INTEL ME

NICOLA CORNA

NICOLA@CORNA.INFO

GITHUB.COM/CORNA

INTEL ME

Cos'è?

Secondo Intel:

Il Motore di gestione Intel® è un microcontroller incorporato (integrato in alcuni chipset Intel) che esegue un sistema operativo microkernel leggero che offre un'ampia varietà di funzioni e servizi ai sistemi basati sui processori Intel®

E ancora...

Le funzioni includono (ma non si limitano a):

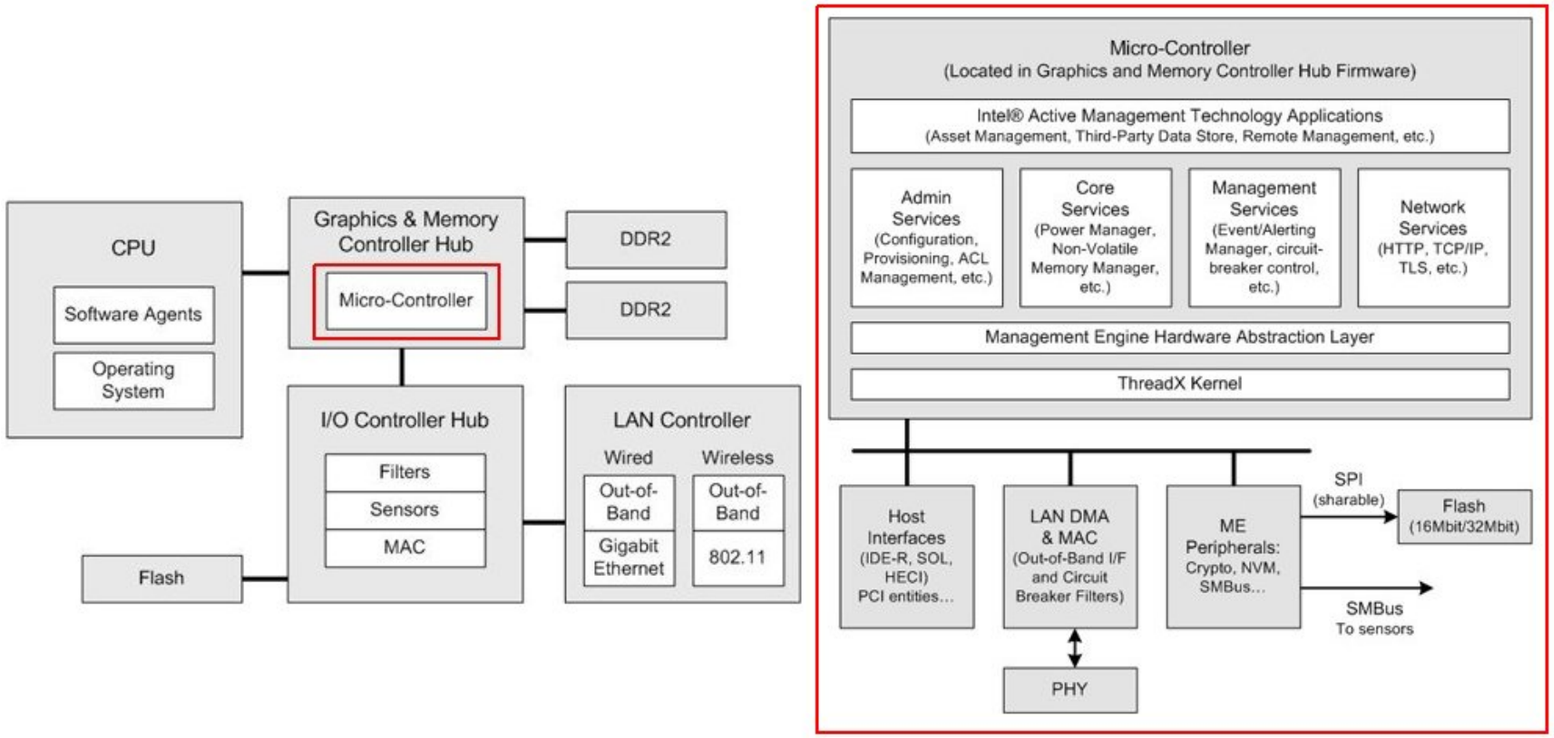
- *Servizi di gestione out-of-band (OOB) a basso consumo*
- *Capability Licensing Service (CLS)*
- ~~*Protezione anti-theft*~~
- *Protected Audio Video Path (PAVP)*

IN CHE SISTEMI È PRESENTE?

- Desktop (Intel (CS)ME)
- Mini-PC (Intel (CS)TXE)
- Server (Intel (CS)SPS)

DOV'È?

- Northbridge fino a Nehalem
- Platform Controller Hub (PCH) successivamente



(Intel, 2009)

GENERAZIONI DI INTEL ME/TXE/SPS

	Gen. 1	Gen. 2	Gen. 3
ME (desktop)	1.x-5.x	6.x-10.x	11.x-12.x
TXE (low-end)		1.x-2.x	3.x
SPS (server)	1.x	2.x-3.x	4.x
Core	ARC	ARC	Intel Quark (x86)
Kernel	ThreadX	ThreadX	Minix 3
Anni	2005-2008	2008-2015	2015-cur

Intel distribuisce principalmente due tipi di firmware per Intel ME:

- "Completo", vPro
  - 5 MB / 6.6 MB
- "Ridotto"
  - 1.5MB / 2 MB

COSA FA?

- ~~Advanced Management Technology (AMT)~~
- ~~Intel Anti-Theft~~
- ~~Intel Boot Guard (BG, o anche IBG)~~
- ~~firmware-based Trusted Platform Module (fTPM)~~
- ~~Quiet System Technology (QST)~~
- Protected Audio Video Path (PAVP)
- Integrated Clock Controller (ICC)
- ...

Per poter offrire queste funzionalità Intel ME ha accesso a:

- Memoria (DMA)
- Network
- Bus PCI
- ROM firmware
- ...

Inoltre per funzionare, Intel ME richiede all'host una piccola area di memoria (~16/32 MB, UMA region) durante la fase di boot.

POWER STATES

M0: host attivo, ME completamente funzionante

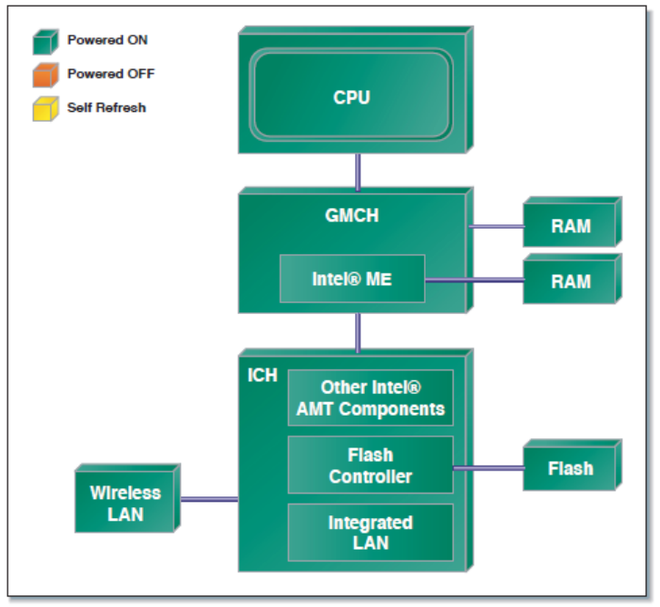


Figure 7.13 An Intel® AMT Computer in S0 and M0 State

Active Platform Management Demystified: Unleashing the Power of Intel VPro Technology, 2009, Intel Press

M1: host in sospensione, ME parzialmente funzionante

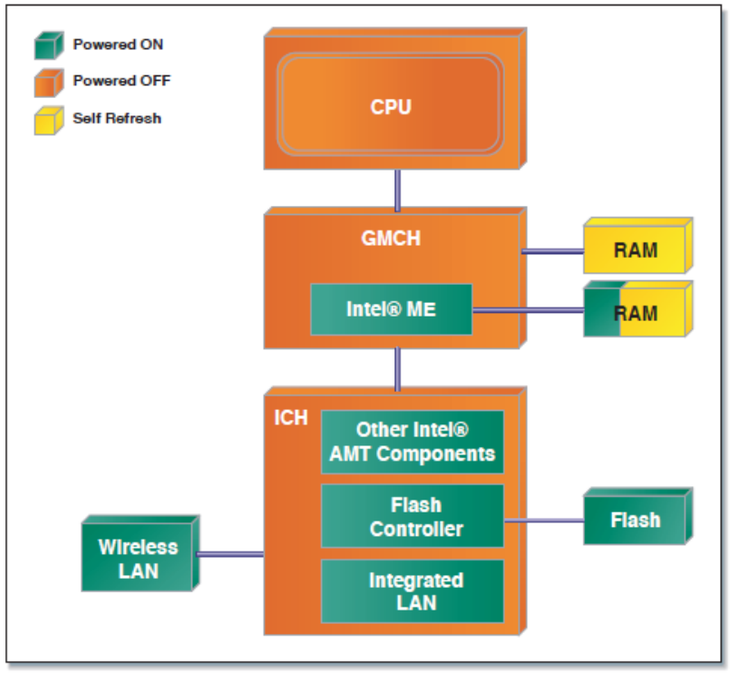


Figure 7.14 An Intel® AMT Computer in S3 and M1 State

Active Platform Management Demystified: Unleashing the Power of Intel VPro Technology, 2009, Intel Press

M-Off: host in sospensione/off, ME off

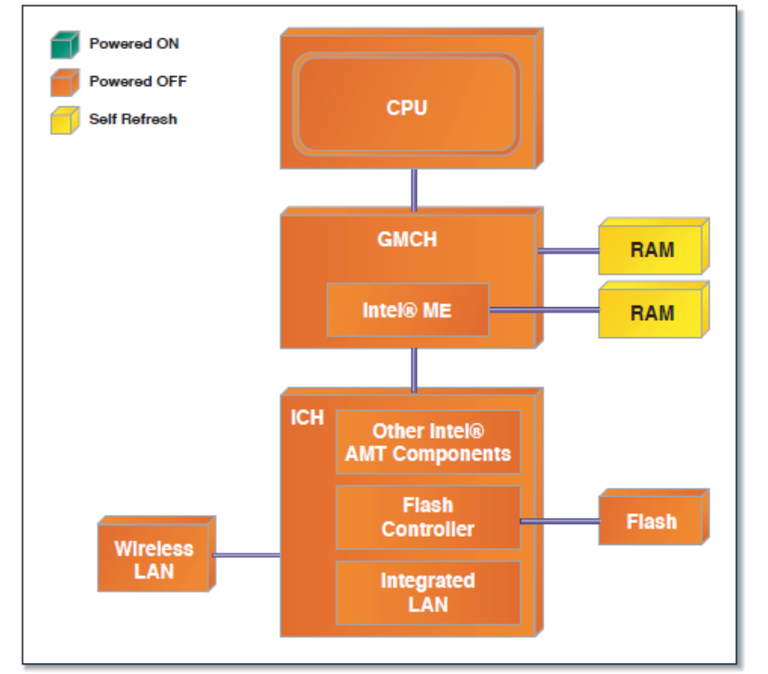


Figure 7.15 An Intel® AMT Computer in S3 and M-Off State

Active Platform Management Demystified: Unleashing the Power of Intel VPro Technology, 2009, Intel Press

M3: aggiunto nelle piattaforme più recenti

- Host in Sx (S3/S4/S5)
- ME alimentato tramite un'alimentazione separata
- Solo la SRAM interna è disponibile

VULNERABILITÀ

- Ring -3 Rootkit - 2009
- Zero-touch provisioning - 2010
- SA-00075 - 2017
- SA-00086 - 2018

POSSO DISABILITARLO?

Dalla pagina F.A.Q. di Libreboot:

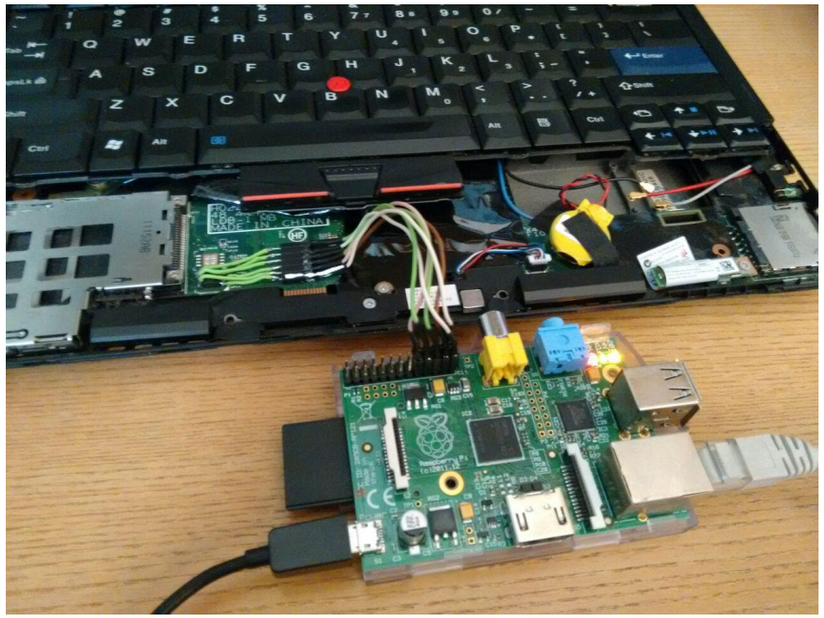
> *Before version 6.0 (that is, on systems from 2008/2009 and earlier), the ME can be disabled by setting a couple of values in the SPI flash memory. [...]*

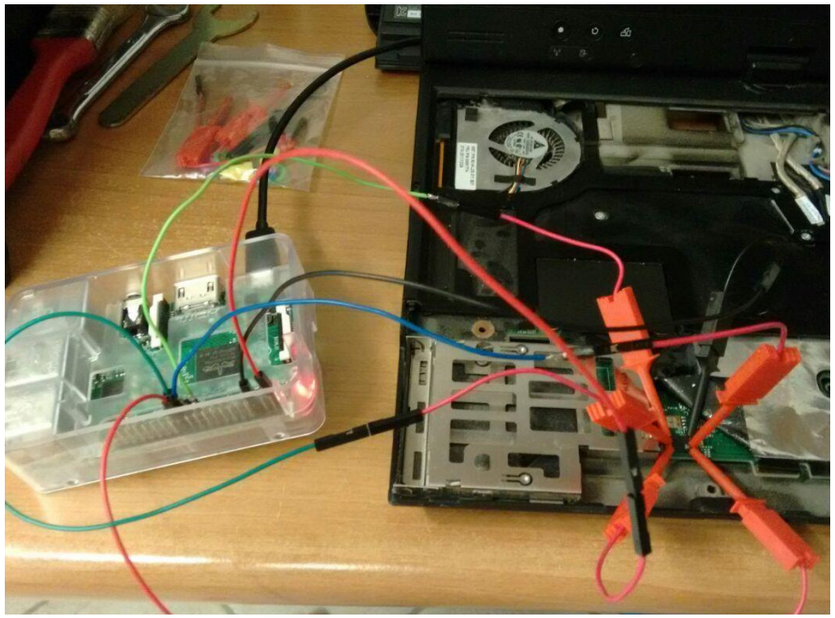
ME firmware versions 6.0 and later [...] include "ME Ignition" firmware that performs some hardware initialization and power management. If the ME's boot ROM does not find in the SPI flash memory an ME firmware manifest with a valid Intel signature, ***the whole PC will shut down after 30 minutes.***

FIRMWARE DI INTEL ME

Per poter analizzare e attaccare Intel ME una delle vie più semplici è tramite il suo firmware, contenuto nello stesso chip del BIOS/UEFI.

Leggerlo e scriverlo è abbastanza semplice con l'aiuto di una qualunque scheda Linux con un'interfaccia SPI master...

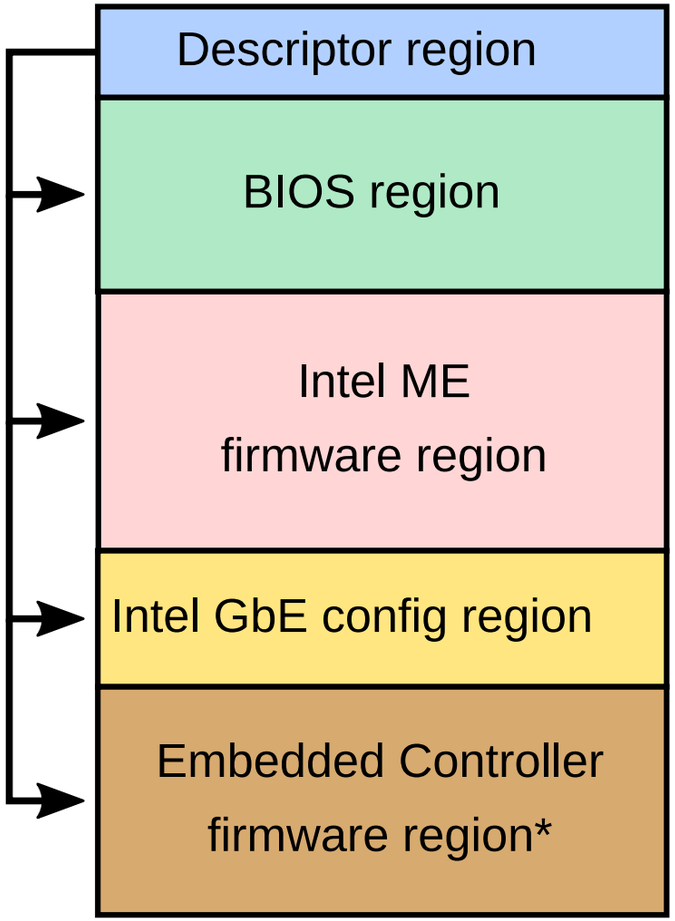




... e flashrom.

```
# flashrom -p linux_spi:dev=/dev/spidev0.0,spispeed=10000 -c W25Q64.V -r dump.bin

flashrom v0.9.9-r1955 on Linux 4.4.10-1-ARCH (armv7l)
flashrom is free software, get the source code at https://flashrom.org

Calibrating delay loop... OK.
Found Macronix flash chip "W25Q64.V" (8192 kB, SPI) on linux_spi.
Reading flash... done.
```

INTEL FLASH DESCRIPTOR (IFD)



*starting from Skylake

Queste regioni possono essere estratte con l'aiuto di ifdtool, parte del progetto coreboot.

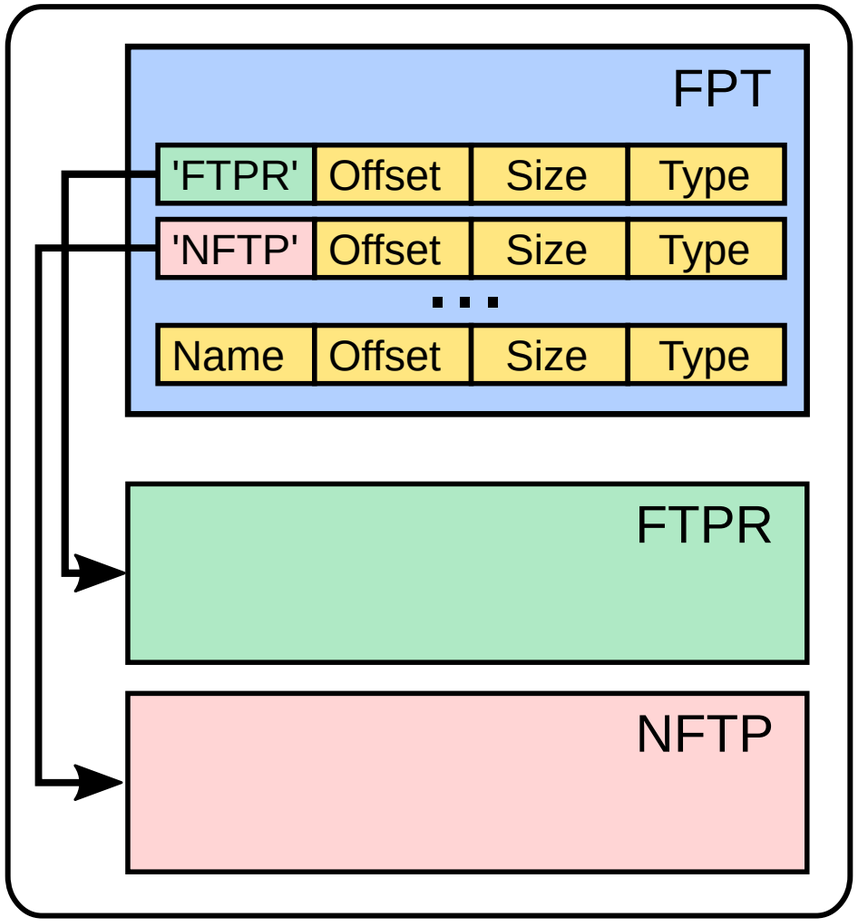
```
$ ifdtool -x dump.bin
File dump.bin is 8388608 bytes
  Flash Region 0 (Flash Descriptor): 00000000 - 00000fff
  Flash Region 1 (BIOS): 00500000 - 007fffff
  Flash Region 2 (Intel ME): 00003000 - 004fffff
  Flash Region 3 (GbE): 00001000 - 00002fff
  Flash Region 4 (Platform Data): 00fff000 - 00000fff (unused)

$ ls -lh flashregion_*
-rw-r--r-- 1 nicola nicola 4,0K mar 12 10:40 flashregion_0_flashdescriptor.bin
-rw-r--r-- 1 nicola nicola 3,0M mar 12 10:40 flashregion_1_bios.bin
-rw-r--r-- 1 nicola nicola 5,0M mar 12 10:40 flashregion_2_intel_me.bin
-rw-r--r-- 1 nicola nicola 8,0K mar 12 10:40 flashregion_3_gbe.bin
```

ME_CLEANER

STEP 1: RIMUOVERE (QUASI) TUTTE LE PARTIZIONI DI INTEL ME

Il primo step è stato provare a rimuovere tutte le partizioni dal firmware di Intel ME ad eccezione di FTPR, la partizione fondamentale.

FIRMWARE PARTITION TABLE (FPT)

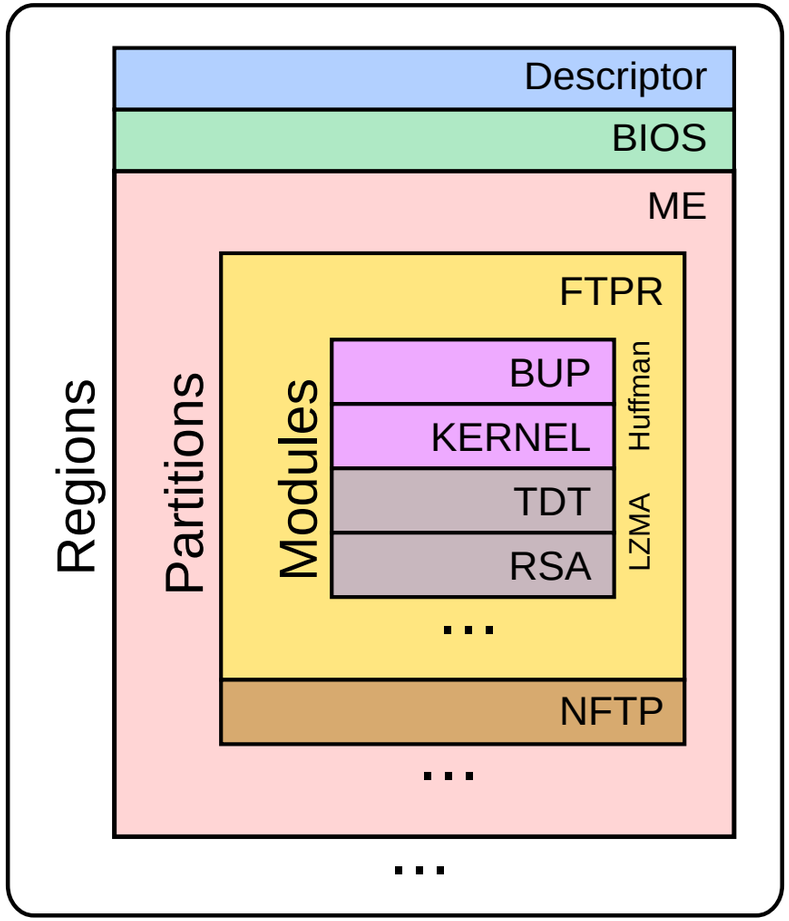


La rimozione è abbastanza semplice, grazie al fatto che:

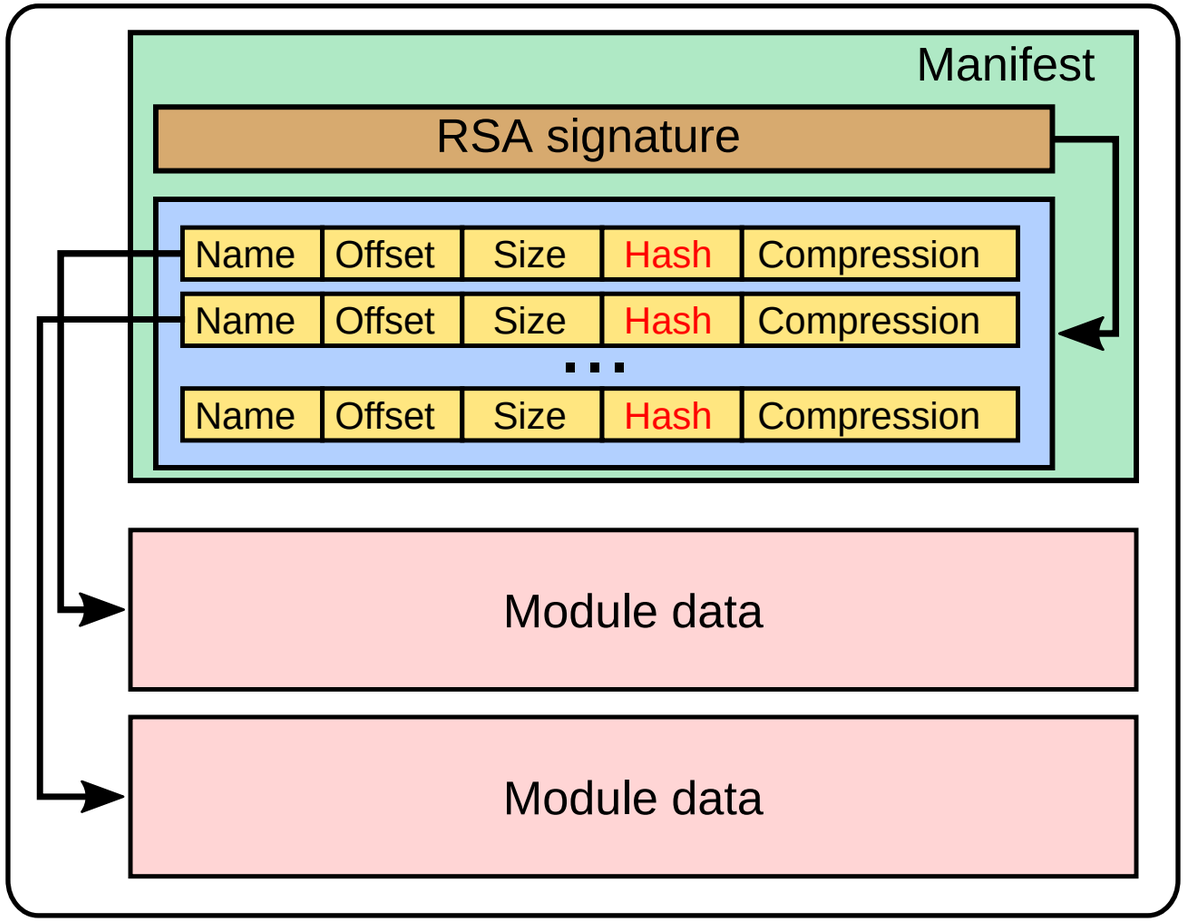
- L'FPT non è firmato (ha solo un checksum)
- Le partizioni sono firmate individualmente
- L'offset e la dimensione di ciascuna partizione sono salvati nell'entry FTP

STEP 2: RIMUOVERE TUTTI I MODULI LZMA

Regioni, partizioni, moduli...



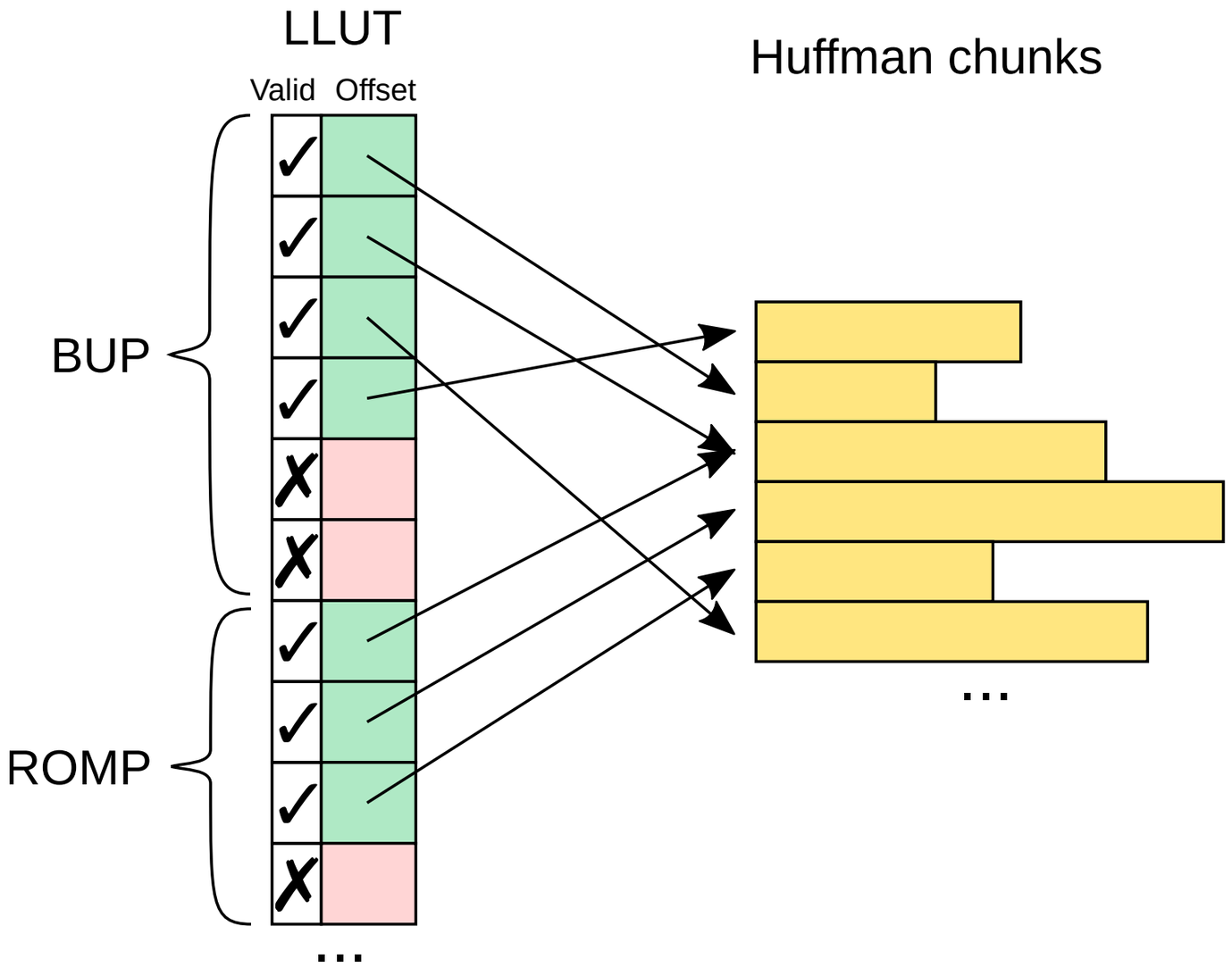
GENERAZIONE 2



Dopo aver rimosso tutti i moduli LZMA rimangono solo i 5 moduli Huffman: ROMP, BUP, KERNEL, POLICY e FTCS.

STEP 3: MODULI HUFFMAN

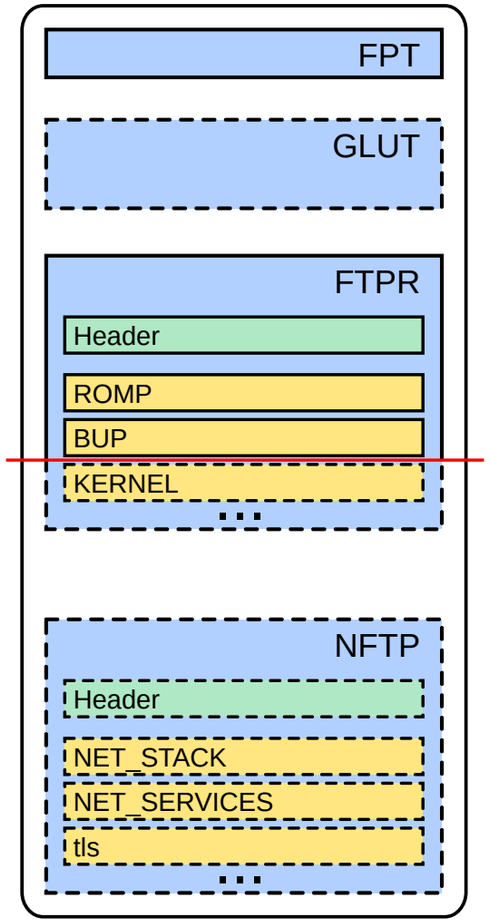
Con l'aiuto del codice sorgente di unhuffme è stato possibile capire la struttura dei moduli Huffman.


LLUT
Valid
Offset
Huffman chunks
BUP
ROMP
...
...

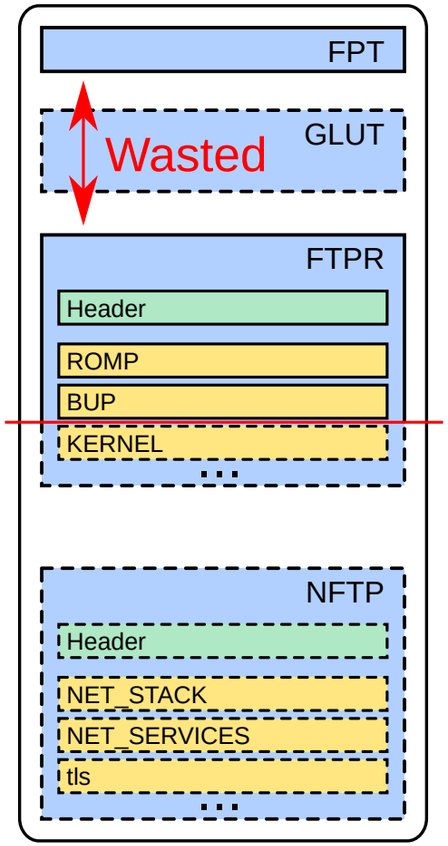
Tramite Trial&Error con l'aiuto di intelmetool è risultato che gli unici moduli realmente necessari per il boot sono:

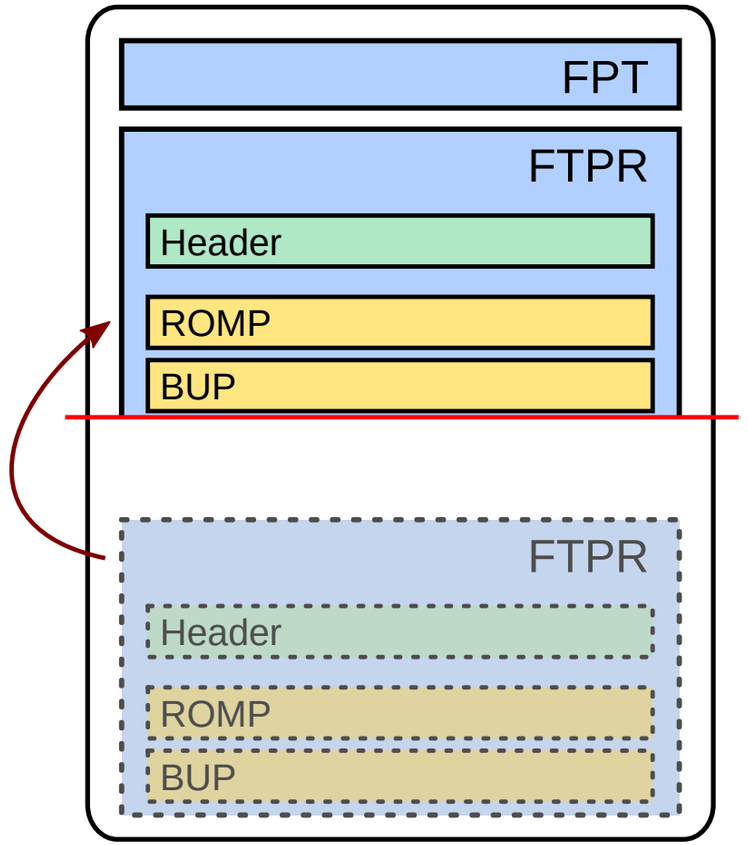
- BUP (BringUP, dove il watchdog dei 30 minuti viene disabilitato)
- ROMP (pochi kB, non sempre presente)

STEP 4: RECUPERARE LO SPAZIO LIBERO



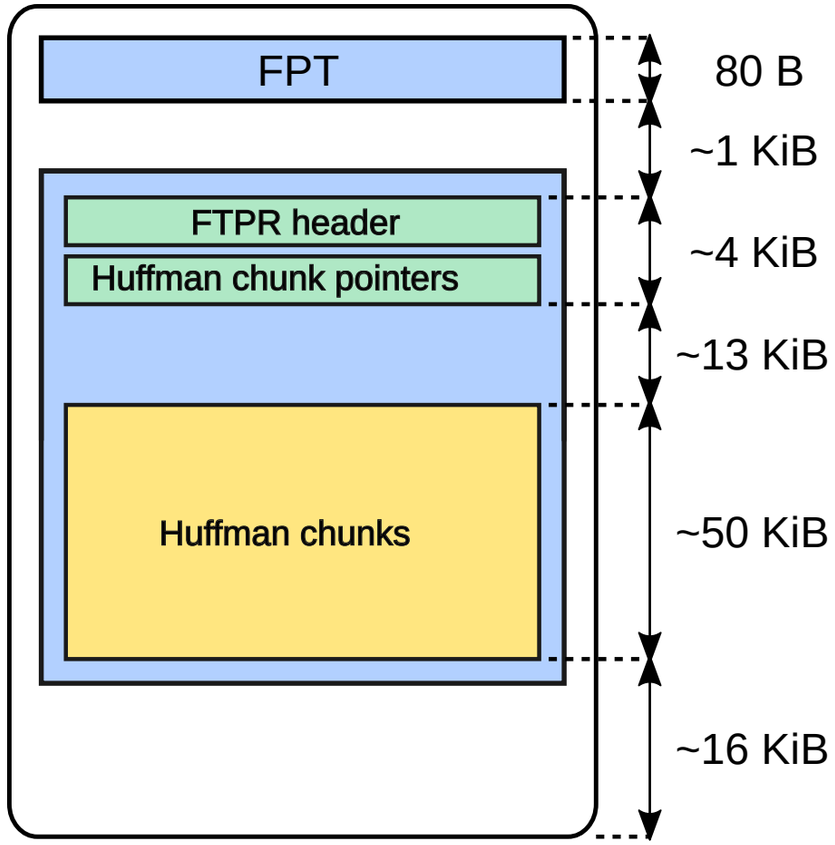
Sfortunatamente tra l'FPT e la partizione FTPR c'era molto spazio libero, sprecato.

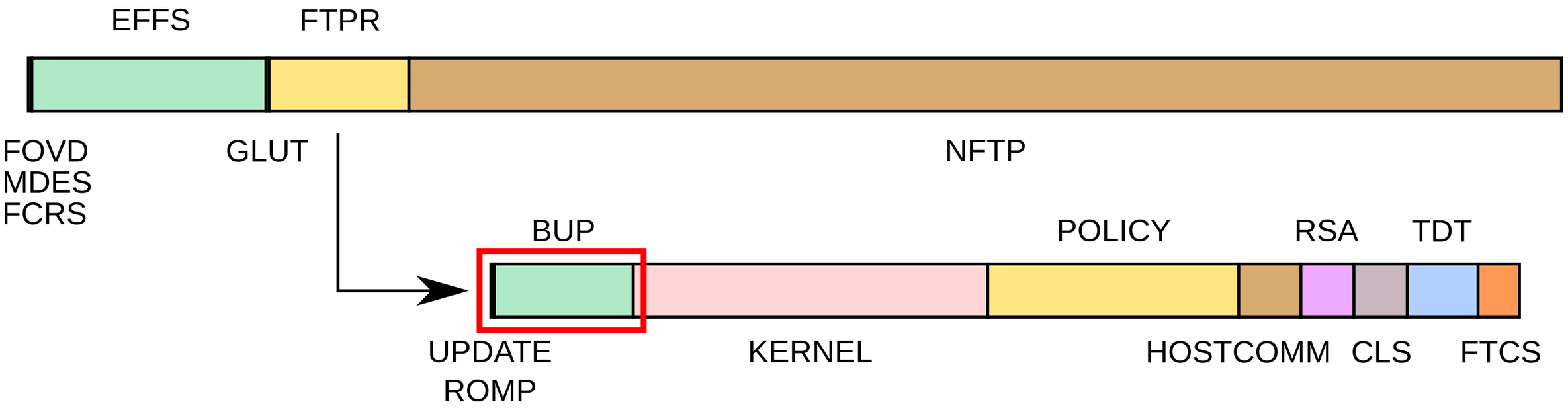



FPT
FTPR
Header
ROMP
BUP
FTPR
Header
ROMP
BUP

Il risultato finale è un firmware con solo la partizione FTPR contenente solo due (e a volte un) modulo, spostati all'indirizzo più basso possibile.

Il firmware di Intel ME, originariamente 5 MiB, ora è 84 KiB.

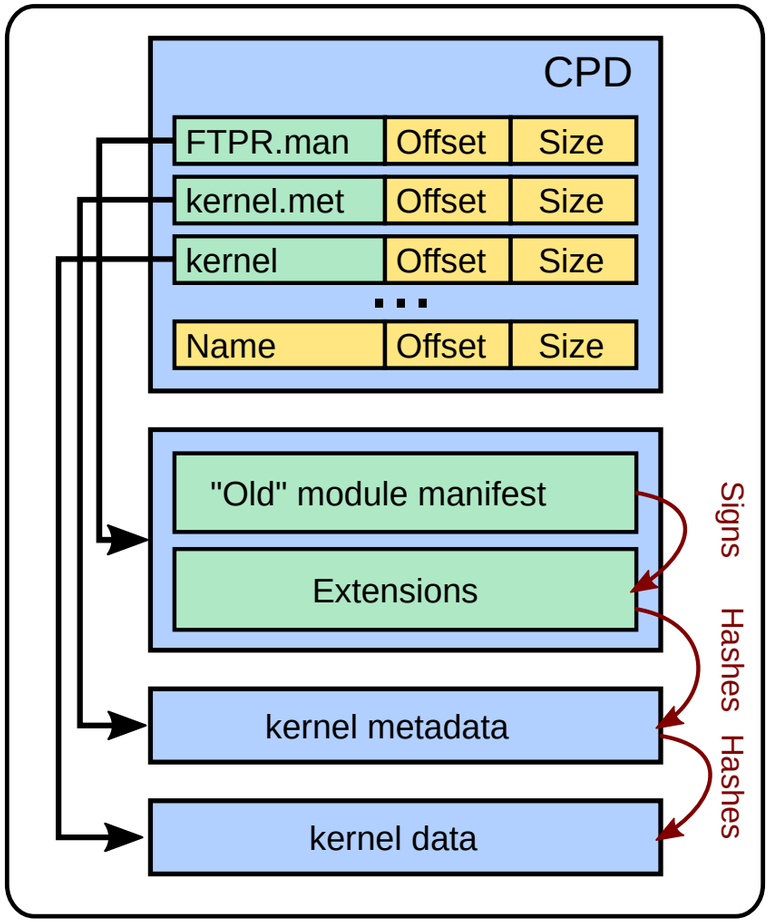



EFFS
FTPR
FOVD
MDES
FCRS
GLUT
NFTP
BUP
POLICY
RSA
TDT
UPDATE
ROMP
KERNEL
HOSTCOMM
CLS
FTCS

GENERAZIONE 3

Lo schema delle partizioni è lo stesso nella terza generazione, quindi è possibile rimuovere tutte le partizioni (ad eccezione dell'FTPR) senza alcuna modifica a me_cleaner.

CODE PARTITION DIRECTORY (CPD)



Dopo alcuni test è risultato che i moduli fondamentali, necessari per poter accendere correttamente il PC, sono:

- syslib
- rbe
- kernel
- bup

KILL-SWITCH

Ad Agosto Positive Technologies ha scoperto che è possibile disattivare Intel ME tramite un kill-switch.

[...] the company [Intel] provides hardware manufacturers with special software, including utilities such as Flash Image Tool (FIT) for configuring ME parameters [...]

From these utilities, you can extract a large number of XML file [...] These files contain a lot of interesting information: the structure of ME firmware and ***description of the PCH strap*** *[...]*

One of the fields, called "reserve_hap", drew our attention because there was a comment next to it: "High Assurance Platform (HAP) enable".

Googling did not take long. The second search result said that the name belongs to a ***trusted platform program linked to the U.S. National Security Agency (NSA)****.*

Our first impulse was to set this bit and see what happens. [...]

After the platform is loaded, the MEInfo utility reports a strange status: "Alt Disable Mode." ***Quick checks showed that ME did not respond to commands or react to requests from the operating system.***

L'HAP bit è disponibile solo nella generazione 3: fortunatamente nella generazione 2 è disponibile il bit AltMEDisable (scoperto da Igor Skochinsky) che dorebbe avere lo stesso effetto.

```
<MeMdesAddr value="0x00" name="ME Debug SMBus Emergency Mode Address"
    help_text="SMBUS address used for ME Debug status writes."/>
<MeMdesEn value="false" name="ME Debug SMBus Emergency Mode Enable"
    help_text="Enable emergency ME Debug status writes over SMBus using
    the address set by ME Debug SMBus Address."/>
<AltMeDisable value="0" name="Reserved[7]" help_text="Reserved, set to '0'."/>
<SRAMZeroingMode value="false" name="ME FW SRAM Zeroing Mode"
    help_text="ME FW SRAM Zeroing Mode"/>
```

RISULTATO

La rimozione di gran parte del firmware di Intel ME riduce la superficie d'attacco e minimizza il codice eseguito da Intel ME.

L'HAP/AltMEDisable bit aumenta la compatibilità delle modifiche con i firmware commerciali.

me_cleaner automatizza tutto il processo di rimozione:

```
$ me_cleaner.py -S -O cleaned.bin dump.bin

Full image detected
The ME/TXE region goes from 0x3000 to 0x500000
Found FPT header at 0x3010
Found 15 partition(s)
Found FTPR header: FTPR partition spans from 0x93000 to 0x108000
ME/TXE firmware version 8.0.13.1502
Public key match: Intel ME, firmware versions 7.x.x.x, 8.x.x.x
Reading partitions list...
 ???? (0x000003c0 - 0x000000400, 0x00000040 total bytes): removed
 FOVD (0x00000400 - 0x000001000, 0x00000c00 total bytes): removed
 MDES (0x00001000 - 0x000002000, 0x00001000 total bytes): removed
 FCRS (0x00002000 - 0x000003000, 0x00001000 total bytes): removed
 EFFS (0x00003000 - 0x00004b000, 0x00048000 total bytes): removed
 NVCL (NVRAM partition, no data, 0x00010511 total bytes): nothing to remove
[...]
 NVTD (NVRAM partition, no data, 0x00001eac total bytes): nothing to remove
 GLUT (0x0004b000 - 0x00004d000, 0x00002000 total bytes): removed
 MDMV (0x0004d000 - 0x000093000, 0x00046000 total bytes): removed
 FTPR (0x00093000 - 0x000108000, 0x00075000 total bytes): NOT removed
 NFTP (0x00108000 - 0x00017d000, 0x00075000 total bytes): removed
[...]
```

```
[...]
Removing partition entries in FPT...
Removing EFFS presence flag...
Correcting checksum (0x05)...
Reading FTPR modules list...
 UPDATE           (LZMA   , 0x0deb1b - 0x0decd9       ): removed
 ROMP             (Huffman, fragmented data, ~2 KiB   ): NOT removed, essential
 BUP              (Huffman, fragmented data, ~54 KiB  ): NOT removed, essential
 KERNEL           (Huffman, fragmented data, ~135 KiB ): removed
 POLICY           (Huffman, fragmented data, ~89 KiB  ): removed
 HOSTCOMM         (LZMA   , 0x0decd9 - 0x0e59ff       ): removed
 RSA              (LZMA   , 0x0e59ff - 0x0eac71       ): removed
 CLS              (LZMA   , 0x0eac71 - 0x0f03ab       ): removed
 TDT              (LZMA   , 0x0f03ab - 0x0f6a8a       ): removed
 FTCS             (Huffman, fragmented data, ~18 KiB  ): removed
 ClsPriv          (LZMA   , 0x0f6a8a - 0x0f6e6c       ): removed
 SESSMGR          (LZMA   , 0x0f6e6c - 0x10575c       ): removed
[...]
```

```
[...]
Relocating FTPR from 0x93000 - 0x108000 to 0x4c0 - 0x754c0...
 Adjusting FPT entry...
 Adjusting LUT start offset...
 Adjusting Huffman start offset...
 Adjusting chunks offsets...
 Moving data...
The ME minimum size should be 94208 bytes (0x17000 bytes)
The ME region can be reduced up to:
 00003000:00019fff me
Setting the AltMeDisable bit in PCHSTRP10 to disable Intel ME...
Checking the FTPR RSA signature... VALID
Done! Good luck!
```

Tramite intelmetool possiamo ottenere lo stato attuale di Intel ME.

Con il firmware di Intel ME originale:

```
# intelmetool -m

 [...]
ME: FW Partition Table       : OK
ME: Bringup Loader Failure   : NO
ME: Firmware Init Complete   : YES
ME: Manufacturing Mode       : YES
ME: Boot Options Present     : NO
ME: Update In Progress       : NO
ME: Current Working State    : Normal
ME: Current Operation State  : M0 with UMA
ME: Current Operation Mode   : Normal
ME: Error Code               : No Error
ME: Progress Phase           : Host Communication
ME: Power Management Event   : Clean Moff->Mx wake
ME: Progress Phase State     : Host communication established
 [...]
```

Con un firmware di Intel ME modificato e l'AltMEDisable bit a 1:

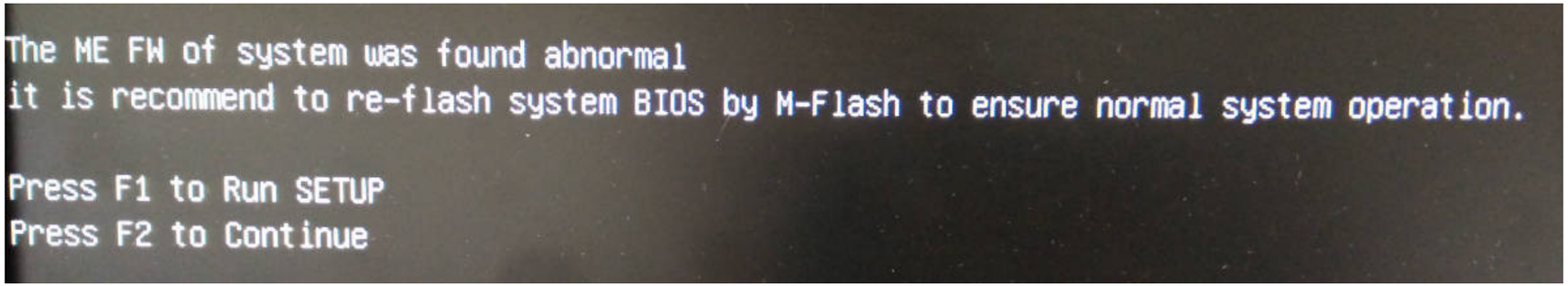
```
# intelmetool -m

 [...]
ME: FW Partition Table       : OK
ME: Bringup Loader Failure   : NO
ME: Firmware Init Complete   : NO
ME: Manufacturing Mode       : YES
ME: Boot Options Present     : NO
ME: Update In Progress       : NO
ME: Current Working State    : Initializing
ME: Current Operation State  : Bring up
ME: Current Operation Mode   : Debug
ME: Error Code               : No Error
ME: Progress Phase           : BUP Phase
ME: Power Management Event   : Clean Moff->Mx wake
ME: Progress Phase State     : Check to see if straps say ME DISABLED
 [...]
```

Alcune funzionalità sono perse:

- Overclocking (ICC)
- Intel AMT
- Intel PAVP (DRM)
- Parti di Intel SGX
- Altro...

A volte possono inoltre insorgere alcuni problemi:

- "Brick"
- Ritardo nel boot
- Rollback automatico delle modifiche introdotte da me_cleaner
- Warning all'avvio



HOWTO

Se volete testare me_cleaner sul vostro PC potete seguire le guide sulla pagina GitHub del progetto.

Q & A

THANK YOU!