Zephyr and Cryptography

Embedded Recipes 2025 - Valerio Setti - vsetti@baylibre.com

Intro

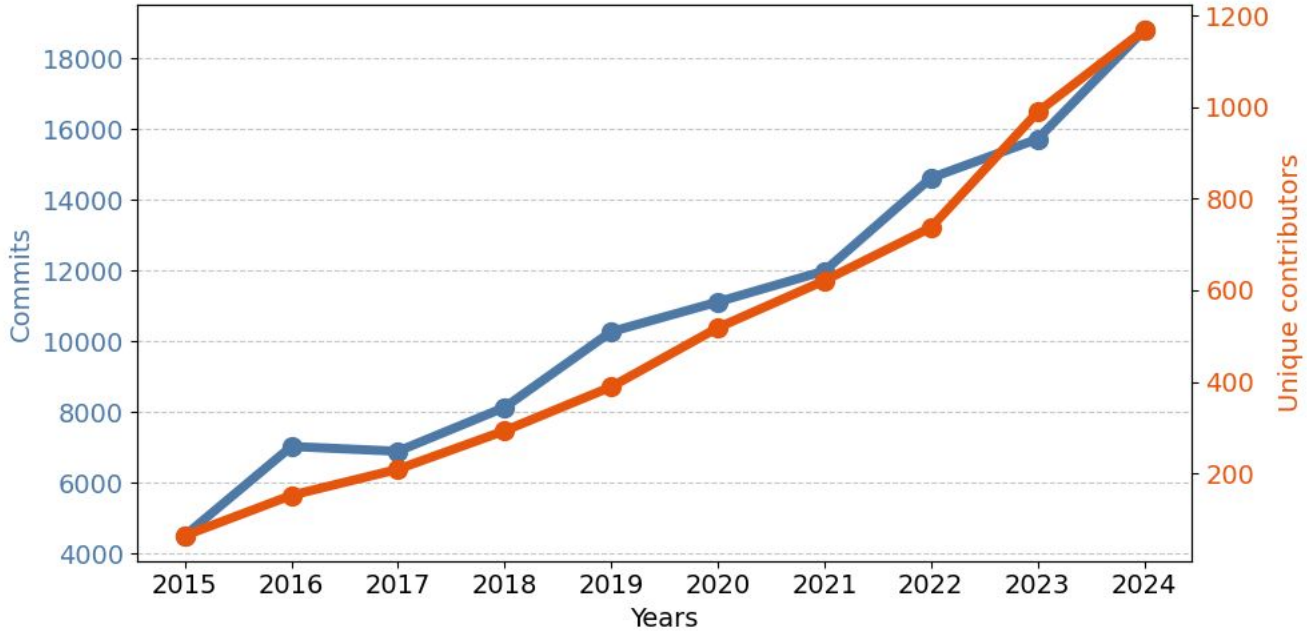
- About BayLibre
  - Open Source consultancy company specializing in all things embedded: Linux Kernel, Yocto, U-Boot, Zephyr, hardware design, compilers and toolchains (GNU)
    - Check the blog page for an up to date list of all our contributors
  - Based in Nice, France, but we have a global presence (US, Canada, UK, Germany, Italy, Taiwan, and more)
- About me
  - Mbed TLS and Zephyr contributor since 2022
  - Part of Zephyr's
    - Security Working Group
    - Security Committee
    - Technical Steering Committee

RTOS options on the market

- Several open source RTOS-es available on the market that support a good amount of platforms
  - FreeRTOS (Amazon AWS; MIT license)
  - ThreadX (Eclipse Foundation; MIT license)
  - ChibiOS (Giovanni Di Sirio; GPL3/proprietary license)
  - NuttX (Apache Software Foundation; Apache 2.0 license)

- Zephyr RTOS
  - Owned by Linux Foundation
  - Apache 2.0 license

Zephyr growth in the last decade



"In 2024, Zephyr had 1,100 unique contributors with more than 50% being first-time contributors" [1]

Current support

- Support covers basically almost all possible 32/64 bit platforms
  - Cortex-M/A, Xtensa, RISC-V just to name a few popular ones
- Many supported boards
  - Currently 743 - continuously increasing
- There are several products already on the market running Zephyr
  - https://www.zephyrproject.org/products-running-zephyr/

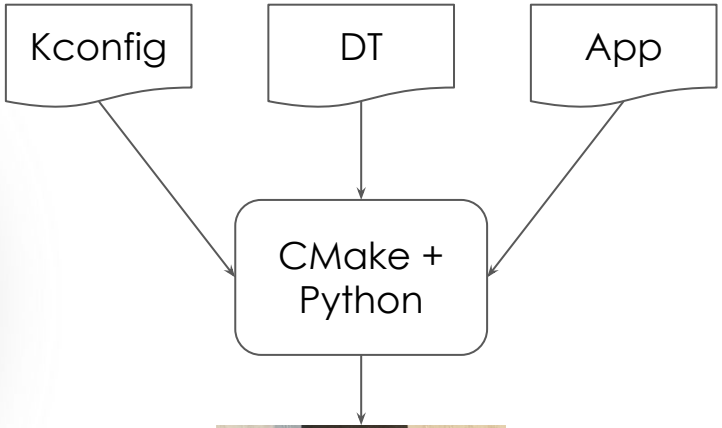
Reasons for Zephyr's success

- Full featured kernel

- High number of optional subsystems and modules
  - Networking stack (Bluetooth, Wi-Fi, …)
  - USB
  - Power Management
  - IPC for AMP both as Zephyr<->Zephyr and Zephyr<->Linux(rpmsg)
  - Non-volatile storage (Classic filesystems like EXT2 or Zephyr's specific ones)
  - Graphics (LVGL)
  - Logging
  - Shell
  - → Crypto ←
  - … and many others…

Development flow

- Configuration
  - Kconfig based configuration files to enable drivers, modules and subsystems and to control their features
  - Device-tree to describe the hardware
  - ≈ embedded Linux
- Build system
  - CMake based
  - Python for checks and code generation
- Developers only need to focus on the application



LILYGO
T-Watch S3

Security in IoT devices (1/3)

- Zephyr is a great platform for Internet-of-Things (IoT) devices
  - Security is essential
- Bluetooth, Bluetooth Low-Energy (BTLE) and Wi-Fi's supplicant require crypto to provide network functionality
- New protocols being introduced for IOT
  - Thread (supported with OpenThread)
  - Matter

Security in IOT devices (2/3)

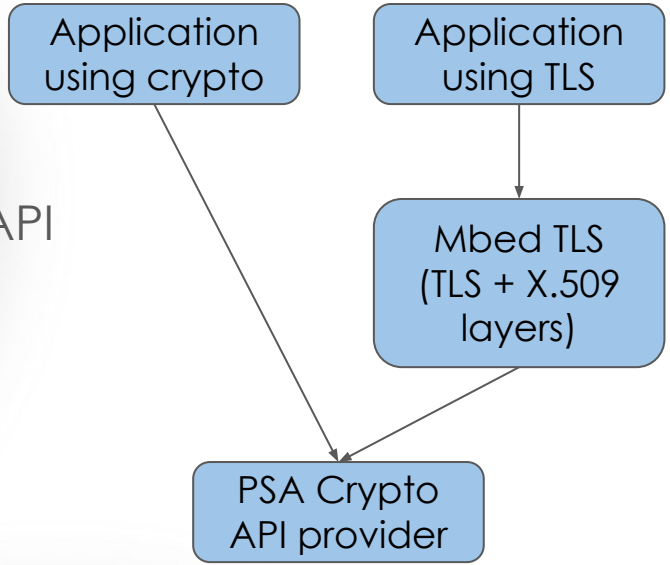
- TLS sockets
  - IOT device needs to connect to or implement a server
  - It must handle all TLS details: certificates, key exchange, [en|de]cryption, …
- JWT (JSON Web Token)
  - Data exchange with servers
- Random value generation
  - HW IPs usually only provide entropy data
  - This can be used as seed to generate random values
  - Provide cryptographically secure random generators
  - Important for a non-predictable behavior

Security in IOT devices (3/3)

- Secure bootloader
  - MCUboot
  - Verifies signature of following images before loading them
  - OTA update
- Secure Storage
  - Allows secure storing of sensitive data (keys, certificates, …)
- Flash management
  - Compute hash of flash memory for integrity check

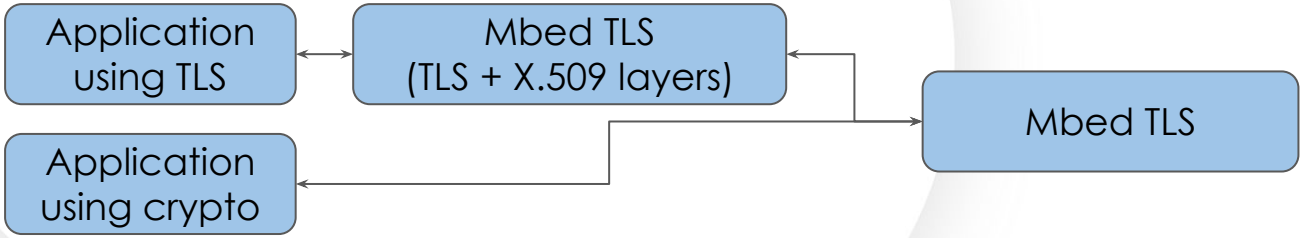
Standard crypto interface

- Need for a crypto provider
  - Directly from the application
  - Indirectly through TLS/DTLS
- Zephyr is transitioning toward PSA Crypto API
  - There's some residual legacy API usage, but that should be removed soon
- PSA (Platform Security Architecture) API
  - Standard interface designed by ARM
  - Provides API also for:
    - Secure Storage
    - Attestation
    - Firmware update

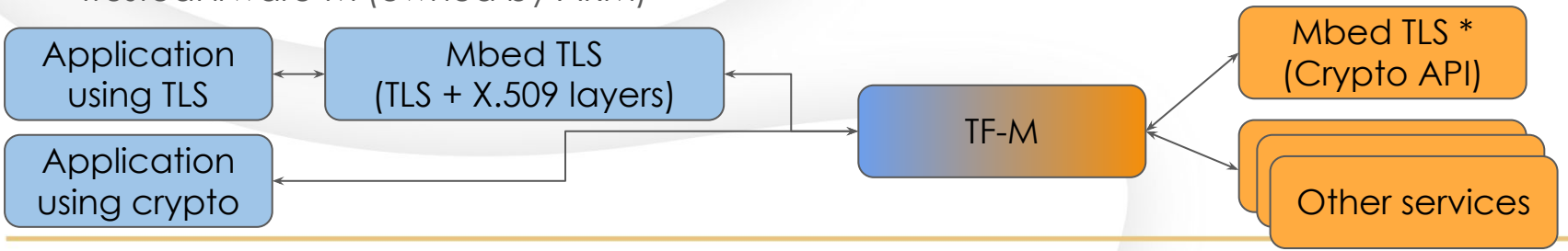


PSA Crypto provider

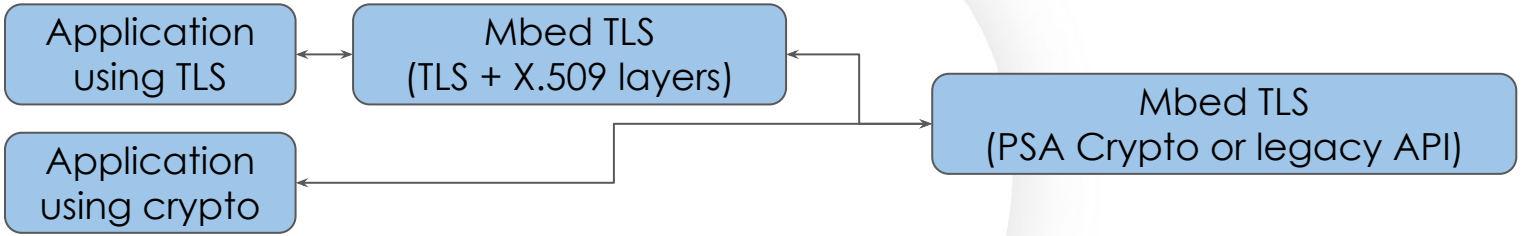
- 1 or 2 options depending on the hardware support
  - Mbed TLS (owned by ARM)



  - TrustedFirware-M (owned by ARM)



Mbed TLS (1/2)



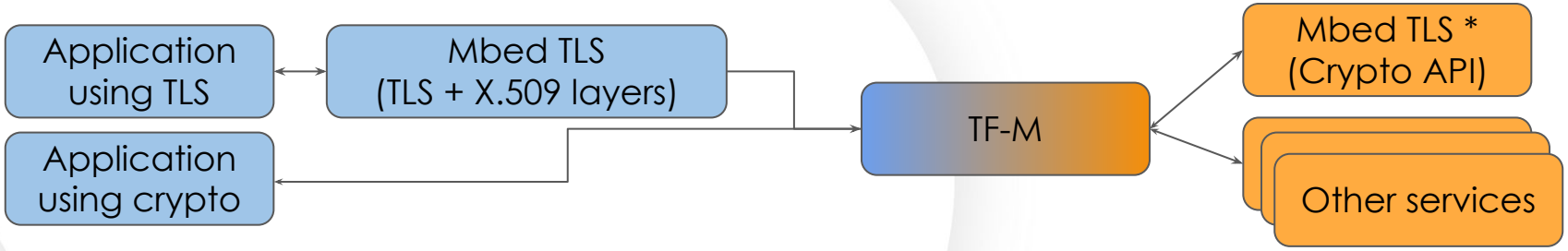
- Provides the full TLS stack
  - TLS + X.509 + Crypto
- Crypto is provided through PSA API or legacy interface
  - Legacy is soon to be removed from the public interface (only internally used)

Mbed TLS (2/2)

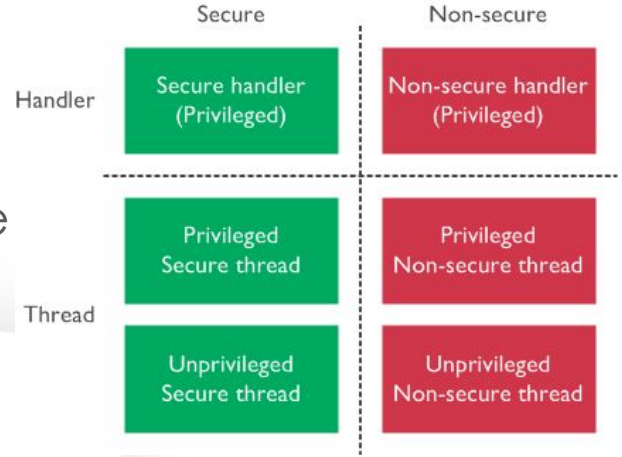
- Software based solution
  - Pro
    - Available for all platforms
  - Cons:
    - It supports few optimizations for some CPU and some alg (ex: AES using ARMv8 Cryptographic Extension), but not in general
    - No support for hardware accelerators out of the box

- Credentials
  - On RAM during execution but memory can be dumped
  - Not securely stored at rest
    - App developer need to take care of this
    - Secure Storage might be a solution

TrustedFirmware-M (1/2)

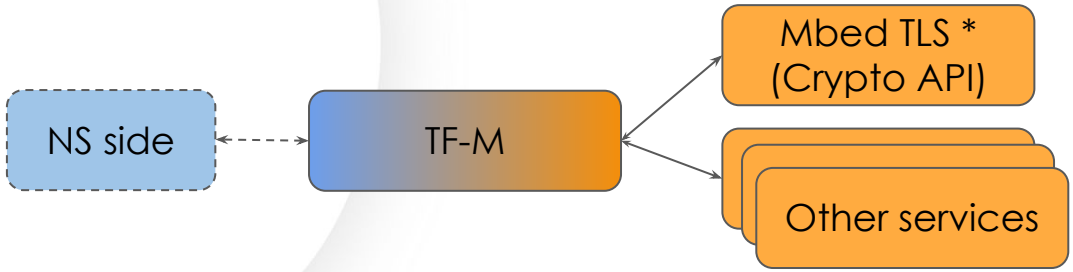


- Only available on ARMv8-M and ARMv8.1-M architectures (ex: Cortex-M33)
  - Based on ARM TrustZone
- Isolation between secure (TF-M) and non-secure code (Zephyr)
  - PSA API call: non secure -> secure -> non secure



TrustedFirmware-M (2/2)

- TF-M implements all the PSA API services
  - Crypto
  - Secure storage
  - Attestation
  - Firmware update
  - Status



- Relies on Mbed TLS to implement and provide PSA Crypto API service
  - Custom patches to allow hardware accelerators to be used (ex: ARM CryptoCell-312)

Future work

- Zephyr
  - Complete the transition to PSA Crypto API in Zephyr and remove all legacy crypto
  - Continue with refactoring/improving integration of Mbed TLS
- Mbed TLS
  - Continue the work for the next release planned September 2025
    - Repo split (TLS/X.509 <-> Crypto)
    - Removal of legacy crypto API
  - After that, add support for hardware accelerators
    - Vendors can easily support their own accelerators
    - Useful for both full Mbed TLS and TF-M scenarios

Thanks for your attention

Questions?