



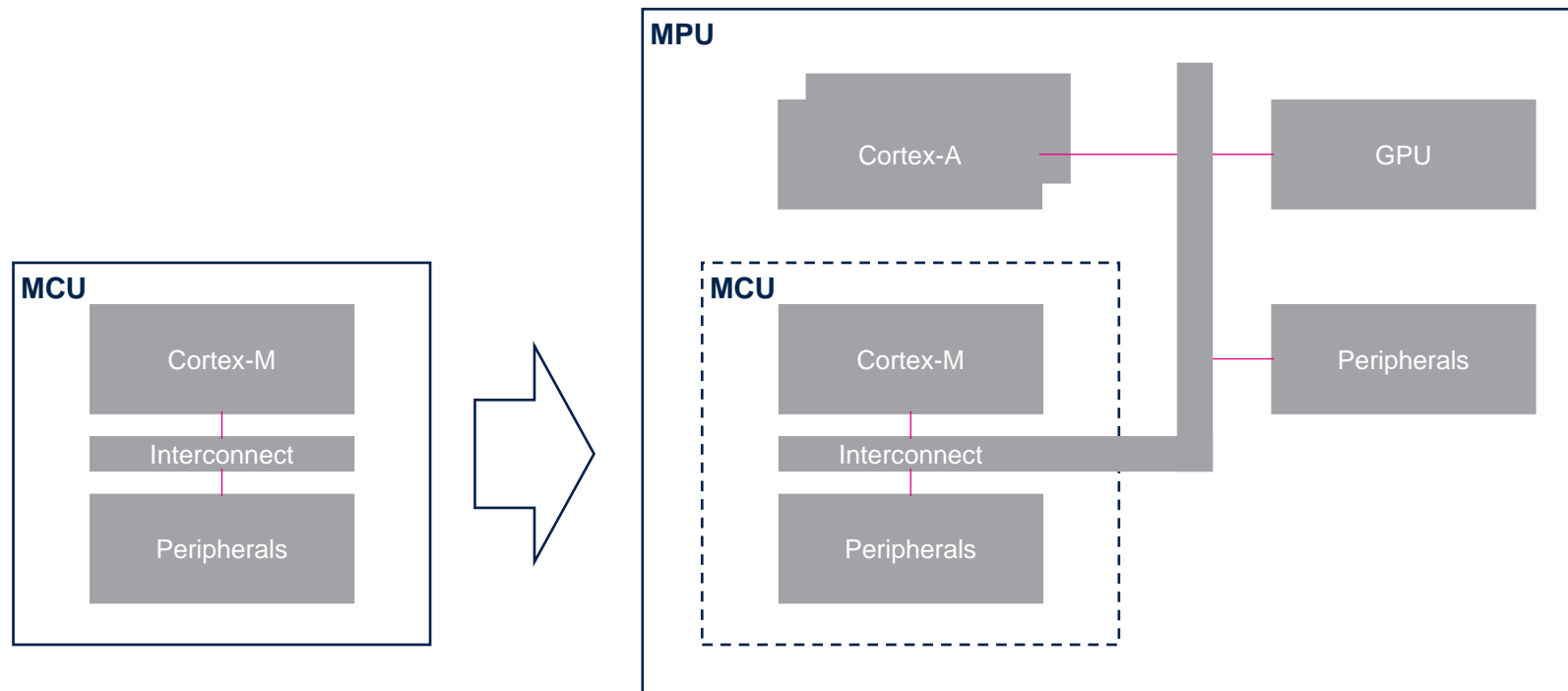
life.augmented

STM32MPU embedded software architecture

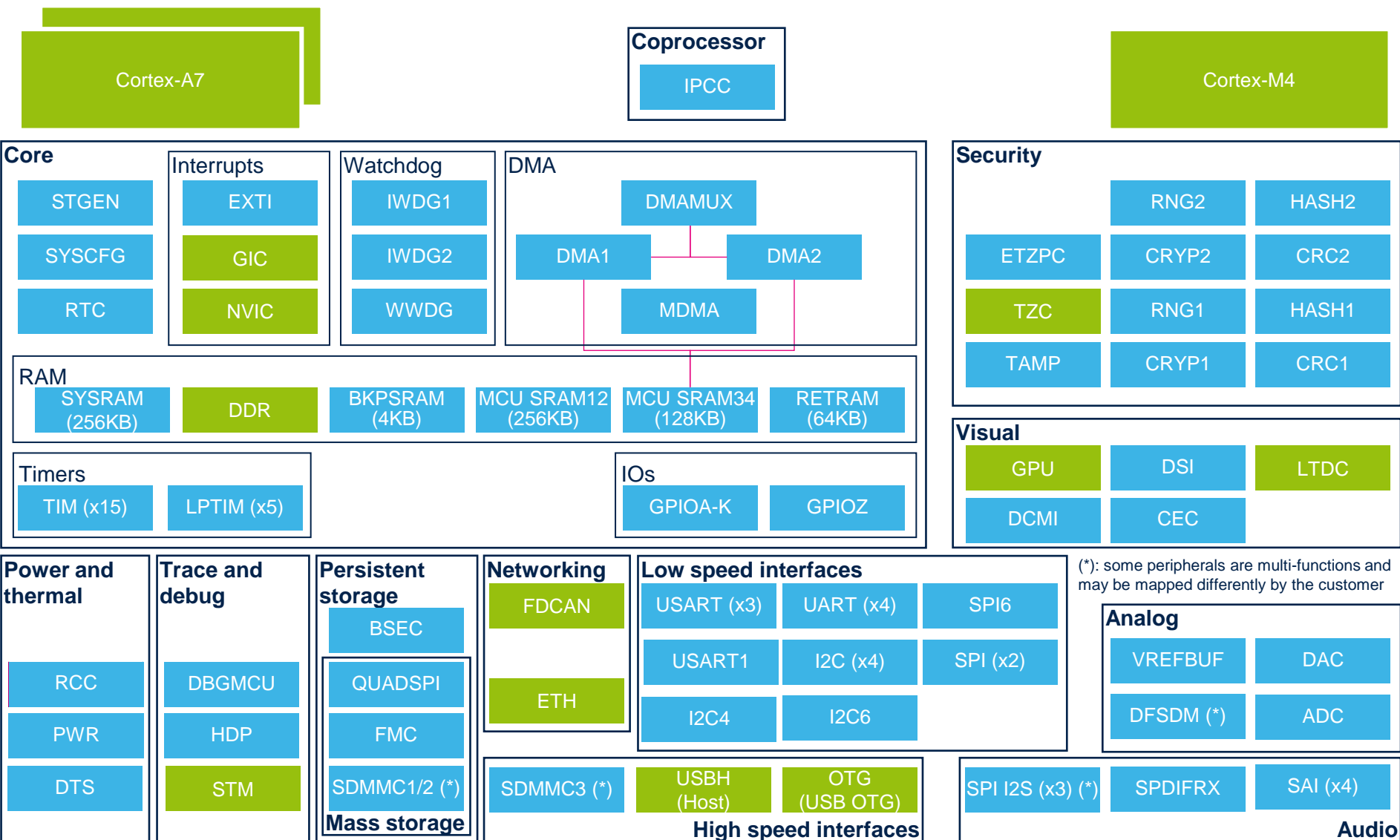
- Peripherals overview and sharing
- Software memory mapping
- STM32MPU Embedded Software

Peripherals overview and sharing

From MCU to MPU

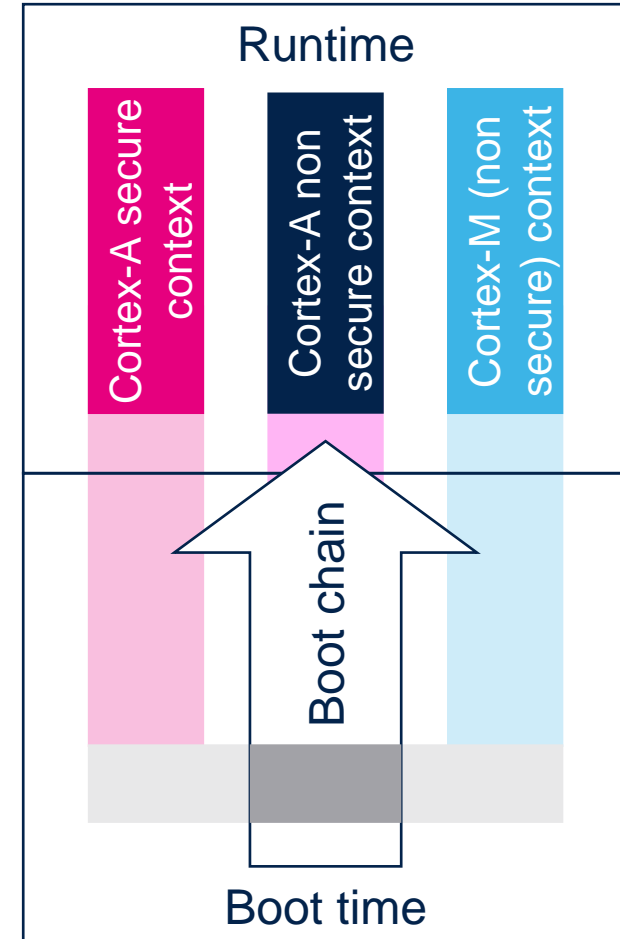


Hardware blocks sourcing



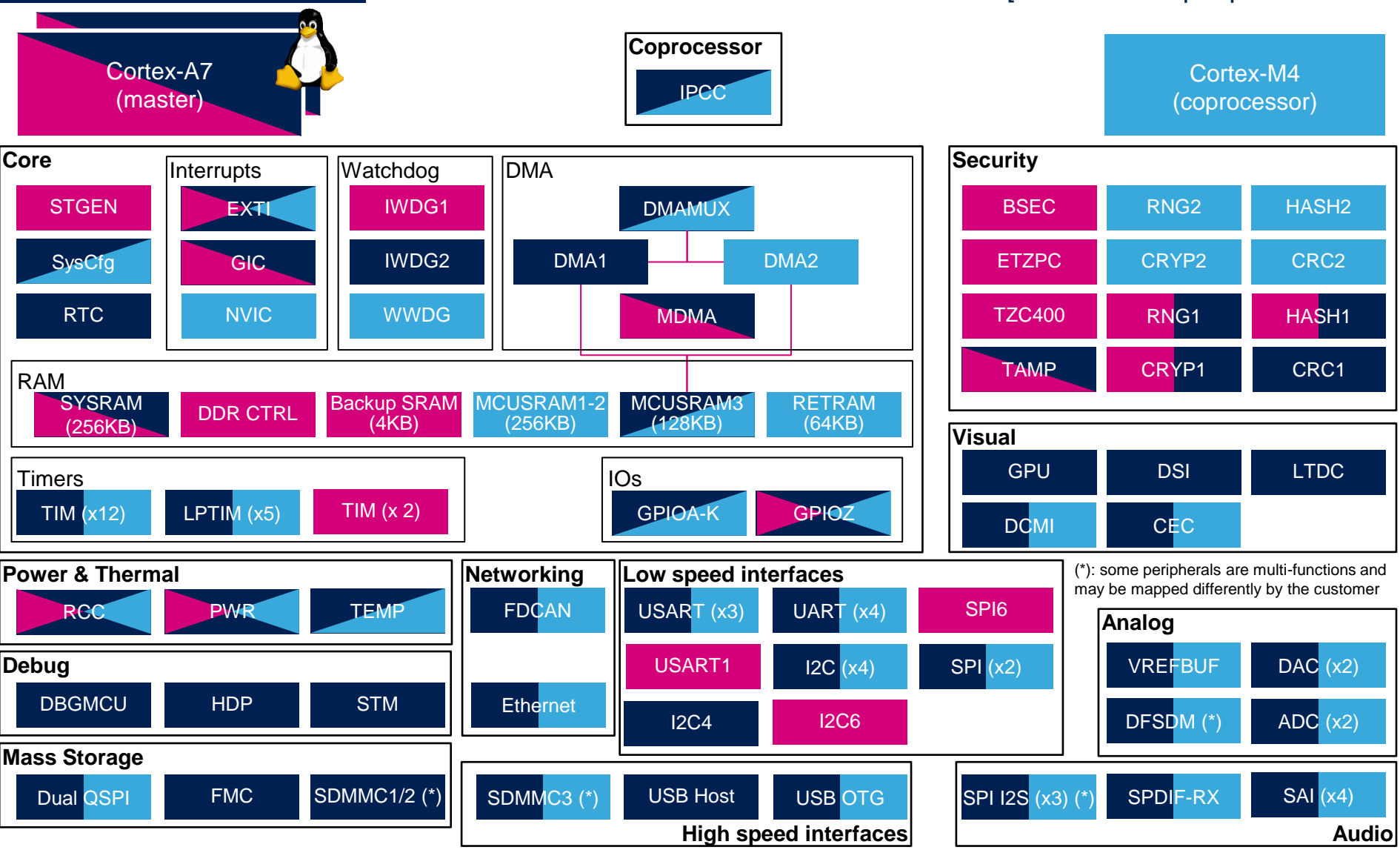
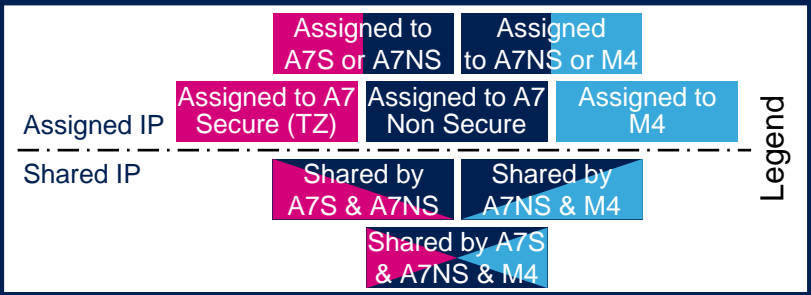
Multiple-core architecture concepts

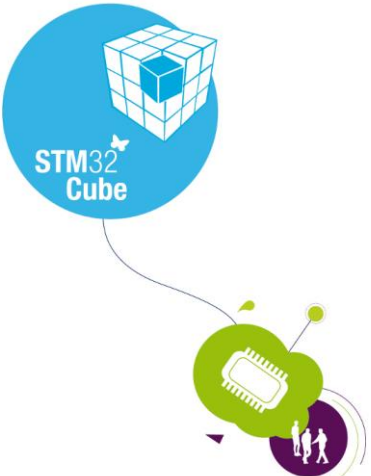
- Hardware execution context
 - « a core and a security mode »
- Firmwares executed runtime contexts
 - Arm Cortex-A secure (Trustzone) executes OP-TEE
 - Arm Cortex-A non secure executes Linux
 - Arm Cortex-M (non secure) executes STM32Cube
- Peripheral assignment to the runtime contexts
 - Assigned or shared



Peripherals sharing

Source: ST Wiki article [STM32MP15 peripherals overview]



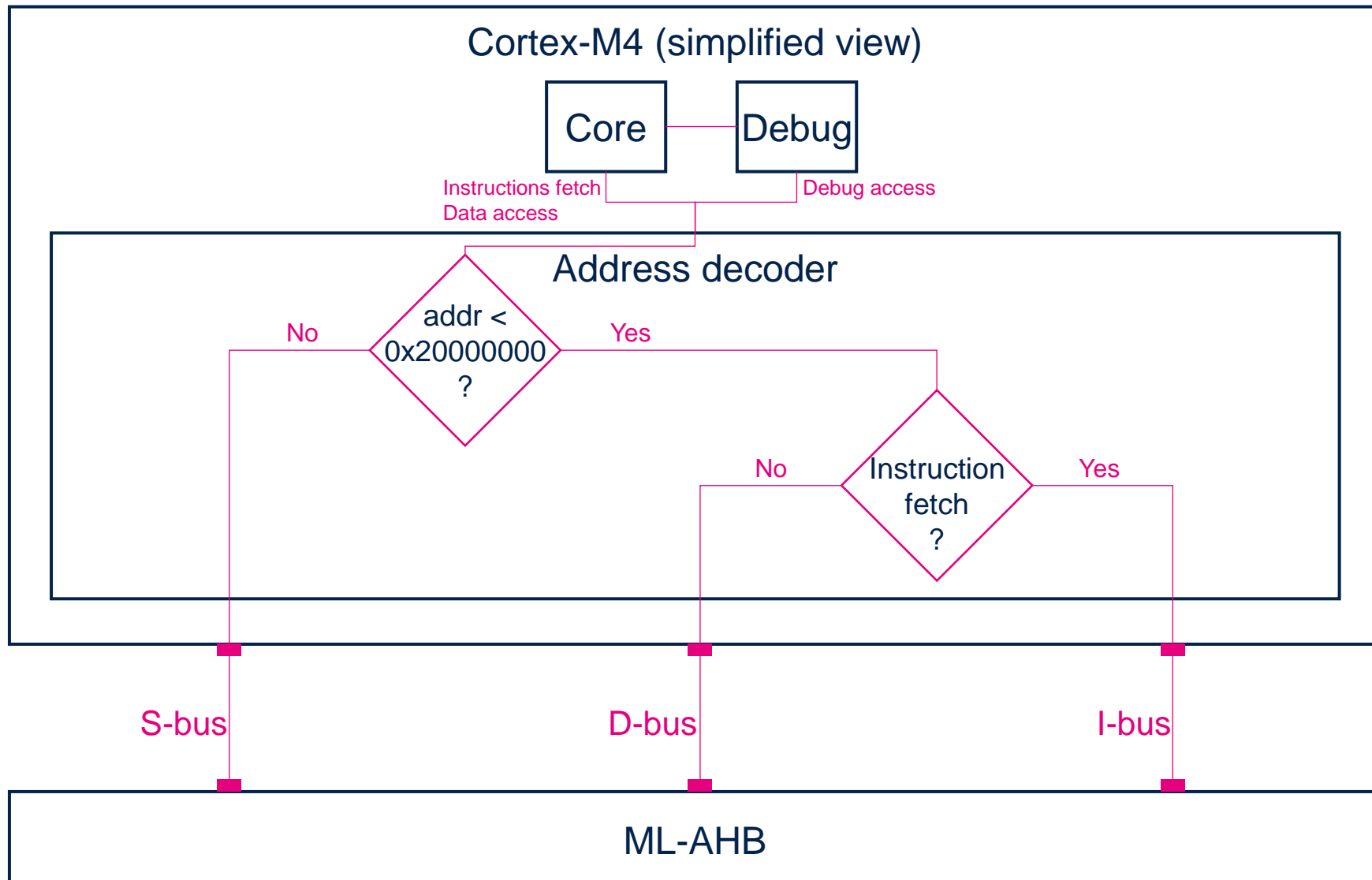


Peripherals assignment via stm32cubemx

Options					
Categories	A->Z				
	Boot ROM	Boot loader	Cortex-A7 secure	A7NS	Cortex-M4
✓ USART1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
✓ USART2	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>
✓ USART3	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>
✓ USART6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>

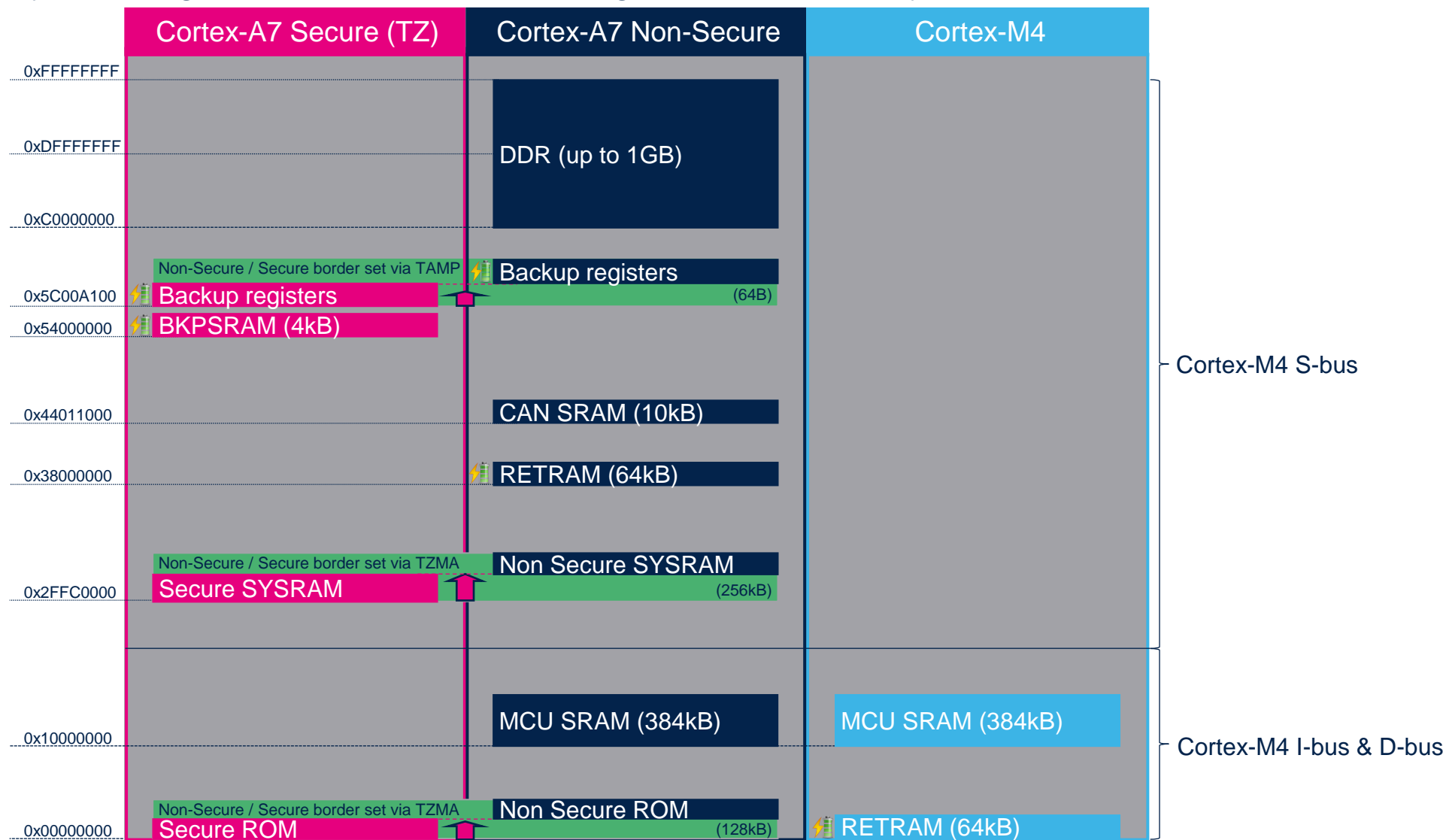
Software memory mapping

Cortex-m4 ports



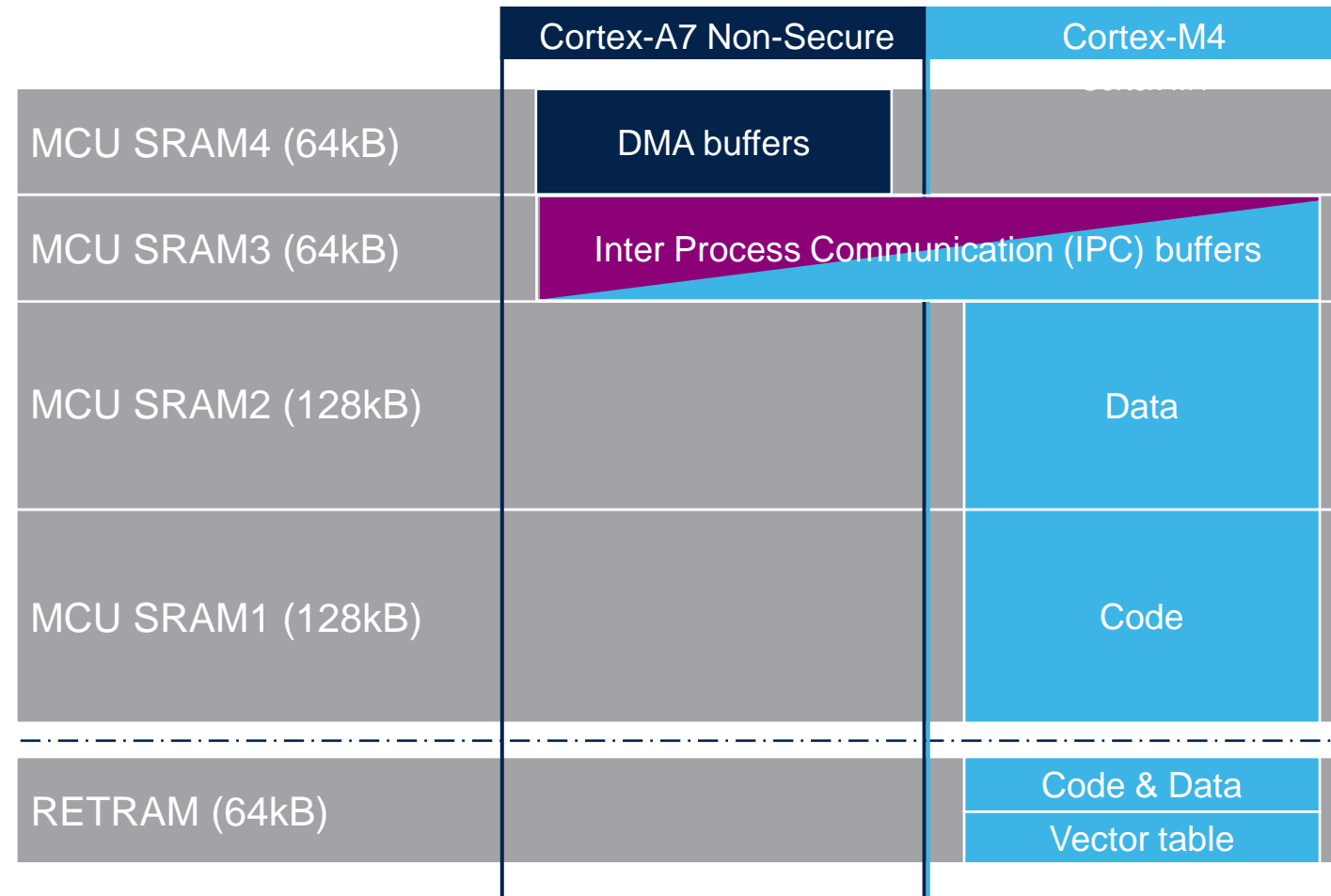
Software memory mapping

- The memory mapping below is a subset of all regions that are really exposed at hardware level.



Shared RAM memory mapping

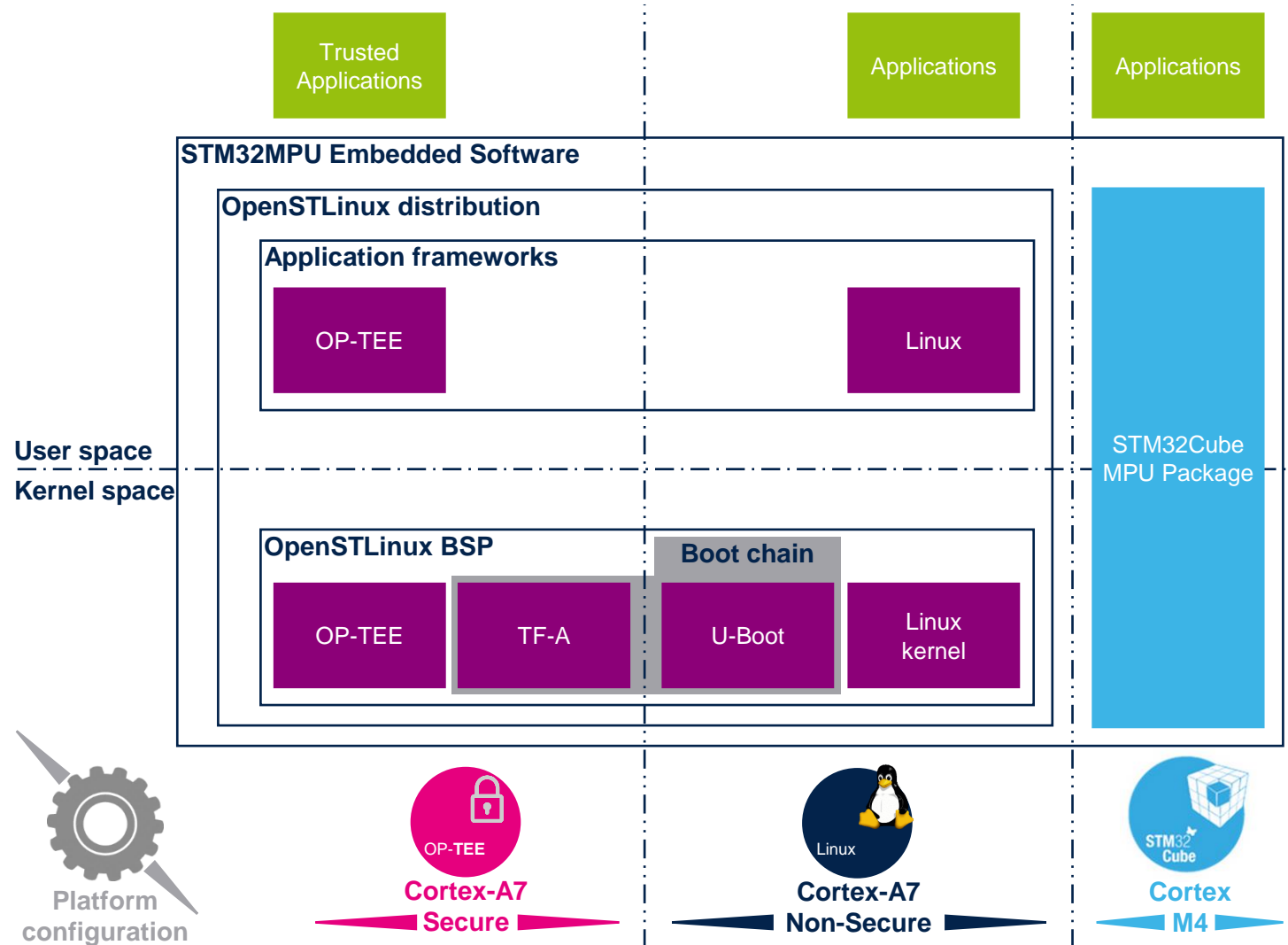
- Notice that each core may not see the same regions at the same address, as already explained on previous slide



- Each customer can of course tune this mapping (regions location and sizes) to fit with his product needs

STM32MPU embedded software

STM32MPU embedded software



3rd Party

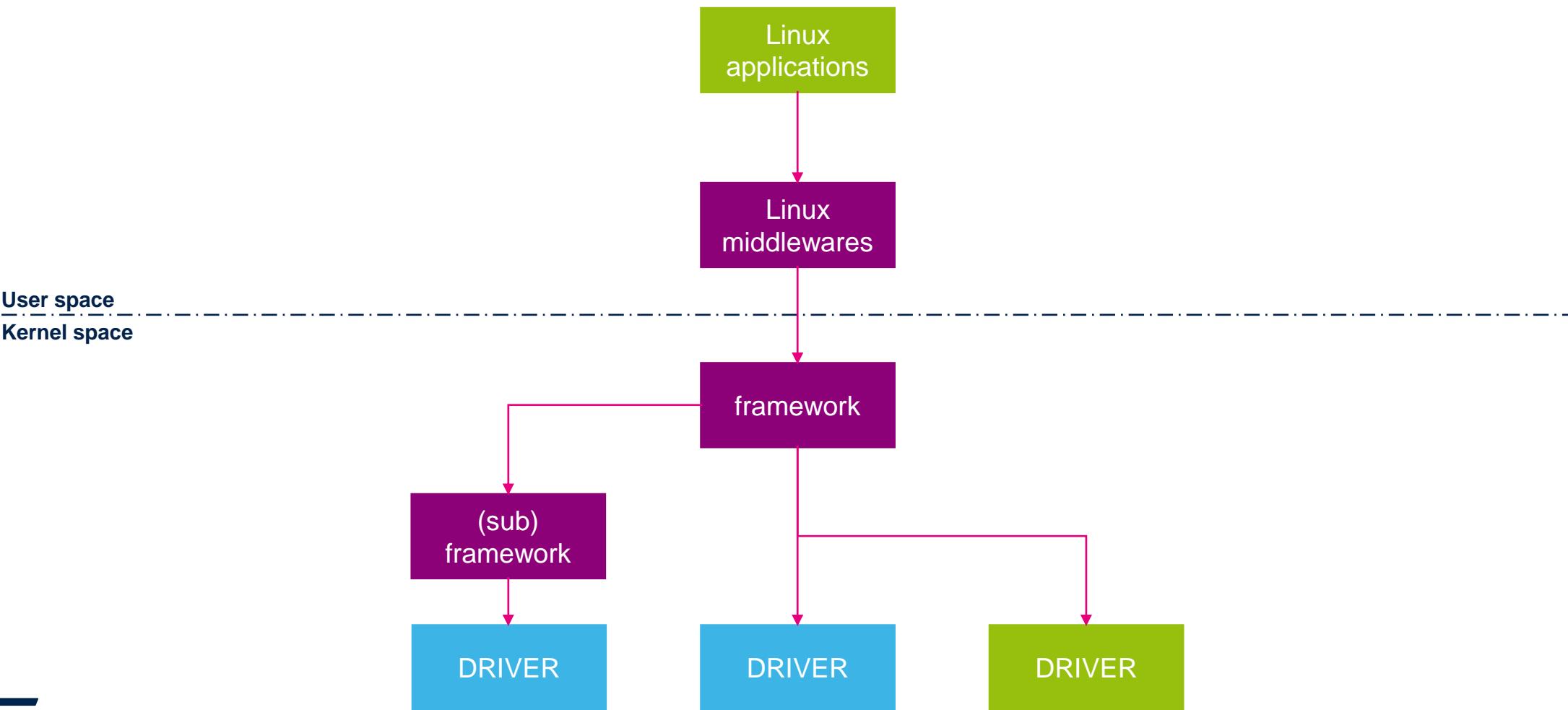
ST

Community

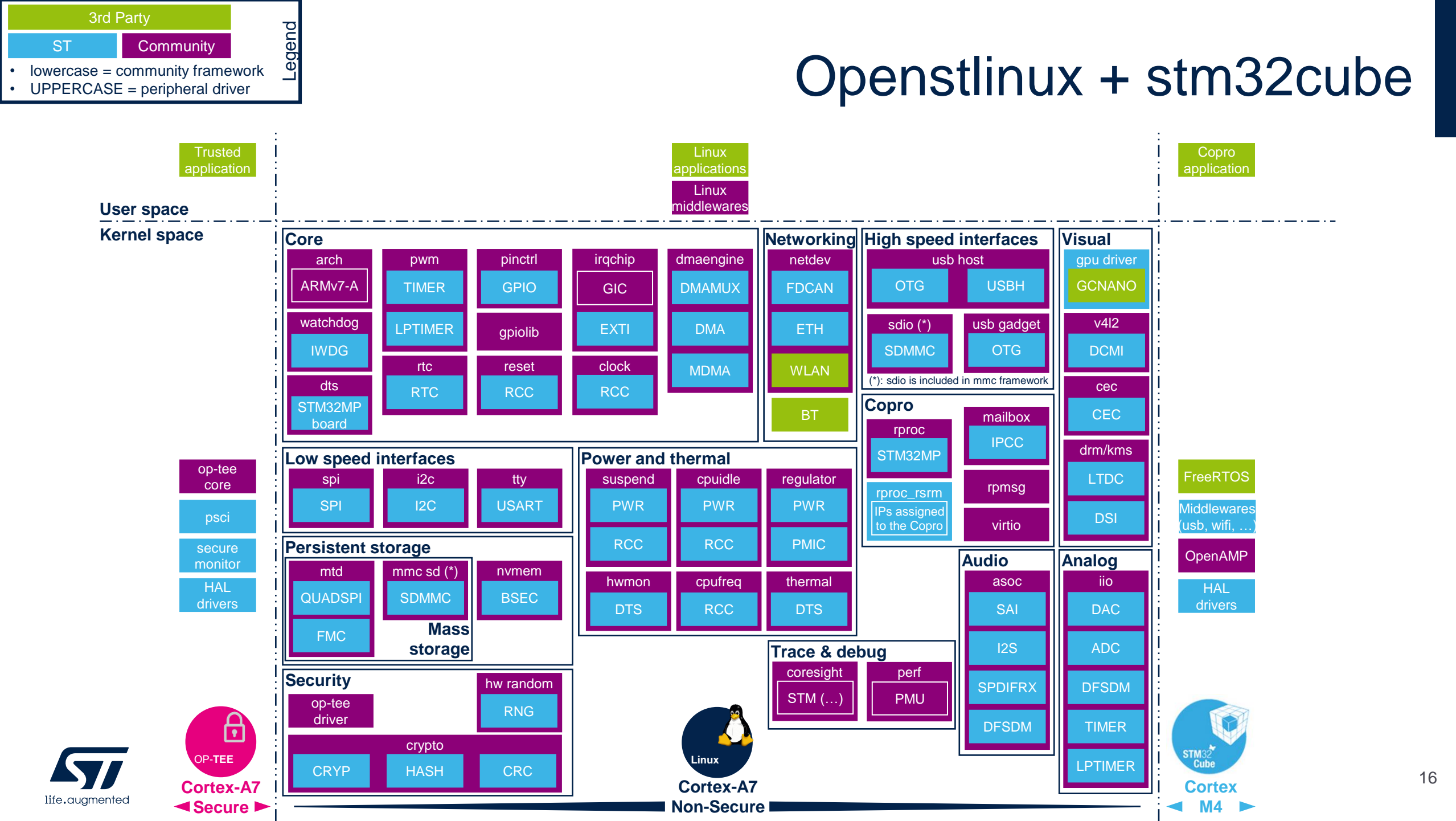
- lowercase = community framework
- UPPERCASE = peripheral driver

Legend

Linux framework & driver

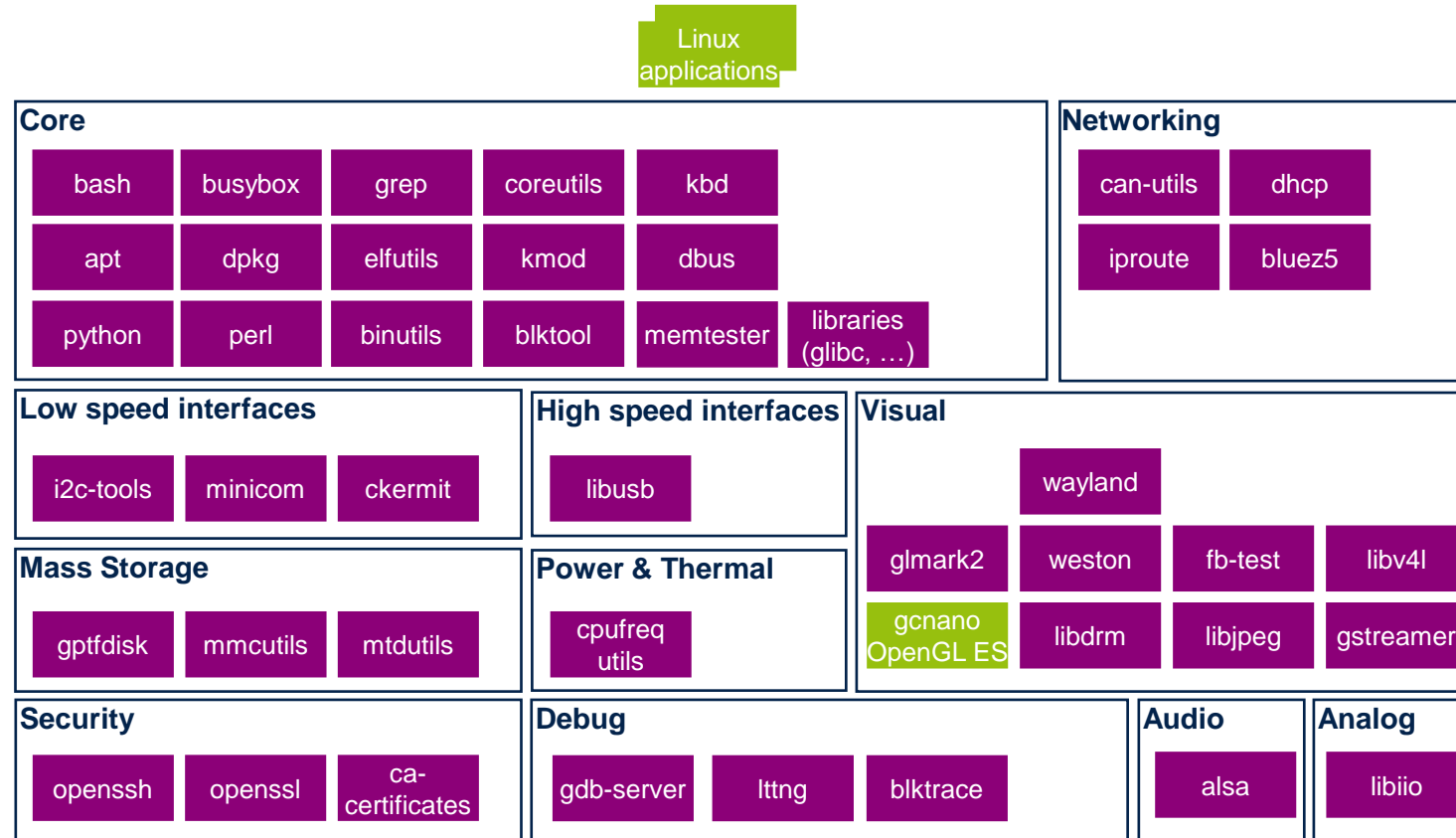


Openstlinux + stm32cube



Open-embedded user space

- The components list shown here is not exhaustive and can be tuned by the customer to fit with applications needs.



User space

Kernel space

