STM32MP1 microprocessor, STM32의 새로운 도약

MMS Korea

Caleb Kang

STM32MP1 introduction

life.augmented

A general purpose MPU

Boosting performances
with Dual Cortex-A7 @ 800MHz

A broader STM32 MPU ecosystem
to reduce development time & cost

Suitable for all Developer Types and Multiple Applications

Developer profile

MCU users new to MPU MCU → MPU

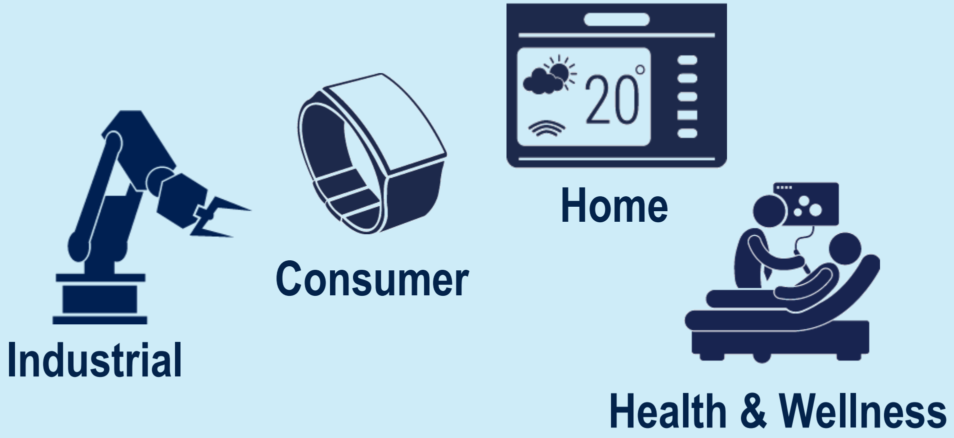
Mixed MCU & MPU users MCU + MPU

Pure MPU users MPU

STM32

STM32 MP1

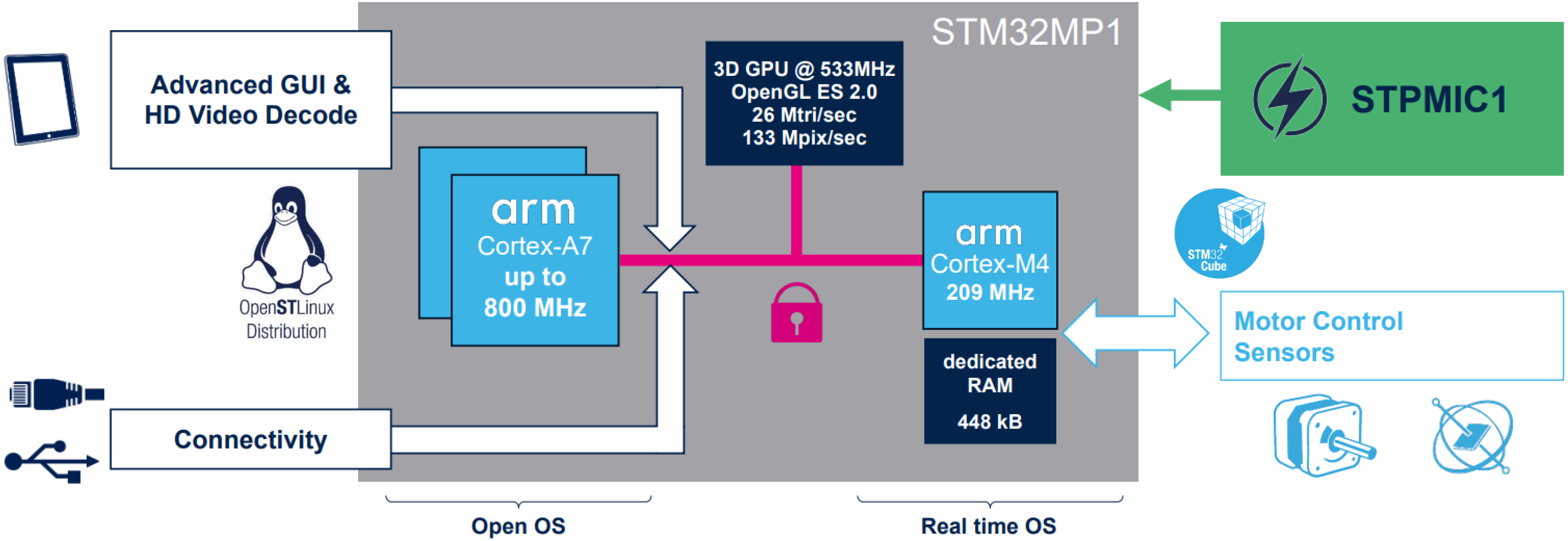
Possible applications



Rich feature set for boosting application possibilities

Graphic and communication
High Performance processing
up to **3040 DMIPS**

Real-time & Low Power applications
260 DMIPS



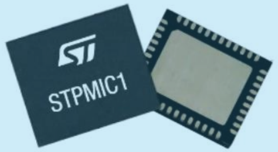
Ready to successful MPU development

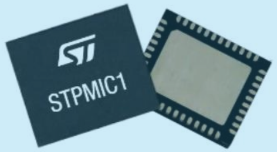
Pin to pin compatibility across all part numbers
Full H/W compatibility with STPMIC1

S/W compatibility
Across the family

Evaluation/Discovery boards
Customer support

Common design properties





@ 650MHz

@ 800MHz

STM32 Cube

STM32 CubeMX

OpenSTLinux Distribution

STM32 CubeProgrammer

+

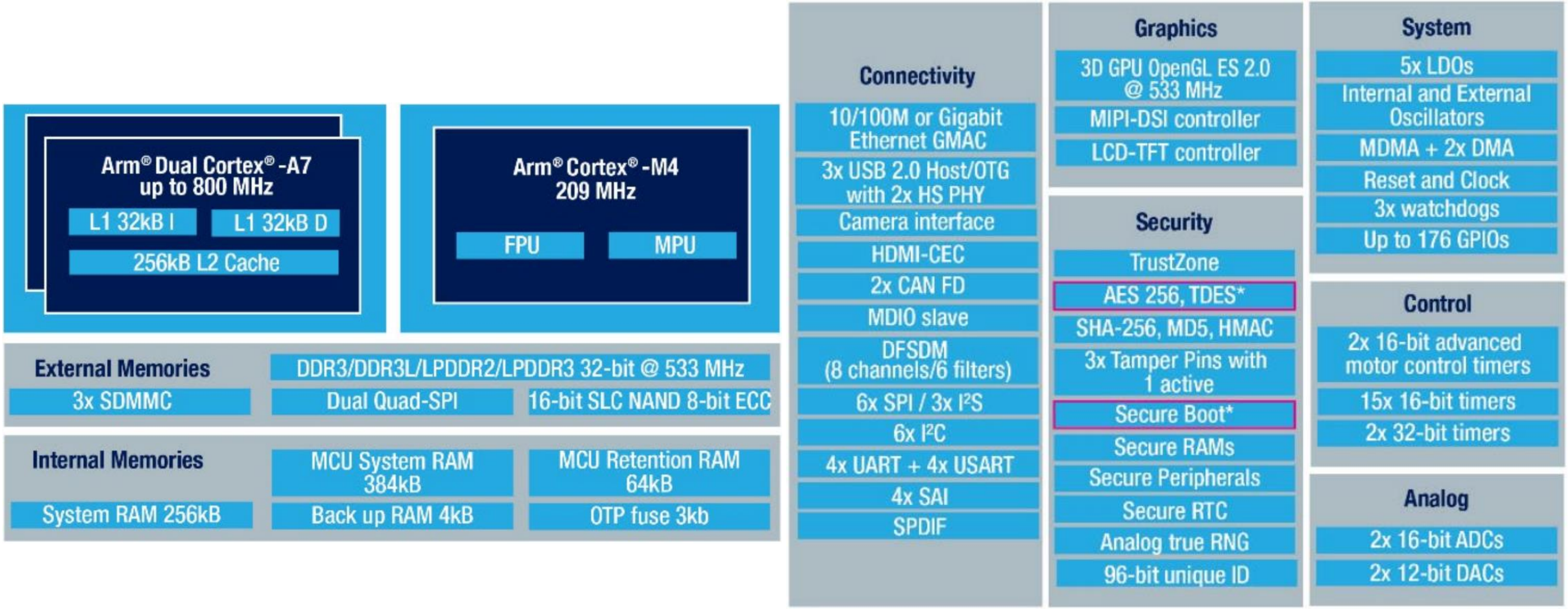
FAE - Worldwide Customer support

community.st.com

STM32MP1 hardware architecture

life.augmented

STM32MP1 microprocessor unit



Arm® Dual Cortex®-A7 up to 800 MHz
- L1 32kB I
- L1 32kB D
- 256kB L2 Cache

Arm® Cortex®-M4 209 MHz
- FPU
- MPU

External Memories
- DDR3/DDR3L/LPDDR2/LPDDR3 32-bit @ 533 MHz
- 3x SDMMC
- Dual Quad-SPI
- 16-bit SLC NAND 8-bit ECC

Internal Memories
- MCU System RAM 384kB
- MCU Retention RAM 64kB
- System RAM 256kB
- Back up RAM 4kB
- OTP fuse 3kb

Connectivity
- 10/100M or Gigabit Ethernet GMAC
- 3x USB 2.0 Host/OTG with 2x HS PHY
- Camera interface
- HDMI-CEC
- 2x CAN FD
- MDIO slave
- DFSDM (8 channels/6 filters)
- 6x SPI / 3x I²S
- 6x I²C
- 4x UART + 4x USART
- 4x SAI
- SPDIF

Graphics
- 3D GPU OpenGL ES 2.0 @ 533 MHz
- MIPI-DSI controller
- LCD-TFT controller

Security
- TrustZone
- AES 256, TDES*
- SHA-256, MD5, HMAC
- 3x Tamper Pins with 1 active
- Secure Boot*
- Secure RAMs
- Secure Peripherals
- Secure RTC
- Analog true RNG
- 96-bit unique ID

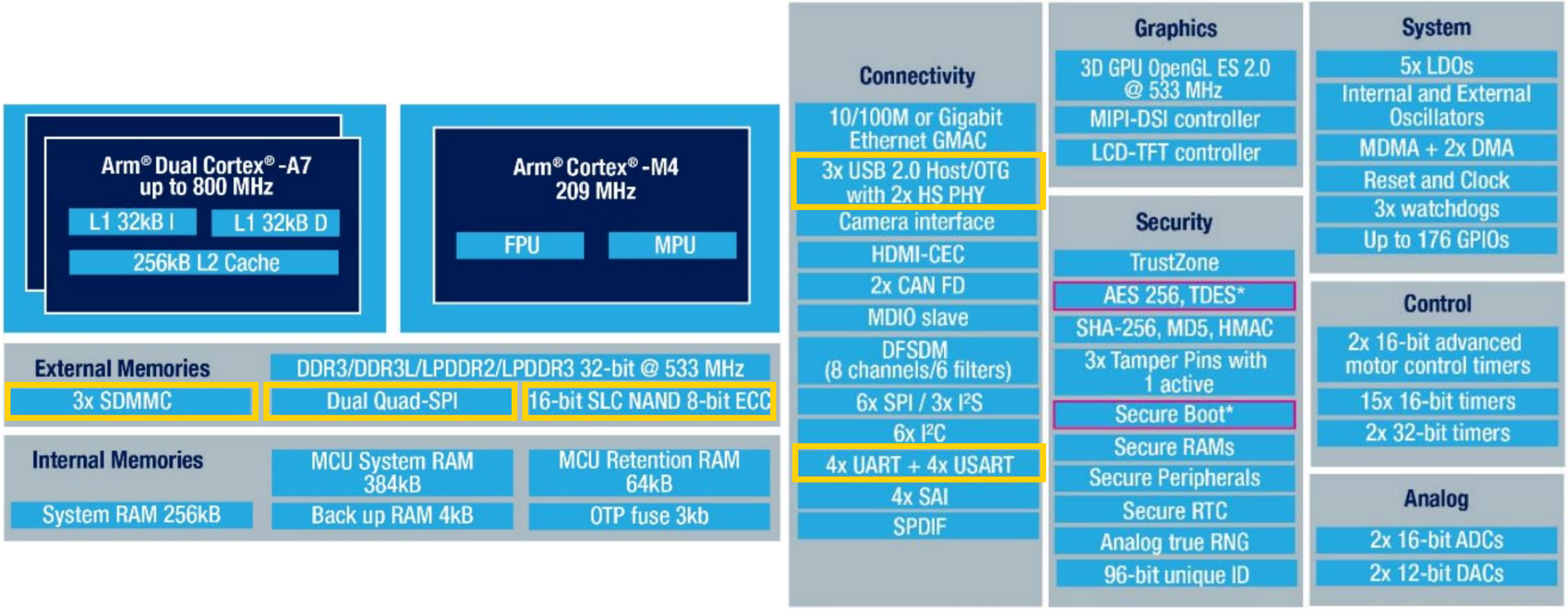
System
- 5x LDOs
- Internal and External Oscillators
- MDMA + 2x DMA
- Reset and Clock
- 3x watchdogs
- Up to 176 GPIOs

Control
- 2x 16-bit advanced motor control timers
- 15x 16-bit timers
- 2x 32-bit timers

Analog
- 2x 16-bit ADCs
- 2x 12-bit DACs

□ *available for STM32MP157C and STM32MP157F only

STM32MP1 peripherals



Arm® Dual Cortex® -A7 up to 800 MHz

L1 32kB I | L1 32kB D

256kB L2 Cache

Arm® Cortex® -M4 209 MHz

FPU | MPU

External Memories

DDR3/DDR3L/LPDDR2/LPDDR3 32-bit @ 533 MHz

3x SDMMC | Dual Quad-SPI | 16-bit SLC NAND 8-bit ECC

Internal Memories

MCU System RAM 384kB | MCU Retention RAM 64kB

System RAM 256kB | Back up RAM 4kB | OTP fuse 3kb

Connectivity

- 10/100M or Gigabit Ethernet GMAC
- 3x USB 2.0 Host/OTG with 2x HS PHY
- Camera interface
- HDMI-CEC
- 2x CAN FD
- MDIO slave
- DFSDM (8 channels/6 filters)
- 6x SPI / 3x I²S
- 6x I²C
- 4x UART + 4x USART
- 4x SAI
- SPDIF

Graphics

- 3D GPU OpenGL ES 2.0 @ 533 MHz
- MIPI-DSI controller
- LCD-TFT controller

Security

- TrustZone
- AES 256, TDES*
- SHA-256, MD5, HMAC
- 3x Tamper Pins with 1 active
- Secure Boot*
- Secure RAMs
- Secure Peripherals
- Secure RTC
- Analog true RNG
- 96-bit unique ID

System

- 5x LDOs
- Internal and External Oscillators
- MDMA + 2x DMA
- Reset and Clock
- 3x watchdogs
- Up to 176 GPIOs

Control

- 2x 16-bit advanced motor control timers
- 15x 16-bit timers
- 2x 32-bit timers

Analog

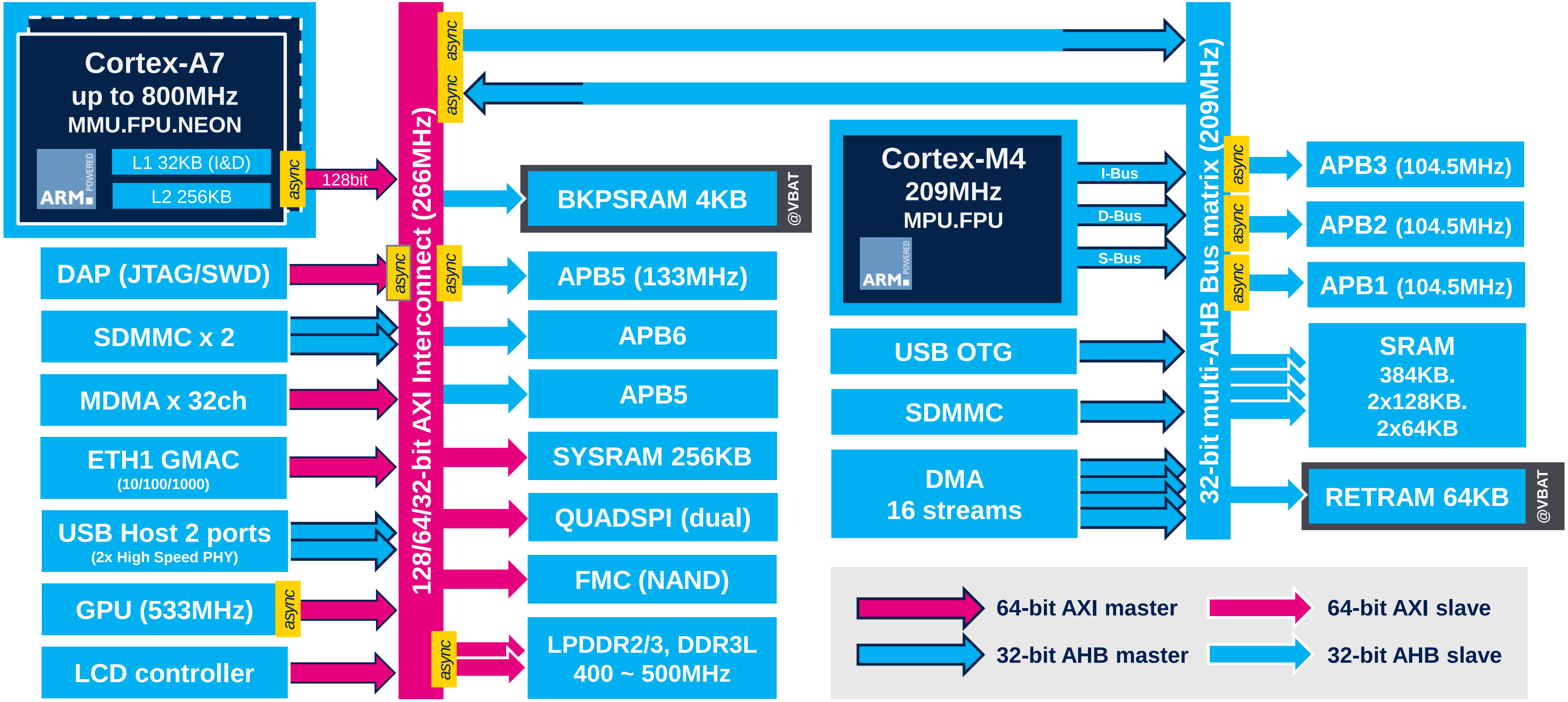
- 2x 16-bit ADCs
- 2x 12-bit DACs

□ *available for STM32MP157C and STM32MP157F only

□ available to boot source

life.augmented

Bus architecture overview

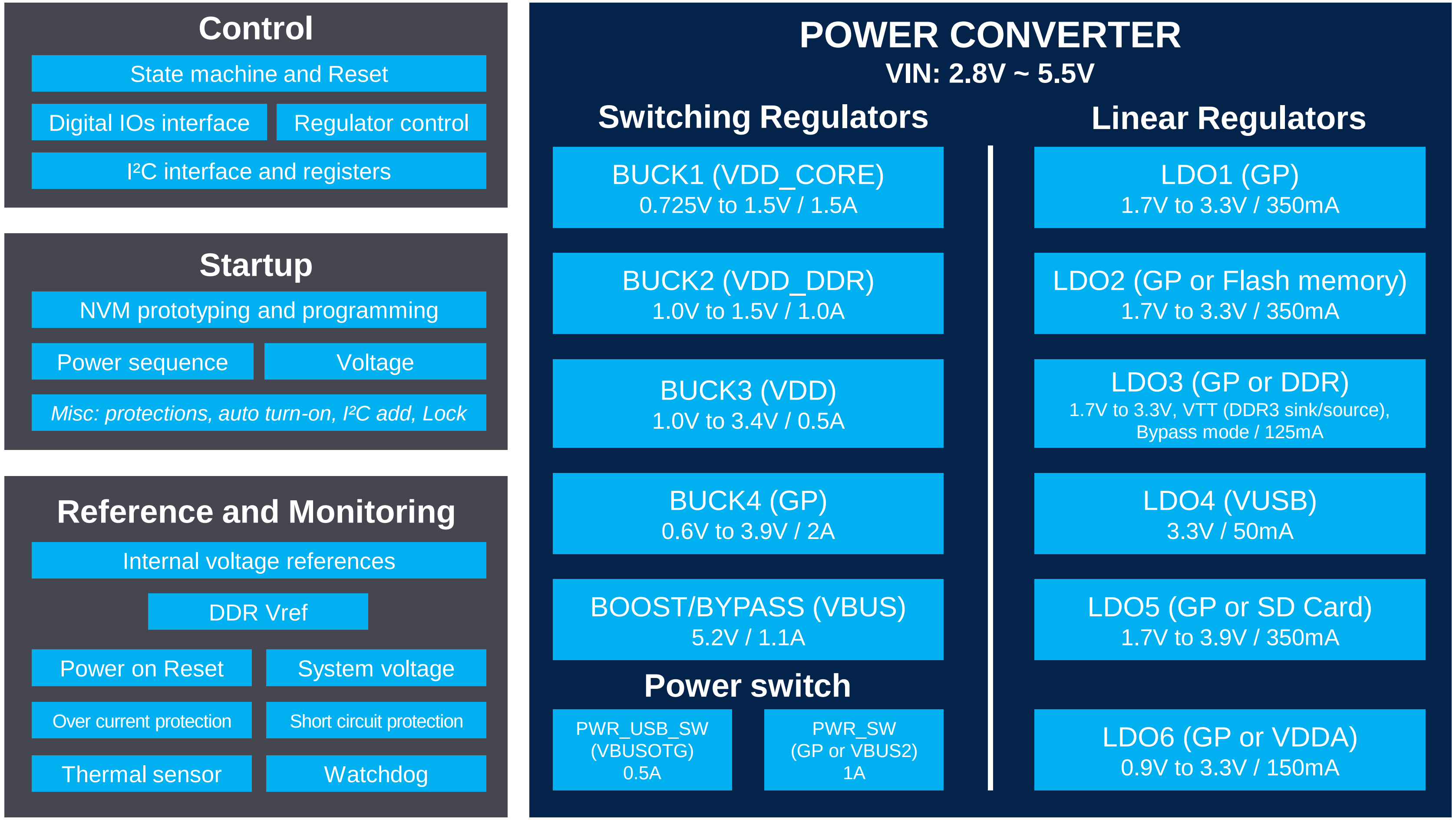


life.augmented

Power supplies

Name	Typical or Range	Description
VDD	**1.7V ~ 3.6V**	Power supply input for I/Os
VDD_PLL / ANA / DSI		Power supply input for PLLs and system analog like RCC, PWR and DSI. To be connected to VDD
VDD_CORE	**1.2V**	Power supply input for Digital Core domain
VDDA	**1.7V – 3.6V**	Analog Power supply input for ADCs, DACs and voltage reference buffers
VDDQ_DDR	**1.2V / 1.35V / 1.5V**	Power supply input for DDR Physical Interface (PHY) and IOs
VDD3V3_USBHS/FS	**3.3V**	Power supply input for USB Physical Interface (PHY) and IOs
Internally generated Power Supplies		
VDDA1V8_REG	**1.8V**	Analog Power Supply input or output, used internally for USB Physical Interface (PHY)
VDDA1V8_DSI		Analog Power supply input for DSI Physical Interface (PHY), to be connected to VDDA1V8_REG
VDDA1V2_DSI_REG	**1.2V**	Analog Power supply output, used internally for DSI PLL
VDDA1V2_DSI_PHY		Analog Power supply input for DSI Physical Interface (PHY), to be connected to VDDA1V2_DSI_REG
VDDA1V1_REG	**1.1V**	Analog Power supply output for USB Physical Interface (PHY)

STPMIC1 block diagram



Control

- State machine and Reset
- Digital IOs interface
- Regulator control
- I²C interface and registers

Startup

- NVM prototyping and programming
- Power sequence
- Voltage
- *Misc: protections, auto turn-on, I²C add, Lock*

Reference and Monitoring

- Internal voltage references
- DDR Vref
- Power on Reset
- System voltage
- Over current protection
- Short circuit protection
- Thermal sensor
- Watchdog

POWER CONVERTER

VIN: 2.8V ~ 5.5V

Switching Regulators

- BUCK1 (VDD_CORE) 0.725V to 1.5V / 1.5A
- BUCK2 (VDD_DDR) 1.0V to 1.5V / 1.0A
- BUCK3 (VDD) 1.0V to 3.4V / 0.5A
- BUCK4 (GP) 0.6V to 3.9V / 2A
- BOOST/BYPASS (VBUS) 5.2V / 1.1A

Power switch

- PWR_USB_SW (VBUSOTG) 0.5A
- PWR_SW (GP or VBUS2) 1A

Linear Regulators

- LDO1 (GP) 1.7V to 3.3V / 350mA
- LDO2 (GP or Flash memory) 1.7V to 3.3V / 350mA
- LDO3 (GP or DDR) 1.7V to 3.3V, VTT (DDR3 sink/source), Bypass mode / 125mA
- LDO4 (VUSB) 3.3V / 50mA
- LDO5 (GP or SD Card) 1.7V to 3.9V / 350mA
- LDO6 (GP or VDDA) 0.9V to 3.3V / 150mA

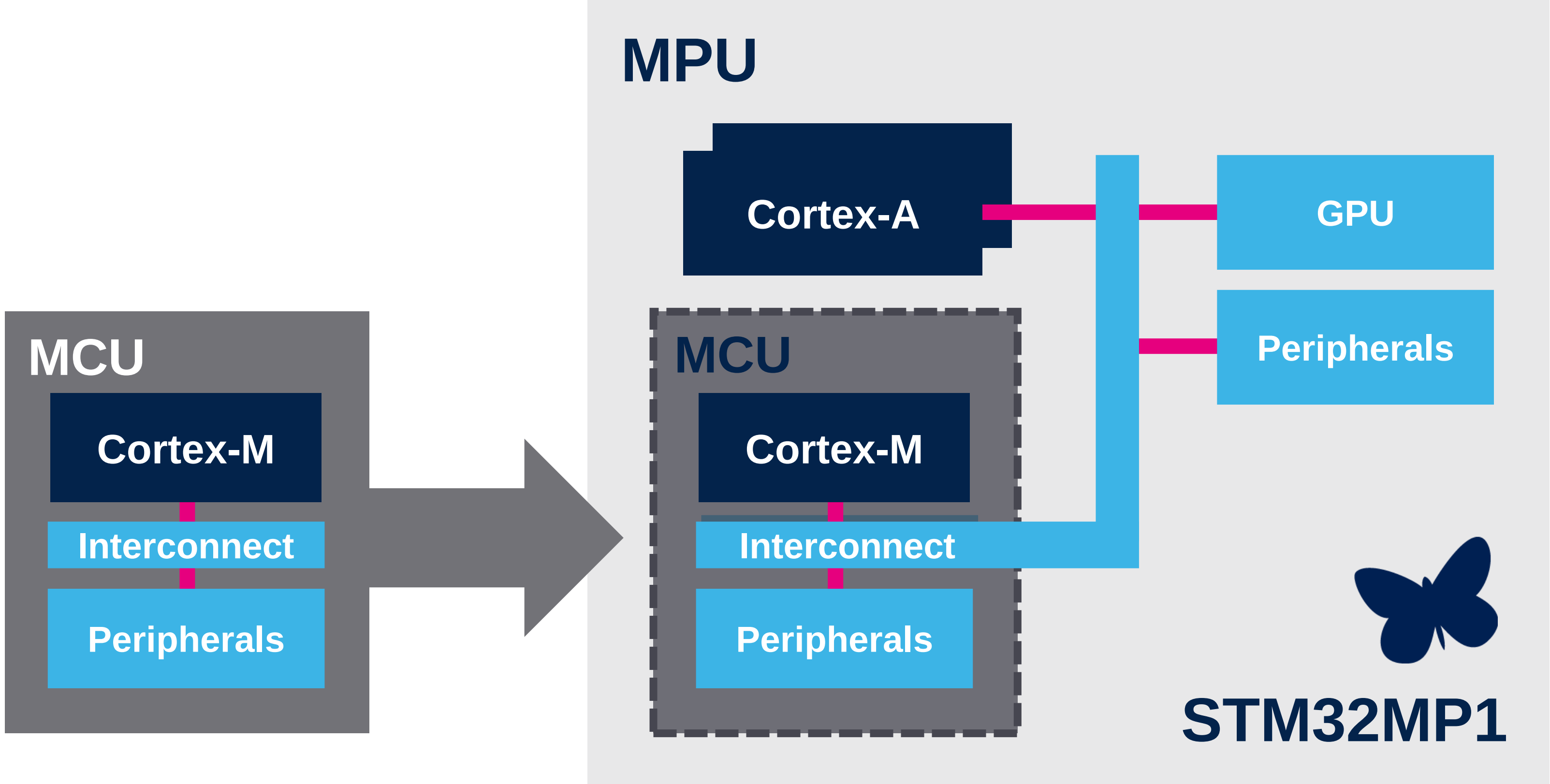
Package: WFQFN 5x6x0.8 44L pitch 0.4mm

OTP fuse

- **OTP Fuses are One Time Programming memory**
  - Initial bits are ‘0’ and are irreversibly programmed to ‘1’
  - Incremental programing of bits in a 32-bit word is possible
- **Handled thru BSEC (Boot and Security) controller IP**
  - Programming, reading, status and locking handled by BSEC
  - Lock mechanism to avoid read and/or program (32-bits granularity)
- **OTP Content**
  - Product configuration and Trimming values set by ST during production
  - Secrets and unique identification numbers set by ST during production
  - Device configuration set by OEM (e.g. MAC address, boot source, security mode, etc...)
  - Secrets set by OEM (e.g. for secure boot)
  - Up to 1184 bits available for other OEM purposes

OpenSTLinux software architecture

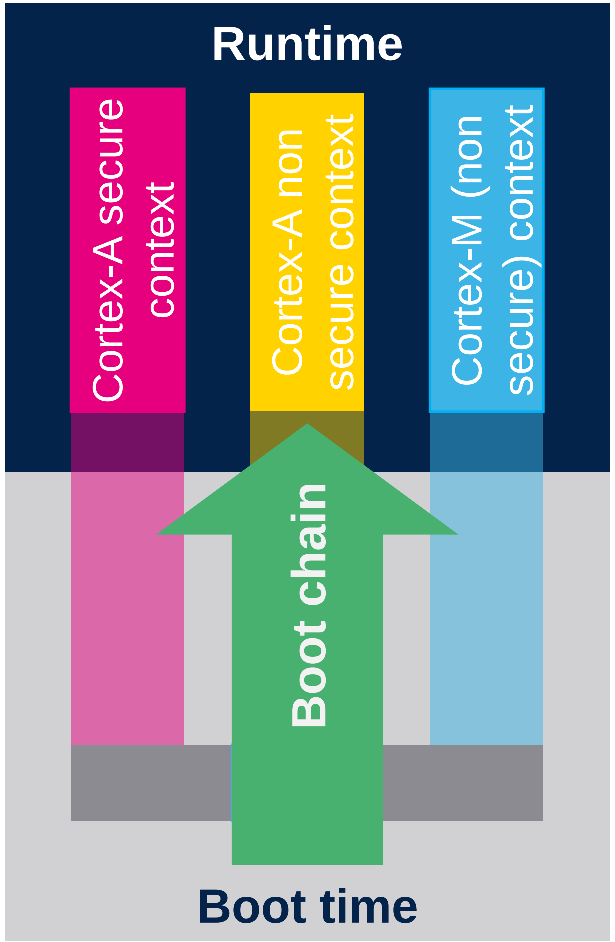
From MCU to MPU



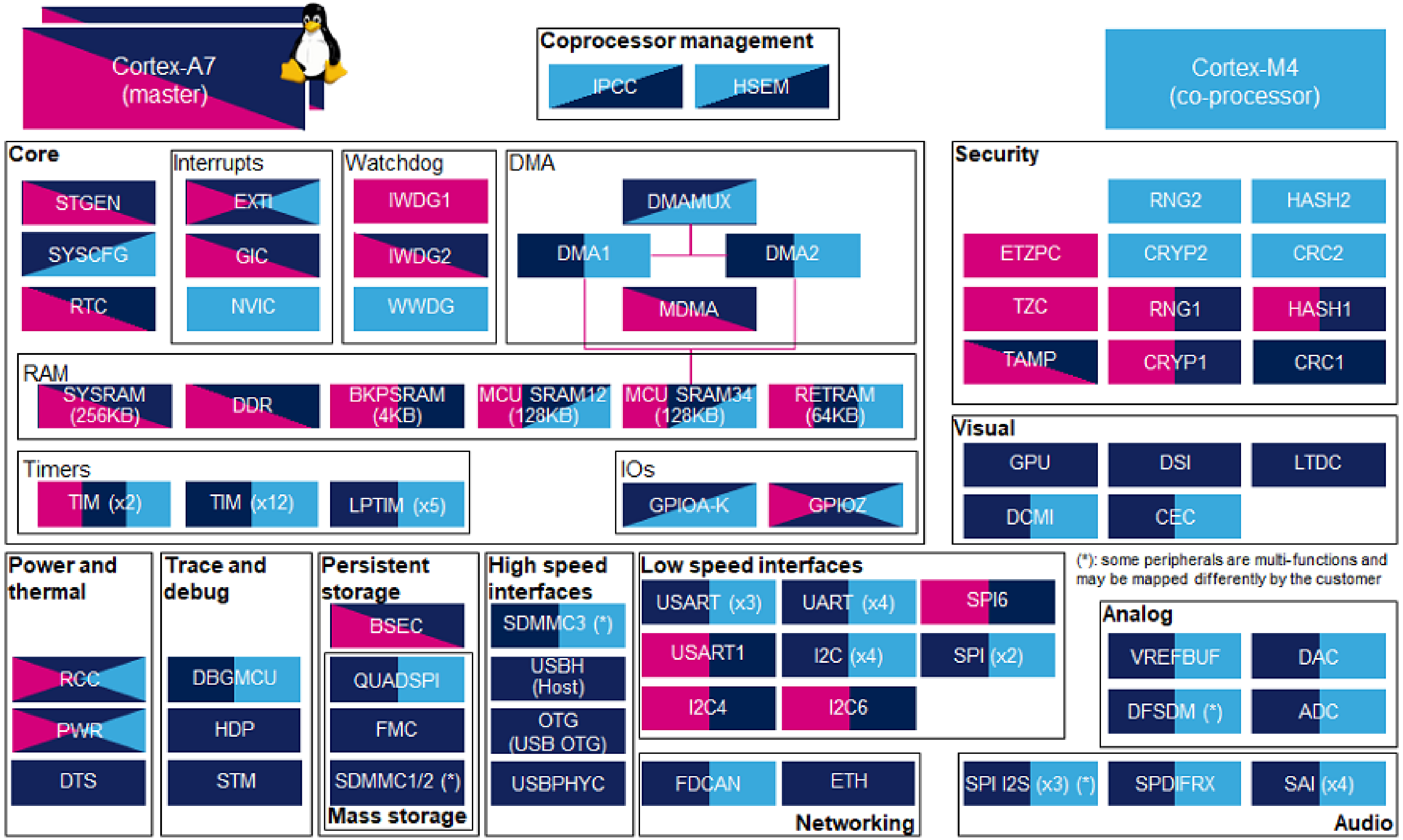
life.augmented

Multiple core architecture concept

- 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



Cortex-A7 (master)

Coprocessor management: IPCC, HSEM

Cortex-M4 (co-processor)

Core: STGEN, SYSCFG, RTC

Interrupts: EXTI, GIC, NVIC

Watchdog: IWDG1, IWDG2, WWDG

DMA: DMAMUX, DMA1, DMA2, MDMA

RAM: SYSRAM (256KB), DDR, BKPSRAM (4KB), MCU SRAM12 (128KB), MCU SRAM34 (128KB), RETRAM (64KB)

Timers: TIM (x2), TIM (x12), LPTIM (x5)

IOs: GPIOA-K, GPIOZ

Security: RNG2, HASH2, ETZPC, CRYP2, CRC2, TZC, RNG1, HASH1, TAMP, CRYP1, CRC1

Visual: GPU, DSI, LTDC, DCMI, CEC

Power and thermal: RCC, PWR, DTS

Trace and debug: DBGMCU, HDP, STM

Persistent storage: BSEC

Mass storage: QUADSPI, FMC, SDMMC1/2 (*)

High speed interfaces: SDMMC3 (*), USBH (Host), OTG (USB OTG), USBPHYC

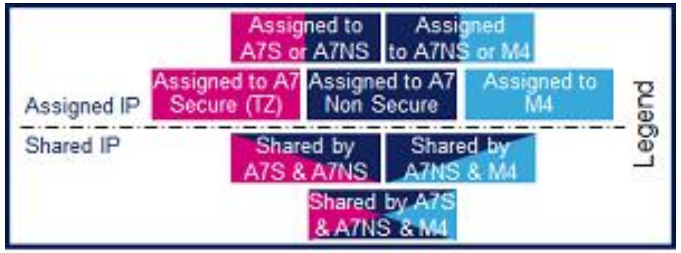
Low speed interfaces: USART (x3), UART (x4), SPI6, USART1, I2C (x4), SPI (x2), I2C4, I2C6

(*): some peripherals are multi-functions and may be mapped differently by the customer

Analog: VREFBUF, DAC, DFSDM (*), ADC

Networking: FDCAN, ETH

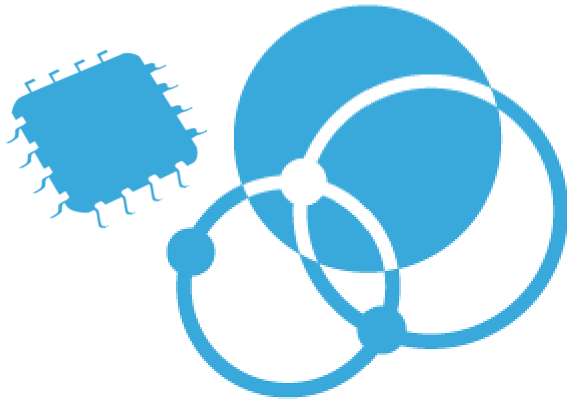
Audio: SPI I2S (x3) (*), SPDIFRX, SAI (x4)



Legend

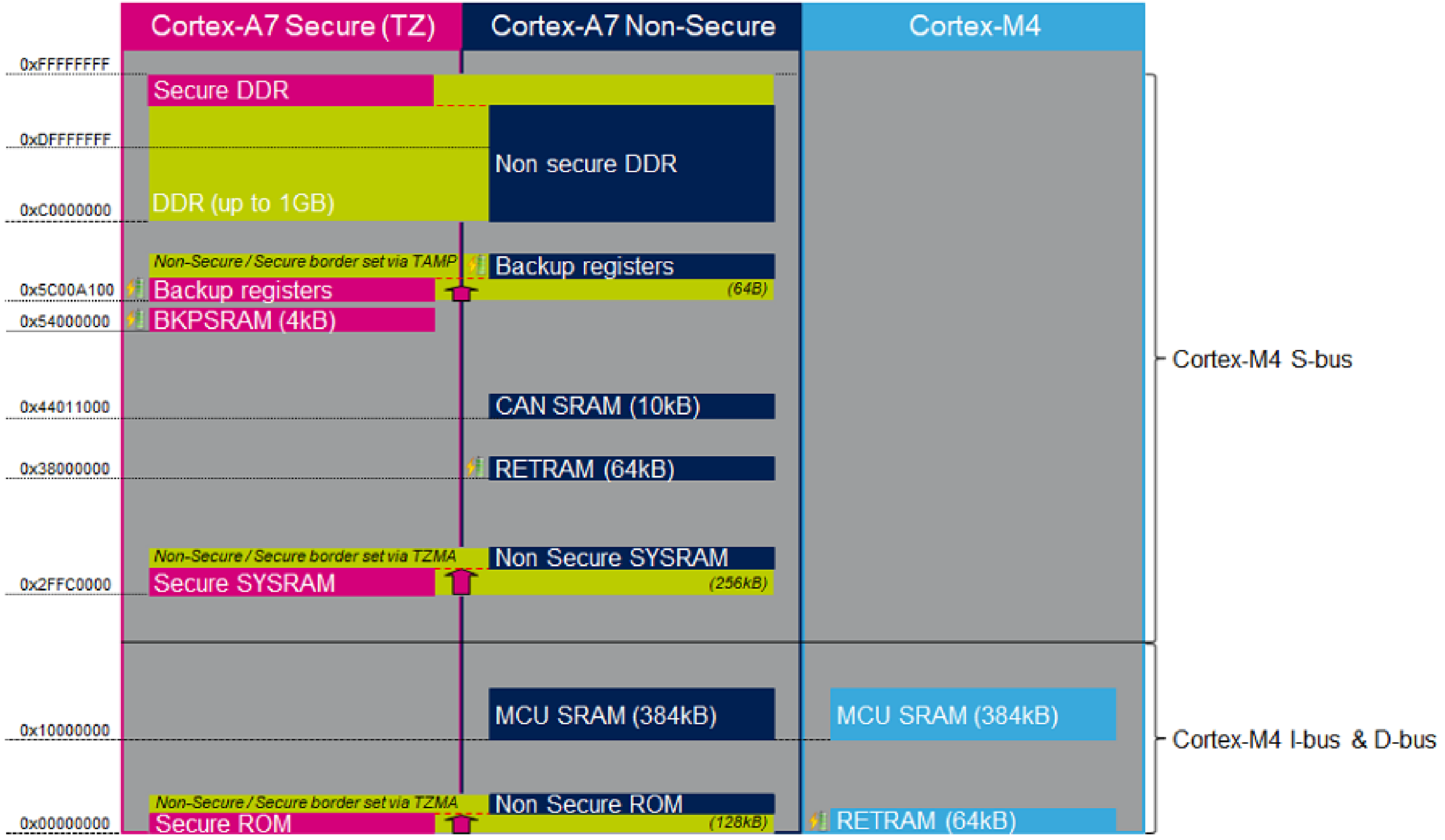
Assigned IP: Assigned to A7S or A7NS, Assigned to A7NS or M4, Assigned to A7 Secure (TZ), Assigned to A7 Non Secure, Assigned to M4

Shared IP: Shared by A7S & A7NS, Shared by A7NS & M4, Shared by A7S & A7NS & M4

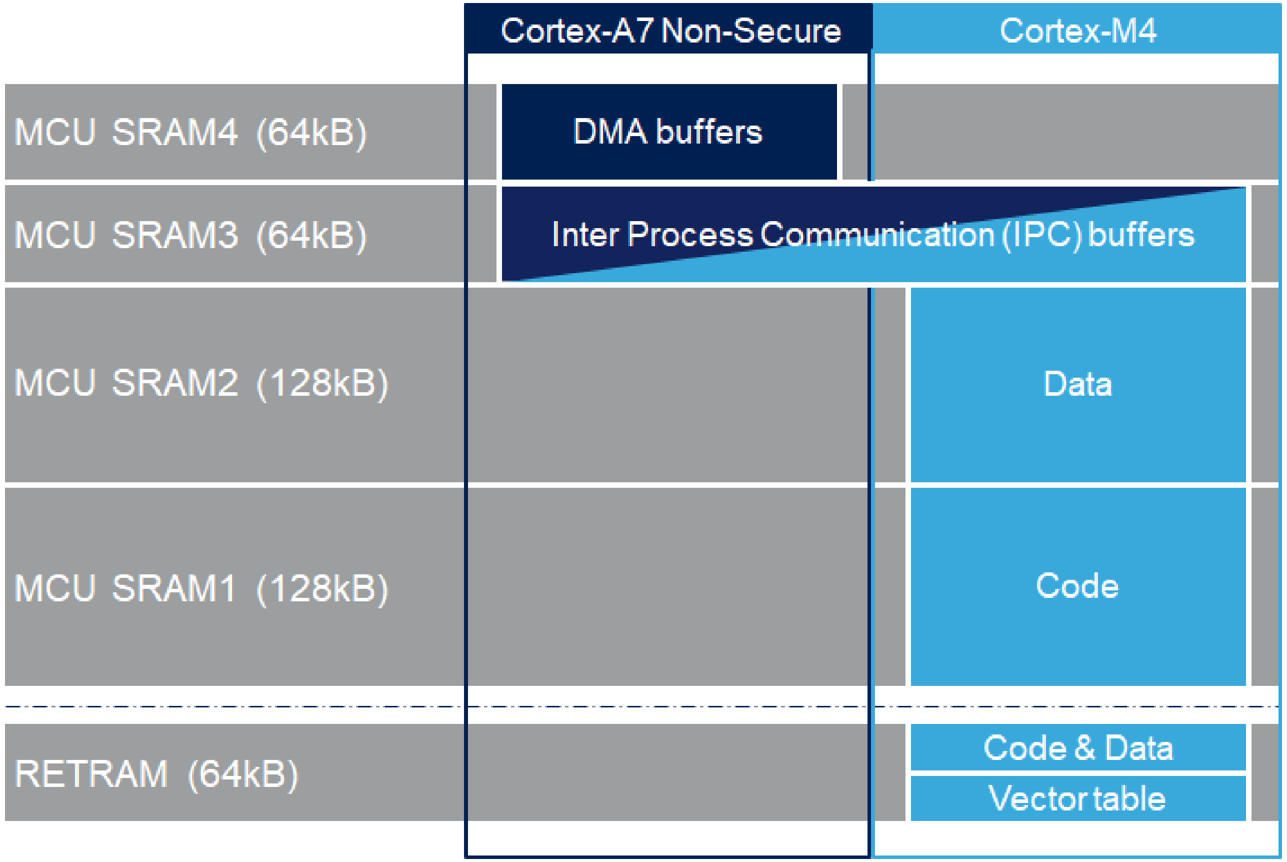


Software memory mapping

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

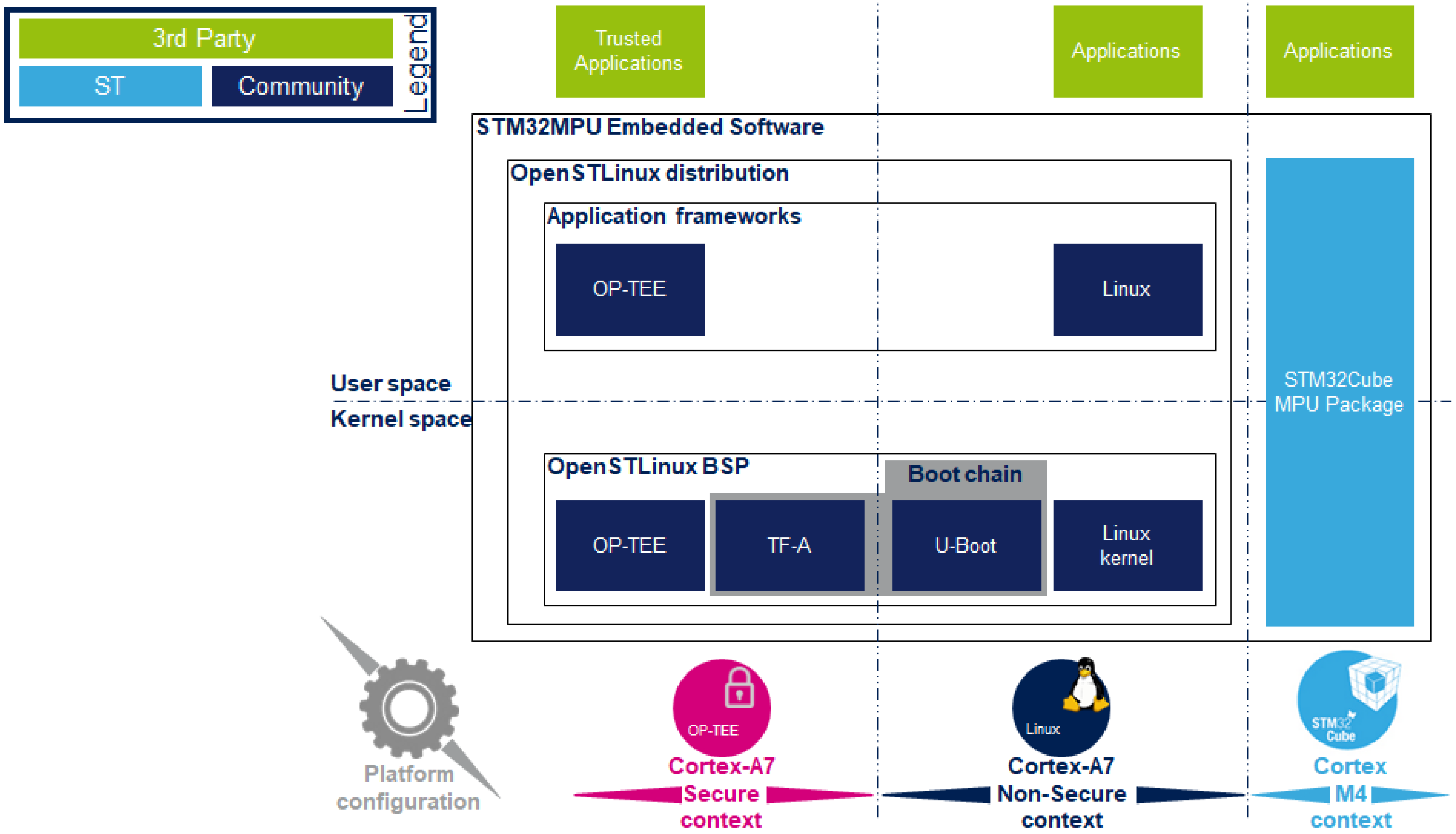


Shared RAM memory mapping



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

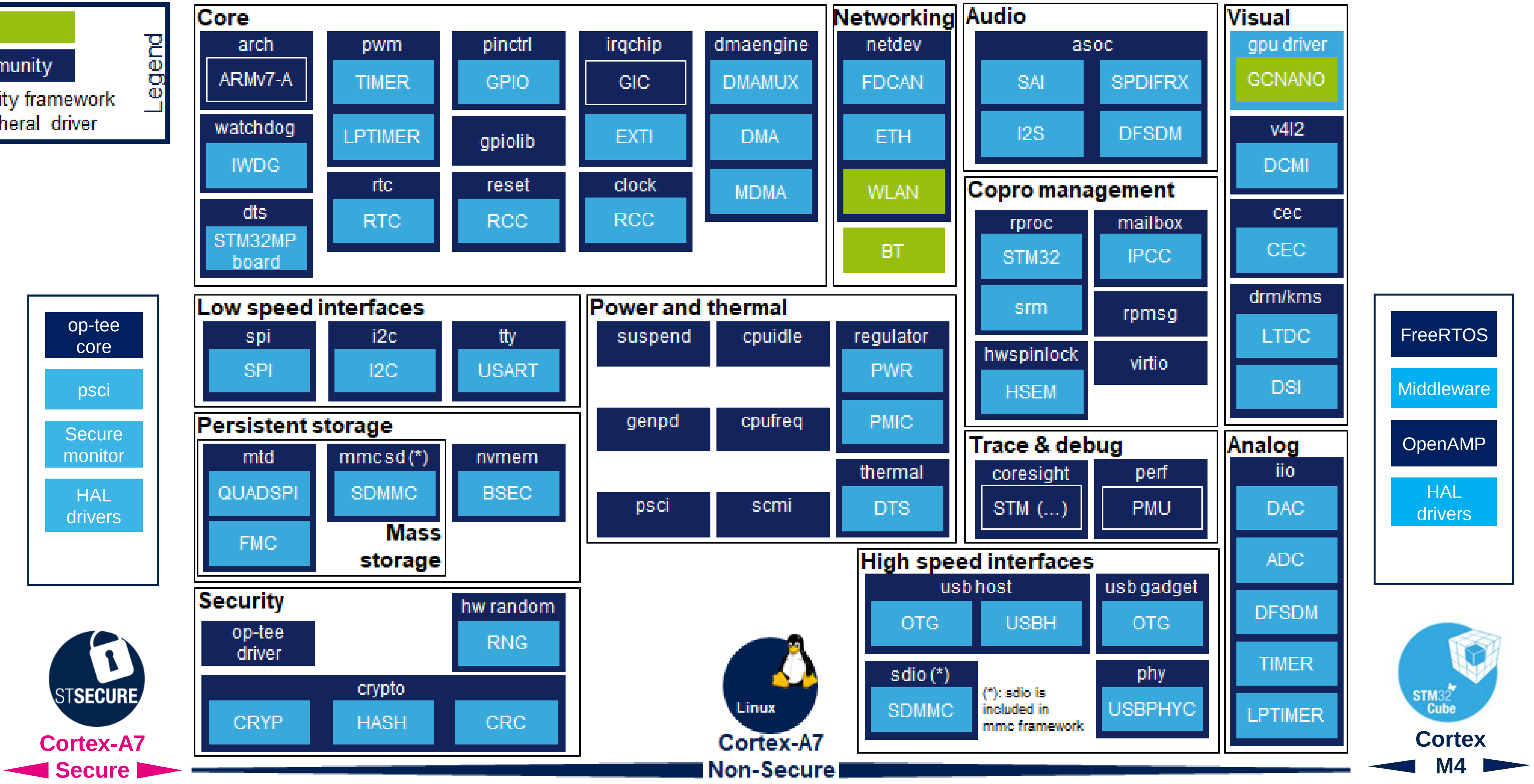
STM32MPU embedded software



OpenSTLinux + STM32Cube

Legend
- 3rd Party
- ST
- Community

- lowercase = community framework
- UPPERCASE = peripheral driver



Core
- arch: ARMv7-A
- watchdog: IWDG
- dts: STM32MP board
- pwm: TIMER, LPTIMER
- rtc: RTC
- pinctrl: GPIO
- gpiolib
- reset: RCC
- irqchip: GIC, EXTI
- clock: RCC
- dmaengine: DMAMUX, DMA, MDMA

Networking
- netdev: FDCAN, ETH, WLAN
- BT

Audio
- asoc: SAI, SPDIFRX, I2S, DFSDM

Visual
- gpu driver: GCNANO
- v4l2: DCMI
- cec: CEC
- drm/kms: LTDC, DSI

Low speed interfaces
- spi: SPI
- i2c: I2C
- tty: USART

Power and thermal
- suspend
- cpuidle
- regulator: PWR, PMIC
- genpd
- cpufreq
- thermal: DTS
- psci
- scmi

Copro management
- rproc: STM32, srm
- mailbox: IPCC
- rpmsg
- hwspinlock: HSEM
- virtio

Persistent storage
- mtd: QUADSPI, FMC
- mmc sd (*): SDMMC
- Mass storage
- nvmem: BSEC

Trace & debug
- coresight: STM (…)
- perf: PMU

Analog
- iio: DAC, ADC, DFSDM, TIMER, LPTIMER

Security
- op-tee driver
- hw random: RNG
- crypto: CRYP, HASH, CRC

High speed interfaces
- usb host: OTG, USBH
- usb gadget: OTG
- sdio (*): SDMMC
- phy: USBPHYC

(*): sdio is included in mmc framework

Cortex-A7 Secure
- op-tee core
- psci
- Secure monitor
- HAL drivers

Cortex-A7 Non-Secure

Cortex M4
- FreeRTOS
- Middleware
- OpenAMP
- HAL drivers

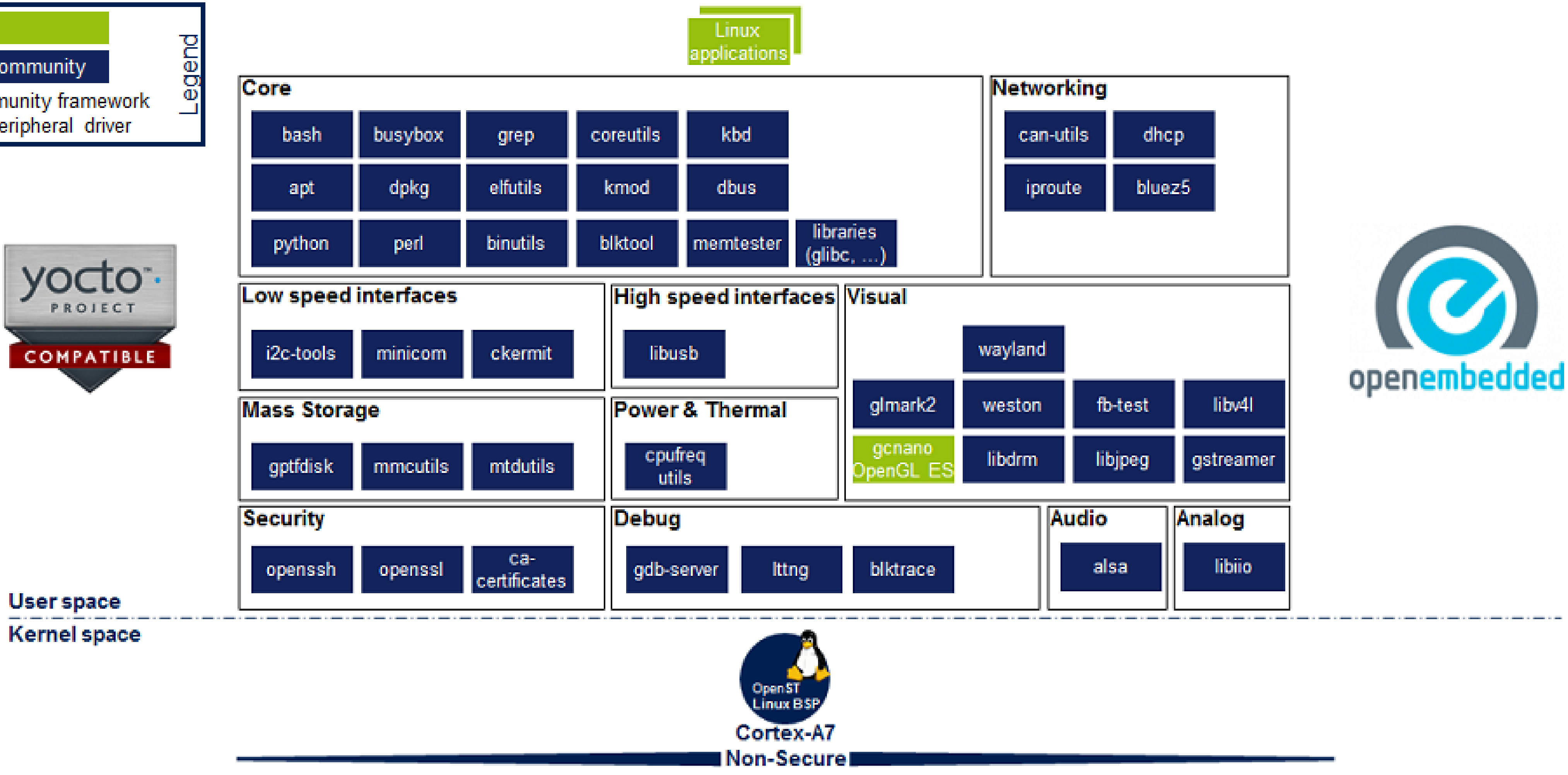
Open-embedded user space

Legend

3rd Party 🟩

ST | Community

- lowercase = community framework
- UPPERCASE = peripheral driver



Linux applications

Core

bash, busybox, grep, coreutils, kbd

apt, dpkg, elfutils, kmod, dbus

python, perl, binutils, blktool, memtester, libraries (glibc, …)

Networking

can-utils, dhcp

iproute, bluez5

Low speed interfaces

i2c-tools, minicom, ckermit

High speed interfaces

libusb

Visual

wayland

glmark2, weston, fb-test, libv4l

gcnano OpenGL ES, libdrm, libjpeg, gstreamer

Mass Storage

gptfdisk, mmcutils, mtdutils

Power & Thermal

cpufreq utils

Security

openssh, openssl, ca-certificates

Debug

gdb-server, lttng, blktrace

Audio

alsa

Analog

libiio

User space

Kernel space

OpenST Linux BSP

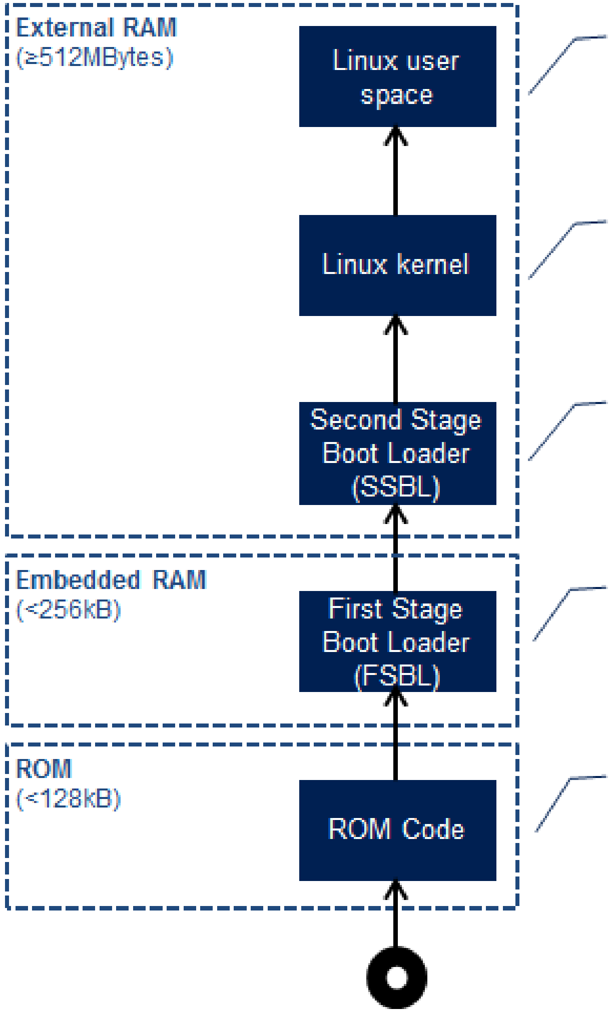
Cortex-A7 Non-Secure

life.augmented

STM32MP1 platform boot

life.augmented

Standard Linux boot chain



- User space services and applications launch

- Linux kernel initialisation (platform device drivers, etc.)
- Root file system (rootfs) mounting
- User space init process launch (/sbin/init)

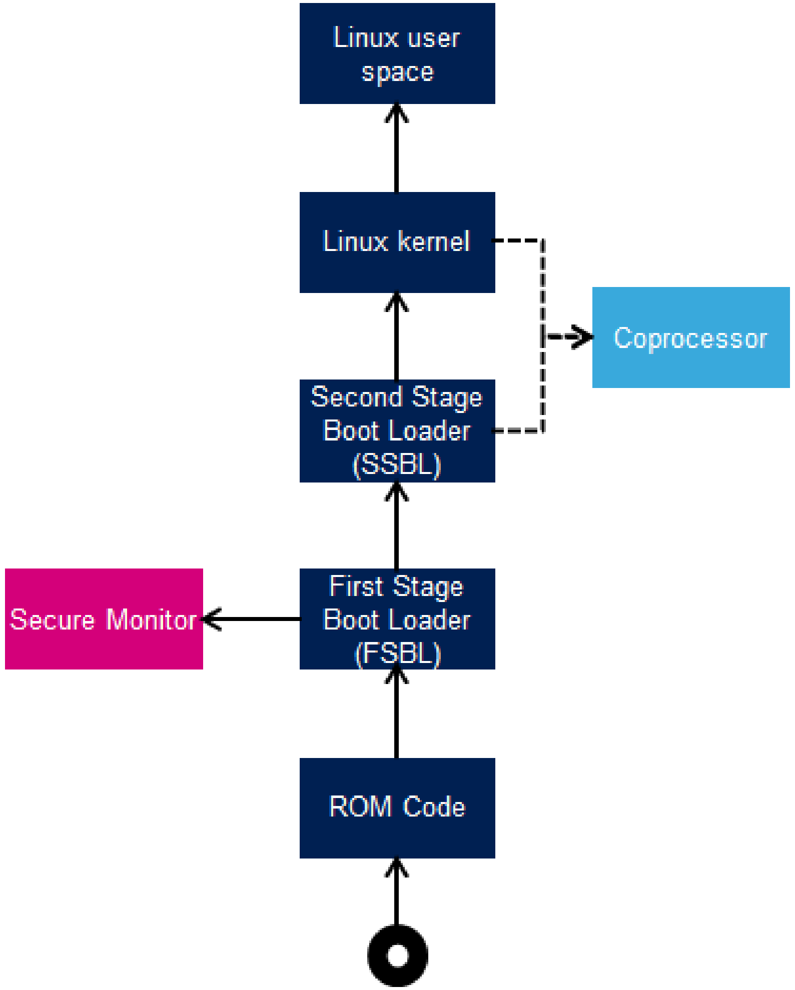
- Boot file system (bootfs) loading from mass storage or Ethernet (TFTP)
- User feedback with boot loader splash screen
- Linux kernel (uImage) launch with its device tree blob (*.dtb)

- Complete clock tree initialization
- External RAM (DDR, LpDDR) controller initialization
- SSBL loading from the boot device (mass storage or serial link)
- SSBL launch

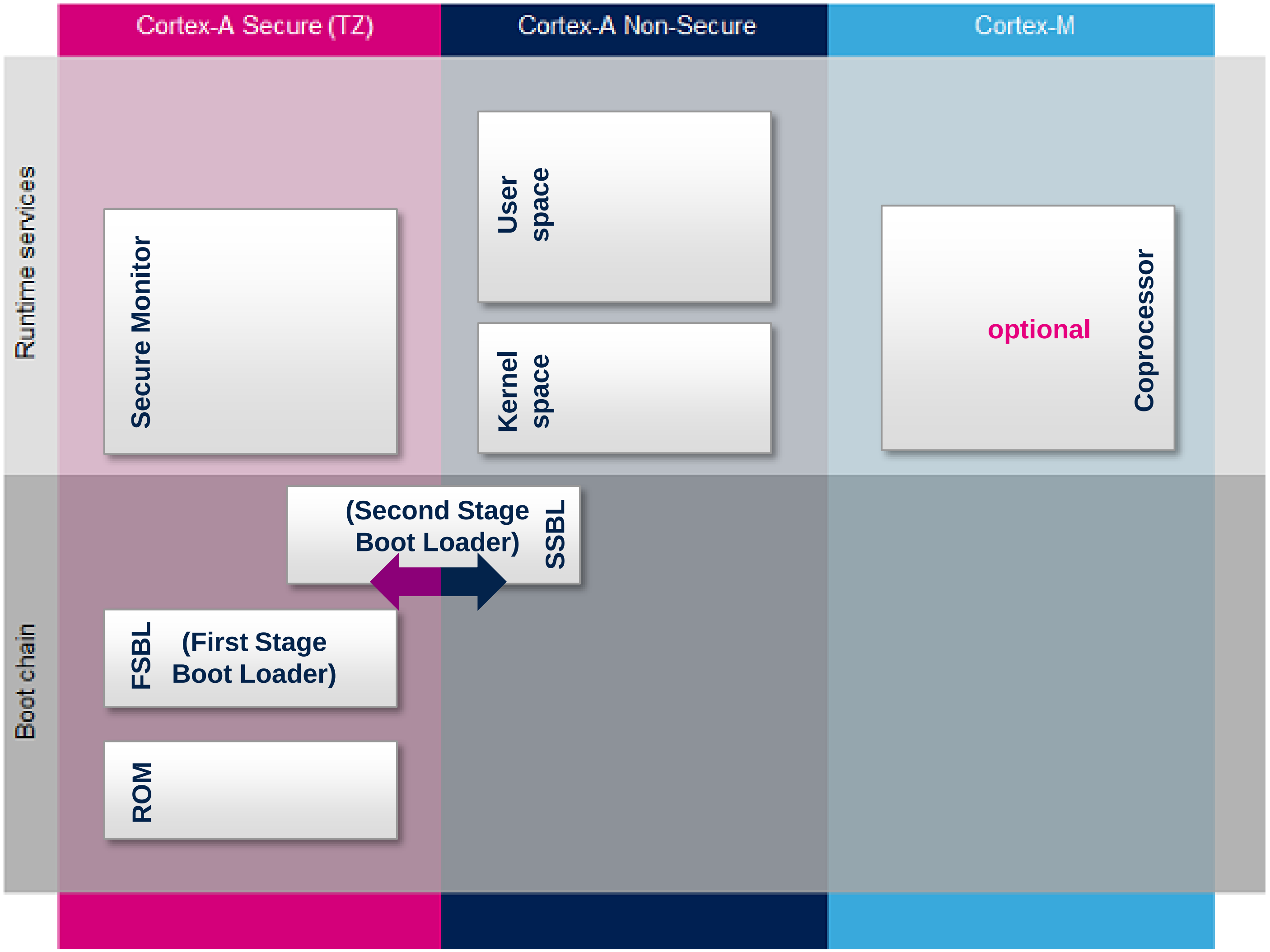
- Basic clock tree initialization
- FSBL loading from the boot device (mass storage or serial link)
- FSBL launch

life.augmented

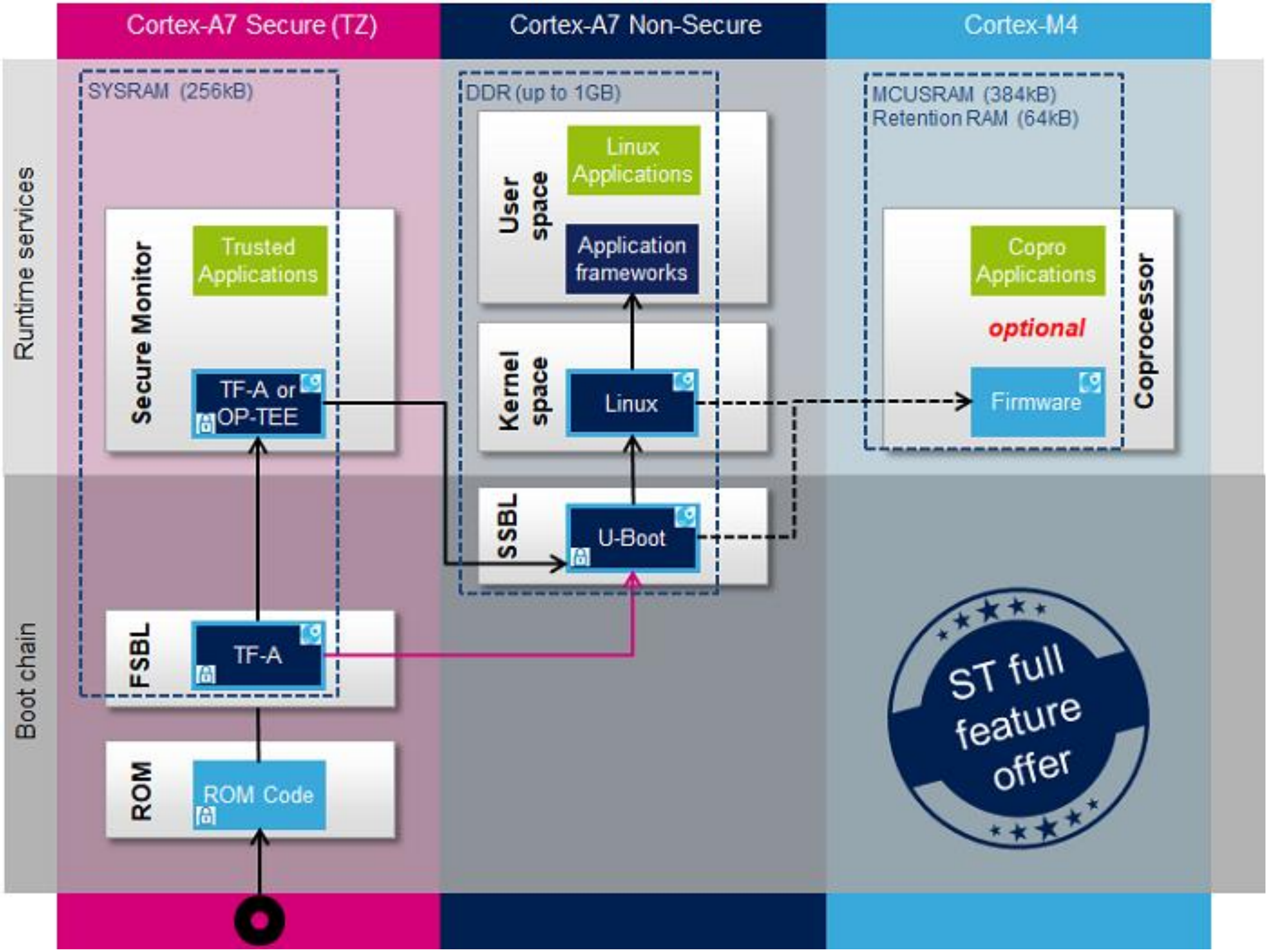
STM32MP1 boot chain (1/2)


Linux user space
Linux kernel
Coprocessor
Second Stage Boot Loader (SSBL)
Secure Monitor
First Stage Boot Loader (FSBL)
ROM Code

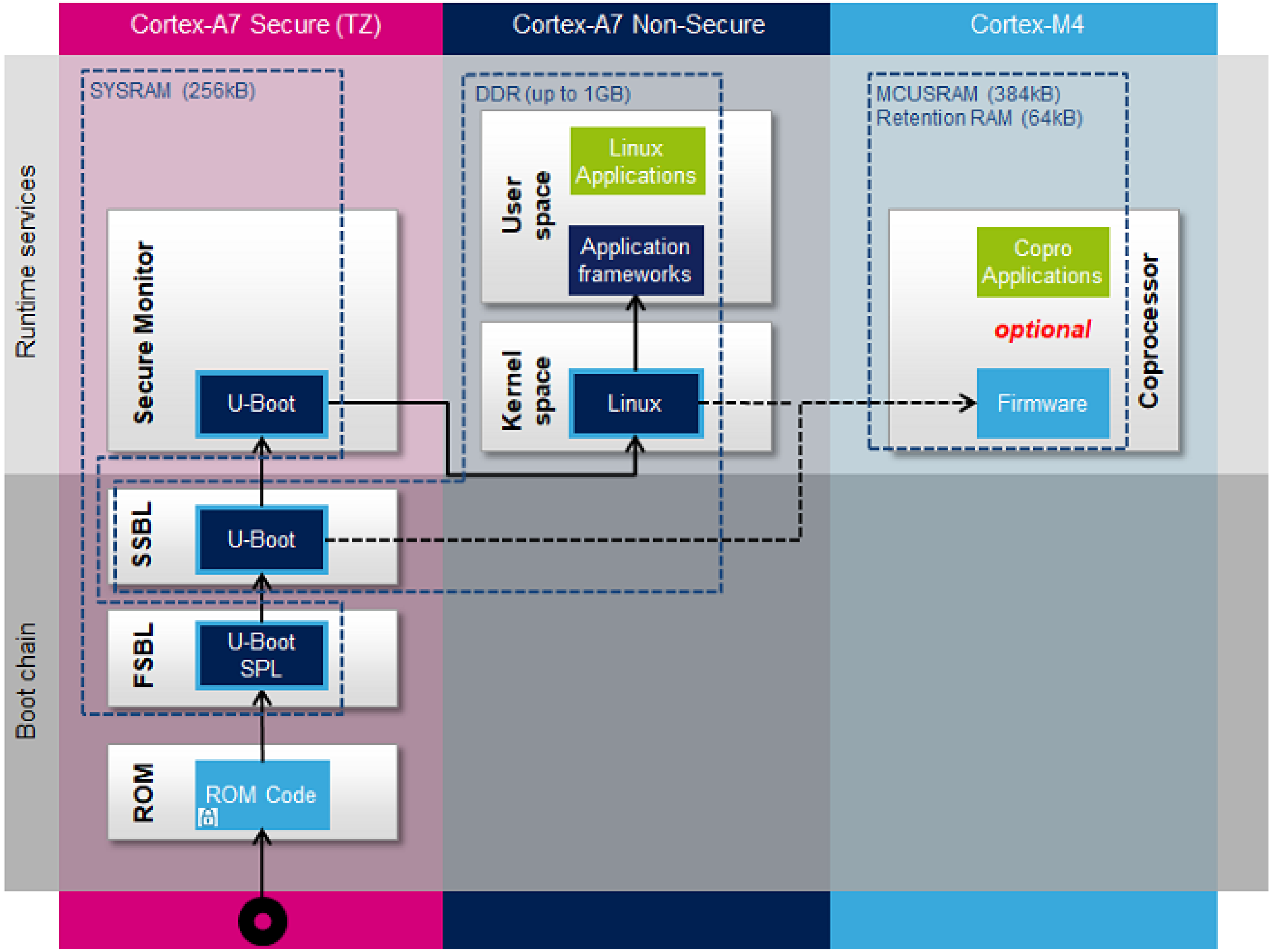
STM32MP1 boot chain (2/2)



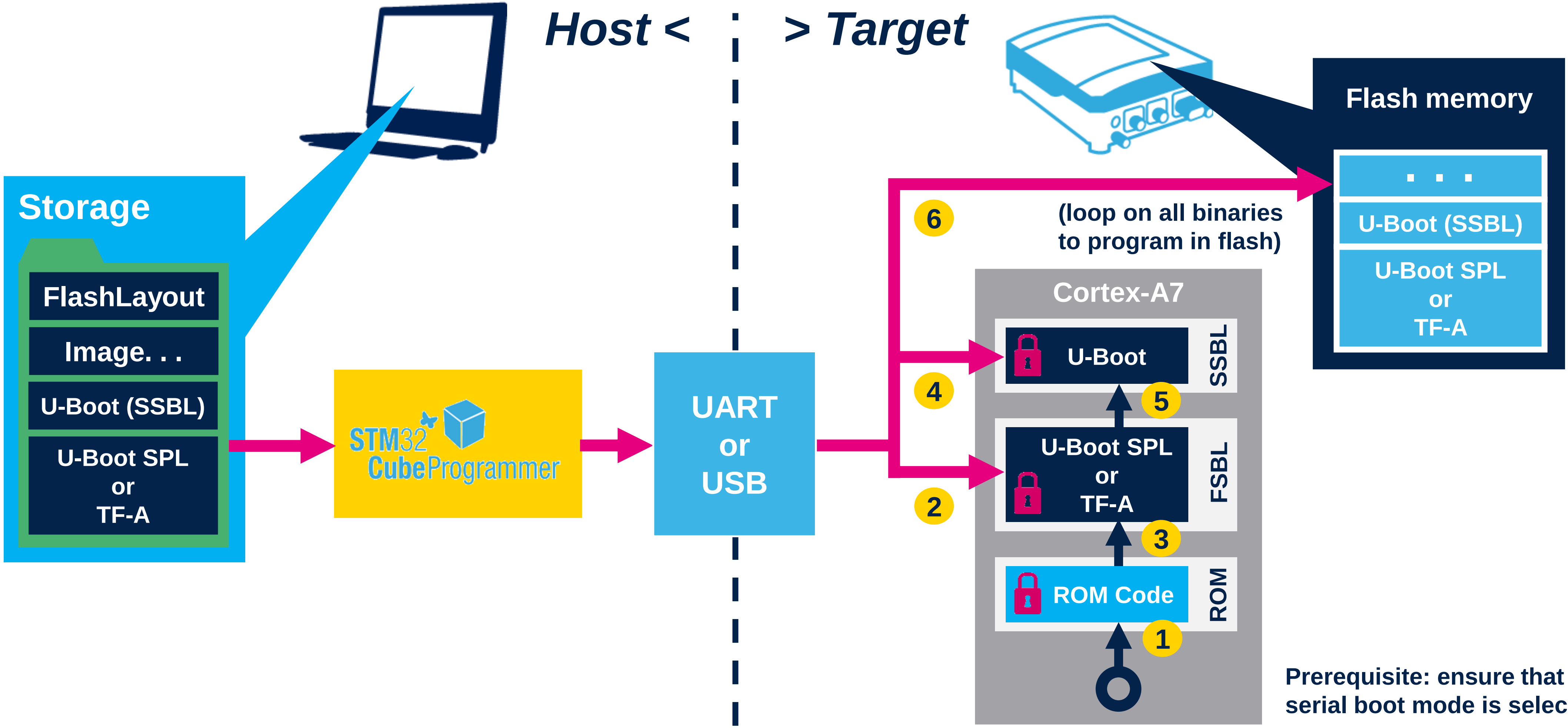
Trusted boot chain



Basic boot chain



STM32CubeProgrammer for IMAGE programming



Prerequisite: ensure that serial boot mode is selected

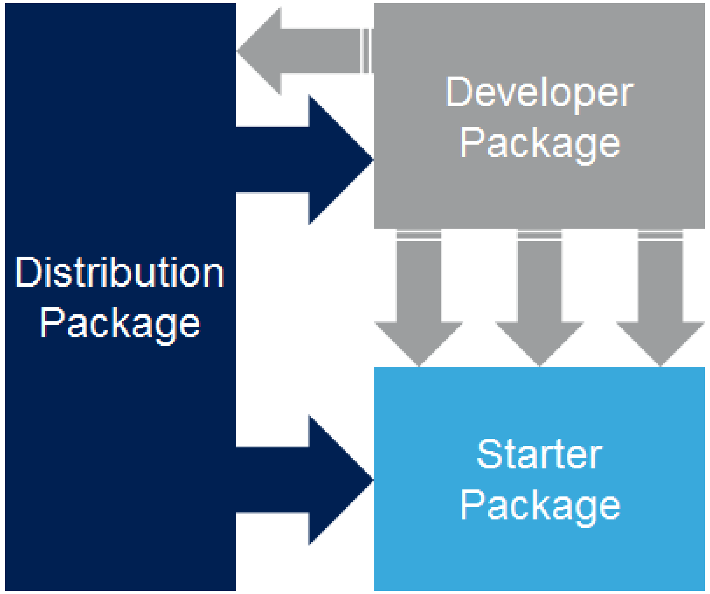
OpenSTLinux distribution

Concept of OpenSTLinux

- OpenSTLinux is a concept for STM32MPU embedded SW package
  - Concept = naming + associated pillars
- Pillars
  - Usage of a standard kernel interface (no proprietary interface)
  - Usage of Open Source software
  - Link to community (upstream)
  - Easy to use
- It supports OpenEmbedded build process
  - Yocto Compatible (target is to have a Board Support Package (BSP) hosted on Yocto server)

Available packages

- **Starter Package**
  - To quickly and easily start with any STM32MP microprocessor device.
  - This Package is generated from the Distribution Package.
- **Developer Package**
  - To add your own developments on top of the STM32MPU Embedded Software distribution, or to replace the Starter Package pre-built binaries.
  - This Package is generated from the Distribution Package.
- **Distribution Package**
  - To create your own Linux® distribution, your own Starter Package and your own Developer Package.



Packages and use cases

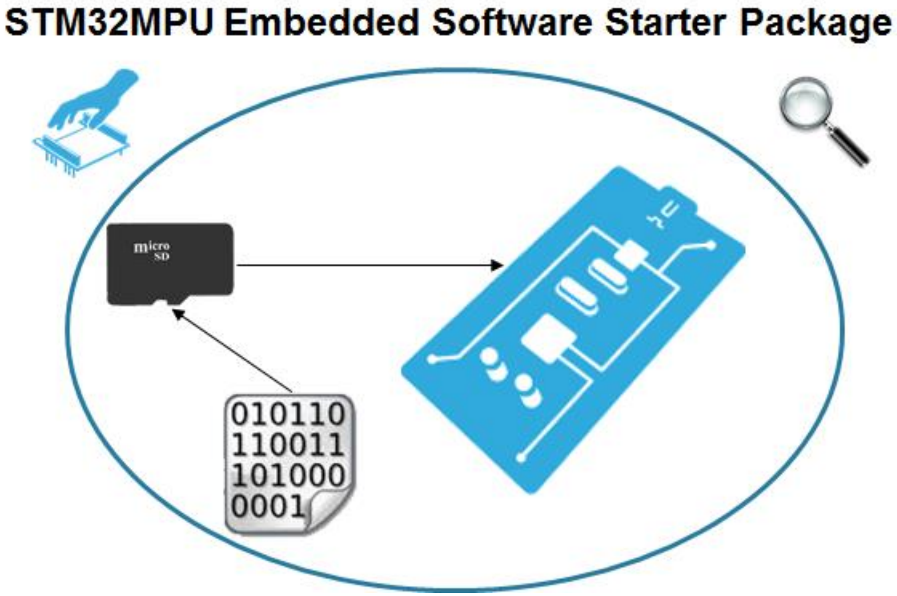
- Packages

Starter Package	Flashable images
Developer Package	Software Development Kit (SDK) and BSP tarballs
Distribution Package	Open Embedded distribution full source

- Rationale
  - Cover all the possible usages from our customers
    - Discover -> Prototype -> Start new hardware -> Productize software
  - Customer receives a board with an OpenSTLinux Starter package
    - Assess board capacities and performance
  - Customer wants to run existing applications = Starter package
  - Customer wants to develop their own application = Developer package
  - Customer wants to start their own hardware = prototype with Developer package then Distribution
  - Productization of software = Distribution package

Starter package

- STM32MPU Embedded Software Starter Package
  - ST images stored with HW diversity flashlayout.zip = Yocto based images

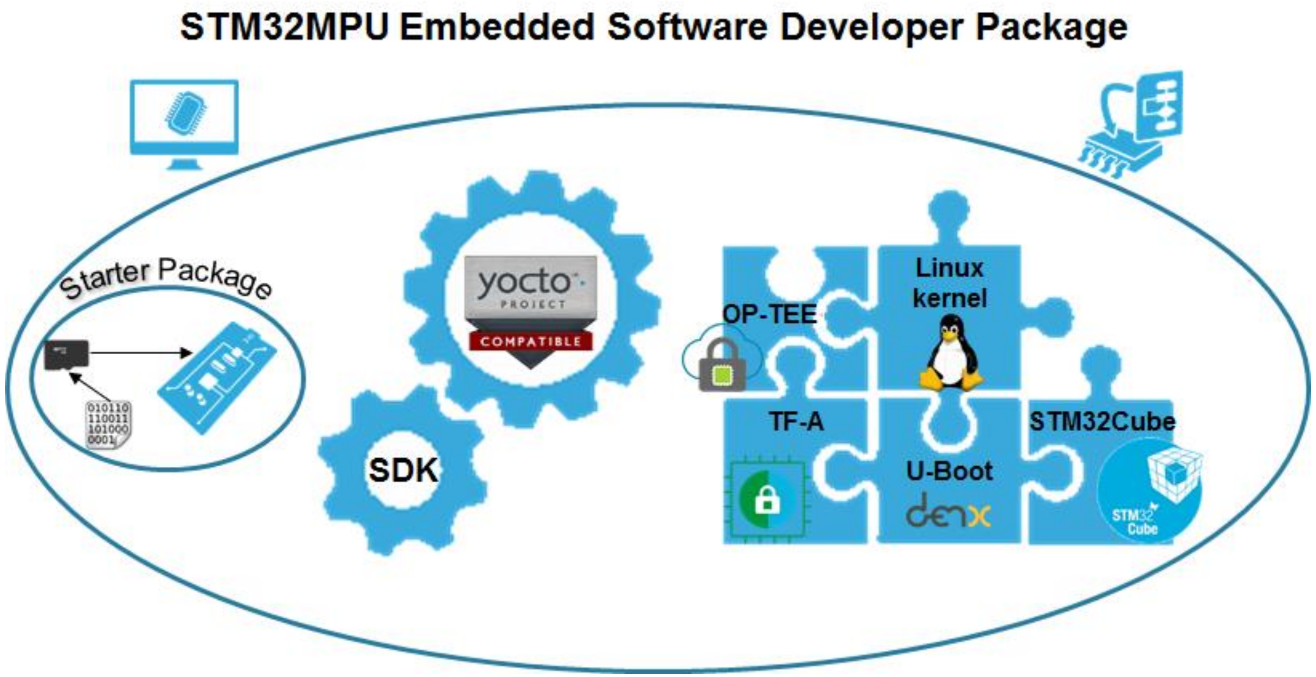


Focus on starter package

- Binaries ready to use by customer: Yocto/OpenEmbedded image (Weston)
  - Demonstrate the capacity of the platform
  - Available on microSD card or directly flashed onto the board
  - At this stage, Yocto starter package is only a prerequisite for developer kit
- Deliverables
  - Set of binaries with a specific flash layout
  - Script to generate an SD card raw image
  - Complete combinations (binaries x flash layouts) of microSD card ready to use images (aka stimg) are not provided by default; there are too many configurations

Developer package

- Yocto based on **st-image-weston**
  - SDK (Toolchain + Includes)
  - Sources (Community Tarball + ST Patches + ST configs)
    - Kernel and Boot (U-Boot, ATF)
  - STM32MP1-M4 Cube



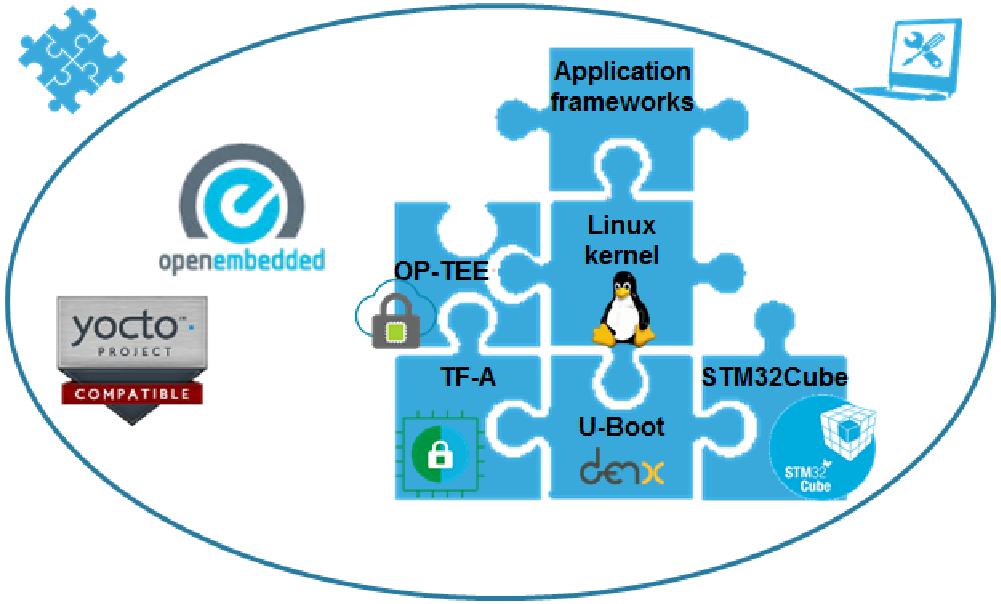
Focus on developer package

- Developer package = SDK (Software Development Kit) + Source code
  - Uses Starter package images
  - Only Yocto/OpenEmbedded SDK, based on Weston images, is provided
  - Source code provided
    - Kernel, U-Boot, ATF, OPTEE (optional), STM32Cube
  - Pre-compiled toolchain
- Release mode depends on the project stages
  - Alpha customer
    - Source code (tar ball from community) + patch
  - Mass market
    - Source code (tar ball from community) + patch
    - Git (ST github) = community content + all patches pending upstream

Distribution package

- Yocto based - (ST layers + ST patches on Git community)
  - oe-manifest
  - meta-st-stm32mp
  - meta-st-openstlinux
  - STM32MP1-M4 Cube
  - meta-st-custo (to customise via appends per customer)

STM32MPU Embedded Software Distribution Package



OpenEmbedded

- Project initiated by the Linux Foundation in 2010 and is still managed by one of its fellows: Richard Purdie.
- Linux-based cross-compilation framework
- Open source (but can be used to build proprietary code)
- It is based on git for software configuration management

OpenEmbedded ecosystem

- People talk about Yocto, Poky or OpenEmbedded and this can be confusing,

Name	Description
OpenEmbedded	Build Framework for embedded Linux Maintained by the community Source version of Poky Setup mainly consolidated for ARM platforms
Yocto	A project that uses OpenEmbedded build system
Poky	Poky is a reference system of the Yocto Project - a collection of Yocto Project tools and metadata that serves as a set of working examples. Poky uses OpenEmbedded Core Poky is maintained by Intel. Setup is mainly consolidated for Intel platforms

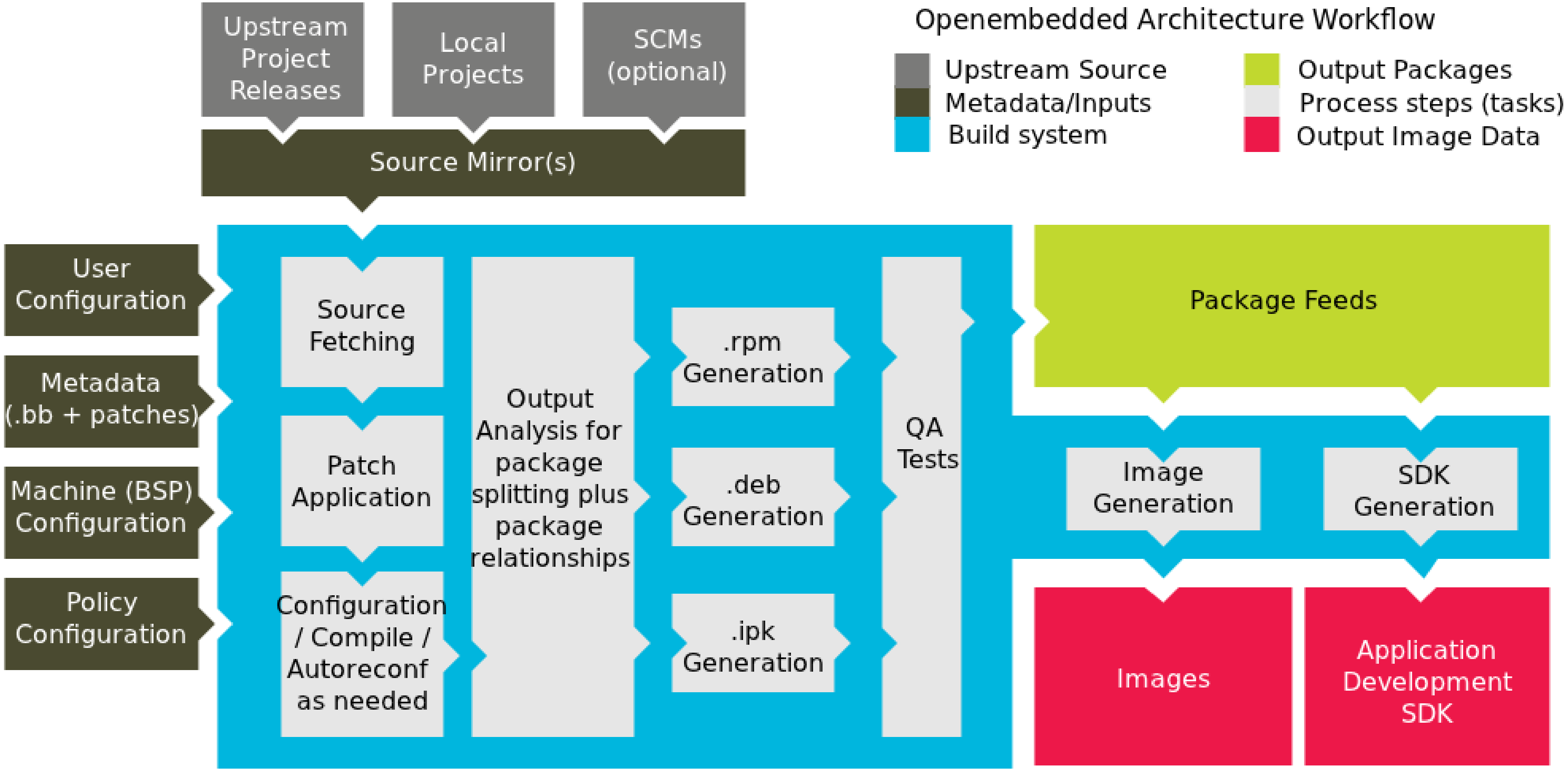
- Some projects works on a Yocto base, some others on a Poky base, but in the end everything is compatible.

OpenEmbedded

- What OpenEmbedded does,
  - Source code download
  - Patch application
  - Cross compilation
  - Package management
- What OpenEmbedded generates,
  - Binary packages
  - Linux-based system images
  - Toolchains
  - SDKs (Software Development Kits)

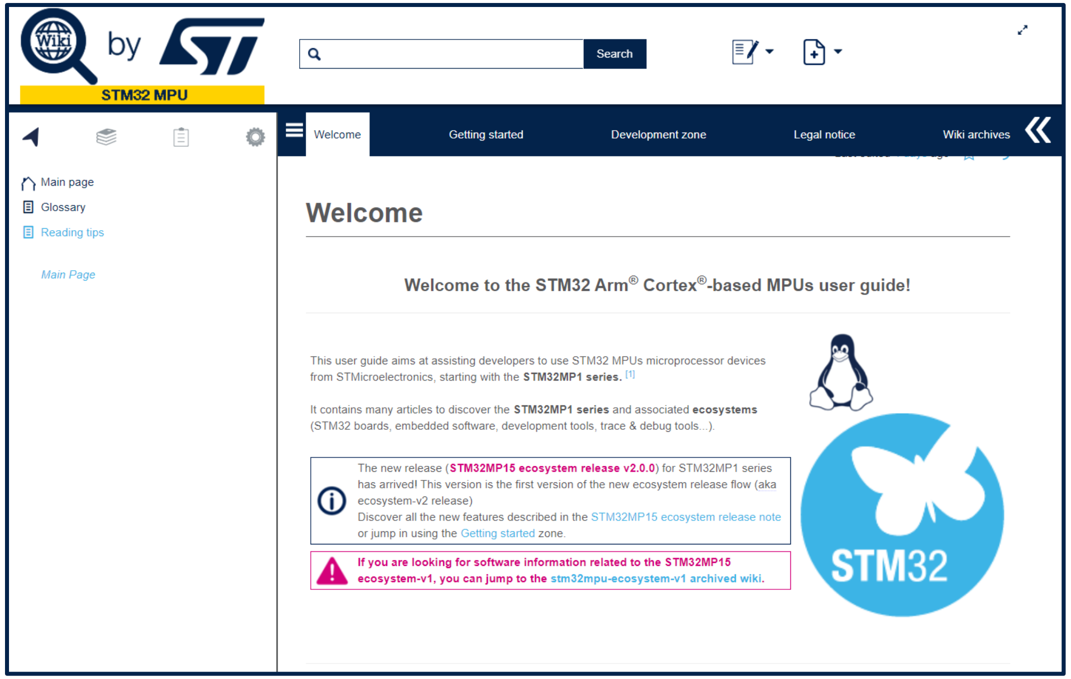
Poky Tool

Compilation flow



STM32MP1 Wiki and community

- STM32MP1 Wiki: **https://wiki.st.com/stm32mpu/index.php/Main_Page**
- STM32 MPU Community: **https://community.st.com/s/topic/0TO0X0000003u2AWAQ/stm32-mpus**



Welcome

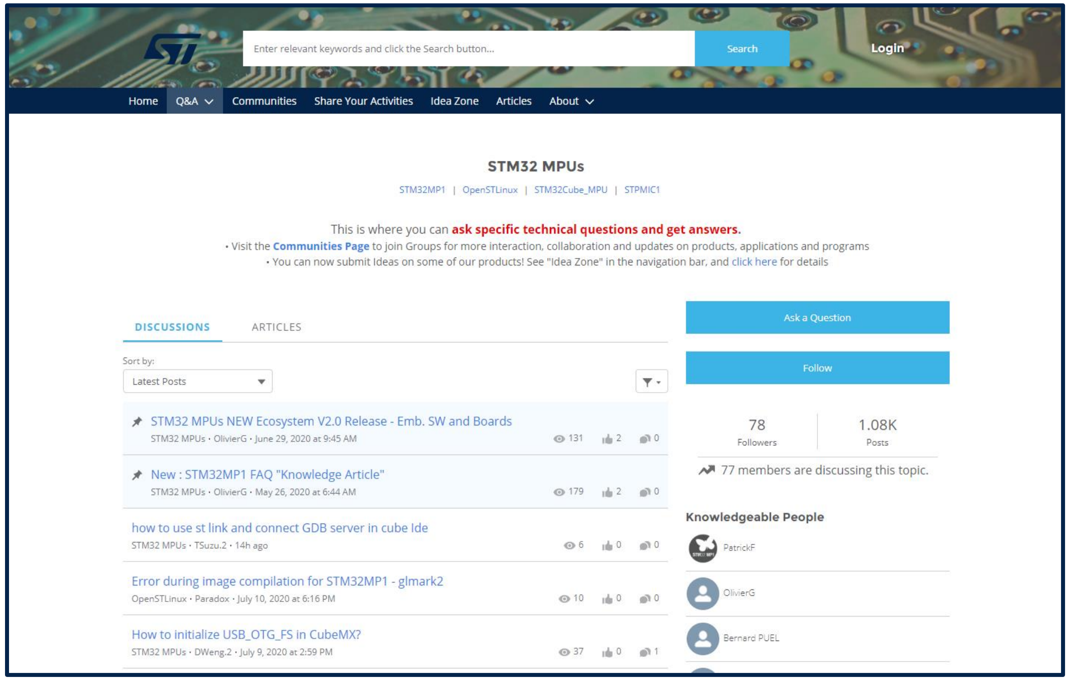
Welcome to the STM32 Arm® Cortex®-based MPUs user guide!

This user guide aims at assisting developers to use STM32 MPUs microprocessor devices from STMicroelectronics, starting with the **STM32MP1 series.** [1]

It contains many articles to discover the **STM32MP1 series** and associated **ecosystems** (STM32 boards, embedded software, development tools, trace & debug tools...).

The new release (**STM32MP15 ecosystem release v2.0.0**) for STM32MP1 series has arrived! This version is the first version of the new ecosystem release flow (aka ecosystem-v2 release)
Discover all the new features described in the STM32MP15 ecosystem release note or jump in using the Getting started zone.

If you are looking for software information related to the STM32MP15 ecosystem-v1, you can jump to the stm32mpu-ecosystem-v1 archived wiki.



STM32 MPUs

STM32MP1 | OpenSTLinux | STM32Cube_MPU | STPMIC1

This is where you can **ask specific technical questions and get answers.**
• Visit the **Communities Page** to join Groups for more interaction, collaboration and updates on products, applications and programs
• You can now submit Ideas on some of our products! See "Idea Zone" in the navigation bar, and click here for details

STM32 MPUs NEW Ecosystem V2.0 Release - Emb. SW and Boards
STM32 MPUs · OlivierG · June 29, 2020 at 9:45 AM 131 2 0

New : STM32MP1 FAQ "Knowledge Article"
STM32 MPUs · OlivierG · May 26, 2020 at 6:44 AM 179 2 0

how to use st link and connect GDB server in cube Ide
STM32 MPUs · TSuzu.2 · 14h ago 6 0 0

Error during image compilation for STM32MP1 - glmark2
OpenSTLinux · Paradox · July 10, 2020 at 6:16 PM 10 0 0

How to initialize USB_OTG_FS in CubeMX?
STM32 MPUs · DWeng.2 · July 9, 2020 at 2:59 PM 37 0 1

78 Followers | 1.08K Posts

77 members are discussing this topic.

Knowledgeable People
PatrickF
OlivierG
Bernard PUEL

Demonstration

life.augmented

Demo

- STM32MP1 DK2 Board
  - ST-LINK embedded debug tool
  - LEDs, push-buttons
  - Ethernet 1-Gbps connector
  - USB Type-C^{TM} OTG connector
  - HDMI® transceiver
  - LCD display with a touch panel
  - microSD^{TM} connector
  - Wi-Fi® and Bluetooth® Low Energy

Thank you

life.augmented