Secure Boot

UNDER CONSTRUCTION

Overview

The purpose of a "*secure*" bootloader is to ensure that first software to run on platform is the one expected. In order to achieve this goal, secure bootloader makes sure that it has not been altered, neither the cryptographic library.

When platform's secure bootloader is now to be trusted, next application to be launched, e.g. (Secure) Uboot or any other application$_{(\text{SLB})}$, shall be granted by secure bootloader. As a consequence, this "*next step application*" must be compliant with "*Secure Bootloader Application Format*".

The "*Power On Reset*"$_{(\text{POR})}$ always points to beginning of "*Secure Bootloader*" then; but it is divided in two parts:

- SBR which is common to all platforms, providing minimum but mandatory security features and capabilities to securely program/update/launch next step application$_{(\text{here SLB})}$. It is located in ROM or OTP.
- SLB which is platform specific, checking final application format$_{(\text{header + signature})}$, initializing devices for primary customer application. One SiFive implementation of SLB is Second Flexible Boot.

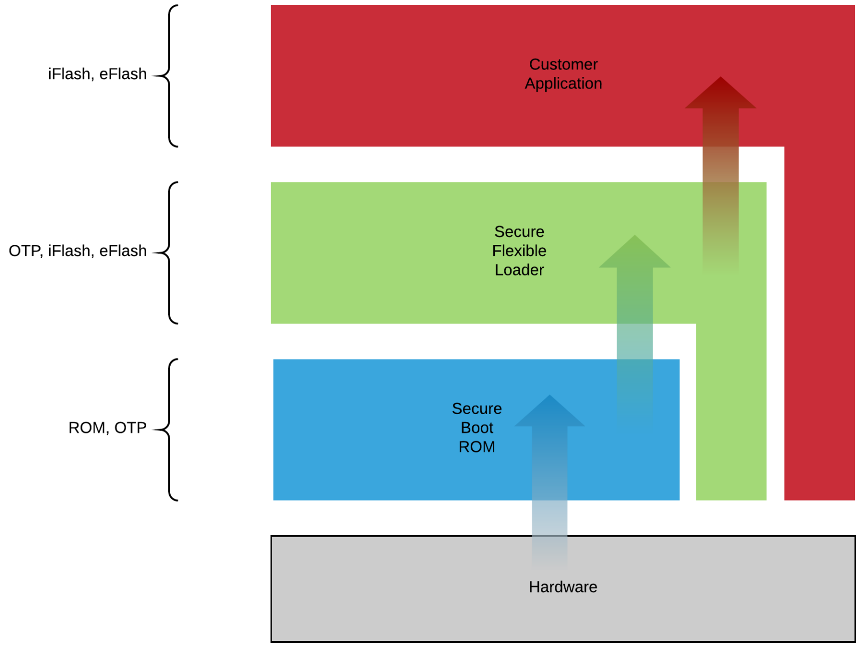
When coming out of "*Low Power*" mode, execution must pass through SBR before resuming execution where it stops before entering "*Low Power*" mode. **To Be Adapted ... platform dependent ...**

[...]

Boot Flow

Here after is shown the nominal flow when platform starts from POR.

[...]



Secure Boot ROM

This bootloader's first part$_{\text{(zero-level)}}$ is mainly tied to SoC/core and as a consequence, its storage location too.

Here is the dedicated page for SBR.

[...]

Secure Flexible Loader

This section will deal with SFL *or equivalent.*

This bootloader's second part$_{\text{(first level)}}$ is adapted to customer's platform needs and especially to customer's application which is the entry point from application level point of view.

Here is the dedicated page for SFL.

[...]

Q&A

- What, Why, How ?
    - Answer is 42.