Code sharing

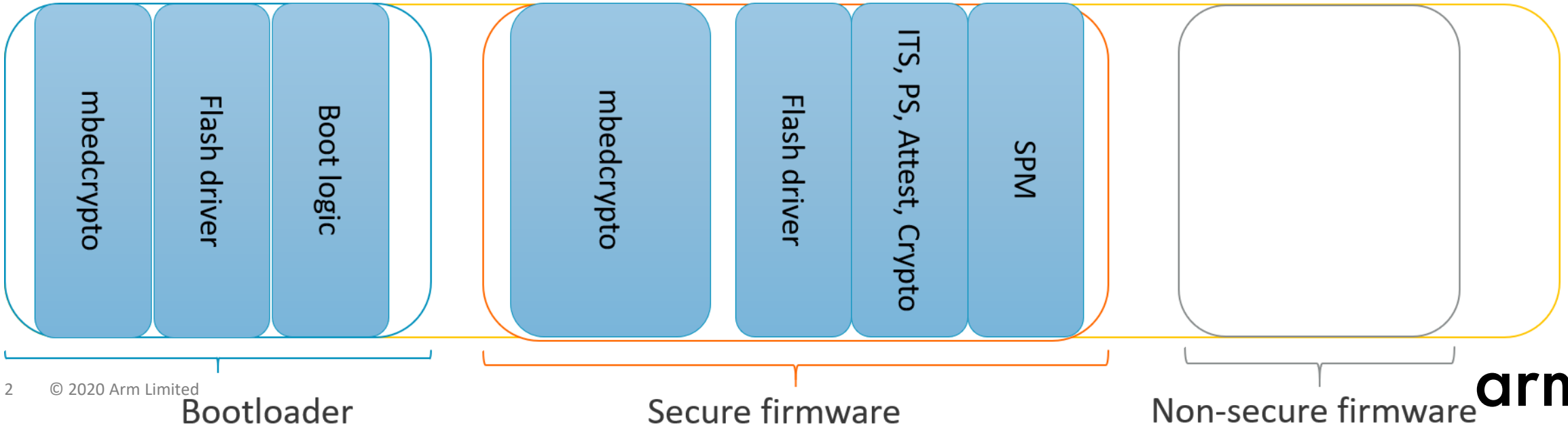
Trusted Firmware M

Tamas Ban
Arm

© 2020 Arm Limited

Motivation

- Cortex-M devices are usually constrained in terms of RAM and flash.
- Secure boot and runtime crypto service has overlapping functionalities.
- Same peripheral drivers are used in bootloader and runtime TF-M.
- Flat memory space, relocation usually not supported.
- **Reuse the common code from bootloader and reduce the memory footprint.**



© 2020 Arm Limited
arm

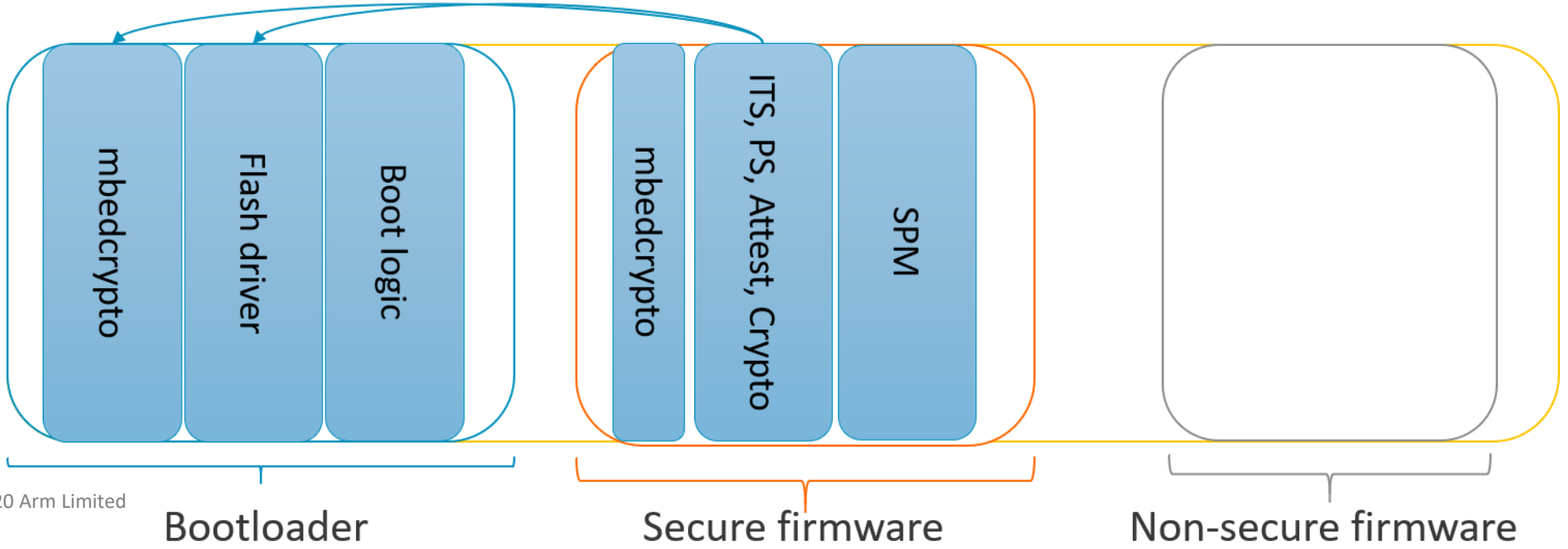
How does it work?

- No standard solution, toolchain dependent
- Manual investigation for shareable code
- Adjust symbol template file, which contains the name of shareable functions
- Sharable function name and address is extracted from bootloader at link time
- The artefacts of shareable code is added to secure firmware at link time
- To avoid symbol collision, shared symbols in secure firmware libraries must be mark as WEAK
- Linker picks up symbols from bootloader code instead of libraries (mbed-crypto, platform_s, etc.)

© 2020 Arm Limited

What type of code can be shared?

- Public functions and global variables
- Easy to share functions with local variables only
- Functions relying on global variables is a bit tricky
- Global variables must be placed to shared symbol section



© 2020 Arm Limited
arm

Gain in flash utilization

- MinSizeRel build type
- Version: 465f73cdedfba5fac6b7430baa4a424d789fed8f + code sharing patches
- MCUboot image encryption is disabled
- Size of secure image:

	ConfigDefault		ConfigProfile-M		ConfigProfile-S	
	ARMCLANG	GNUARM	ARMCLANG	GNUARM	ARMCLANG	GNUARM
CODE_SHARING=OFF	130012	123440	83612	80244	55008	50352
CODE_SHARING=ON	120560	114368	77524	74340	53484	49088
Difference	**9452**	**9072**	**6088**	**5904**	**2524**	**1264**

- MCUboot image encryption is enabled: saving is up to ~13-15KB

© 2020 Arm Limited
arm

Useability

- If bootloader is immutable then bug in shared code cannot be fixed with firmware upgrade.
- Global variables must be to shard symbol section.
- Shared global variables might need to be reinitialized in SPE explicitly, low level start-up does not do it.
- Compiler flag alignment(?)
- Shared code artefacts must be archived because they are needed when new secure image is built.
- https://review.trustedfirmware.org/c/TF-M/trusted-firmware-m/+/4587

© 2020 Arm Limited