UEFI and Linux
Platform Interface Advancement Through Collaborations
Kirk Bresniker
HP Fellow, Vice President and Chief Technologist (HP Servers)
Sept 16, 2013
Agenda

UEFI Overview
- Org Chart
- Timeline

UEFI Updates
- Specifications Update
- Test Update & Plugfests
- ACPI

Open Source Developments
- Open Source UEFI Reference
- Linux on x64
- Linaro

Challenges
- Balance between standards and differentiations
- Preboot User Interface Designs
- UEFI Secure boot
- Dual Boot
- UEFI for 32-bit x86

Call to Action
The UEFI Forum Organization Chart

Board of Directors (11 Promoters)

Industry & Communications WG

UEFI Specification WG

Platform Initialization WG

Test WG

Security Subteam

Configuration Subteam

Network Subteam

Shell Subteam

ARM Binding Subteam

Officers:
President: Mark Doran (Intel); VP (CEO): Dong Wei (HP)
Secretary: Jeff Bobzin (Insyde); Treasurer: Bill Keown (Lenovo)

Platform Initialization (PI): interfaces produced & consumed by firmware only; promote interoperability between firmware components

UEFI: pre-OS (and limited runtime program interfaces) between UEFI applications(incl. OSes)/drivers and system firmware

ACPI: tables and primary runtime interpreted control methods provided by system firmware to the OS for system configuration, power management and error handling
UEFI Industry Transition

1995
HP/Intel needed a boot architecture for Itanium servers that overcame BIOS PC-AT limitations
Intel created EFI with HP and others in the industry, made it processor agnostic (x86, ia64)

1997-2000

2004
Tianocore.org, open source EFI community launched

2005
Unified EFI (UEFI)
The UEFI Forum, with 11 promoters, was formed to standardize EFI, extended to x64

2009
UEFI extended to ARM AArch32

2012
Windows 8 and ubiquitous native UEFI adoption for client PCs (Boot Performance, Secure Boot focused)

2013
250 members and growing! Linux Distros extended support for UEFI Secure Boot.
First Linux Foundation hosted UEFI Plugfest. UEFI extended to ARM AArch64.
Industry needs a venue for ACPI.next

Future
UEFI as the converged firmware infrastructure
UEFI Updates
Specifications Update

**UEFI v2.4 (July 2013)**
- ARM AArch64 Binding
- Adapter Information Protocol
- Capsule to deliver update image to Firmware Management Protocol
- Disk IO 2 Protocol to support Async IO
- Support NVM Express devices

**PI v 1.3 (July 2013)**
- I²C Bus Protocol
- NVM Express Disk Info GUID
- PCI Enumeration Complete GUID
Test Update and Plugfests

**SCT v2.3.1C (July 2013)**
Target compliance with UEFI Specification v2.3.1C
Collective effort with limited resources
Intel and ARM contributed most of the resources
Some interfaces are not covered

**SCT v2.4**
Target compliance with UEFI Specification v2.4
Target date: July 2014
Member contributions welcome!

**Plugfests in 2013**
Taipei Plugfest (hosted by AMI) – March
UEFI Summerfest (hosted by Microsoft) – July
New Orleans Plugfest – later this week!

Hosted by The Linux Foundation and The UEFI Forum
Synchronize Interface Definitions with Diverse Ecosystem Participations

The UEFI Forum is by far the de facto industry standards body for platform interfaces

Industry needs a venue for ACPI.next
Open Source UEFI Reference

TianoCore EDK 2
Reference implementation for UEFI/PI Specs
BSD License
Hosted at SourceForge: tianocore.org
Covers core UEFI/PI framework
Supports virtual environment:
  • NT32, OVMF (Open Virtual Machine Firmware)
  • AArch64 Foundation, AArch64 Base FVP (Fixed Virtual Platform)
Supports reference boards:
  • Omap 35xx, BeagleBoard, ARM Real View, ARM Versatile Express
  • MinnowBoard (www.minnowboard.org)
Linux on x64

Distros are supporting UEFI Boot
Distros are supporting UEFI Secure Boot via UEFI CA
- Fedora 18+
- Ubuntu 12.10+
- OpenSuSe 12.3+
- Linux Foundation enablement

Self-signing for the kernel development
Linaro (www.linaro.org)  
- A Linux-on-ARM Not-for-Profit

**UEFI as the preferred boot model for ARMv8 servers**

UEFI enablement is championed by the Linaro Enterprise Group (LEG)

- https://wiki.linaro.org/ARM/UEFI
- https://wiki.linaro.org/LEG/Engineering/Kernel/UEFI

- Experiment on more physical platforms:
  - ARM Versatile Express (A5, A9, A15)
  - Samsung Origen, Arndale
  - TI PandaBoard

- Support KVM
- Enable UEFI SCT to run on ARM systems

**ACPI as the preferred runtime interface for ARMv8 servers**

ACPI enablement is also championed by the LEG

- https://wiki.linaro.org/LEG/Engineering/Kernel/ACPI
- https://wiki.linaro.org/LEG/Engineering/Kernel/ACPI/RASandACPI

- Main focus on system configuration, power management and RAS support
Challenges
Balance between Standardization and Differentiation

• UEFI intends to support a wide range of platforms from the embedded devices to the enterprise servers
• UEFI standardizes on the platform interfaces for interoperability rather than the implementations
• UEFI standardizes on the human interface infrastructure (HII) rather than the actual user interfaces
  – UEFI Shell is an exception, but the Shell Specification still defines various support levels
• UEFI is extensible for product differentiations
  – Vendor-specific features can be supported via vendor-specific extensions
• All good, but...
  – Implementation and user interface differences, if not properly managed, can present challenges to users

Almost all challenges related to the Linux support are due to these differences not properly managed
Preboot User Interface Designs

Boot Options
UEFI provides a very flexible Multi-Boot framework
EDK 2 provides a baseline reference BDS (Boot Device Selection) with OS Multi-Boot support
Vendors are actively differentiating their BDS designs, often at the product line or even at the product level
However, if not done properly, Multi-Boot support may not be present or fully supported

System Features Management
UEFI HII provides the human interface infrastructure
Vendors are creative in their user interface designs based on HII, from text based to fully GUI based
Getting into the Preboot User Interface itself is a challenge due to Fast Boot
No standard presentations of where and how system features are managed

UEFI Multi-Boot is not a commonly tested feature on client PCS
Navigating and understanding the user interface require effort
UEFI Secure Boot

Feature Support
UEFI Form publishes a whitepaper

• Rootkit attack is a real threat
• UEFI Secure Boot
• As of this date, no one has claimed or demonstrated an attack that can circumvent UEFI Secure Boot on a system where it is properly implemented and enabled

A very useful optional UEFI feature to have
• Physically-present users can turn it off on general-purpose x64 systems

User Interface
Getting into the Preboot User Interface itself is a challenge due to Fast Boot

No standard presentation of where and how UEFI Secure Boot is managed

• Nomenclature not standardized
• Setup Mode/User Mode/"Custom Mode” Mode management varies
• User Key Enrollment may or may not be present
  – Is this better managed via OS App?
• Factory Key Restore may or may not be present

UEFI Secure Boot attempts to prohibit the execution of unprotected code prior to the engagement of the operating system
Dual Boot of Windows 8 and Linux with UEFI Secure Boot Enabled

Intel video on YouTube:
- A very good example based on Intel reference implementation
- Provide a nice overview and basic steps
- [http://www.youtube.com/watch?v=eAnlhkbMang](http://www.youtube.com/watch?v=eAnlhkbMang) (Part 1)
- [http://www.youtube.com/watch?v=dwlbf1VRJ60](http://www.youtube.com/watch?v=dwlbf1VRJ60) (Part 2)
- [http://www.youtube.com/watch?v=_cEwj8bBBC4](http://www.youtube.com/watch?v=_cEwj8bBBC4) (Part 3)

User Interfaces vary from system to system
- Example: HP EliteBook and ProBook Laptops (instructions available from dong.wei@hp.com)
UEFI for 32-bit x86

Why

An interesting combination in the market:
- Non-traditional market segments (e.g., tablets) need the modern UEFI infrastructure
  - No dependencies on traditional BIOS
  - No desire to carry on the CSM burden
- CPU architecture remains 32-bit today for whatever reasons

End up with Class 3 UEFI for 32-bit x86 (e.g., HP ElitePad, Envy x2)

Complications

Fortunately, these are all closed system
- IHVs are spared of supporting 32-bit x86 or EBC UEFI Option ROMs

OSVs would need to support UEFI for 32-bit x86 to be able to run on these systems

Customers are also confused:
- For traditional PCs, OSVs only support x64 UEFI
- But for these closed systems, Android and Windows 8 support UEFI for 32-bit x86
- May present a problem for large scale deployment in enterprises
Call to Action
Call to Action

Participate in the UEFI Development

UEFI Forum Membership
• Specifications (UEFI, PI, ACPI.next?), Tests

Open Source Contribution
• TianoCore
• Linaro

Help Improve User Experience

Work with OEM/IBVs to
• Support user-friendly UEFI Multi-Boot
• Support user-friendly UEFI Secure Boot Management user interface
Q&A

Questions can also be directed to:

Dong Wei
HP Fellow
Vice President and CEO of the UEFI Forum
dong.wei@hp.com

He will be in New Orleans this week for the UEFI Plugfest as well
Thank you