Second Level Boot

UNDER CONSTRUCTION

Overview

For better comprehension, **S**econd **L**evel **B**oot is generic term. **S**ecure **F**lexible **L**oader is one (SiFive) implementation of SLB.

The SBR is going to check for SFL$_{(or\ equivalent)}$ security integrity before launching it. Of course if this process fails, SBR does not launch SFL and goes into "*download mode*".

Boot Policy

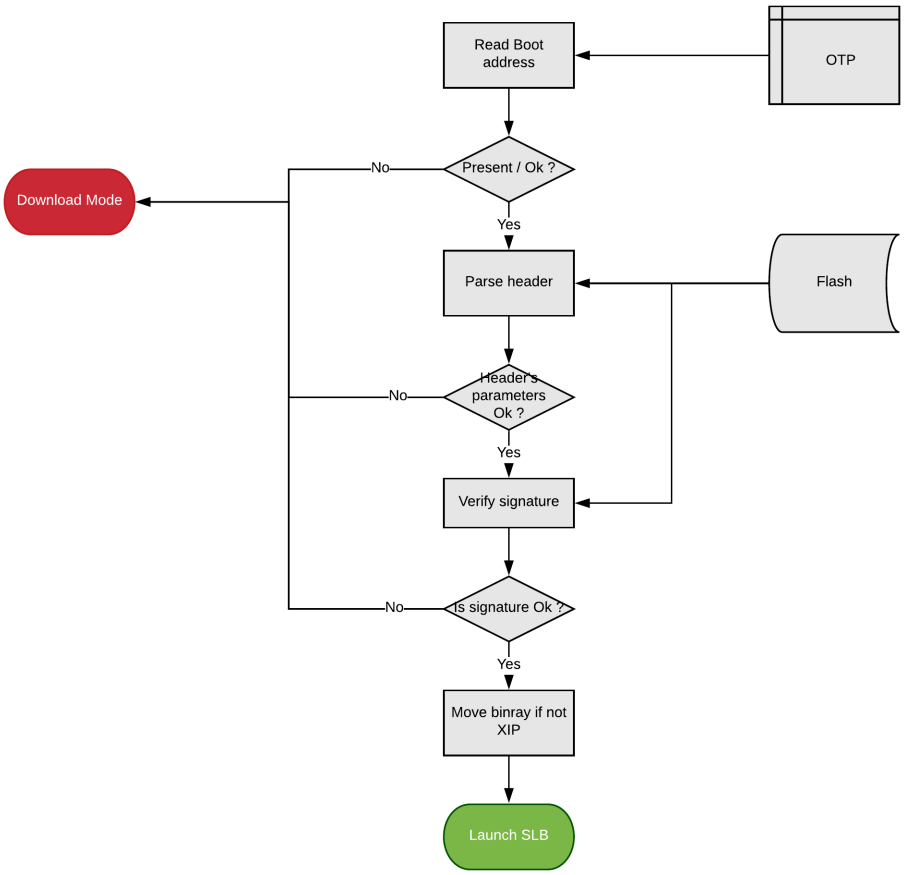
Boot policy determines how SBR access to binary image stored on platform and what it expects to find in SLB storage location.

Consequent access to binary is possible thanks to "*Boot_Address*". This parameters is stored in platform's non-volatile internal memory$_{(e.g.\ OTP)}$. It has defined depth to allow user to chane in time this value. It is completely platform dependent.

Raw Boot

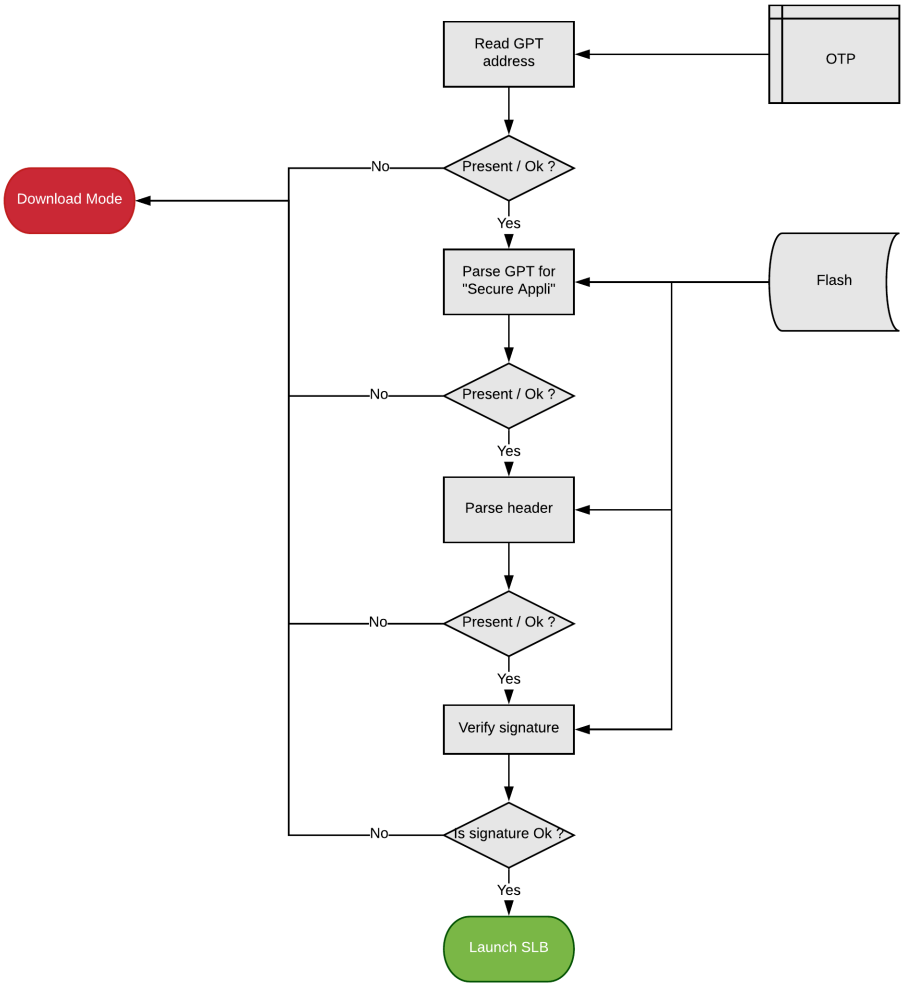
The "Raw Boot" characteristic is to be independent from any file system or software storage partitioning management. It means it is not based upon MBR nor on GPT.

The "*Boot_Address*" gives the address where to to directly find header of the binary to be checked and launched.



GPT Boot

Boot is based on GPT stored in external/internal memory(e.g. SPI flash). Note that specific partition identifier

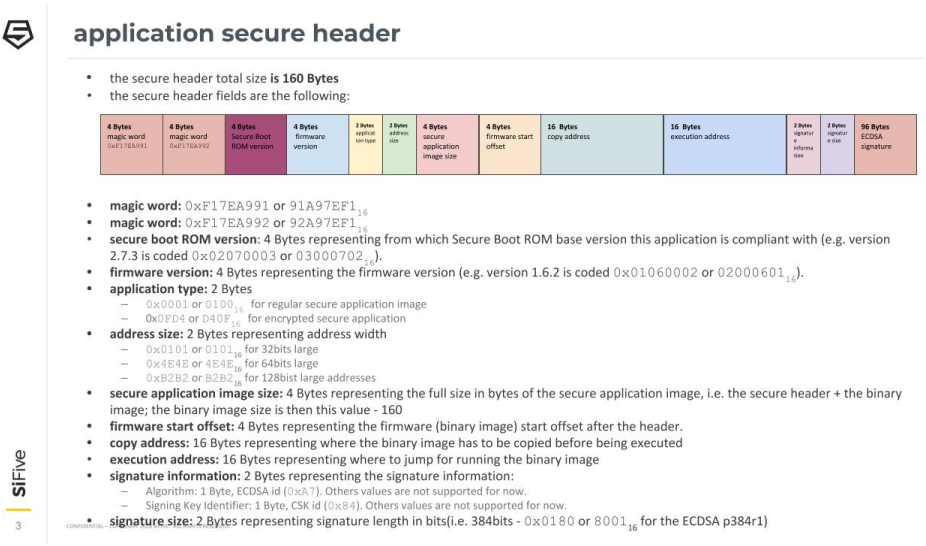


Format

[...]

This "secure" format applies to SFL,launched by SBR, and customer application, launched by SFL.

Header



Exit SBR

[...]

SBR has now successfully checked $SLB_{(SFL)}$, it restores used IP(s) to its/their "*start state*".

When SLB has been checked successfully and is now ready to jump at "*jump_address*" found in header.