Secure Boot in OpenPower Systems

Claudio Carvalho
cclaudio@br.ibm.com
IBM Linux Technology Center

Linux Developer Conference BR / August 3, 2019

Outline

- **Introduction**
  OpenPower Systems
  Secure Boot
  OpenPower Secure Boot Domains
- **Firmware Secure Boot**
- **OS Secure Boot**
- **Final Considerations**

Disclaimer

- This work represents the view of the author and does not necessarily represent the view of IBM
- All design points disclosed herein are subject to finalization and upstream acceptance. The features described may not ultimately exist or take the described form in a product
- IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

OpenPOWER (Ready) Systems

- *https://openpowerfoundation.org*
- OpenPOWER foundation is an open technical membership organization
  - *“Through the growing open ecosystem of the POWER Architecture and its associated technologies, the OpenPOWER Foundation facilitates its Members to share expertise, investment and intellectual property to serve the evolving needs of all end users.”*
- OpenPOWER Ready: mark to indicate that the product meets the minimum set of characteristics and should be interoperable with other OpenPOWER products
- OpenPOWER Ready Systems: e.g. S812LC, S822LC, LC921, LC922
  - Firmware must be a modest derivative of https://github.com/open-power/op-build
  - Linux on Power. Trusted Boot and Secure Boot OpenPOWER Ready
- OpenPOWER Ready Software, I/O Adapter, etc

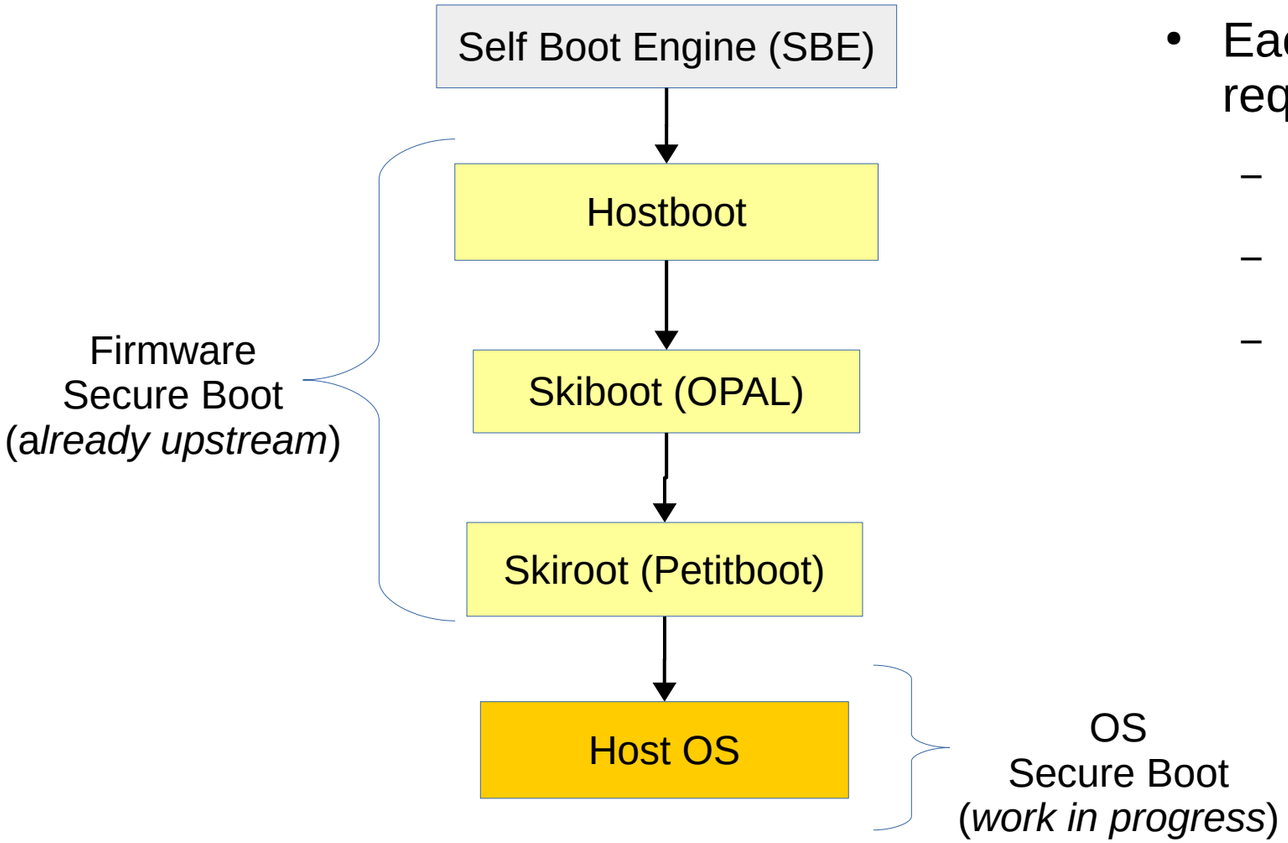
What is Secure Boot?

- Technology that aims to prevent untrusted code from loading during the platform boot.

- Uses cryptography functions to ensure that only code signed with trusted keys is started, otherwise the boot is aborted.

- Establishes a CHAIN OF TRUST from firmware up to the Operating System (OS)

OpenPower Secure Boot Domains

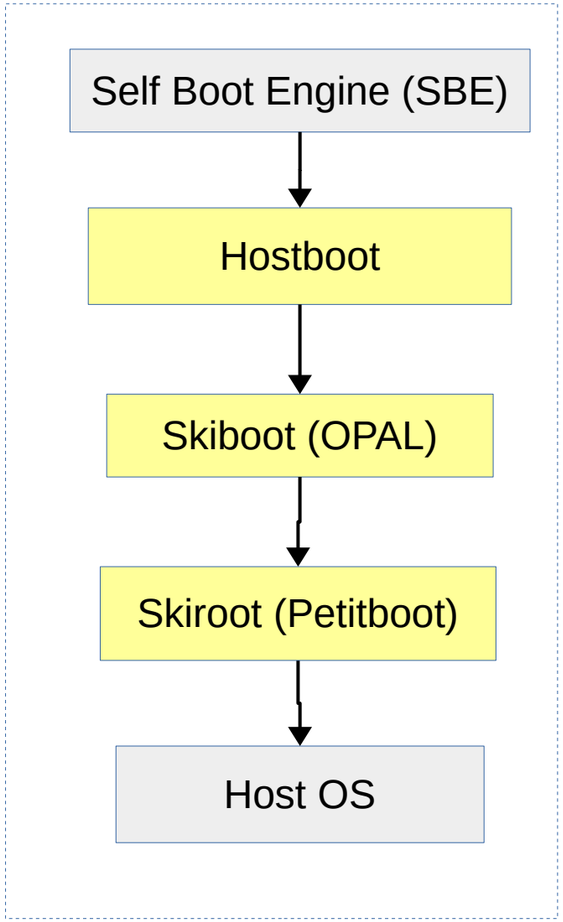


Simplified OpenPower Boot Flow [1]

- Each secure boot domain has its own requirements for:
  - Key management
  - Image signing
  - Image verification

[1] - https://github.com/open-power/docs/blob/master/hostboot/P9_Boot_Flow_OpenPOWER.pdf

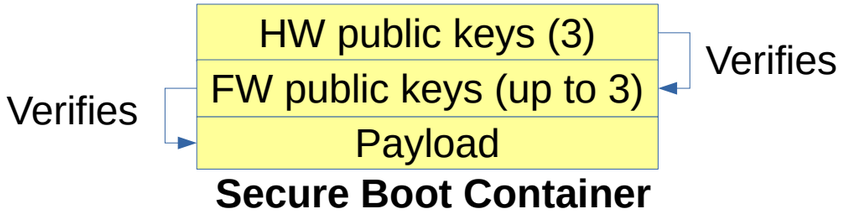
Firmware Secure Boot: Overview



Simplified OpenPower Boot Flow

- Stored in protected memory:
  - Root of trust: hardware public keys hash - SEEPROM
  - Secure Boot Container Verification code – OTPROM

- *op-build* builds firmware components and sign them following the secure boot container layout
- Firmware components are stored in a flash memory (PNOR – Processor NOR)
- FW secure boot is enabled by a hardware setting in the motherboard (platform dependent)



Secure Boot Container

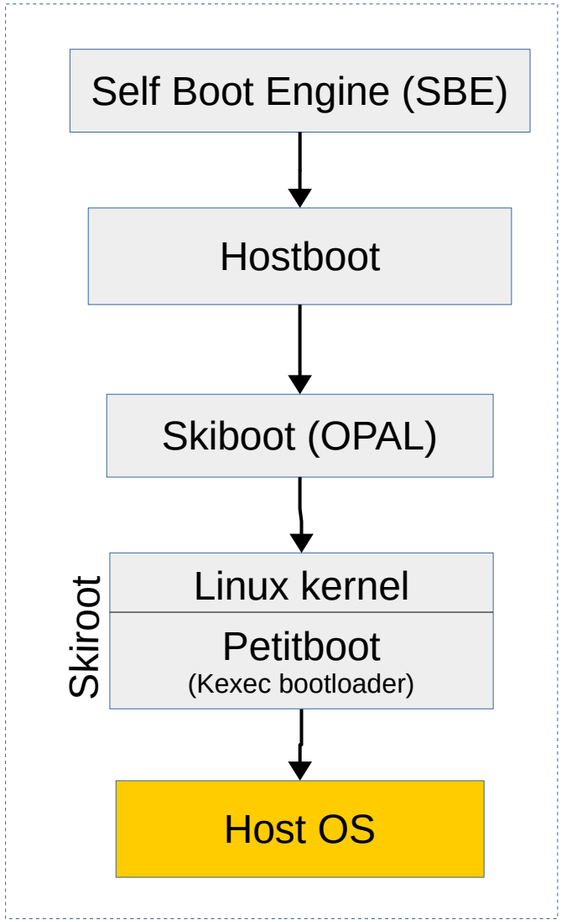
Firmware Secure Boot: Upstream Code

Secure mode disabled
Secure boot will not be enforced

This is the skiroot

```
[cclaudio@localhost ~]$ grep STB /sys/firmware/opal/msglog
[    69.056932895,3] STB: container NOT VERIFIED, resource_id=4 secureboot not yet initialized
[    69.256328750,5] STB: Found ibm,secureboot-v2
[    69.256387874,5] STB: secure mode off
[    69.256409780,6] STB: Found CVC @ 200ffd1d0000-200ffd1dffff
[    69.256411167,6] STB: Found CVC-sha512 @ 200ffd1d0040, version=1
[    69.256412497,6] STB: Found CVC-verify @ 200ffd1d0050, version=1
[    69.256431826,5] STB: Found tpm0,i2c_tpm_nuvoton evLogLen=2174 evLogSize=65536
[    69.383155960,5] STB: trusted mode on
[    70.511731190,5] STB: IMA_CATALOG verified
[    70.511936383,5] STB: IMA_CATALOG hash calculated
[    71.043208171,5] STB: IMA_CATALOG measured on pcr2 (tpm0, evType 0x5, evLogLen 2257)
[    71.383439064,5] STB: CAPP verified
[    71.383707310,5] STB: CAPP hash calculated
[    71.426871893,5] STB: CAPP measured on pcr2 (tpm0, evType 0x5, evLogLen 2333)
[    79.462183541,5] STB: BOOTKERNEL verified
[    79.492754100,5] STB: BOOTKERNEL hash calculated
[    80.024420917,5] STB: BOOTKERNEL measured on pcr4 (tpm0, evType 0x5, evLogLen 2415)
[    80.453220510,5] STB: EV_SEPARATOR measured on pcr0 (tpm0, evType 0x4, evLogLen 2491)
[    80.497174564,5] STB: EV_SEPARATOR measured on pcr1 (tpm0, evType 0x4, evLogLen 2567)
[    81.028419907,5] STB: EV_SEPARATOR measured on pcr2 (tpm0, evType 0x4, evLogLen 2643)
[    81.071664532,5] STB: EV_SEPARATOR measured on pcr3 (tpm0, evType 0x4, evLogLen 2719)
[    81.114942755,5] STB: EV_SEPARATOR measured on pcr4 (tpm0, evType 0x4, evLogLen 2795)
[    81.158264748,5] STB: EV_SEPARATOR measured on pcr5 (tpm0, evType 0x4, evLogLen 2871)
[    81.201673492,5] STB: EV_SEPARATOR measured on pcr6 (tpm0, evType 0x4, evLogLen 2947)
[    81.244920149,5] STB: EV_SEPARATOR measured on pcr7 (tpm0, evType 0x4, evLogLen 3023)
[cclaudio@localhost ~]$ lsprop /sys/firmware/devicetree/base/ibm,secureboot/
hw-key-hash-size 00000040 (64)
trusted-enabled
compatible              "ibm,secureboot-v2"
phandle                 000000b3 (179)
hw-key-hash             40d487ff 7380ed6a d54775d5 795fea0d
                        e2f541fe a9db06b8 466a42a3 20e65f75
                        b4866546 0017d907 515dc2a5 f9fc5095
                        4d6ee0c9 b67d219d fb708535 1d01d6d1
name                    "ibm,secureboot"
[cclaudio@localhost ~]$
```

OS Secure Boot: Overview



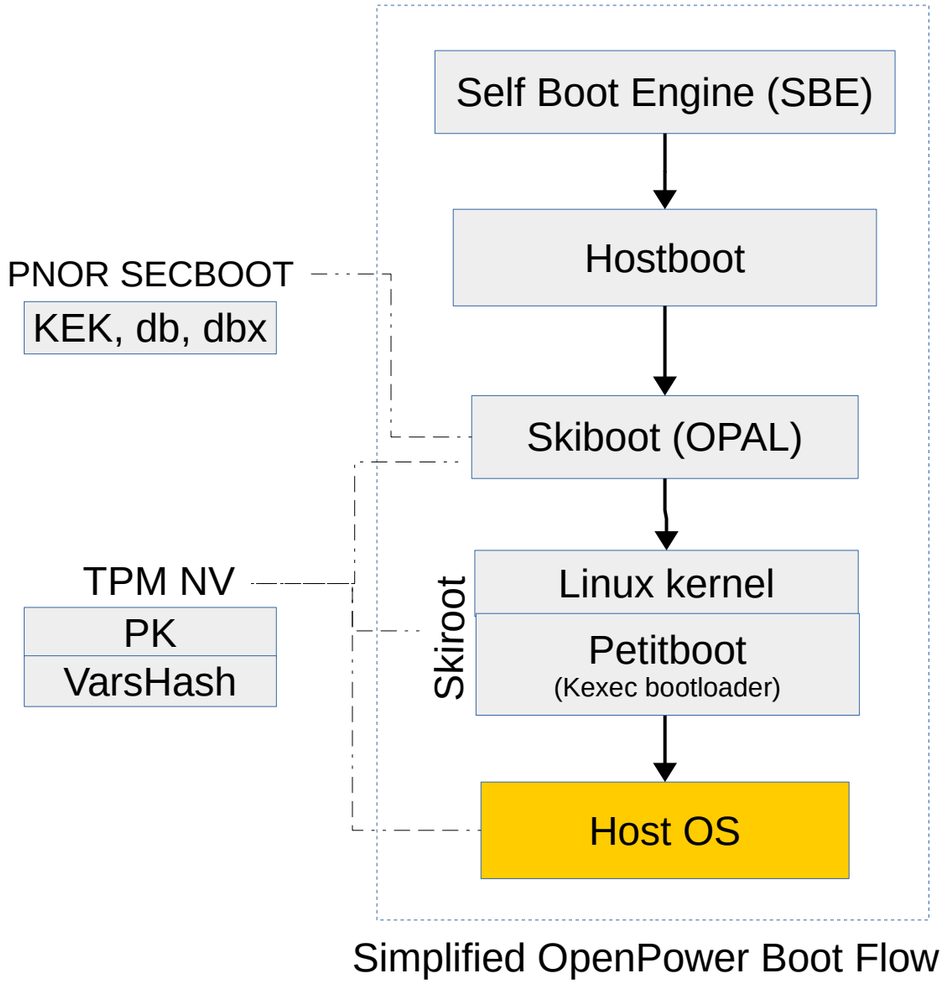
Simplified OpenPOWER Boot Flow

- OS Secure Boot: work-in-progress
- Skiroot is a linux kernel with embedded initramfs that runs Petitboot – a kexec bootloader.

Current design:

- OS kernel will be signed with *sign-file*, the same tool used to sign kernel modules. The signature is appended.
- OS kernel will be verified by IMA-appraisal.
- Key management:
  - Multiple OS kernels, multiple keys
  - Petitboot interface to manage keys
  - OS secure boot keys will be stored in PNOR and TPM (Trusted Platform Module)
- OS secure boot can be enabled only if FW secure boot is enabled

OS Secure Boot: Variables Policy

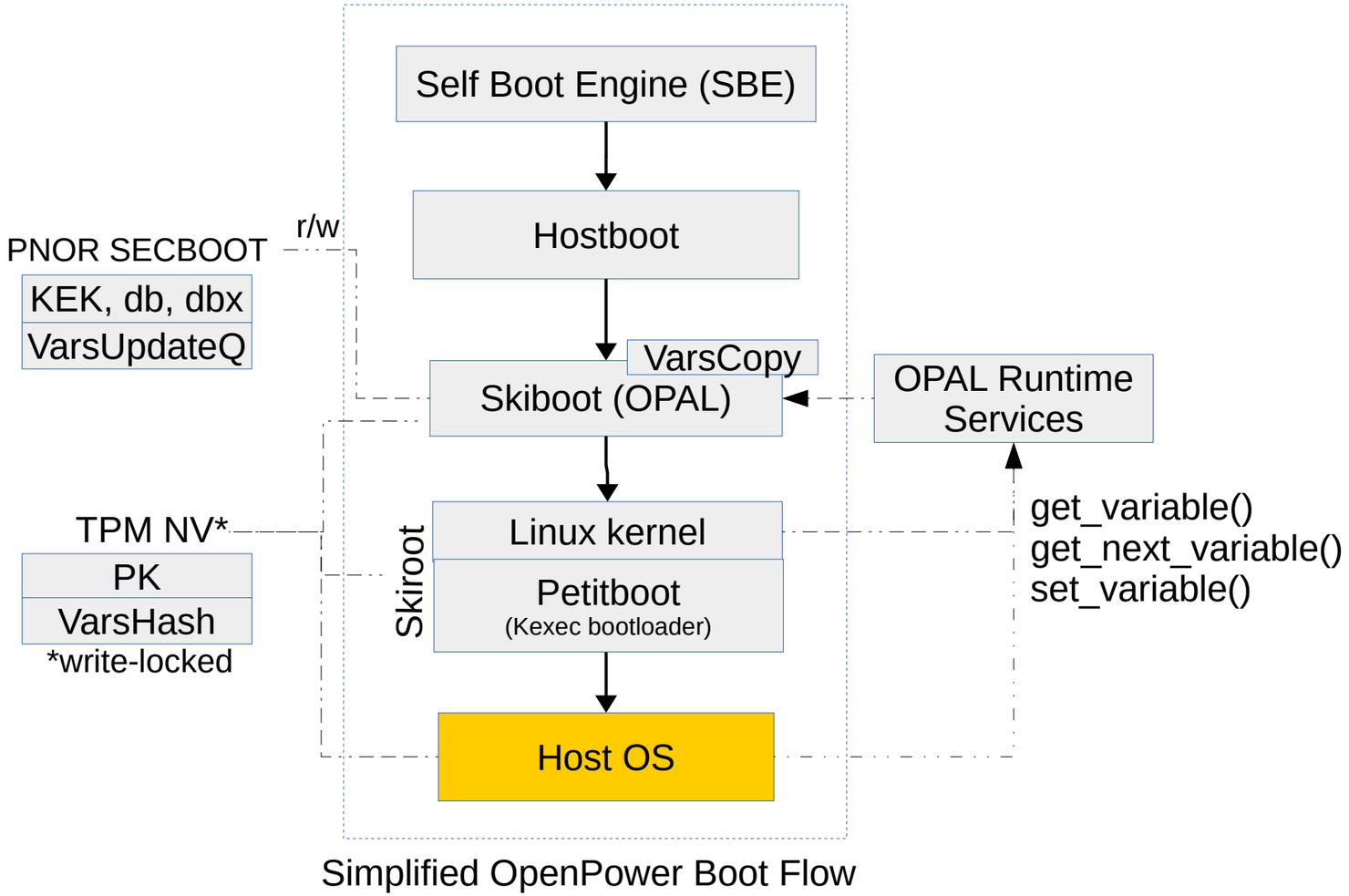


Simplified OpenPower Boot Flow

Current design:

- Secure boot variables: X.509 certificates
- Stored in the protected memory - TPM NVRAM:
  - Platform Key (PK)
    - Root of trust for the OS Secure Boot
    - When PK is set, OS Secure boot policy is enforced
  - SHA512 hash of the PNOR variables
- Stored in unprotected memory - PNOR:
  - Key Exchange Key (KEK)
  - Authorized Signature Database (db)
  - Forbidden Signature Database (dbx)

OS Secure Boot: Variables Update

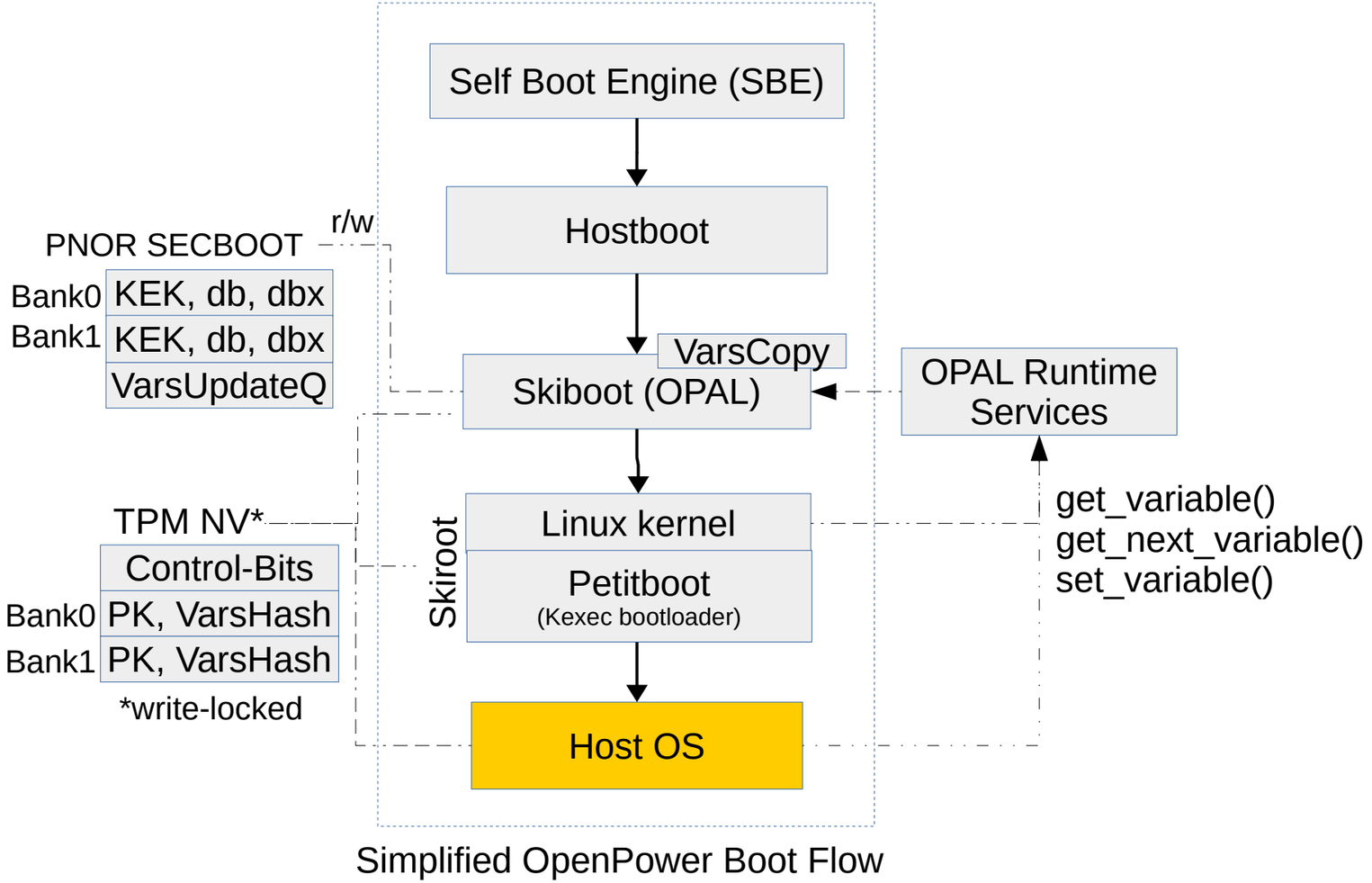


Simplified OpenPower Boot Flow

Current design:

- Skiboot checks the integrity of the variables and keeps an in-memory copy of them.
- Skiboot write locks the TPM NV secure boot indices at boot time until next boot.
- OPAL runtime services:
  - set_variable() enqueues signed variable updates that are processed only in the next boot by skiboot

OS Secure Boot: Atomic Variable Updates

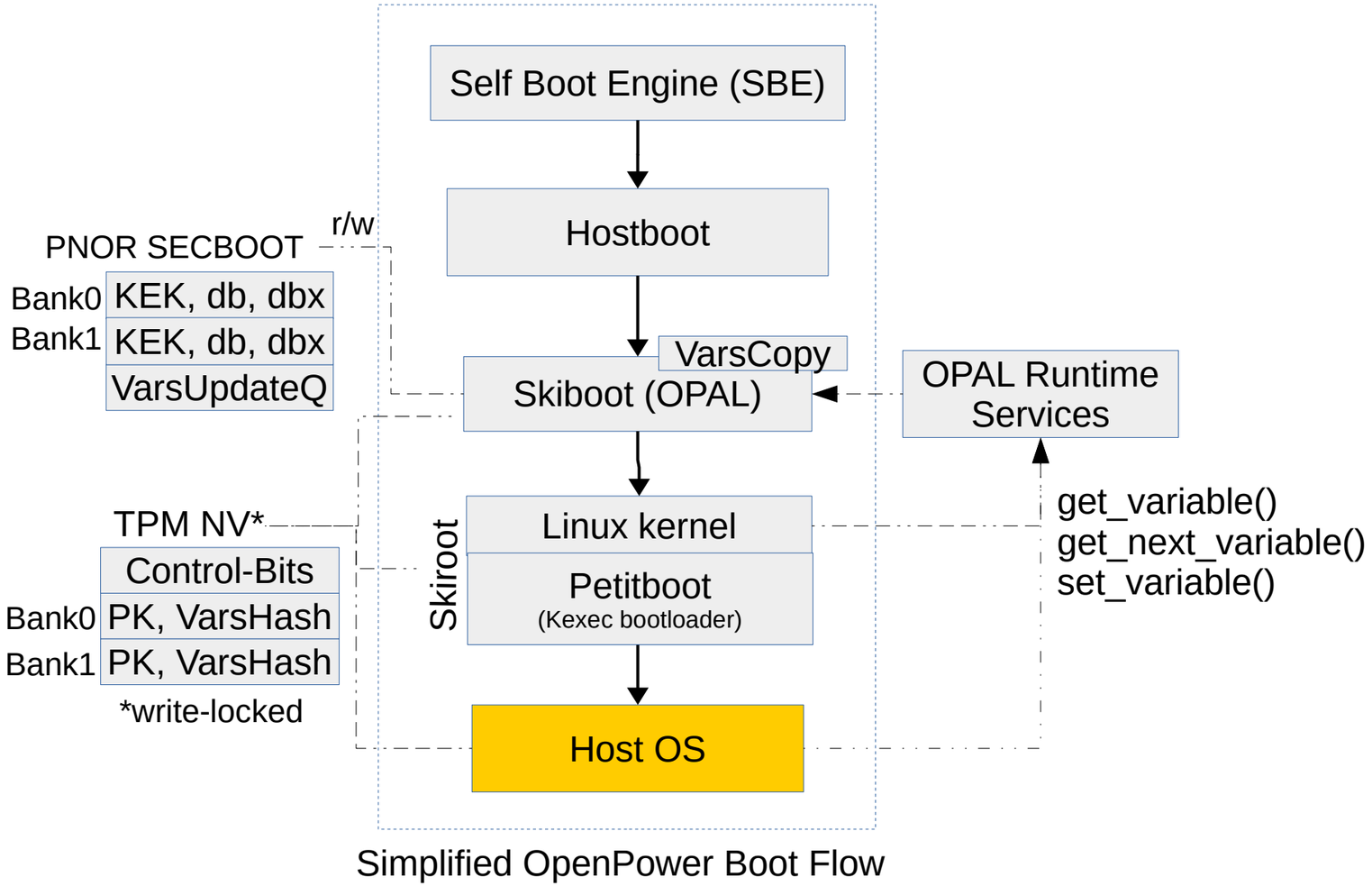


Simplified OpenPower Boot Flow

Current design:

- A variable update requires writes to multiple places
- Atomic update:
  - 2 variable banks
  - Active bank bit in the TPM Control-Bits
- At boot time, skiboot:
  1) Applies *VarsUpdateQueue* to *VarsCopy*
  2) Query TPM for staging bank
  3) Write updated VarsCopy to the staging bank
  4) Flip active bank bit in the TPM

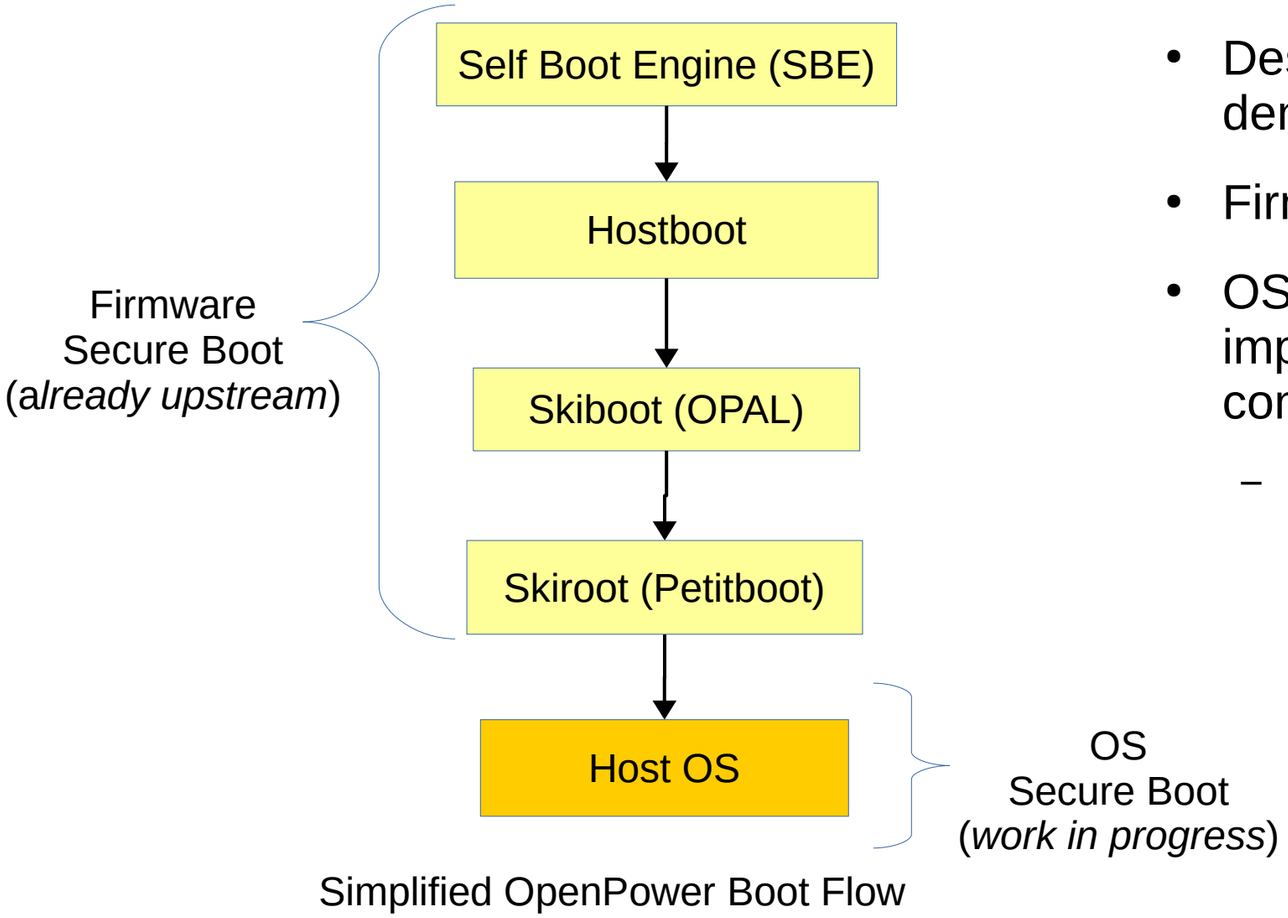
OS Secure Boot: Challenges



Simplified OpenPower Boot Flow

- Reuse of existing userspace tools for key management.
- Reuse of existing secure boot kernel code.
- Linux kernel:
  - Interface between skiboot and kernel.
  - Interface between kernel and userspace for key management.
  - Safely revoke keys.

Final Considerations



Simplified OpenPower Boot Flow

- Design and develop secure boot from scratch demands a big effort
- Firmware secure boot is working since Power8
- OS secure boot work is in-progress, being implemented and discussed with upstream communities
  - TPM NV has been shown a valuable resource

References

OpenPOWER Foundation
https://openpowerfoundation.org

OpenPOWER Firmware
https://github.com/open-power

POWER9 Boot Flow
https://github.com/open-power/docs/blob/master/hostboot/P9_Boot_Flow_OpenPOWER.pdf

Protecting System Firmware with OpenPOWER Secure Boot
https://www.ibm.com/developerworks/library/l-protect-system-firmware-openpower/index.html

Trusted Platform Module TCG Working Group
https://trustedcomputinggroup.org/work-groups/trusted-platform-module/

Using the TPM NVRAM to Protect Secure Boot Keys in OpenPower Systems
https://lssna18.sched.com/event/FLYK

Questions?

Thank you! Obrigado!

Claudio Carvalho
cclaudio@br.ibm.com
IBM Linux Technology Center

Backup Slides

Processor NOR (PNOR)

```
[cclaudio@rino Downloads]$ pflash -F image.pnor -i
Flash info:
-----------
Name         = /home/cclaudio/Downloads/image.pnor
Total size   = 64MB     Flags E:ECC, P:PRESERVED, R:READONLY, B:BACKUP
Erase granule =  0KB           F:REPROVISION, V:VOLATILE, C:CLEARECC

TOC@0x00000000 Partitions:
-----------
ID=00           part 0x00000000..0x00002000 (actual=0x00002000) [----R-----]
ID=01           HBEL 0x00008000..0x0002c000 (actual=0x00024000) [E-----F-C-]
ID=02          GUARD 0x0002c000..0x00031000 (actual=0x00005000) [E--P--F-C-]
ID=03          NVRAM 0x00031000..0x000c1000 (actual=0x00090000) [---P--F---]
ID=04        SECBOOT 0x000c1000..0x000e5000 (actual=0x00024000) [E--P------]
ID=05          DJVPD 0x000e5000..0x0012d000 (actual=0x00048000) [E--P--F-C-]
ID=06           MVPD 0x0012d000..0x001bd000 (actual=0x00090000) [E--P--F-C-]
ID=07           CVPD 0x001bd000..0x00205000 (actual=0x00048000) [E--P--F-C-]
ID=08            HBB 0x00205000..0x00305000 (actual=0x00100000) [EL--R-----]
ID=09            HBD 0x00305000..0x00425000 (actual=0x00120000) [EL--------]
ID=10            HBI 0x00425000..0x013e5000 (actual=0x00fc0000) [EL--R-----]
ID=11            SBE 0x013e5000..0x014a1000 (actual=0x000bc000) [ELI-R-----]
ID=12          HCODE 0x014a1000..0x015c1000 (actual=0x00120000) [EL--R-----]
ID=13           HBRT 0x015c1000..0x01bc1000 (actual=0x00600000) [EL--R-----]
ID=14        PAYLOAD 0x01bc1000..0x01cc1000 (actual=0x00100000) [-L--R-----]
ID=15     BOOTKERNEL 0x01cc1000..0x02bc1000 (actual=0x00f00000) [-L--R-----]
ID=16            OCC 0x02bc1000..0x02ce1000 (actual=0x00120000) [EL--R-----]
ID=17        FIRDATA 0x02ce1000..0x02ce4000 (actual=0x00003000) [E-----F-C-]
ID=18           CAPP 0x02ce4000..0x02d08000 (actual=0x00024000) [EL--R-----]
ID=19        BMC_INV 0x02d08000..0x02d11000 (actual=0x00009000) [------F---]
ID=20           HBBL 0x02d11000..0x02d18000 (actual=0x00007000) [EL--R-----]
ID=21       ATTR_TMP 0x02d18000..0x02d20000 (actual=0x00008000) [------F---]
ID=22      ATTR_PERM 0x02d20000..0x02d28000 (actual=0x00008000) [E-----F-C-]
ID=23        VERSION 0x02d28000..0x02d2a000 (actual=0x00002000) [-L--R-----]
ID=24    IMA_CATALOG 0x02d2a000..0x02d6a000 (actual=0x00040000) [EL--R-----]
ID=25        RINGOVD 0x02d6a000..0x02d8a000 (actual=0x00020000) [----------]
ID=26        WOFDATA 0x02d8a000..0x0308a000 (actual=0x00300000) [EL--R-----]
ID=27    HB_VOLATILE 0x0308a000..0x0308f000 (actual=0x00005000) [E-----F-CV]
ID=28           MEMD 0x0308f000..0x0309d000 (actual=0x0000e000) [EL--R-----]
ID=29           SBKT 0x0309d000..0x030a1000 (actual=0x00004000) [EL--R-----]
ID=30           HDAT 0x030a1000..0x030a9000 (actual=0x00008000) [EL--R-----]
ID=31         UVISOR 0x030a9000..0x031a9000 (actual=0x00100000) [-L--R-----]
ID=32         OCMBFW 0x031a9000..0x031f4000 (actual=0x0004b000) [EL--R-----]
ID=33       UVBWLIST 0x031f4000..0x03204000 (actual=0x00010000) [-L--R-----]
ID=34    BACKUP_PART 0x03ff7000..0x03fff000 (actual=0x00000000) [----RB----]
```

Last Slide