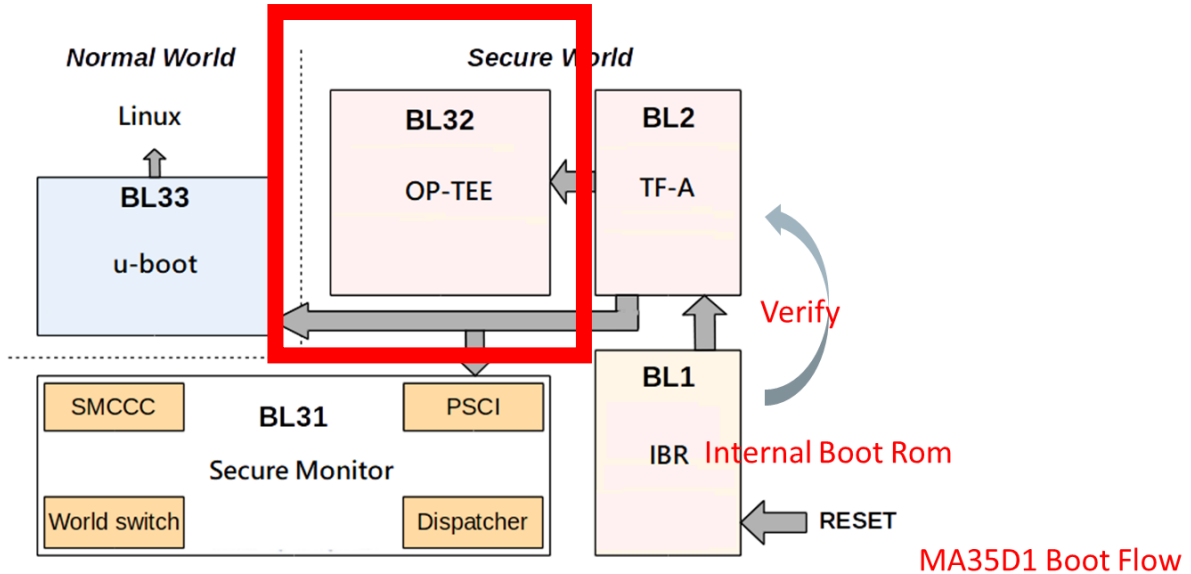
OP-TEE

Introduction

OP-TEE(BL32)

- Initial during TF-A boot sequence
- Provide isolation from the non-secure OS(Linux)
- Provide Trusted Applications



MA35D1 Boot Flow

Source: https://www.cnblogs.com/arnoldlu/p/14175126.html

TEE (Trusted Execution Environment)

- OP-TEE (Arm Linaro)
- Trusty TEE (Google Android)
- QSEE (Qualcomm)
- TEEgris (Samsung)
- Beanpod ISEE (MTK)
- iOS Secure Enclave (Apple) – Separate processor

TRUSTED EXECUTION ENVIRONMENTS (TEE)
An Introduction to TEE functionality and how GlobalPlatform supports it.

THE TECHNOLOGY

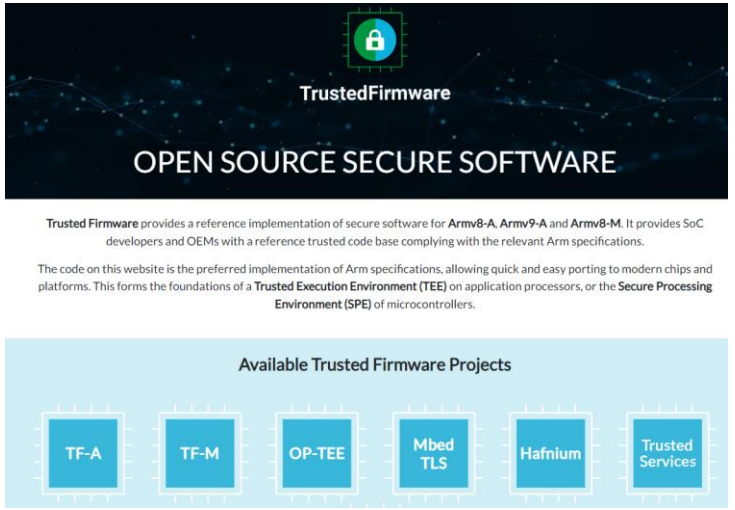
The TEE is a secure area of the main processor of a connected device that ensures sensitive data is stored, processed and protected in an isolated and trusted environment. As such, it offers protection against software attacks generated in the Rich Operating System (Rich OS).

The TEE's ability to offer safe execution of authorized security software, known as 'trusted applications' (TAs), enables it to provide end-to-end security by protecting the execution of authenticated code, confidentiality, authenticity, privacy, system integrity and data access rights. Comparative to other security environments on the device, the TEE also offers high processing speeds and a large amount of accessible memory. The primary purpose of the isolated execution environment, provided by the TEE, is to protect device and TA assets.

Source: https://globalplatform.org/wp-content/uploads/2018/05/Introduction-to-Trusted-Execution-Environment-15May2018.pdf

OP-TEE – Overview

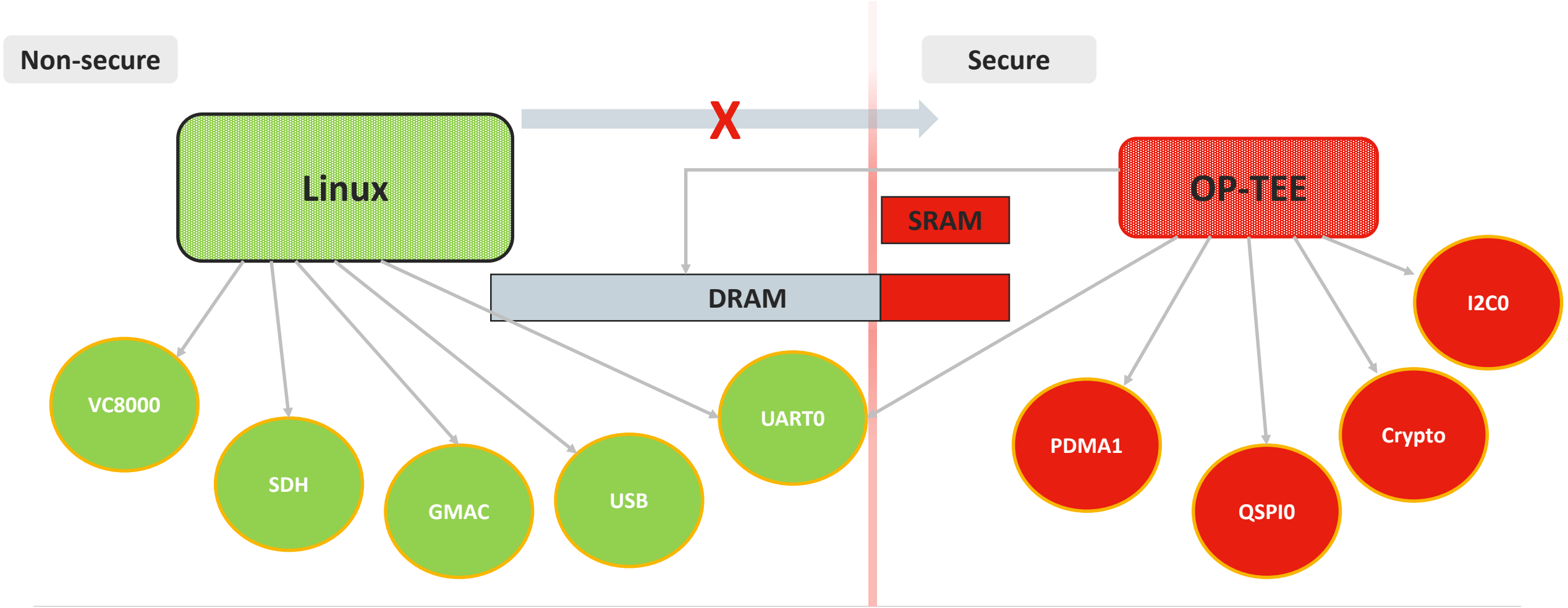
- Open-source Portable Trusted Execution Environment
- An OS running in secured world
- Designed as companion to Linux kernel which is referred to as the Rich Execution Environment (REE)
- Rely on the Arm TrustZone technology (ARMv7/ARMv8)
- Why we need OP-TEE?



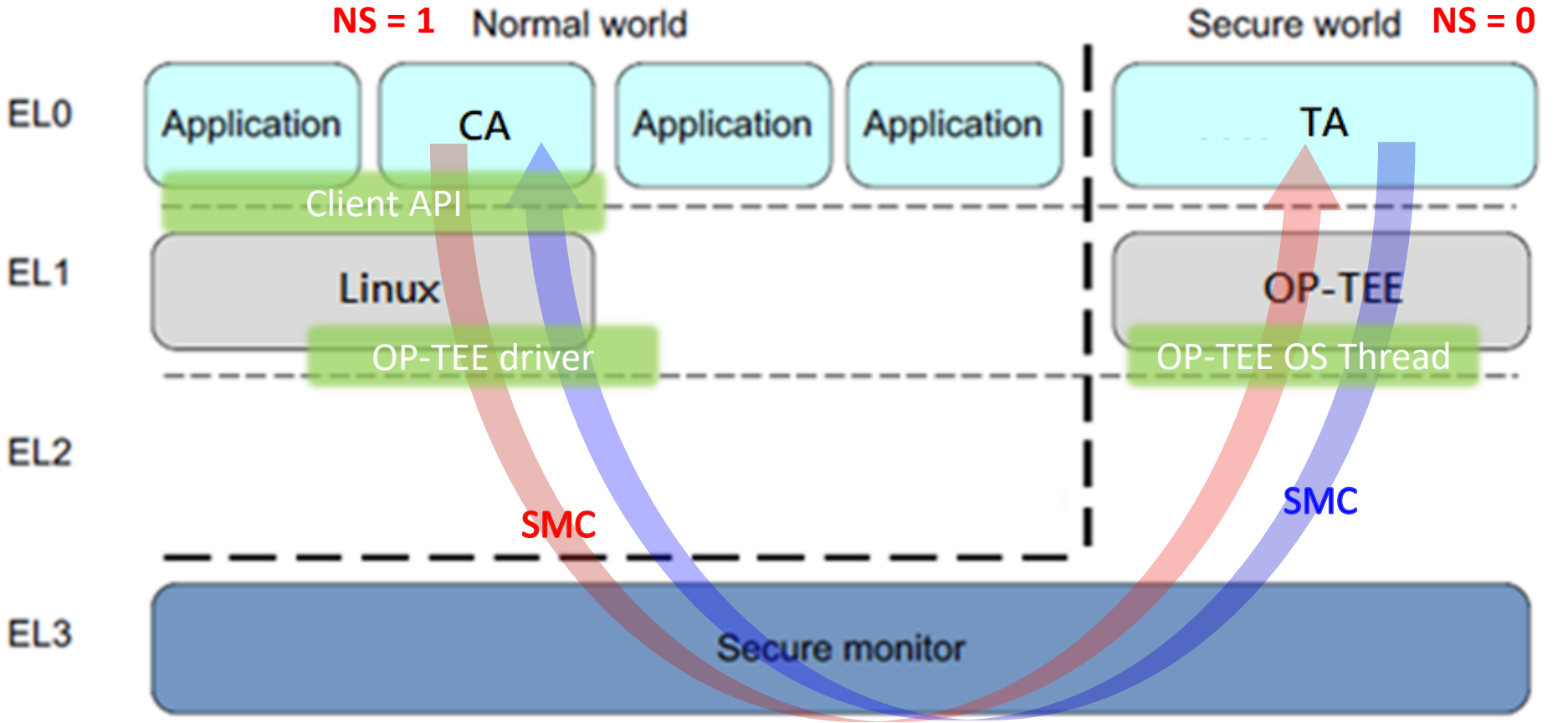
TEE Application

- Fingerprint recognition
- DRM video playback
- Mobile payments
- Trusted UI

MA35D1 Secure/Non-secure IP



Client Application ⟷ Trusted Application



① Push non-secure world context
② Pop secure world context

① Push secure world context
② Pop non-secure world context

CA & TA

- Optee_client/libteec

```
TEEC_Result TEEC_InitializeContext(
    const char* name,
    TEEC_Context* context)

void TEEC_FinalizeContext(
    TEEC_Context* context)

TEEC_Result TEEC_OpenSession (
    TEEC_Context* context,
    TEEC_Session* session,
    const TEEC_UUID* destination,
    uint32_t connectionMethod,
    const void* connectionData,
    TEEC_Operation* operation,
    uint32_t* returnOrigin)

void TEEC_CloseSession (
    TEEC_Session* session)

TEEC_Result TEEC_InvokeCommand(
    TEEC_Session* session,
    uint32_t commandID,
    TEEC_Operation* operation,
    uint32_t* returnOrigin)
```

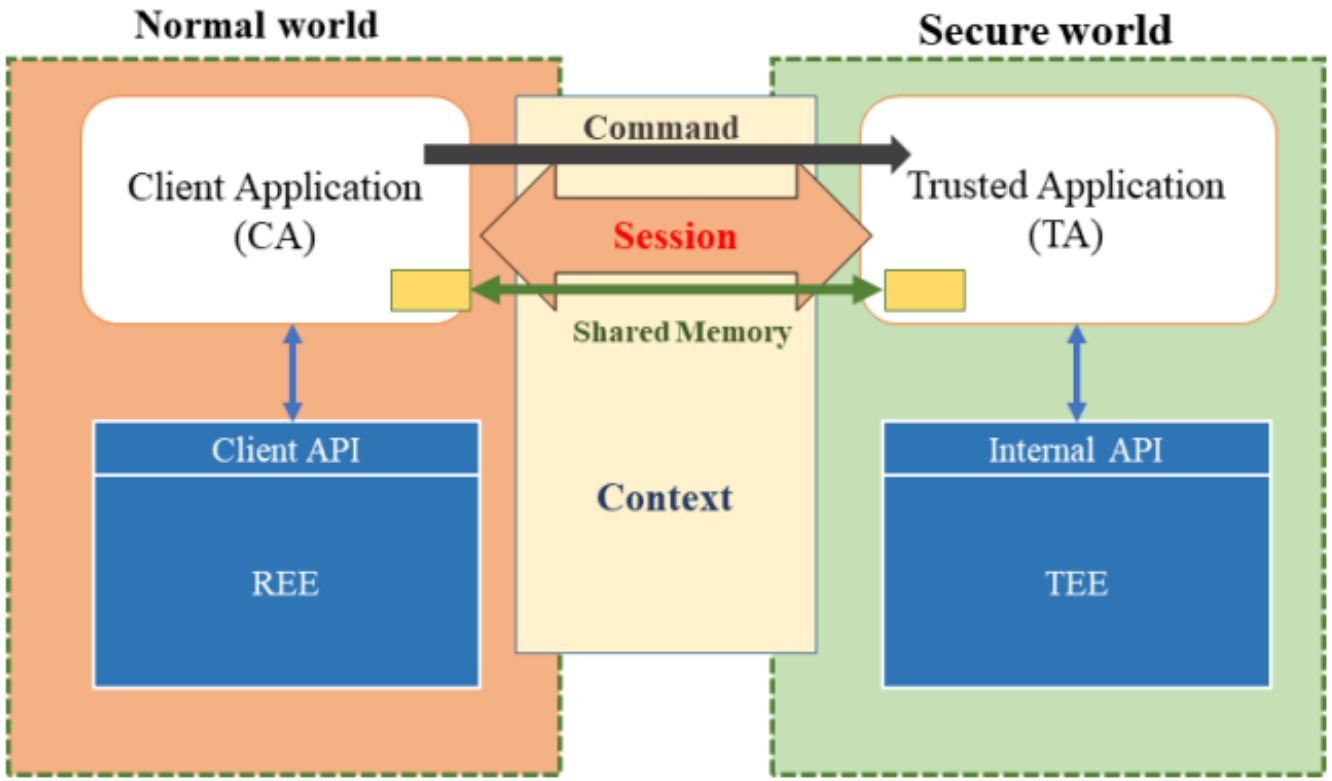
1

5

2

4

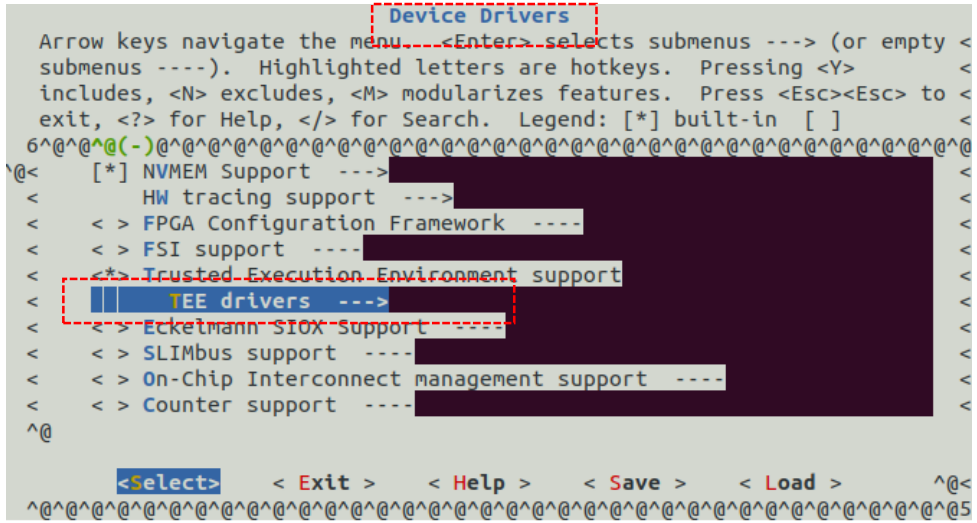
3



Build OP-TEE with Yocto

- Check the TEE is enabled in Linux Kernel

```
$ bitbake linux-ma35d1 -c menuconfig
$ bitbake linux-ma35d1 -C compile
```



- Add OP-TEE stuff into the Yocto build
  1. Open the local.conf *(~/yocto/build/conf/local.conf)*
  2. Insert `MACHINE_FEATURES_append = "optee"`
  3. Save

OP-TEE in Linux

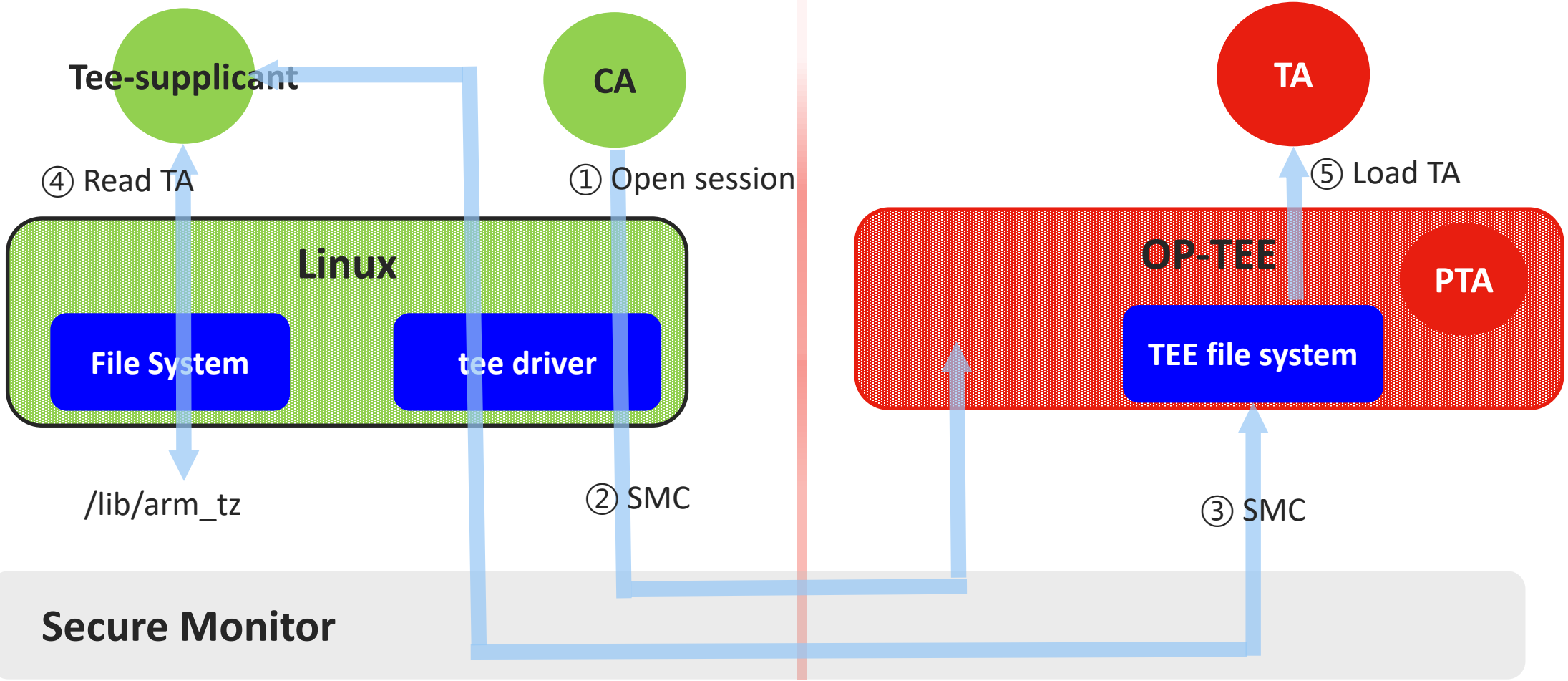
- Check optee device

```
root@ma35d1-evb:/dev# ls tee*
tee0       teepriv0
```

- TEE subsystem running in background

```
416 root      2052 S    /sbin/agetty -o -p -- \u --noclear t
417 root      3252 S    -sh
419 root      2332 S    /usr/bin/tee-supplicant
420 root      6724 S    /lib/systemd/systemd-logind
425 root         0 IW   [kworker/u4:3-ev]
427 root         0 IW   [kworker/1:0-ggn]
```

Run-time Load TA



CA & TA (from MA35D1 Linux)

- CA (Client Application)

```
root@ma35d1-evb:~# ls /usr/bin/optee*
/usr/bin/optee_example_acipher      /usr/bin/optee_example_hello_world    /usr/bin/optee_example_random
/usr/bin/optee_example_aes          /usr/bin/optee_example_hotp           /usr/bin/optee_example_secure_storage
```

- TA (Trusted Application)

```
root@ma35d1-evb:~# ls /lib/optee_armtz/
484d4143-2d53-4841-3120-4a6f636b6542.ta  8aaaf200-2450-11e4-abe2-0002a5d5c51b.ta  d17f73a0-36ef-11e1-984a-0002a5d5c51b.ta
528938ce-fc59-11e8-8eb2-f2801f1b9fd1.ta  a4c04d50-f180-11e8-8eb2-f2801f1b9fd1.ta  e13010e0-2ae1-11e5-896a-0002a5d5c51b.ta
5b9e0e40-2636-11e1-ad9e-0002a5d5c51b.ta  a734eed9-d6a1-4244-aa50-7c99719e7b7b.ta  e626662e-c0e2-485c-b8c8-09fbce6edf3d.ta
5ce0c432-0ab0-40e5-a056-782ca0e6aba2.ta  b3091a65-9751-4784-abf7-0298a7cc35ba.ta  e6a33ed4-562b-463a-bb7e-ff5e15a493c8.ta
5dbac793-f574-4871-8ad3-04331ec17f24.ta  b689f2a7-8adf-477a-9f99-32e90c0ad0a2.ta  f157cda0-550c-11e5-a6fa-0002a5d5c51b.ta
614789f2-39c0-4ebf-b235-92b32ac107ed.ta  b6c53aba-9669-4668-a7f2-205629d00f86.ta  f4e750bb-1437-4fbf-8785-8d3580c34994.ta
731e279e-aafb-4575-a771-38caa6f0cca6.ta  c3f6e2c0-3548-11e1-b86c-0800200c9a66.ta  ffd2bded-ab7d-4988-95ee-e4962fff7154.ta
873bcd08-c2c3-11e6-a937-d0bf9c45c61c.ta  cb3e5ba0-adf1-11e0-998b-0002a5d5c51b.ta
```

UUID defined source header of CA and TA

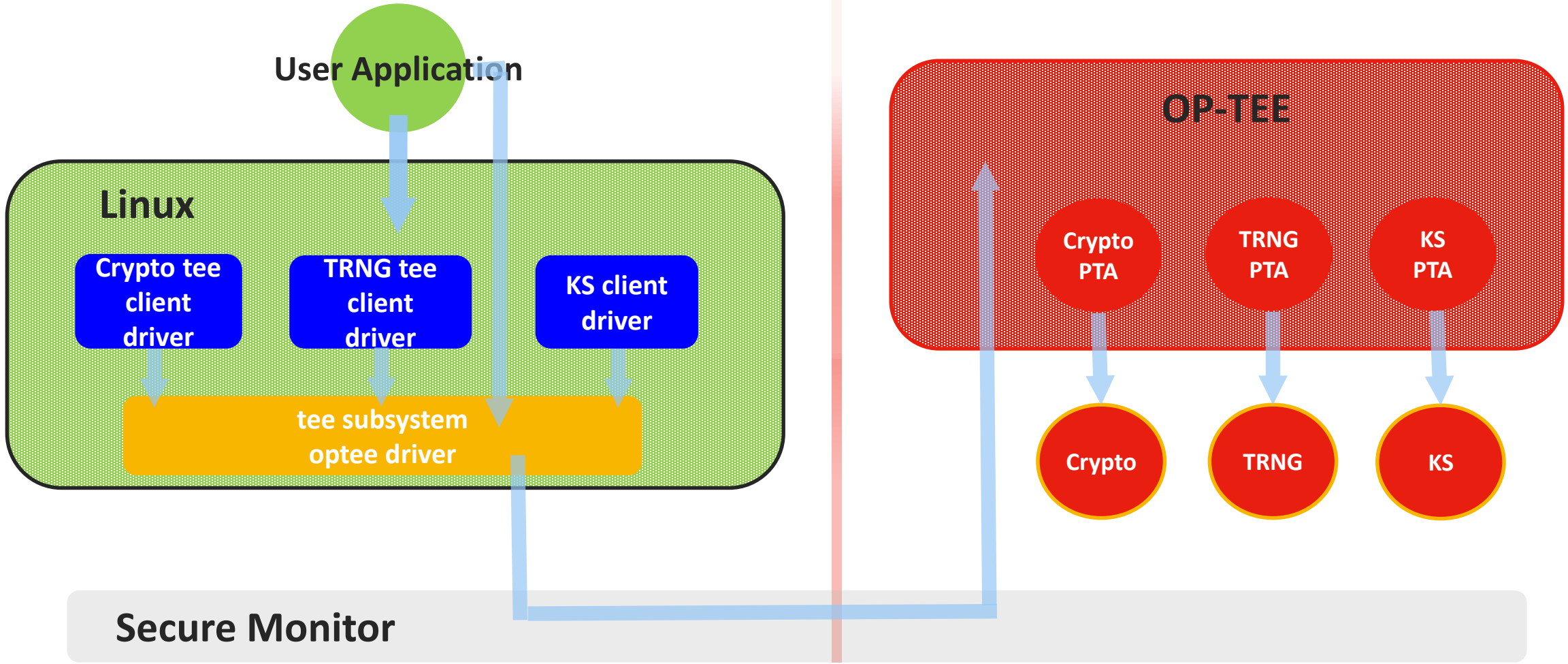
```
/* UUID of the trusted application */
#define TA_SECURE_STORAGE_UUID \
                { 0xf4e750bb, 0x1437, 0x4fbf, \
                        { 0x87, 0x85, 0x8d, 0x35, 0x80, 0xc3, 0x49, 0x94 } }
/*
```

TA sign and encrypt

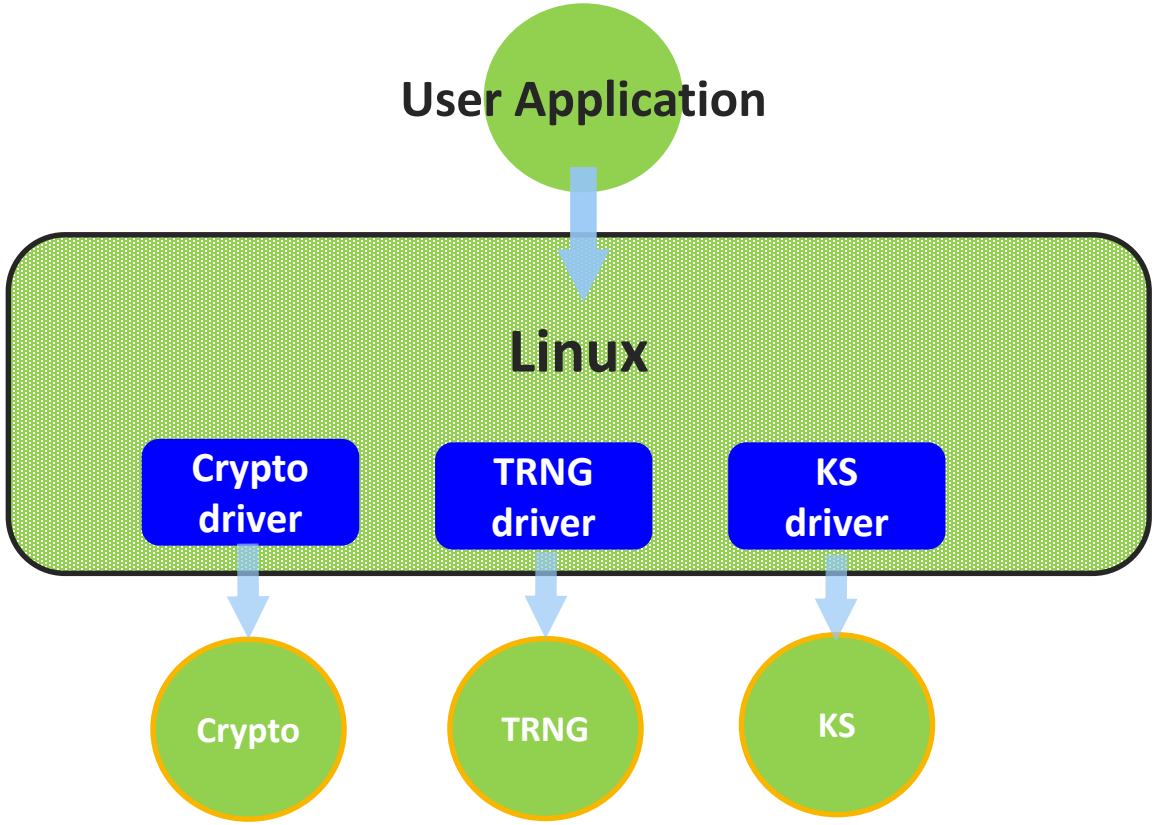
- Sign and encrypt on compile-time
  - RSA private key: optee-os/keys/default_ta.pem
  - Encrypt key TA_ENC_KEY: optee-os/ta/arch/arm/link.mk
  - Sign and encrypt by: optee-os/scripts/sign_encrypt.py

- Verify and decrypt on run-time
  - RSA public key in optee-os image
  - OP-TEE OS decrypt key: tee_otp_get_ta_enc_key()

Secure Crypto, TRNG, KS



Non-secure Crypto, TRNG, KS



OP-TEE

Secure Monitor

Crypto

Crypto Accelerator

- PRNG
  - Can take seed from TRNG
- HMAC/SHA, SHA3, MD5, SM3
- AES, SM4
- ECC, SM2
- RSA

side-channel attack protection ability

Joy of innovation

NUVOTON

Thank You

Danke

Merci

ありがとう

Gracias

Kiitos

감사합니다

धन्यबाद

شكرا لك

הדות