ORACLE

Linux Secure Launch for Arm From x86 to Arm

Ross Philipson – Oracle Corporation
Daniel Smith – Apertus Solutions LLC

September 18, 2024

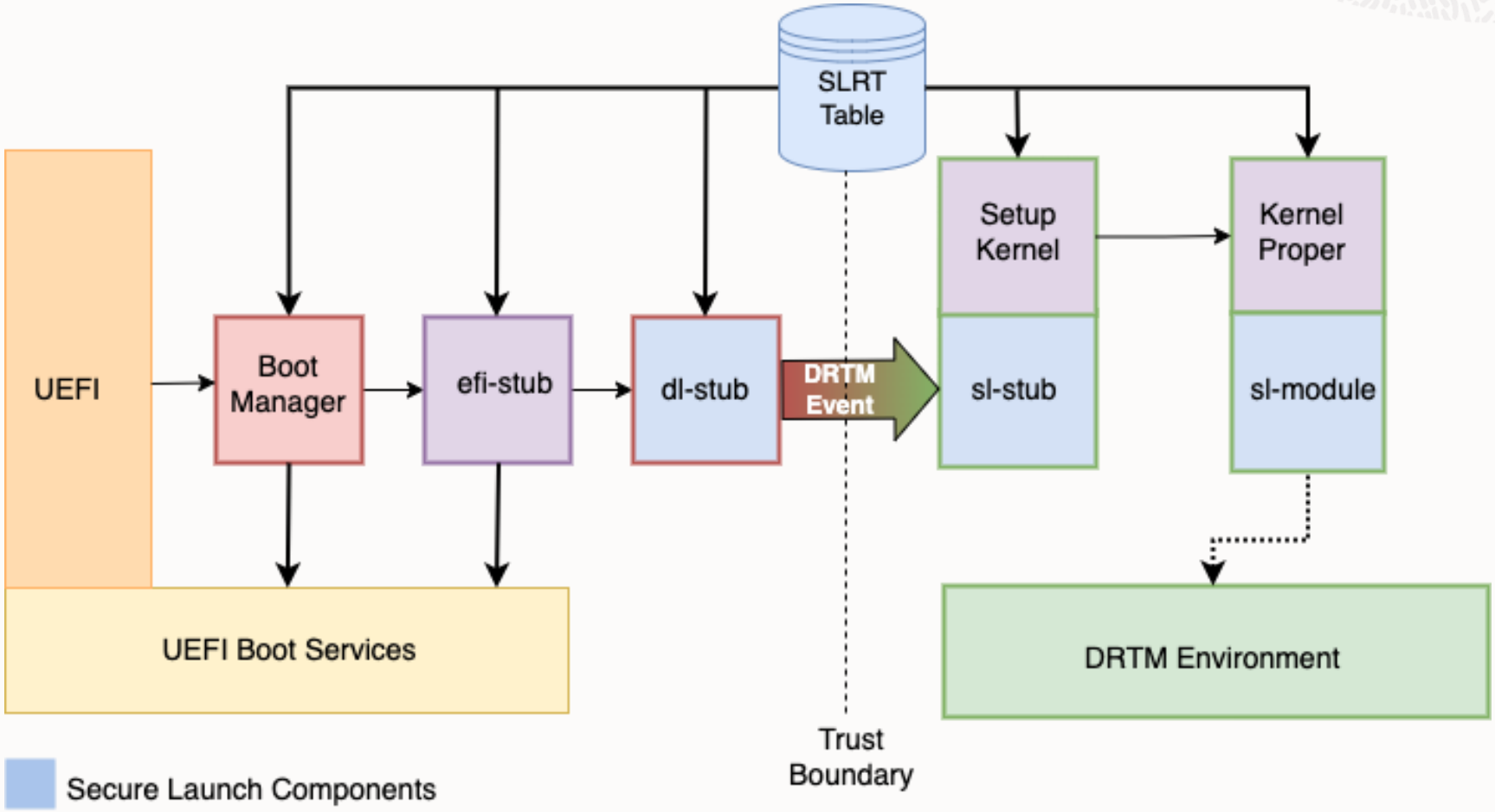
A Quick Introduction to TrenchBoot

- At a very high level, TrenchBoot is a project working to provide a common way to use TCG Dynamic Launch across:
  - Architectures, e.g. x86, Arm and Power.
  - Platforms, e.g. Intel and AMD.
  - Operating Systems/Kernels, e.g. Linux and Xen.

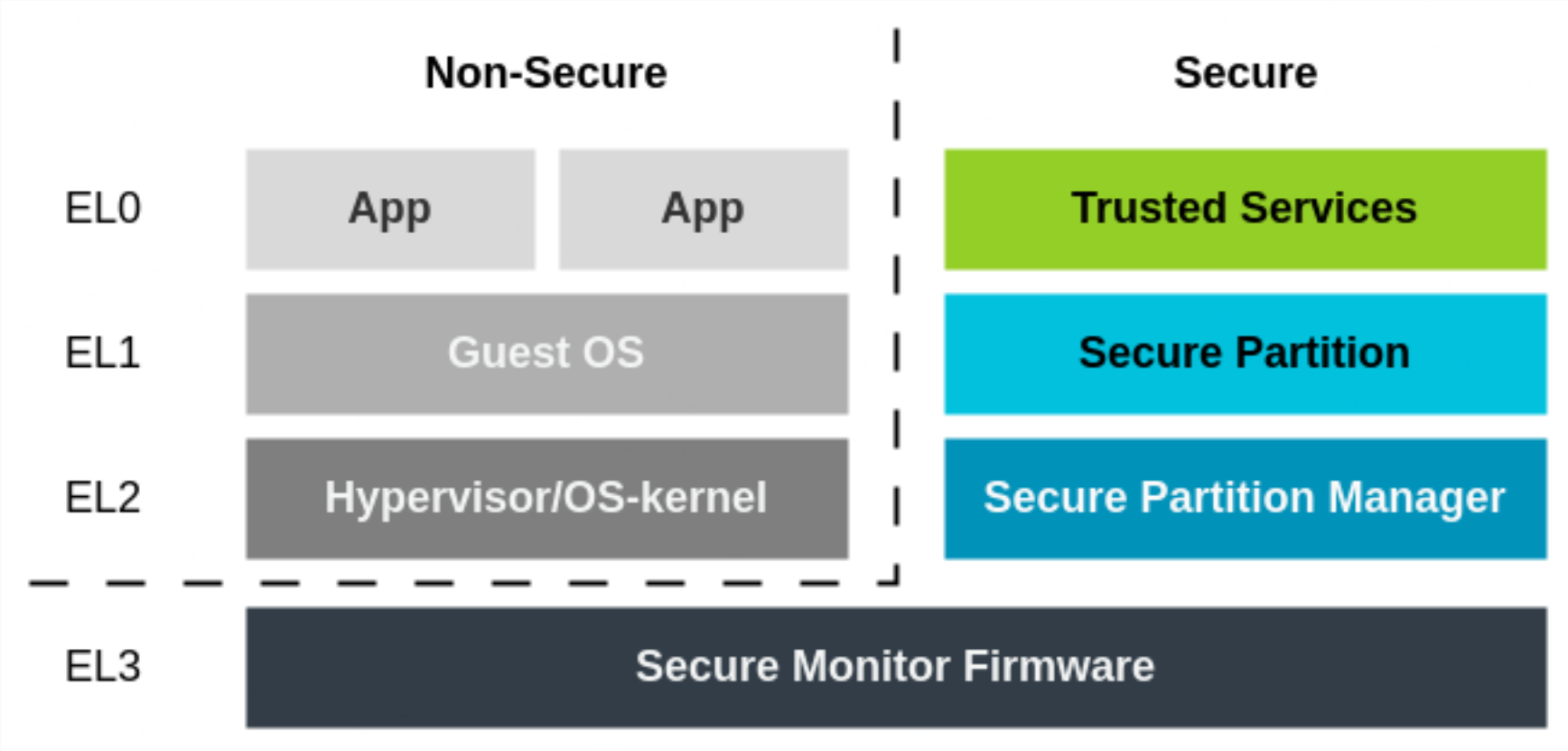
Linux Secure Launch

- Linux Secure Launch is the name under which the work is being done to add the ability for the Linux kernel to be directly started by a Dynamic Launch.
- The first platform/architecture supported by Linux Secure Launch was Intel x86.
  - AMD support is feature complete, awaiting for AMD DRTM documentation to be published.
- Many challenges have been encountered in the upstreaming process.
  - Positive results, e.g. Secure Launch Resource Table (SLRT), have come from working through the challenges.
- The resulting implementation translates well across architectures.

Linux Secure Launch Overview



Review: Arm Exception Levels and Security States



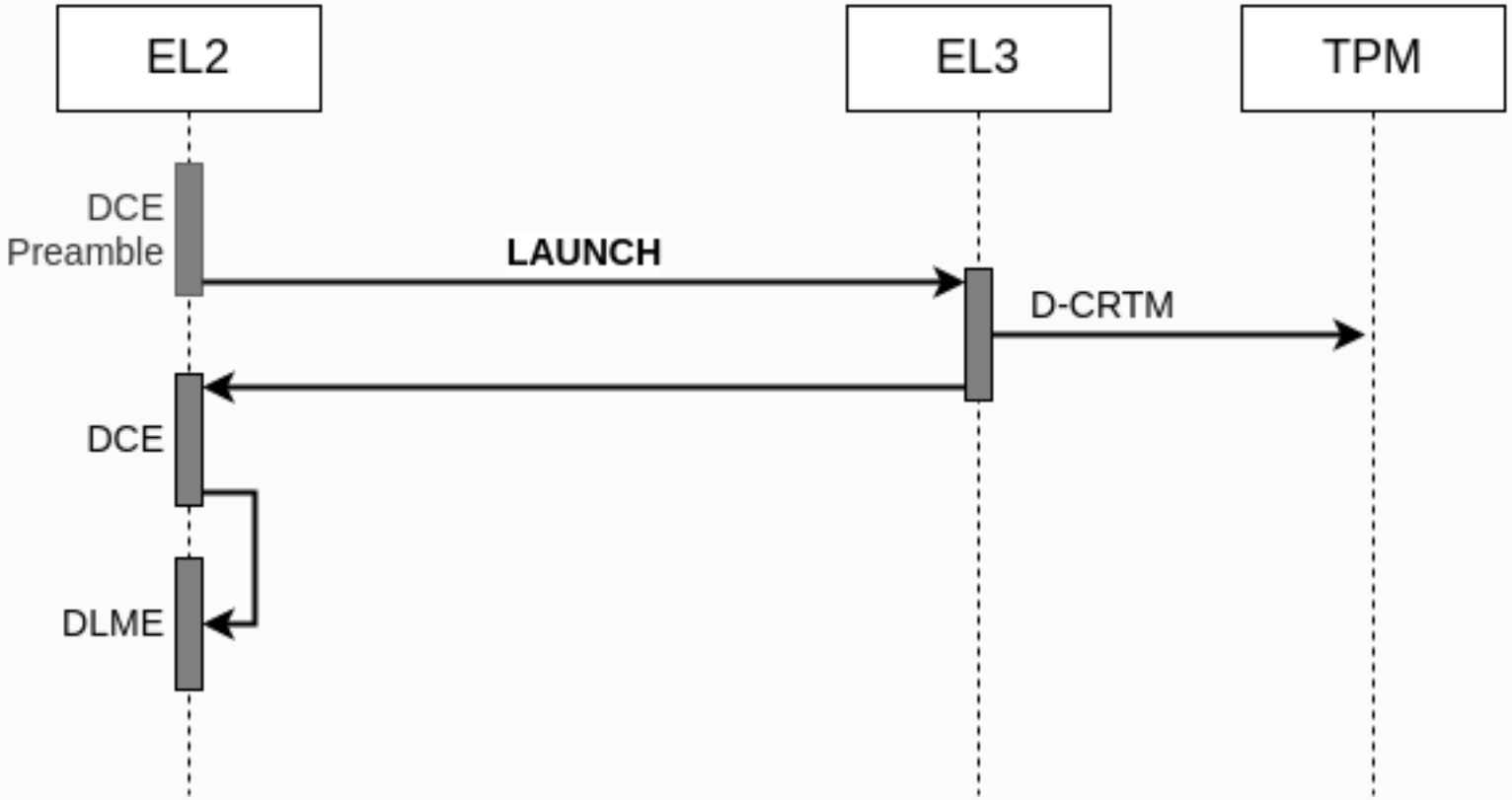
ARM/DRTM Terminology

- Terminology:
  - DRTM: Dynamic Root of Trust for Measurement
  - DLE: Dynamic Launch Event
  - D-CRTM: Dynamic Core Root of Trust for Measurement
  - DCE: DRTM Configuration Environment
  - DLME: Dynamically Launched Measured Environment
  - DLME Region: Region where the DLME Image and Data will reside
  - Normal World: The Non-secure privilege levels (Non-secure EL0, EL1, and EL2)
  - Secure World: The environment that is provided by the Secure privilege levels
  - Preamble: Module(s) responsible for setup and initiation of DLE.
  - Locality: A mechanism in a TPM that supports an access privilege hierarchy
  - PE: Processing element

Initial DRTM Architecture for Arm

- The DRTM Architecture allows for both hardware and firmware implementation approaches that a vendor may provide.
- The initial design is a firmware-backed DRTM and Normal World DCE.
- The next generation of chips from hardware vendors are expected to have hardware-backed DRTM support.

Firmware-backed Implementation Dynamic Launch

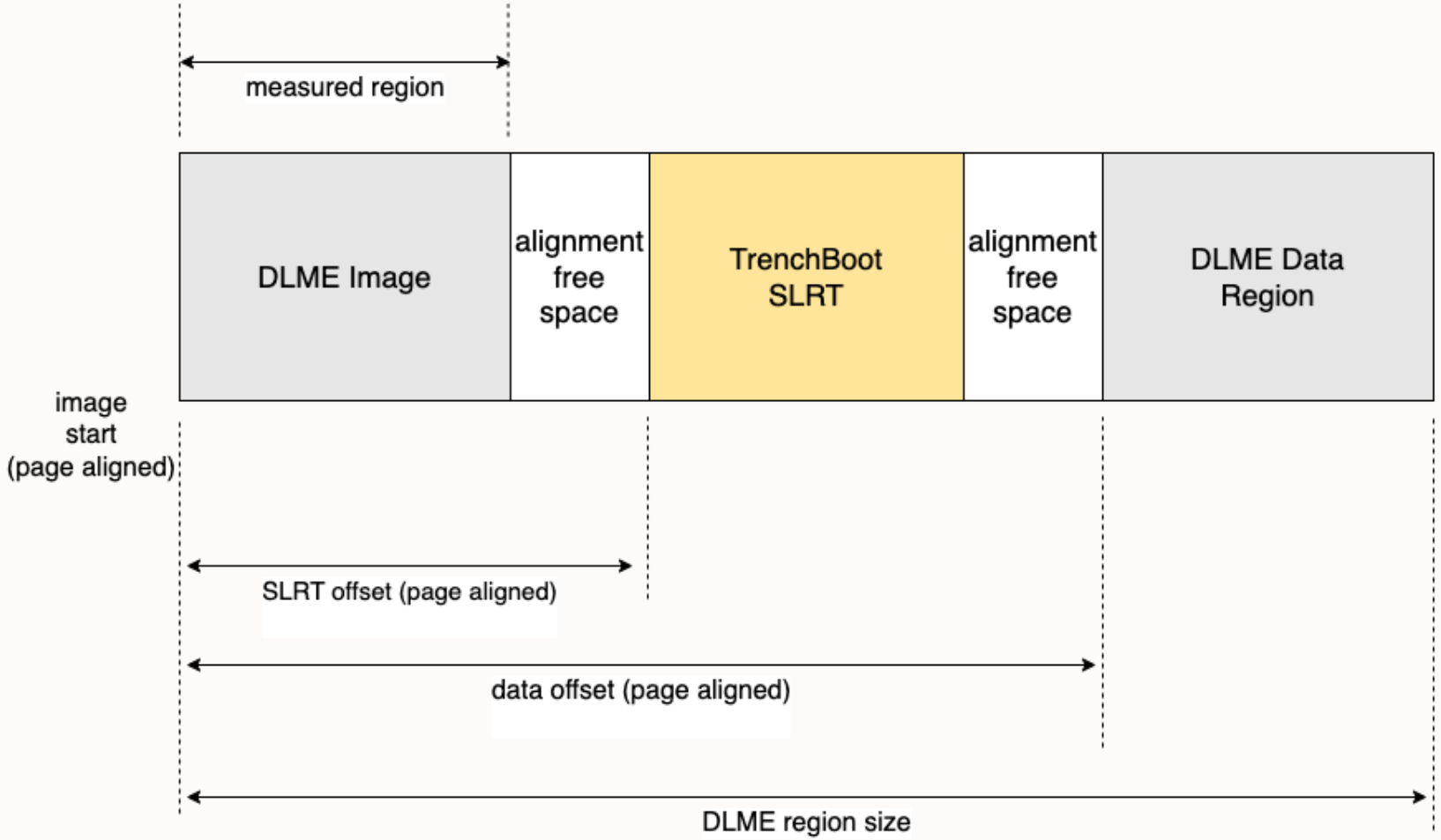


DRTM Launch Setup (DCE Preamble)

- GRUB will be used as the loader to prepare the environment.
  - The DRTM_VERSION will be used to report/log version info.
  - Inspection of the DRTM_FEATURES to report/log the available features.
    - This will confirm whether it needs to load a Normal World DCE
  - Set up the DCE, DRTM_PARAMETERS structure, and allocate DLME Region.
    - Request firmware-based hashing, falling back to TPM hashing.
    - Request complete DMA memory protection (which removes any requirement to predict memory relocations and memory management of the Linux kernel).
  - Populate the TrenchBoot SLRT between the DLME and the DLME Data Region.
  - Comply with system state required for Dynamic Launch.
    - Ensure execution on the Boot PE and that all other PEs are disabled/off.

DLME Region Layout

GRUB will layout the DLME Region in accordance to DRTM specification. The specification allows for allocating space in the middle of the region:



Normal World DCE

- As noted, the design and implementation of the normal world DCE is being delegated to Oracle by the vendor.
- The Vendor must provide details on how the EL3 D-CRTM will be invoking EL2 DCE.
  - This will drive the design for the normal world DCE before jumping into the DLME.
- The design approach will devise a protocol between a normal world DCE and Linux Secure Launch.
  - In the future, if a vendor solution included their own Secure World DCE, then the TrenchBoot project may need to produce a light-weight version of the Normal World DCE to bridge between vendor Secure World DCE and Linux Secure Launch.
  - The ideal outcome is to produce a Normal World DCE that would allow booting through the standard Arm Linux entry point for the kernel without modification.

DLME - Linux Secure Launch

- Provided all DRTM capable platforms will be UEFI based, the existing Arm64 entry point for the kernel will be used.
  - In this case, Dynamic Launch can be detected by locating the SLRT GUID in the EFI configuration tables.
  - If this is not the case, then a separate entry point for invoking the Linux Arm64 kernel may be needed.
- There will be an early call to a Secure Launch specific function, as with x86, to do measurements and validation.
  - Since the TPM is accessed via SMC calls to the secure firmware, the extends can be done at the time of measurement during the early kernel boot phase.
- As this device is a firmware-based implementation and does not impose a Dynamic Launch-specific CPU state, adjustments to the Arm SMP bring-up code will not be necessary.

Assumptions and Challenges

- The approach presented here for Linux Secure Launch for Arm is rough estimation built on the existing x86 implementation and the DRTM Architecture for Arm document (ARM DEN 0113).
- Constraints on the design:
  - Vendor documentation and hardware access.
    - Waiting for vendor to provide documentation/details regarding D-CRTM to DCE handoff to see how we can meet those requirements.
    - DRTM Architecture for Arm made assumptions of tight coupling between D-CRTM and DCE implementations.
  - Familiarity with Linux early startup for Arm is limited, but quickly getting up to speed.
- Mobile TPM usage under DRTM.
  - Assumption is that DRTM Service call to close locality 2 will move the TPM to locality 0.

Current Upstream Effort and Project Goals

- Version 10 of the Linux Secure Launch patch set was submitted to LKML on August 26, 2024
  - This submission is currently for Intel/TXT x86 platforms only.
  - The preamble support in GRUB for version 10 was rebased on a recent version and posted to the TrenchBoot project.
  - A “Quick Start” guide was introduced to the TrenchBoot project to help get capable systems configured with the latest work to perform a dynamic launch.
- Version 11 of the Linux Secure Launch patch set was submitted to LKML on September 13, 2024
  - The primary focus was to address and document concerns about the usage of the SHA-1 algorithm.
- AMD support for dynamic launch is actively being worked on. The expectation is that it will be posted upon acceptance of the Intel x86 work upstream.
- Arm support (as noted) is in the early stages but also being actively worked on.

Community Question

- Arm questions?
- X86 questions?

Thank you

Ross Philipson - ross.philipson@oracle.com

Daniel Smith - dpsmith@apertussolutions.com