Virtualizing Arm TrustZone on KVM

Chun-Yen Lin, Shih-Wei Li

國立臺灣大學
National Taiwan University

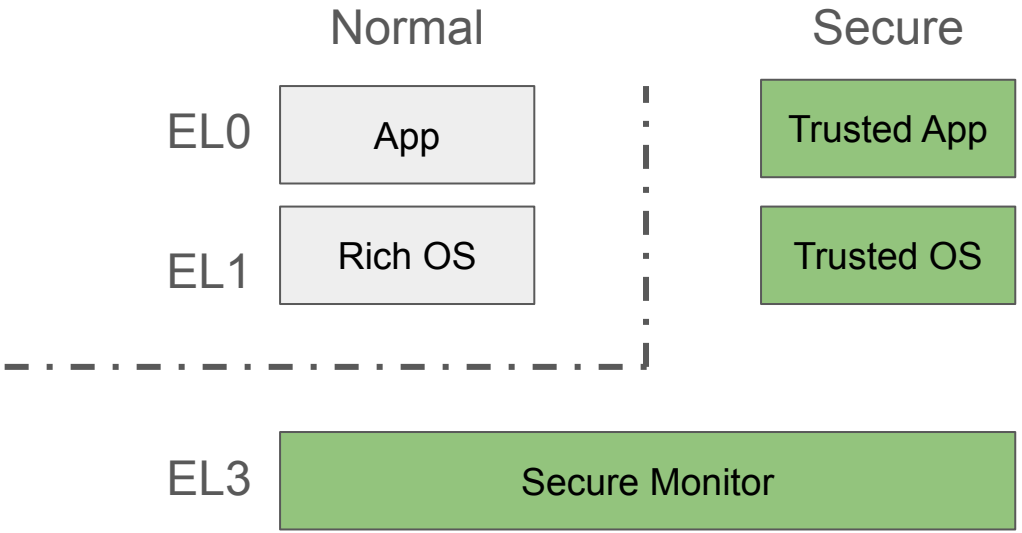
SSLab
Secure Systems Laboratory

Outline

- Introduction to Arm TrustZone and OP-TEE
- Motivation & Our Goals
- Design
- Evaluation
- Future Plan

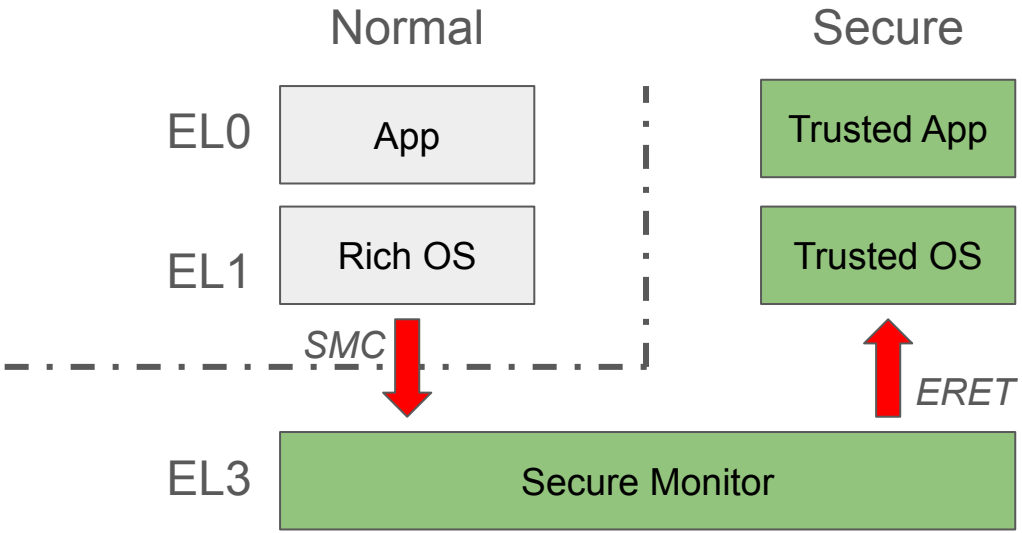
Arm TrustZone

- Isolation between worlds



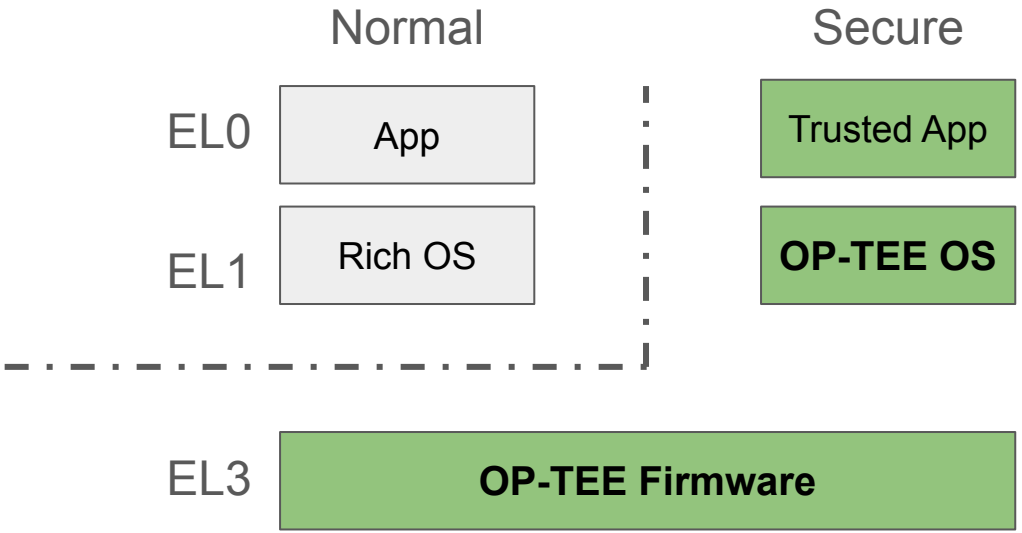
Arm TrustZone

- CPU can switch to secure worlds by making a SMC (secure monitor call)



Arm TrustZone

- TrustZone is typically paired with OP-TEE as its secure OS



OP-TEE

- Trusted Execution Environment (TEE) based on Arm TrustZone
- Developers can implement Trusted Applications (TAs) in the secure world
- OP-TEE firmware handles the `SMC` at EL3

Motivation

- Currently QEMU/TCG support the emulation of TrustZone, but KVM doesn't
  - We can virtualize TrustZone CPU features at KVM and reuse the existing QEMU emulation on TrustZone hardware to create a virtual TrustZone environment

Our Goals

- Extend KVM to expose a virtual TrustZone to VMs
- Set up an OP-TEE VM on our virtualized TrustZone environment
  - To demonstrate the ability of our virtualization framework

Design - Sensitive Instructions

- `ERET` and `MSR/MRS` with EL3 system registers
  - May cause undefined behavior
- `SMC`
  - Should be executed by EL1 then be trapped to EL3
- `HLT` (Semihosting call in Arm64)
  - May cause the CPU to stop from executing

Design - Sensitive Instructions Handling

- Trap-and-emulate to handle sensitive instructions
  - Not all sensitive instructions would be trapped
  - We adapted a para-virtualization method to replace sensitive instructions with `HVC`
    - `ERET` and `HLT`

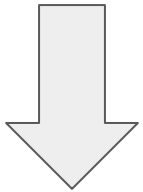
Design - Virtual System Registers

- Most of the EL3 system registers are also banked at EL1

EL3 Register	EL1 Equivalent Register
SPSR_EL3	SPSR_EL1
VBAR_EL3	VBAR_EL1
SCTLR_EL3	SCTLR_EL1
ELR_EL3	ELR_EL1
SP_EL3	SP_EL1

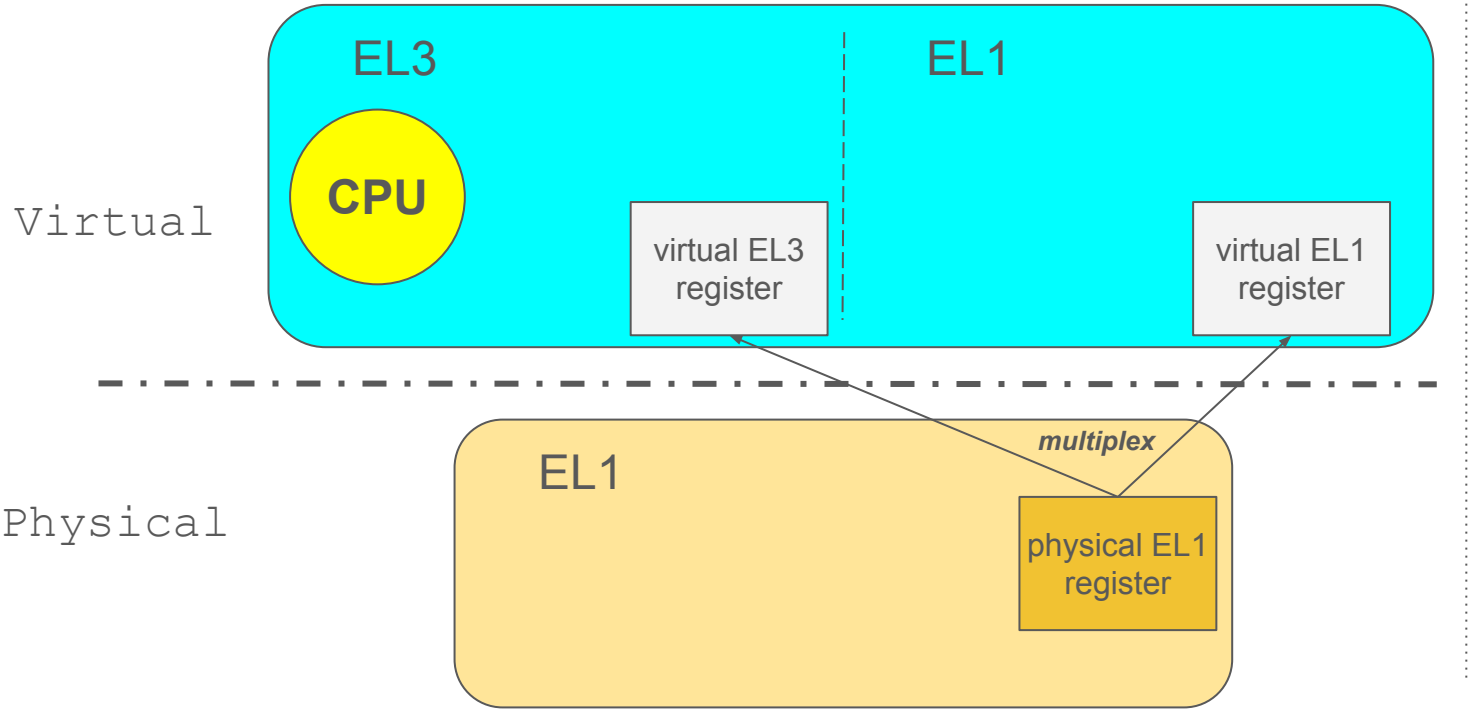
Design - Virtual System Registers

```
MSR SPSR_EL3, x0
```

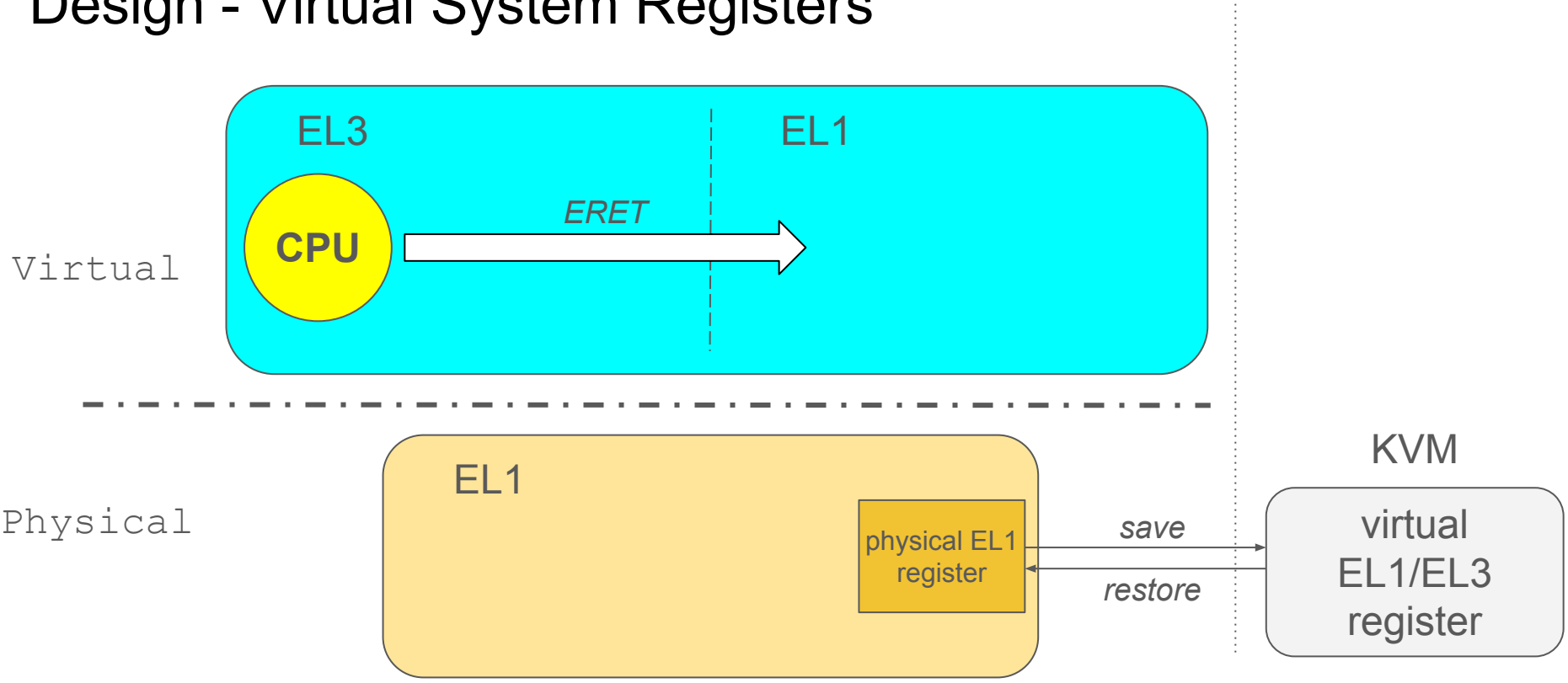


```
MSR SPSR_EL1, x0
```

Design - Virtual System Registers



Design - Virtual System Registers



Design - Virtual System Registers

- Some register are only existing in EL3
  - SCR_EL3, MDCR_EL3
  - We store these registers' value at KVM, guest VM can access them through `HVC`

SMC

1. Save current PC to virtual ELR_EL3
2. Save current process state to virtual SPSR_EL3
3. Set PC to virtual VBAR_EL3 with corresponding offset
4. Context switch, save current EL1 registers and restore virtual EL3 register on hardware EL1

ERET

1. Restore PC from virtual ELR_EL3
2. Restore process state from virtual SPSR_EL3
3. Context switch, save current EL3 registers and restore virtual EL1 register to hardware EL1

QEMU

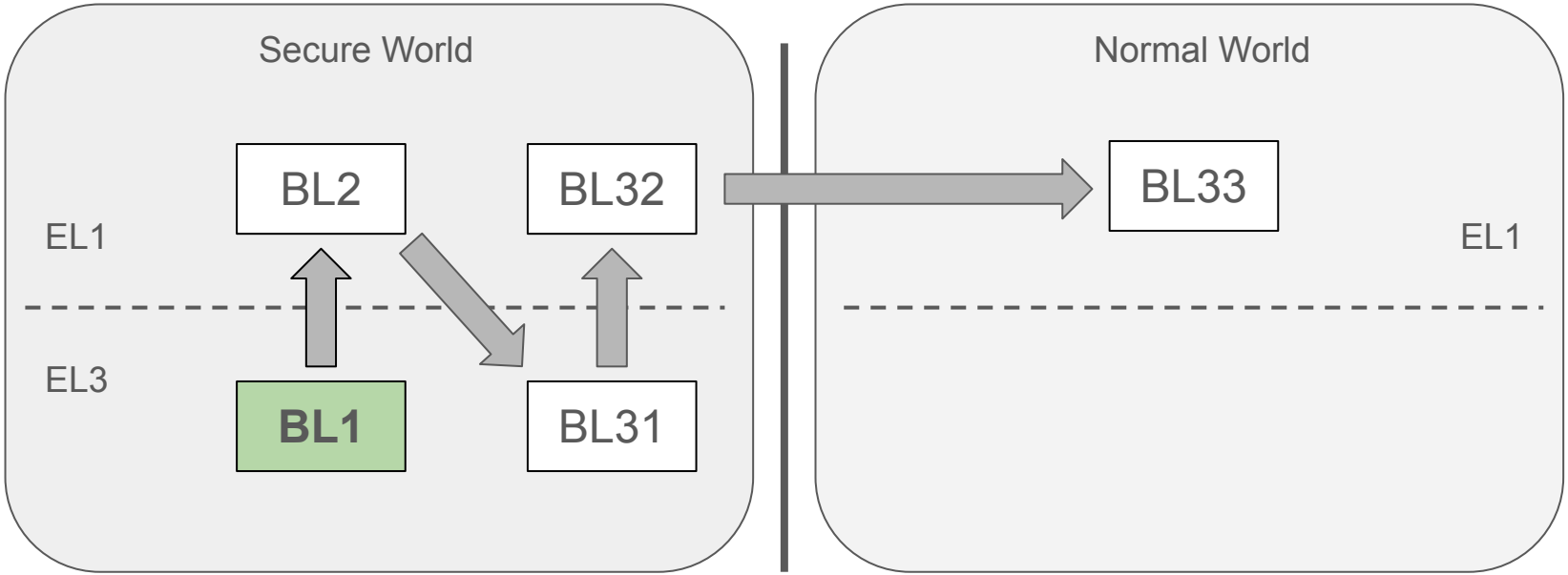
- Add a virtual secure memory region to emulate secure memory of TrustZone
  - Map secure UART and GPIO onto it
- Handle semihosting call
  - Replace `HLT` with `HVC`
  - Return to QEMU from KVM
  - Handle semihosting call at QEMU

Paravirtualization of OP-TEE

- Most modification is about the **ARM Trusted Firmware**
  - The firmware of OP-TEE
  - Handles the early boot stages, sets up critical security features for OP-TEE, loads the OP-TEE OS into the Secure World
  - Handles the transition between the Secure world and the Normal World
  - Consists of several bootloaders

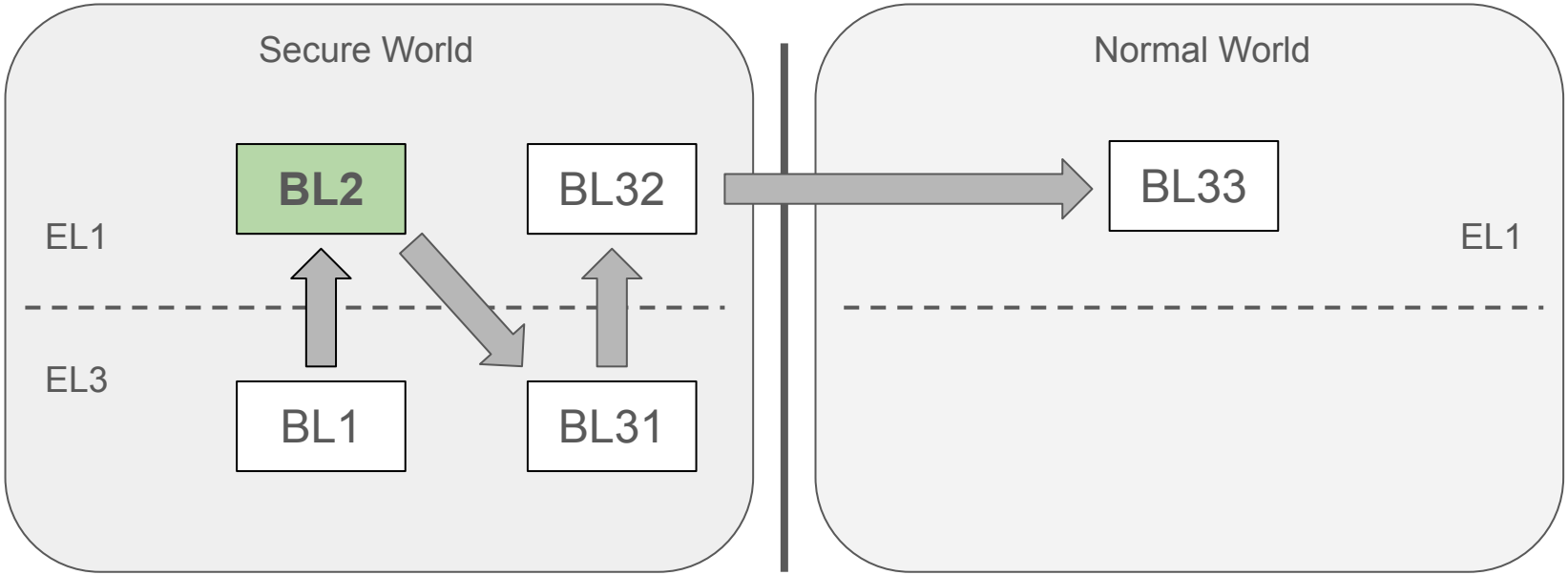
Bootloader 1

Initialize the early hardware components and security features



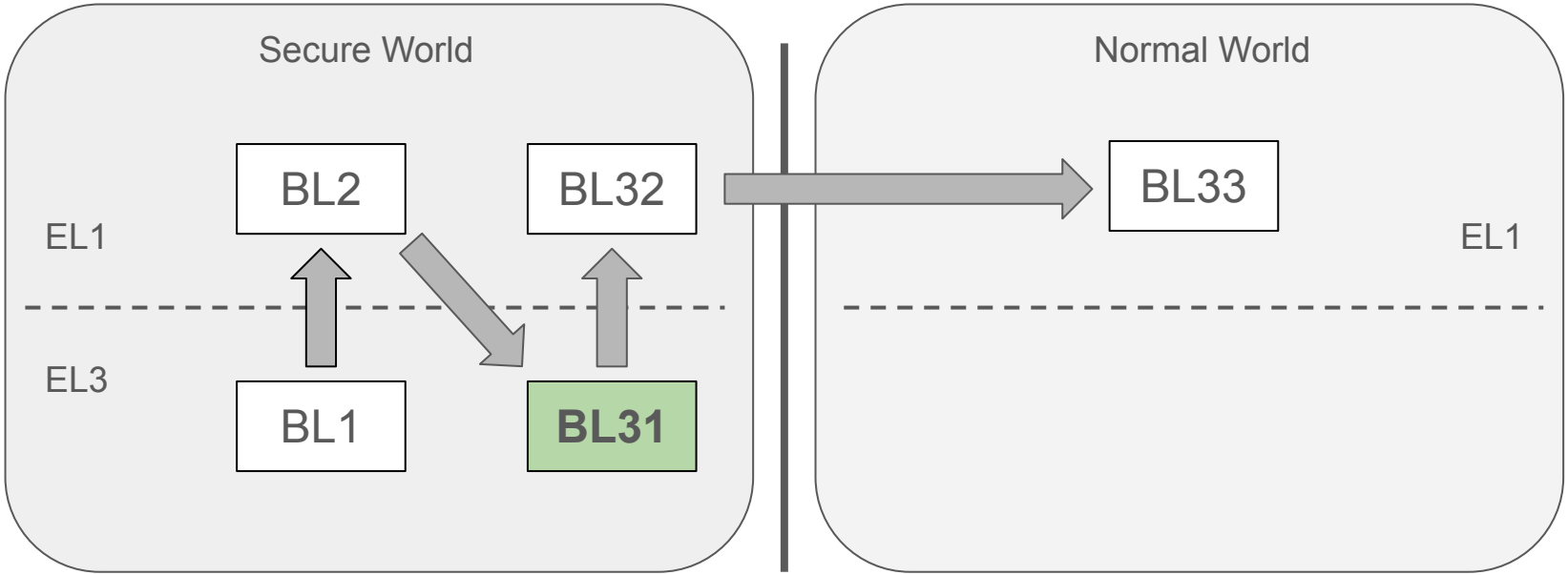
Bootloader 2

Load and verify further boot loaders and system software



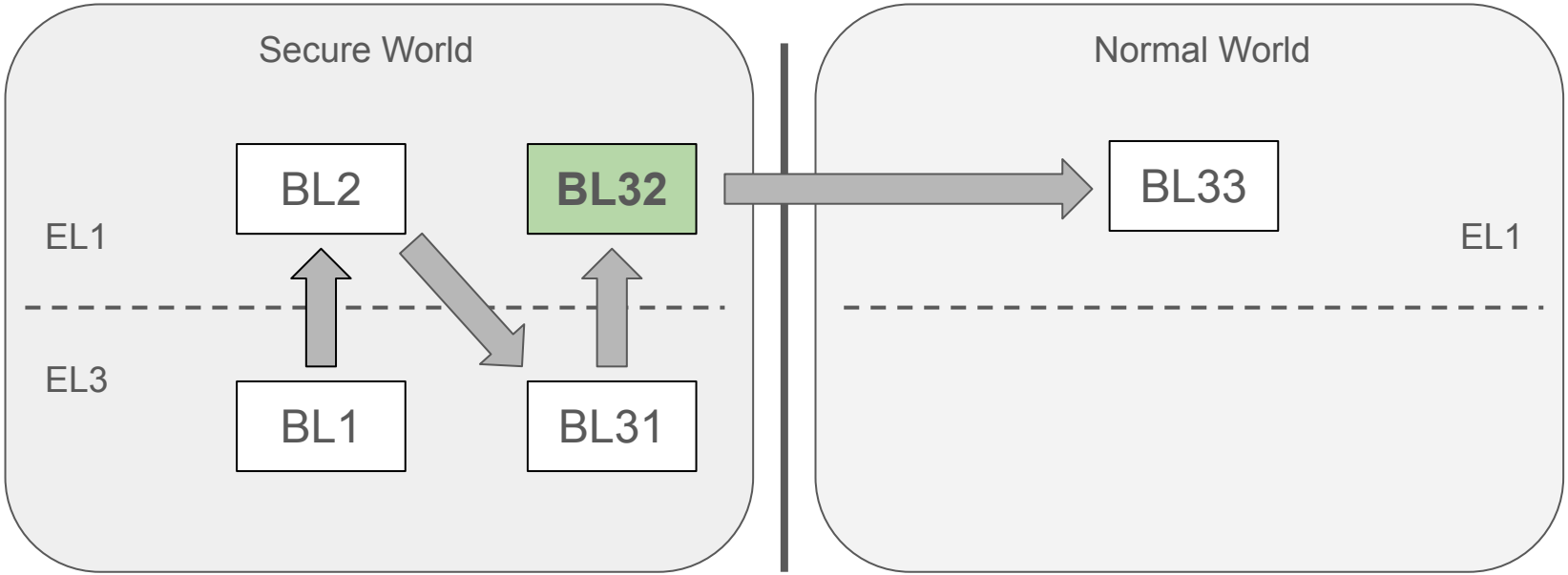
Bootloader 31

EL3 Runtime Firmware, the core part of TF-A that handles SMC and do secure world and normal world context switching



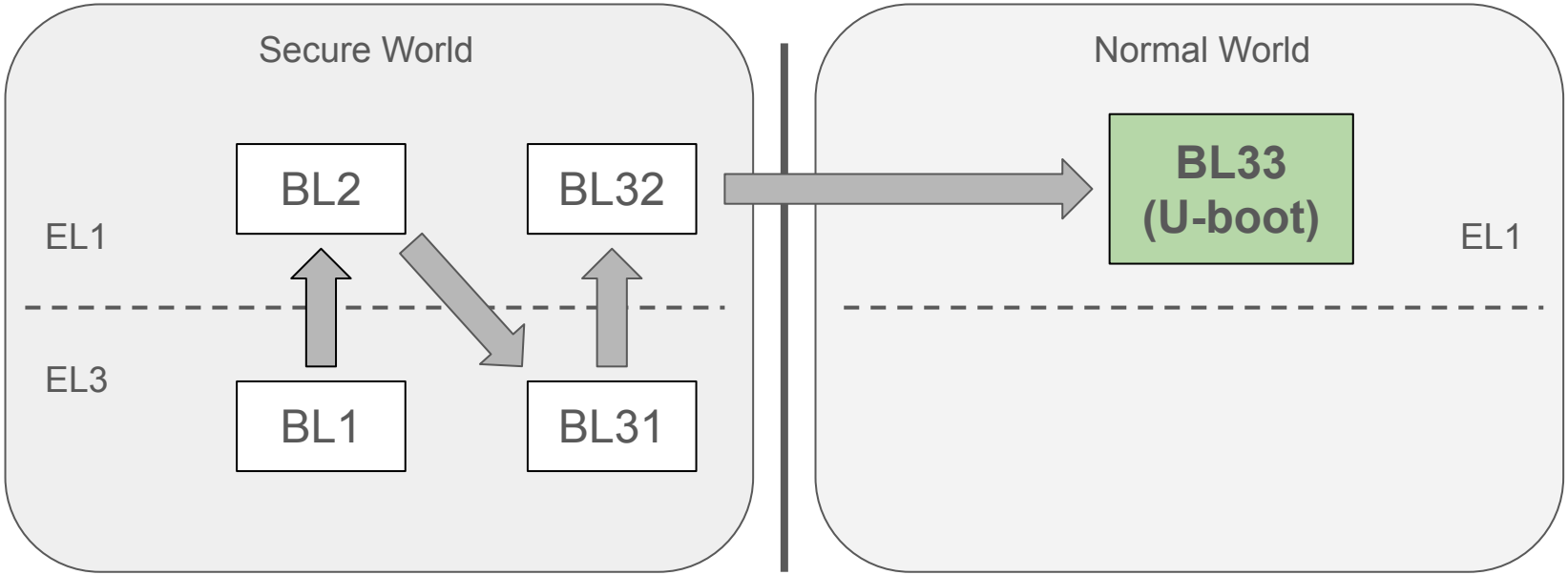
Bootloader 32

Load and initialize the OP-TEE OS



Bootloader 33 (U-boot)

Load and initialize the normal world OS (Linux)



Performance Evaluation - Setup

- Hardware
  - AVA Developer Platform
    - Arm Neoverse N1
    - 32 cores and 32GB RAM
- Software
  - KVM Linux 5.15
  - QEMU v8.0.0
  - Guest VM
    - OP-TEE 4.0.0
    - Trusted Firmware-a 2.9
    - Linux 6.2

Performance Evaluation - applications

Application	Description
acipher	Generates an RSA key pair and encrypts a supplied string
aes	Runs an AES encryption and decryption
hello_world	A simple Trusted Application to answer a hello command
hotp	Generates a HMAC based One Time Password
random	Generates a random UUID
secure_storage	Reads/writes raw data into the OP-TEE secure storage
plugins	Interacts with Linux syslog service as a plugin

Performance Evaluation - Results

Exec. Time (ms)	TCG	KVM
acipher	365	30
aes	266	32
hello_world	221	23
hotp	279	30
random	211	22
secure_storage	561	51
plugins	10327	10093

Future Plan

- Extend our approach to KVM-based confidential VM (e.g. pKVM or Arm CCA)
- Extend QEMU to virtualize secure IO devices
  - TrustZone Address Space Controller (TZASC) can define memory regions and prevent unauthorized access

Thanks!
Questions?