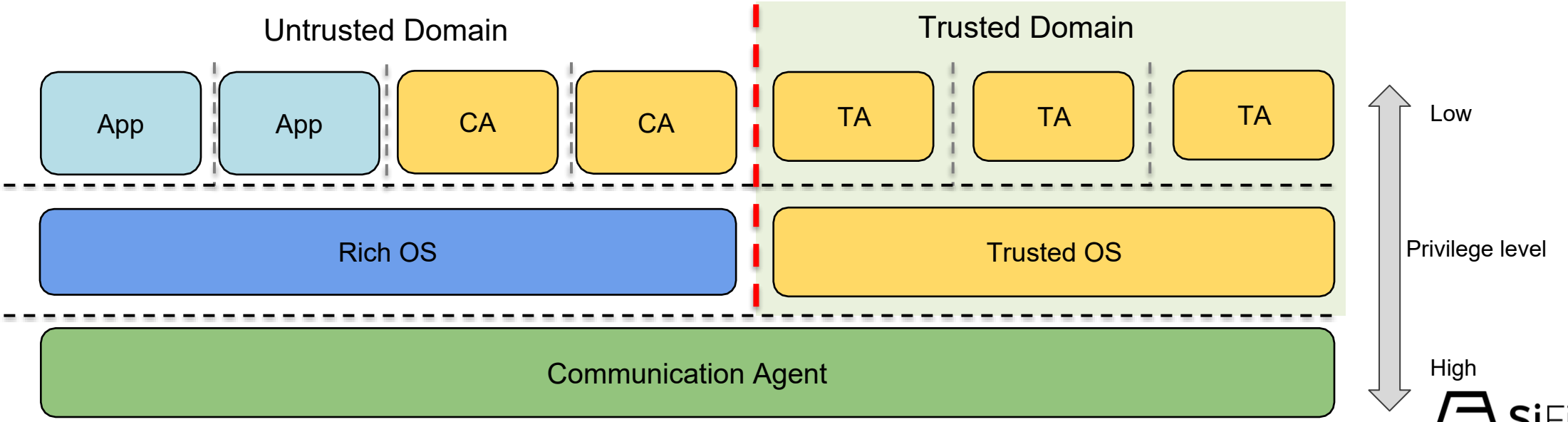
- OpenSBI
- U-Boot
- OP-TEE OS
- Linux

Successfully passes all regression tests (v4.6.0) https://github.com/OP-TEE/optee_test on QEMU Virt and SiFive platforms.

SiFive

Introduction to TEE

- Trusted OS that runs alongside but isolated from an REE, ensures confidentiality and maintains integrity
- Hardware-enforced isolation technology
  - ARM® TrustZone®
  - RISC-V PMP, SiFive WorldGuard
- Communication Agent
  - **Trusted Firmware A**: handles SMC (Secure Monitor Call)
  - **OpenSBI**: MPXY SBI extension handles SBI call



SiFive

RISC-V OP-TEE System Architecture

Interaction flow between Linux, OP-TEE OS, and GlobalPlatform defined Trusted Applications

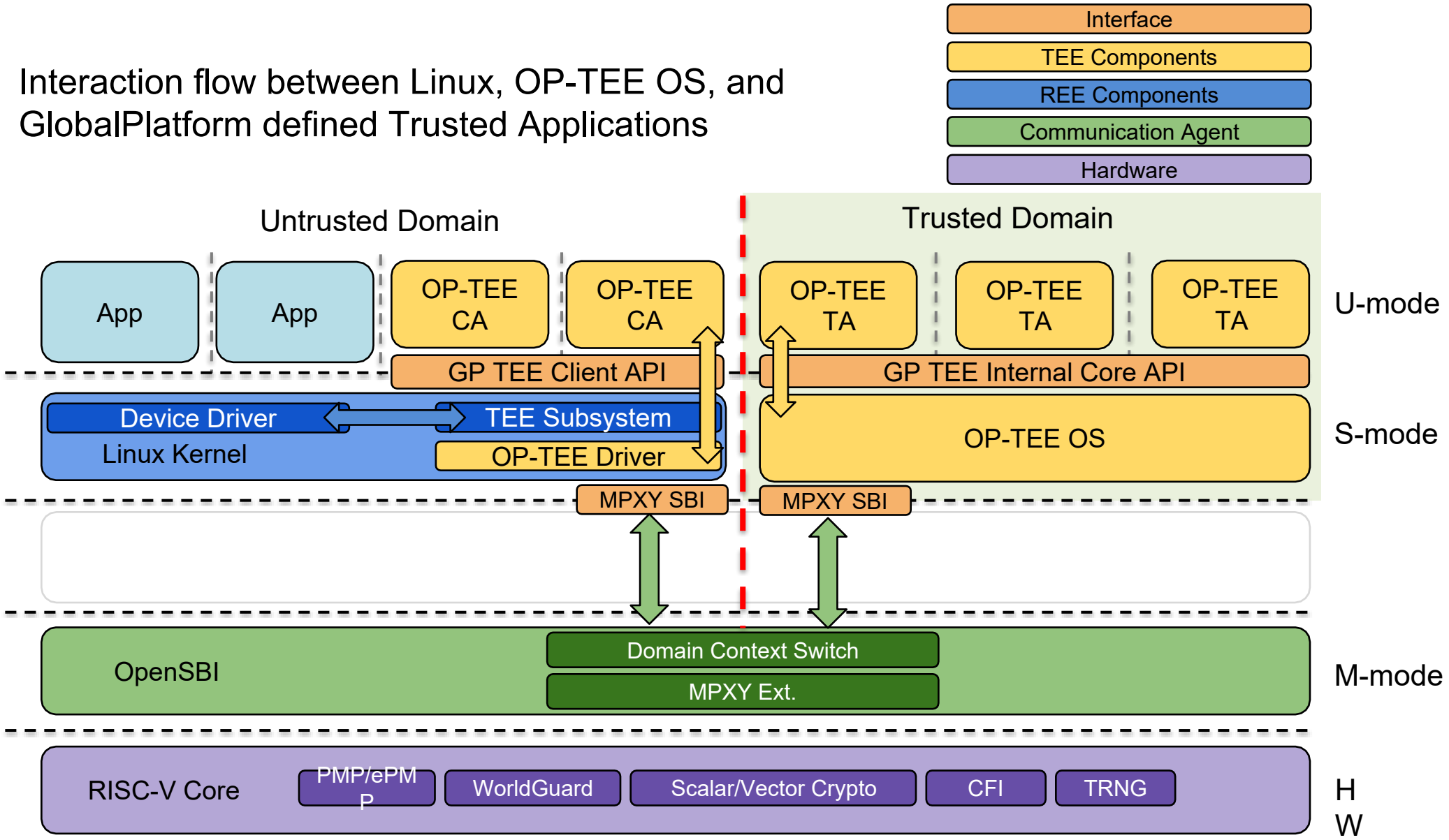


Image credits: https://riscv.org/blog/2024/10/towards-generic-risc-v-tee-ecosystem-with-penglai-and-op-tee/

SiFive

Supervisor Domain Isolation

OpenSBI Domain is a generic framework that supports

- **DT-based physical memory partitioning**
  - Regions are statically allocated
  - Assign CPU associated memory/device regions to domains
- **Domain context switching**
  - Save/restore S-mode CSRs

From the kernel's perspective, trusted memory is **reserved** and cannot be mapped into the kernel's virtual address space.

```
opensbi-domains {
        compatible = "opensbi,domain,config";
        tmem: tmem {
                compatible = "opensbi,domain,memregion";
                base = <0x0 0xF1000000>;
                order = <24>; // 16MiB
        };
        allmem: allmem {
                compatible = "opensbi,domain,memregion";
                base = <0x0 0x0>;
                order = <64>;
        };
        tdomain: trusted-domain {
                compatible = "opensbi,domain,instance";
                regions = <&allmem 0x38>;
                possible-harts = <&cpu0 &cpu1>;
                next-addr = <0x0 0xF1000000>; /* optee_os: CFG_TDDRAM_START */
                next-mode = <0x1>;
        };
        udomain: untrusted-domain {
                compatible = "opensbi,domain,instance";
                regions = <&tmem 0x0>, <&allmem 0x38>;
                possible-harts = <&cpu0 &cpu1>;
                boot-hart = <&cpu0>;
                next-addr = <0x0 0x81200000>; /* u-boot: CONFIG_TEXT_BASE */
                next-mode = <0x1>;
        };
};
```


SiFive

Supervisor Domain Isolation

OpenSBI boot log:

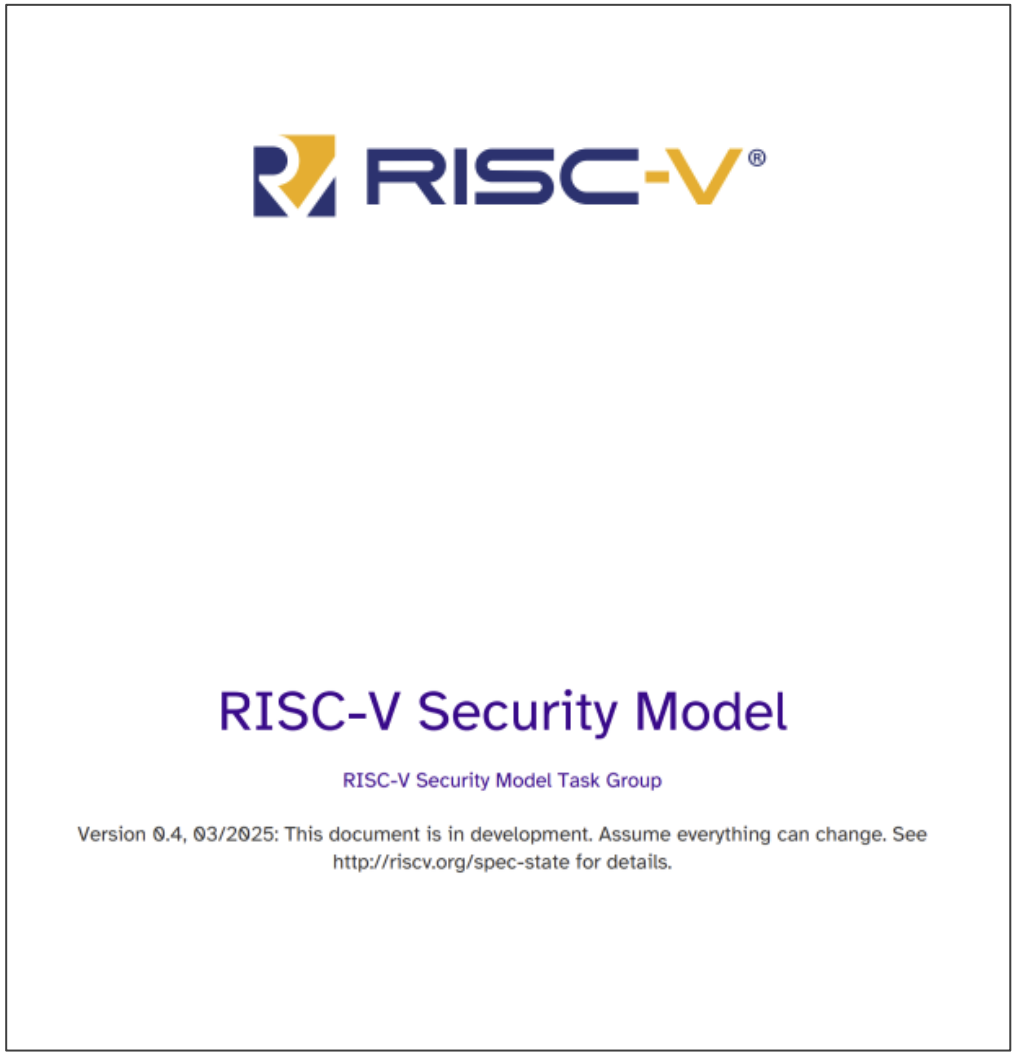
```
  Domain1 Name            : trusted-domain
  Domain1 Boot HART         : 0
  Domain1 HARTs           : 0*,1*
  Domain1 Region00          : 0x0000000002000000-0x000000000200ffff M: (I,R,W) S/U: ()
  Domain1 Region01          : 0x0000000080180000-0x000000008019ffff M: (R,W) S/U: ()
  Domain1 Region02          : 0x0000000080100000-0x000000008017ffff M: (R,X) S/U: ()
  Domain1 Region03          : 0x0000000000000000-0xffffffffffffffff M: () S/U: (R,W,X)
  Domain1 Next Address        : 0x00000000f1000000
  Domain1 Next Arg1         : 0x0000000081299ce0
  Domain1 Next Mode          : S-mode

  Domain2 Name            : untrusted-domain
  Domain2 Boot HART         : 0
  Domain2 HARTs           : 0,1
  Domain2 Region00          : 0x0000000002000000-0x000000000200ffff M: (I,R,W) S/U: ()
  Domain2 Region01          : 0x0000000080180000-0x000000008019ffff M: (R,W) S/U: ()
  Domain2 Region02          : 0x0000000080100000-0x000000008017ffff M: (R,X) S/U: ()
+ Domain2 Region03          : 0x00000000f1000000-0x00000000f1ffffff M: () S/U: ()
  Domain2 Region04          : 0x0000000000000000-0xffffffffffffffff M: () S/U: (R,W,X)
  Domain2 Next Address        : 0x0000000081200000
  Domain2 Next Arg1         : 0x0000000081299ce0
  Domain2 Next Mode          : S-mode
```


SiFive

Security Constraint in RISC-V Architecture

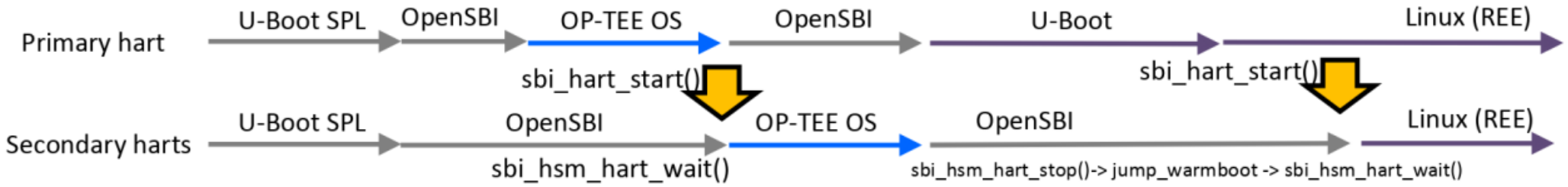
- Allow Root domain to access resources assigned to any domain, while preventing itself from unintended access to resources assigned to a different domain (privilege escalation)
- Prevent other domains from accessing resources assigned to Root domain
- Block resources assigned to TEE domain from access by Normal domain
- Allow resources assigned to Normal domain to be accessible to Normal domain (r/w/x), and to TEE domain (r/w) (default sharing rule)
- Ensure resources assigned to a single TA, or a guest TEE, are not be accessible by a different TA, or guest TEE, without consent.

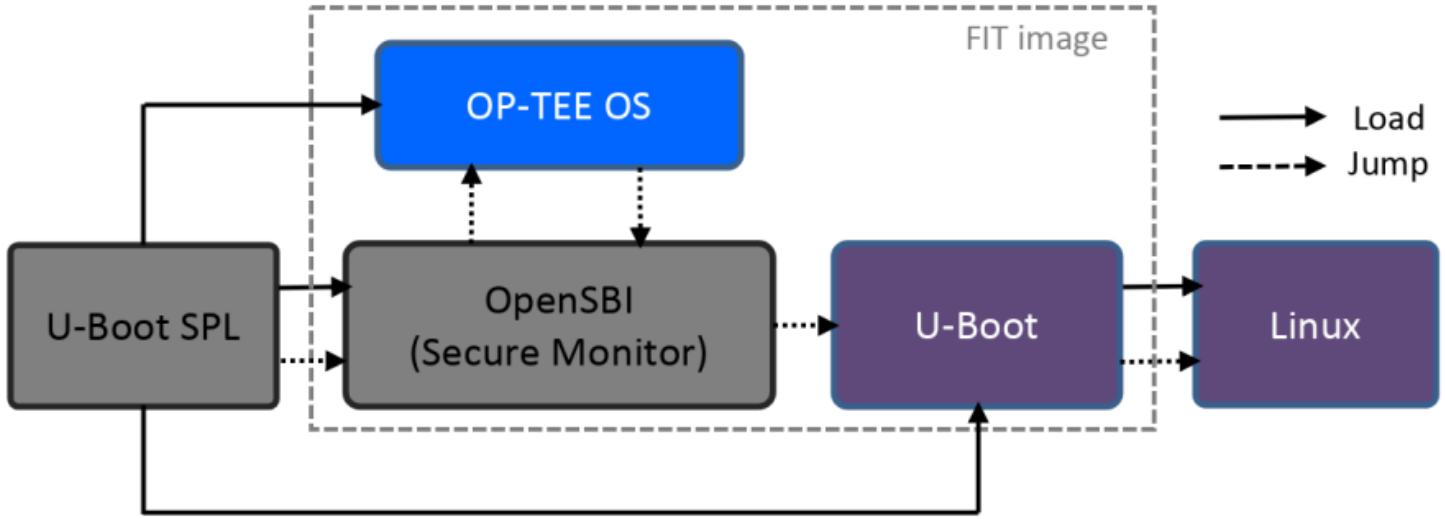


SiFive

Boot Process

- U-Boot FIT image (u-boot.itb) allows including OP-TEE binary (tee.bin) which will be loaded by U-Boot SPL





SiFive

OP-TEE Configurations

- Memory layout configurations
  - **CFG_TDDRAM_START**
  - **CFG_TDDRAM_SIZE**
  - **CFG_TEE_RAM_VA_SIZE**
- **CFG_CORE_NB_CORE**: Number of HARTs runs Linux/OP-TEE
  - e.g. SiFive Unmatched board (1xS71 + 4xU74MC)
  - CFG_CORE_NB_CORE=4
- **CFG_NUM_THREADS**: Number of thread
- **CFG_SEMIHOSTING**: Can be used as secure console on QEMU virt machine
- **CFG_RISCV_SBI_CONSOLE**: SBI console useful during early stage bring-up
- **CFG_CORE_UNSAFE_MODEXP & CFG_TA_MBEDTLS_UNSAFE_MODEXP**: Control which modular exponentiation algorithm is used
  - y: Enables a faster, but potentially unsafe algorithm (not constant-time; vulnerable to timing attacks)
  - n: Uses a slower, but safe algorithm (constant-time; resistant to timing attacks)

Debugging Configurations

- **DEBUG=1**: Compiler optimization -O0
- **CFG_TEE_CORE_DEBUG**: Activates assertions and lock checks and detects errors as early as possible
- **CFG_TEE_{CORE|TA}_LOG_LEVEL**: Set to 4 to enable the highest verbosity
- **CFG_CORE_DEBUG_CHECK_STACKS**: Detects stack overflow
- **CFG_WITH_STACK_CANARIES**: Compiler instrumentation with -finstrument-functions
- **CFG_{CORE|TA}_STACK_PROTECTOR**: Compiler stack protection mechanisms with -fstack-protector to prevet buffer overflow attack
- **CFG_{CORE|TA}_ASLR**: Enabled by default, if Zkr is supported
  - **CFG_CORE_ASLR_SEED**: provides consistent VA offset for debugging

RISC-V OP-TEE Upstream Status

Work Item	Status
Boot SMP (with non-contiguous HART id)	Supported
Domain Context Switch (OpenSBI)	Supported
Zkr Seed	Supported
Core compiler generated stack canaries	Supported
TA compiler generated stack canaries	Supported
Core ASLR	Supported
TA ASLR	Supported
AIA	Basic driver w/o domain specific interrupt handling

RISC-V OP-TEE Roadmap

Work Item	Status
MPXY (with RPMI MM+ReqFwd)	In Progress
Smepmp	In Progress
SiFive WorldGuard	In Progress
External interrupt directed Domain Context switching	Planned
RISC-V Crypto	Planned
TA Floating Point	Planned
ASAN	Planned
RISC-V Pointer Masking	Planned
RISC-V CFI Landing Pads	Planned
RISC-V CFI Shadow Stack	Planned
OP-TEE Documentation for RISC-V architecture	Planned

Acknowledgements

NXP: Initial RISC-V Support by Marouene Boubakri

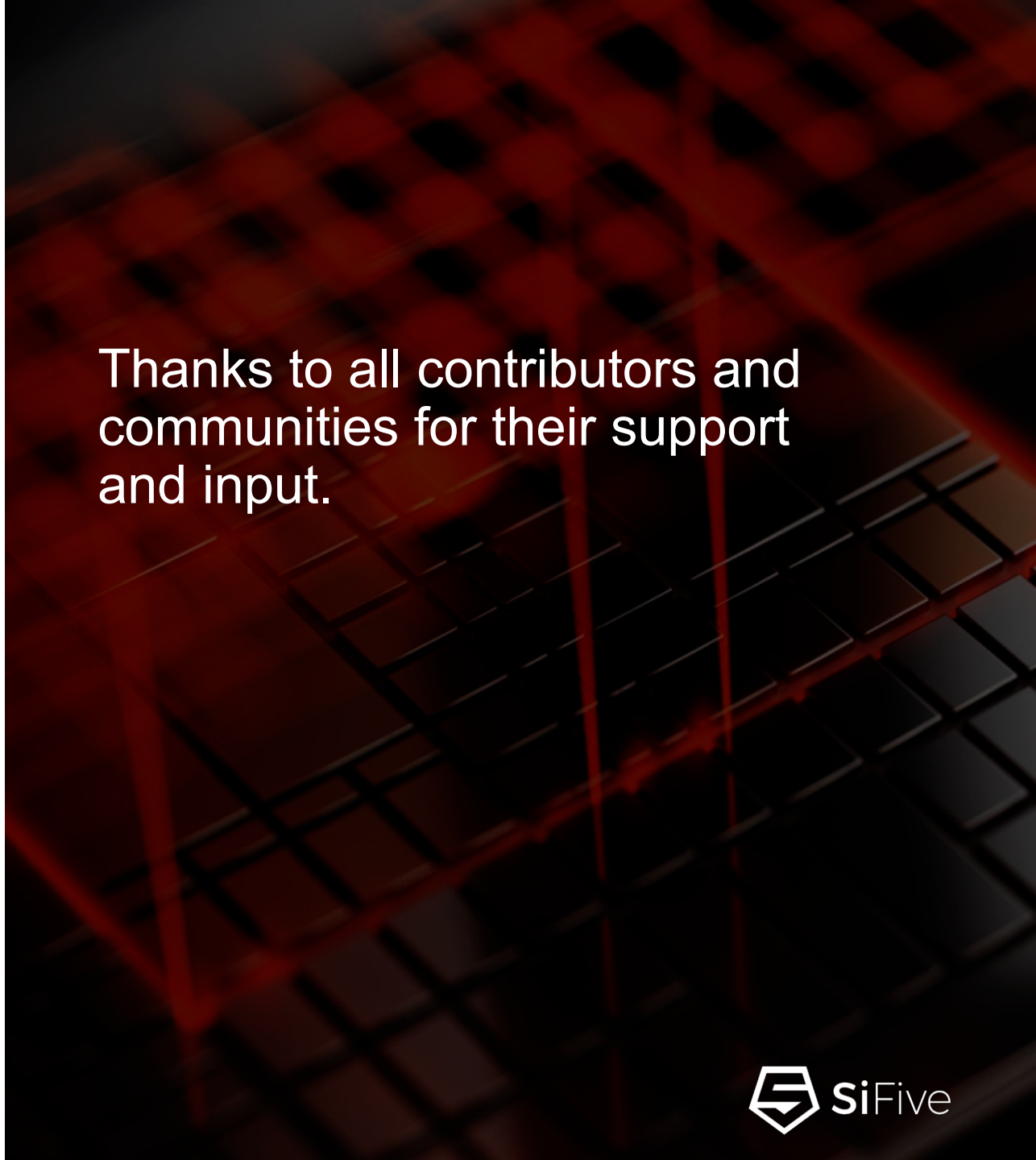
Andes Technology: Code review and maintenance by Alvin Chang

PengLai Enclave Team & Intel: Domain Context Switch by Qingyu Shang & Yong Li

Nuclei System Technology & ESWIN Computing: Improvements

BOSC: AIA (APLIC/IMSIC) driver support by Huang Borong

ARM/Linaro Community: Code review & feedbacks



Reference

- Lightning Talk: A Trusted OS for RISC-V ? OP-TEE is a Candidate - Marouene Boubakri, NXP: https://www.youtube.com/watch?v=uR1YwN47yY8
- Marouene Boubakri, Fausto Chiatante, Behassen Zouari, “Open Portable Trusted Execution Environment framework for RISC-V,” 2021 IEEE 19th Int. Symp. on Parallel and Distributed Processing with Applications (ISPA), 2021
- RISC-V Security Model: https://github.com/riscv-non-isa/riscv-security-model/releases/download/v0.4/riscv-platform-security-model.pdf
- Towards Generic RISC-V TEE Ecosystem with Penglai and OP-TEE: https://riscv.org/blog/2024/10/towards-generic-risc-v-tee-ecosystem-with-penglai-and-op-tee

SiFive

Thank you!

Yu-Chien Peter Lin
<peter.lin@sifive.com>

SiFive