arm CCA

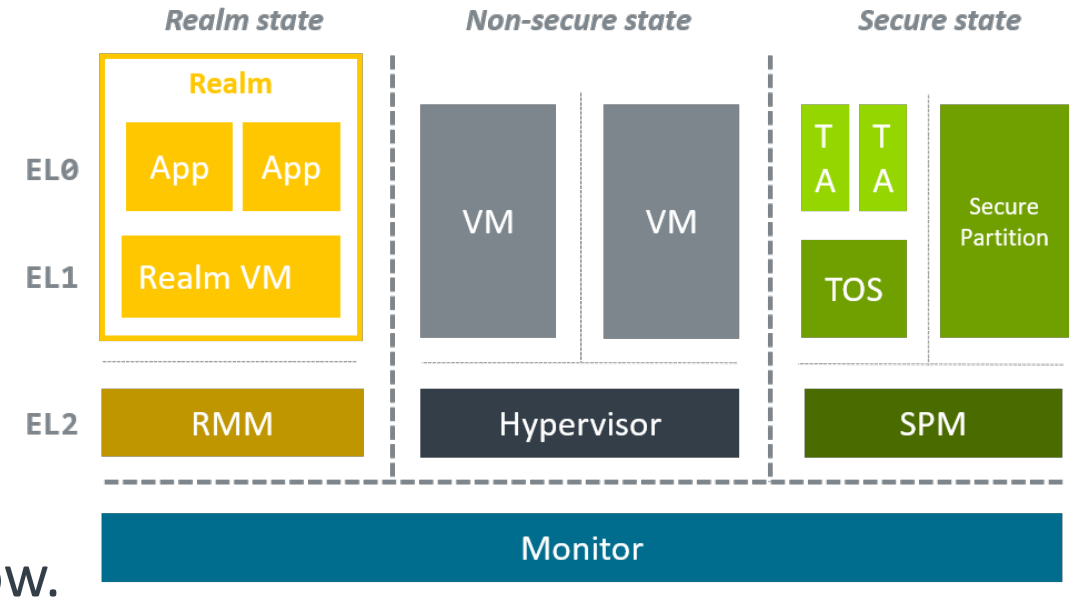
Arm CCA

# TF-A Monitor firmware

Soby Mathew

# Agenda

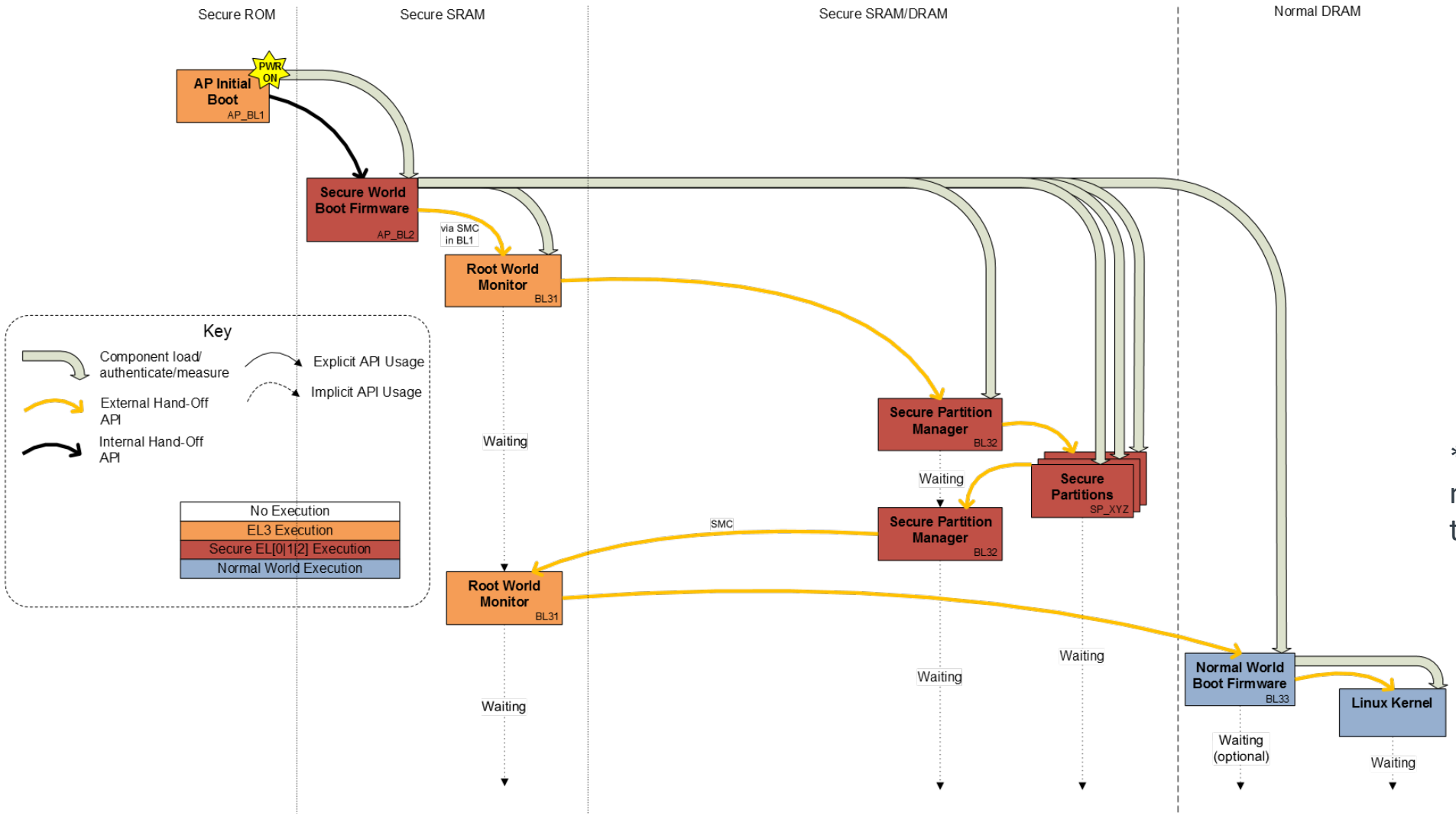
1. Discuss the prospective TF-A changes needed for Arm CCA
   a) Introduce boot flow architecture
   b) Overview of GPT initialization during boot flow
   c) Discuss the 4 world design and its impact on TF-A
2. Prototype implementation in TF-A
   a) Boot flow for the prototype
   b) RMM Dispatcher
   c) Test Realm payload
3. Testing the prototype implementation
4. Pointer to the TF-A branch and development flow.



# Bootflow for Arm CCA systems

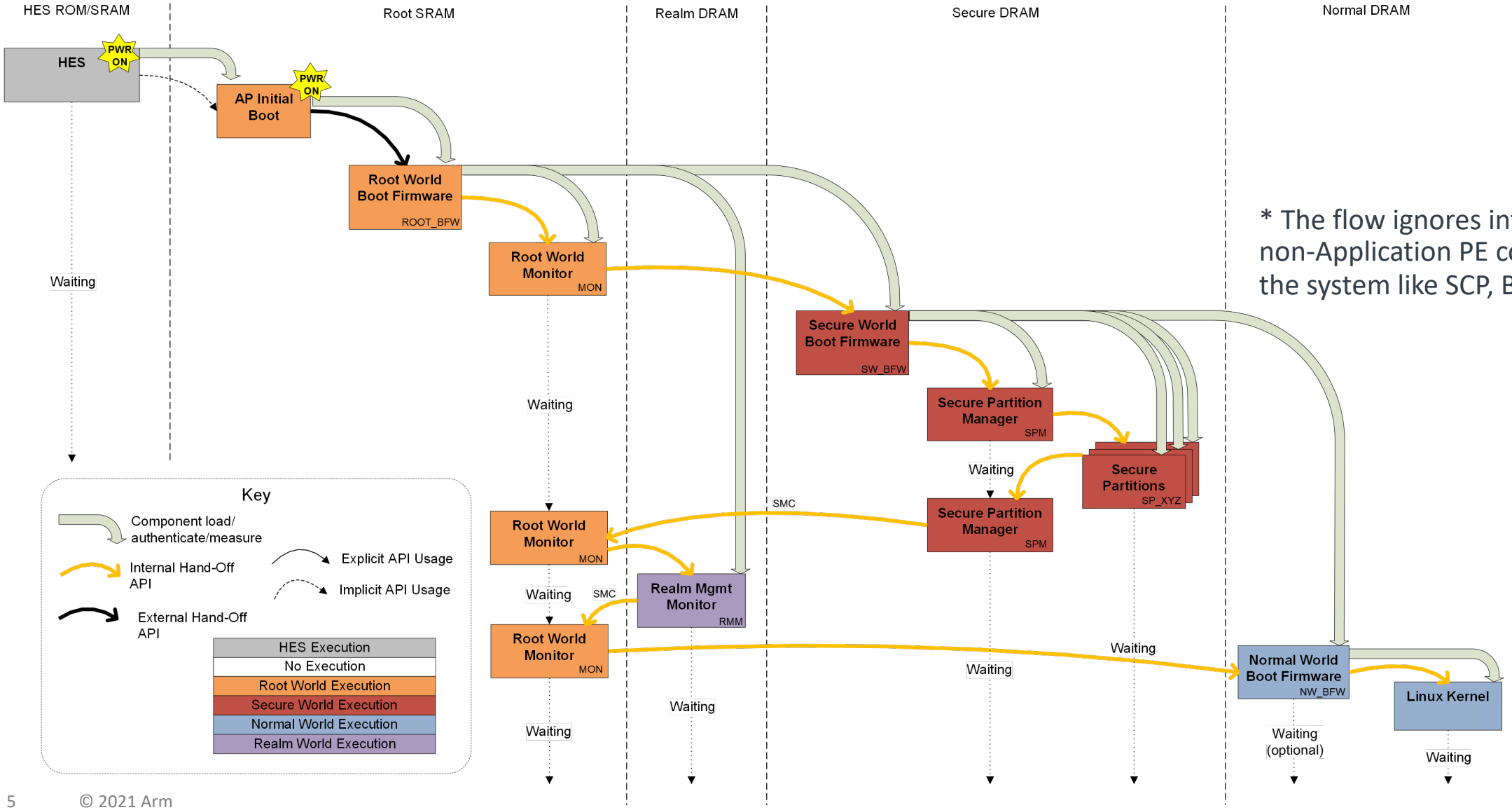
- Arm CCA introduces additional requirements for the bootflow.
  - The Boot firmware must be aware of the 4 Physical Address Spaces (PAS)
  - The Monitor firmware in EL3 and the RMM must be loaded by boot firmware running in Root PAS.
- The Granule Protection Tables (GPT) must be initialized before allocating memory to Secure or Realm worlds for loading their respective firmware.
- Root world boot firmware is responsible for recording measurements of the firmware components it loads.
  - In a HES (Hardware Enforced Security) enabled implementation, the measurements are recorded via service provided by the HES module.

# Typical ARMv8-A reference bootflow with SPM*



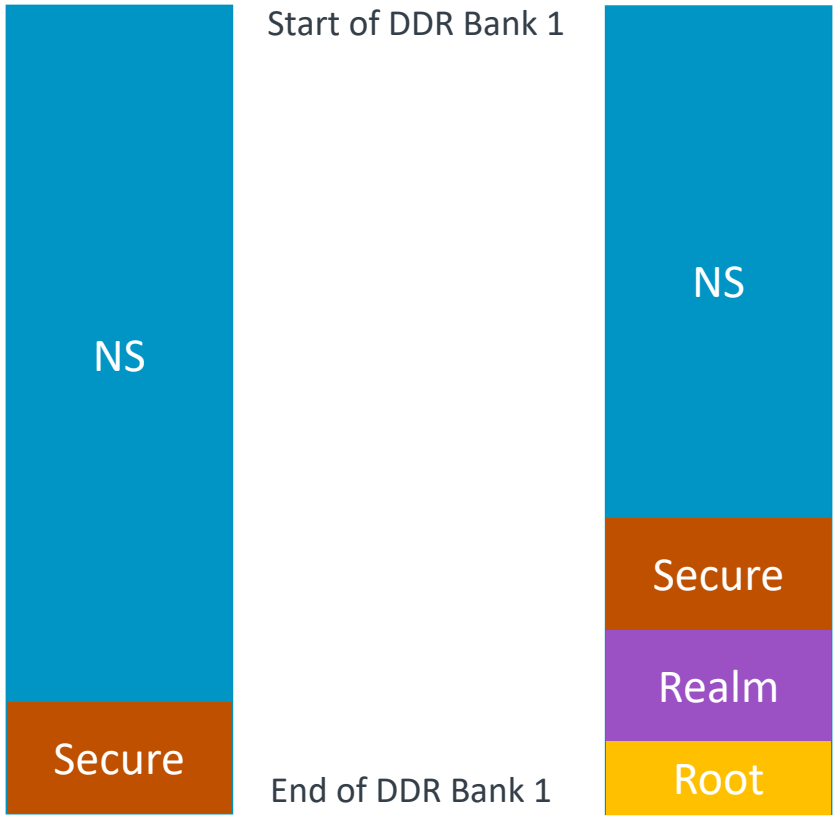

* The flow ignores interaction with non-Application PE components in the system like SCP, BMC etc.

# Proposed Arm CCA reference bootflow*




* The flow ignores interaction with non-Application PE components in the system like SCP, BMC etc.

# TF-A GPT initialisation during boot



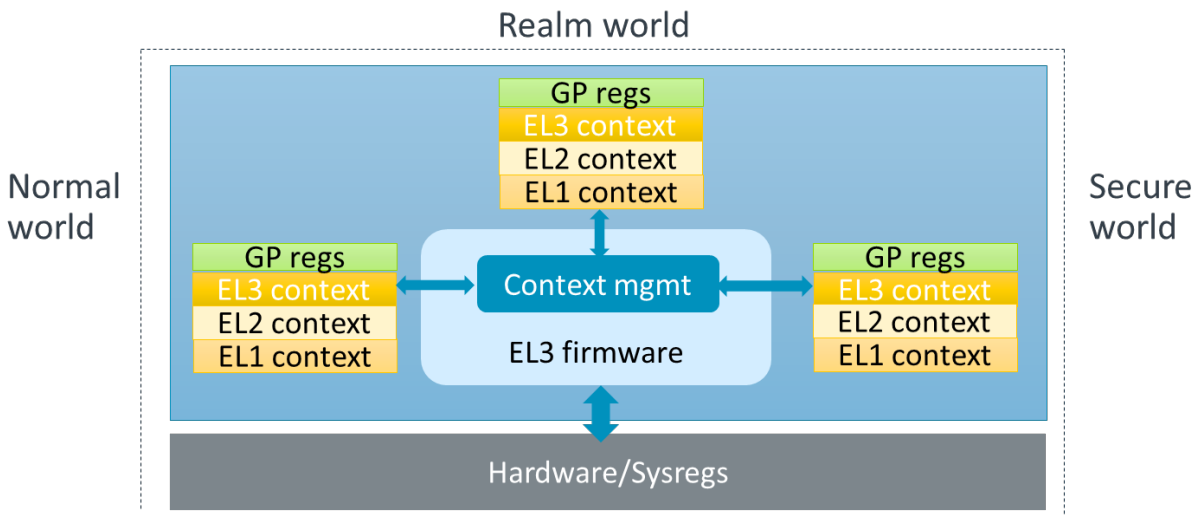

Carveout for TrustZone systems

Carveout for Arm CCA systems

- The GPT is initialized by ROOT_BFW and Root, Realm and Secure world carveouts need to be created.
  - The Root carveout will be used for L1 GPT tables
  - Any additional Root world data in DDR must have additional integrity protection implemented in software.
  - The Realm world carveout is for RMM.
- The Monitor will have a runtime service to dynamically transition memory from one PAS to another PAS – Granule Transition Service.
  - RMM will use this service to delegate/undelegate granules.
  - The SW_BFW can use this service to expand the secure carveout to load the SPM and SPs, or for implementing Trustzone with dynamic memory by SPM.

# Supporting 4 worlds in Monitor firmware

- Monitor needs to switch between Normal, Secure and Realm worlds
  - Add a new CPU context for the Realm world in context mgmt. library.
  - Support new world identity in:
    - SMC handler call path
    - Interrupt handler call path
    - Context management world switch code.
- Introduce the GPT library and add support for migrating memory regions from one world to another at runtime – GTS.
- Enhance translation table lib support to include Root and Realm world attributes.



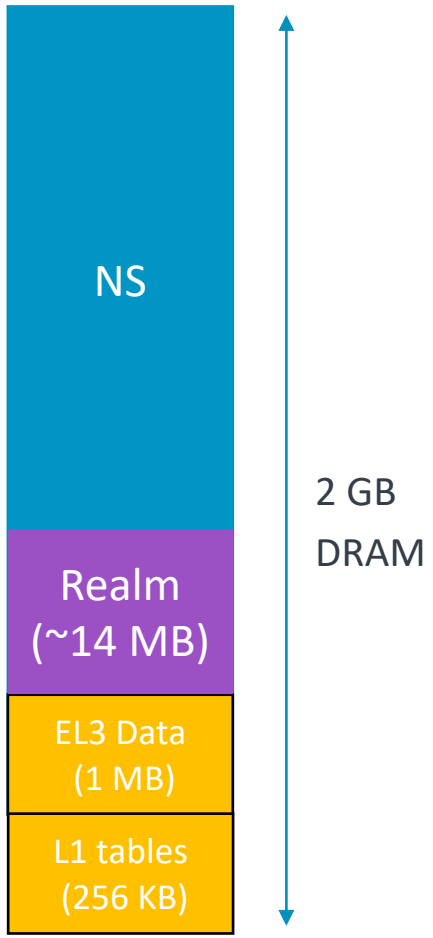

Conceptual view of context mgmt. in EL3

# Supporting 4 worlds in Monitor – Further changes

- Enhance interrupt management in EL3 and External Abort for the 4-world scenario.
  - This affects SDEI and RAS handling code in TF-A. This is an area of investigation.
  - Both Realm and Secure worlds are mutually distrusting.
  - Realm world can get pre-empted by Secure world interrupts. The Secure world dispatcher (SPM Dispatcher) needs to be aware of this.
- Add support for additional dispatcher component for Realm world – RMM Dispatcher (see later slide)
- Impacts to Secure Partition Manager
  - Introduction of Realm world and dynamic migration of memory from one PAS to another has some implications for SPM.
- Usage of DRAM by Monitor (BL31) should have additional integrity protection implemented.
  - Some of the frameworks in TF-A for save and restore of context in DDR needs enhancements for this.

# TF-A Arm CCA prototype – Boot details

- The prototype replaces the Secure world with Realm world
  - Focus is to configure an RME-enabled PE correctly and boot.
  - The Secure world image (BL32) is replaced with Realm world binary.
- BL2 is modified to run at EL3 and load firmware images in Realm and Non-secure address space.
- Introduces a library to create and configure GPT.
  - The library has a similar interface to xlat_table lib. Takes a list of PAS regions with region attributes to configure.
  - Creates L0 block entries for non-DDR regions with ACCESS_ALL attributes. L1 tables are created for the 2 GB of DDR region.
- BL2 creates L0 & L1 GPT tables with a carveout for Realm and Root world as shown in the figure.



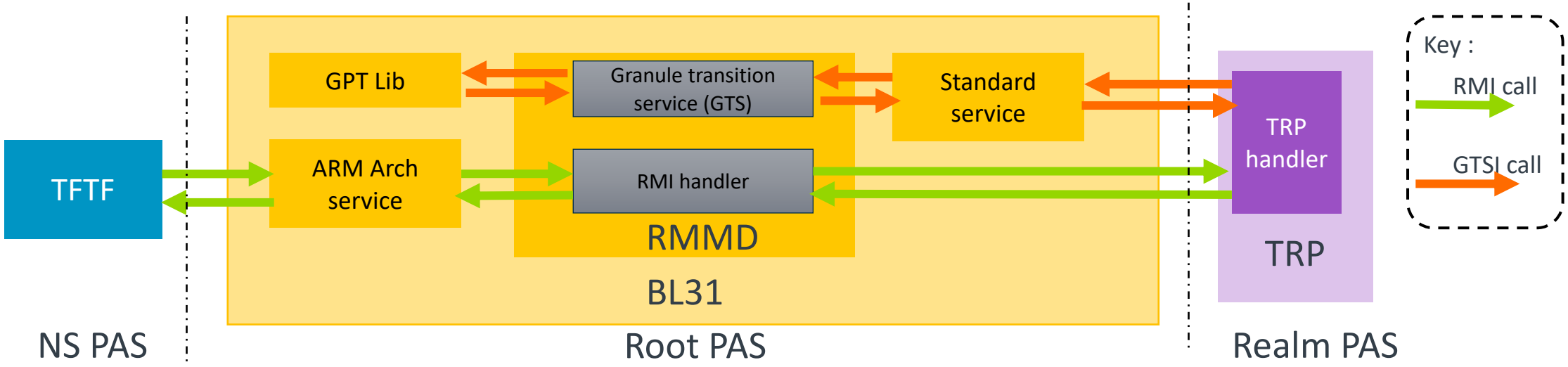
# TF-A Arm CCA prototype – RMM Dispatcher (RMMD)

- The RMM dispatcher is common and is to be included with monitor if RME is enabled.
  - This is similar to the goal to have a common SPM Dispatcher, for platforms with S-EL2.
- The RMI calls are routed between Non Secure and Realm worlds by the RMMD.
- Implements a Granule Transition Service (GTS) to migrate granules from one PAS to another.
  - SMC_ASC_MARK_REALM and SMC_ASC_MARK_NON_SECURE can be invoked from Realm world.
- The PSCI power management operations are registered and notified to RMMD via publish-subscribe mechanism
  - Currently we think only PSCI_CPU_ON finish event needs to be propagated to RMM.
- Currently the Monitor and RMM interface is minimal. This will be enhanced and formalized as part of TF-A implementation.

# TF-A Arm CCA prototype – Test Realm Payload (TRP)

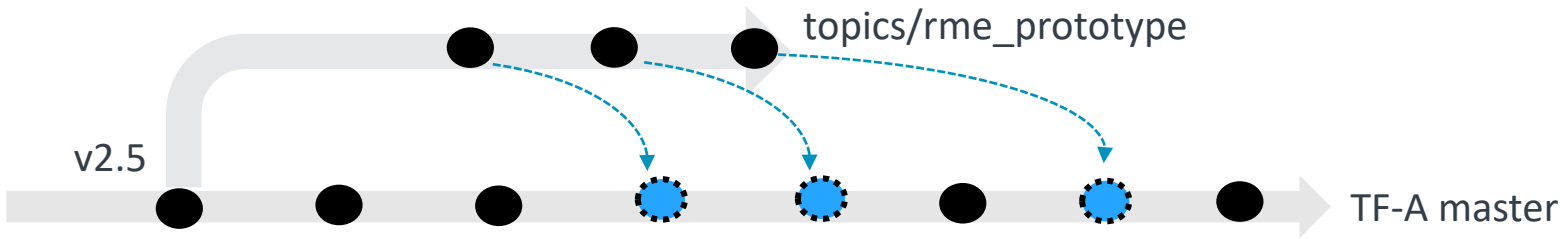
- This is a test payload analogous to Test Secure Payload (TSP).
- Aim of this payload is to test Root monitor interface and functionality.
- Current functionality includes:
  - Initialize UART and C runtime environment for execution in Realm world.
  - Implements a handler which can respond to RMI_VERSION, RMI_DELEGATE and RMI_UNDELEGATE calls from Normal world.
  - Built as the default payload for Realm world if `BL32` is not specified.

# TF-A Arm CCA prototype – Runtime test with TF-A-Tests




- TFTF is one of the test payloads in TF-A-Tests (test framework for TF-A)
- Tests are added to TFTF to issue the following RMI calls :
  - RMI _VERSION
  - RMI_DELEGATE : This call is used to migrate granule from NS to Realm PAS. The TRP will issue SMC_ASC_MARK_REALM to BL31 on receipt of this RMI cmd.
  - RMI_UNDELEGATE : This call is the counterpart of RMI_DELEGATE, used to migrate granule from Realm to NS. The TRP will issue SMC_ASC_MARK_NONSECURE to BL31.

# TF-A Arm CCA prototype – Branch and upstream plans




- TF-A and TF-A-Tests prototypes available at tf.org
  - TF-A branch: https://git.trustedfirmware.org/TF-A/trusted-firmware-a.git/log/?h=topics/rme_prototype
    - All changes for RME in TF-A controlled by `ENABLE_RME` config flag
    - The docs folder has instructions to build and run this on an RME capable FVP.
  - TF-A Tests branch: https://git.trustedfirmware.org/TF-A/tf-a-tests.git/log/?h=topics/rme_prototype
- We expect to start contributing patches to TF-A master soon.
  - The branch is expected to remain static and may be updated on a need basis.
- Updates to TF-A master will use the regular upstream contribution flow:
  - RFC patches in gerrit and discussion on ML.
  - Biweekly TF-A Tech Forum can be utilized to discuss design and code details.

arm

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