presented by

UEFI ARM Update

UEFI Summerfest – July 15-19, 2013
Presented by Dan Handley (ARM)

Agenda

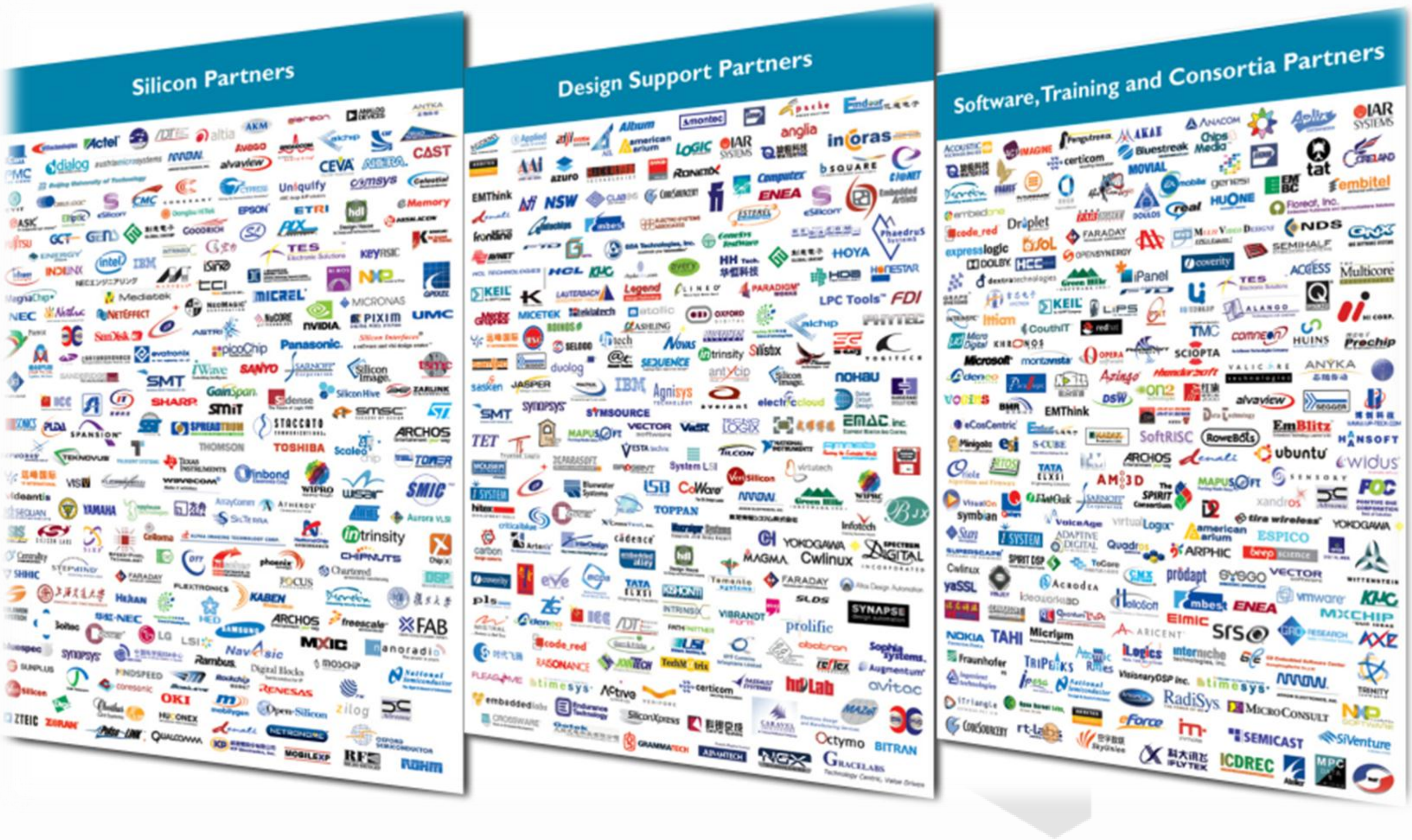
- ARM Economics
- ARM UEFI Strategy
- Current Status
- Future Work
- Questions

Economics

- What are the ARM numbers?

– Processors shipped in 2012	: **~8.7 B** (~7.9 B in '11)
– Processors shipped in total	: **>30 B**
– Processor licenses	: **~960** (850 in '12)
– Semiconductor partners	: **310** (290 in '12)
– Process technology	: **14 – 250 nm**
– Connected community members	: **1000+** (950 in '12)

Economics (1000+)



ARM UEFI Strategy

Why UEFI on ARM?

- Driving forces for UEFI on ARM
  - Processor and system complexity increasing
  - Support existing partners' ARM processor-based UEFI solutions
  - Help standardize boot process for ARM processor-based platforms
  - Improve hw-sw interface for OS that target the ARM architecture

- Advantages to ARM partners and OEMs
  - Write once per platform, saving costs in bootloader development
  - UEFI specification written down and peer reviewed
  - Tested UEFI drivers available from 3rd party peripherals providers
  - Provides an environment for manufacturing tests

ARM UEFI Vision

- Provide standard ARM architectural support
  - Correctness in implementation within ARMv7-A and ARMv8-A architectures
  - Future-proof through standardized (rather than proprietary) reference software
  - Focus on reducing fragmentation and overall partner support costs
- Provide reference ports of UEFI for ARM development platforms
- Support BIOS (and other) partners' UEFI development
  - Directly and through Linaro

ARM Engineering Strategy

- UEFI support for the ARM Architecture
  - Maintain ARM packages and docs in Tianocore EDK2 repository
  - Implement support for new ARM architectures, CPUs and system IP
  - Implement common UEFI features or applications for ARM
  - Maintain SCT for ARM and validate on standard platforms
  - Align with relevant ARM Platform Design Documents (PDDs)
- UEFI support for ARM platforms
  - Porting for new ARM development platforms
  - Maintained within EDK2 (for standard platforms) or other neutral repository
- Help partners with UEFI platform code management and development

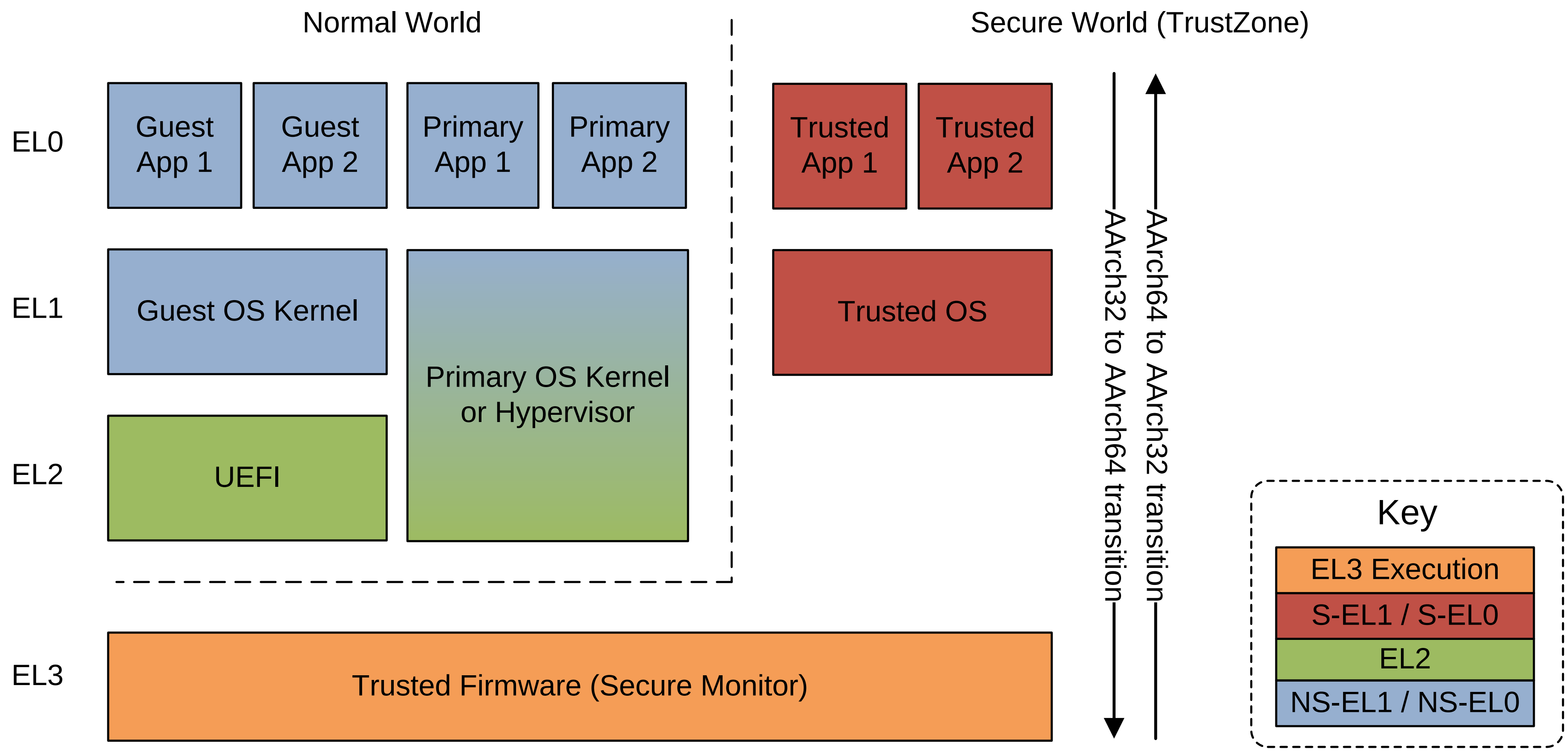
New Technologies

- **big.LITTLE**
  - Heterogeneous computing technology providing both high performance and extreme power efficiency, serving dynamic computing demands (32-bit & 64-bit)

- **Virtualization**
  - Includes Large Physical Address Extensions (LPAE), second level of MMU page table translations and support for hypervisors (32-bit & 64-bit)

- **ARMv8-A / AArch64**
  - Brings 64-bit support to the ARM Architecture increasing the register file, media instructions, addressing range and cryptography instructions (64-bit)

Example ARMv8-A Stack



Current Status

Specification

- ARM Binding Sub-Team (ABST) activities:
  - Created AArch64 UEFI Bindings
    - Support now available in UEFI 2.4
  - Virtualization Protocol Proposal
    - Standardize the way to start a hypervisor from AArch32 UEFI
  - Boot Architecture
    - Discussions around the standardization of the ARM Boot Architecture

Existing EDK2 Features

- ARMv7-A architectural support
  - Maintained by ARM since February 2011
  - With help from Apple, HP, Linaro, ...
- Standard implementations for ARM hardware IP
  - All Cortex-A class processors, caches, interconnects, memory controllers, ...
- ARM development platform support
  - Models, Versatile Express based systems (A9x4, A15x2 + A7x3)
- TrustZone initialization, big.LITTLE
- Booting ATAG and FDT Linux kernels
- Toolchain support (ARM, GNU, XCode)
- Debug (GDB, DS-5 integration)
  - http://blogs.arm.com/software-enablement/884-uefi-debug-made-easy/
- SCT port to ARM (integrated with main SCT package)
- Using any CPU as the primary

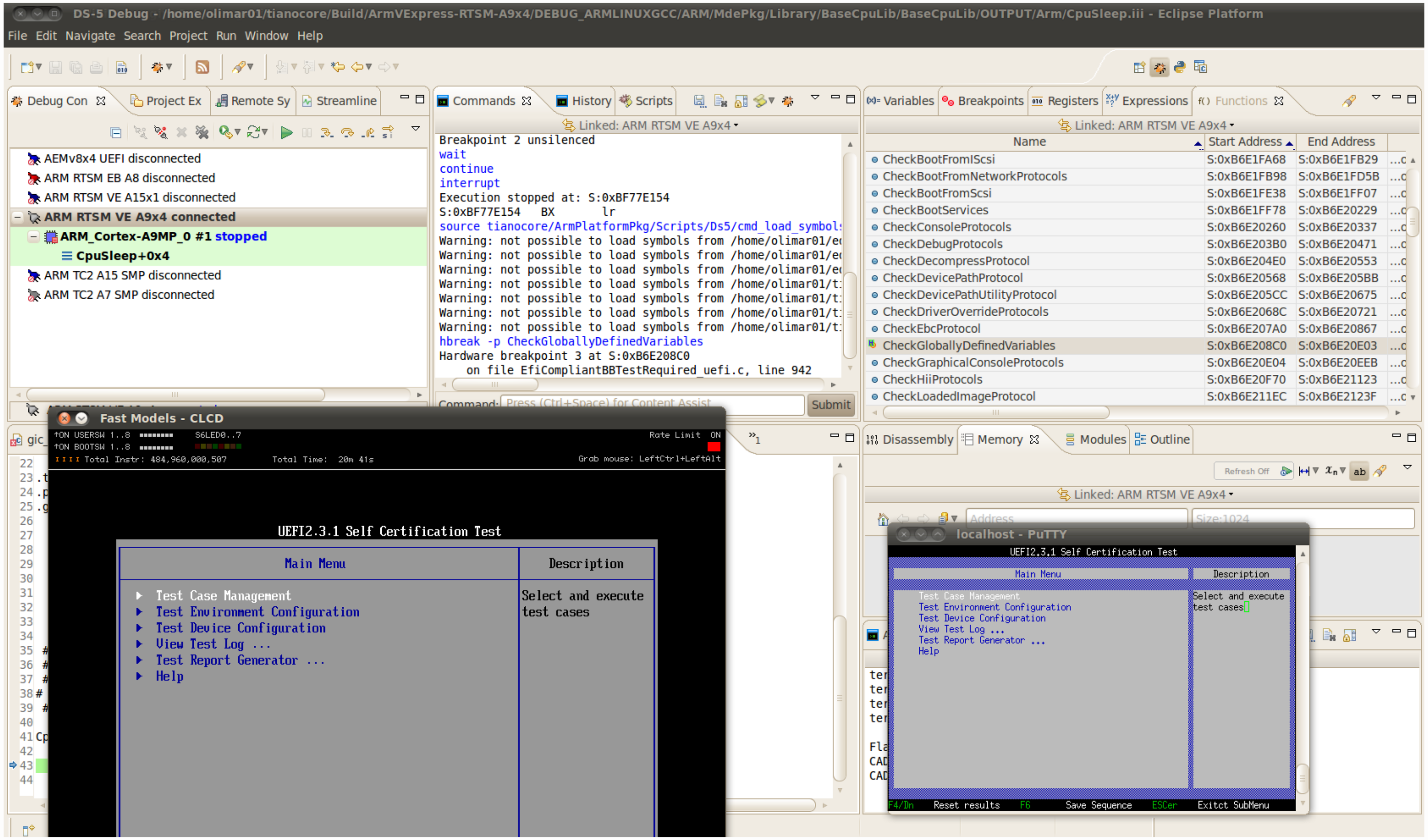
Current ARM EDK2 Focus

- Adding support for AArch64 in EDK2 and SCT
- Implementation of the ARM Virtualization Protocol proposal
- Aligning EDK2 with latest UEFI Specification
- Improving protocol support/compliance
- Enabling the ARM Ecosystem through Open Source contributions

AArch64 EDK2 support

- ARM recommends UEFI for all AArch64 systems
- Available to licensees for last few months
- Publication of UEFI 2.4 spec unblocks public release
  - Upstreaming to EDK2 imminent
- Focus is on ARM Fast Models for now
  - Fixed Virtual Platforms (FVPs)
  - AArch64 hardware not widely available yet
  - Platform support will be available from neutral repository

Fast Model Example


DS-5 Debug - /home/olimar01/tianocore/Build/ArmVExpress-RTSM-A9x4/DEBUG_ARMLINUXGCC/ARM/MdePkg/Library/BaseCpuLib/BaseCpuLib/OUTPUT/Arm/CpuSleep.iii - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Debug Con
Project Ex
Remote Sy
Streamline
Commands
History
Scripts
Variables
Breakpoints
Registers
Expressions
Functions
Disassembly
Memory
Modules
Outline
Refresh Off
Address
Size:1024
Command:
Press (Ctrl+Space) for Content Assist
Submit
gic_
Fast Models - CLCD
localhost - PuTTY

- AEMv8x4 UEFI disconnected
- ARM RTSM EB A8 disconnected
- ARM RTSM VE A15x1 disconnected
- **ARM RTSM VE A9x4 connected**
  - **ARM_Cortex-A9MP_0 #1 stopped**
    - **CpuSleep+0x4**
- ARM TC2 A15 SMP disconnected
- ARM TC2 A7 SMP disconnected

Linked: ARM RTSM VE A9x4

```
Breakpoint 2 unsilenced
wait
continue
interrupt
Execution stopped at: S:0xBF77E154
S:0xBF77E154   BX       lr
source tianocore/ArmPlatformPkg/Scripts/Ds5/cmd_load_symbol:
Warning: not possible to load symbols from /home/olimar01/ec
Warning: not possible to load symbols from /home/olimar01/ec
Warning: not possible to load symbols from /home/olimar01/ec
Warning: not possible to load symbols from /home/olimar01/t:
Warning: not possible to load symbols from /home/olimar01/t:
Warning: not possible to load symbols from /home/olimar01/t:
Warning: not possible to load symbols from /home/olimar01/t:
hbreak -p CheckGloballyDefinedVariables
Hardware breakpoint 3 at S:0xB6E208C0
    on file EfiCompliantBBTestRequired_uefi.c, line 942
```

Linked: ARM RTSM VE A9x4

Name	Start Address	End Address	
CheckBootFromIScsi	S:0xB6E1FA68	S:0xB6E1FB29	...c
CheckBootFromNetworkProtocols	S:0xB6E1FB98	S:0xB6E1FD5B	...c
CheckBootFromScsi	S:0xB6E1FE38	S:0xB6E1FF07	...c
CheckBootServices	S:0xB6E1FF78	S:0xB6E20229	...c
CheckConsoleProtocols	S:0xB6E20260	S:0xB6E20337	...c
CheckDebugProtocols	S:0xB6E203B0	S:0xB6E20471	...c
CheckDecompressProtocol	S:0xB6E204E0	S:0xB6E20553	...c
CheckDevicePathProtocol	S:0xB6E20568	S:0xB6E205BB	...c
CheckDevicePathUtilityProtocol	S:0xB6E205CC	S:0xB6E20675	...c
CheckDriverOverrideProtocols	S:0xB6E2068C	S:0xB6E20721	...c
CheckEbcProtocol	S:0xB6E207A0	S:0xB6E20867	...c
CheckGloballyDefinedVariables	S:0xB6E208C0	S:0xB6E20E03	...c
CheckGraphicalConsoleProtocols	S:0xB6E20E04	S:0xB6E20EEB	...c
CheckHiiProtocols	S:0xB6E20F70	S:0xB6E21123	...c
CheckLoadedImageProtocol	S:0xB6E211EC	S:0xB6E2123F	...c

Linked: ARM RTSM VE A9x4

```
↑ON USERSW 1..8  ▪▪▪▪▪▪▪▪▪   S6LED0..7                                   Rate Limit  ON
↑ON BOOTSW 1..8  ▪▪▪▪▪▪▪▪▪
I I I I Total Instr: 484,960,000,507     Total Time:  20m 41s     Grab mouse: LeftCtrl+LeftAlt

                       UEFI2.3.1 Self Certification Test

              Main Menu                          Description

  ▶ Test Case Management                    Select and execute
  ▶ Test Environment Configuration          test cases
  ▶ Test Device Configuration
  ▶ View Test Log ...
  ▶ Test Report Generator ...
  ▶ Help
```

```
                 UEFI2.3.1 Self Certification Test

               Main Menu                        Description

      Test Case Management                 Select and execute
      Test Environment Configuration       test cases
      Test Device Configuration
      View Test Log ...
      Test Report Generator ...
      Help

F4/Dn    Reset results    F6       Save Sequence    ESCer    Exitct SubMenu
```

Fixed Virtual Platforms

- **Current: Two main flavours of FVP**
  - “AEMv8-RTSM-VE”: primary development platform
  - Foundation: Free of charge entry offering (http://www.arm.com/fvp)
  - The key development platforms for key software activity
    - AArch64 tools (GNU and ARM), UEFI and Linux kernel
    - Linux filesystem and related packages

- **2013 H2: Address broader needs for software eco-system**
  - System Architecture (Platform Design Documents)
  - Dual cluster capability, power management emulation
  - Low-level software frameworks:
    - Support for all exception levels (secure world, virtualization support)
    - Power State Coordination Interface (PSCI)
  - "VE" => "Base" platform (generic AEMv8 and Cortex-A53/57 variants)

ARM Virtualization Protocol

- Problem: Need to make Virtualization Extensions available to OS
- For AArch64, can just run UEFI and OS in "Hyp Mode" (EL2)
- For AArch32, existing systems run UEFI in "SVC Mode" (EL1)
- Protocol allows new OS loaders to escalate UEFI into "Hyp Mode"
  - While providing compatibility with existing OS
- Can already start Linux KVM from AArch32 UEFI
  - Solution not yet complete

Future Work

- Create and manage regular stable branches
- Support latest ARM System IP
  - GICv3, interconnect, memory controllers, ...
- Improved virtualization support
  - Virtio drivers (block device, network)
  - VM booting via UEFI
- Maintenance, Consolidation, Housekeeping, Integration, Upstreaming, ...

Summary

- UEFI is a compelling solution for ARM and its partners
  - Recommended bootloader for AArch64
- ARM is investing in both specification and implementation
  - Keeping up to date with new technologies (big.LITTLE, Virtualization, AArch64, ...)
- AArch64 implementation available publically imminently
- ARM models used to drive standardization
- Supporting BIOS (and other) partners, directly and through Linaro

Questions?

Thanks for attending the UEFI Summerfest 2013

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

presented by

ARM