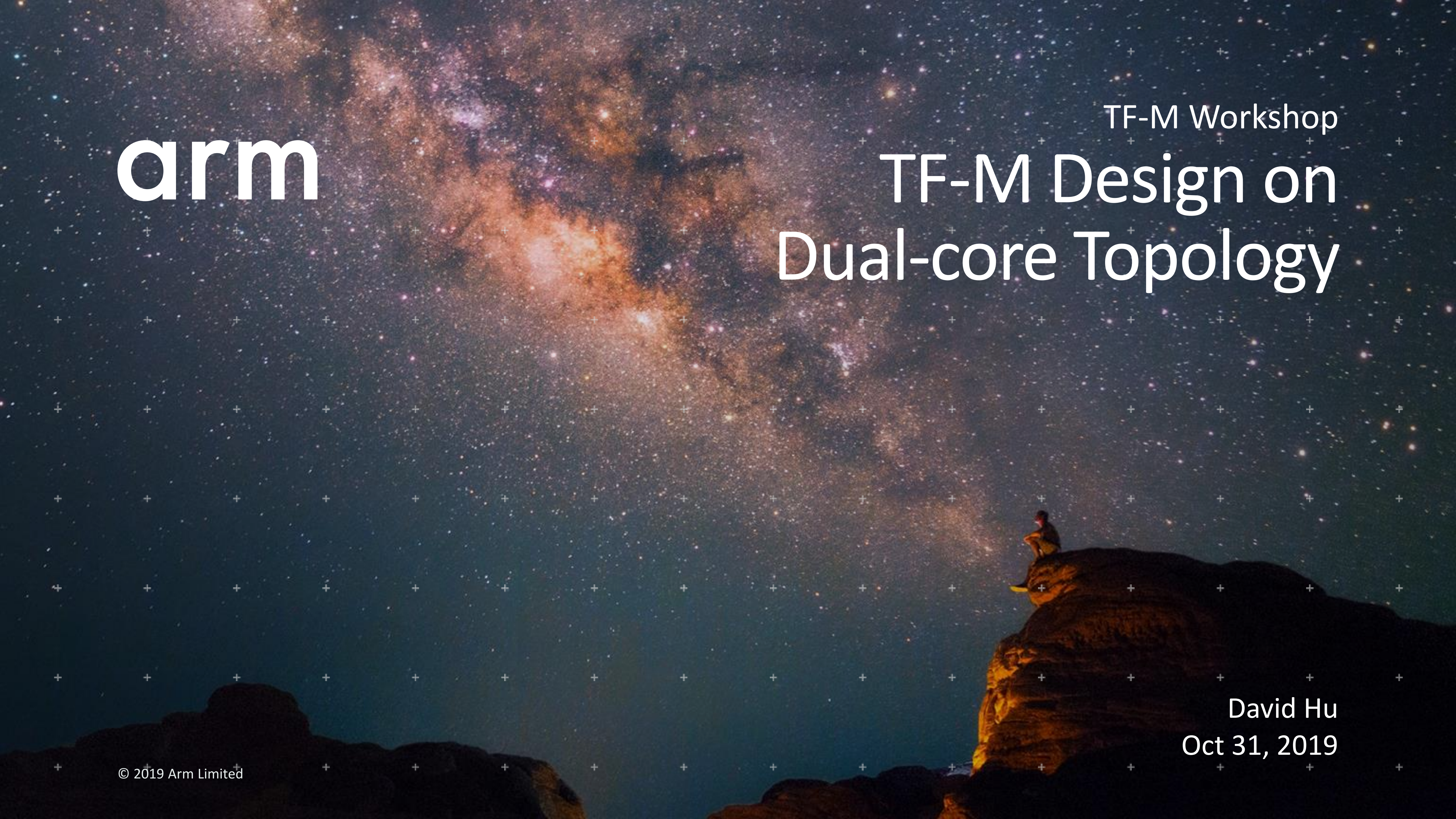


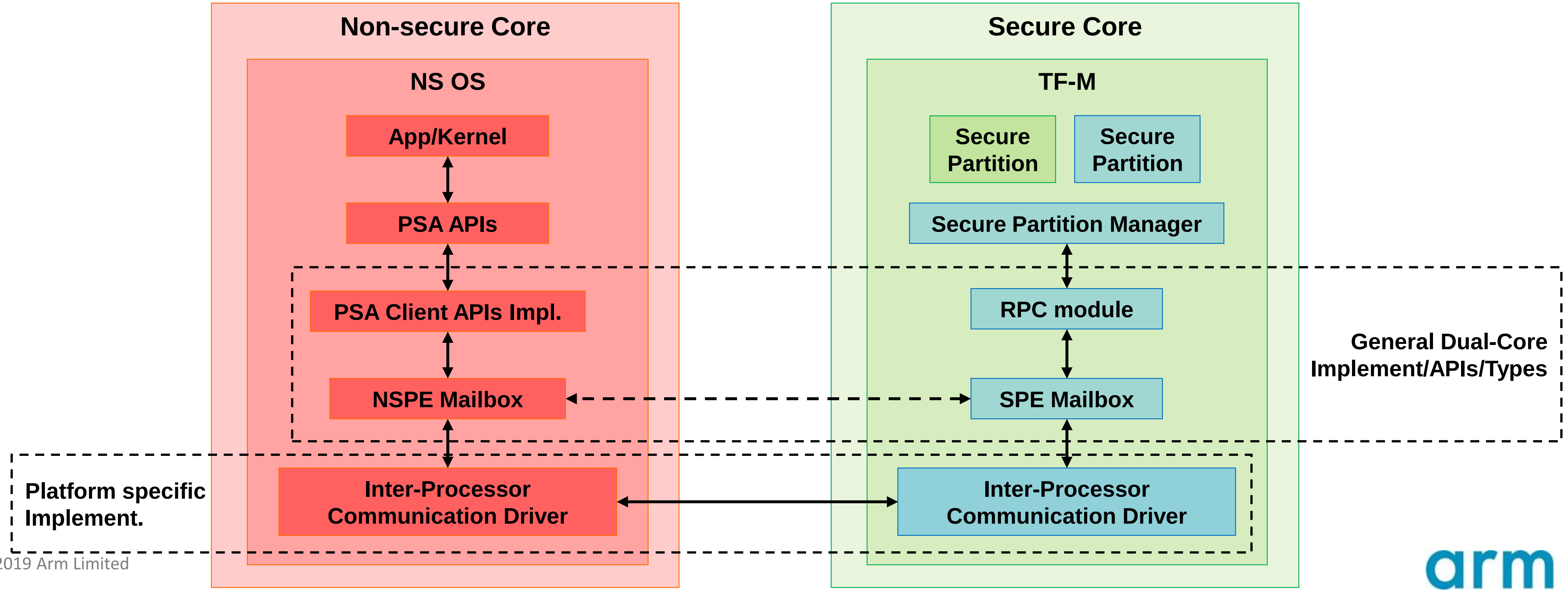
TF-M Workshop

TF-M Design on Dual-core Topology

David Hu
Oct 31, 2019

General dual-core design overview

- A lightweight mailbox dedicated for TF-M for inter-core communication
- Dual-core specific PSA Client API implementation on Non-Secure Side
- Share the same TF-M SPM interface/behavior as that in single Armv8-M TF-M
- Design document online



Current progress and further features

- Current progress
  - Technical collaboration with Cypress on PSoC 6 platform
  - Implement dual-core TF-M Isolation Level 2 feature on Cypress PSoC 6 platform on feature branch.
  - Merging general features from feature branch back to master branch

- Further general features
  - Multiple outstanding PSA Client calls from NS
  - Secure interrupt on dual-core topology
  - Dual-core specific test cases (*Not currently planned*)

Open Discussion

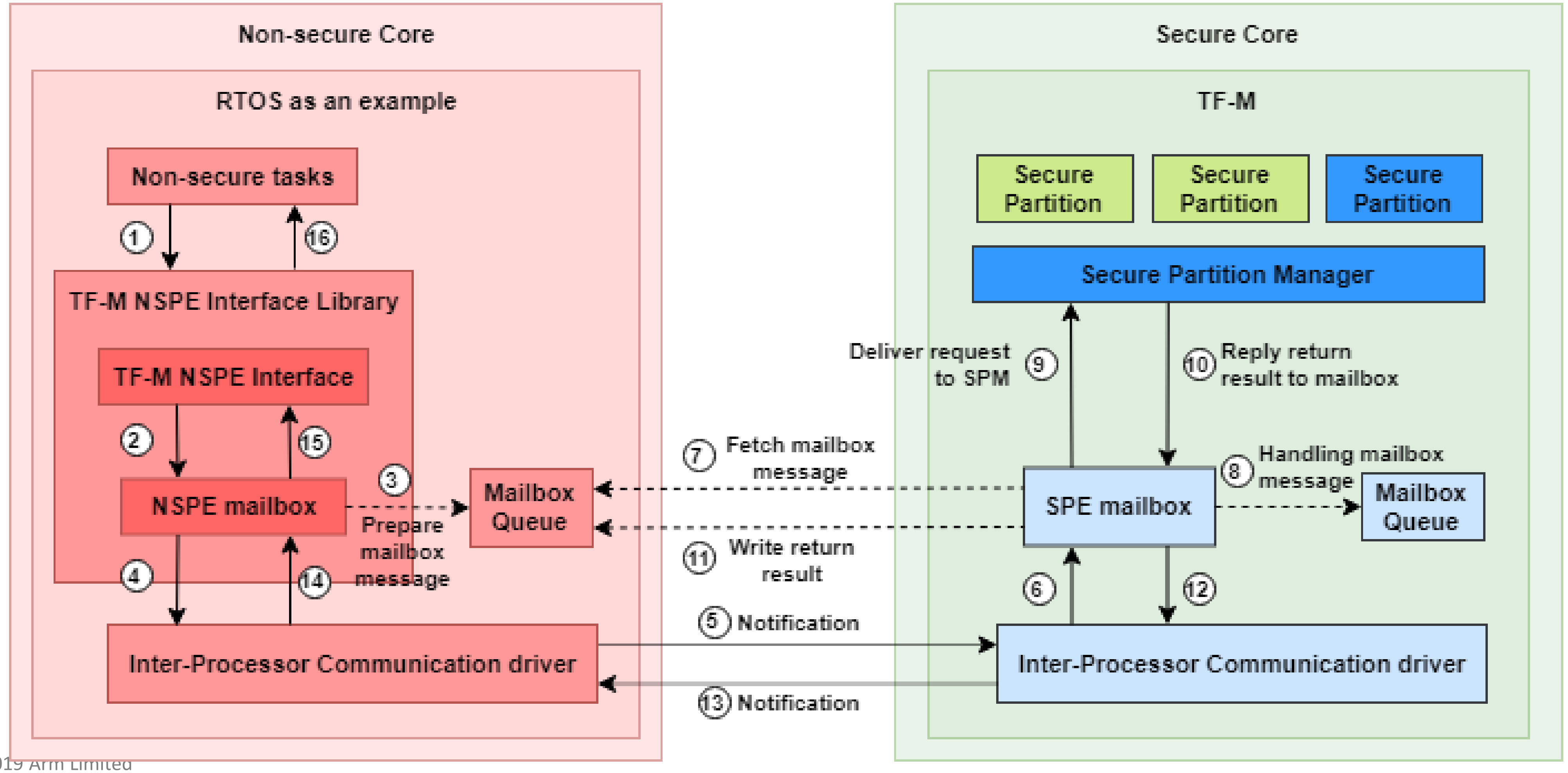
- Multiple outstanding PSA Client call from NS
  - Support multiple NS threads to send PSA Client request simultaneously
  - Changes to TF-M (Mailbox/SPM): recognize the multiple PSA Client call requests/repliesS.
  - Platform/NS OS specific thread waiting/waking mechanism on NS side.
- Pieces of potential enhancement/feature
  - Secure core enters low-power during idle
  - Identification of NS tasks inside TF-M
- Contributing to dual-core features

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה

Back-up Slides

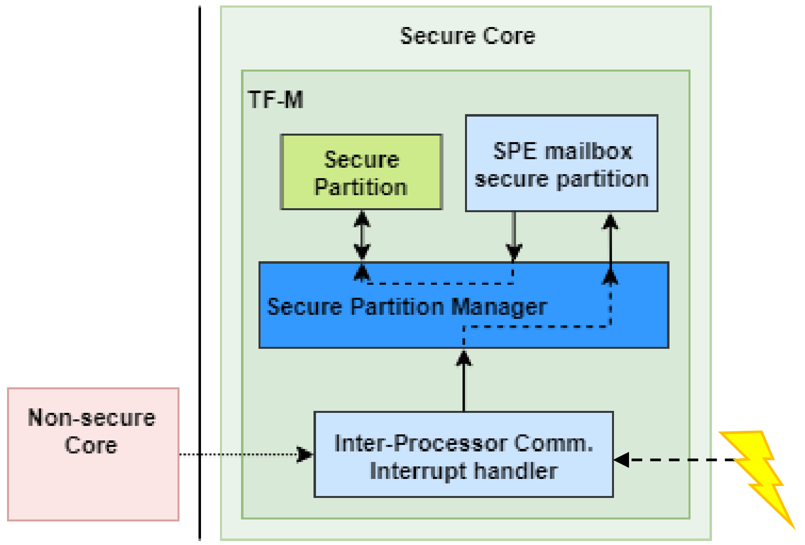
Implementation details

- Key workflow of handling secure service



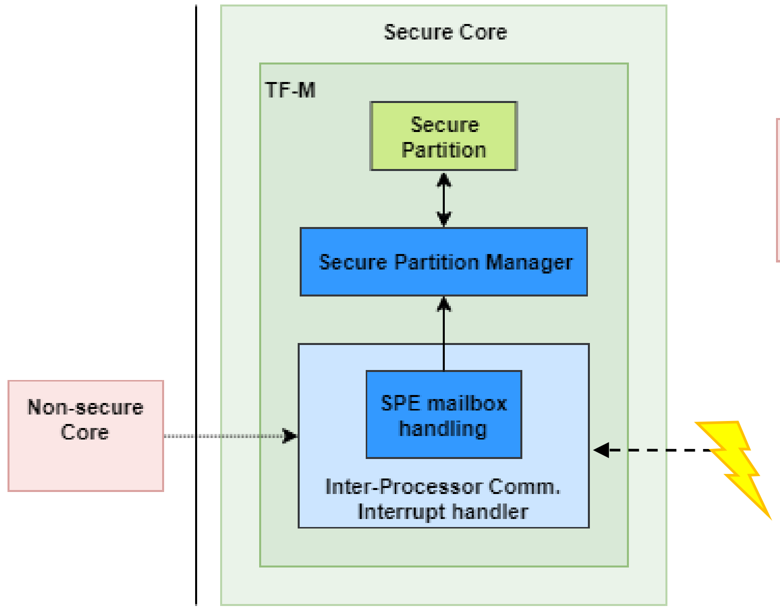
Implementation details (cont.)

- Handle inbound mailbox message in TF-M PendSV handler
  - Decrease the cost brought by Inter-Processor Communication
  - Avoid blocking other urgent secure interrupts
  - Simplify platform specific implementation



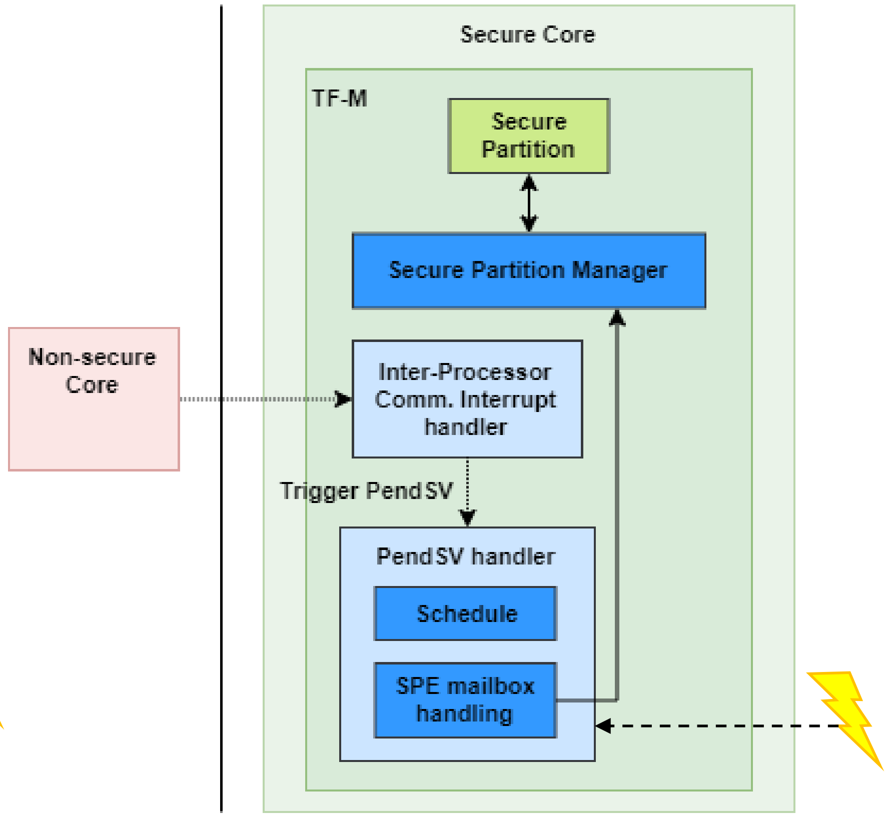
Put mailbox handling in a secure thread as some common IPC/RTOS projects do

- **Long latency**
- **High Complexity**



Put mailbox handling in platform specific IPC dedicated interrupt handler

- **Each platform has to implement the handling**
- **Difficult to set IPC interrupt priority**
  - **Mailbox response vs. Other interrupt source**
  - **Diverse requirements in different use cases/platforms**

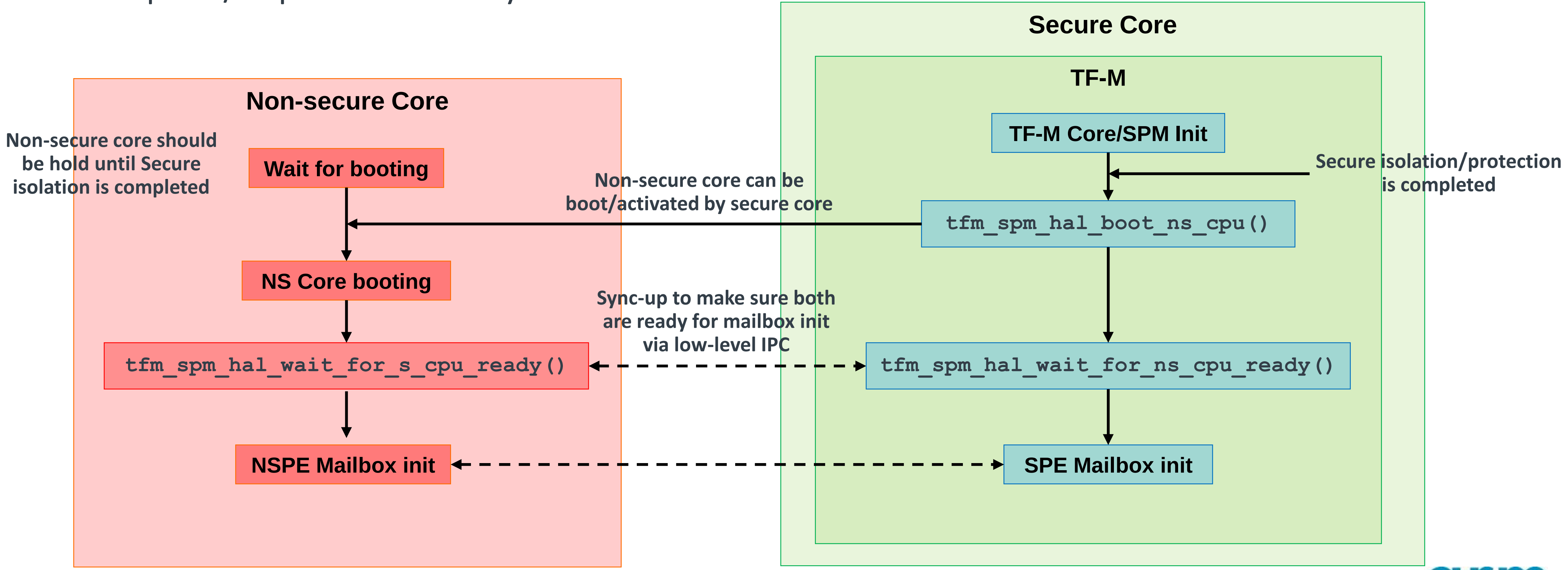


Put mailbox handling in "Half bottom" of PendSV handler

- **Simplify platform IPC handler: Just trigger PendSV**
- **Reasonable latency**
- **Put PendSV into low priority to avoid blocking other urgent events**

Implementation details (cont.)

- Booting a dual-core platform
  - Design document
  - Platform specific implementation of general APIs
  - Assumption/requirements on system architecture

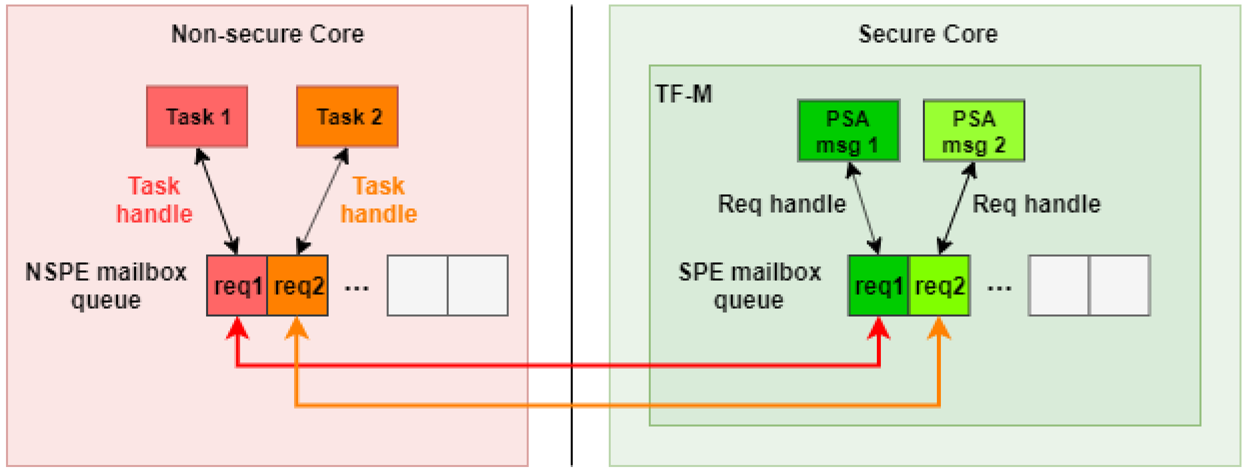


Secure Core
TF-M
TF-M Core/SPM Init
Secure isolation/protection is completed
tfm_spm_hal_boot_ns_cpu()
Non-secure core can be boot/activated by secure core
Non-secure Core
Non-secure core should be hold until Secure isolation is completed
Wait for booting
NS Core booting
Sync-up to make sure both are ready for mailbox init via low-level IPC
tfm_spm_hal_wait_for_s_cpu_ready()
tfm_spm_hal_wait_for_ns_cpu_ready()
NSPE Mailbox init
SPE Mailbox init

Multiple ongoing PSA Client request from NS

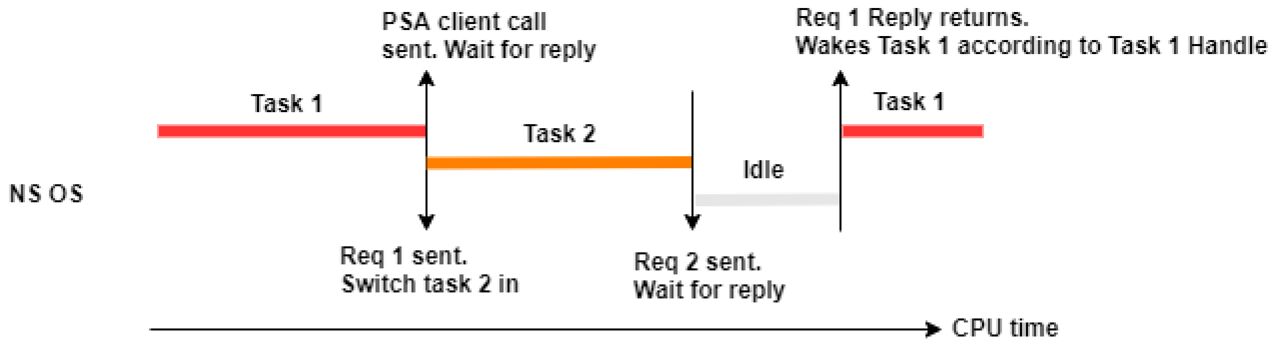
- Enhance concurrency on non-secure core
  - Support of multiple ongoing PSA client call requests

Non-secure task handle stored in mailbox message identifies non-secure task when reply is returned



Mailbox appends mailbox message handle to the corresponding PSA message

The handle identifies the mailbox message when PSA client call is completed


PSA client call sent. Wait for reply
Req 1 Reply returns. Wakes Task 1 according to Task 1 Handle
Task 1
Task 2
Task 1
Idle
NS OS
Req 1 sent. Switch task 2 in
Req 2 sent. Wait for reply
CPU time

Porting TF-M on a dual-core platform

- Platform specific Inter-Processor Communication (IPC) functionalities
  - IPC interrupt
  - Notification functionalities to implement mailbox HAL APIs.

- Platform specific implementation of mailbox HAL APIs
  - Mailbox initialization
  - Synchronization and critical section protection between cores

- Booting sequence
  - Based on platform specific inter-processor communication

- Integration of Non-secure and Secure binaries
  - No veneer binary
  - Additional export files for mailbox
  - Different build configs on dual cores if porting NS demo and test of TF-M