
Plug-fest
MULTIPLY YOUR INNOVATION
AND MAXIMIZE YOUR POTENTIAL
MULTIPLY YOUR KNOWLEDGE

UEFI 2.1 and PI 1.0 Details and Differences

Michael A. Rothman

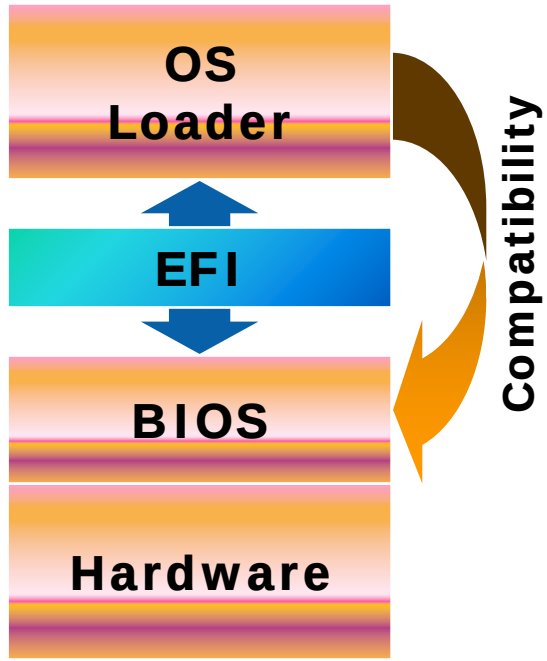
One of the many UEFI guys

Agenda

- A look at EFI and UEFI Overview
- UEFI 2.1 New Content and Changes
- Concept Demo
- PI 1.0 Content and Changes
- Future Development and Test Plans

Brief History On EFI

- Interface specification
  - Implementation agnostic
- Abstracts BIOS from OS
  - Decouples development
- Compatible by design
  - Evolution, not revolution
- Modular and extensible
  - OS-Neutral value add
- Provide efficient Option ROM Replacement
  - Common source for multiple CPU architectures



EFI is the successor to BIOS

Unified EFI Forum, Inc. created for standardization

A Washington non-profit Corporation

- Develops, promotes and manages evolution of Unified EFI Specification
- Continue to drive low barrier for adoption

Promoter members:

- AMD, AMI, Apple, Dell, HP, IBM, Insyde, Intel, Lenovo, Microsoft, Phoenix

Tiered Membership:

- Promoters, Contributors and Adopters

More information: www.uefi.org

Industry momentum for BIOS standardization

UEFI Membership

Promoters: board and corporate officers

Contributors:

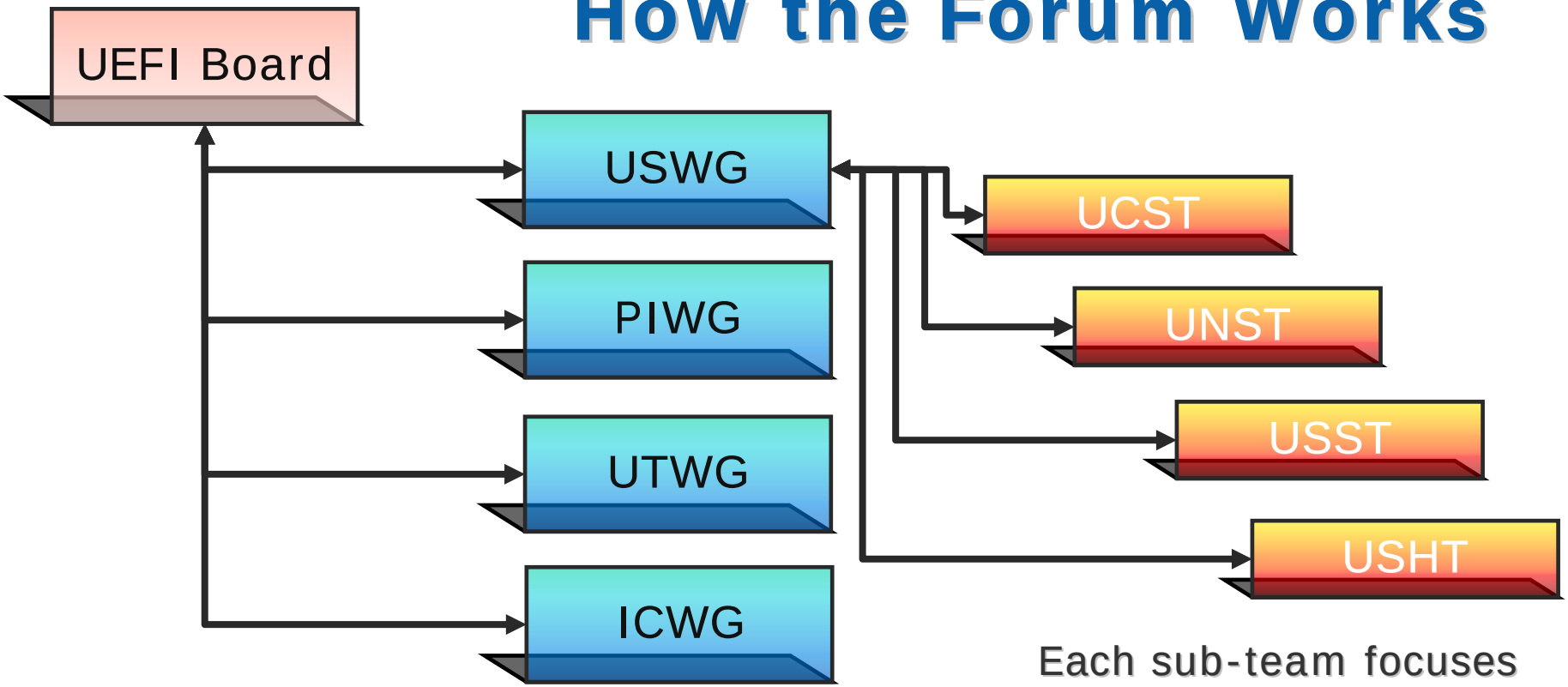
- Corporations, groups or individuals wanting to participate in UEFI
- Chance to join work groups and contribute to spec or test development
- Early access to drafts and work in progress

Adopters:

- Any entity wanting to implement the specification

Membership is open / encourages industry participation

How the Forum Works



Publications/Decisions ratified by the board

Each work group approves/delivers different content to the public.

Each sub-team focuses on specific topics and contributes material to the work group.

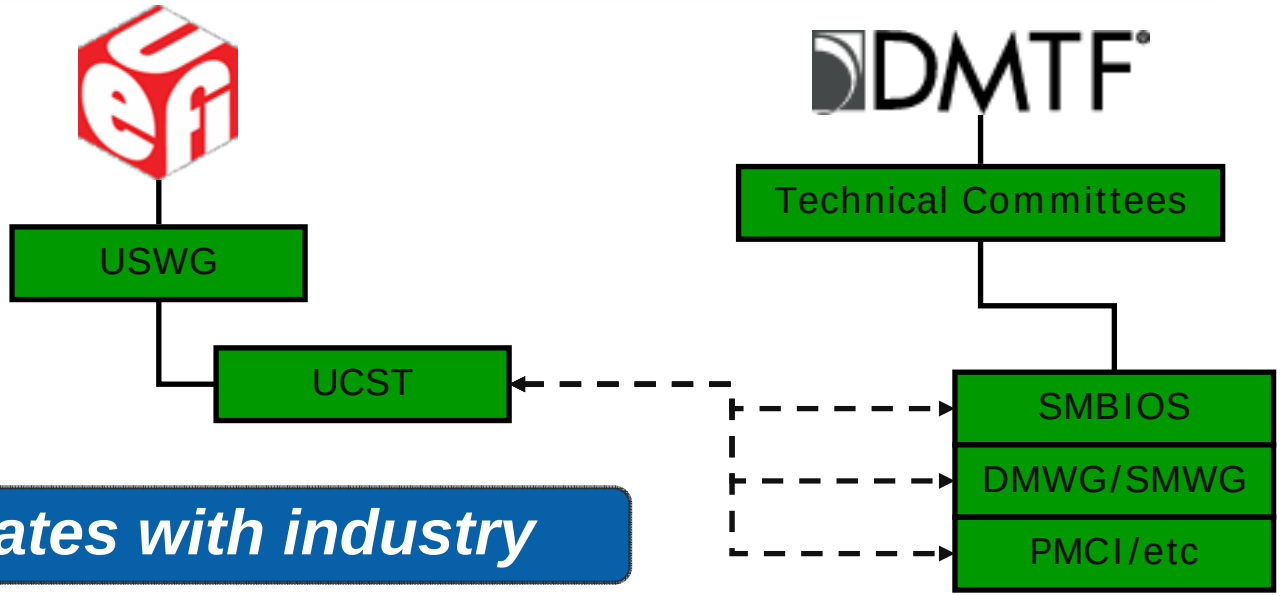
Many groups working together to Standardize Firmware

UEFI interactions with Industry

DMTF and UEFI Forum Work Together to Advance IT Standards

Two Leading Technology Standards Bodies Form Alliance to Address Platform Inventory and Configuration Management Requirements

PORTLAND, Ore. – June 18, 2007 – Two leading technology standards groups have joined forces to help developers speed deployment of standards-based solutions for the end-to-end management of distributed enterprise computing. The Distributed Management Task Force, Inc. (DMTF®) and the UEFI Forum (UEFI™) today announced a plan to align key technical specifications, thereby promoting interoperable management solutions to help lower costs and simplify operations in the data center.



UEFI collaborates with industry

Agenda

- A look at EFI and UEFI Overview
- UEFI 2.1 New Content and Changes
- Concept Demo
- PI 1.0 Content and Changes
- Future Development and Test Plans

UEFI Configuration Infrastructure

- Introduced the Human Interface Infrastructure (HII)

- Problem Statement
  - No standard/interoperable mechanism to address pre-boot based issues like:
    - Localization
      - Standard delivery of string packages
    - Fonts
      - Create standard glyph support along with optional font styles
    - Shared Configuration Infrastructure
      - Alleviate the burden for many configuration engines in a system (e.g. add-in device no longer needs to delay boot or poll for hot-keys, etc)
  - Should be able to also address:
    - Human -> Machine system configuration
      - Think Setup
    - Machine -> Machine system configuration
      - Think Automation

Human Interface Infrastructure

Goals:

- A simplified method for localization.
- Forms Representation that can support complex configuration.
- Allows for configuration in pre-boot, runtime, and remotely.
- Ability for various drivers from different sources (including add-in cards) to interact with configuration infrastructure
- Support User Interface on a wide range of display devices

Introducing a UI/Configuration Infrastructure

Configuration of Add-in Devices

- Device Access APIs

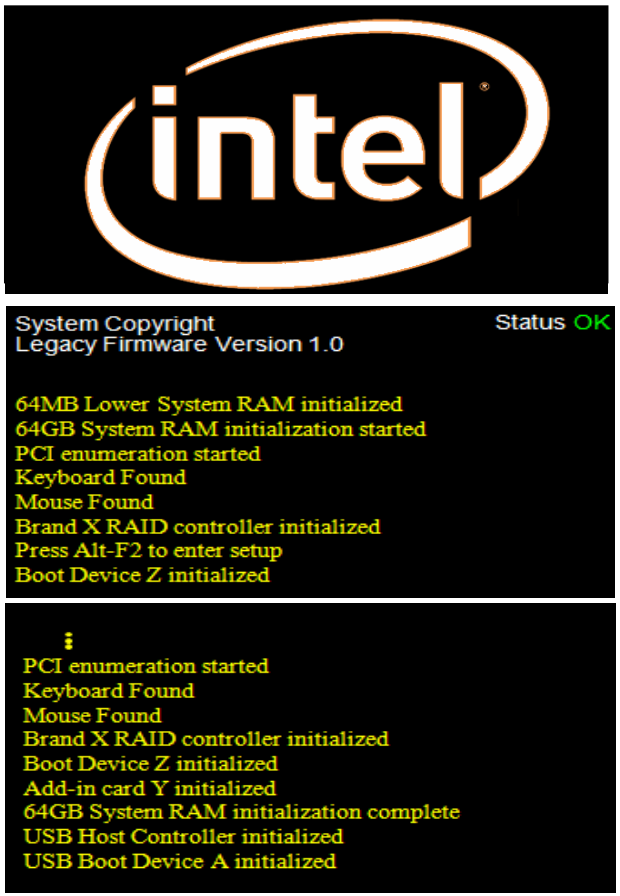
Introduces abstractions to allow the platform BIOS to interact both with the motherboard as well as various other agents (e.g. Add-in device) in the system.

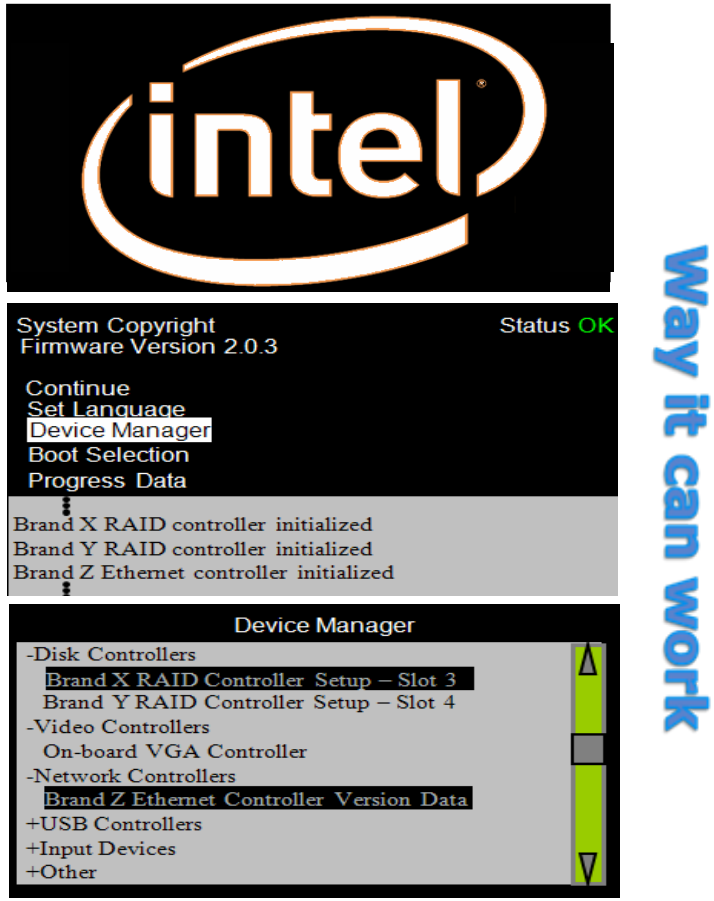
Standard way to programmatically interact with IHV add-in devices.

```
typedef struct {
  EFI_HII_EXTRACT_CONFIG       ExtractConfig;
  EFI_HII_ROUTE_CONFIG         RouteConfig;
  EFI_HII_FORM_CALLBACK        Callback;
} EFI_HII_CONFIG_ACCESS_PROTOCOL;
```

Example usage of this methodology

Way it has worked

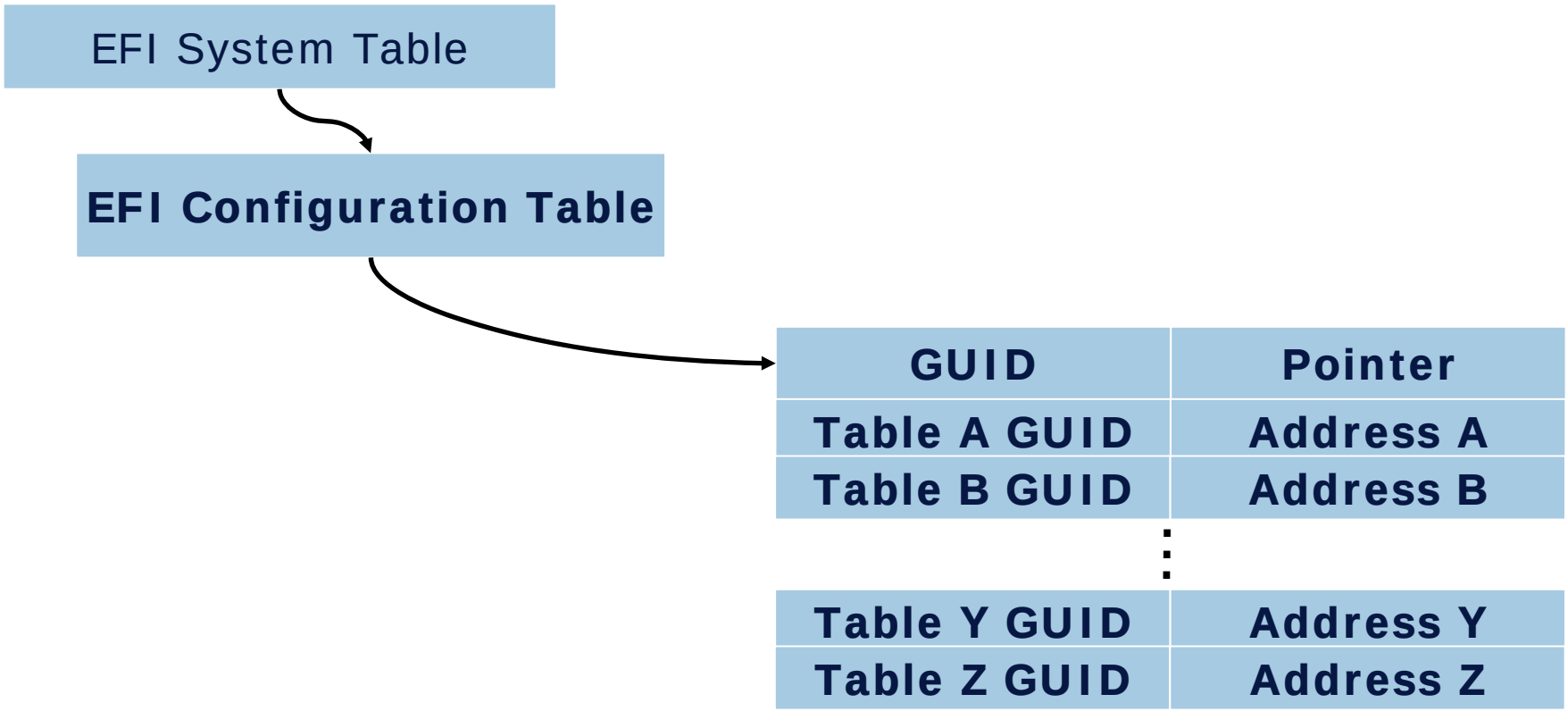




Way it can work

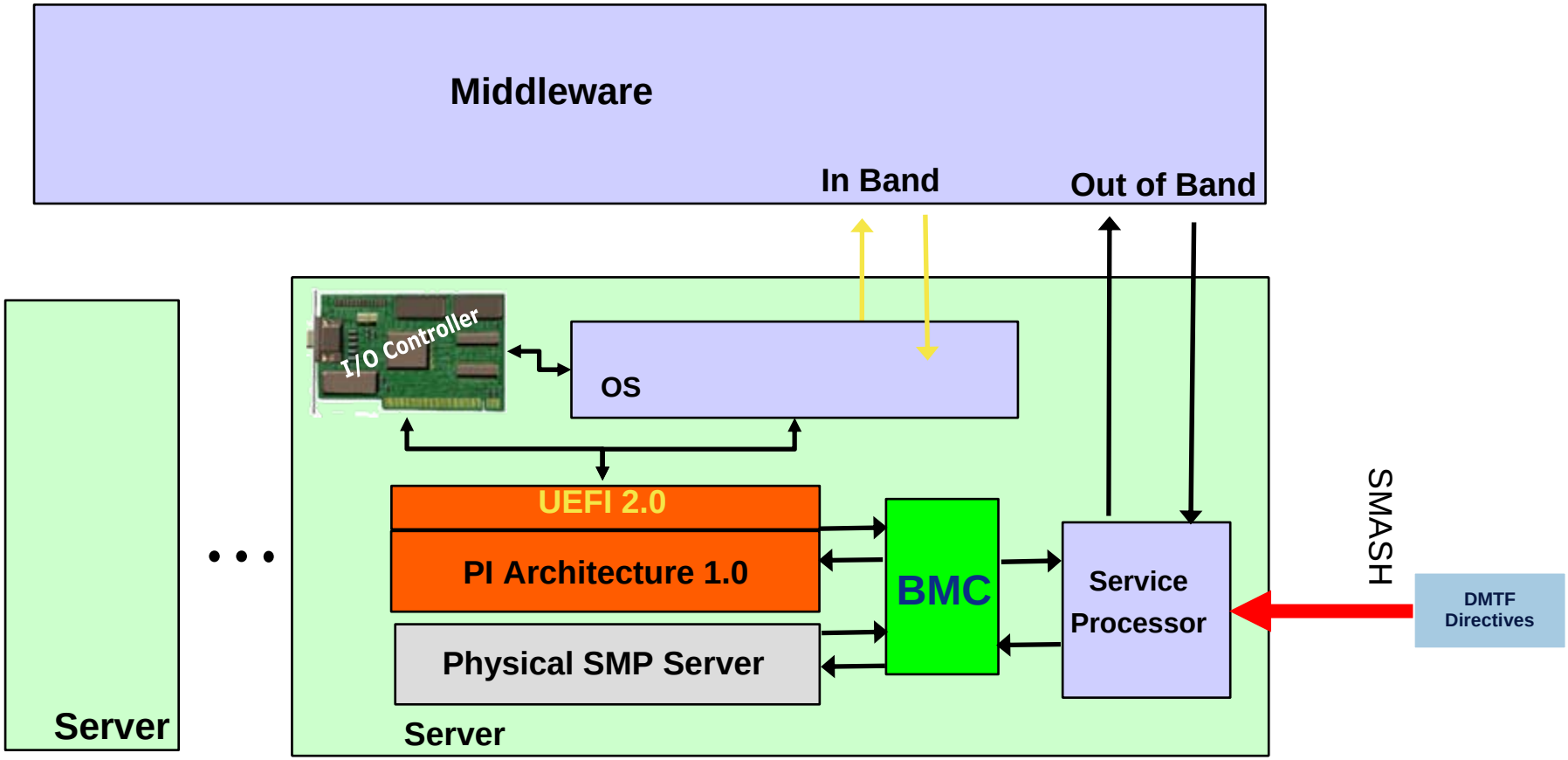
Evolving the infrastructure capabilities

Local Configuration Infrastructure



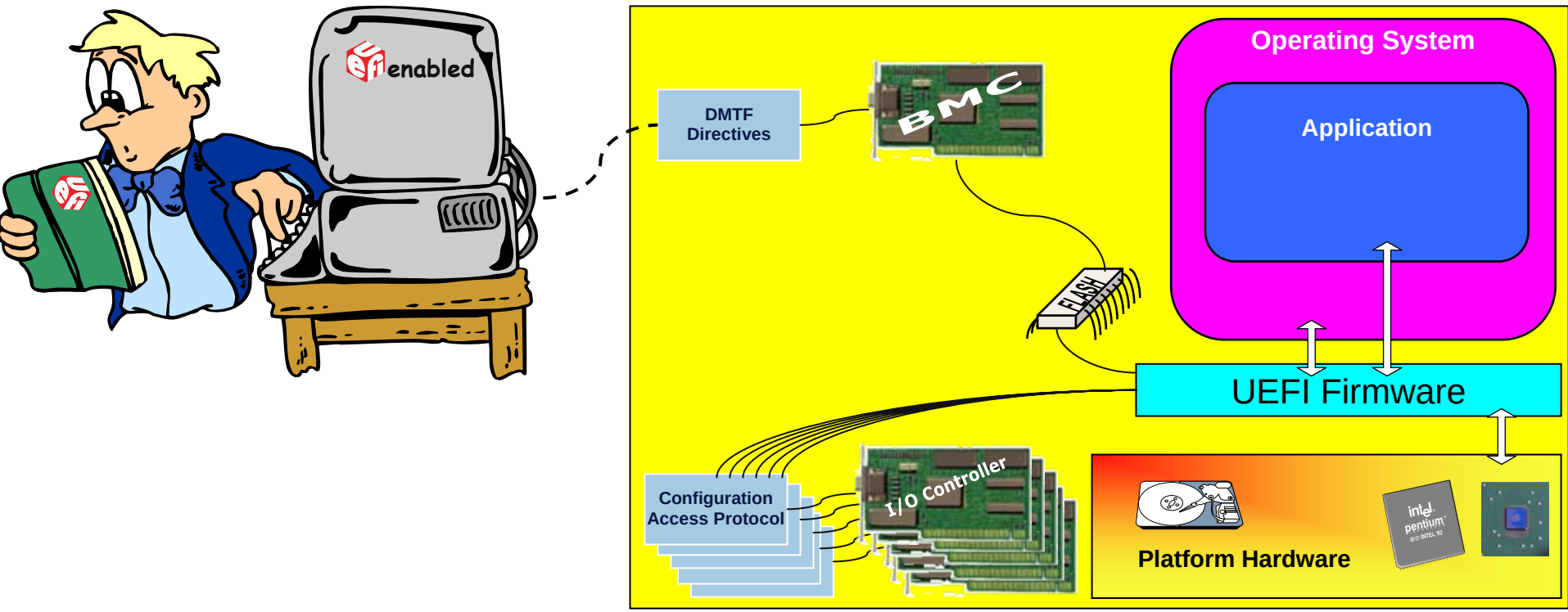
Standard method to pass interesting state data up through to the OS

Basic network-based configuration interactions



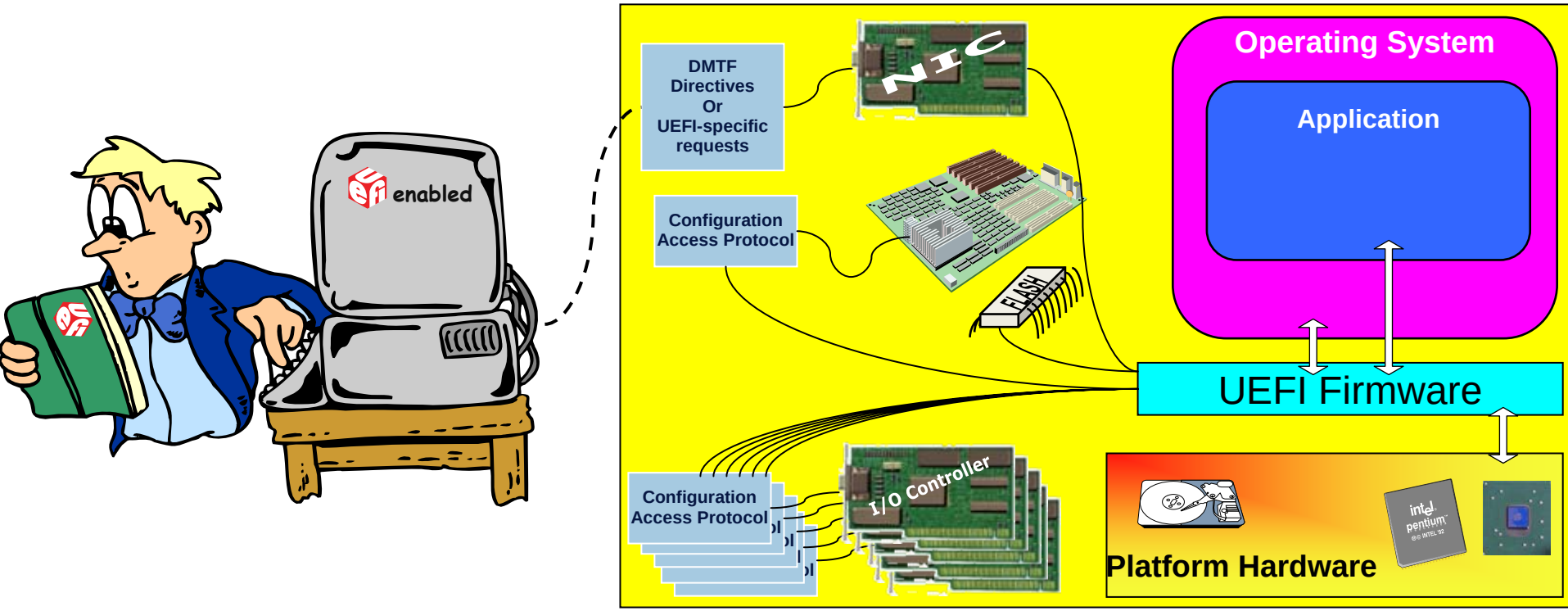
Advanced Usage Models

- Platforms with a service processor (e.g. ME/BMC)



Advanced Usage Models

Platforms without a service processor (e.g. ME/BMC)



Advanced network-based configuration interactions

Call to Action!

We need IHV support to evolve this capability

Localization


Spanish


US English


French

String ID #4	String Representation	H	E	L	L	O		W	O	R	L	D	
	Unicode Encoding	0x0048	0x0045	0x004C	0x004C	0x004F	0x0020	0x0057	0x004F	0x0052	0x004C	0x0044	0x0000

String ID #4	String Representation	H	O	L	A		M	U	N	D	O	
	Unicode Encoding	0x0048	0x004F	0x004C	0x0041	0x0020	0x004D	0x0055	0x004E	0x0044	0x004F	0x0000

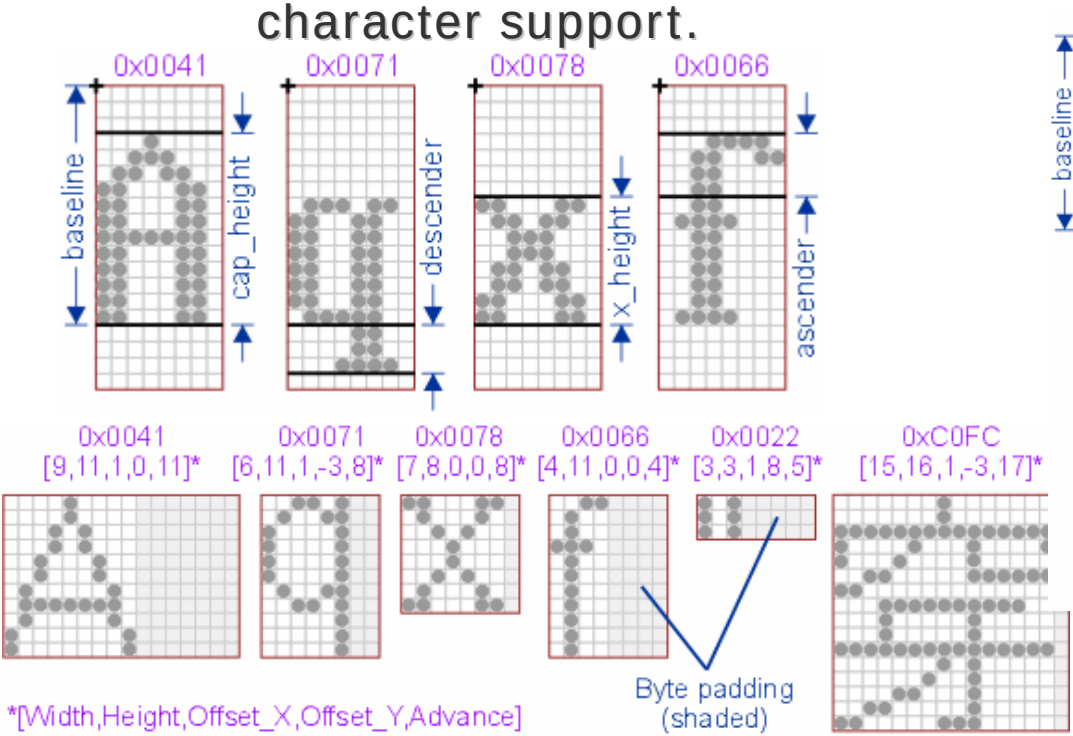
String ID #4	String Representation	你	好	世	界	
	Unicode Encoding	0x4F60	0x597D	0x4E16	0x754C	0x0000

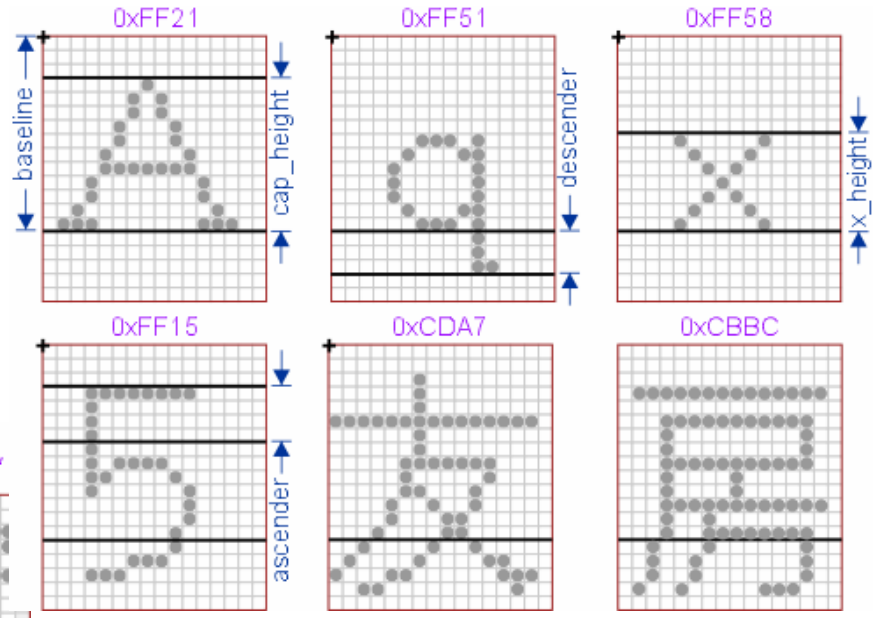
Providing input support for international venues

Glyphs

Standard Glyph Definitions:

- We can now avoid the limitations of the previous INT 10h character support.

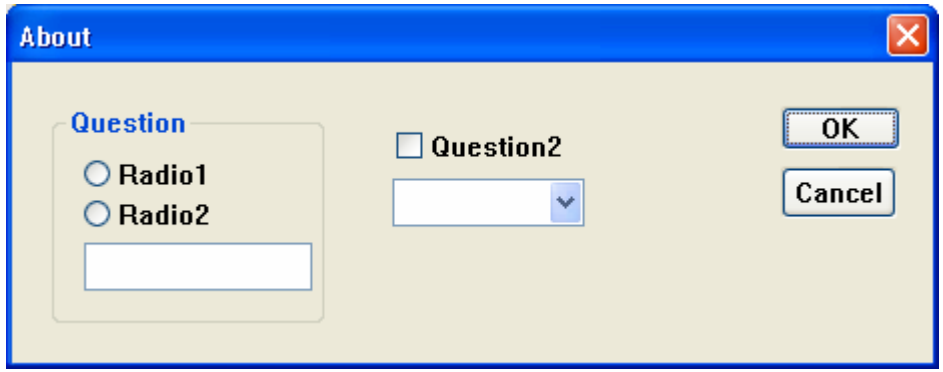




Providing output support for international venues

Forms

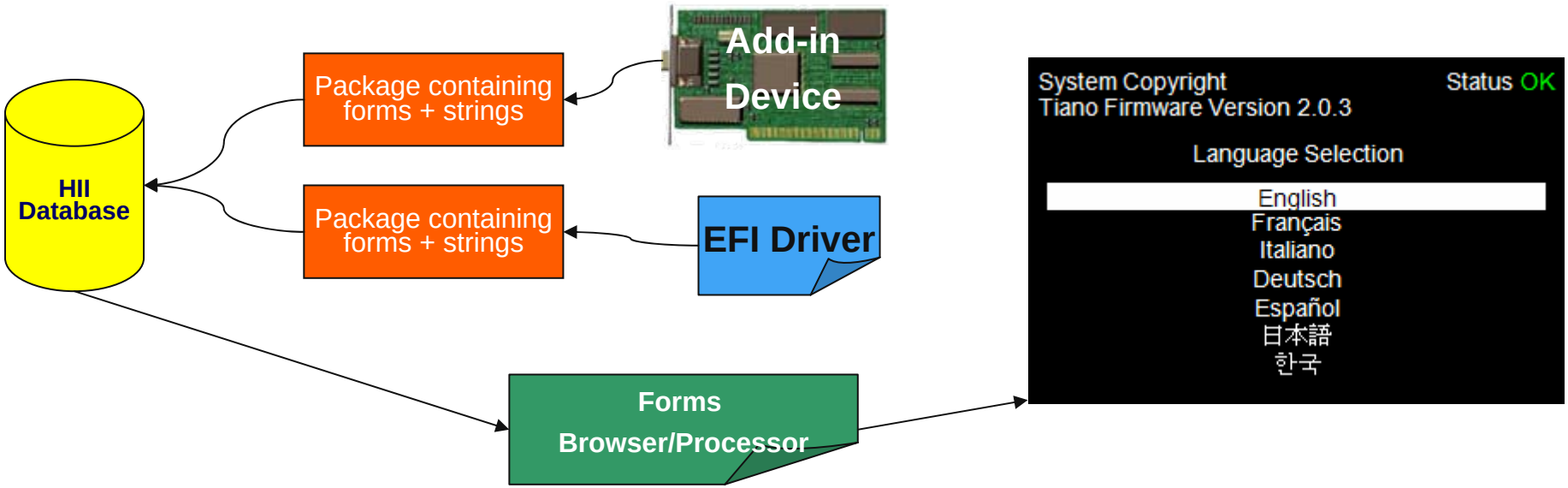
- Forms-based model for setup question descriptions
  - Must meet BIOS requirements
    - Scalable UI display support (Server Front Panel to local high resolution monitor).
    - Small encoding size
  - Encoding that is Self Describing
  - Position Independent
  - Can support scripting
  - Extensible syntax



- Exact look and feel defined by the browser and not defined in UEFI 2.1.
  - Developer/OEM/IHV defines questions to ask and what strings to display
  - Browser determines “how” to display the questions

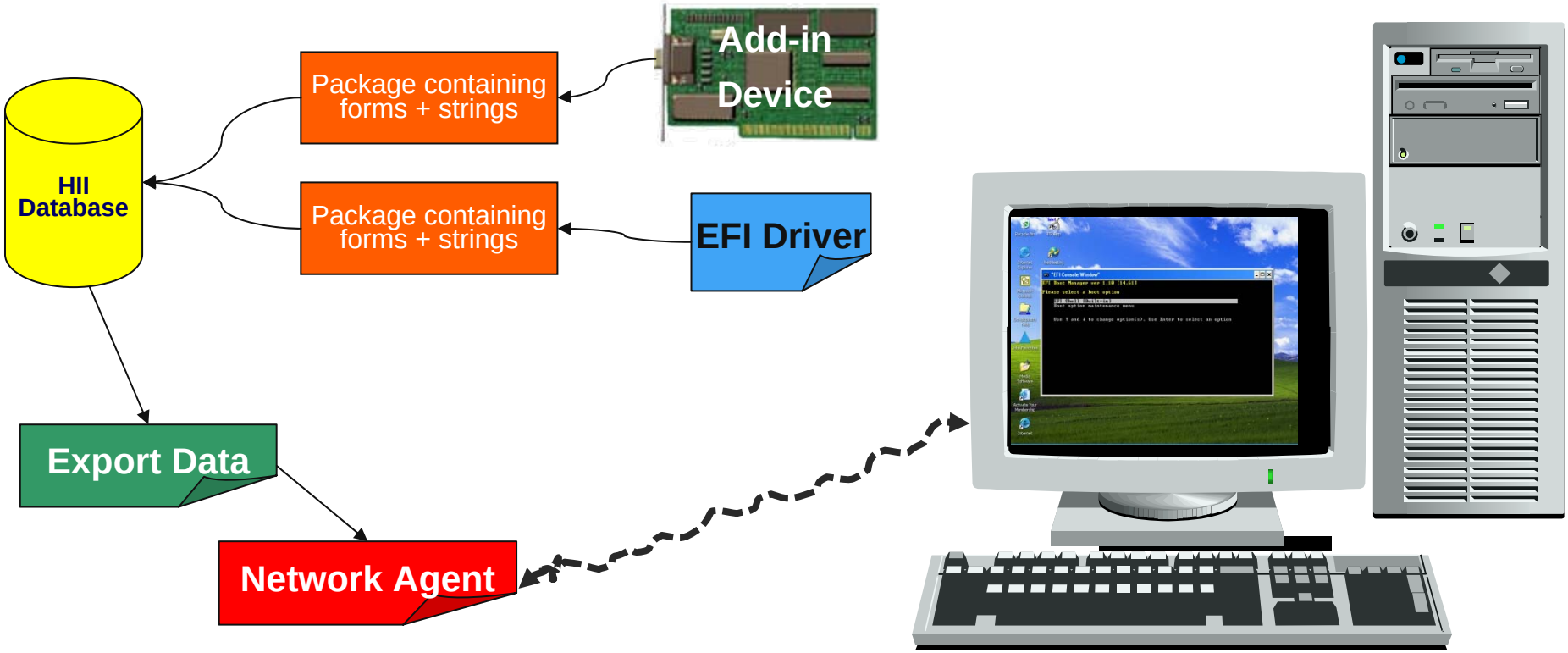
Enable UI infrastructure without dictating look-and-feel

Local Configuration



Local UI/configuration interaction enabled

Remote Configuration



Remote UI/configuration interaction enabled

Variable Updates

- Four new architectural variables defined
  - HwErrRecSupport
    - ✓ Defines if Hardware Error Record Persistence supported
    - ✓ 0 – No support exists / 1- Support exists
    - ✓ Firmware codebase responsible for setting this value
  - HwErrRec####
    - ✓ Hardware error record entry. #### is a printed hex value
    - ✓ A standard format for the error record is also defined in UEFI 2.1
  - Key####
    - ✓ Associate a key press with a single boot option. #### is a printed hex value.
  - BootOptionSupport
    - ✓ Determines if a platform supports optionally treating boot targets as applications Associate a key press with a single boot option. #### is a printed hex
- Two new variable attributes defined
  - EFI_VARIABLE_HARDWARE_ERROR_RECORD
    - ✓ Indicates that a variable is a hardware error record
  - EFI_VARIABLE_AUTHENTICATED_WRITE_ACCESS
    - ✓ Adds capability for a platform owner to ensure that variables are only updated in a owner-authorized fashion.

Standardized error records and authenticated variables

Extended Simple Input Support

- Adjustments to UEFI input support included:
  - Registration for a hot-key event
    - ✓ Ability to establish notifications when a particular key combination is pressed.
  - Clarifications/extensions for EFI Scan Codes
    - ✓ Introduced some new keys (e.g. VolumeUp, Mute, Eject, etc)
  - Retrieve added key state information
    - ✓ Key shift state information
      - For example, RIGHT_CONTROL_PRESSED
    - ✓ Key toggle state information
      - For example, CAPS_LOCK_ACTIVE
  - Ability to set various state values
    - ✓ Ability to toggle certain internal key state values such as CAPS_LOCK_ACTIVE.

Expanded key input support

Application Registration Support

- Enable third-party applications to register for execution
  - LOAD_OPTION_CATEGORY_BOOT
    - ✓Boot options which are to be treated as part of the normal boot process.
  - LOAD_OPTION_CATEGORY_APP
    - ✓Executables which are not part of the normal boot process.
- Ability to associate a hot-key with a boot target.

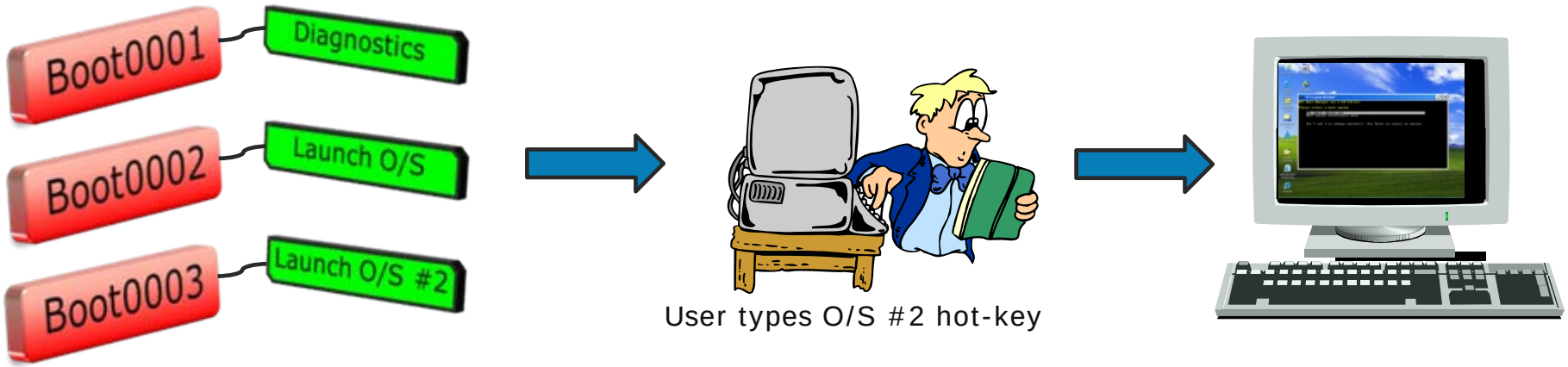


Table Support Updates

- Standardize the installation of ACPI tables in UEFI
  - ACPI Table protocol added since there are multiple agents in the system which might wish to install static ACPI tables, therefore standardization is desired.
- Add standard UEFI ACPI table format
  - To prevent ACPI namespace collision, a UEFI ACPI table format is defined. This allows creation of ACPI tables without colliding with tables reserved in the namespace.
- Alert when UEFI Configuration Table changes.
  - Adjust the underlying InstallConfigurationTable support so that when it is called, it will signal an event indicating that a specific GUID's configuration entry is being updated.

Enable Table Change Notification and Cleaner ACPI Table Usage

Absolute Pointer Protocol

- Add support for an absolute pointer protocol.
  - Absolute X/Y coordinate support for devices such as digitizers, PDA's, pen-based systems, etc.

A pointer does not mean only a mouse anymore

More Details in the UEFI 2.1 Specification

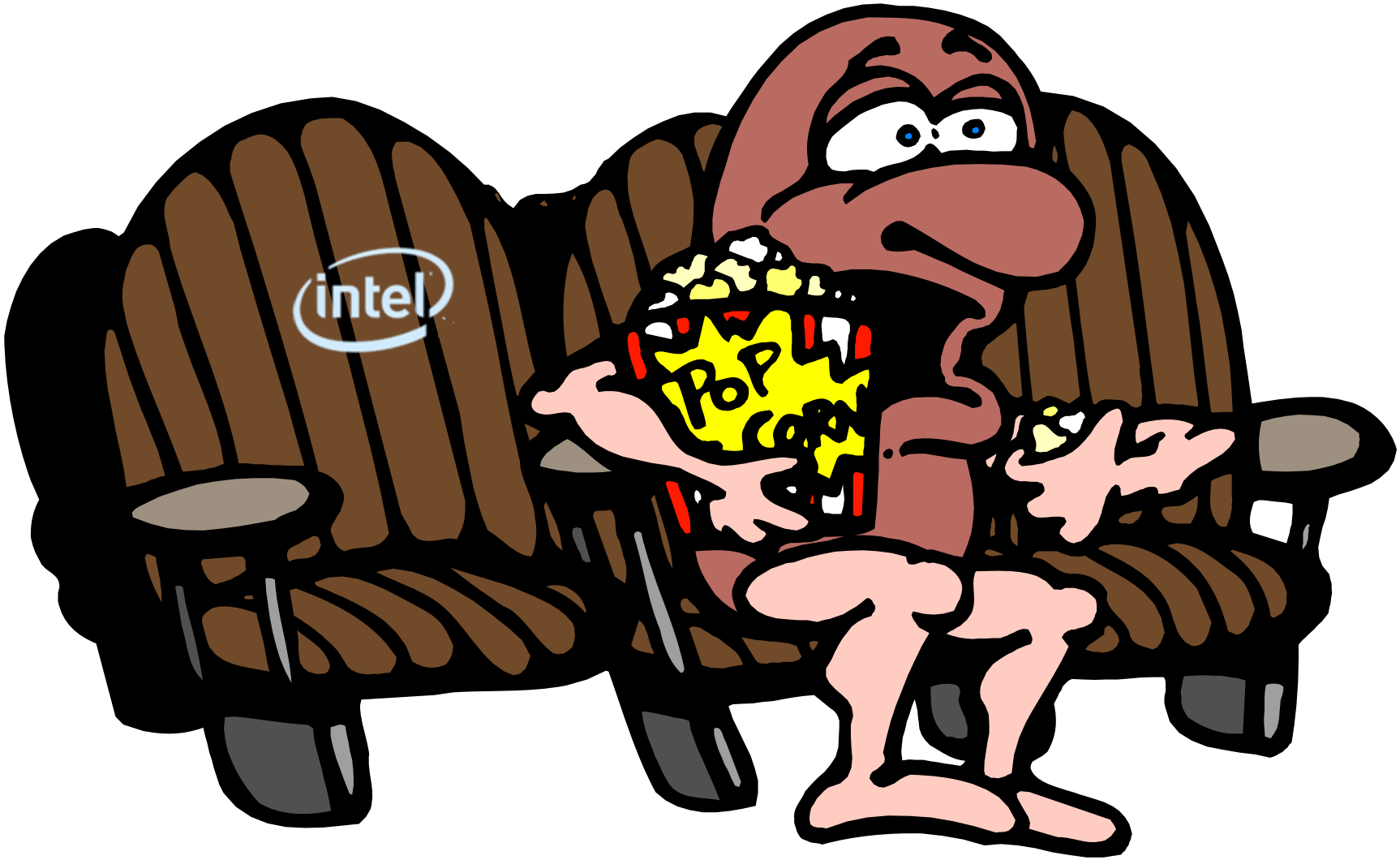
- See the UEFI web site (www.uefi.org) for a more comprehensive list of the changes between UEFI 2.0 and UEFI 2.1.

The UEFI Standard Continues To Evolve

Agenda

- A look at EFI and UEFI Overview
- UEFI 2.1 New Content and Changes
- Concept Demo
- PI 1.0 Content and Changes
- Future Development and Test Plans

Concept Video

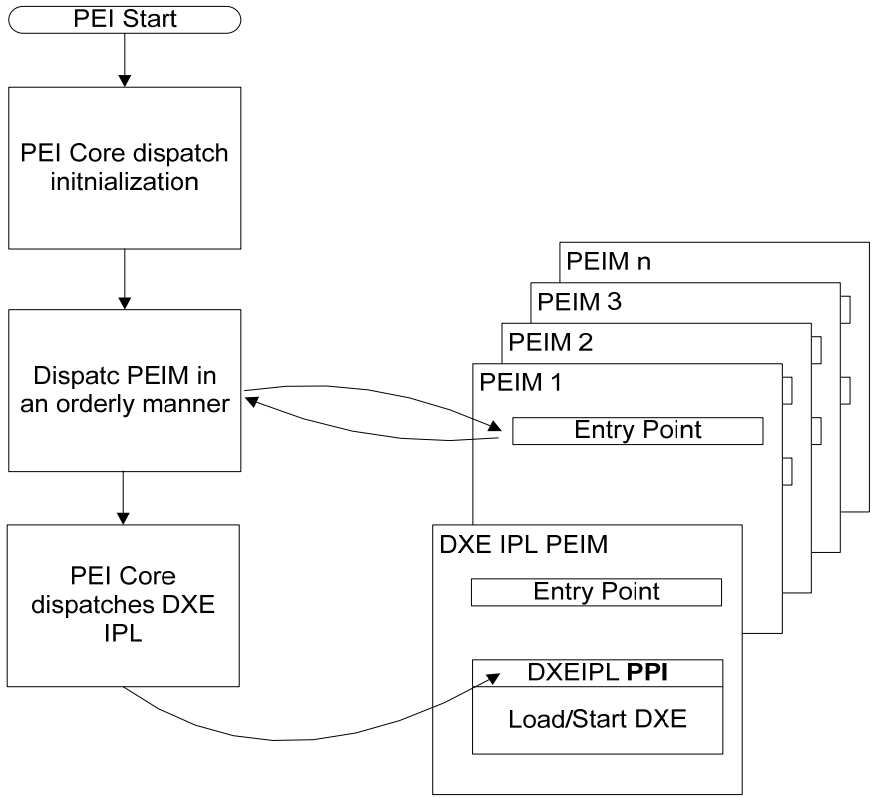


Agenda

- A look at EFI and UEFI Overview
- UEFI 2.1 New Content and Changes
- Concept Demo
- PI 1.0 Content and Changes
- Future Development and Test Plans

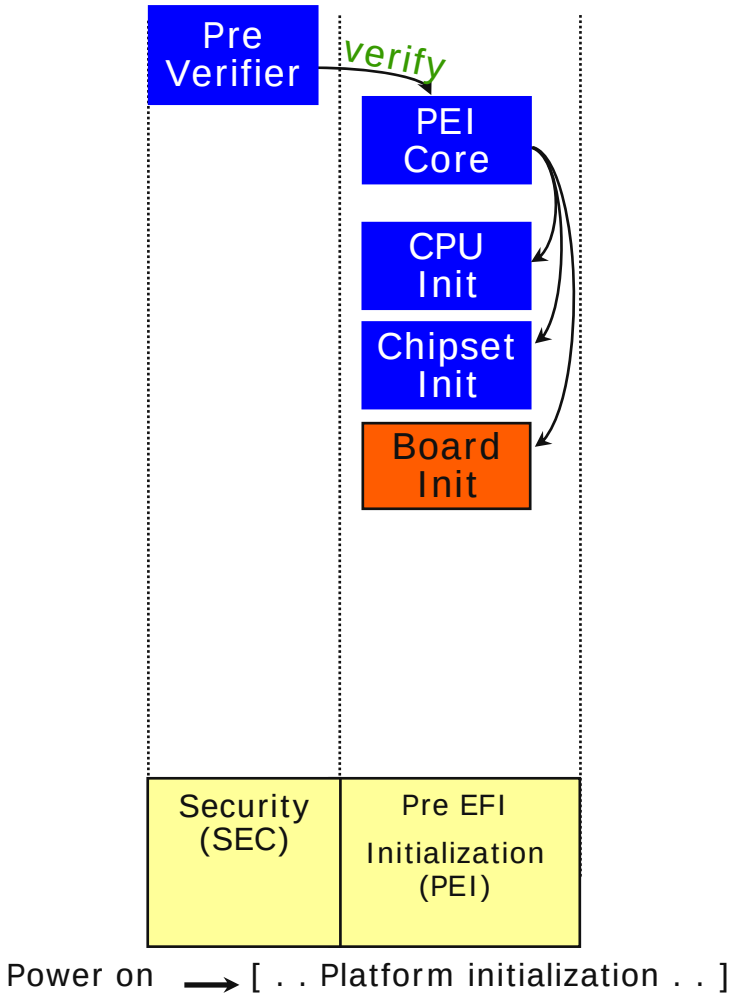
PEI Theory of Operation

- Consumes reset, INIT, MCA
- Small, tight startup code
  - Starts as XIP from ROM
- Leverage new architectural support in upcoming IA CPUs
  - “Cache in lieu of RAM”
  - Gets us to C closer to reset
- Core locates, validates, and dispatches PEIMs
- Primary goals
  - **A standard method for delivering silicon modules.**
  - Discover boot mode
  - Launch modules that initialize main memory
  - Discover & launch DXE core



PI specification describes architecture starting from the reset vector

Early Boot Overview



Transition from PEI to DXE

- PEI gives way to DXE
  - Hand off from one to the other, PEI dematerializes
  - Work deferred to DXE whenever possible
- Memory map and resources discovered in PEI passed on to DXE
- Hand Of Blocks (HOBs)
  - set of linked data structures
  - Memory, firmware stores, platform resources, boot mode, etc.
- Last PEI Module is Initial Program Load for DXE
  - HOB list passed in as argument to DXE “main”

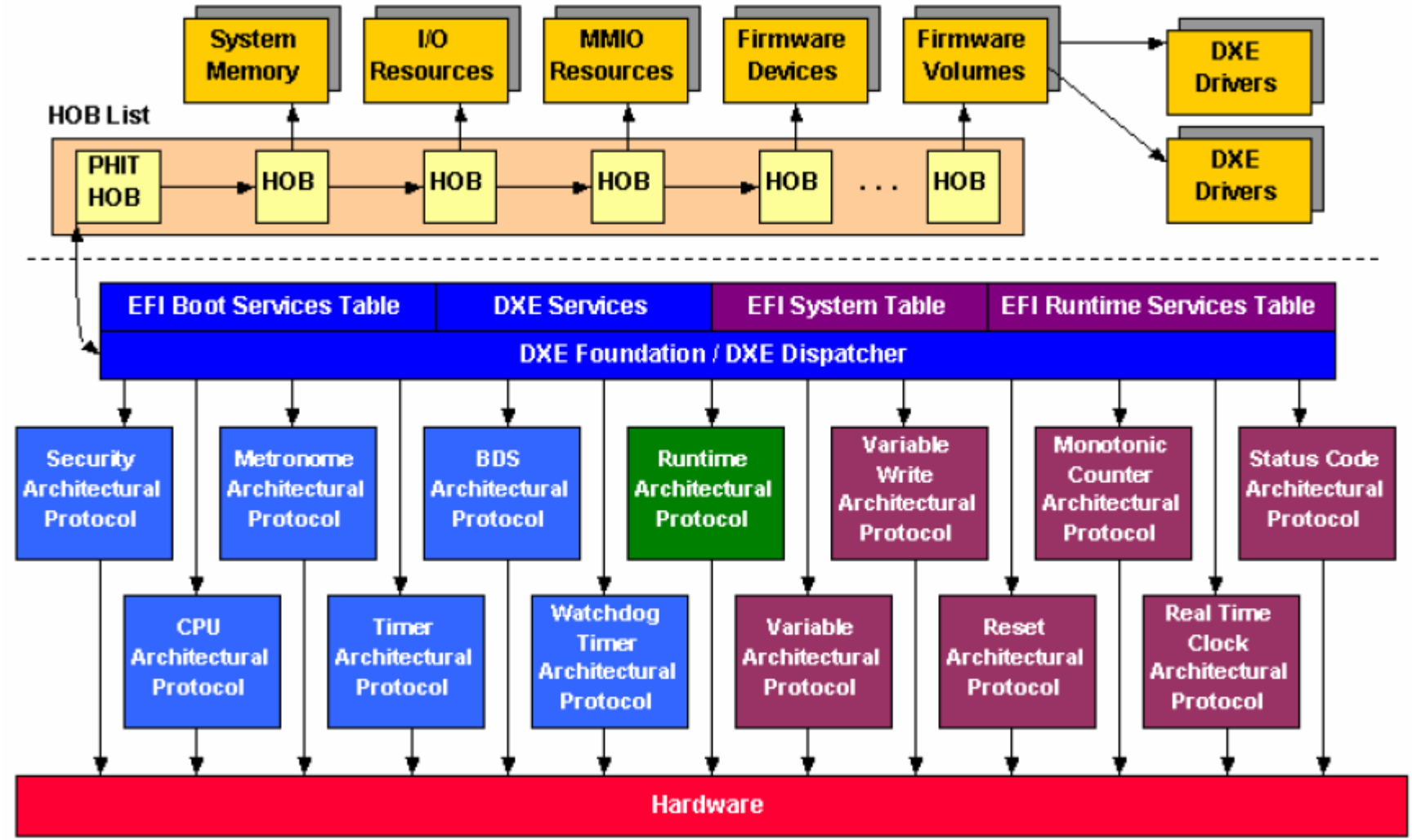
Formal hand-off defined between early and later initialization phases

DXE Properties

- Depends only on HOB list
  - State initialization passed in from PEI
- No hard coded addresses in DXE
  - Foundation code can be loaded anywhere
- No hardware specifics in DXE Foundation
  - Access to hardware abstracted by a set of architectural protocols (APs)
  - APs implemented as drivers
  - Only DXE Foundation may call APs
  - APs encapsulate CPU, chipset, board specifics

DXE's view of the system is strictly based on PEI's HOB data

DXE Overview

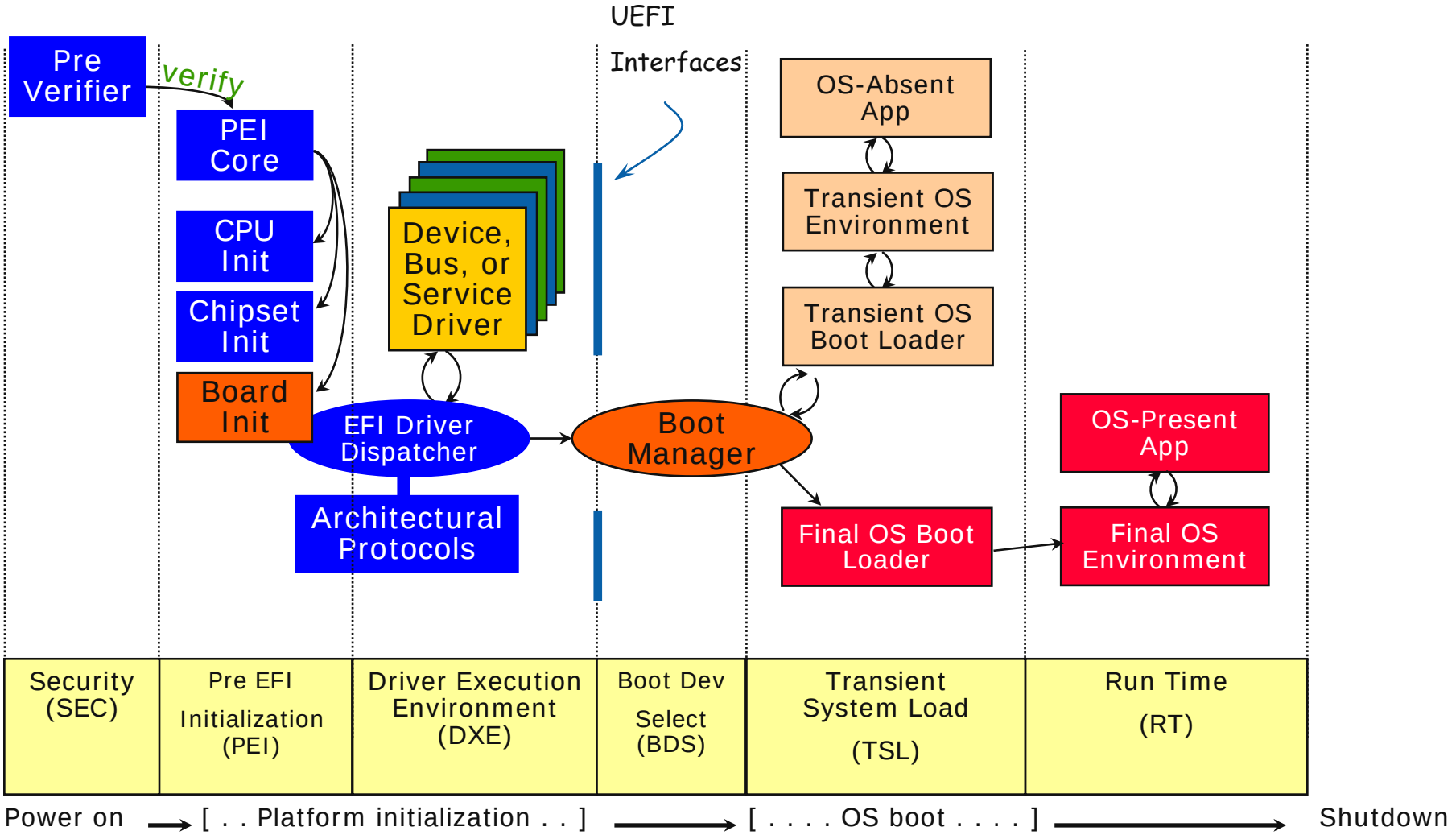


DXE Theory of Operations

- First goal: determine boot target
  - Required boot device and console devices
- Loads drivers to construct environment that can support boot manager and OS boot
- Dependencies provide driver ordering
  - Grammar-based description of drivers' requirements
  - Including patch or override operations e.g. with "before/after" dependencies
- EFI drivers with no dependency started last
  - Compatibility for UEFI drivers, IHV cards etc.
- Dispatch completes as fast as practical
  - Required hardware init performed by driver on call to entry point
  - EFI driver entry points just register protocol
  - Defer initialization of boot devices until we know which are needed
- When all required drivers are loaded go to boot manager to attempt to boot

DXE instantiates UEFI interfaces and launches the boot target

Overall View of Boot Time Line



Overview of Differences – PI 1.0 Vs. Framework Components

	Component	Actions / Exceptions
✓	Compatibility	Do not access internals of the firmware files Do not use ReportStatusCode
✓	PEI File System	Minor change to the file header and firmware volume header
✓	PPI Updates	PCI PPI for Extended PCI-express New PPI – Terminate End of Temp Memory
✓	DXE Service Table	Removed Report Status Code service
✓	New Architectural Protocol	Capsule AP / QueryVariableInfo
✓	HOB definitions	More Firmware volume information Remove Capsule HOB definition

PI 1.0 Introduces Standards To Early Boot

Agenda

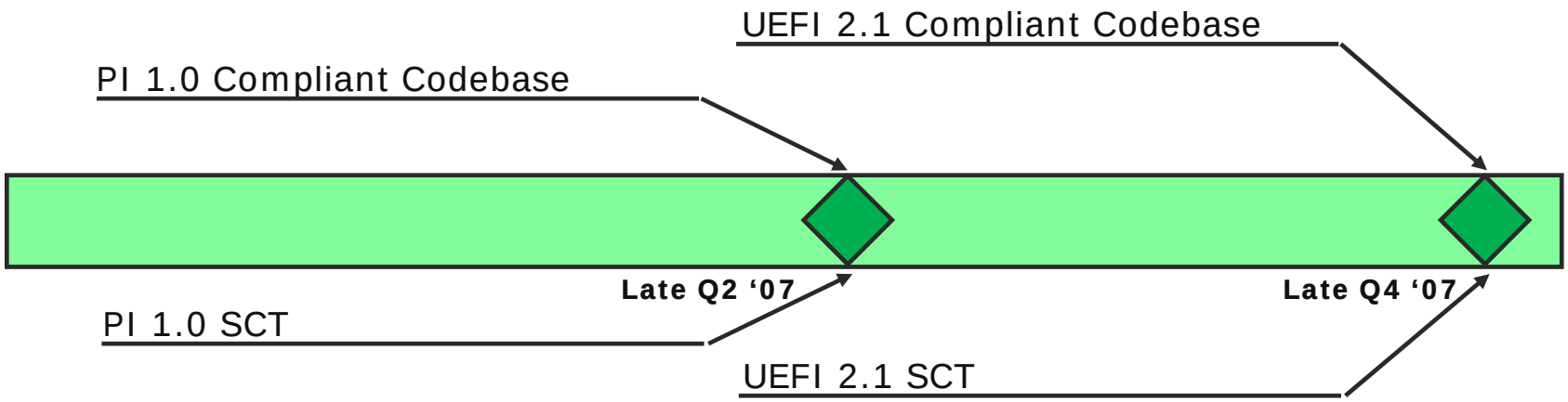
- A look at EFI and UEFI Overview
- UEFI 2.1 New Content and Changes
- Concept Demo
- PI 1.0 Content and Changes
- Future Development and Test Plans

Some Future work items

- In the various UEFI workgroups there are many ongoing efforts.
  - USWG
    - ✓ More security content
    - ✓ Continued HII enhancements for interacting with other standards-based namespaces.
    - ✓ Continued evolution of networking components (e.g. IPv6, PXE, IPSec, etc)
  - PIWG
    - ✓ More work on added standardization of firmware interaction/use of various technologies such as PCI, ACPI, SMM, MP, and S3

Lots of work ongoing and encourage added participation

Deploying Code



Self Certification Tests (SCT) for PI 1.0 and UEFI 2.1 will be completed in conjunction with their associated codebases.

Check UEFI.org and Tianocore.org for new material

Summary

UEFI 2.1 Spec is complete and available

- Ongoing work to improve interoperability standards for the UEFI Specification.

PI 1.0 Spec is complete and available

- Ongoing work on additional standards to augment and improve Platform Initialization (PI) Specification.

UEFI Forum is the standard place where firmware evolution is discussed

UEFI x64 OS updates

- **Windows***
  - See Microsoft and IBM UEFI session (EFIS001) for details on Windows* support schedule in Windows Server 2008 and Vista SP1
- **Linux open source**
  - Support released part of main kernel 2.6.24+. See www.kernel.org
  - Intel is working with Red Hat* to address UEFI in the next major release of RHEL*.
  - Intel is working with Novell to address UEFI in the next major release of SLES*
  - ***Linux UEFI Tools:***
    - ELILO – See Sourceforge project for x64 support http://www.sourceforge.org/ELILO
    - GNUEFI library – needed to build ELILO. Soureforge project http://sourceforge.net/projects/gnu-efi
    - GRUB 1.0 for UEFI boot – open source submitted (review in progress)
    - Binutils 2.17.50.0.14.tar.bz2 or Higher www.kernel.org/pub/linux/devel/binutils
- **Apple MAC OS***
  - MAC OS X with native UEFI support (available since 2006)
- **HP-UX* and Open VMS***
  - All Intel® Itanium Processor Platform versions

Join and deploy UEFI

UEFI Forum encourages active participation

- Although anyone can get and read the specs

Become a Contributor

- Early access to specs in progress
- Provide input and direction for spec work
  - Via email or participation in WG deliberations
- Not an obligation to commit resources or product
  - …although that would be great if you do!

Become an Adopter

- Simple sign up for a license to implement

Call to Action!

- **UEFI Testing Event**
  - In Sunnyvale California Sept 25-28 2007
  - Purpose
    - Provide the an opportunity to allow implementers of UEFI to test their implementations among the UEFI community
    - Testing of UEFI systems and platforms with UEFI Add in Cards in different configurations for UEFI compliance as well
    - Testing install and boot to a variety of UEFI Operating systems
  - More on this event: www.uefi.org

Additional sources of information on this topic:

- **Visit the UEFI Community in the IDF showcase**
- **More web based info:**
  - www.tianoCore.org
  - www.uefi.org
  - www.intel.com/technology/framework
- **Technical book from Intel Press:**
  - *"Beyond BIOS: Implementing the Unified Extensible Firmware Interface with Intel's Framework"*

    For more info: www.intel.com/intelpress

- This Session presentation (PDF) is available from www.intel.com/idf. Some sessions will also provide Audio-enabled presentations after the event.

Additional UEFI /Framework Sessions Moscone West 2007:

Session	EFI #	Company	Time
UEFI 2.1 and UEFI Platform Initialization (PI) 1.0 - Details and Differences	S004	Intel	10:00 AM
"Zero to OS in a Flash" - Intel's Framework solution for HPC and Embedded Applications	S002	Intel	11:00 AM
UEFI Benefits for IBM* Product Development and Microsoft Update on Windows* UEFI Support	S001	IBM / Microsoft	3:00 PM
PC Client, Revolutionary Embedded Software, Pathway to the Future	S003	Hewlett Packard	4:10 PM
Q&A open forum Chalk Talk Room	C001	Intel	5:10 PM

Please fill out the Session Evaluation Form for your chance to win a $500 Gift card!

How?

- Use your IDF Flash Drive
- Go to an IDF Internet Station
- Go to www.Intel.com/go/myidfeval

There will be daily drawings for Gift cards – The more evaluations you fill out the more chances to win!

See drawing terms and condition in Program Guide for more information including alternative means of entry.

Risk Factors

This presentation contains forward-looking statements. All statements made that are not historical facts are subject to a number of risks and uncertainties, and actual results may differ materially. Please refer to our most recent Earnings Release and our most recent Form 10-Q or 10-K filing available on our website for more information on the risk factors that could cause actual results to differ.

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.
- Intel may make changes to specifications and product descriptions at any time, without notice.
- All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- Intel, Intel Inside and the Intel logo are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-


intel
Leap ahead™