IOT BZH

Introduction to OP-TEE

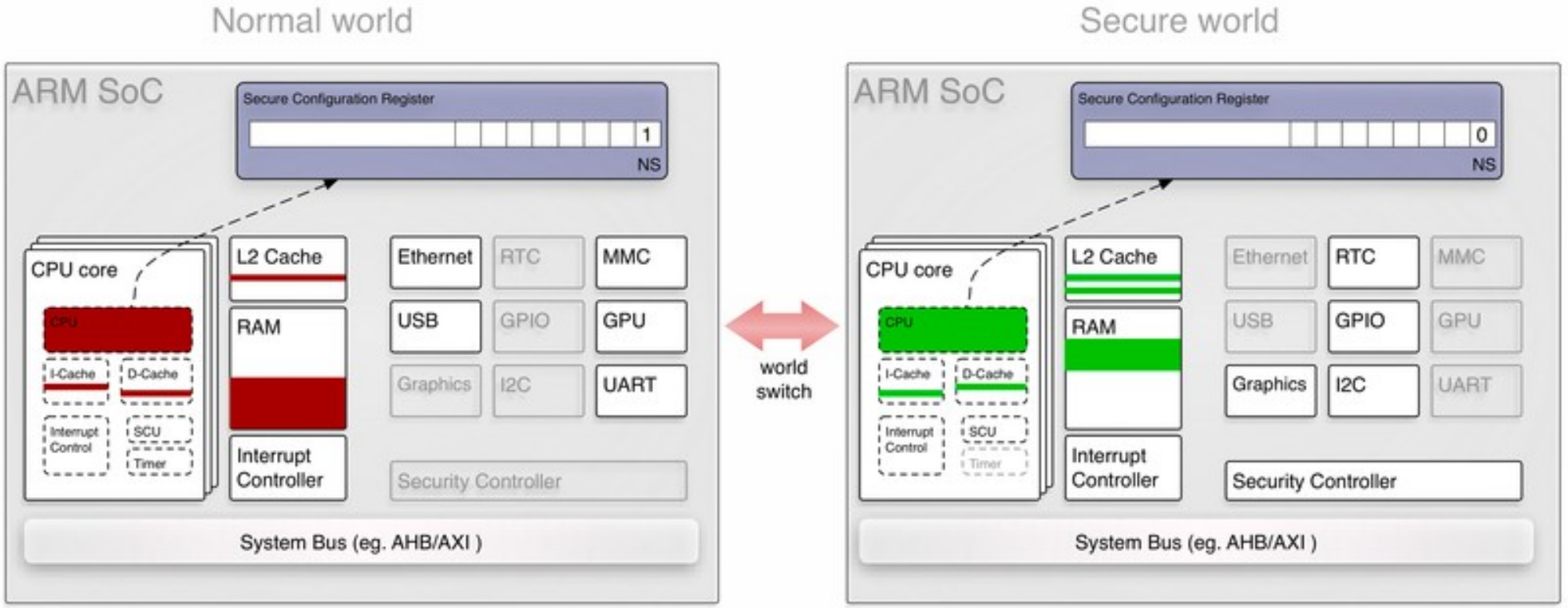
25 May 2016

OP-TEE

- **OP-TEE**
  - Open-source Portable Trusted Execution Environment,
  - Implements the Global Platform API on top of ARM TrustZone,
  - Initiated by ST in 2007, then handled by Linaro (sources on GitHub).

- **Architecture**
  - Security by isolated execution, introducing two contexts:
    - Rich Execution Environment (normal world),
    - Trusted Execution Environment (secure world).
  - A software part handles context switches :
    - “secure monitor” (armv7) or “ARM trusted firmware” (armv8).
  - Both worlds communicates through:
    - Sequences of messages (ioctl()),
    - Shared memory.

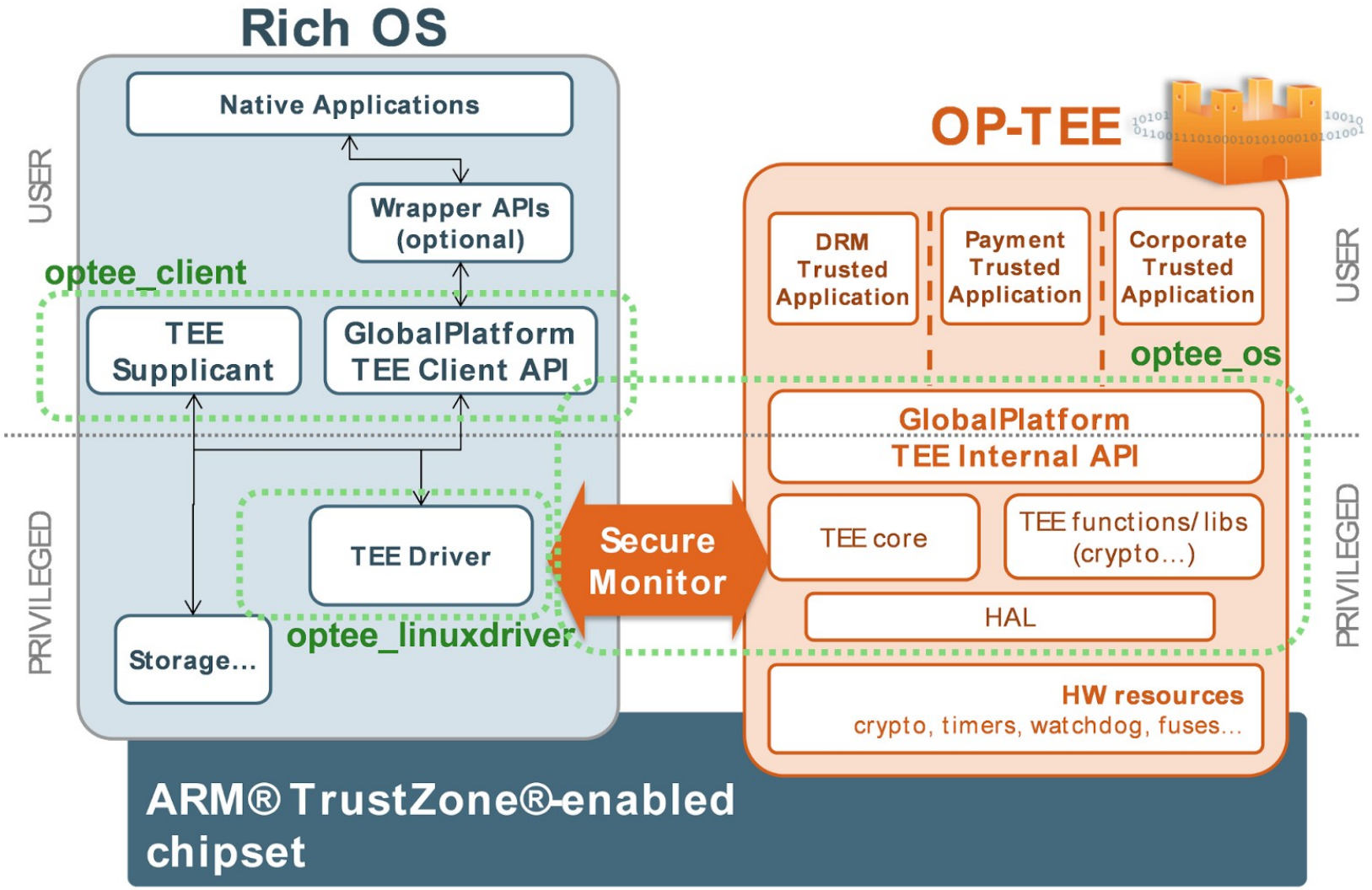
ARM Trustzone hardware isolation



Credit:
http://genode.org/documentation/articles/trustzone

- **Principle**
  - The SoC memory mapping / peripherals visibility can be configured for both worlds,
  - CPU contexts, Core Exceptions Levels,
  - Depends on SoC design, sometimes BootRom use this internally.

OP-TEE Software architecture



OP-TEE

- **OP-TEE OS Characteristics**
  - Sequential execution of commands from Client App, no re-entrance,
  - Checks inputs (commands/datas) received from REE,
  - Strong isolation of TA, stack protections, tasks creation on each TA entries points,
  - Use Secure-RAM HW capability,

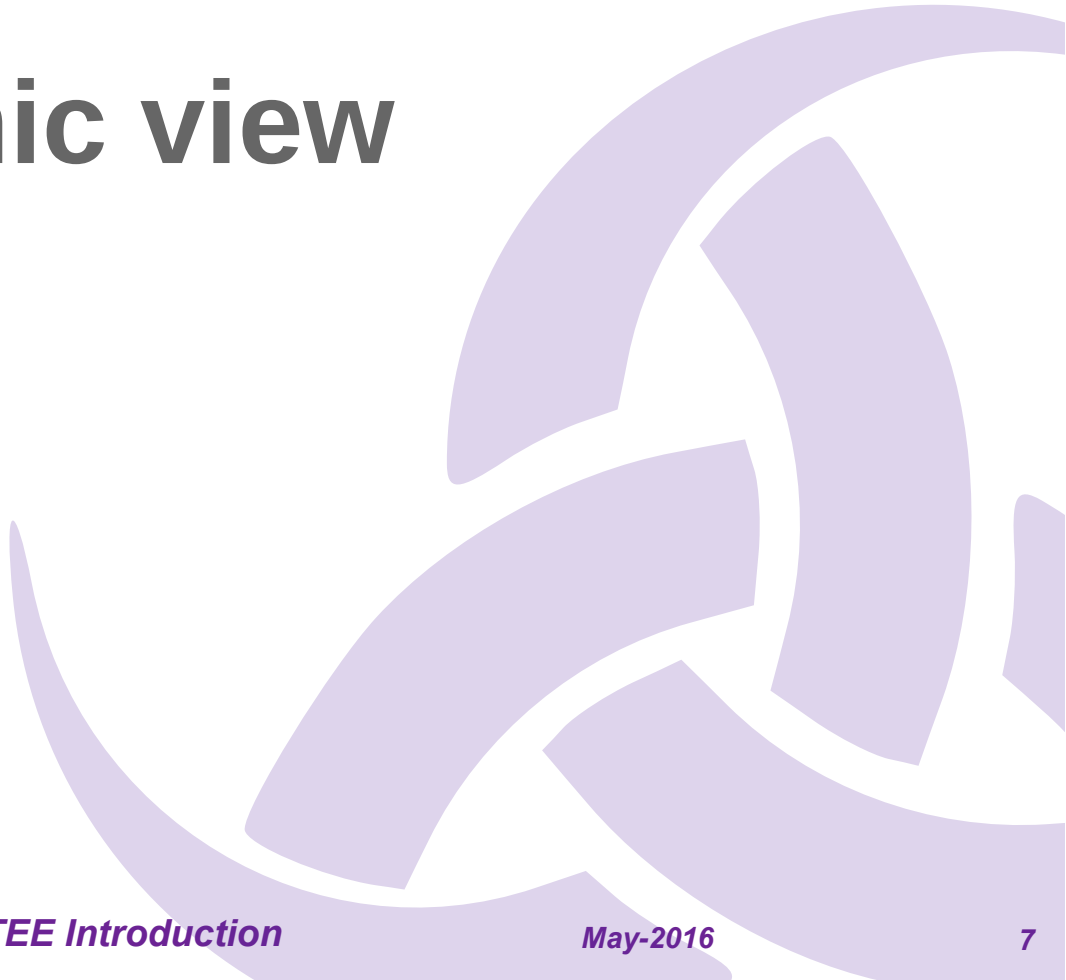
- **Secured Applications**
  - Two binaries blobs:
    - User space program (Normal world),
    - TA: Trusted Application (Secure world).
  - TA are signed, and identified by a UUID,
  - TA integrity is checked by the trusted OS before execution.

- **OS Design**
  - Client library (libteec.so),
  - Kernel Driver (optee.ko),
  - Trusted OS (bare metal C code)

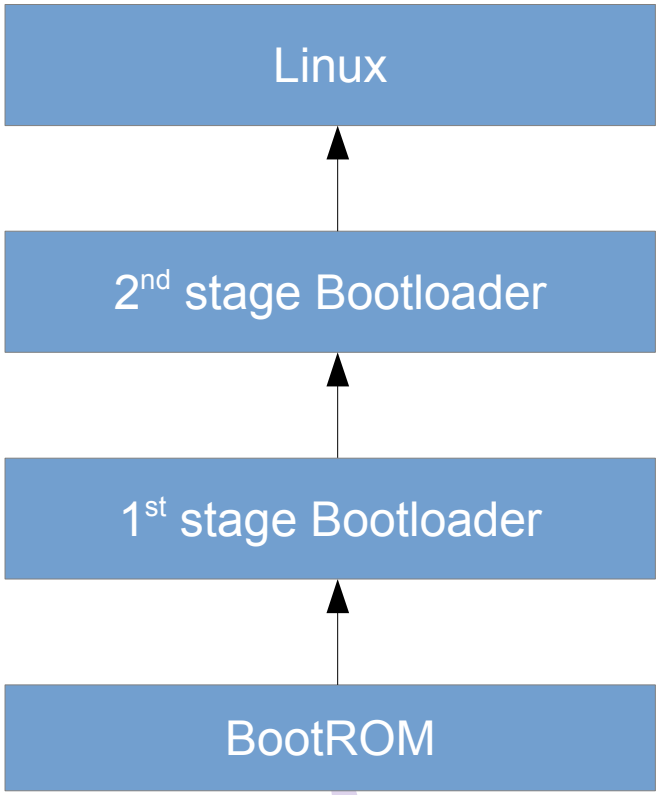
Global Platform

- **Features**
  - Protected storage,
  - SW isolation,
  - Device integrity.
- **TEE Core API specify**
  - Trusted Core Framework API,
  - Trusted Storage API for Data and Keys,
  - Cryptographic Operation API,
  - Time API,
  - Arithmetical API.
- **TEE Client API**
- **Others**
  - Access Control, UI API.
- Specifications are accessible: https://www.globalplatform.org/

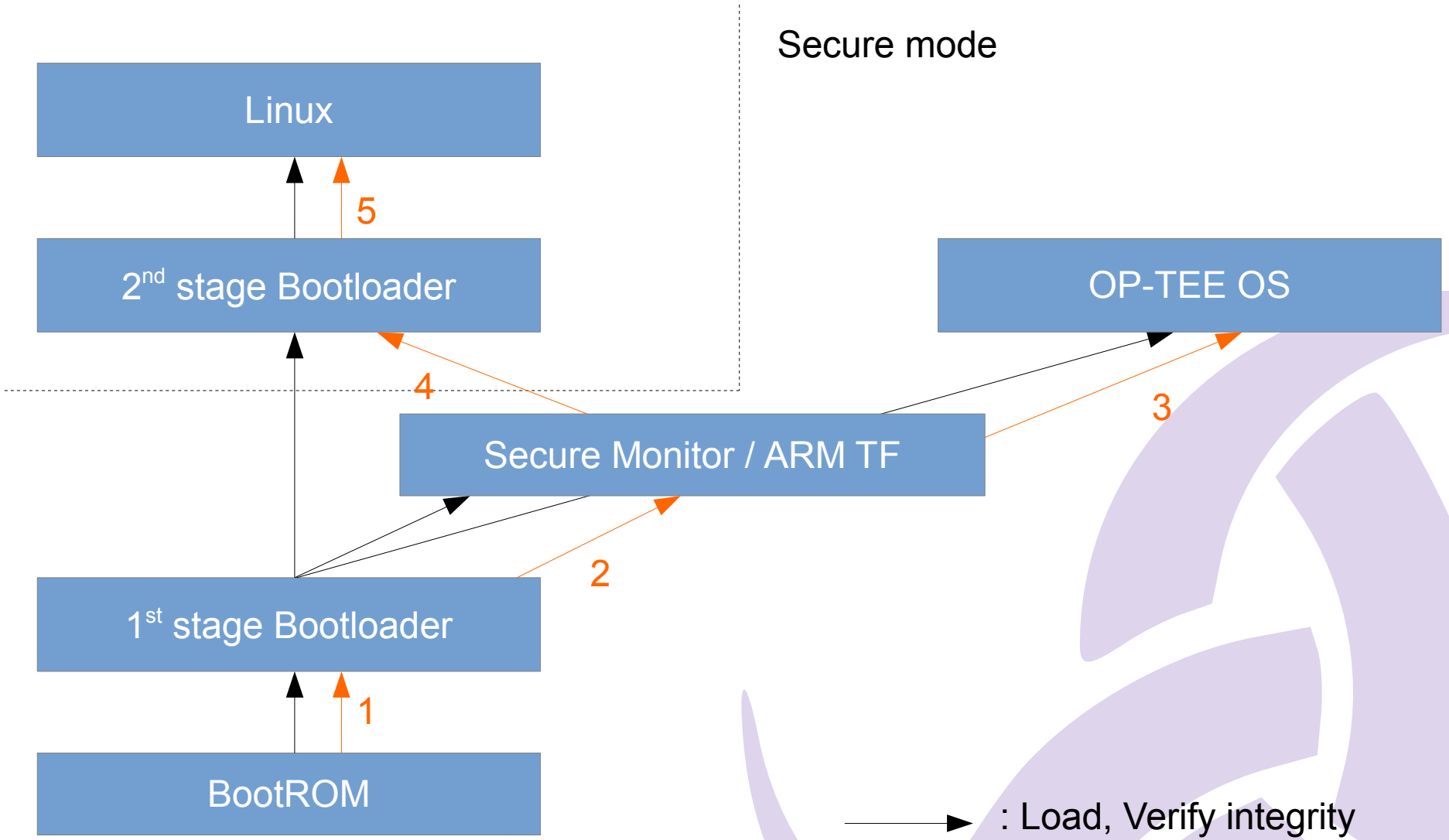
Dynamic view



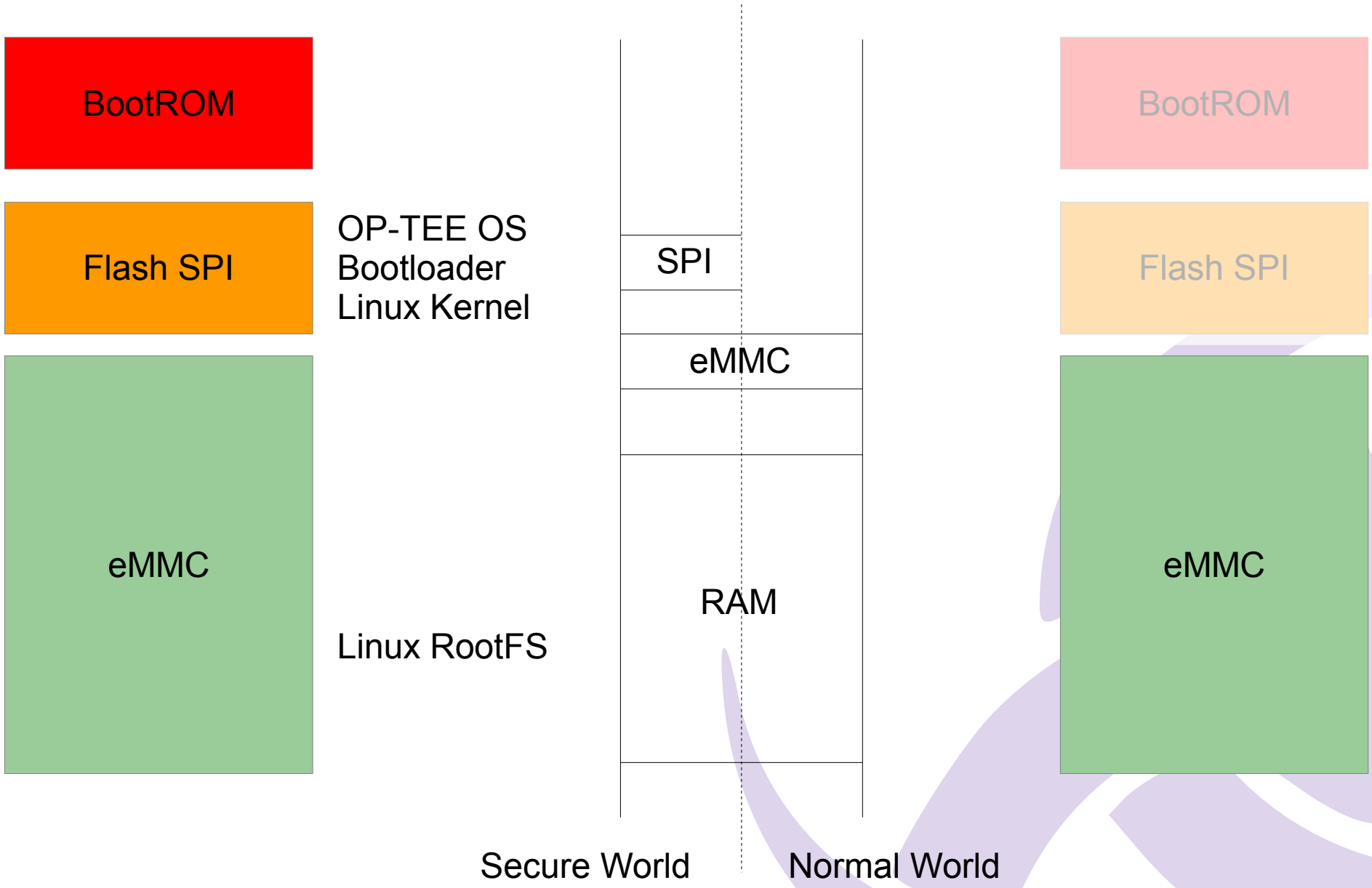
Typical boot sequence



Boot sequence



Data vs. peripherals isolation



Data storage

- **Secure storage**
  - Using Normal world rootfs + cryptography,
  - Using eMMC RPMB (Jedec-84 A) partition,
- **A Storage usage policy may be defined**
  - In regards of distro. packages & SW architecture,
- **Installation strategy to perform the update**
  - Single vs. Dual copy updates,
  - Recovery mode, rollback, persistence,
  - Sw update package format,
  - ...

Status

- **Integration in Yocto/AGL**
  - Layer enabling a QEmu machine with OP-TEE OS + samples apps:

    https://github.com/iotbzh/meta-optee

- **Open points**
  - Security API commonly available in Intel TXT & ARM, architecture that can enforce SOTA,
  - Key management,
  - Updates package format, generation from Yocto,

References

External links:

- http://fr.slideshare.net/linaroorg/hkg15311-optee-for-beginners-and-porting-review
- http://fr.slideshare.net/linaroorg/lcu14-302-how-to-port-optee-to-another-platform

Sources repositories:

- https://github.com/OP-TEE/
- https://github.com/OP-TEE/optee_os/tree/master/documentation
- https://github.com/ARM-software/arm-trusted-firmware