

presented by



Standardized Firmware for ARMv8 based Volume Servers

UEFI Spring Plugfest – March 29-31, 2016

Presented by Jonathan Zhang, Robert Hsu

Cavium Inc. & AMI

Agenda



- Why standardized FW
- Standardized FW for system features
- Standardized FW engineering process
- Option ROM Challenge
- Questions



Why standardized FW




Market for ARMv8 volume servers



HPC

A circular inset image showing various high-performance computing (HPC) server components, including server racks and individual server units.

Cloud Compute

A circular inset image showing a server rack with a server unit, representing cloud computing infrastructure.

Telco

A circular inset image showing a server rack with a server unit, representing telecommunications infrastructure.

Storage

A circular inset image showing a server rack with a server unit, representing storage infrastructure.

OCP

A circular inset image showing a server rack with a server unit, representing Open Compute Project (OCP) infrastructure.

Web Hosting

A circular inset image showing a server rack with a server unit, representing web hosting infrastructure.

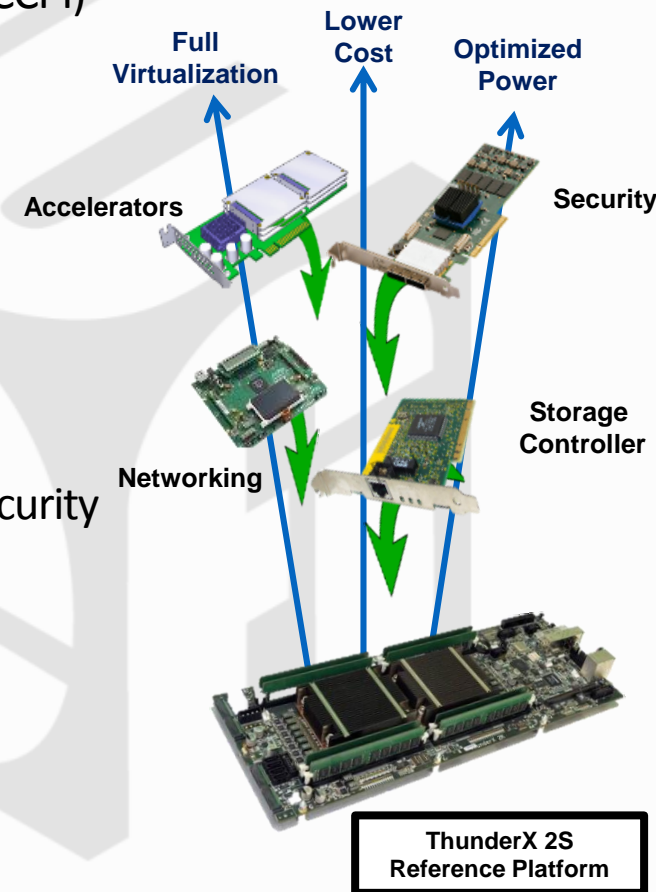
Implementation of ARMv8-A Cavium ThunderX[®]



- Up to 48 full custom ARMv8 cores
- Multi-socket capable with Cavium Cache Coherent Interconnect (CCPI)
- Up to 4x 72-bit DDR3/4 Memory Controllers
 - 1 TB system memory in 2S config
- Family Specific I/O's including 40G/10GE, PCIe Gen3, SATA 6G
- Standards based low latency Ethernet fabric
- virtSOC™: Virtualization from Core to I/O
- Platform : Single & Dual Socket
- Family Specific Accelerators : Storage/Networking / Compute / Security

The benefits of this Workload Specific approach

- Efficiency (Performance, Latency, Power, and Scalability)
- Best in Class Optimized solution for the specific workload

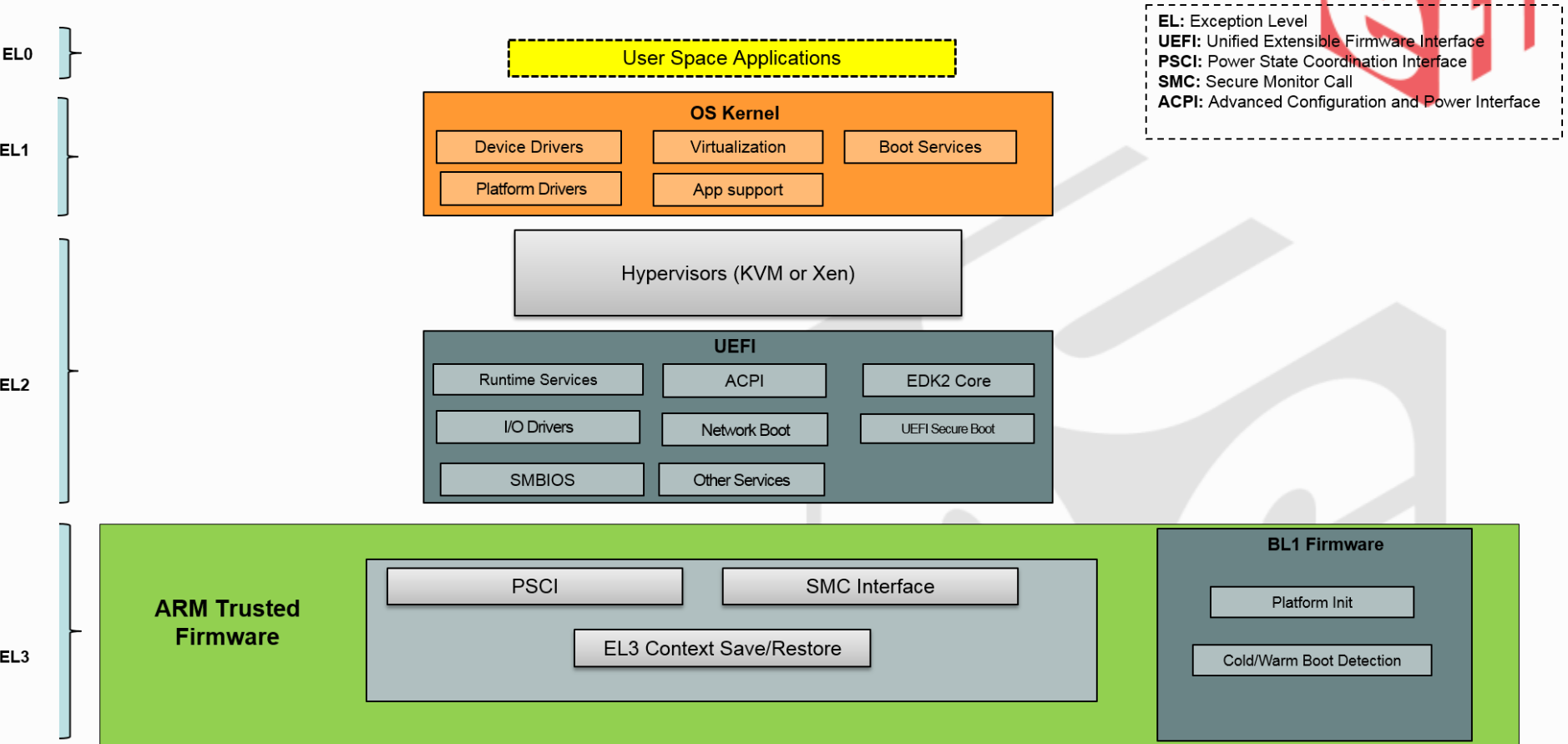


AMI – Cavium partnership



- ThunderX Enablement - UEFI based off of Tianocore model
 - Upstream code collaboration and integration
 - Engagement directly w/ Linaro, ARM and broader community
 - Limited features and no support
- Full commercial model w/ AMI – active deployment w/ ODMs
 - Single Socket and Dual Socket support
 - Full system management via MegaRAC software and tools
 - Out of Band MGT via IPMI integrated w/ UEFI/ACPI boot
 - Foundation for all future designs – OEM support in place
- Field collaboration & Customer support fully enabled
 - Collaboration on customer designs
 - Worldwide field teams fully trained and ramped up

Standardized SW Architecture



To-do: Add Secure EL1 and Secure EL0.

UEFI/ACPI Published Standards



UEFI FORUM ANNOUNCES THE AVAILABILITY OF UPDATED SPECIFICATIONS: UEFI V2.6 AND ACPI V6.1

Wednesday, March 9, 2016

New specifications enhance ever-growing mobility and manageability of computing systems for consumer and enterprise levels.

Beaverton, Ore.—March 9, 2016—Today, the UEFI Forum announced availability of the Advanced Configuration and Power Interface (ACPI) Specification v6.1 and the Unified Extensible Firmware Interface (UEFI) Specification v2.6. The new specifications continue to advance by keeping pace with market demand for enhanced mobility and manageability of computing systems for customer and enterprise levels.

“UEFI and ACPI specification enhancements will ripple through the industry by expanding support for new hardware, new platforms and OS designs,” states Mark Doran, president, UEFI Forum. “Our target platforms are the building blocks of embedded, business and personal computing ecosystems, and the UEFI v2.6 and ACPI v6.1 reinforce our mission to modernize the booting and power management processes.”

The ACPI Specification v6.1 now includes:

- Interrupt-signaled events for expanded hardware-reduced platform support and improved system-on-chip designs.
- Standardized ARMv8-A processor support for “firmware-first” hardware error handling and reporting, including SEA and SEI notification types in the Hardware Error Source Table (HEST).

UEFI Specification v2.6 now includes:

- Enriched ability for agents in the system to provide better user interface support prior to launching of the OS through additional Image and Font information.
- Formal API definition for RAM Disk Protocol.
- New Wireless MAC Connection Protocol interface simplifies wireless network support and future radio technology versions.
- ARM error reporting extensions for the Common Platform Error Record (CPER), allowing ARMv8-A systems to implement “firmware-first” hardware error handling and reporting.

Download the specifications [here](#) to learn more about these and other updates.

ARMv8 Server Standards



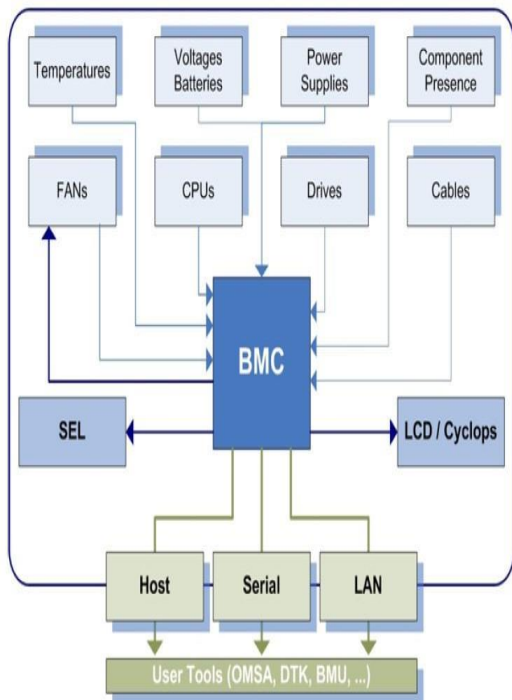
- ARM SBBR (Server Base Boot Requirements), ver. 1.0,
<http://infocenter.arm.com/help/topic/com.arm.doc.den0044b/index.html>
- ARM SBSA (Server Base System Architecture), ver. 3,
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0029/index.html>,
available to registered ARM customers.



Section Heading

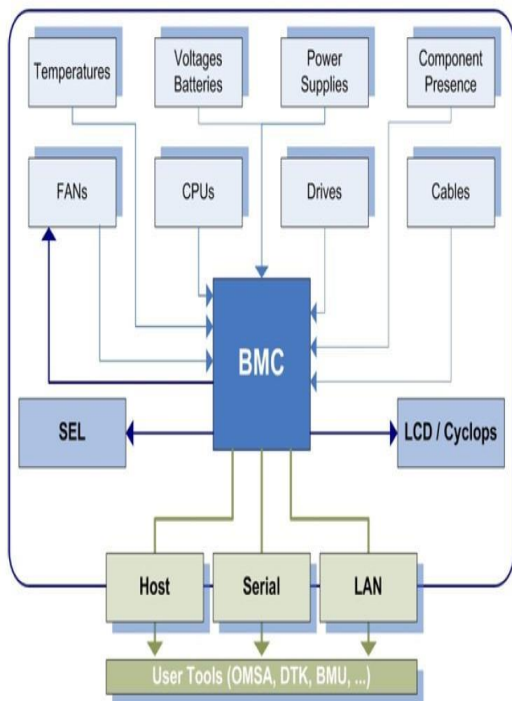
Standardized FW for system features

Standardized FW for out of band management



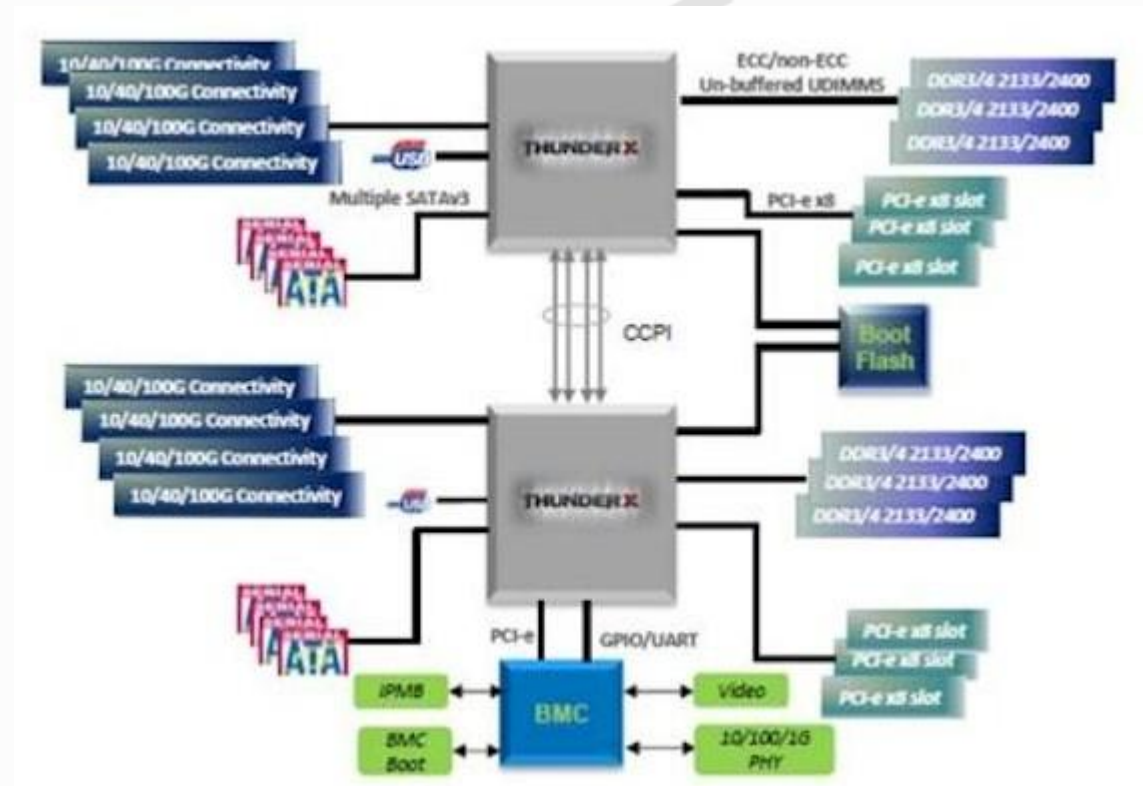
- BMC features to enable out of band management
 - IPMI 2.0
 - Serial over Lan
 - SSIF
 - CPLD Flashing (takes less than a minute)
 - Centralized Fan control
 - Centralized Web interface to monitor and manage the system health.
 - Remote Media redirection
 - Remote KVM with complete console redirection

Standardized FW for out of band management



- System design goal of minimizing need for BMC customization due to board/rack configuration difference
 - CPU SW (OS and FW) supports server management standards (IPMI, smbios, etc.) to allow for standardized BMC FW.
 - SoC specific processing/functions are done in CPU SW.
 - BMC not participating in system functions (such as host initiated shutdown), except of monitoring, reporting functions.
- Lack of standardization of how in-band and out-of-band management work together, needs to be addressed.

Standardized FW for dual socket



Standardized FW for dual socket



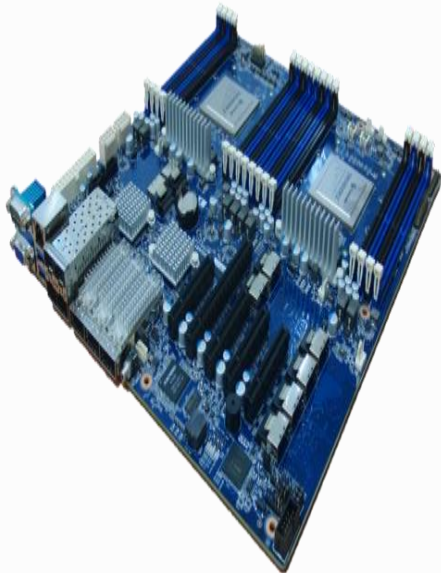
- NUMA support for arm64 platforms in Linux based on ACPI SRAT/SLIT tables.
- Improvement (Linux) in progress.

Standardized FW for PCIe devices

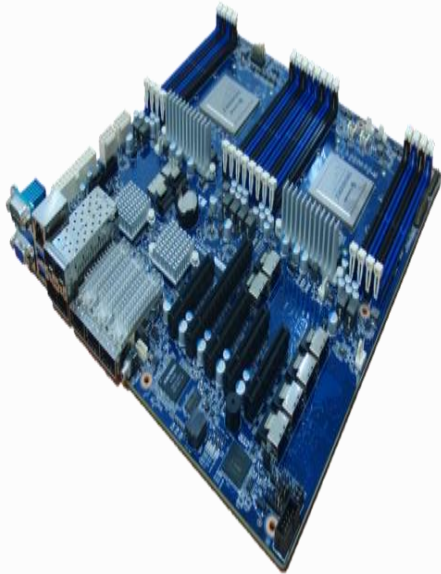


Supported PCIe devices:

- VNIC devices
- Storage devices
- Bootable PCIe devices
- NVMe based PCIe storage devices
- PCIe devices with OptionROM



Standardized FW for PCIe devices



- Achieved:
 - Enumeration of PCIe devices in UEFI shell.
 - PCIe devices configuration in UEFI shell.
 - Full functioning of PCIe devices in OS.
- To-do:
 - More standardization of PCIe in SBSA.
 - PCIe device description (for ARM based server) in ACPI.

Standardized FW for RAS



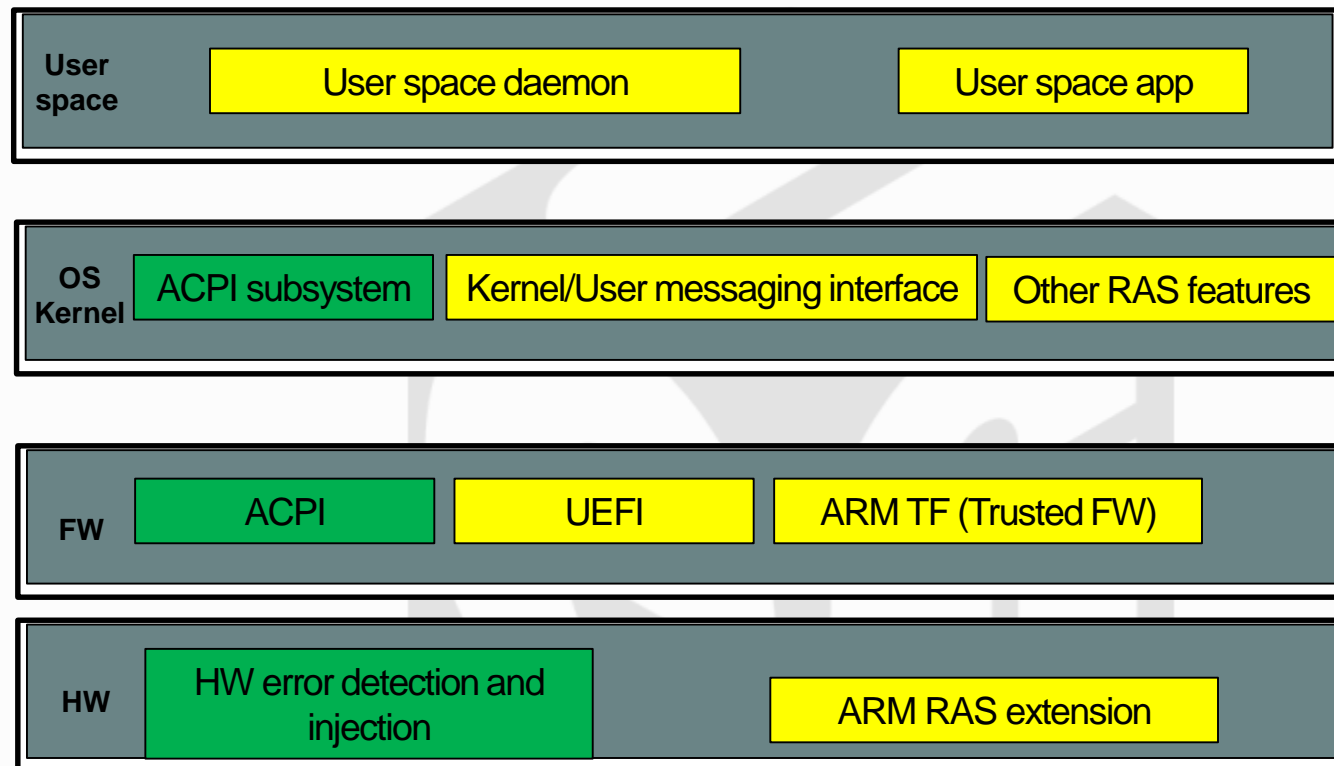
- RAS requirements:
 - Reliability – Minimize impact of error
 - Availability – Uninterrupted system operation and consistent workload performance
 - Serviceability – Scheduled maintenance is okay, but not unscheduled downtime.
- Limitations of EDAC (non- standard) approach:
 - Error visibility.
 - OS support.
 - CPU cycle.
- EDAC approach is perfect for embedded application, but not for volume servers which should take firmware first approach.
- Another possible approach is to have BMC handling all RAS events.

Standardized FW for RAS

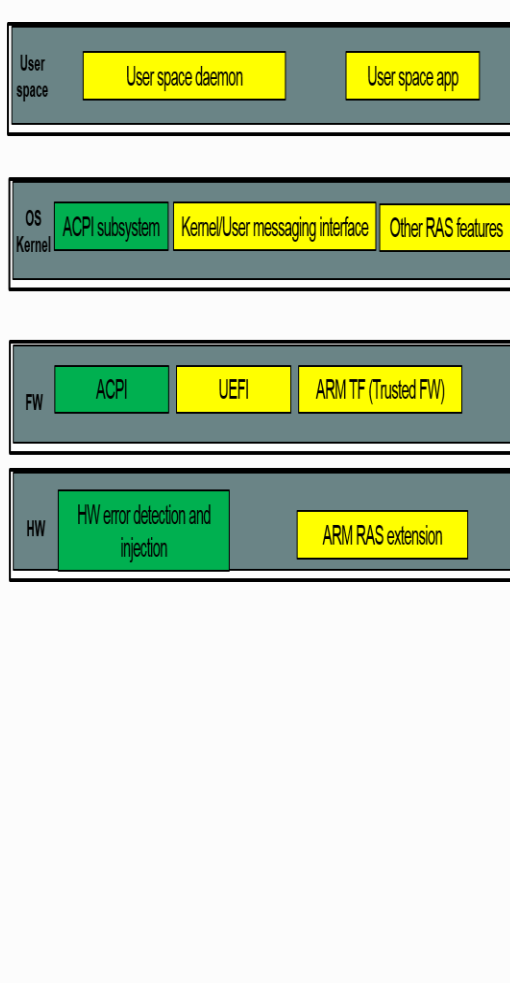


mature

Need some work

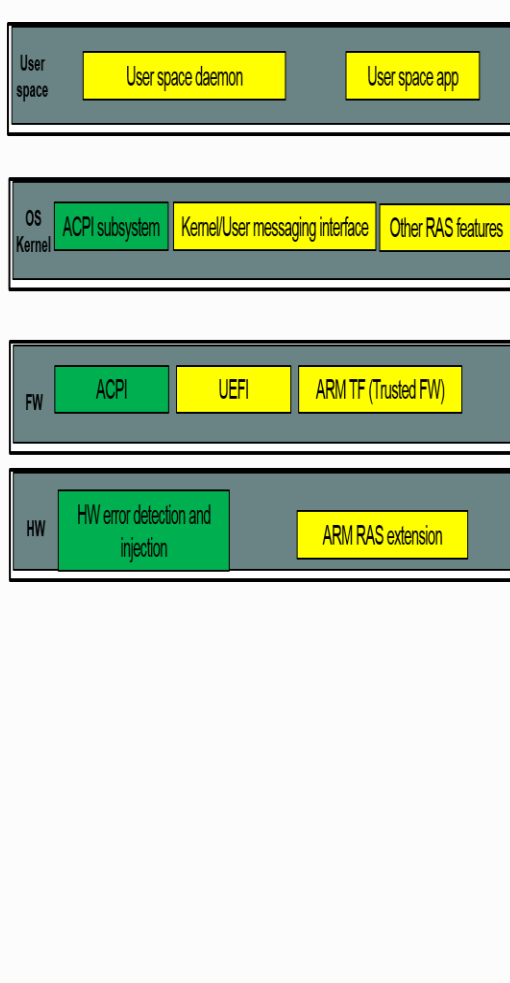


Standardized FW for RAS



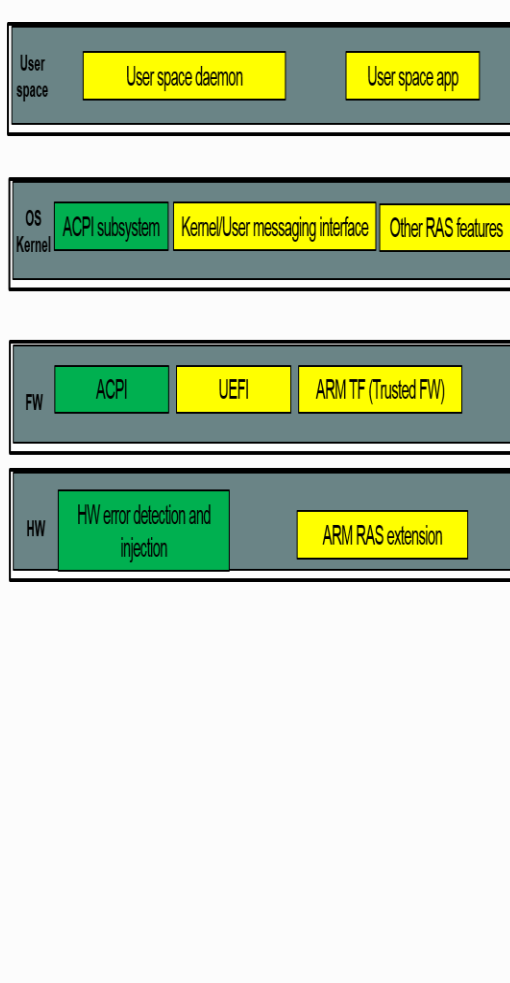
- ARM RAS extension:
 - instruction for “Error Synchronization Barrier”
 - RAS related registers in core.
- To-do:
 - Review ACPI/UEFI specs to see what needs to be added to support RAS extension.
 - Develop standard ARM TF and UEFI code to support RAS extension.

Standardized FW for RAS



- ACPI/UEFI spec updates extended support for ARM based SoC
 - ACPI 6.1 (APEI) published in Jan. 2016.
 - UEFI 2.5 (CPER) published in Jan. 2016.
- To-do:
 - OS support for the new specs.
 - Develop standard ARM TF and UEFI code to support the new spec.

Standardized FW for RAS



- Two approaches to achieve firmware first HW error handling:
 - A Control Processor: a dedicated core runs platform FW to achieve the goal.
 - MM (Management Mode): AP processors get into MM to execute platform code to achieve the goal. More standardization, less HW complexity.
- **Call for action:**
 - Join ABST and PIWG.
 - Get PIWG spec update published.
 - Develop standardized ARM TF and UEFI code to support ARM trust zone based MM.

Standardized FW for other system features



- UEFI 2.5.
- Boot from SATA, USB, SD/MMC, network (PXE)
- Run time services.
- To-do:
 - Secure boot
 - Secure FW update

Standardized FW for other system features



APTIO™: AMI's UEFI BIOS on Cavium Thunder X

● Standard Features

- - Setup
- - PCI
- - USB
- - IPMI
- SPI
- AHCI
- PXE
- NVME

```
Aptio Setup Utility - Copyright (C) 2016 American Megatrends, Inc.
Main Advanced Chipset Security Boot Save & Exit Event Logs
*****
> Discard Changes and Exit      *A
> Save Changes and Reset      *
> Discard Changes and Reset    *
> Save Options                 *
> Save Changes                 *
> Discard Changes              *
> Restore Defaults             *
> Save as User Defaults        *
> Restore User Defaults        *
> Boot Override                *
UEFI OS (P0 - NDC VDS003ABVZ-011FA0)
  Uunatu (P0 - HNS72806PLA380)
UEFI: Iastion Nano Pro PMAP, Partition 1
UEFI: Built-in EFI Shell
  Launch EFI Shell from filesystem device
*****
```

Standardized FW for ACPI features

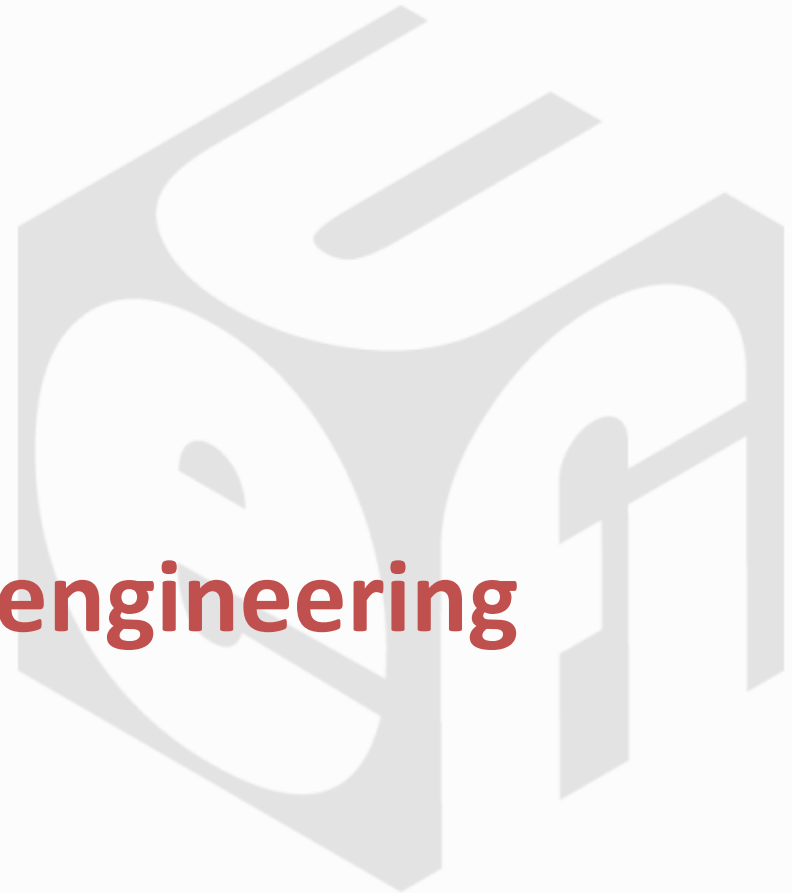


- ACPI Tables:
 - DSDT, SSDT
 - MADT, GTDT, IORT
 - SRAT, SLIT
 - MCFG
 - SPCR, DBG2
- System states – G0 (S0) and G2 (S5)
- Processor states – c0 and c1
 - To do: LPI (Lower Power Idle states were introduced in ACPI 6.0)
- Device states – in development

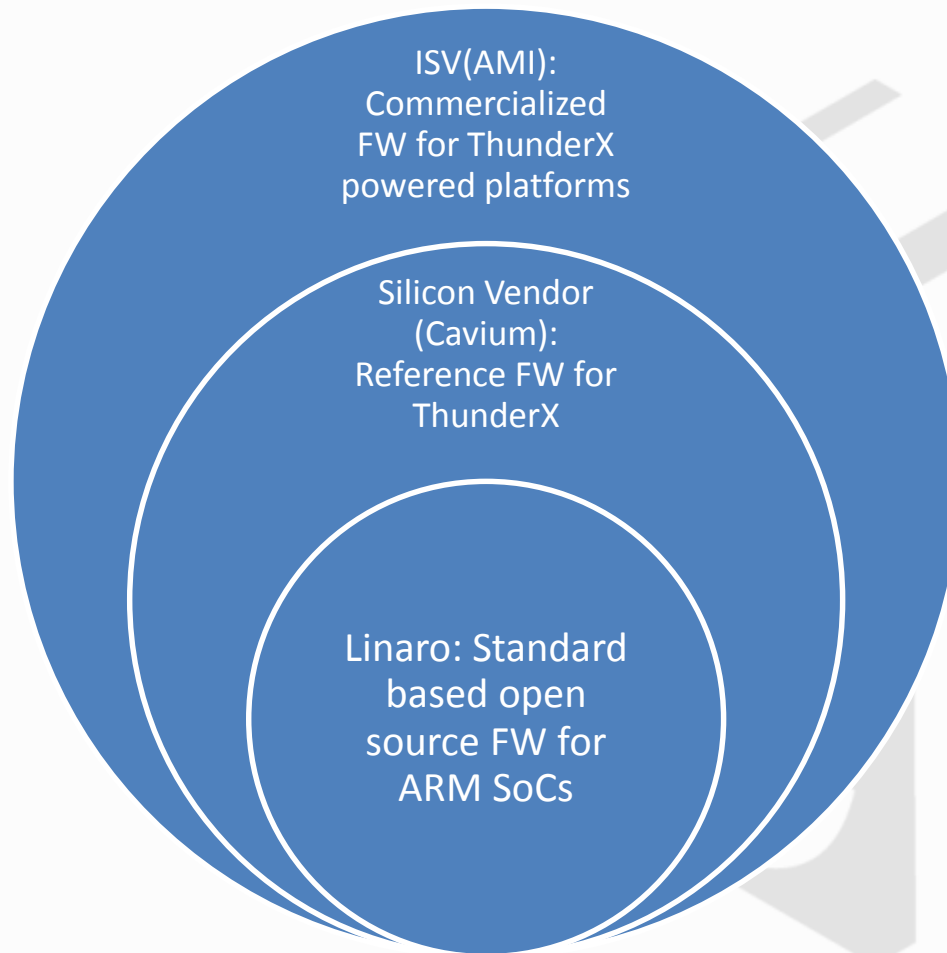


Section Heading

Standardized FW engineering process



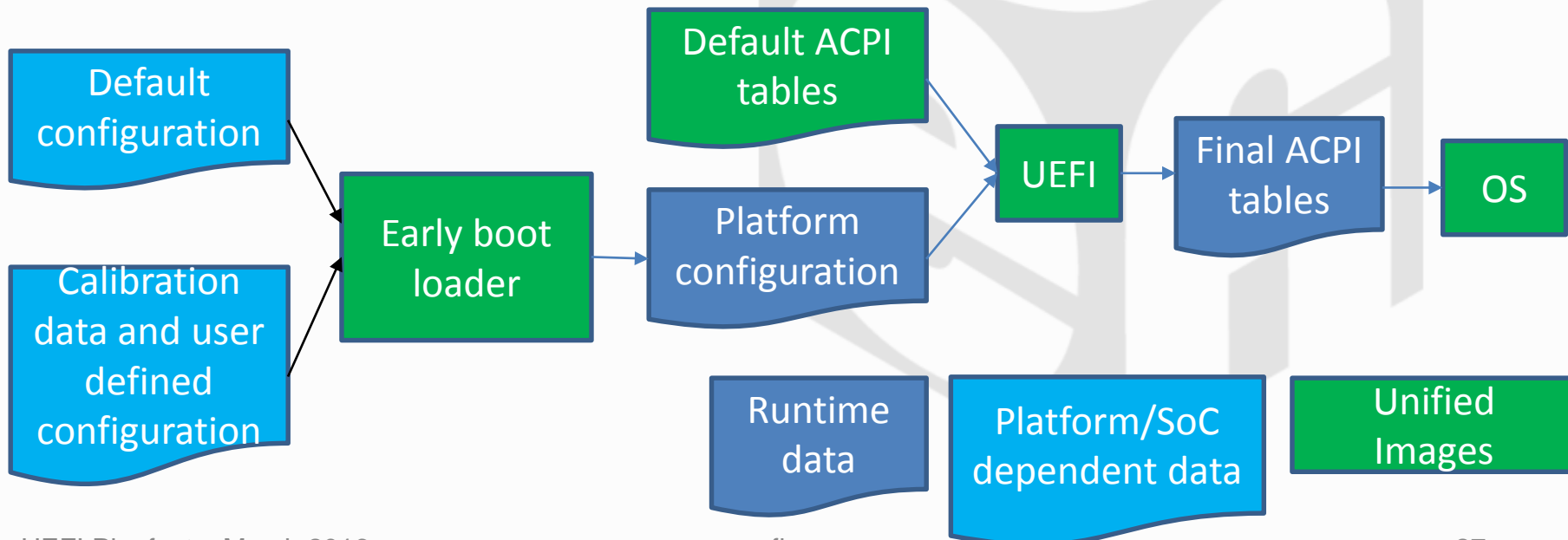
Standardization vs. featurization



Platform/SoC proliferation



- Problem: How to support ODM/Platform/SoC proliferation while managing the code base effectively.
- Solution: Disciplined SW architecture (standard based) and consistent design/coding practice.



APTIO™: AMI's UEFI BIOS on Cavium Thunder X



```
ApTio Setup Utility - Copyright (C) 2016 American Megatrends, Inc.
Main Advanced Chipset Security Boot Save & Exit Event Logs
*****
> Discard Changes and Exit          F4
> Save Changes and Reset           F2
> Discard Changes and Reset       F4
> Save Options                     F9
> Save Changes                     F10
> Discard Changes                   F4
> Restore Defaults                  F8
> Save as User Defaults             F5
> Restore User Defaults             F6
> Boot Override                    F12
UEFI OS (P0 - NDC VDS003ABVZ-011FA0)
ubuntu (P0 - HDS72800PLA380)
UEFI: Iastion Nano Pro PMAP, Partition 1
UEFI: Built-in EFI Shell
> Launch EFI Shell from filesystem device
*****
```

● Visual eBIOS (VeB)

- BIOS Development Utility

● BIOS features based on AMI eModules

- Provide drop-in UEFI Feature
- Source Level control of Modules
- Easy expansion of UEFI projects
- Use VeB Wizards to create eModules
- Delete unused eModules.

● SVN: On Demand Source Code Access

- SVN allows a customer to receive source updates
- Access is integrated into VeB for easy updating of modules



Section Heading

Option ROM Challenges with ARMv8

Current Industry Status



- Virtually all PCIe add-in cards currently have some type of x86 native Option ROM. 2 typical formats:
 - Legacy BIOS Option ROM
 - UEFI Option ROM
- The x86 native implementations today cannot be supported as they exist on ARMv8 servers
- UEFI provides mechanisms to accommodate multiple Option ROM images
- End users are deploying ARMv8 based servers today and requiring clear path to achieving equivalent card and feature support for ARMv8 servers

Call for action



- There are 3 potential paths to enable non-x86 architectures:
 - EBC Option ROM format
 - Native port to target architecture
 - X86 emulation to run existing native Option ROM format
- Requires coordinated industry engagement and commitment:
 - UEFI Forum: consistent and defined standards
 - ARM: coordination across ARM partner community
 - IHVs: reference implementations aligned with defined Option ROM format
 - Firmware vendors: full UEFI implementation and validation
 - Silicon/Server Vendors: validation of system level solution

Thanks for attending the
UEFI Spring Plugfest 2016



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

presented by

