presented by

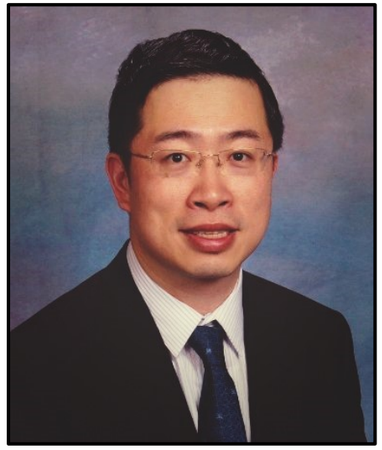
Building a System that “Just Works” – The Arm Firmware Ecosystem

UEFI 2020 Virtual Plugfest

May 20, 2020

Presented by Dong Wei (Arm) and Samer El-Haj-Mahmoud (Arm)

Meet the Presenters



Dong Wei
Standards Architect and Fellow
Member Company: Arm

Samer El-Haj-Mahmoud
Senior Principal Architect
Member Company: Arm

Agenda

- Arm Base Boot Requirements (BBR)
- Arm Open Source Firmware Projects
- Case Study: SBBR on Edge Devices

Arm Base Boot Requirements (BBR)

Goals

- Define a BBR spec to cover ‘A’ profile markets beyond server
- Continue the current EBBR spec with the community development approach
  - BBR spec refers to EBBR spec as needed
- BBR Spec
- Recipes
  - SBBR
  - ESBBR
  - EBBR
  - LBBR

- Establish interface requirements
  - PSCI, SMCCC (Common for all)
  - UEFI (for SBBR recipe)
  - ACPI (for SBBR recipe)
  - Exceptions (for ESBBR recipe)
  - SMBIOS
  - Devicetree (reference DT Spec)

System Firmware Landscape

Vertical

Custom Linux

Horizontal

Vertical

yocto PROJECT

ubuntu

SUSE

vmware ESXi

redhat

Microsoft

aws

facebook

Google

ORACLE Linux

chromeOS

FreeBSD

EBBR

ESBBR

SBBR

LBBR

U-Boot

NetBSD

LinuxBoot

coreboot

TrustedFirmware.org

Edge

Cloud & Datacenters

arm

Recipes

- SBBR
  - PSCI, SMCCC, UEFI, ACPI, SMBIOS interfaces
  - Windows Client/Server, RHEL require
  - SLES, Ubuntu, CentOS, Fedora, OpenSUSE, Debian, VMware ESXi, NetBSD, FreeBSD support
- ESBBR
  - SBBR with exceptions
  - VMware ESXi, Windows (IoT), SLES, Ubuntu, CentOS, Fedora, OpenSUSE, Debian, NetBSD, FreeBSD
- EBBR
  - PSCI, SMCCC, UEFI, DT interfaces
  - Fedora, OpenSUSE, Ubuntu, Debian, OpenWRT, Yocto, Windriver, Mentor
- LBBR
  - PSCI, SMCCC, LinuxBoot, DT or ACPI interfaces
  - Google, Facebook

SBBR

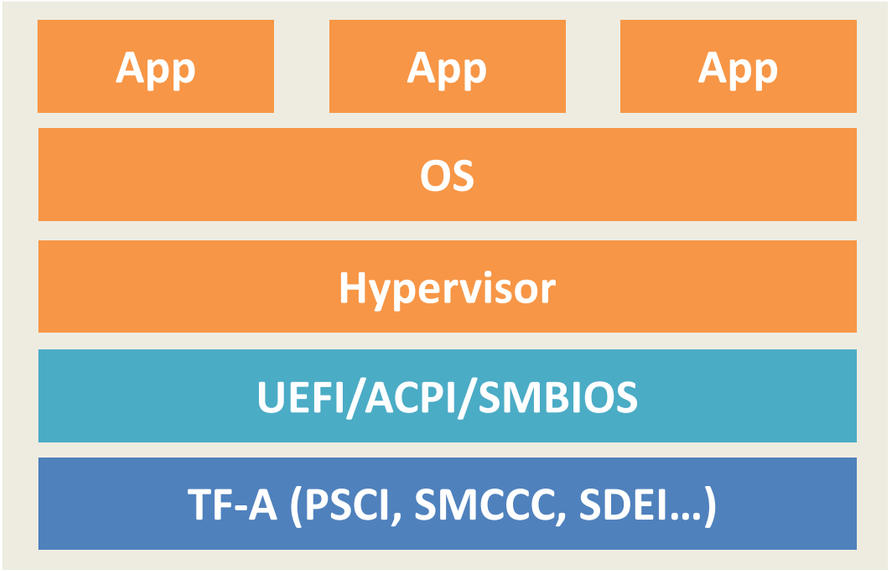
arm Developer

Arm Specs

- PSCI
- SMCCC
- TF-A
- Arm FFH
- Arm MM

https://developer.arm.com/products/architecture/system-architecture/server-system-architecture

- Firmware requirements for Windows, Red Hat, VMWare, SUSE, etc..
- **Horizontal Integration** requires standard firmware interfaces. Focus on interface requirements, not implementation



Industry Standards

- UEFI
- ACPI
- SMBIOS
- TCG FW spec
- PCIe FW spec

EBBR

arm Developer

Arm Specs

- PSCI
- SMCCC
- TF-A

https://github.com/ARM-software/ebbr

The goal is to establish consistent boot ABIs and behavior so that supporting new hardware platforms does not require custom engineering work.

EBBR is a subset of SBBR requirements. EBBR requirements have been implemented by the U-Boot project with Devicetree.

Industry Standards

- UEFI

LBBR

arm Developer

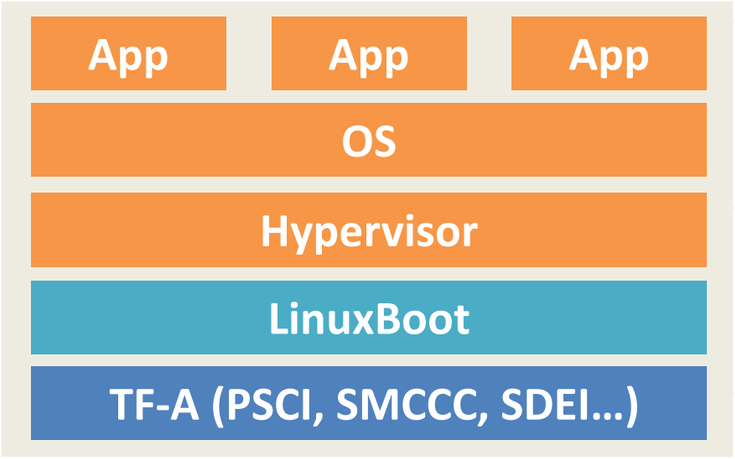
Arm Specs

- PSCI
- SMCCC
- TF-A
- Arm FFH (??)
- Arm MM (??)

LinuxBoot (https://www.linuxboot.org/) is system firmware implemented with the Linux kernel and a userspace runtime instead of EDK2 or U-Boot.

LinuxBoot on Arm normally replaces all non-secure firmware and can directly call TF-A APIs to control the platform. It still provides an ACPI or DT description.

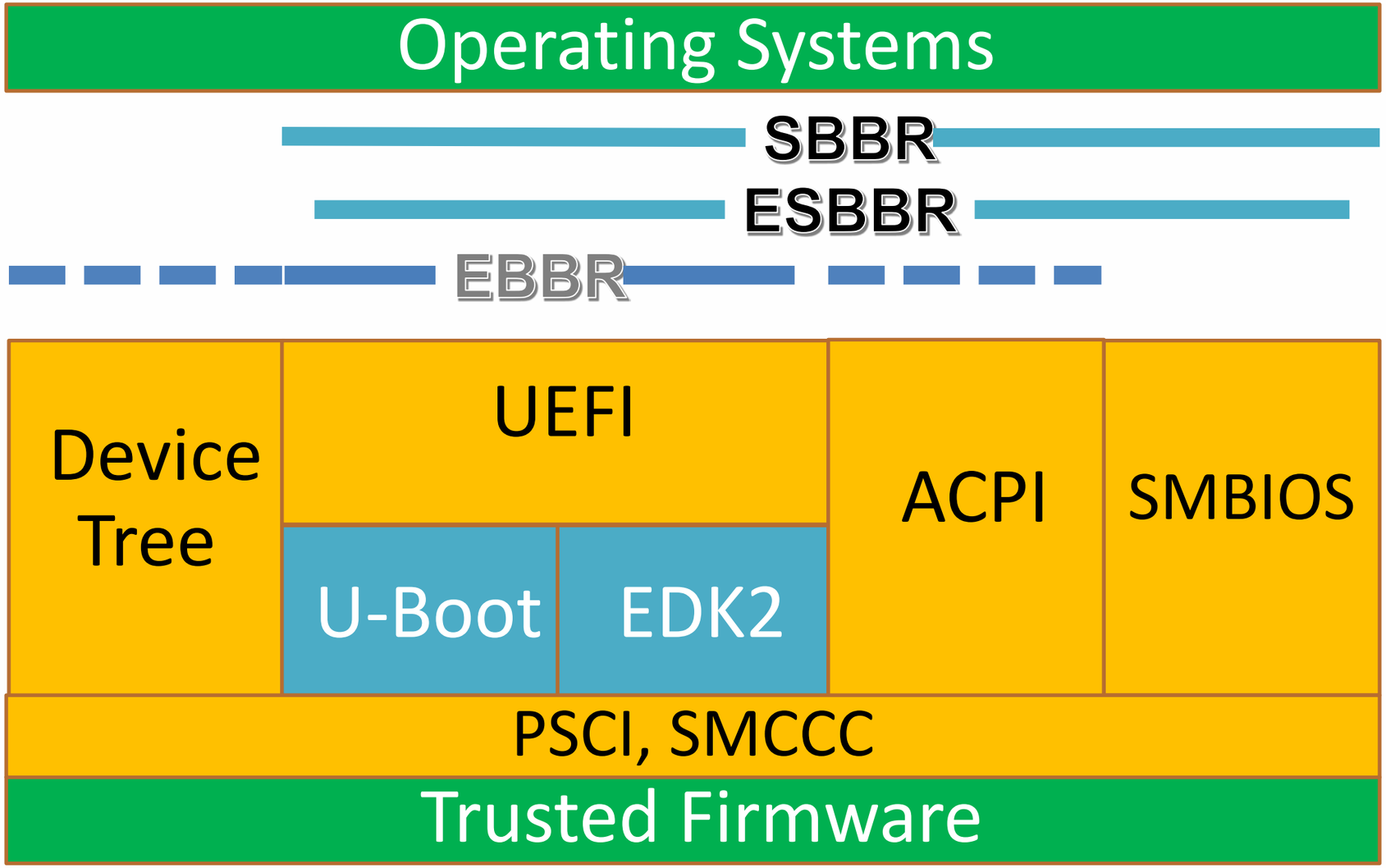
LinuxBoot doesn't implement all of SBBR. OSes that require the UEFI ABI may not be supported, unless UEFI ABI is also implemented in LinuxBoot.



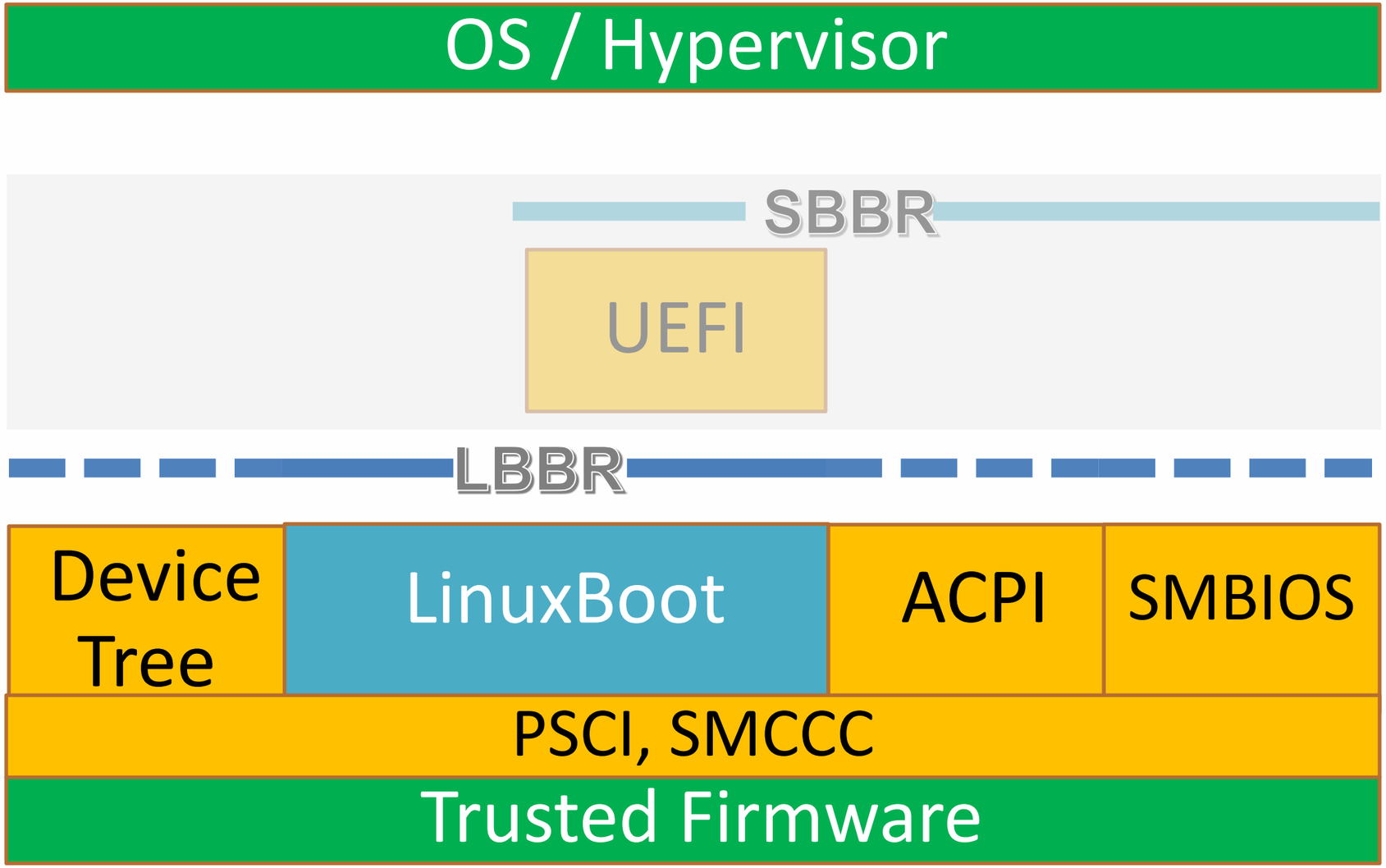
Industry Standards

- ACPI
- Devicetree
- SMBIOS
- TCG FW spec (?)
- PCIe FW spec (?)

Recipe Relationships



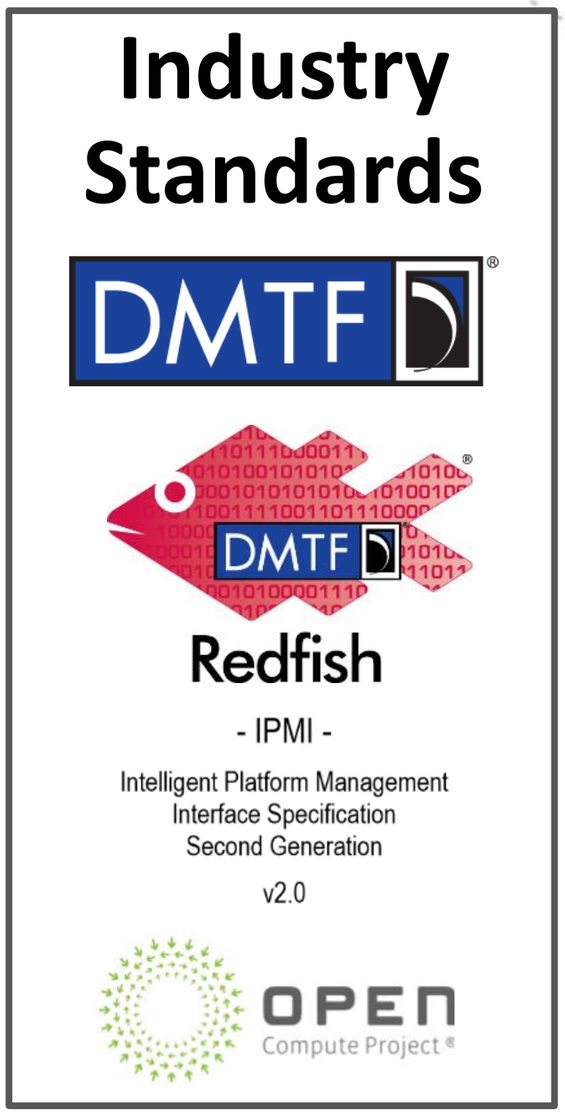
Recipe Relationships



SBMR - Server Base Manageability Requirements

https://developer.arm.com/products/architecture/system-architecture/server-system-architecture

- Hardware and Firmware requirements for standard system management of SBSA/SBBR compliant servers.
- v1.0 Release March 2020
- Provides Foundation for standardized common capabilities, and allows value-add on top
- Builds on top of prevalent industry standards for systems management
  - DMTF Redfish
  - DMTF Management Component Transport Protocol (MCTP)
  - DMTF Platform Level Data Model (PLDM)
  - OCP Hardware Management
  - IPMI



Open Source System Firmware on Arm

Arm Open Source Firmware

- Arm systems support firmware solutions with **multiple boot models**, and that can be **open source** OR **commercial**.
- Arm's strategy is to encourage partners to provide **full open source** firmware implementations, regardless of the boot model
- Open source firmware options on Arm systems include:

TrustedFirmware	• Open source for Secure World firmware
TianoCore / EDK2	• Open source for UEFI, ACPI, SMBIOS standard system firmware
U-Boot	• Open source for embedded systems firmware
LinuxBoot	• Open source for cloud providers Linux-based firmware
OpenBMC	• Open source BMC firmware

Trusted Firmware

- https://trustedfirmware.org/
- Open Source, Open Governance Community Project
- Evolution of former Open Source “Arm Trusted Firmware” project
- Reference implementation of Secure world software for Armv7 & Armv8 architectures (both A/M-Profiles)
- Membership open to all
- Governance overseen by a board of member representatives
- Technical direction overseen by TSC

TrustedFirmware.org

~~ARM Trusted Firmware~~

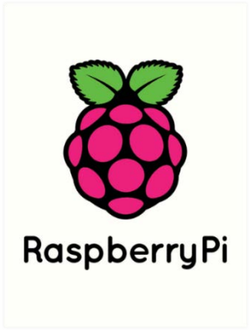
TrustedFirmware.org

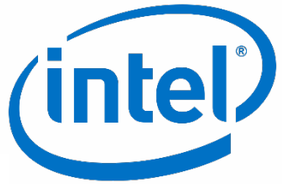
(Open governance Community project)

TrustedFirmware Diverse Community

- 30+ platform ports from 16+ vendors !
- 25+ partners contributing

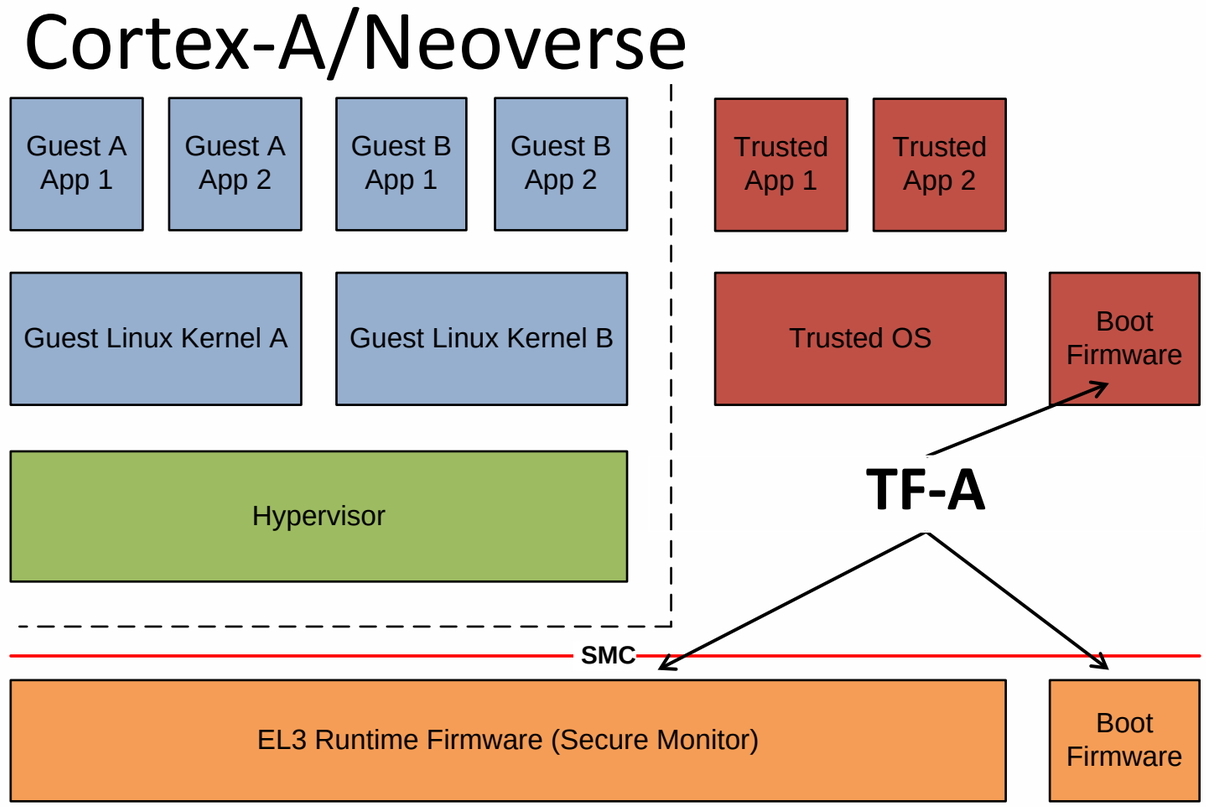




Trusted Firmware-A (TF-A)

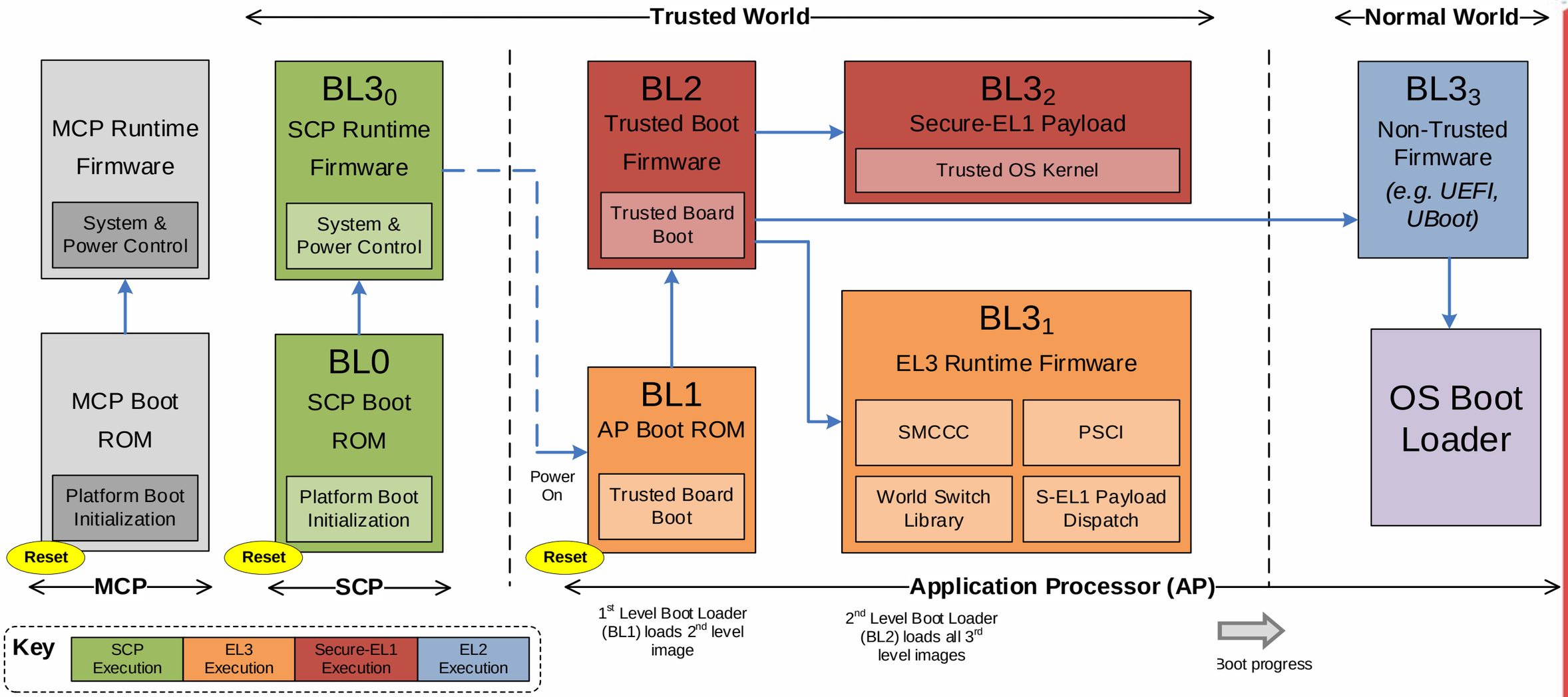
- Secure world reference software for all Arm Cortex-A & Neoverse processors across all market segments.
- Trusted boot flow and runtime firmware providing standard implementation of Arm specifications:
  - SMCCC (SMC Calling Convention)
  - TBBR (Trusted Board Boot Requirements)
  - PSCI (Power State Coordination Interface)
  - SCMI (System Control & Management Interface)
  - SPCI (Secure Partitions Client Interface)



https://git.trustedfirmware.org/TF-A/trusted-firmware-a.git/about/

https://git.trustedfirmware.org/TF-A-Tests/trusted-firmware-a.git/about/

TF-A Boot Flow

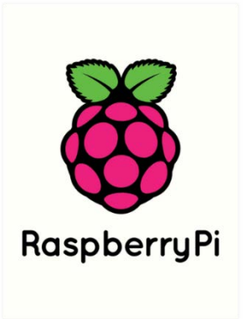


TianoCore

- https://www.tianocore.org/
- Community project supporting open source implementation of Unified Extensible Firmware Interface (UEFI) firmware
- Covering multiple standards: UEFI, PI, ACPI, SMBIOS, UEFI Shell, etc.
- Main project: **EDK2**. Modern, feature-rich, cross-platform firmware development environment for the UEFI and PI specifications.
- BSD-2-Clause-Patent license

Arm support on TianoCore

- Growing Arm implementations on edk2 and edk2-platform
  - Complete/partial platforms, silicon drivers, libraries, support code
  - Diverse community participation, continuous increase

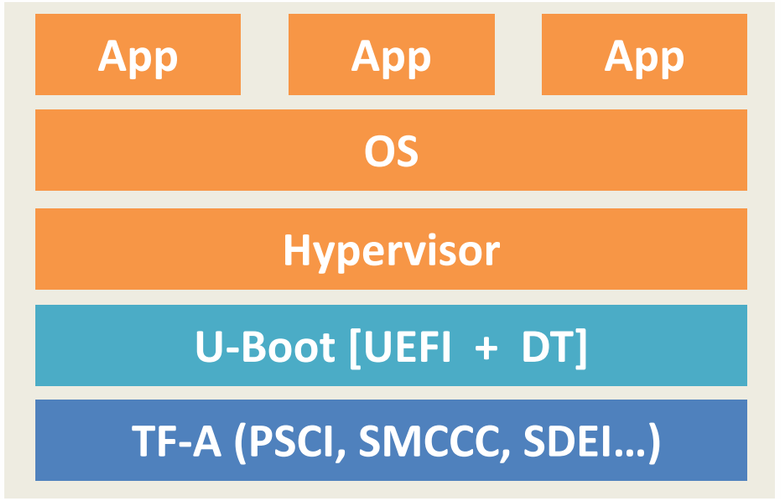



Arm on UEFI Showcase – Raspberry Pi

- https://rpi4-uefi.dev/
- Arm64 standards firmware for Rasbperry Pi
  - RPi3: EBBR (EDK2 UEFI + Devicetree)
  - RPi4 : SBBR (EDK2 UEFI + ACPI), with Devicetree option
- Goal is to make the Pi "ServerReady" - Support standard OSes
- Fully open source (TianoCore + TF-A)
- Community driven collaboration (including VMware, Arm, and others in the community)
  - EDK2 up-stream: https://github.com/tianocore/edk2-platforms/tree/master/Platform/RaspberryPi/
  - Discord community channel: **#rpi4-uefi-dev (**https://discordapp.com/invite/fqRhc8y)

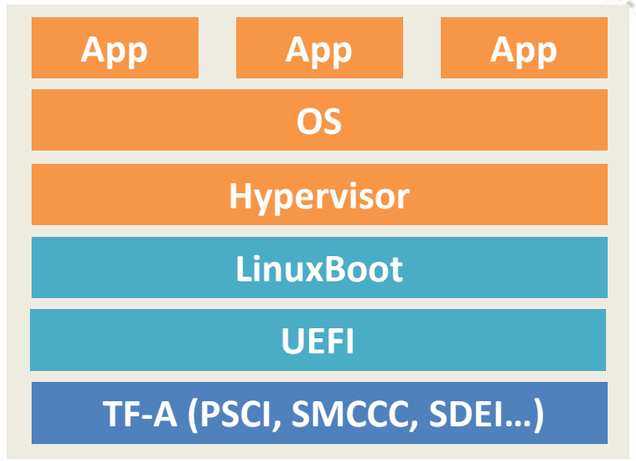
U-Boot Firmware

- https://www.denx.de/wiki/U-Boot
- “Universal Bootloader.” Open source, GPL
- Supports multiple architectures (including Arm/Arm64)
- Portable, easy to port/debug
- Many (100s) boards up-streamed
- Suitable for embedded devices (predominantly vertically integrated ecosystem)
- U-Boot implements UEFI ABI as required by EBBR
  - Support both Arm64 and x64
  - https://gitlab.denx.de/u-boot/u-boot/blob/master/doc/uefi/uefi.rst
  - Allows standard OS bootloader (like GRUB) to load and boot standard OS

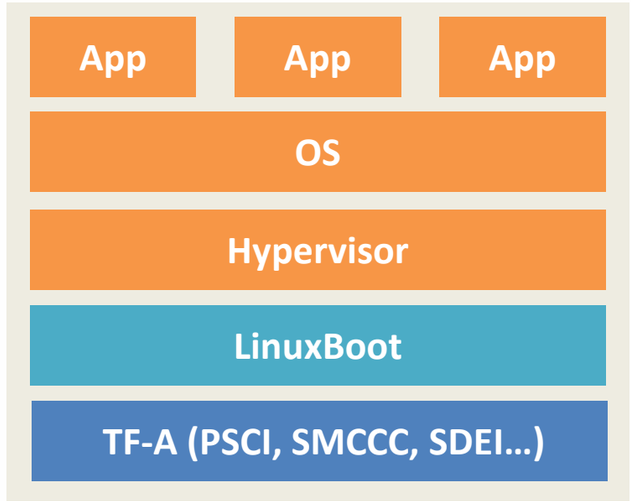


LinuxBoot

- https://linuxboot.org/
- LinuxBoot is a firmware for servers that replaces specific firmware functionality like the UEFI DXE phase with a Linux kernel and runtime
- Re-use existing Linux drivers code (without the need to write DXE/UEFI drivers)
- Linux usermode using u-root https://github.com/u-root/u-root
- Two approaches on Arm servers:
  - LinuxBoot in UEFI FV (replace UEFI Shell binary with LinuxBoot binary)
  - Direct load from TF-A to LinuxBoot (no UEFI)
- It is still possible to implement UEFI/APCI/SMBIOS/DT ABIs (or carry “blobs”) in LinuxBoot for final OS consumption



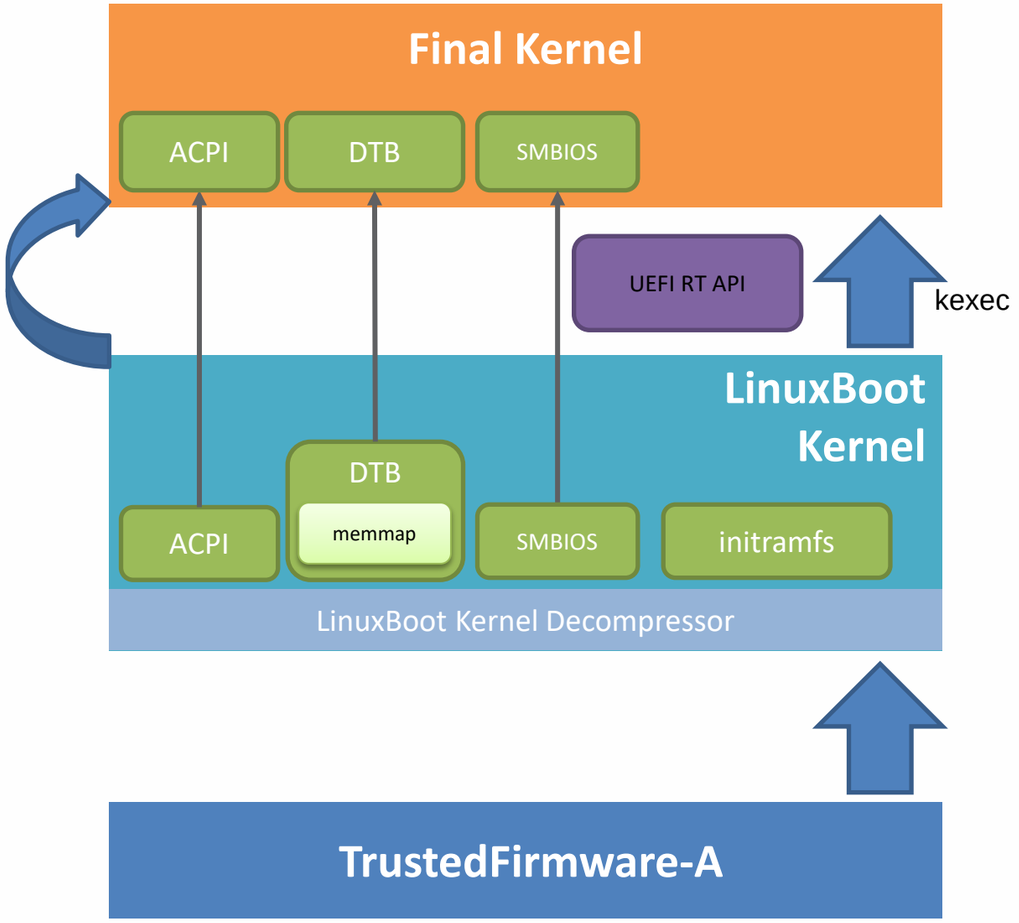
Approach #1 – LinuxBoot in UEFI FV



Approach #2 – Skip UEFI

LinuxBoot and UEFI

- In addition to ACPI/SMBIOS/DT payloads pushed to the final OS, LinuxBoot *could* publish UEFI ABI to the final OS
  - UEFI is an API spec. DXE/PI are not required to implement UEFI (or UEFI Runtime Services)
  - Similar to U-Boot's UEFI implementation (EBBR)
  - Enables OS functionality that depends on UEFI runtime APIs



Questions?

Thanks for attending the UEFI 2020 Virtual Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by