Realm Management Extension (RME) Support in TF-A
Agenda

• Brief introduction to Arm CCA
  • Realm Management Extension (RME)
• TF-A changes for RME
Introduction to Arm CCA & RME
Arm Confidential Compute Architecture (CCA)

• Provides hardware-backed secure execution environments called **Realms**

• Code and data in realms is shielded from observation or modification by privileged software and hardware agents (Hypervisor, Host OS, TrustZone)

• Realms are supported at the VM level
  • Hypervisor manages Realm VM resources (scheduling, memory) but can not access those resources

• The platform and realms are attestable

Arm CCA – TrustZone recap

- Two Physical Address Spaces (PAS) and security states: Secure & Non-secure

- Isolation boundary is based on security state

<table>
<thead>
<tr>
<th>Security State</th>
<th>Non-secure PAS</th>
<th>Secure PAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-secure</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Secure</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Physical memory access rules

- No architectural mechanism to dynamically move memory/devices between Secure and Non-Secure PAS
  - Memory for Secure PA is typically statically carved out
Arm CCA – Realm Management Extension (RME)

- Adds two new PAS and security states: **Root & Realm**

- Secure and Realm are mutually distrusting

<table>
<thead>
<tr>
<th>Security State</th>
<th>Non-secure PAS</th>
<th>Secure PAS</th>
<th>Realm PAS</th>
<th>Root PAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-secure</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Secure</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Realm</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Root</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Physical memory access rules
Arm CCA – Realm Management Extension (RME)

- Physical memory access control enforced by HW **Granule Protection Check (GPC)** by MMU

- PAS assignment of every granule (page) of physical memory is described in a **Granule Protection Table (GPT)**
  - GPT is controlled by the monitor in EL3

- Memory can move between physical address spaces dynamically
  (Memory delegation/undelegation)

Arm CCA – Realm Management Monitor (RMM)

• Realm world firmware used to manage the execution of Realm VMs and their interaction with the hypervisor

• Two interfaces:
  • **Realm Management Interface (RMI)** with the Non-secure host
    • Used by host hypervisor to request management control from the RMM (creation, population, execution, and destruction of the realms)
  • **Realm Service Interface (RSI)** with Realm VMs
    • Channel to provide services to Realms such as cryptographic services and attestation
    • channel for memory management requests
EL3 firmware (TF-A) responsibilities

• Loading and running RMM firmware
• Switching between Normal, Secure and Realm worlds
• Management of GPT
Boot flow changes – Typical boot flow with SPM

**Secure ROM**
- AP Initial Boot BL1

**Secure SRAM**
- Secure World Boot Firmware BL2
- EL3 Monitor BL31

**Secure SRAM/DRAM**
- SPM BL32
- Secure Partitions

**Normal DRAM**
- Normal World Boot Firmware BL33
- Linux Kernel

**Key**
- Component load/authenticate/measure
- Hand-off API

- EL3 Execution
- Secure E10/1/2 Execution
- Normal World Execution

© 2021 Arm
Boot flow changes – New boot flow with RMM
Boot flow changes – New boot flow with RMM

- BL2 needs to run at EL3 (Root)
- Can’t use the BL2_AT_EL3 build option
Boot flow changes – New boot flow with RMM

- BL2 loads RMM
- RMM initialized by BL31, similar to SPM
TF-A Changes – Enhancements

• Context management changes
  • New CPU context for Realm world

• Translation library changes
  • Enhanced v2 xlat table library to include Root and Realm world attributes

• Support for RMM image
  • Support for RMM image in fiptool

• New build option, ENABLE_RME
TF-A Changes – New additions

• RMM Dispatcher (RMMD)
  • New standard service, handles RMI SMC calls

• GPT library
  • Enables and initializes the GPT
  • Support for granule transition

• Test Realm Payload (TRP)
  • A small test payload that implements RMM functionalities
  • Supports transitioning granules from Non-secure to Realm world and vice versa
TF-A Changes – Platform changes

- RME support for the FVP platform
  - New memory carve-out
Testing the changes

- New tests added to **TF-A-Tests**
  - Test the RMI interface – Single CPU and multi-CPU granule transition tests

- All patches can be found at [trustedfirmware.org](http://trustedfirmware.org)
Resources

- https://connect.linaro.org/resources/arm-cca
- https://developer.arm.com/documentation/ddi0615/latest
Thank You
Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה