presented by

UEFI Updates and Open Source Firmware Evolutions on Arm

Spring 2019 UEFI Plugfest

April 8-12, 2019

Presented by Dong Wei & Matteo Carlini (Arm)

Agenda

- UEFI SBBR & EBBR Updates
- Open Source Firmware evolutions on Arm
  - Trusted Firmware Updates
  - Secure Software Isolation Recap
  - Firmware challenges for Edge
- Questions?

UEFI Updates:
Server Base Boot Requirements & Embedded Base Boot Requirements

Arm Specs

- PSCI
- SMCCC
- Arm FFH
- Arm MM

SBBR: Server Base Boot Requirements

Operating systems running on standard server hardware require standard firmware interfaces to be present in order to boot and function correctly. The Server Base Board Boot Requirements (SBBR) document describes these firmware requirements. The SBBR covers UEFI, ACPI and SMBIOS industry standards as well as standards specific to Arm, such as PSCI. Together with SBSA, the SBBR provides a standard based approach to building Arm servers and their firmware. The specification is developed in conjunctions with partners across the industry.

For more information, please visit:
https://developer.arm.com/products/architecture/system-architecture/server-system-architecture

License

Arm Confidential Proprietary Notice for drafts and Arm Non-Confidential Proprietary Notice for released final spec.

Contribution

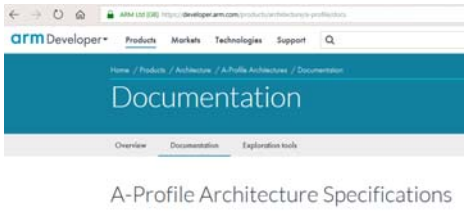
Members of the Arm Server Advisory Committee may submit Engineering Change Requests (ECRs) and the Committee decides to approve/reject the ECRs. There is a mailing list and a monthly conference call.

Release

Current version 1.1, ServerReady v1.0 launched Oct 2018
Working on content for the next version.

Industry Standards

- UEFI
- ACPI
- SMBIOS
- TCG FW spec
- PCI FW spec



Arm Specs

- PSCI
- SMCCC
- TF-A

EBBR: Embedded Base Boot Requirements

The Embedded Base Boot Requirements specification defines requirements for embedded systems to enable inter-operability between SoCs, hardware platforms, firmware implementations, and operating system distributions. The aim is to establish consistent boot ABIs and behavior so that supporting new hardware platforms does not require custom engineering work.

EBBR is intended for embedded and things-edge devices

For more information, please visit:
https://github.com/ARM-software/ebbr

License

Contributions are accepted under the same with sign-off under the Developer's Certificate of Origin.

Contribution

Anyone may contribute to EBBR. Discussion is on the boot-architecture@lists.linaro.org and arm.ebbr-discuss@arm.com mailing list, and there is a weekly conference call.

Release

v1.0 in March 2019
Secure Boot and Capsule Services for Firmware Update are of interest in future versions

Industry Standards

devicetree
.org

Open Source Firmware Evolutions on Arm

Trusted Firmware Updates

- ~~Arm Trusted Firmware~~

  → **TrustedFirmware.org**

  – Open Governance project

  – Reference implementation of secure world software for Armv7-A, Armv8-A and Armv8-M

  – Trusted Firmware-A for A-class (TF-A)

  – Trusted Firmware-M for M-class (TF-M)

  – Foundation of Arm's Platform Security Architecture (PSA)

  – https://www.trustedfirmware.org/

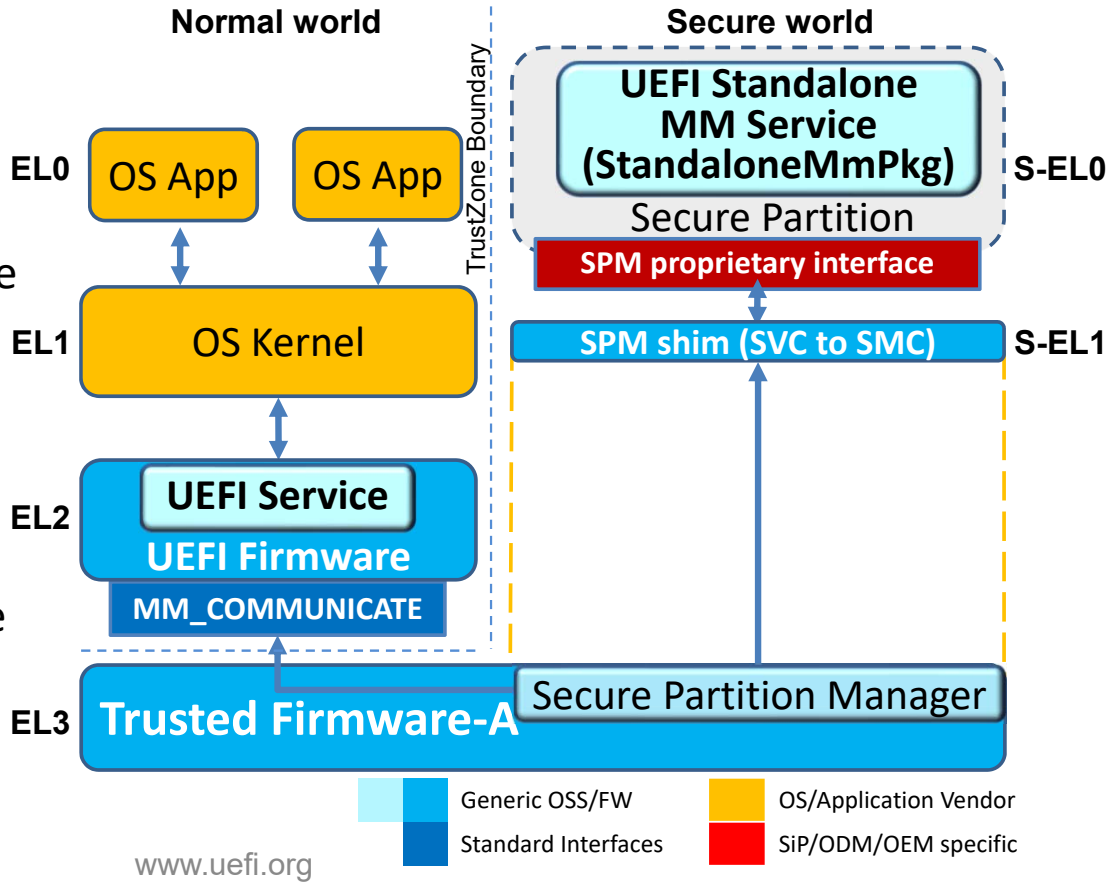
Secure Software Isolation Recap

- As the firmware on the Application Processor (AP) is getting bigger and more complex to audit, there is a strong need for:
  - Separation of liability amongst services through isolation
  - Restriction of the level of privilege available to each service
- Software architecture support needed to provide isolation between components in the Arm Secure world
  - Applicable to existing Armv8.3 and earlier Armv8 architecture
  - Ready to support Armv8.4 Secure-EL2 virtualization extension
- Standard interfaces at component boundaries to enable
  - Distinct software to interoperate and be audited separately
  - Removal of vendor specific code from secure firmware

Secure Partitions Recap – Single SP

Secure Partitions (SP):

- Unprivileged software sandbox environment running in the Secure world
- Isolated execution context
- Limited access to system resources
- Execute UEFI image with Standalone Management Mode support to execute secure management services
- Leverage Arm MM Specification: MM_COMMUNICATE SMC to request partition services
- Code reuse between normal/secure world whenever possible
- Reduced services code into privileged Trusted Firmware (EL3)
- **EDK2 StandaloneMmPkg**



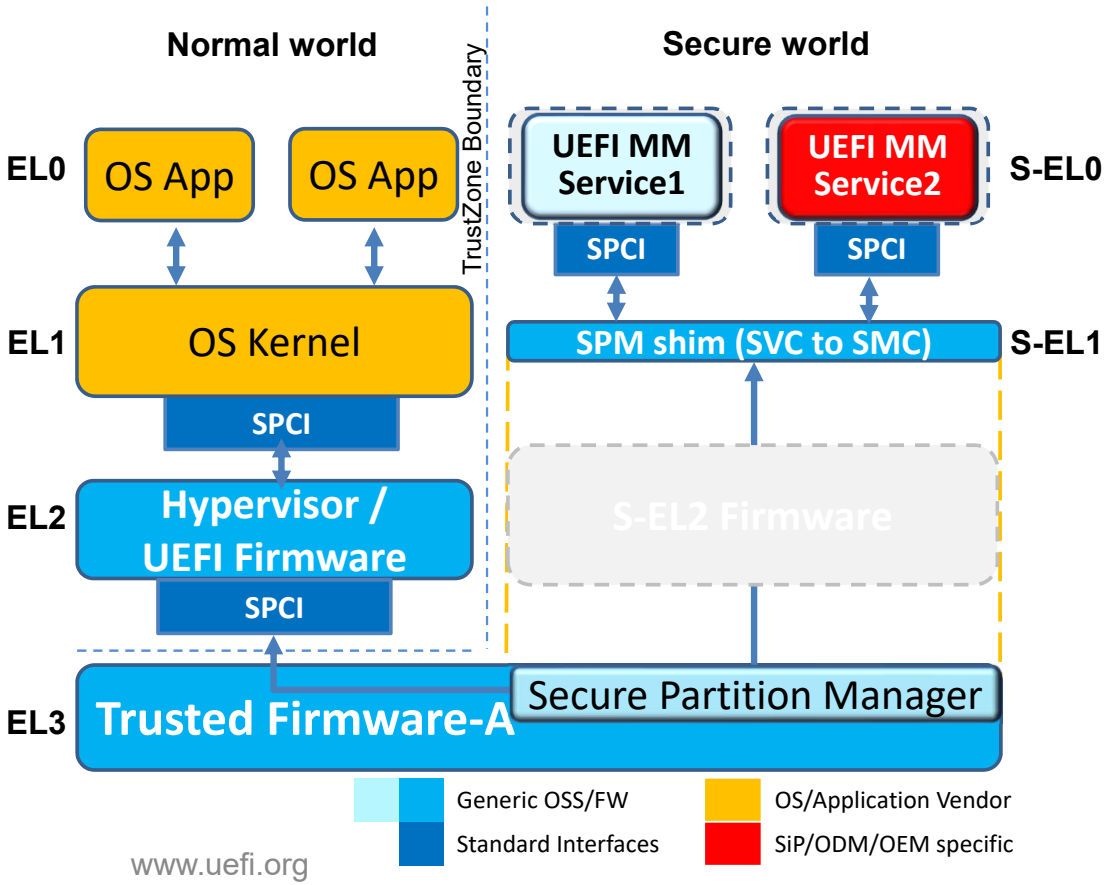
Secure Partitions Recap – Multiple SPs

Secure Partition Manager (SPM):

- Runs at EL3 as part of the Runtime portion of Trusted Firmware-A
- Enforces principle of least privilege
- Initializes SP at boot time and manages runtime requests
- Enables communication between service requestors and providers
- Implements **Secure Partition Client Interfaces (SPCI)**, MM evolution

Use-cases:

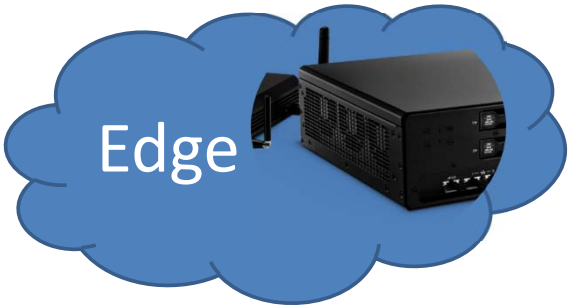
- Secure Variable access / FW Update
- fTPM implementation
- Errata handling / RAS Errors handling
- BMC communication



End game on Armv8 UEFI systems?

Here comes the Edge

…still few problems to solve…



- Firmware specific challenges in the Edge space
  1. UEFI Standalone MM Services plus concurrent Trusted OS/Apps
  2. UEFI & ACPI on legacy U-Boot/DT Edge devices
- Multiple Firmware Signing Domains for Secure Services
  - Common problem to Edge/Infrastructure/Cloud

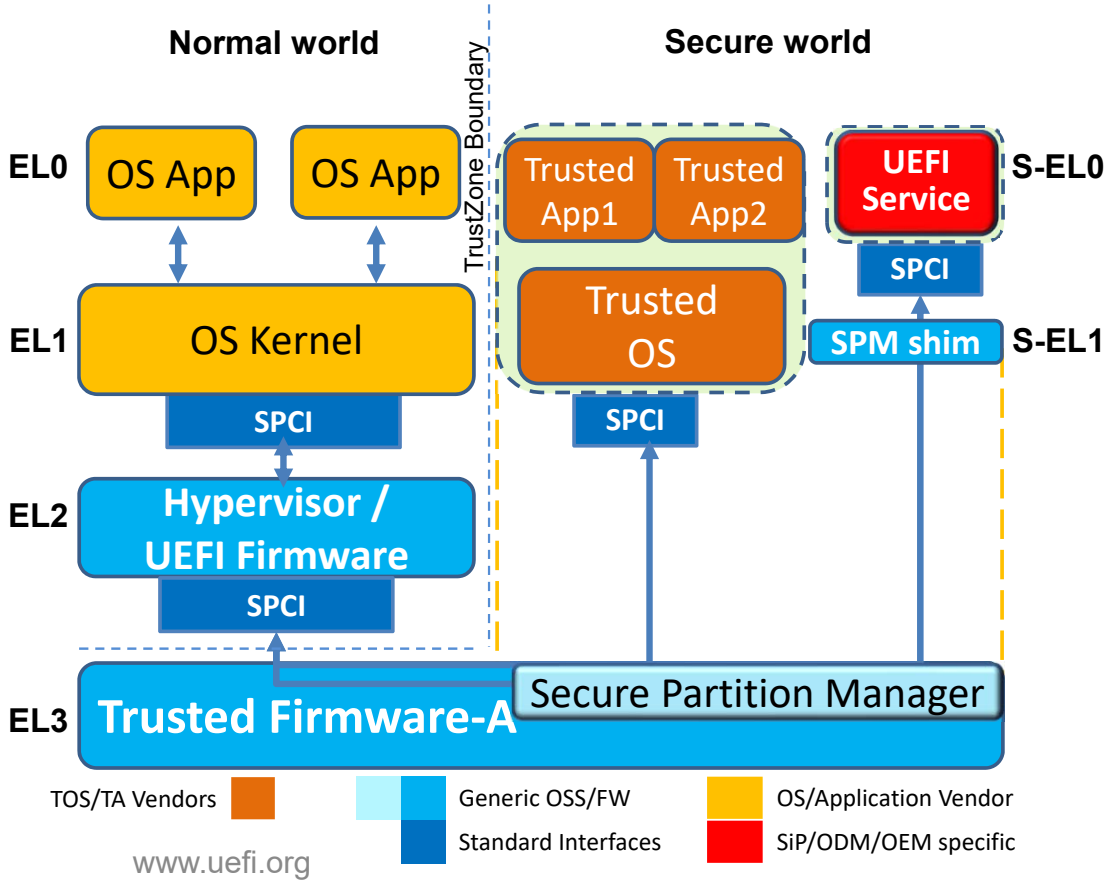
Edge: UEFI Services plus Trusted OS

Deployment requirements with mixed scenarios of:

- Trusted OS – handling dedicated Trusted Applications for specific security tasks
- Secure Partition(s) running UEFI Standalone MM services for handling separate platform functions

Secure Partition Client Interface accommodates TOS specific needs, enabling

- Standardised communication between any Secure Partition and Normal/Secure world
- Migration path towards Armv8.4 S-EL2 virtualization extension



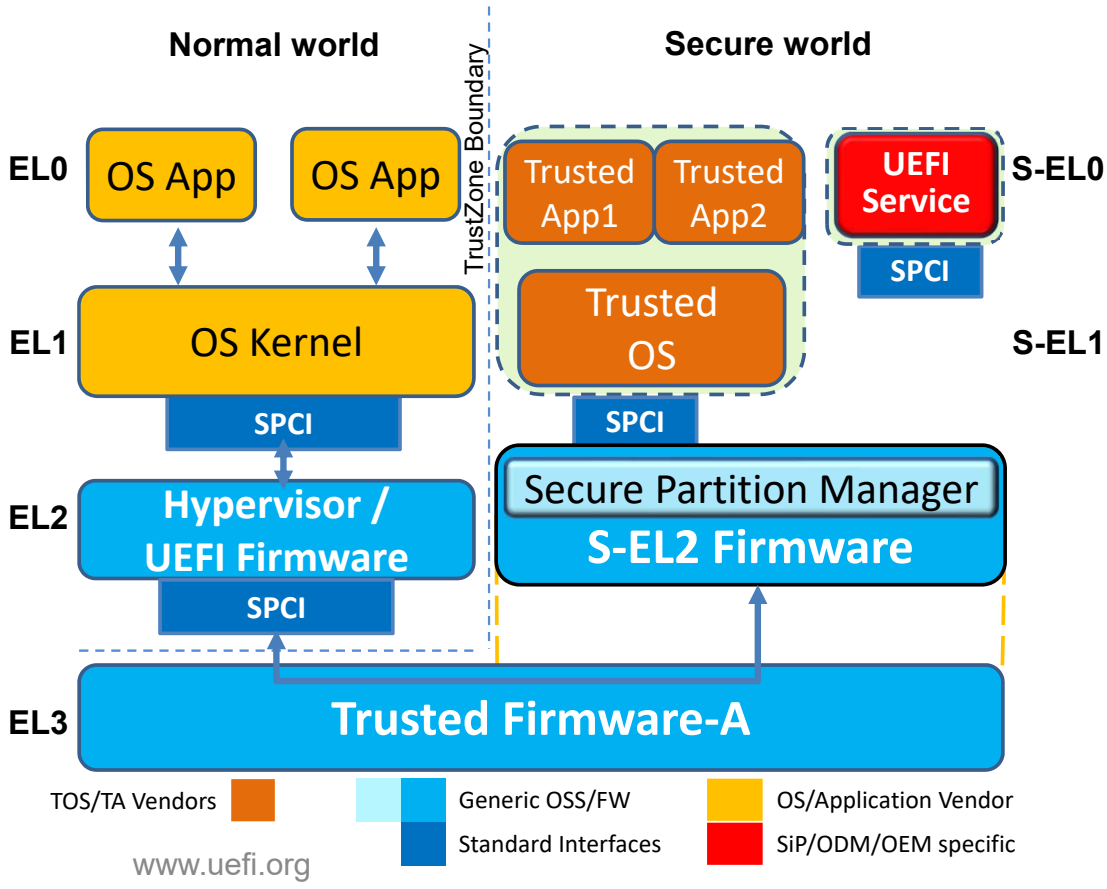
Edge: The road to HW Virtualization

Armv8.4 architecture introduces a Secure-EL2 virtualization extension

Coupled with secure SMMUv3.2 & GICv3.1 virtualization extension, this will allow HW enforced isolation and virtualization based security in the Secure world

The related Software architecture will enable scenarios with:

- TEE/TOS coexistence with Standalone MM secure services running into fully isolated Secure Partitions at either S-EL0/S-EL1

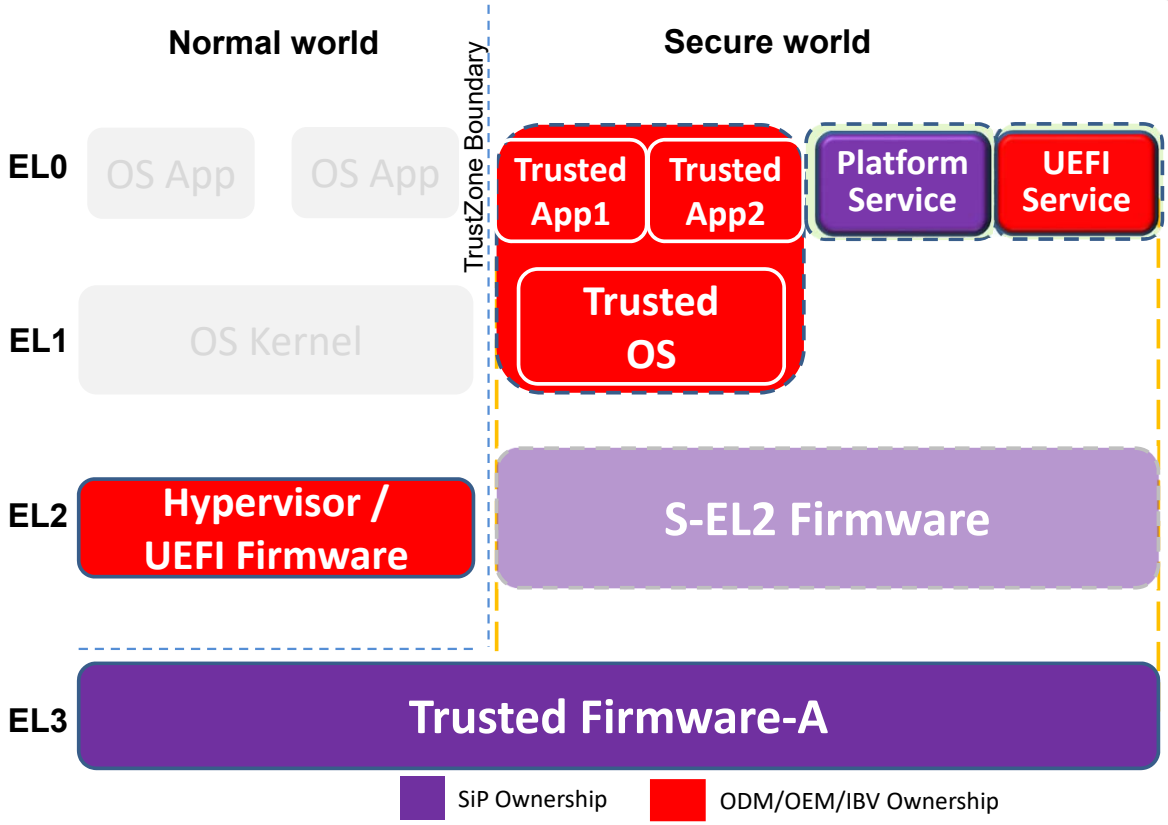


Multiple Firmware Signing Domains

Problem: different signing owners for different Secure FW entities:

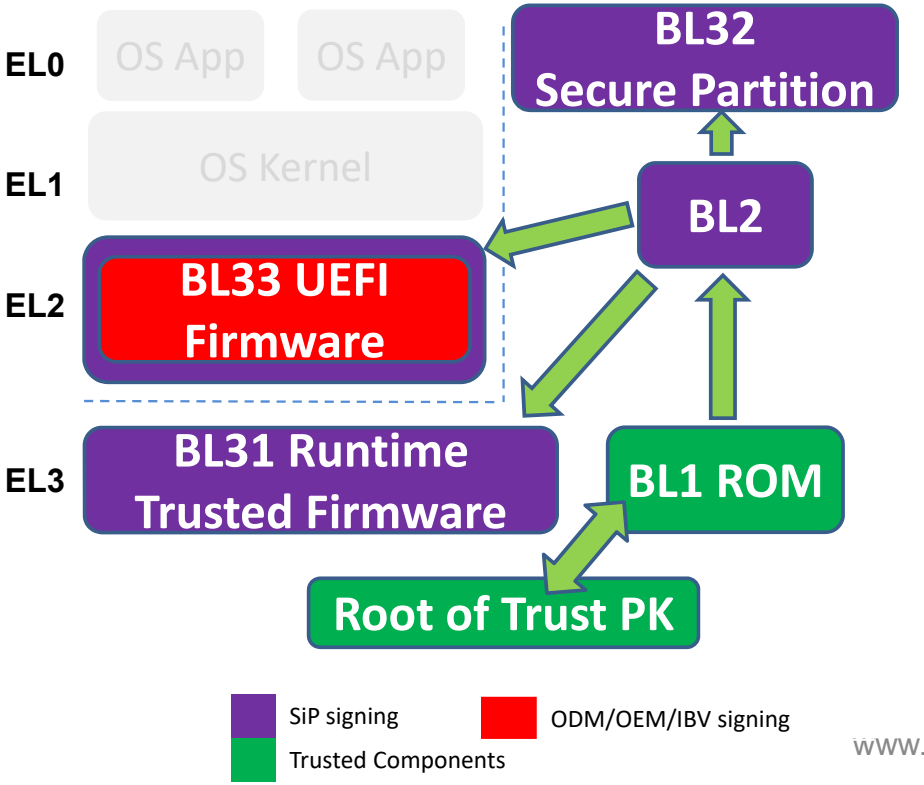
- SiP
  - Trusted Firmware-A
  - Platform Services
- ODM/OEM/IBV
  - UEFI Firmware
  - UEFI Standalone MM Services
  - (Trusted OSs + TAs)

Each signing entity might be independent from any other: no cascade-signing for independent deployment

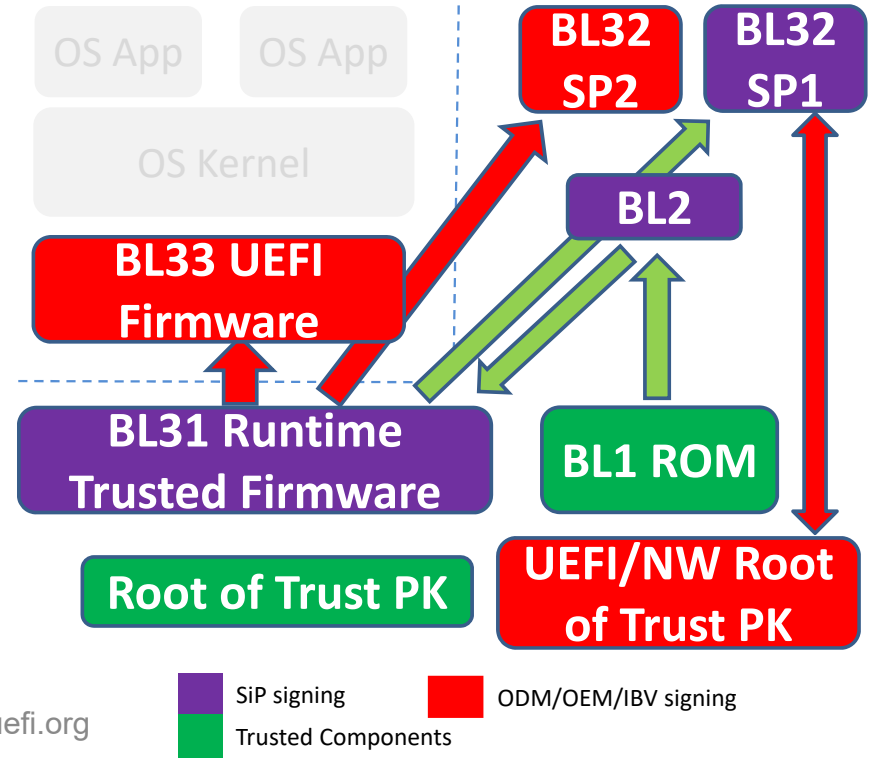


Signing Domains and Chain of Trust

Existing single cascading CoT for Trusted Boot Flow



Investigating CoT split with different signing domains on Trusted Boot



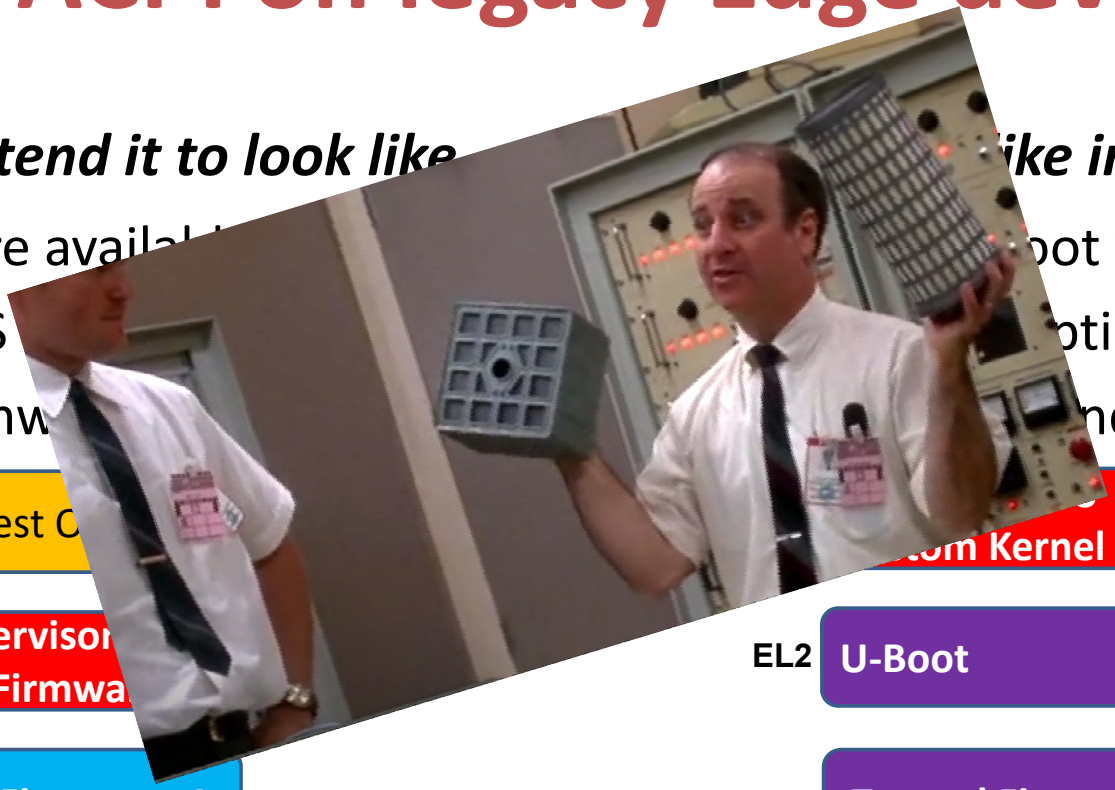
UEFI & ACPI on legacy Edge devices?

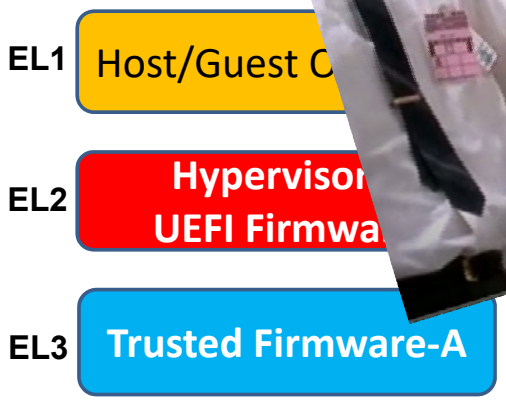
How we pretend it to look like

UEFI Firmware availa

Generic HLOS

Different Firmw





ike in reality…

oot flow, NO UEFI

otion, NO ACPI

ndled with FW