Firmware update design
Agenda

• Firmware update current scenario in Arm platforms
• Overview of firmware update flow
  - Flash controlled by Secure world
  - Flash controlled by Non-secure world
• Firmware update - metadata
• Firmware update - ABIs and state machine
• Firmware update tasks in TF-A
• Unit/Integration tests executed
Firmware update current scenario – Recovery mechanism
Firmware update – Basic Flow

1. Device Management System sends approved firmware build to the Cloud.
2. The Cloud sends firmware update downloads to Device 1, Device 2, and Device N.
Firmware update components

• Update Client
  - Receives new firmware images and initiate the FW update operation

• Update Agent
  - Receives FW images from client and write them to NV storage (Flash)
Firmware Update - Secure world control

- Update Agent in the Secure World
- Client executes in the Non-secure World
Firmware Update – Non-Secure world control

EL0

Non-secure

New FW

EL1

Client and Updater

Non-secure NV memory

bank₁

bank₀

metadata

EL2

EL3
Firmware update metadata

FW update metadata
- crc_32
- version
- active_index
- previous_active_index
- image_entry[#images]

FW image properties
- image_uuid
- accepted
- reserved

FW image entry
- image_type_uuid
- location_uuid
- image_bank_info[#banks]
ABIs and State machine

These ABIs are a contract between caller (Client) and and the callee (Update Agent)

- fwu_discover
- fwu_begin_staging
- fwu_end_staging
- fwu_cancel_staging
- fwu_open
- fwu_write_stream
- fwu_read_stream
- fwu_commit
- fwu_accept_image
- fwu_select_previous
TF-A BL2 involvement in Firmware update

- GPT parser support enablement
- Hardware and Software CRC32 support
- Loading and parsing of Firmware update metadata
- Select the updatable images in non-volatile storage by reading active index (as a part of metadata)
- Avoid NV-counter update in trial run state
Boot Flow 1/2 - BL2 Execution

1. Load FWU metadata to global metadata structure
2. Do sanity check on FWU metadata
3. CRC32 validation on FWU metadata
4. CRC or sanity check failures, repeat step 1 to 3 on backup metadata, if that still fails the checks, then panic.

Set I/O policy for fwu metadata image source

Set I/O policies for all the images loaded by BL2 using active firmware bank index
Boot Flow 2/2 – Trial run detection

- **BL2 common code**
  - bl2_main
  - bl2_image_load_v2
  - bl_common

- **Drivers**
  - auth_mod
  - fwu

- **Flow**
  - bl2_load_images0
    - load all images as per image_info array
  - load_auth_image0
    - load and authenticate image
  - auth_mod_verify_img0
    - Authenticate the image
  - auth_nvctr()
    - Get whether system running in regular or trial state to decide on platform NV counter update
  - fwu_is_trial_run_state()
Firmware bank boot index decision

Each boot_index assignment tried with max_failed_boots

1. boot_index = active_index
2. boot_index = previous_active_index if,
   active_index != previous_active_index, otherwise step 3
3. boot_index = <recovery bank index> [IMP defined]
4. Propogate boot index to update agent using platform defined way
Verification

• Manually created GPT image with two FIP and two FWU metadata images
• Created sample FWU metadata binary to check various flow in the code using FVP model
  - loading of partition table
  - loading of metadata in SRAM
  - CRC32 calculation and verification of metadata
  - regular state vs trial state
  - select FIP A vs FIP B
  - avoid NV counter update in trial run state
• Patches are posted externally for review:
Guid Partition Table Scheme

- LBA 0
- LBA 1
- LBA 2
- LBA 33
- LBA 34
- LBA -34
- LBA -33
- LBA -2
- LBA -1

Protective MBR
- Primary GPT Header
  - Entry 1
  - Entry 2
  - Entry 3
  - Entry 4
  - Entries 5–128

Partition 1

Partition 2

Remaining Partitions
- Entry 1
- Entry 2
- Entry 3
- Entry 4
- Entries 5–128

Secondary GPT Header

manishbadarkhe@manishbadarkhe-VirtualBox:~/.trusted-firmware-a$ gdisk -l sample-fwu-gpt.img
GPT fdisk (gdisk) version 1.0.4

Partition table scan:
MBR: protective
BSD: not present
APM: not present
GPT: present

Found valid GPT with protective MBR; using GPT.
Disk sample-fwu-gpt.img: 2048 sectors, 1024.0 KiB
Sector size (logical): 512 bytes
Disk identifier (GUID): 85A7AAE8-DF95-44A8-AA89-F8DB67842AC2
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 2014
Partitions will be aligned on 2-sector boundaries
Total free space is 361 sectors (180.5 KiB)

<table>
<thead>
<tr>
<th>Number</th>
<th>Start (sector)</th>
<th>End (sector)</th>
<th>Size</th>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>833</td>
<td>400.0 KiB</td>
<td>8300</td>
<td>FIP_A</td>
</tr>
<tr>
<td>2</td>
<td>834</td>
<td>1633</td>
<td>400.0 KiB</td>
<td>8300</td>
<td>FIP_B</td>
</tr>
<tr>
<td>3</td>
<td>1634</td>
<td>1643</td>
<td>5.0 KiB</td>
<td>8300</td>
<td>FWU-Metadata</td>
</tr>
<tr>
<td>4</td>
<td>1644</td>
<td>1653</td>
<td>5.0 KiB</td>
<td>8300</td>
<td>Bkup-FWU-Metadata</td>
</tr>
</tbody>
</table>
Prototype – Flash in secure side

- This is currently tested on QEMU platform, with u-boot and running StMM on top of OPTEE
- Driver is implemented in StMM to flash the images in secure flash
Ongoing tasks

• Boot index decision - Max boot retry with active FIP, switch back to previous active FIP
• Arm platform – Recovery flow implementation
• Integrate TF-A patch work with a total compute platform stack to exercise full firmware update flow
Thank You
Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
شكرًا
धन्यवाद
תודה