presented by

arm

UEFI updates and Secure Software Isolation on Arm

Fall 2018 UEFI Plugfest

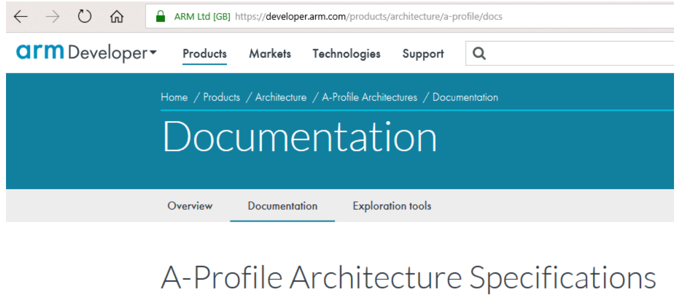
October 15 – 19, 2018

Presented by Dong Wei & Matteo Carlini (Arm)

Agenda

- UEFI SBBR & EBBR Updates
- Secure Software Status on Arm
- Secure Software Isolation architecture
- Single & Multiple Secure Partitions use-cases
- Armv8.4 Secure-EL2 virtualization extension

UEFI SBBR & EBBR Updates


ARM Ltd [GB] https://developer.arm.com/products/architecture/a-profile/docs
arm Developer
Products
Markets
Technologies
Support
Home / Products / Architecture / A-Profile Architectures / Documentation
Overview
Documentation
Exploration tools

Documentation

A-Profile Architecture Specifications

Arm Specs

- PSCI
- SMCCC
- Arm TF-A
- 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/platform-design/server-and-infrastructure

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.

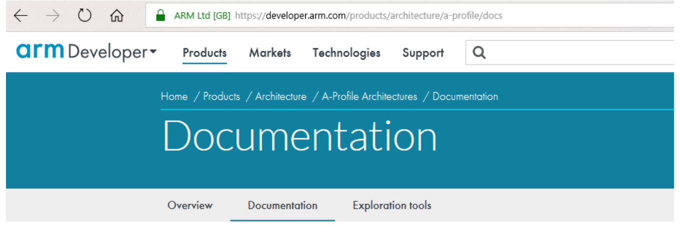
Industry Standards

- UEFI
- ACPI

- SMBIOS

- TCG FW spec

- PCI FW spec


ARM Ltd [GB] https://developer.arm.com/products/architecture/a-profile/docs
arm Developer
Products
Markets
Technologies
Support
Home / Products / Architecture / A-Profile Architectures / Documentation
Overview
Documentation
Exploration tools

Documentation

A-Profile Architecture Specifications

Arm Specs

- PSCI
- SMCCC
- Arm 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.

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

License

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License (CC-BY-SA-4.0). To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/

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.

Industry Standards

devicetree
.org

Secure Software Isolation on Arm

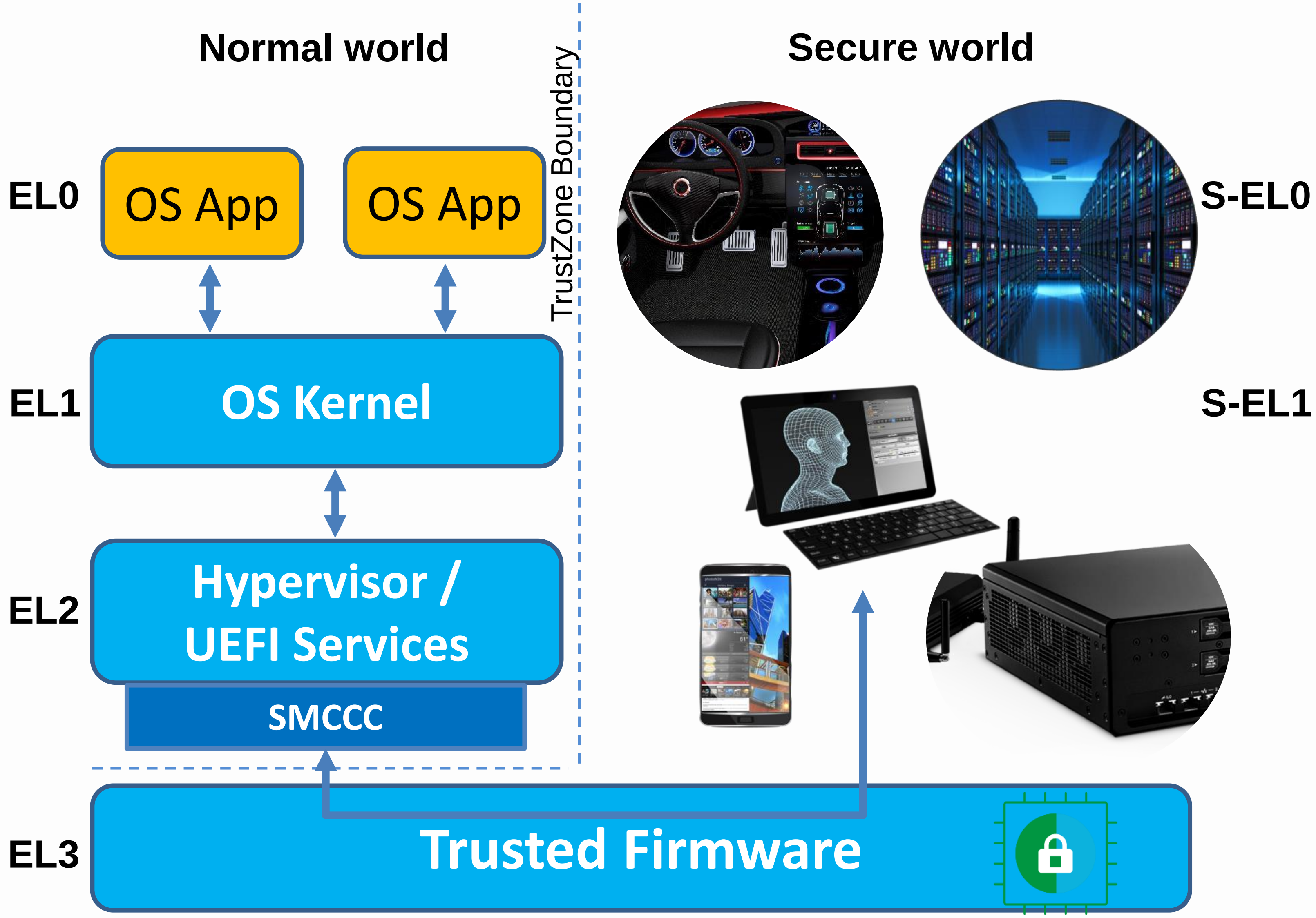
Arm Software Architecture recap

KVM

Xen®

tianocore

https://www.trustedfirmware.org/



🟦 Generic OS Software

🟦 Standard Interfaces

🟨 Application provider specific

Secure world Software Status

Highly fragmented environment

Secure services coming from different sources (silicon vendors, ODMs, OEMs, Open-source)

No isolation among different EL3, Secure-EL1, Secure-EL0 services

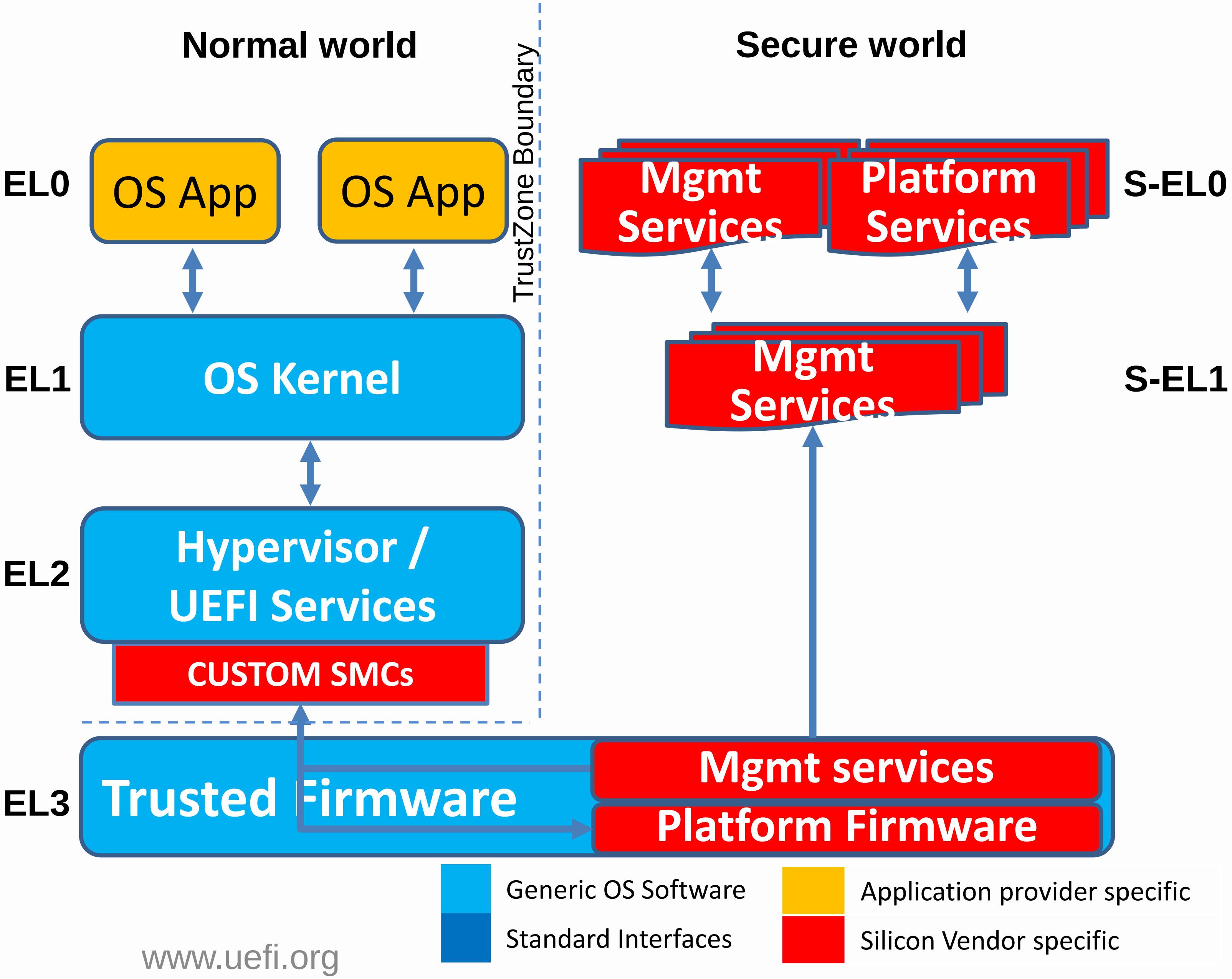
Custom SMCs & custom interfaces

- Interop problems and huge integration effort

No principle of least privilege

Firmware increased size and complexity

Security auditing becomes harder



Platform Firmware Services Use-cases

- Security related services
  - Secure storage access (UEFI Variables, Firmware Update)
  - Verified & Measured Boot (TPM / fTPM)
  - Cryptographic services
- Management Services
  - Errata handling
  - BMC communication
  - RAS Error Handling
  - System Control Processor (SCP) communication driver
  - SCP in the Secure world
- Others (RNG, …)

Secure Software Isolation

- 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

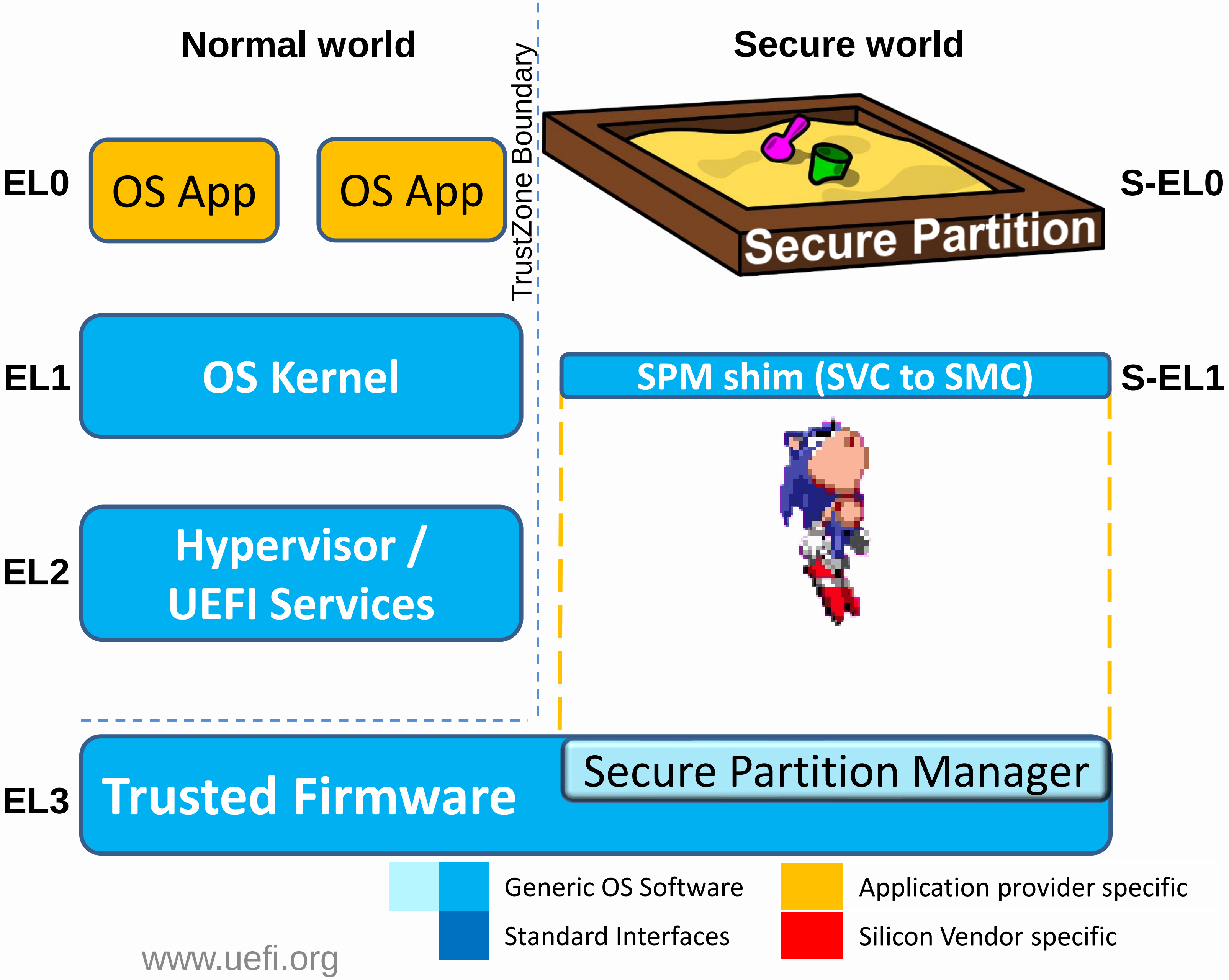
Secure Partitions Software Architecture

Secure Partition (SP):

- Unprivileged software sandbox environment running in the Secure world
- Isolated execution context
- Limited access to system resources (defined by the underlying SPM)

Secure Partition Manager (SPM):

- Runs at EL3 and owns S-EL1
- Enforces principle of least privilege
- Responsible for initializing a SP at boot time and managing runtime requests
- Responsible for enabling communication between service requestors and providers at runtime



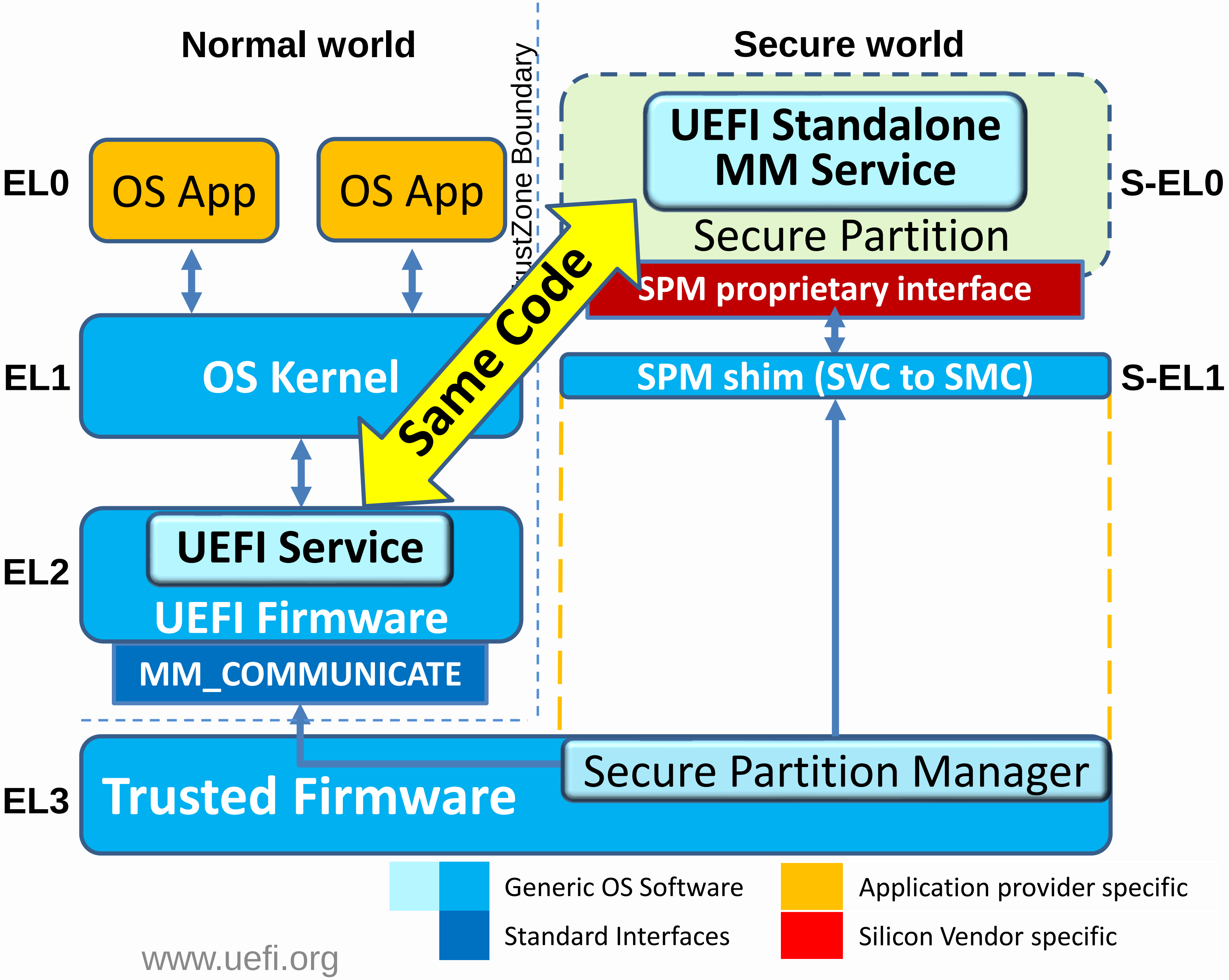
Single Partition Use-case (<=Armv8.3)

Single uniprocessor secure partition

- Execute a UEFI image with Standalone Management Mode (MM) support to execute secure management services
- Included in Trusted Firmware-A boot flow as BL32 image
- Run-to-completion runtime model
- UEFI code reuse between Normal/Secure world
- Reduced services / vendor specific code into privileged firmware (EL3)

Leverage the Arm MM Interface spec

- MM_COMMUNICATE SMC to request partition services



Multiple Partitions Use-case (<=Armv8.3)

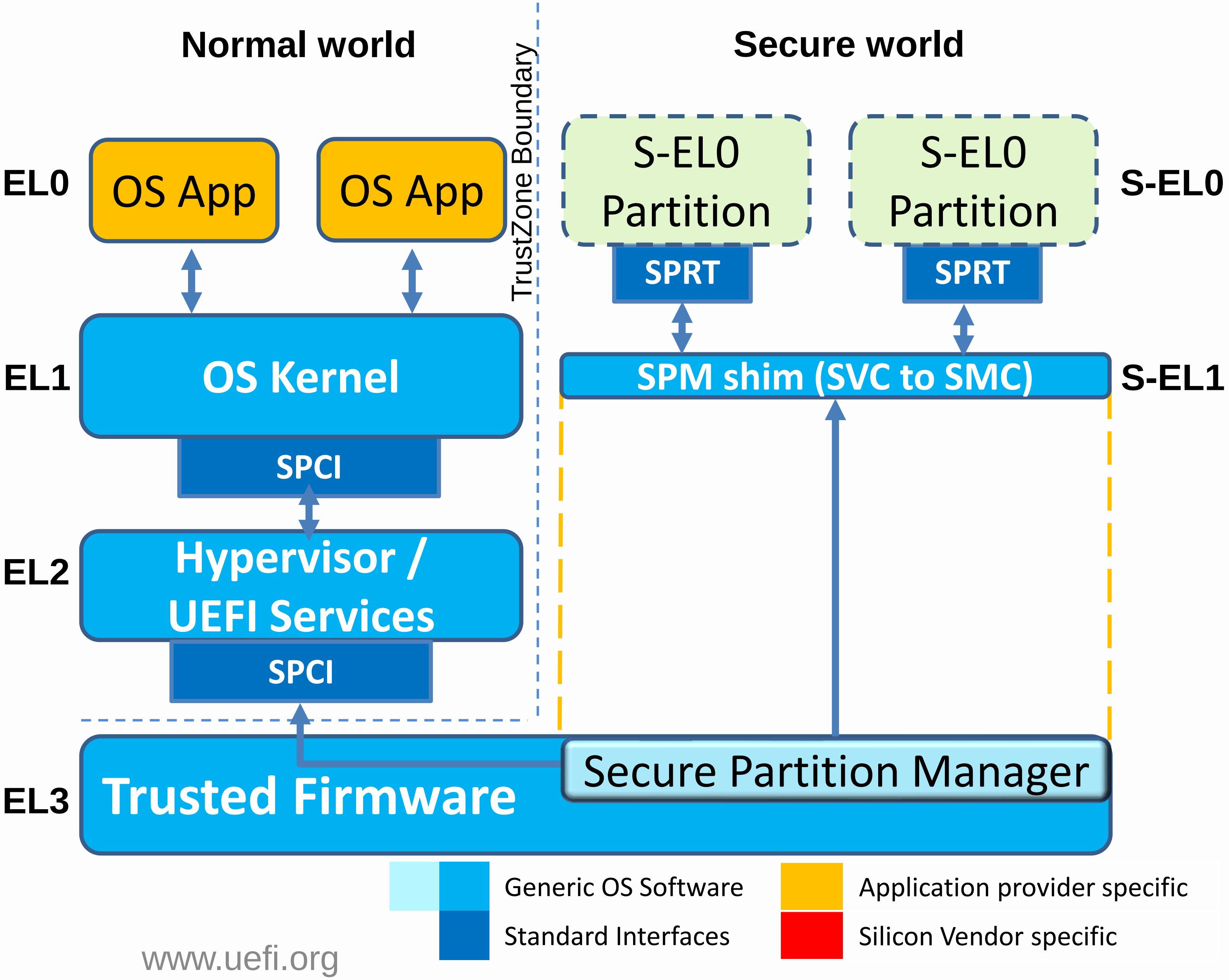
Multiple isolated Secure Partitions enabling concurrent Secure Services to run at S-EL0

Secure Partition Client Interface (SPCI)

- ABIs between Normal world clients and providers of services in Secure Partitions
- Avoids vendor specific drivers in Normal world EL2 and EL3 firmware
- Provides a SMC based transport for vendor specific drivers in Rich OS

Secure Partition Runtime interface (SPRT)

- Describes the run time model that each SP depends upon to implement secure services
- Describes ABIs between SPs and SPM to initialize SPs, dispatch requests (interrupts) to a SP and obtain responses



www.uefi.org

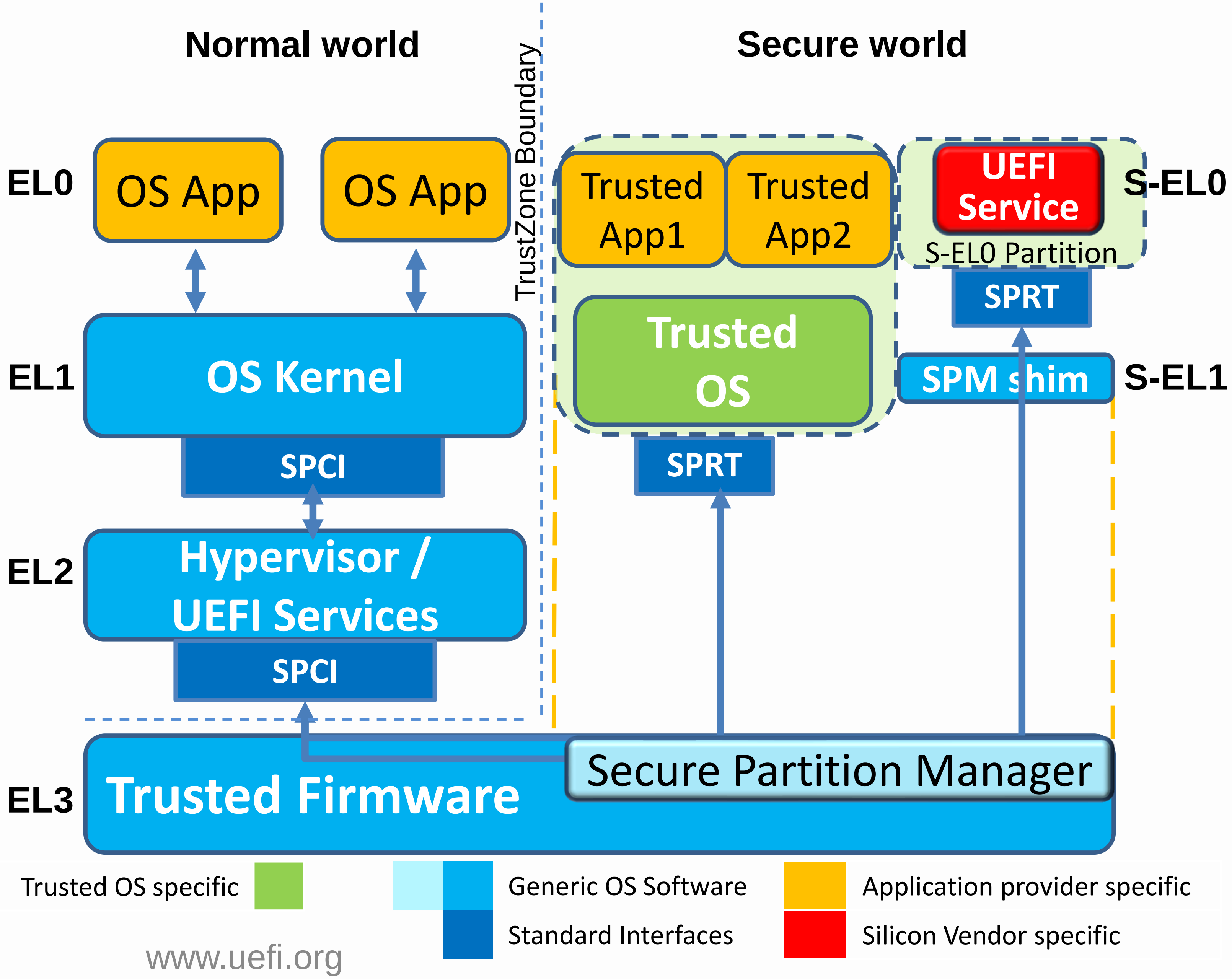
Edge/Networking Use-case (<=Armv8.3)

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 conceptually separate secure functions like Secure variable access, Firmware update

Migration path towards Armv8.4 Secure-EL2 extension

Any other input /use-cases / requirements?



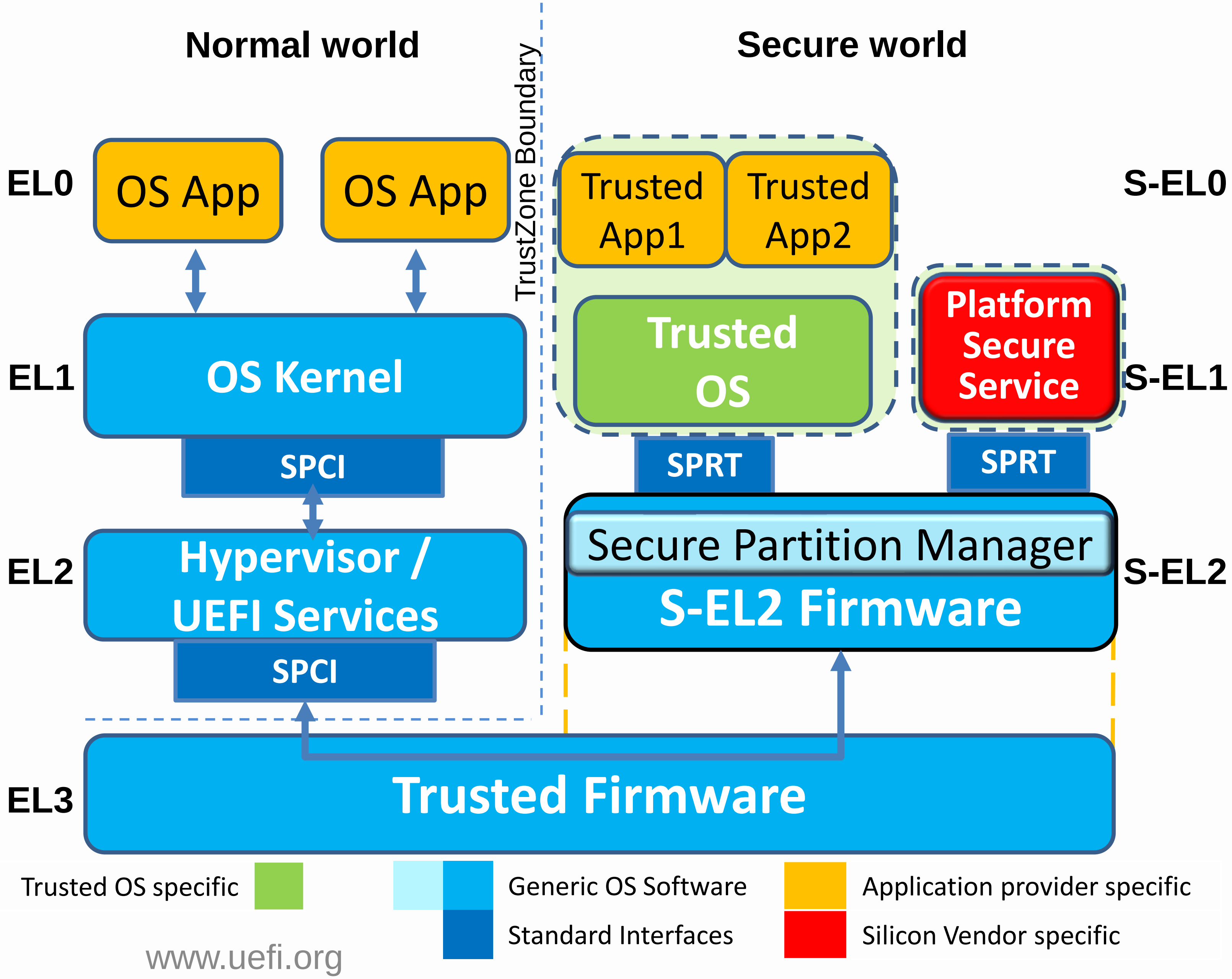
Armv8.4 S-EL2 Virtualization Extension

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



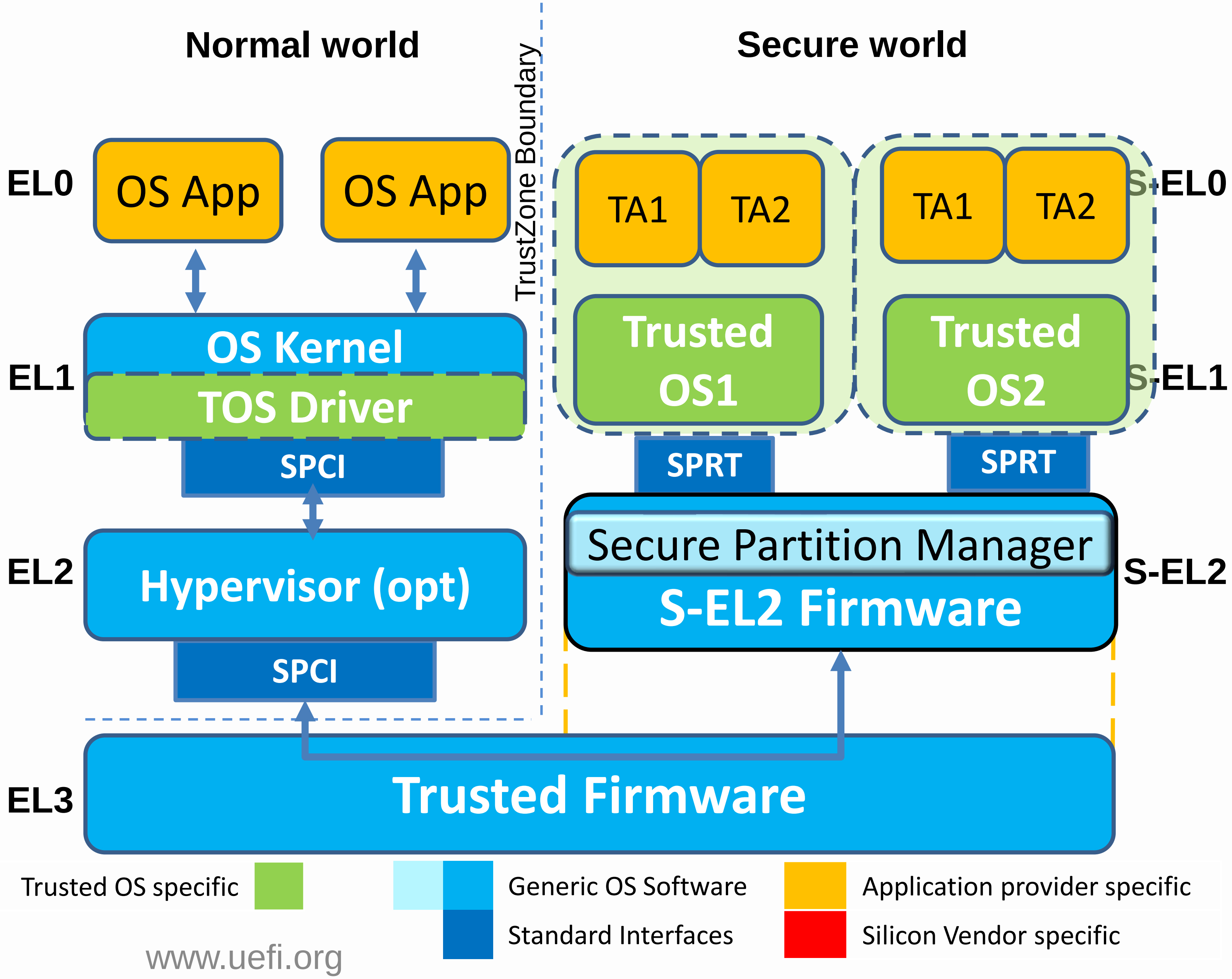
Armv8.4 S-EL2 – Multiple TEEs

Primary use-case for S-EL2 virtualization extension:

- Multiple mutually untrusted TEEs / Trusted OSs running in parallel

Mostly predominant in the mobile market segment

- Trusted OSs owned & provided by different vendors
- Diverse ownership model
- Different Trusted application providers for specific TOS



References

- Arm MM Interface Specification
  - http://infocenter.arm.com/help/topic/com.arm.doc.den0060a/DEN0060A_ARM_MM_Interface_Specification.pdf
- Trusted Firmware-A – Secure Partition Manager design document
  - https://github.com/ARM-software/arm-trusted-firmware/blob/master/docs/secure-partition-manager-design.rst
- EDK2 StandaloneMmPkg Core package
  - https://github.com/tianocore/edk2/tree/master/StandaloneMmPkg
- Secure Partition Client Interface specification Alpha1 available on DropZone
  - https://connect.arm.com/dropzone/systemarch/DEN0077A_Secure_Partition_Interface_Specification_1.0_Alpha_1.pdf
- Secure Partition Run Time Alpha specification under development
  - Expected to be available end of Oct’18
- Armv8.4 S-EL2 whitepaper available for download
  - https://community.arm.com/processors/b/blog/posts/architecting-more-secure-world-with-isolation-and-virtualization

Questions? (uefi@arm.com)

Thanks for attending the Fall 2018 UEFI Plugfest

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

presented by

arm