



arm

Standards

removing “value-less” differentiation

March 2019
GENIVI

Why do we need a standards-based approach?

Arm architecture supports a very diverse variety of devices



Diversity is good, but uncontrolled diversity is bad, particularly for servers

- Servers are very different to embedded devices – you have to install standard OSs which may even pre-date the SoC
- Installation process needs to ‘just work’
- Modifying the operating to suit the HW is not a viable option, as it is in embedded

Servers rely on standards to solve this - Common rules for hardware and for firmware

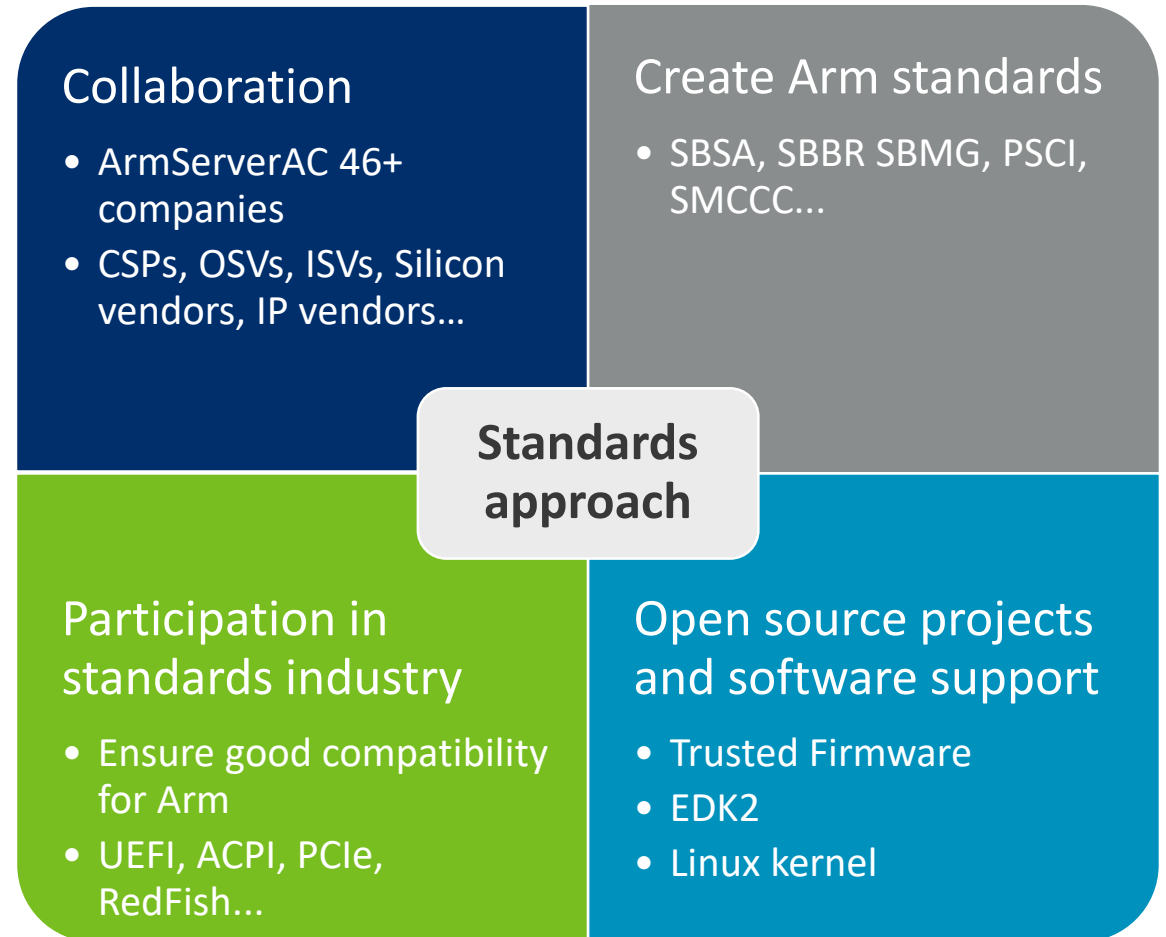
What is the Arm standards-based approach to servers

Arm takes a collaborative standards-based approach to servers

We collaborate with companies across the server ecosystem to create Arm standards for servers

We ensure existing industry standards work well with the Arm architecture

Support open source projects for software and firmware



So what's in the Server Base System Architecture?

Hardware requirements for Server SoCs, aimed at making OS “just work™”

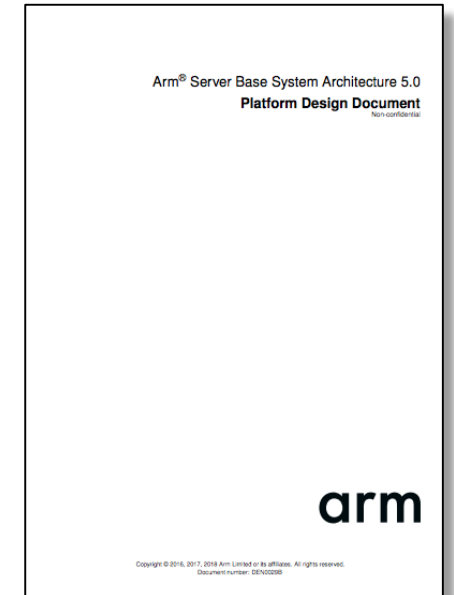
Number of levels of compliance for HW requirements in:

- Processor features and memory subsystem features
- GIC and SMMU revision features
- PCIe integration features
- Base versions for other peripherals (USB, SATA)
- Security features
- Power semantics

Levels represent the passage of time, as new architectures arrive, new levels are added

- New one every year, same as Arm ARM

Developed in the Arm Server AC: basically every company in the Arm ecosystem that's involved in server. But biggest input is from OSVs



So what's in the Server Board Boot Requirements?

Firmware requirements for Server SoCs, aimed at making OS “just work™”

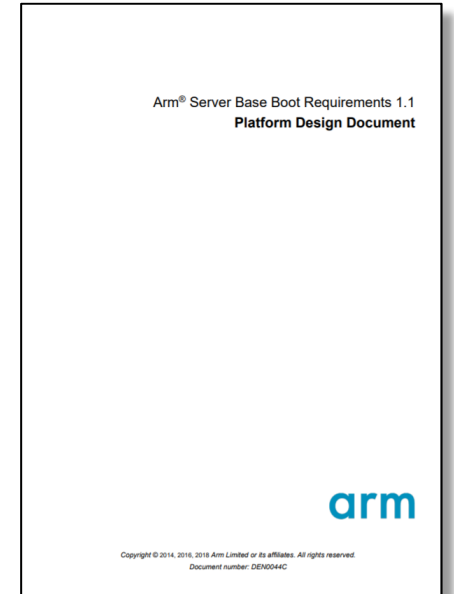
Mainly points at requirement from other FW specs

- Industry specs: ACPI, UEFI, SMBIOS
- Arm specs: PSCI, SDEI

Roughly yearly cadence matching industry spec releases

More info on SBSA and SBRR:

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



ServerReady

We provide testing and certification to check for SBSA and SBBR compliance

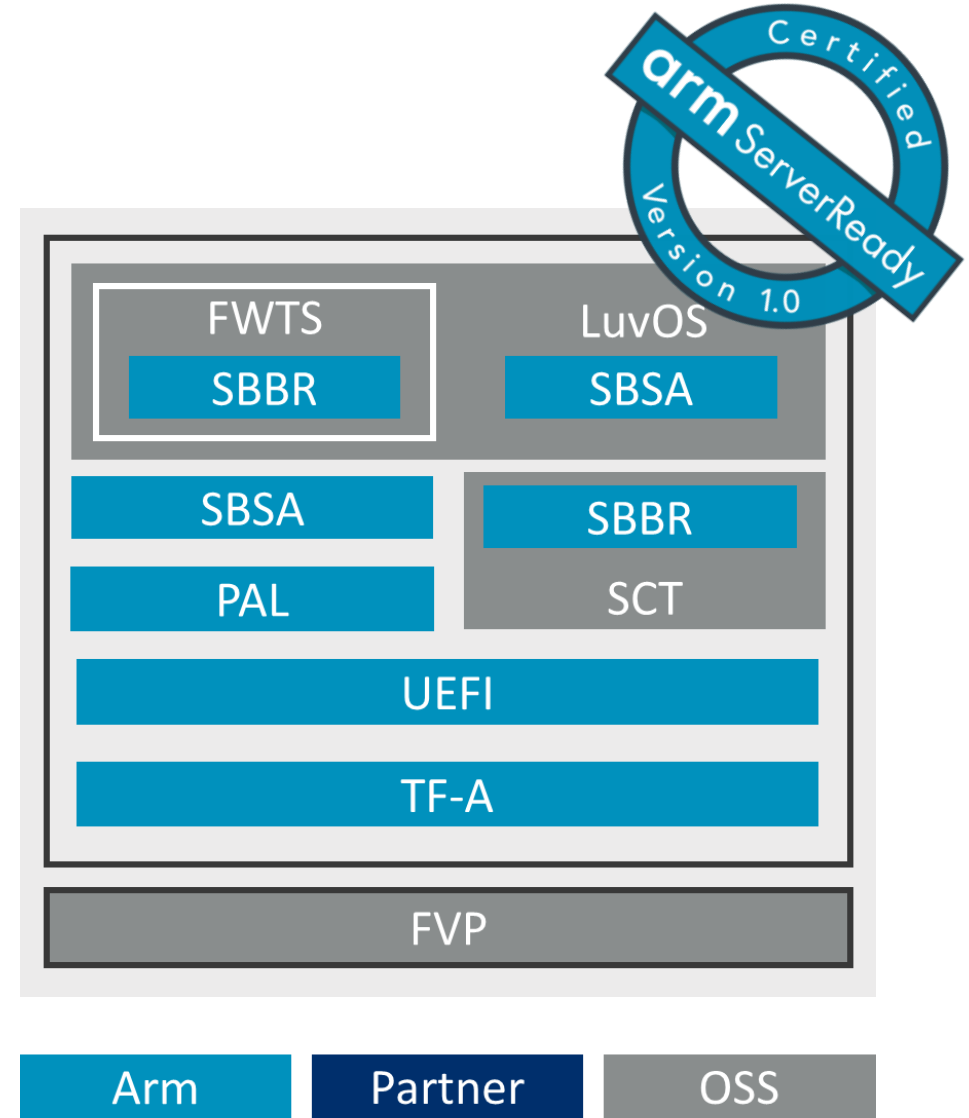
SBSA hardware requirements (CPU, GIC, SMMU, PCIe...) properties

SBBR defined FW requirements (UEFI, ACPI and SMBIOS tests)

The test suites are hosted in GitHub and are open source (Apache v2):

<https://github.com/ARM-software/sbsa-acss>

<https://github.com/ARM-software/arm-enterprise-acss>



Strong OS support



Availability across multiple architectures

Red Hat Enterprise Linux 7.5 is **simultaneously available across all supported architectures, including ... 64-bit Arm.**

SUSE Blog

It's all coming together for Arm in High Performance Computing

By: Jay Kruemcke | 2,948 views

SLES 12 for HPC is tailored for HPC workloads by including the [HPC Module](#). The HPC Module consists of a number of HPC packages that are fully supported on 64-bit Arm

Ubuntu 18.04 LTS optimised for security, multi-cloud, containers & AI



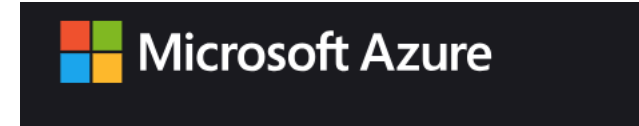
26th April 2018, London, UK: [Ubuntu 18.04 LTS](#) – the newest version of the most widely used Linux for workstations, cloud and IoT, is now available.



June 24, 2018

Oracle Linux 7 for Arm is now Generally Available

We released Oracle Linux 7 for Arm General Availability. We have been making previews available for a few months now but **the time has come to put support behind it and make clear to customers and partners that this is a real product, not just a preview.**



“We’re announcing that we are driving innovation with ARM server processors for use in our datacenters”

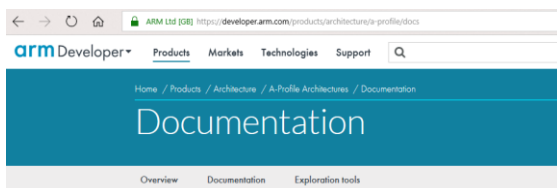
ESXi on Arm? Yes, ESXi on Arm. VMware teases bare-metal hypervisor for 64-bit Arm servers

No, we're not pulling your leg

By Chris Williams, Editor in Chief 27 Aug 2018 at 19:52

13 SHARE





A-Profile Architecture Specifications

Arm Specs

- PSCI
- SMCCC
- Arm TF

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

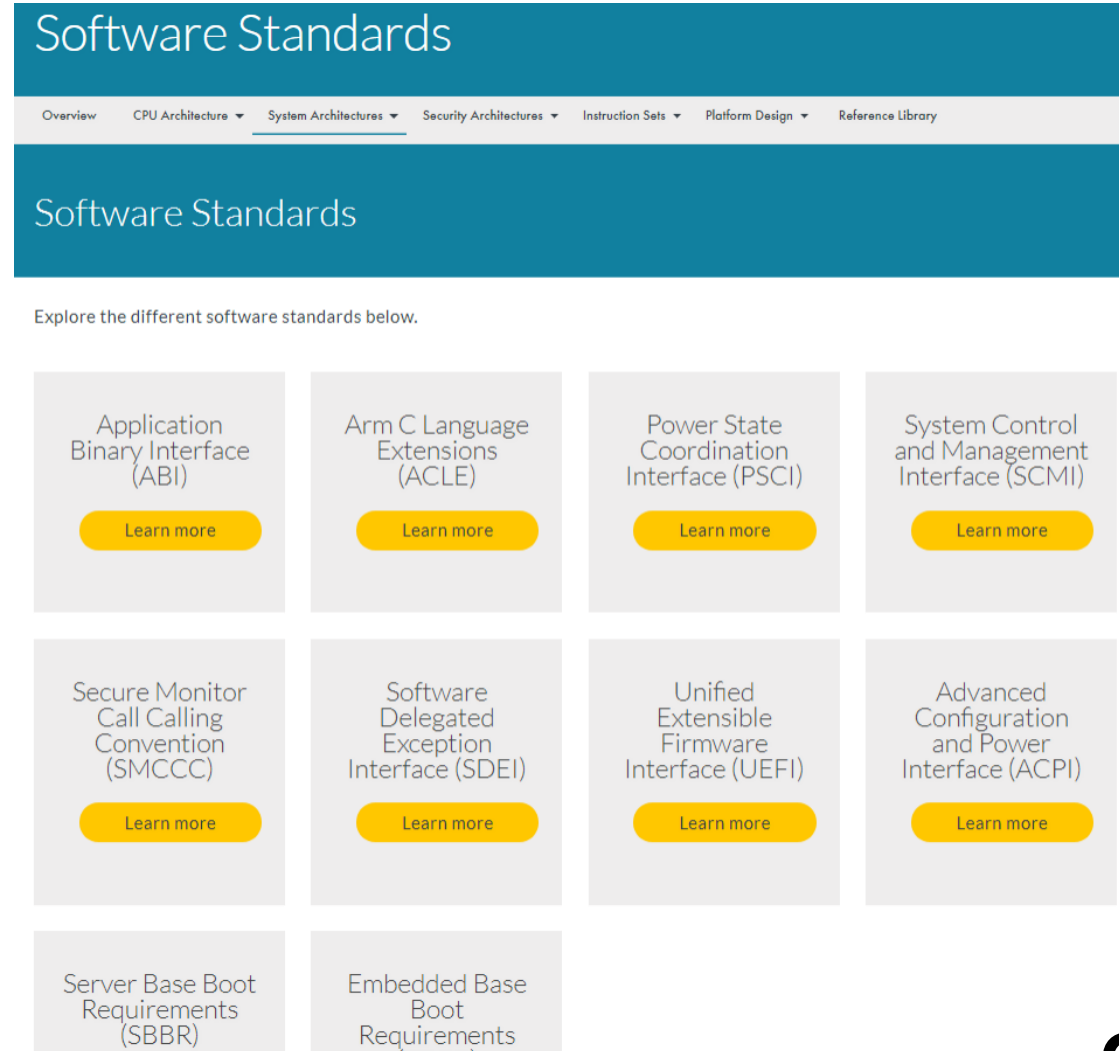


We do quite a lot more...

Developer is a good reference:

<https://developer.arm.com/products/architecture/system-architectures/software-standards>

In FW key standards are PSCI, SCMI, SDEI, and SMCCC

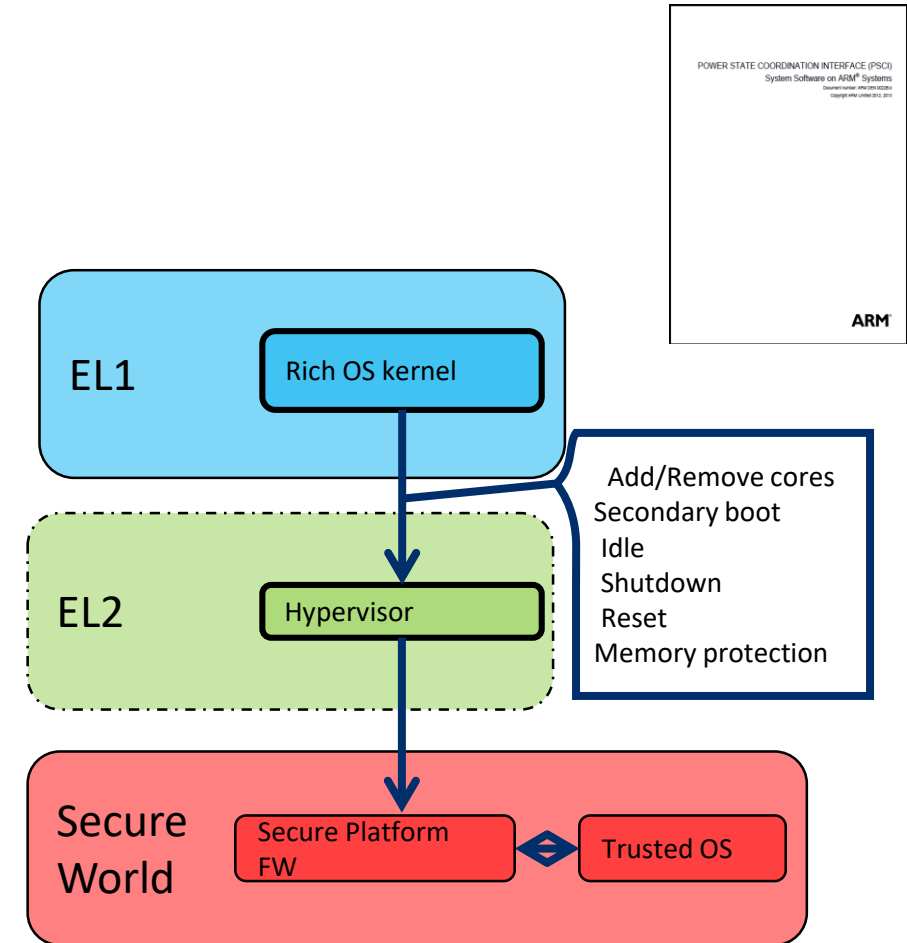


The screenshot shows the 'Software Standards' page on the Arm Developer website. The page has a dark teal header with the title 'Software Standards'. Below the header is a navigation bar with links: Overview, CPU Architecture, System Architectures (selected), Security Architectures, Instruction Sets, Platform Design, and Reference Library. The main content area has a teal background with the title 'Software Standards' and the text 'Explore the different software standards below.' Below this is a grid of 12 cards, each representing a software standard with a 'Learn more' button.

Application Binary Interface (ABI)	Arm C Language Extensions (ACLE)	Power State Coordination Interface (PSCI)	System Control and Management Interface (SCMI)
Secure Monitor Call Calling Convention (SMCCC)	Software Delegated Exception Interface (SDEI)	Unified Extensible Firmware Interface (UEFI)	Advanced Configuration and Power Interface (ACPI)
Server Base Boot Requirements (SBBR)	Embedded Base Boot Requirements (EBBR)		

Power State Coordination Interface

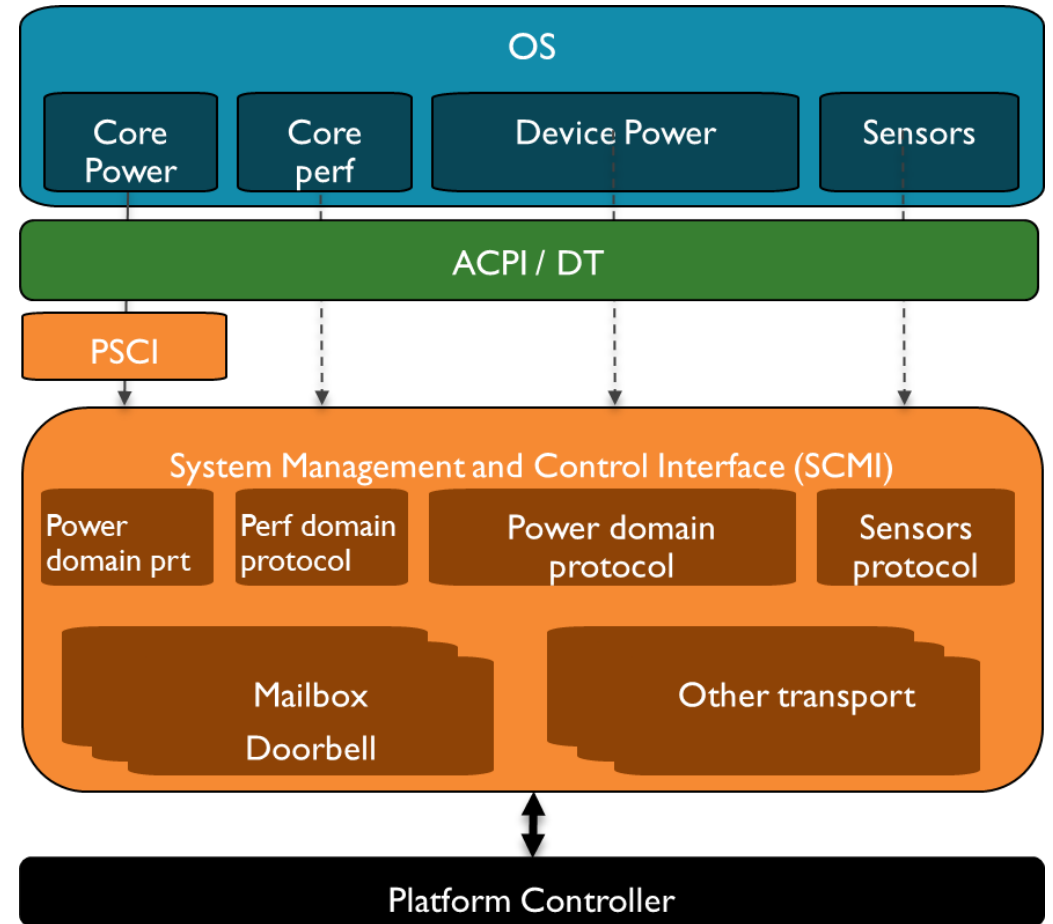
- Defines a standard interface for making power management requests across exception levels/operating systems
- Supports virtualisation and a communications with between normal and secure world
- Heavily adopted by OSVs and silicon vendors: Linux, Windows, Vmware, Xen, KVM, FreeBSD...
- Reference code in Trusted FW project : <https://www.trustedfirmware.org/>



<https://developer.arm.com/products/architecture/system-architectures/software-standards/psci>

SCMI system control and management interface

- SCMI is an standardizes common power and sensor functions
- Integrates with PSCI and provides: DVFS, device power, sensors, clock control among others
- Aimed at embedded
- Has some parallels with ACPI if you are writing power controller FW
- Aimed mainly at power controllers
- Linux support and github reference implementation for the FW



Software Delegated Exception Interface (SDEI)

Arm A-class architecture does not provide an NMI

SDEI is FW based interface that provides equivalent functionality

- At guest to hypervisor boundaries
- At hypervisor to FW boundary

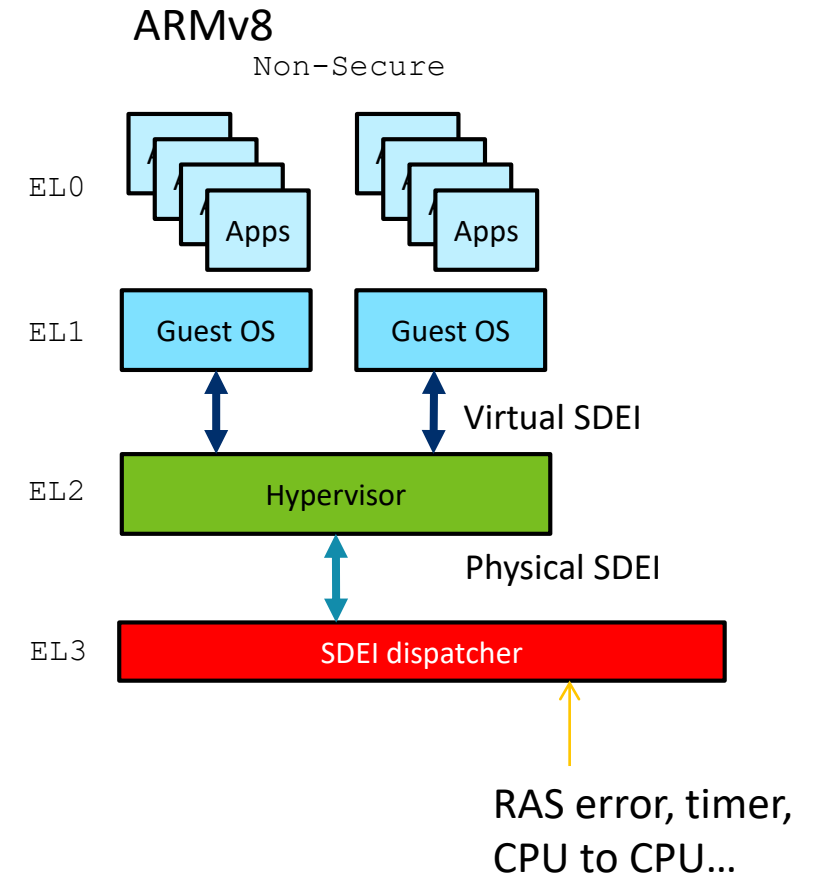
Usecases: RAS, WDog, CPU to CPU

Upstream support for SDEI in trusted FW

Basic SDEI driver support has been upstreamed

RAS still on list

<https://developer.arm.com/products/architecture/system-architectures/software-standards/sdei>



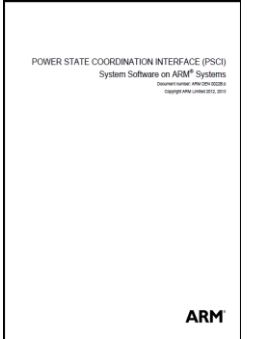
SMC Calling Convention

ABI to request runtime FW services

PSCI and SDEI sit on top of this

Partitions services and vendors that operating in Trustzone

Recently we added a FW workaround discovery process on top of this



Thank You!

Danke!

Merci!

谢谢!

ありがとう!

Gracias!

Kiitos!

arm

SBSA Level 0 - summary

Now deprecated though many requirements apply to higher levels

Processor Element (processors cores)

- Up to 8 PEs
- AArch64 at all Exception Levels
- Must to provide EL2
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 4 PMU counters, 4 breakpoints, 4 watchpoints
- PMU overflow to SPI or PPI

System level requirements

Interrupts

- GICv2 (hence 8 processor limit)
- No standard for MSI support

IO virtualization

- SMMUv1 (no PCIe ATS!)

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

Other

Power

- System specific wake up timer

Peripherals

- System specific UART
- System specific WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later

SBSA Level 0 - summary

Now deprecated though many requirements apply to higher levels

Processor Element (processors cores)

- Up to 8 PEs
- **AArch64 at all Exception Levels**
- **Must to provide EL2**
- **Advanced SIMD extensions.**
- **Instruction Caches VIPT or PIP**
- **16-bit ASID support.**
- **4KB and 64KB translation granules at stage 1 and stage 2.**
- **All PEs are coherent and in the same Inner Shareable domain.**
- **Where export restrictions allow, cryptography extensions.**
- **Little-endian support.**
- 4 PMU counters, 4 breakpoints, 4 watchpoints
- PMU overflow to SPI or PPI

System level requirements

Interrupts

- GICv2 (hence 8 processor limit)
- No standard for MSI support

IO virtualization

- SMMUv1 (no PCIe ATS!)

Memory

- **No deadlocks when accessed by processor or device**
- **MMIO Peripherals 64Kb apart**

Other

Power

- System specific wake up timer

Peripherals

- System specific UART
- System specific WDog
- **EHCI v1.0 or later**
- **XHCI v1.0 or later**
- **AHCI v1.3 or later**

SBSA Level 1 - summary

Now deprecated though many requirements apply to higher levels

Processor Element (processors cores)

- Up to 8 PEs
- **AArch64 at all Exception Levels**
- **Must to provide EL2**
- **Advanced SIMD extensions.**
- **Instruction Caches VIPT or PIP**
- **16-bit ASID support.**
- **4KB and 64KB translation granules at stage 1 and stage 2.**
- **All PEs are coherent and in the same Inner Shareable domain.**
- **Where export restrictions allow, cryptography extensions.**
- **Little-endian support.**
- **6 PMU counters, 6 breakpoints, 4 watchpoints**
- **PMU overflow to SPI or PPI**

System level requirements

Interrupts

- GICv2 (hence 8 processor limit)
- **GICv2-M to support MSI (minimal of 32)**

IO virtualization

- SMMUv1 (no PCIe ATS!)

Memory

- **No deadlocks when accessed by processor or device**
- **MMIO Peripherals 64Kb apart**

Other

Power

- **Arm standard wake up timer**

Peripherals

- **Arm standard UART**
- **Arm standard WDog**
- **EHCI v1.0 or later**
- **XHCI v1.0 or later**
- **AHCI v1.3 or later**

SBSA Level 1 - summary

Now deprecated though many requirements apply to higher levels

Processor Element (processors cores)

- Up to 8 PEs
- **AArch64 at all Exception Levels**
- **Must to provide EL2**
- **Advanced SIMD extensions.**
- **Instruction Caches VIPT or PIP**
- **16-bit ASID support.**
- **4KB and 64KB translation granules at stage 1 and stage 2.**
- **All PEs are coherent and in the same Inner Shareable domain.**
- **Where export restrictions allow, cryptography extensions.**
- **Little-endian support.**
- **6 PMU counters, 6 breakpoints, 4 watchpoints**
- **PMU overflow to SPI or PPI**

System level requirements

Interrupts

- GICv2 (hence 8 processor limit)
- GICv2-M to support MSI (minimal of 32)

IO virtualization

- SMMUv1 (no PCIe ATS!)

Memory

- **No deadlocks when accessed by processor or device**
- **MMIO Peripherals 64Kb apart**

Other

Power

- **Arm standard wake up timer**

Peripherals

- **Arm standard UART**
- **Arm standard WDog**
- **EHCI v1.0 or later**
- **XHCI v1.0 or later**
- **AHCI v1.3 or later**

SBSA Level 2 - summary

Now deprecated though many requirements apply to higher levels

Processor Element (processors cores)

- **Up to 2²⁸ PEs**
- **AArch64 at all Exception Levels**
- **Must to provide EL2**
- **Advanced SIMD extensions.**
- **Instruction Caches VIPT or PIP**
- **16-bit ASID support.**
- **4KB and 64KB translation granules at stage 1 and stage 2.**
- **All PEs are coherent and in the same Inner Shareable domain.**
- **Where export restrictions allow, cryptography extensions.**
- **Little-endian support.**
- **6 PMU counters, 6 breakpoints, 4 watchpoints**

System level requirements

Interrupts

- **GICv3 (generic MSI support!)**
- **Standard interrupt numbers**

IO virtualization

- **SMMUv2**

Memory

- **No deadlocks when accessed by processor or device**
- **MMIO Peripherals 64Kb apart**

Other

Power

- **Arm standard wake up timer**

Peripherals

- **Arm standard UART**
- **Arm standard WDog**
- **EHCI v1.0 or later**
- **XHCI v1.0 or later**
- **AHCI v1.3 or later**

SBSA Level 3 - summary

Most systems today are level 3 compliant

Processor Element (processors cores)

- Up to 2²⁸ PEs
- AArch64 at all Exception Levels
- Must to provide EL2 and EL3
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 6 PMU counters, 6 breakpoints, 4 watchpoints
- CRC32, cond. Scalar vector ext.

System level requirements

Interrupts

- GICv3 (generic MSI support!)
- Standard interrupt numbers

IO virtualization

- SMMUv2 or SMMUv3 (ATS!)

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

Other

Power

- Arm standard wake up timer

Peripherals

- Arm standard UART
- Arm standard WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later

Firmware:

- Optional level 3 FW secure wake up timer, WDog and UART

SBSA Level 3 - summary

Most systems today are level 3 compliant

Processor Element (processors cores)

- Up to 2²⁸ PEs
- AArch64 at all Exception Levels
- Must to provide EL2 and EL3
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 6 PMU counters, 6 breakpoints, 4 watchpoints
- CRC32, cond. Scalar vector ext.

System level requirements

Interrupts

- GICv3 (generic MSI support!)
- Standard interrupt numbers

IO virtualization

- SMMUv2 or SMMUv3 (ATS!)

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

Other

Power

- Arm standard wake up timer

Peripherals

- Arm standard UART
- Arm standard WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later

Firmware:

- Optional level 3 FW secure wake up timer, WDog and UART

SBSA Level 4 - summary

Next generation mainly being built to this or level 5

Processor Element (processors cores)

- Up to 2²⁸ PEs
- AArch64 at all Exception Levels
- Must to provide EL2 and EL3
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 6 PMU counters, 6 breakpoints, 4 watchpoints
- CRC32, cond. Scalar vector ext.

- Standard RAS
- 16 bit VMID
- Virtual host extension
- Cond. pointer signing
- Cond. Clean to point of serialization

System level requirements

Interrupts

- GICv3 (generic MSI support!)
- Standard interrupt numbers

IO virtualization

- SMMUv3 (ATS!)
- PCIe for all assignable devices

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

Other

Power

- Arm standard wake up timer

Peripherals

- Arm standard UART
- Arm standard WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later

SBSA Level 4 - summary

Next generation mainly being built to this or level 5

Processor Element (processors cores)

- Up to 2^{28} PEs
- AArch64 at all Exception Levels
- Must to provide EL2 and EL3
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 6 PMU counters, 6 breakpoints, 4 watchpoints
- CRC32, cond. Scalar vector ext.

- Standard RAS
- 16 bit VMID
- Virtual host extension
- Cond. pointer signing
- Cond. Clean to point of serialization

System level requirements

Interrupts

- GICv3 (generic MSI support!)
- Standard interrupt numbers

IO virtualization

- SMMUv3 (ATS!)
- PCIe for all assignable devices

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

Other

Power

- Arm standard wake up timer

Peripherals

- Arm standard UART
- Arm standard WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later

SBSA Level 5 - summary

Next generation mainly being built to this or level 4

Processor Element (processors cores)

- Up to 2^{28} PEs
- AArch64 at all Exception Levels
- Must to provide EL2 and EL3
- Advanced SIMD extensions.
- Instruction Caches VIPT or PIP
- 16-bit ASID support.
- 4KB and 64KB translation granules at stage 1 and stage 2.
- All PEs are coherent and in the same Inner Shareable domain.
- Where export restrictions allow, cryptography extensions.
- Little-endian support.
- 6 PMU counters, 6 breakpoints, 4 watchpoints
- CRC32, cond. Scalar vector ext.

- Standard RAS
- 16 bit VMID
- Virtual host extension
- Cond. pointer signing
- Cond. Clean to point of serialization
- Nested virtualization improvements
- Pointer signing
- Coresight standard support
- Cond. MPAM
- Crypto additions (SHA3/512 SM3/4)
- Activity monitor extension

System level requirements

Interrupts

- GICv3 (generic MSI support!)
- Standard interrupt numbers
- No non-standard extensions

IO virtualization

- SMMUv3 (ATS!)
- PCIe for all assignable devices

Memory

- No deadlocks when accessed by processor or device
- MMIO Peripherals 64Kb apart

System counter scaling

Other

Power

- Arm standard wake up timer

Peripherals

- Arm standard UART
- Arm standard WDog
- EHCI v1.0 or later
- XHCI v1.0 or later
- AHCI v1.3 or later