ORACLE

DRTM on AMD Server Platforms

Alec Brown, Jag Raman & Ross Philipson
Oracle Linux
Feb 4, 2024

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

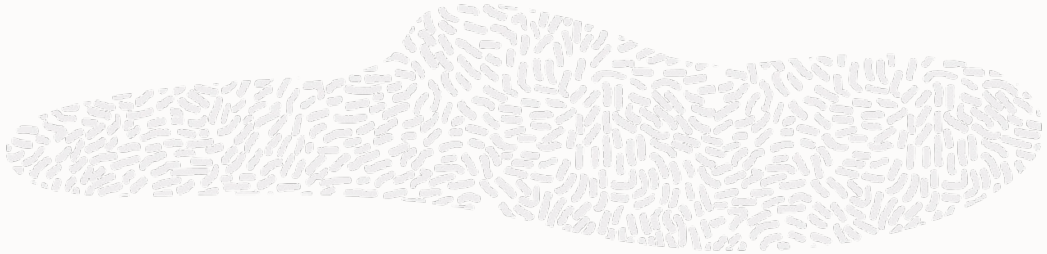
AMD’s statement

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc.

AMD MAKES NO REPRESENTATION OR WARRANTIES WITH RESEPCT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILTIES FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Agenda

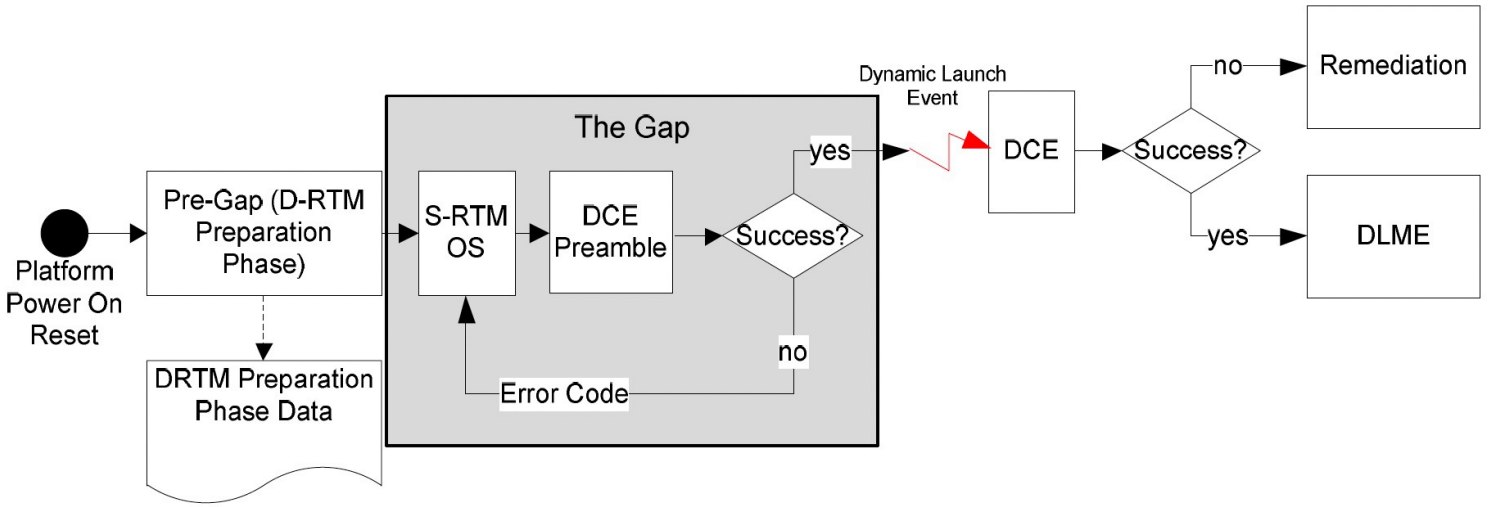


- Dynamic Root of Trust Measurement (DRTM)
- DRTM present status
- DRTM: Intel vs AMD
- DRTM with AMD's ASP
- GRUB
- Secure Kernel Loader (SKL)
- Secure Launch Kernel
- Linux Upstream Status
- Questions

Dynamic Root of Trust Measurement (D-RTM)

TCG Specification

- Trusted Computing Group (**TCG**) defines the broad requirements for D-RTM
- "*A platform-dependent function that initializes the state of the platform and provides a new instance of a root of trust for measurement without rebooting the platform. The initial state establishes a minimal Trusted Computing Base.*"



Nominal Timeline defined by the TCG

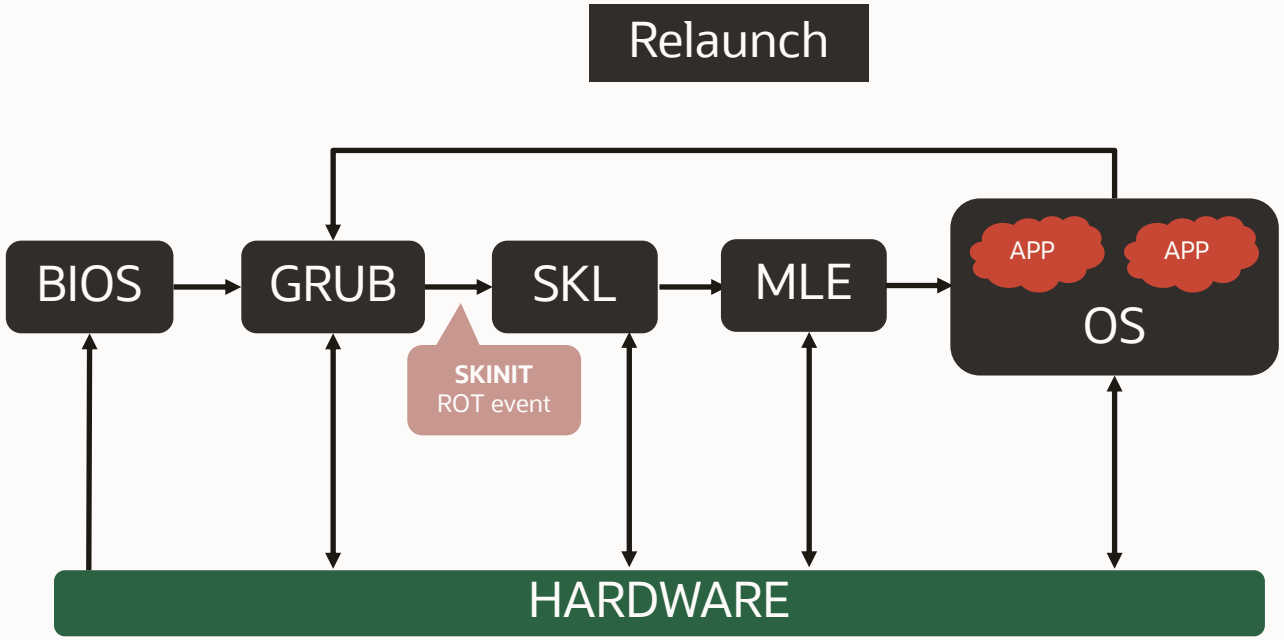
- ***The Gap***, a period in time where we haven't validated the Computed Base.
- Dynamic Launch Event: DRTM starts recording measurements into the PCR after the ***DL Event***
- If successful, DRTM presents the user with a Dynamically Launched Measured Environment (**DLME**).

Dynamic Root of Trust Measurement (DRTM)

TrenchBoot

- The location of the ROT is dynamic in DRTM; it is flexible, and the security architecture can choose where ROT lies.
- In the AMD platforms, the ROT starts with executing the **SKINIT** instruction.
- For TrenchBoot project on AMD, GRUB launches the DRTM sequence by executing the SKINIT instruction.
- The SKINIT instruction verifies the Signature of the SKL.

TrenchBoot with AMD's HW



SKL: Secure Kernel Loader

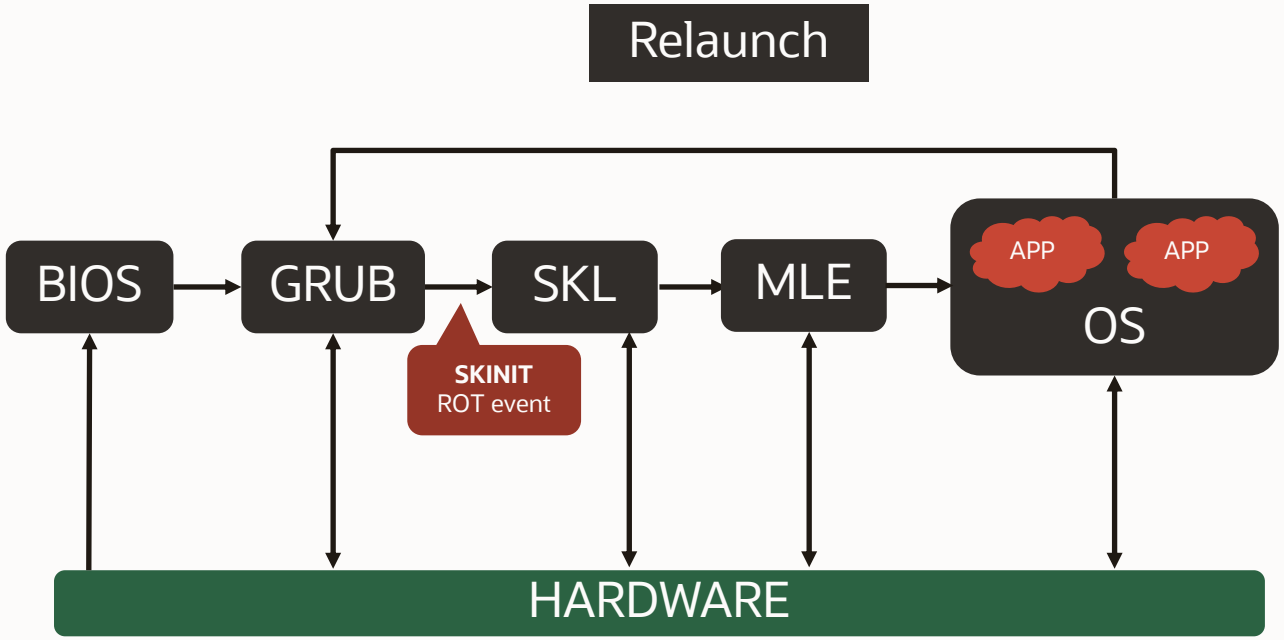
MLE: Measured Launch Environment

Dynamic Root of Trust Measurement (DRTM)

TrenchBoot

- The location of the ROT is dynamic in DRTM; it is flexible, and the security architecture can choose where ROT lies.
- In the AMD platforms, the ROT starts with executing the **SKINIT** instruction.
- For TrenchBoot project on AMD, GRUB launches the DRTM sequence by executing the SKINIT instruction.
- The SKINIT instruction verifies the Signature of the SKL.

TrenchBoot with AMD's HW



SKL: Secure Kernel Loader

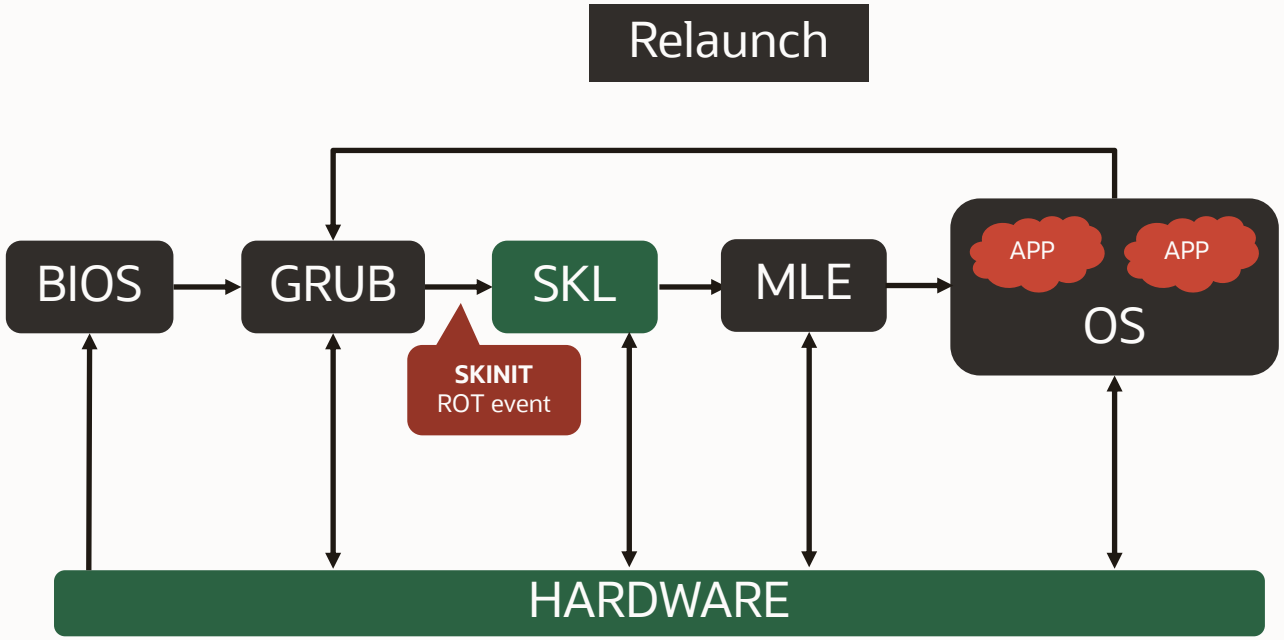
MLE: Measured Launch Environment

Dynamic Root of Trust Measurement (DRTM)

TrenchBoot

- The location of the ROT is dynamic in DRTM; it is flexible, and the security architecture can choose where ROT lies.
- In the AMD platforms, the ROT starts with executing the **SKINIT** instruction.
- For TrenchBoot project on AMD, GRUB launches the DRTM sequence by executing the SKINIT instruction.
- The SKINIT instruction verifies the Signature of the SKL.

TrenchBoot with AMD's HW



SKL: Secure Kernel Loader

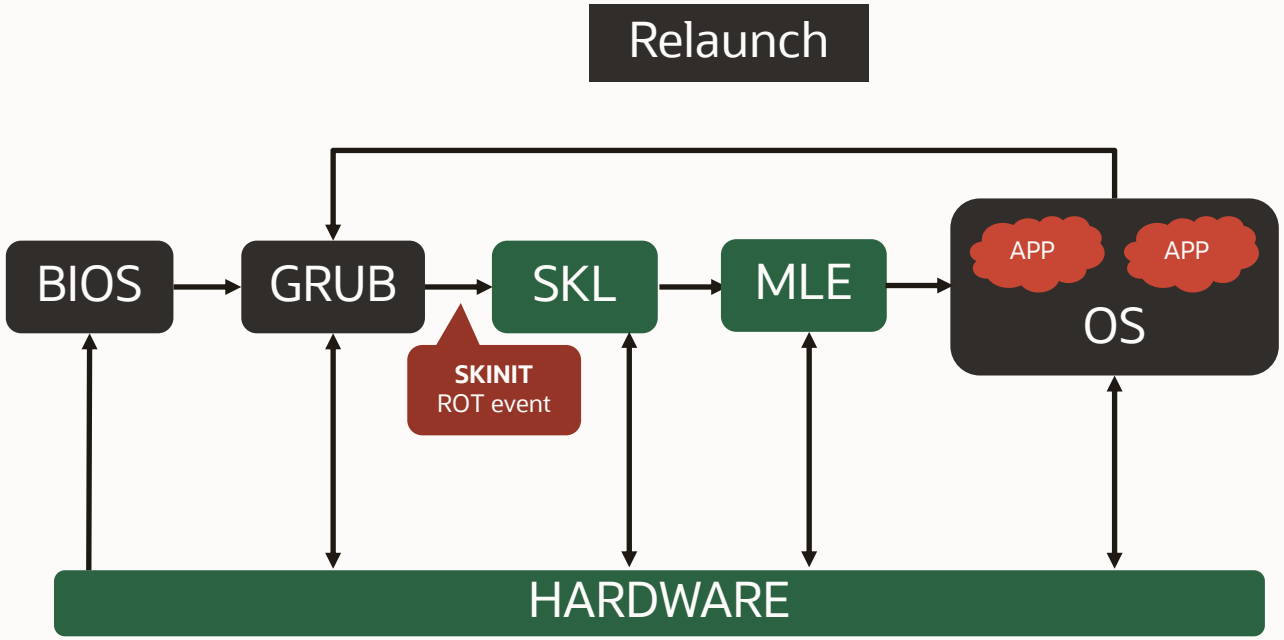
MLE: Measured Launch Environment

Dynamic Root of Trust Measurement (DRTM)

TrenchBoot

- The location of the ROT is dynamic in DRTM; it is flexible, and the security architecture can choose where ROT lies.
- In the AMD platforms, the ROT starts with executing the **SKINIT** instruction.
- For TrenchBoot project on AMD, GRUB launches the DRTM sequence by executing the SKINIT instruction.
- The SKINIT instruction verifies the Signature of the SKL.

TrenchBoot with AMD's HW



SKL: Secure Kernel Loader

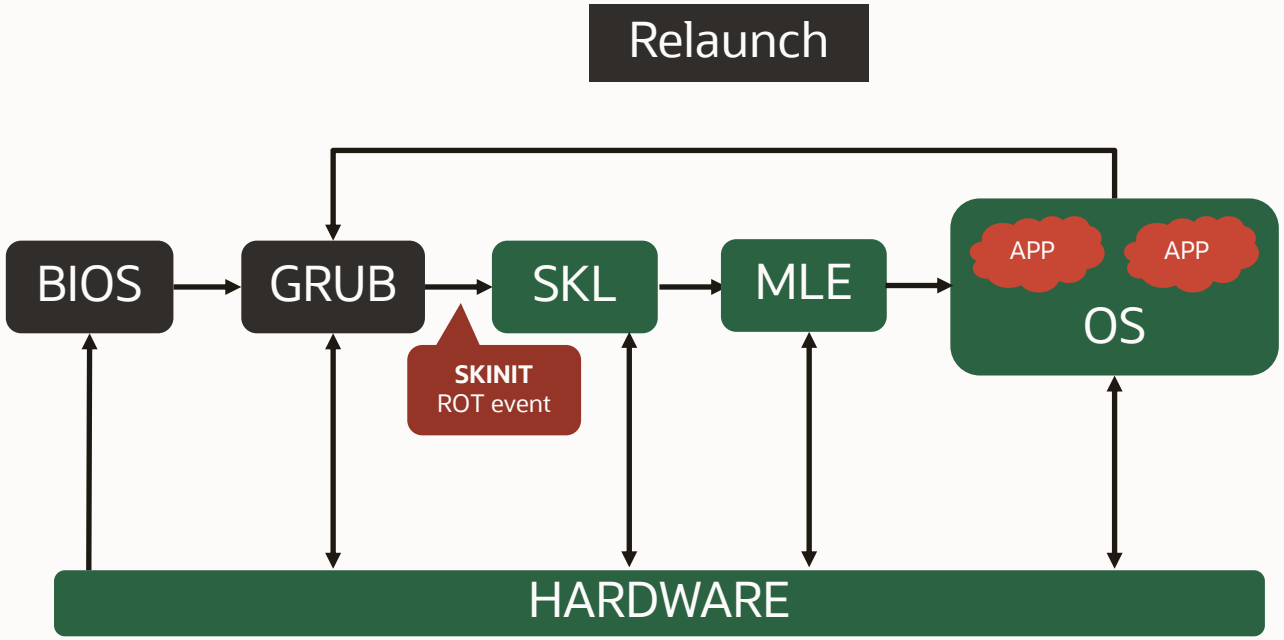
MLE: Measured Launch Environment

Dynamic Root of Trust Measurement (DRTM)

TrenchBoot

- The location of the ROT is dynamic in DRTM; it is flexible, and the security architecture can choose where ROT lies.
- In the AMD platforms, the ROT starts with executing the **SKINIT** instruction.
- For TrenchBoot project on AMD, GRUB launches the DRTM sequence by executing the SKINIT instruction.
- The SKINIT instruction verifies the Signature of the SKL.

TrenchBoot with AMD's HW

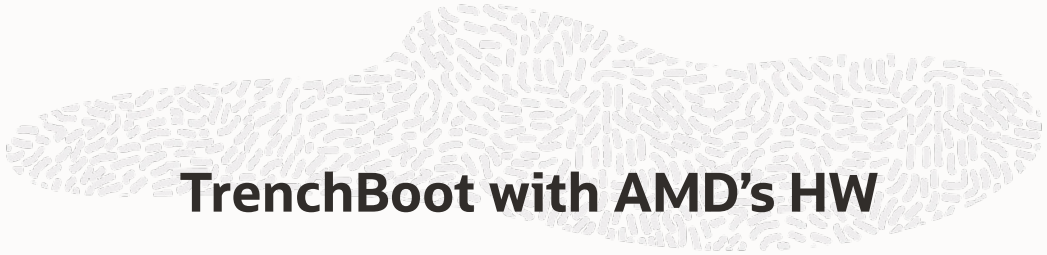


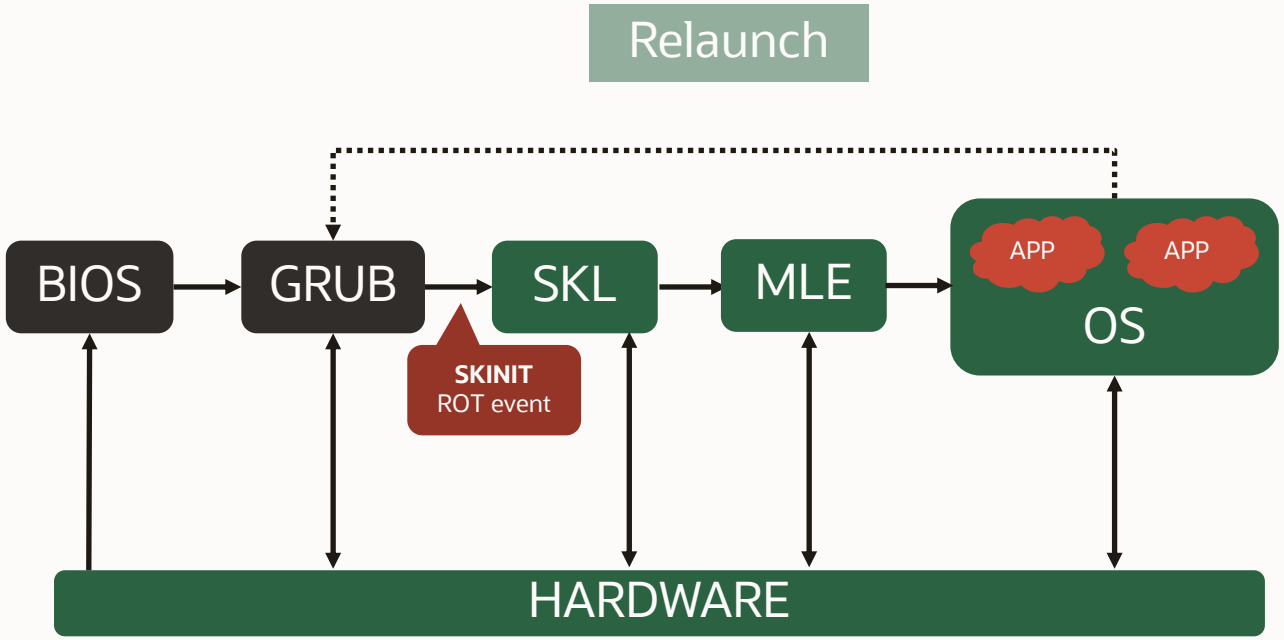
SKL: Secure Kernel Loader

MLE: Measured Launch Environment

DRTM present status

- DRTM comprises two phases: **Power-On** and **Relaunch**.
- The Power-on phase interfaces with AMD's **ASP** hardware; it authenticates and measures software modules.
- The Relaunch phase involves saving and restoring the user state and vectoring back to the GRUB. **Ross Philipson** is working on it.
- The Power-on phase is now available.

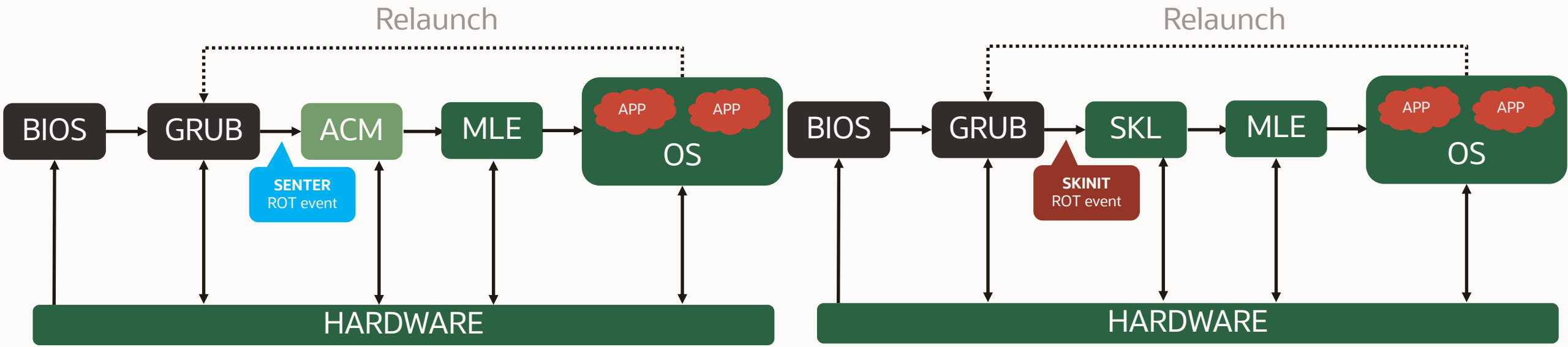




SKL: Secure Kernel Loader

MLE: Measured Launch Environment

DRTM: Intel vs AMD

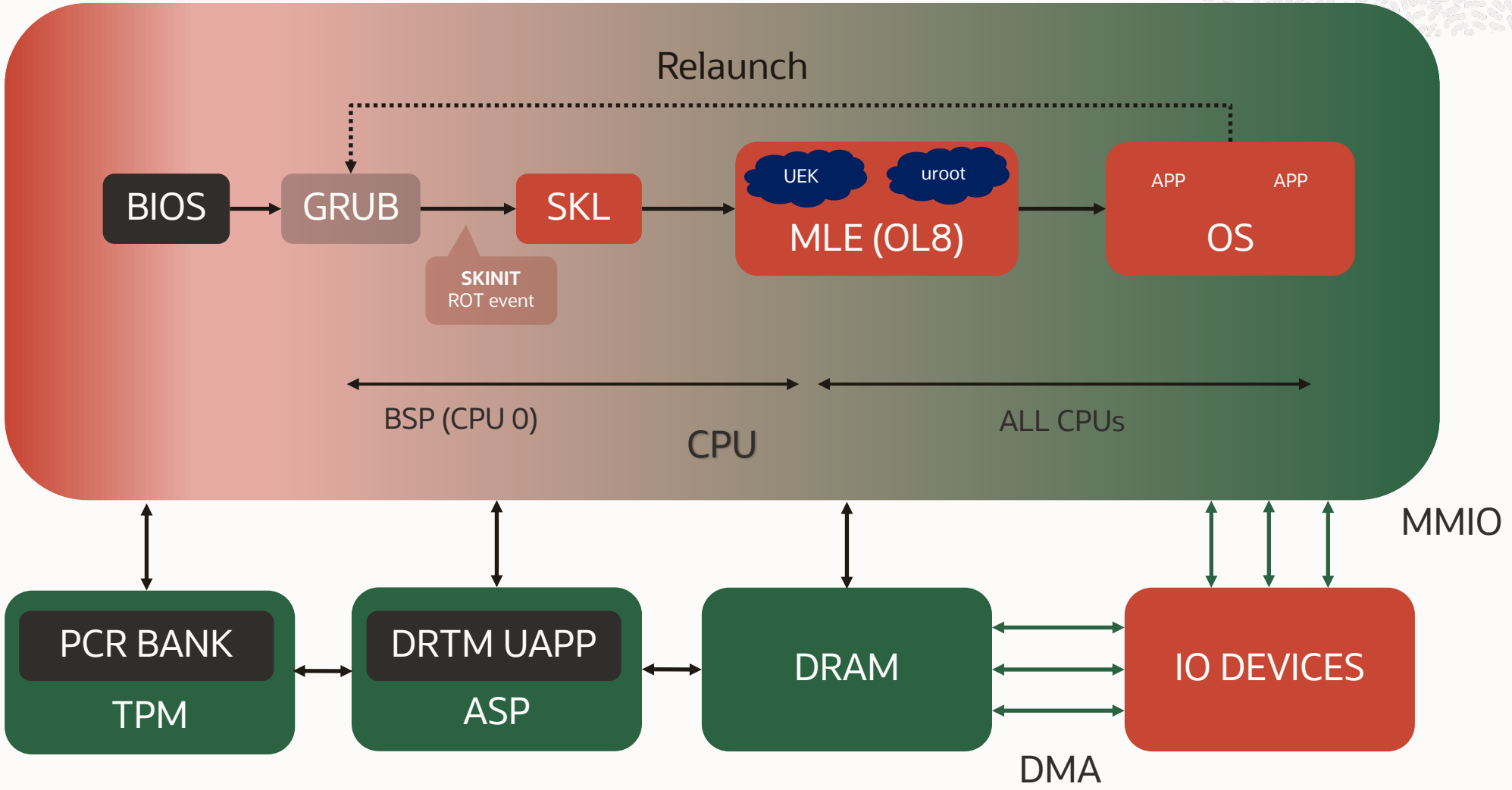


ACM: Authenticated Code Module

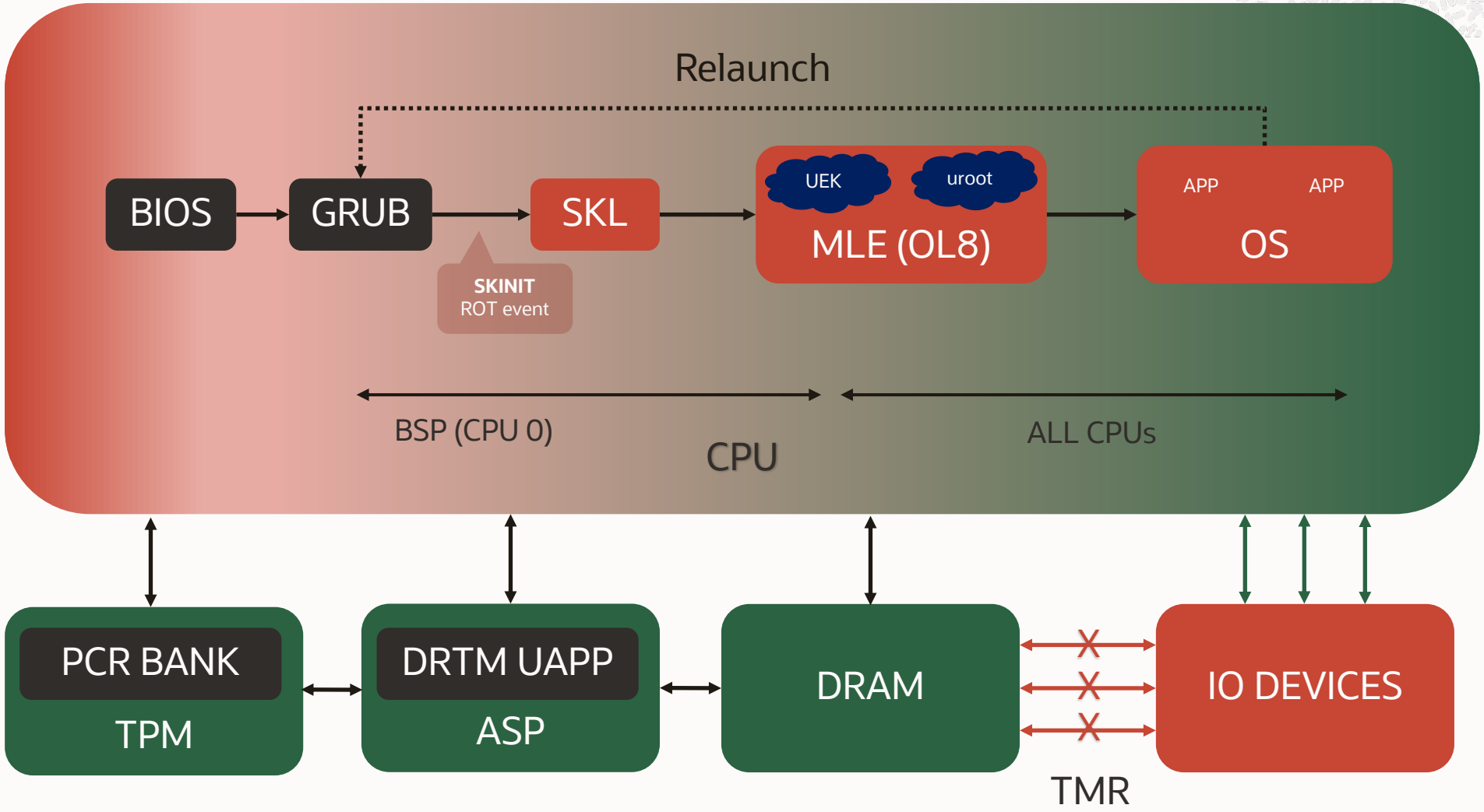
SKL: Secure Kernel Loader

MLE: Measured Launch Environment

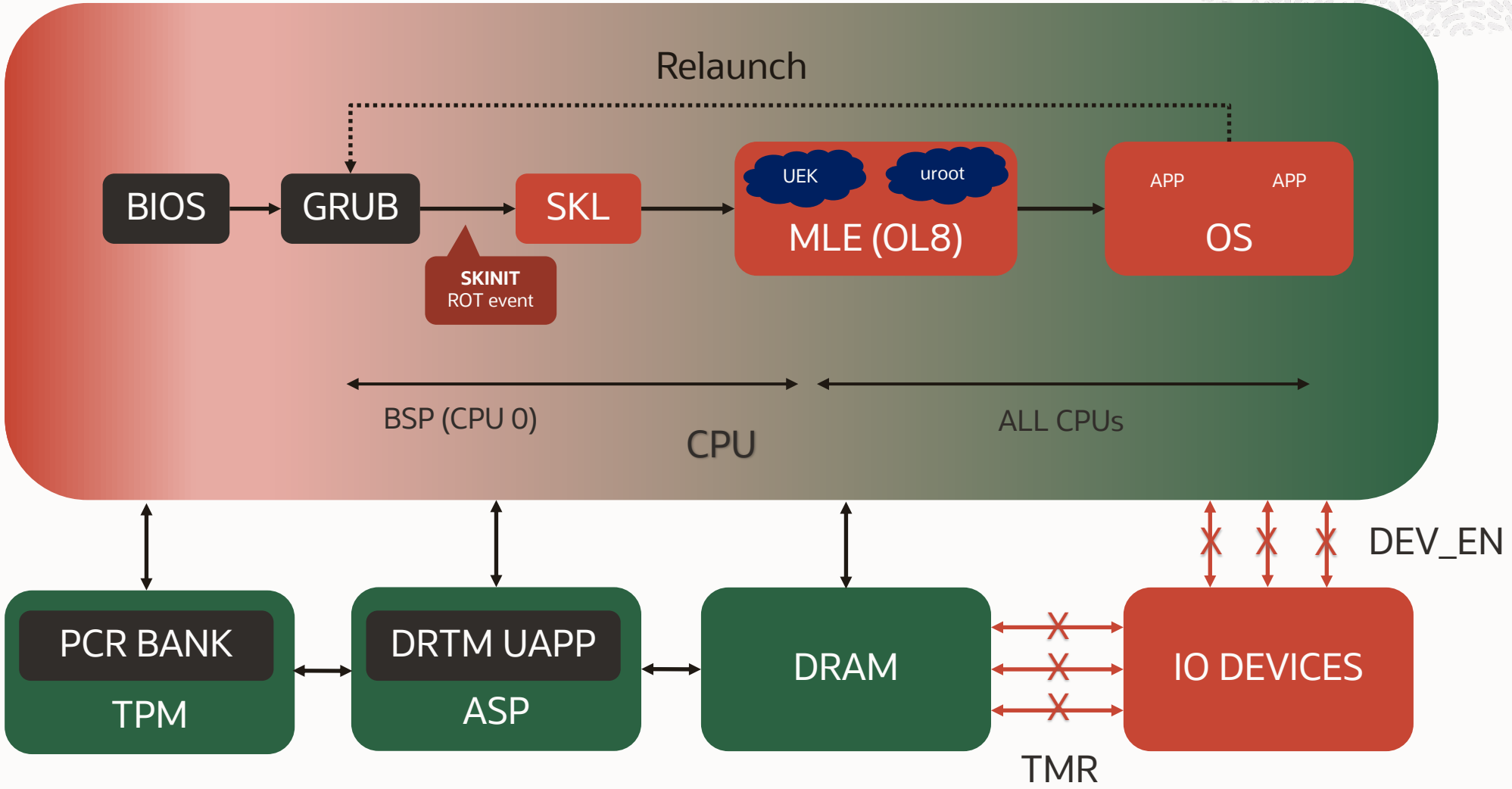
DRTM with AMD's ASP



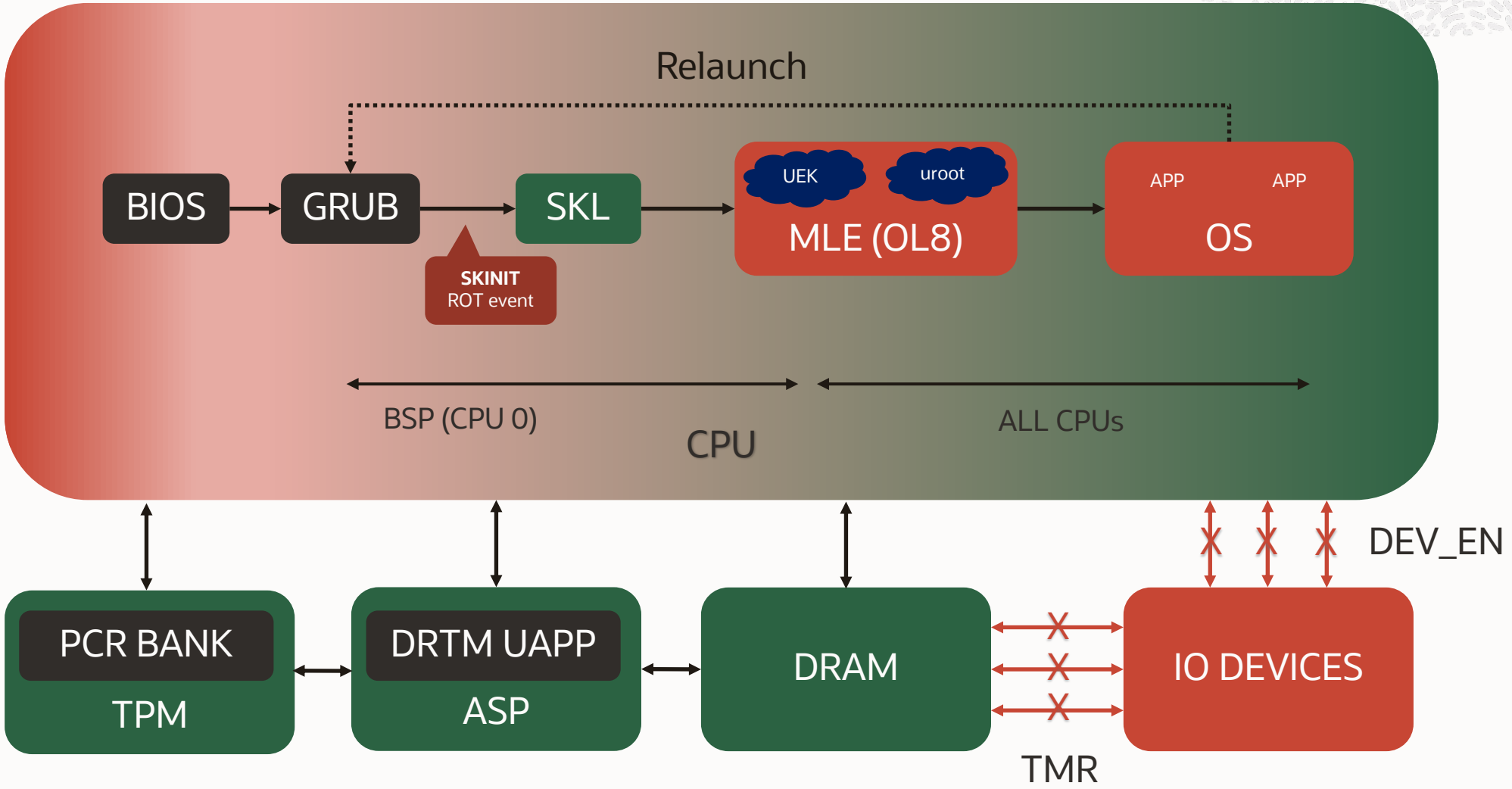
DRTM with AMD's ASP



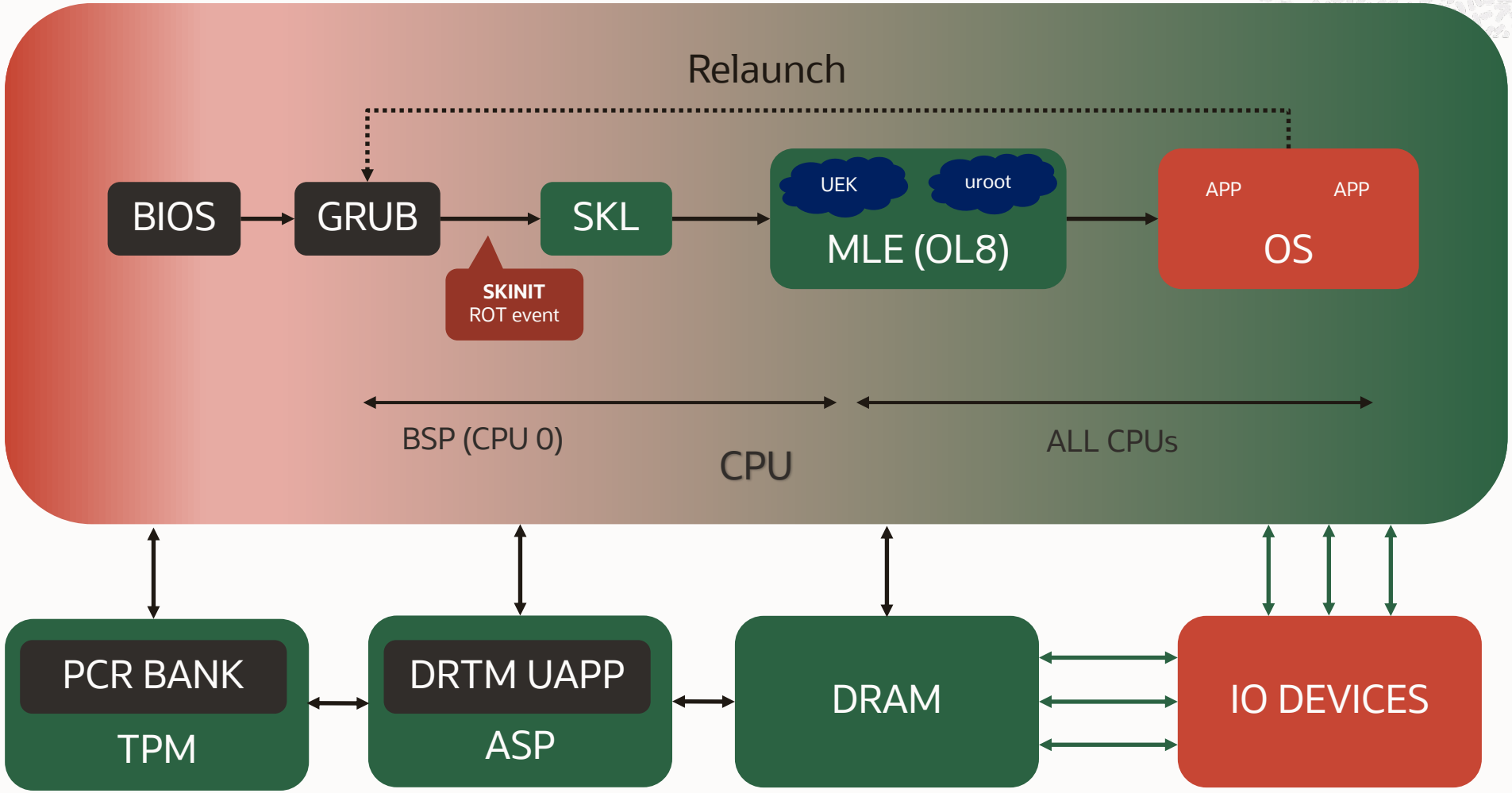
DRTM with AMD's ASP



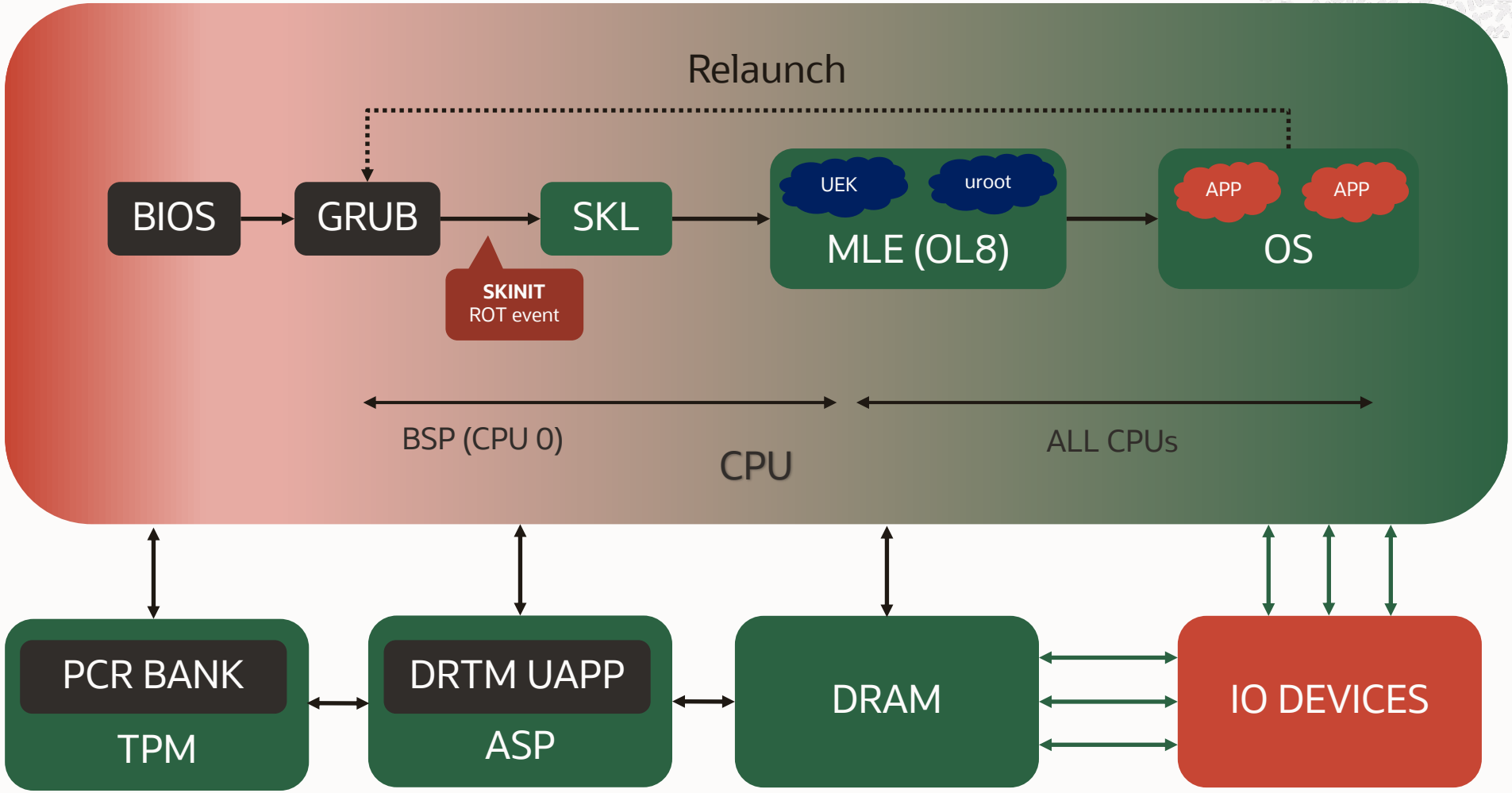
DRTM with AMD's ASP



DRTM with AMD's ASP

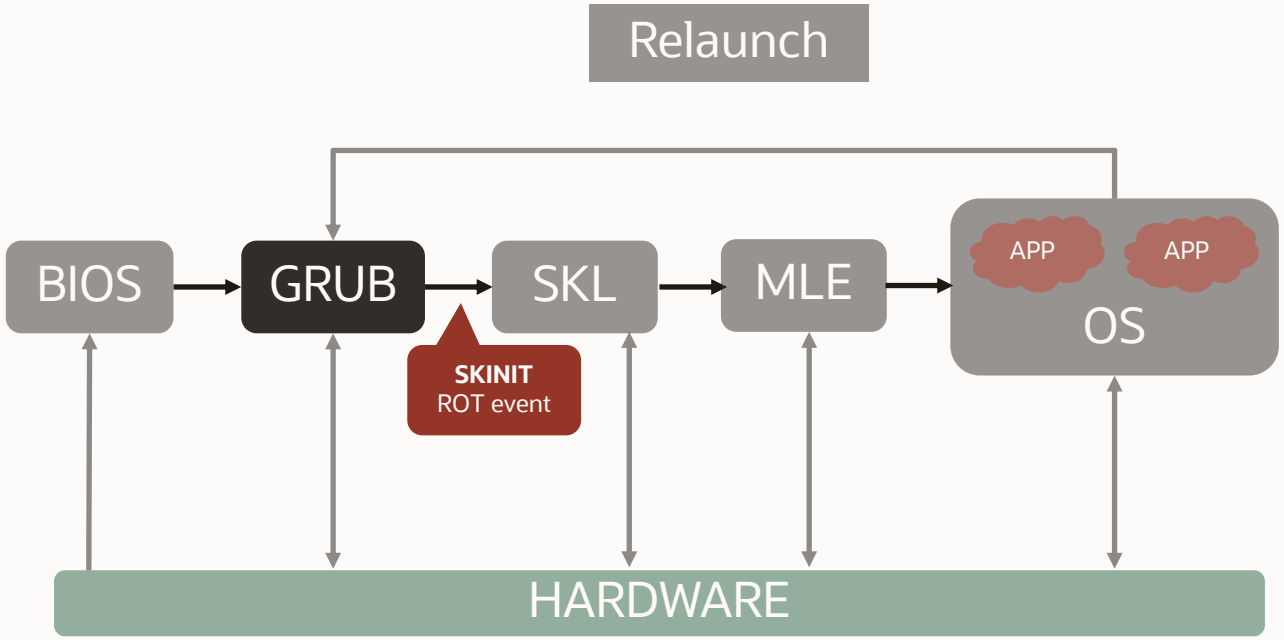


DRTM with AMD's ASP



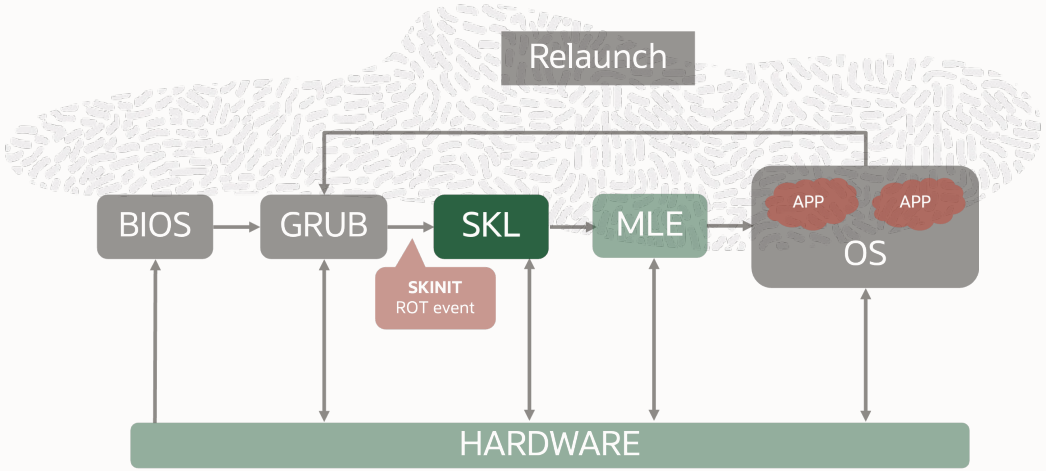
GRUB
AMD: Overview

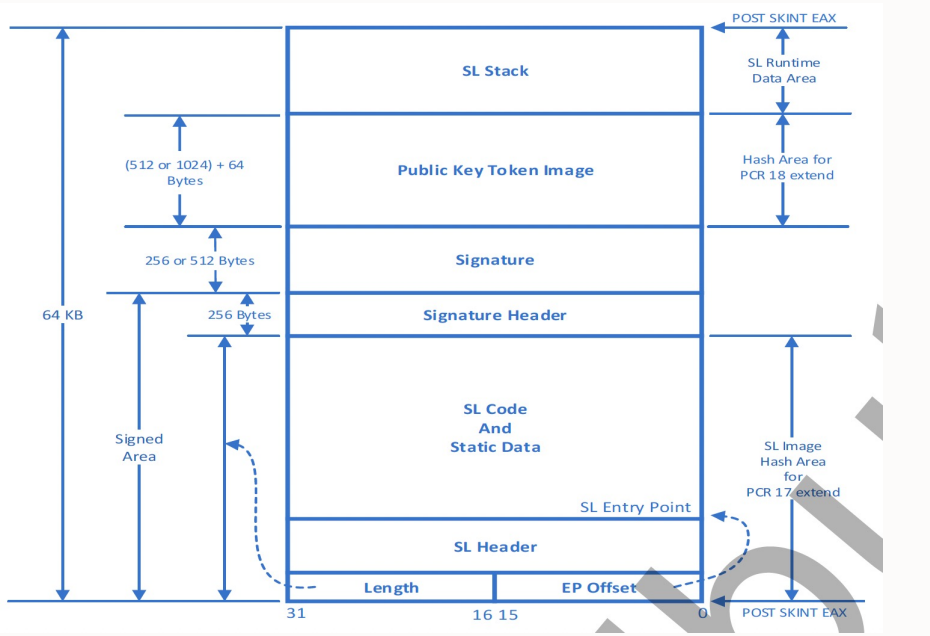
- Check CPU supports SKINIT
- Set ICR, CPU control registers, and fill SKL tags with bootloader data
- Setup DRTM service through ASP
  - DRTM Service Initialization
  - DRTM Get Capability
  - DRTM Setup Trusted Memory Region
- Runs the SKINIT instruction by passing the physical address of the SKL binary to it via the EAX register



Secure Kernel Loader (SKL)

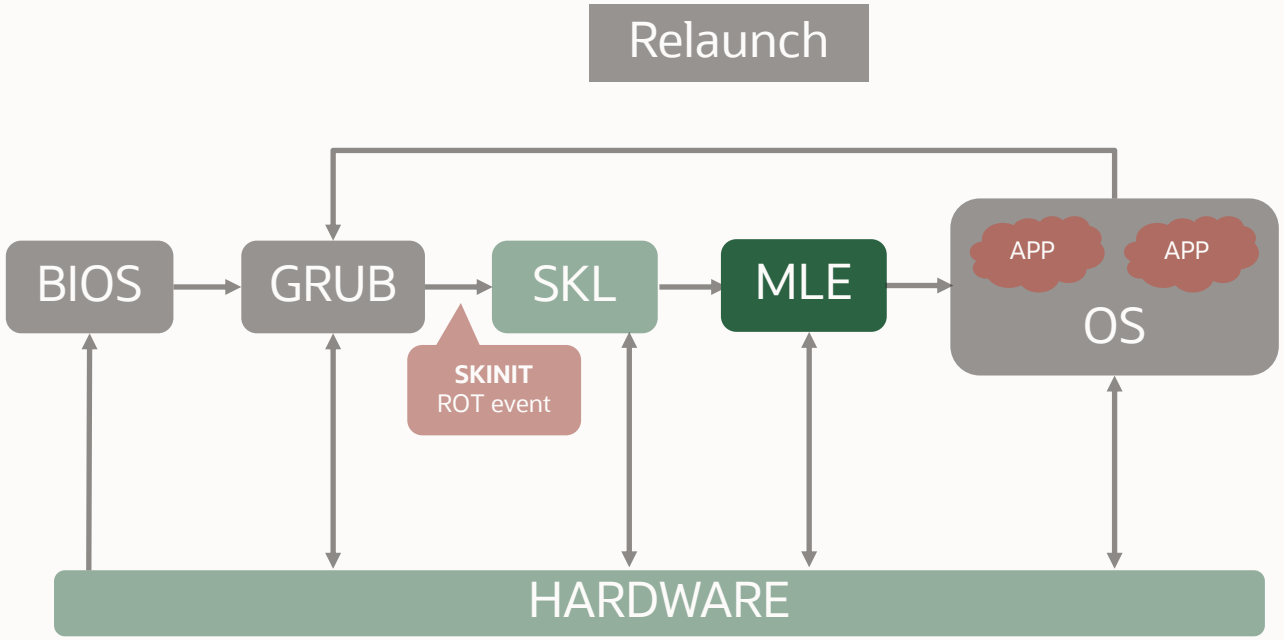
- Executes with SL_DEV protection
- Kicks off the state machine in the DRTM UAPP
- Measures the MLE kernel and extends the measurements to PCRs
- ASP validates the signature embedded in SKL before vectoring into it.
- AMD allows vendors to sign SKL. Oracle developed a tool to sign and package SKL.





Secure Launch Kernel (a.k.a MLE / DLME)

- Runs on the Bootstrap Processor (**BSP**)
- Enables DMA by releasing TMRs
- Locks TPM locality 2 and ends the DRTM state machine in the ASP
- Clears *INIT_REDIRECTION* in VM_CR MSR
- Wakes up the *Application Processors* (**APs**) using the startup IPI
- U-root makes policy decisions
- Two flavors: *Provisioning* & *SecureLaunch*



Linux Upstream Status

- Current Secure Launch patch set for Linux submission is Intel/TXT only.
- Version 7 of the patch set was posted to LKML in November 2023.
- Primarily contained AP startup changes using MWAIT/MONITOR per Thomas Gleixner's suggestion.
- The UEFI/Dynamic Launch stub support was removed in v7 because of substantial changes to the EFI startup code in the setup kernel.
- An effort is underway with the UEFI Linux kernel maintainers to redesign a new solution.
- These changes will be posted to LKML in patch set version 8 in the next few months.

Thank you


ORACLE