

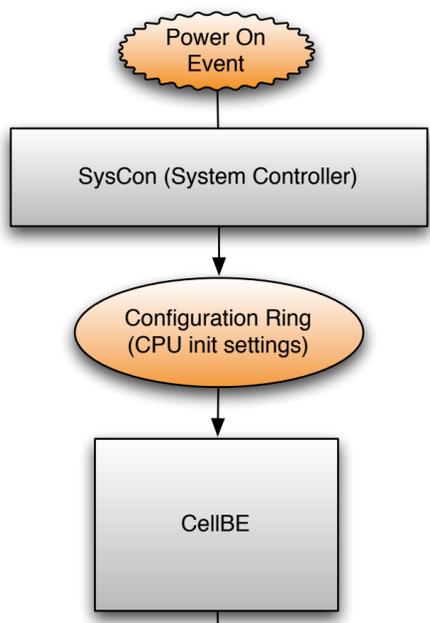
CellBE Secure Boot Process

Time

Power On Reset Sequence (POR)

Reset Vector for secure boot
 NAND: 0x240_1FC0000
 NOR: 0x240_1FFC0000

Location Offset
 NAND: 0x0
 NOR: 0xFC0000

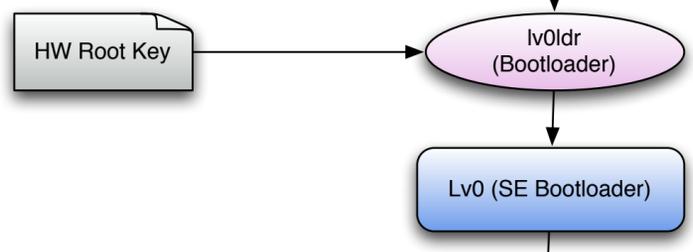


Loads configuration ring and calibrates I/O controller, will not do anything further in POR
 ref. CellBE HIG - 2.2

This contains the reset vector, which is the address of Iv0ldr. It also contains the initial register and cpu settings, these are passed to the CellBE
 ref. CellBE HIG - 2.3.4

Boot Sequence

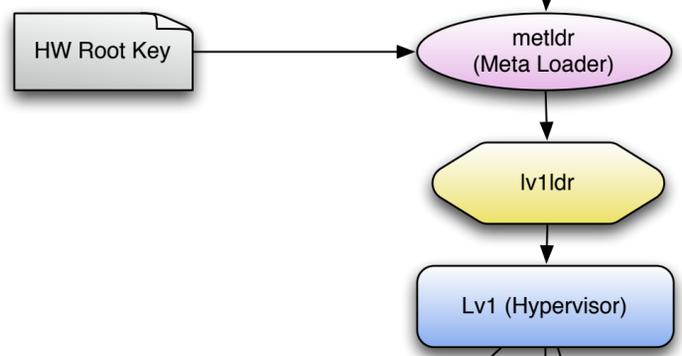
The HW Root Key is stored inside the CellBE hardware. This key is unique to each CellBE and is used to decrypt and verify SPU Secure Loaders such as metldr and Iv0ldr



SPU loads and executes Iv0ldr according to the reset vector provided. This differs to the process in non secure boot, where the ROM code is executed by PPU
 ref. CellBE HIG - 2.2.1

Lv0 is decrypted to the PPU RAM by Iv0ldr. Lv0ldr then starts the PPU executing Lv0 from RAM at address 0x100. There is no extra loader used on the SPU in this case.

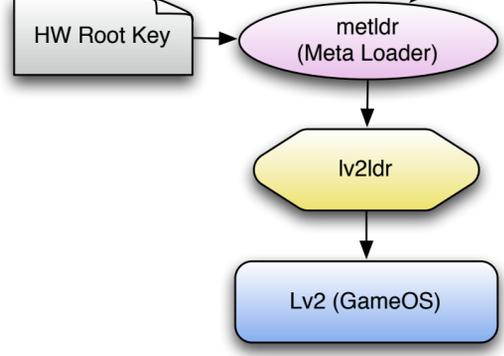
Hypervisor Init



Metldr is loaded to an isolated SPU to facilitate loading of each SPU Isolated Loader.
 ref. IBM Secure SDK Documentation

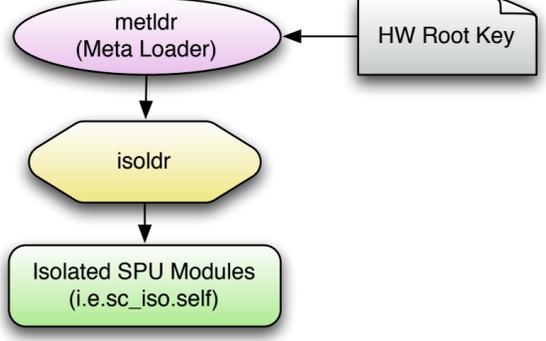
From now on access to hardware resources must be done via the Hypervisor. Direct access is no longer possible.

Kernel Init



The Game Operating System kernel runs on top of the hypervisor. Both this kernel and the hypervisor stay present in memory while all userland operations run on top of them.

Isolated SPU Init

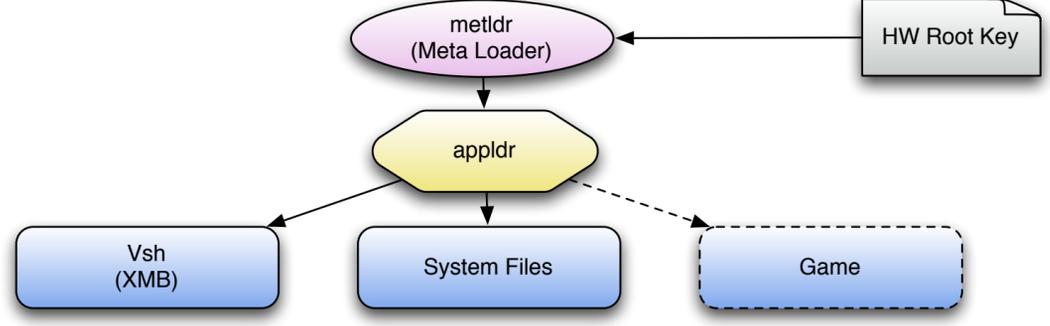


SPU Secure Loaders are loaded by CellBE hardware to the LS address 0x400. The CellBE decrypts, authenticates and then executes them at 0x400.

SPU Isolated Loaders are loaded by metldr to a high LS address. Metldr then zeros itself out and jumps to their entrypoint to begin their execution.

Isolated SPU Modules are loaded by isoldr to a low LS address. Isoldr then zeros itself out and jumps to their entrypoint to begin their execution.

Userland Init



System files like vsh (XMB) or games call back to the GameOS, which then call back to the hypervisor for certain operations. The processes in userland after boot are more dynamic, but this is outside the scope of this document.

Loading a game is optional

