Overview of the UEFI Forum

Presented by Dong Wei (ARM)

Agenda

- Introduction
- Background Information
- UEFI Forum/DMTF Work Register
- Mapping UEFI and Redfish™
- Summary

Background Information

UEFI Forum Overview

- Non-profit industry forum
- Founded in 2005
- Formed to standardize EFI and extend to x64
- Forum maintains all specification development
- Currently at over 330 member companies and individual adopters

Why Become a UEFI Member?

Membership Profiles

- System Manufacturers (server, client, mobile, IoT)
- Silicon Providers
- Firmware Vendors
- Computer Peripheral/Hardware Vendors
- Software Vendors
- Operating System Developers
- Industry Advisors
- Best Practices Stewards
- Academics

Membership Levels

- Adopter (complimentary)
  - Access to the Members-only web area
  - Invitations to member events
  - Access to UEFI technical tools and design guides
- Contributor ($2500 annual fee)
  - Adopter benefits, plus:
    - Participation in UEFI Work Groups, by invitation
    - Participation in email reflectors
    - Access to draft specifications

Promoter Members

MARK DORAN
President
Intel

DONG WEI
Vice President
ARM

JEFF BOBZIN
Secretary
Insyde Software

BILL KEOWN
Treasurer
Lenovo

GARY SIMPSON
Advanced Micro Devices, Inc.

STEFANO RIGHI
American Megatrends, Inc.

ANDREW FISH
Apple

RICHARD HOLMBERG
Dell

KEVIN DEPEW
Hewlett Packard Enterprise

LAN WANG
HP, Inc.

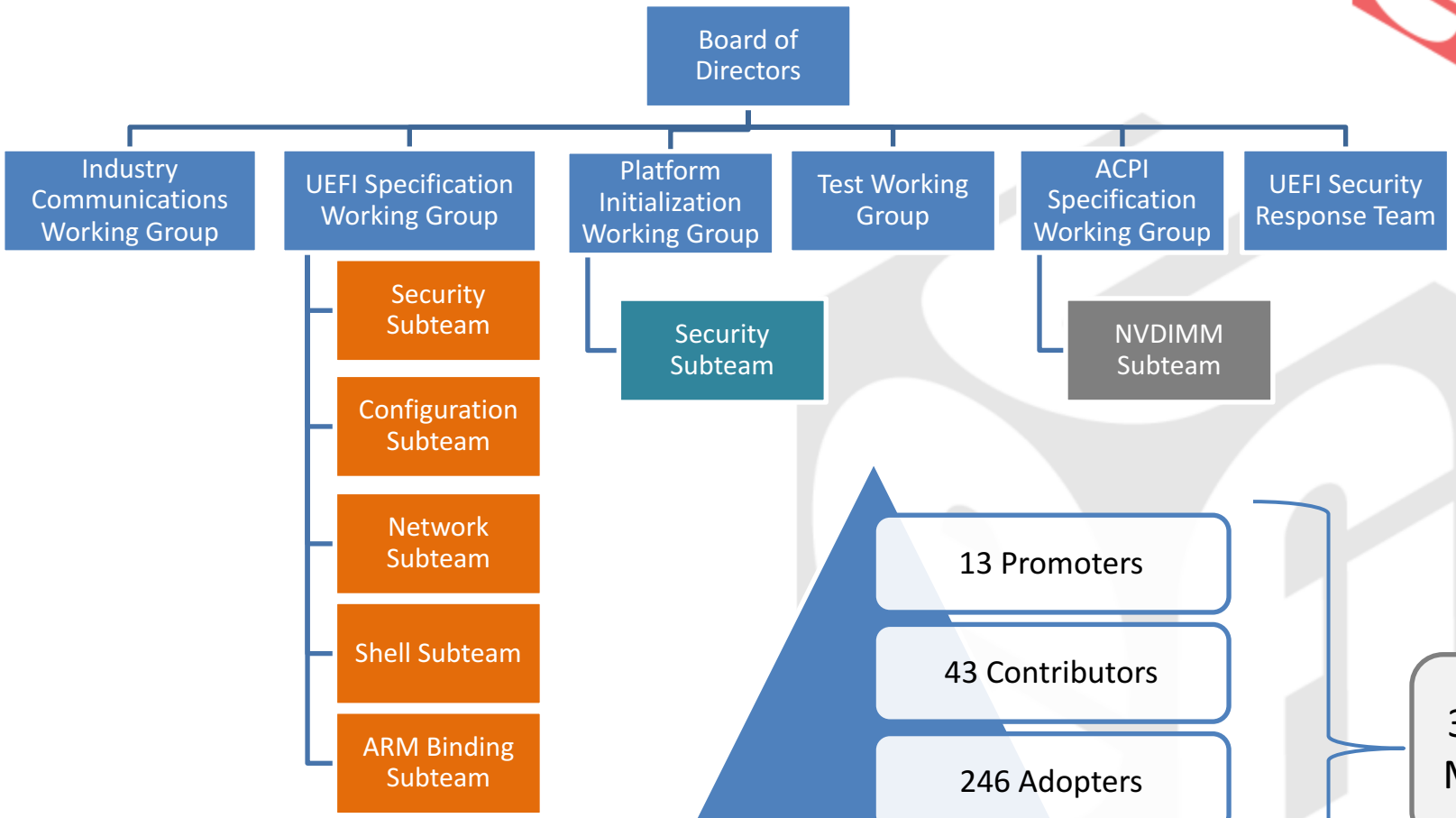
JEREMY KERR
IBM

TOBY NIXON
Microsoft

DICK WILKINS
Phoenix Technologies

Contributor Members

UEFI Forum Overview



UEFI & ACPI Specifications

- **Unified Extensible Firmware Interface (UEFI)**
  - Defines firmware interface in pre-OS space
  - Standardizes platform interfaces for interoperability
  - Extensible across all platforms
  - Architecture-agnostic
    - Currently officially supports IA64, ia32, x64, ARM AArch32 and ARM AArch64
    - RISC-V support coming
- **Advanced Configuration and Power Interface (ACPI)**
  - Key element in OS-directed configuration and Power Management (OSPM)
  - Flexible mechanisms for device discovery, thermal management and reliability, availability and supportability (RAS) features
  - Enables platform technologies to evolve independently in the operating system and hardware

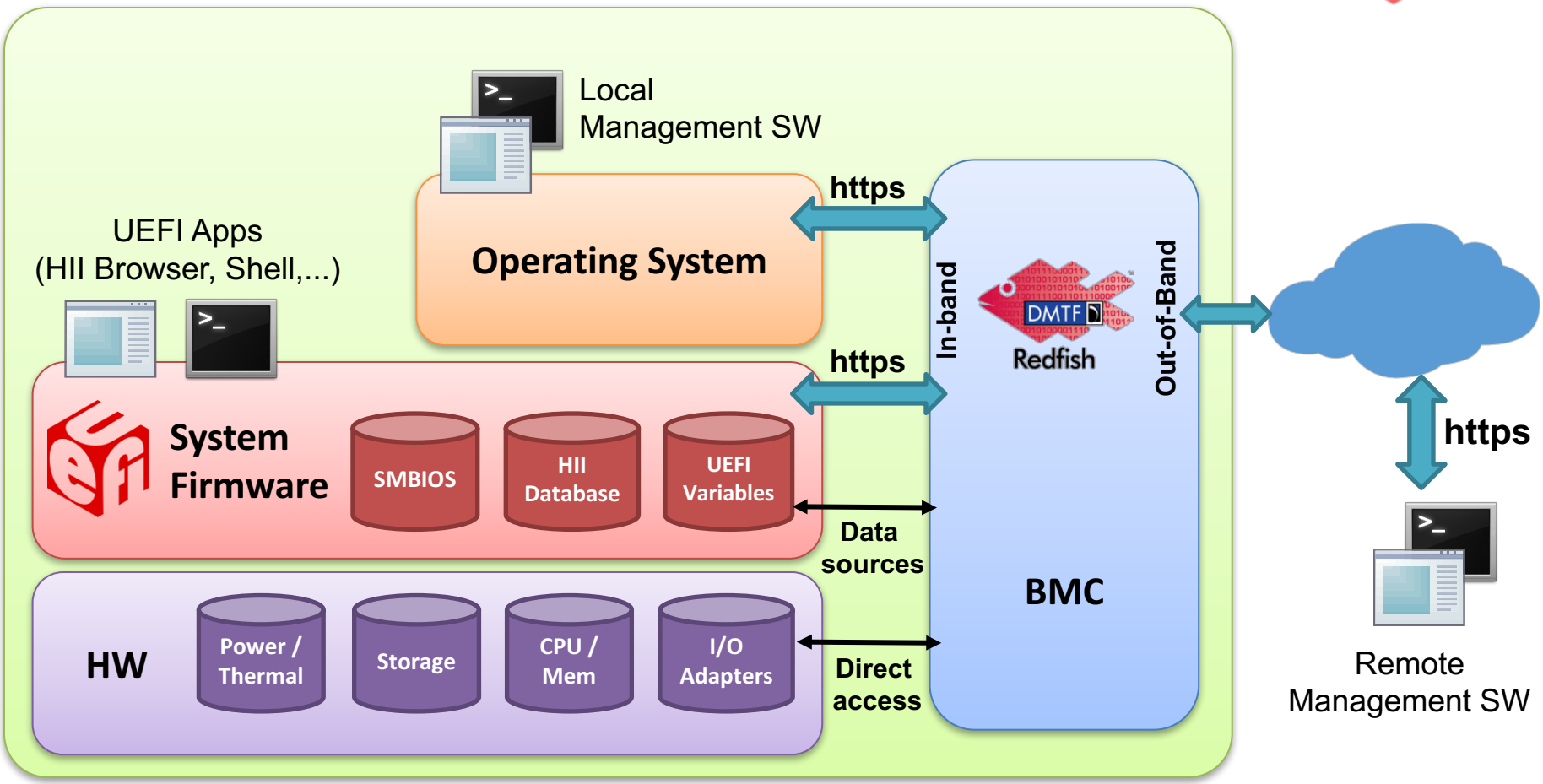
UEFI Forum/DMTF Work Register

Mission/Benefits

- Alliance Partner Mission
  - The goal of this coordination is to incorporate requirements for modeling UEFI compliant firmware implementations into system and firmware models developed by the DMTF.
- Alliance Benefits
  - Ensure the DMTF and UEFI standards are coordinated and address platform inventory and configuration management requirements
  - Consistent interpretation of ACPI specifications
  - Promote DMTF and UEFI standards to member companies
  - UEFI participation at the DMTF Developers Conference and in various working groups
  - Participate in Alliance Partner Summit

Mapping UEFI and Redfish™

Systems Management with Redfish™ and UEFI

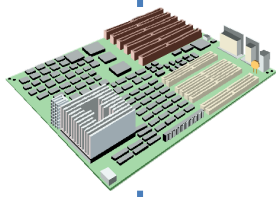


UEFI REST communication

- EFI_REST_PROTOCOL and BMC Device Path
  - Standard in-band access to a REST API, like Redfish
  - Abstracts BMC-specific access methods from UEFI (proprietary)
  - EFI_HTTP_UTILITY_PROTOCOL to build/parse HTTP headers

- UEFI HTTP/TLS
  - EFI_HTTP_PROTOCOL and EFI_TLS_PROTOCOL to Implement HTTPs communication between UEFI and BMC
  - Using HW (physical) or SW (virtual, proprietary) NIC

HII / Redfish™ Flow



Register HII Forms, Strings

HII Database

Config Access Protocol

HII Browser

Config Access Protocol

HII Pkgs: Forms, Strings (display, x-UEFI)

HII Config (x-UEFI)

Keyword Handler Protocol

HII Keywords to Redfish JSON

HII Forms/Strings to Redfish JSON

HII Keywords (x-UEFI)

Bios Attributes

JSON

Bios Attributes Registry

BMC

DMTF Redfish

HII / Redfish™ Mapping

- x-UEFI name/value pairs translated to Redfish BIOS Attributes
  - OData v4 Restrictions on Attribute Names and Enums (Oneof possible values)
    - Alphanumeric, and “_”. No spaces or other special characters
    - Cannot start with a digit
- HII Forms (IFR) and String Packages to Attribute Registry
  - Mostly 1-to-1
  - One Registry per Display Language
  - Limited IFR Dependency rules support

HII Settings :: Attributes

https://<ip>/redfish/v1/Systems/1/BIOS


```
{
  "@odata.id": "/redfish/v1/Systems/1/Bios",
  "@odata.type": "#Bios.v1_0_0.Bios",
  "AttributeRegistry": "BiosAttributeRegistryP89.v1_0_0",
  "Actions": {
    "#Bios.ResetBios": {
      "target": "/redfish/v1/Systems/1/Bios/Actions/Bios.ResetBios"
    },
    "#Bios.ChangePassword": {
      "target": "/redfish/v1/Systems/1/Bios/Actions/Bios.ChangePassword"
    }
  },
  "Attributes": {
    "BootMode": "Uefi",
    "EmbeddedSata": "Raid",
    "NicBoot1": "NetworkBoot",
    "NicBoot2": "Disabled",
    "PowerProfile": "MaxPerf",
    "ProcCoreDisable": 0,
    "ProcHyperthreading": "Enabled",
    "ProcTurboMode": "Enabled",
    "UsbControl": "UsbEnabled"
  }
}
```


Reset to defaults

- EFI_IFR_RESET_BUTTON_OP (global reset of entire Form Set)

Passwords

- Special case to avoid exposing clear text password in Attributes
- Must provide old and new passwords (authenticate as BIOS user/admin)

Attributes

- Direct translation of HII Keywords name/value pairs (with Odata rules)
- Value can be numeric, string, Boolean, or enum (one of)
- OEM/IBV specific

HII Forms :: Attribute Registry

https://<ip>/redfish/v1/Registries/ BiosAttributeRegistryXYZ.v1_0_0

```
{
  "@odata.type":
"#AttributeRegistry.v1_0_0.AttributeRegistry",
  "Description": "This registry defines a
representation of UEFI HII BIOS Attributes",
  "Id": "BiosAttributeRegistryXYZ.v1_0_0",
  "Language": "en",
  "Name": “System X BIOS Attribute Registry",
  "OwningEntity": “Lenovo",
  "RegistryVersion": "1.0.0",
  "Attributes" : [
     ...
  ],
  "Menus" : [
     ...
  ],
  "Dependencies" : [
     ...
  ]
}
```

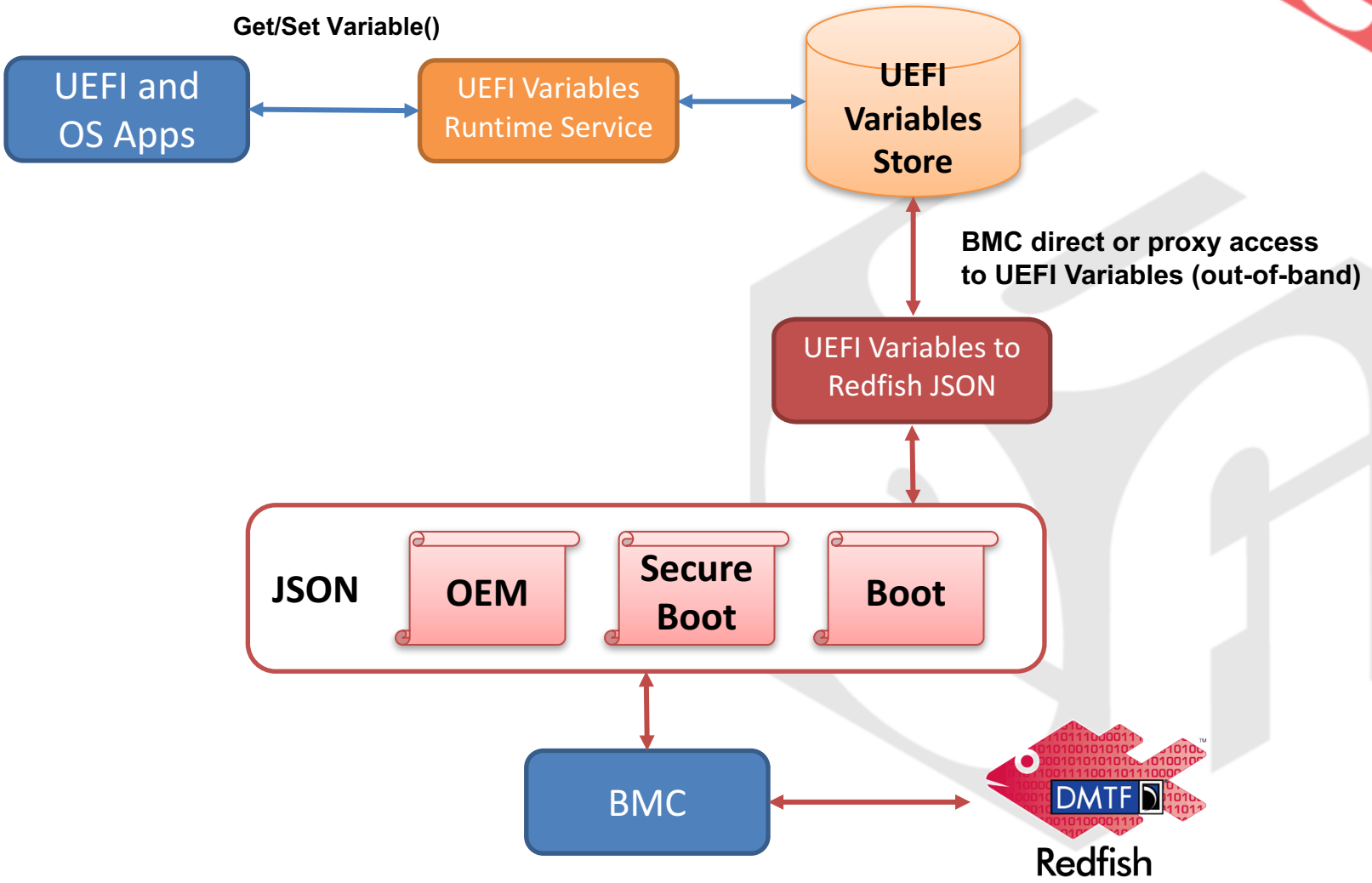
Localized Display Strings

- Each Attribute Registry instance maps to 1 language in the HII String Packages

IFR Forms and String Packages

- 1 Attributes[] entry for each HII IFR Question
- 1 Menus[] entry for each HII Form
- Dependencies[] describe conditions between questions (Such as GrayoutIf, SuppressIf, Map, etc..)

UEFI Variables / Redfish ™ Flow



UEFI BootOrder and BootNext

https://<ip>/redfish/v1/Systems/1

```
{
 "Boot": {
    "BootSourceOverrideEnabled": "Once",
    "BootSourceOverrideMode": "UEFI",
    "BootSourceOverrideTarget": "Pxe",
    "BootSourceOverrideTarget@Redfish.AllowableValues": [
        "None",
        "Pxe",
        "Floppy",
        "Cd",
        "Usb",
        "Hdd",
        "BiosSetup",
        "Utilities",
        "Diags",
        "UefiTarget",
        "SDCard",
        "UefiHttp"
    ],
    "UefiTargetBootSourceOverride": "UEFI device path"
  }
}
```

Boot Order override

- "Disabled" = No override
- "Once" = BootNext
- "Continuous" = Move to top of BootOrder (or create BootNext on every boot)

Boot Mode: UEFI / LegacyBios

Boot device selection:

- "Pxe", "UefiHttp" → 1st or any NIC
- "Usb" → USB Class
- "SDCard" →
- "Hdd" → Fixed media, iSCSI
- "Cd" → optical drives
- "Diags" → Embedded UEFI Diagnostics
- "Utilities" → Embedded management tool
- "BiosSetup" → Samer as OSIndications to enter UEFI Boot Manager
- "UefiTarget" → When selected, specify the complete UEFI Device Path in UEfiTargetBootSourceOverride below

UEFI Device Path (text representation) that goes in BootNext variable

UEFI SecureBoot

https://<ip>/redfish/v1/Systems/1/BIOS/SecureBoot


```
{
  "@odata.id": "/redfish/v1/Systems/1/SecureBoot",
  "@odata.type": "#SecureBoot.v1_0_0.SecureBoot",
  "Name": "UEFI Secure Boot",
  "Actions": {
    "#SecureBoot.ResetKeys": {
      "target": "/redfish/v1/Systems/1/SecureBoot/
Actions/SecureBoot.ResetKeys",
      "ResetKeysType@Redfish.AllowableValues": [
         "ResetAllKeysToDefault",
         "DeleteAllKeys",
         "DeletePK"
       ]
    },
  },
  "SecureBootEnable": false,
  "SecureBootCurrentBoot": "Disabled",
  "SecureBootMode": "UserMode",
  "Oem": {}
}
```


SecureBoot Key Management

- **DeletePK** = Puts system in SetupMode
- **DeleteAllKeys =** Remove all entries form PK/KEK/DB/DBX
- **ResetAllKeysToDefault =** PK/KEK/DB/DBX content reset

UEFI SecureBoot

- True : SecureBoot = 1 (Enabled)
- False : Secureboot = 0 (Disabled)

Applies on next reboot

UEFI SecureBoot Current Status

SecureBoot Mode (UEFI 2.6)

Read-only, current mode:
UseMode, SetupMode, AuditMode, DeploMode

Firmware Updates

- /redfish/v1/UpdateService/FirmwareInventory
  - FirmwareInventory
  - SimpleUpdate Action
- Model is
  - Give FW binary/package URI (HTTPs/FTP) to BMC
  - BMC downloads and applies (or stages) FW
- Back-end could be
  - UEFI Firmware Management Protocol (FMP)
  - UEFI Capsules
  - Other Technologies

Summary

Summary

- UEFI Forum is a widely-participated industry standard consortium
  - It owns the definition and promotion of the UEFI Specification and its Test Suite
  - In addition, it owns the definition and promotion of the Advanced Configuration and Power Interface (ACPI) and Platform Initialization Specification (PI Specifications)
- UEFI Forum and DMTF has a Work Register
  - UEFI Forum is an Alliance Partner to DMTF
- UEFI Forum and DMTF are collaborating in many areas, including Redfish

Thanks for attending the DMTF APTS

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

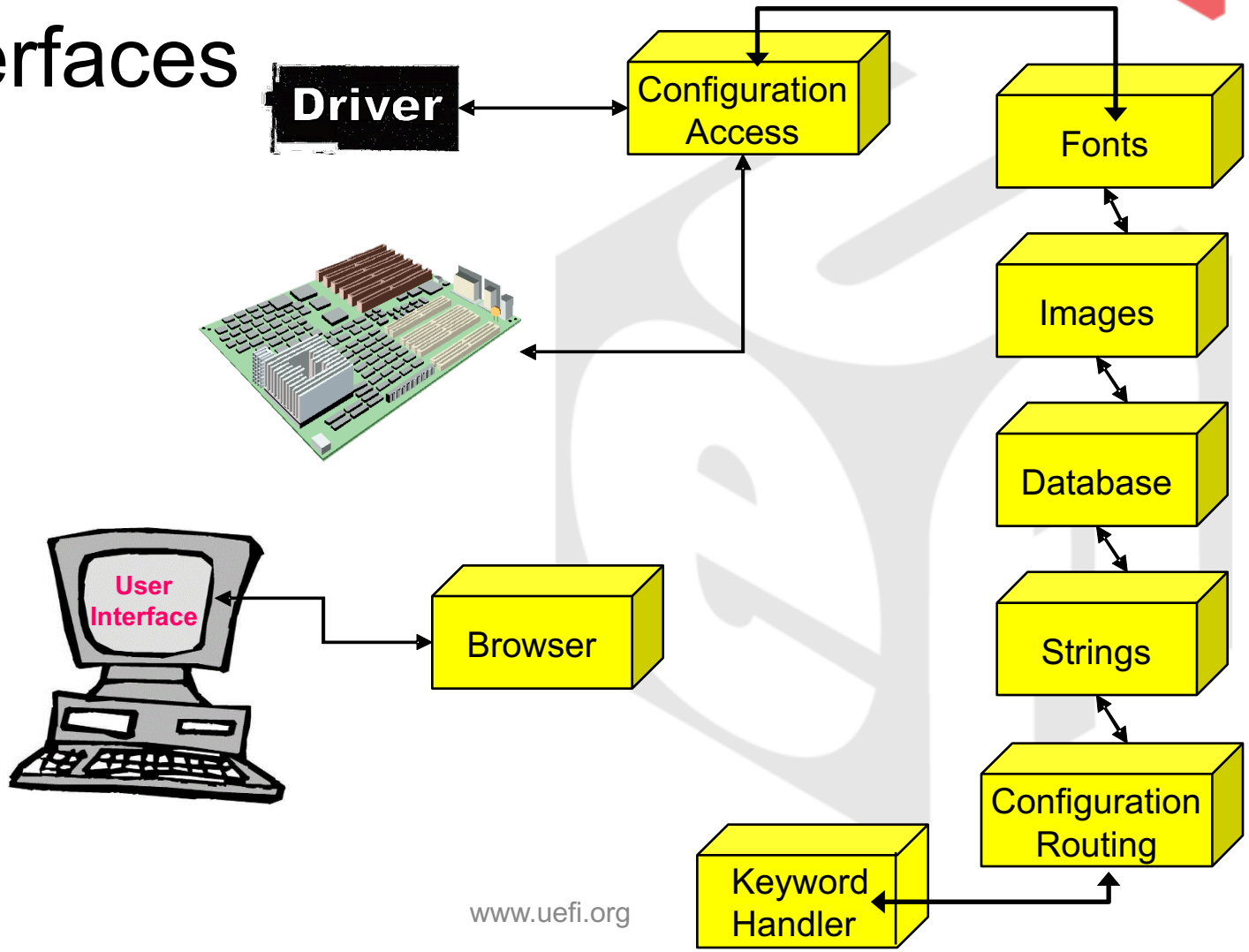
presented by

Dong Wei, ARM, The UEFI Forum

HII Data Flow

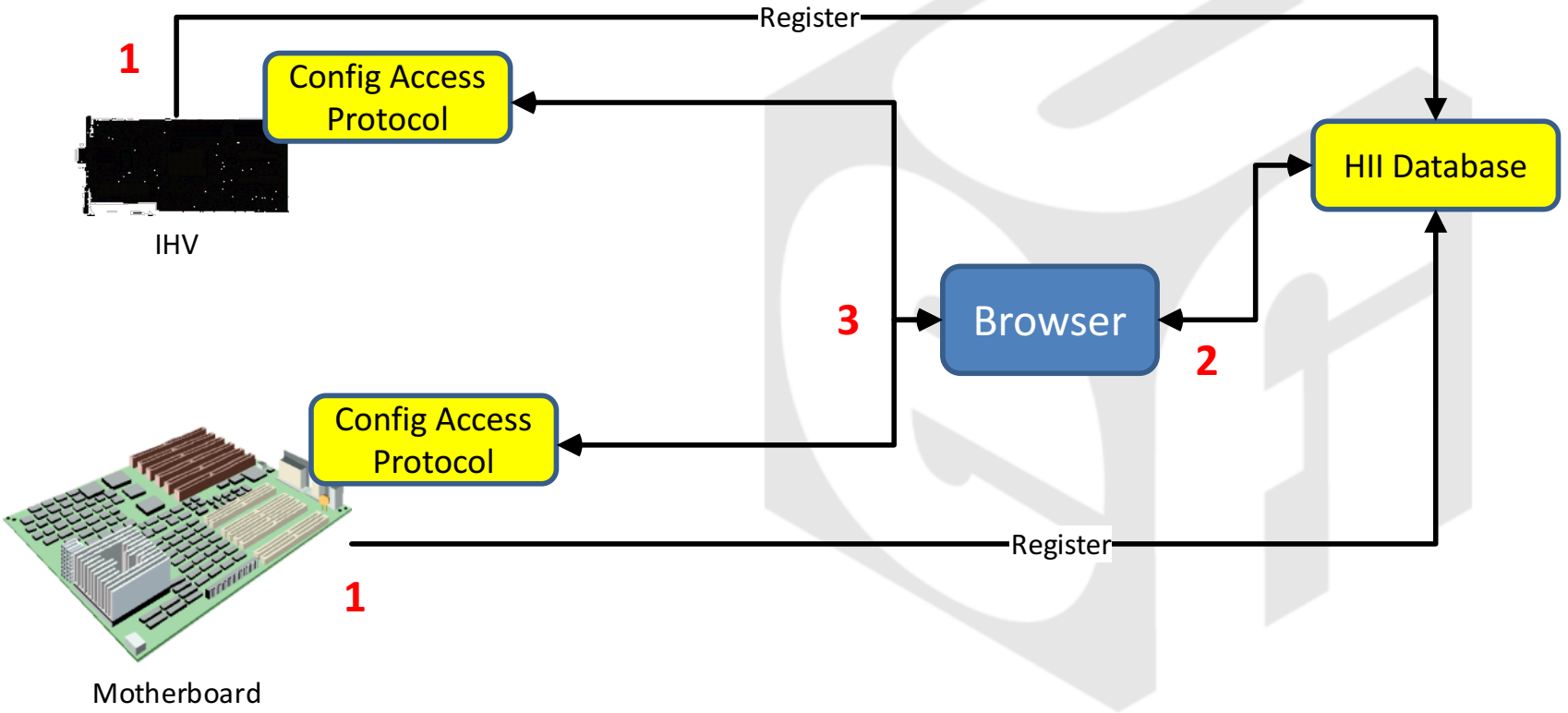
HII Data Flow

- Interfaces



HII Data Flow

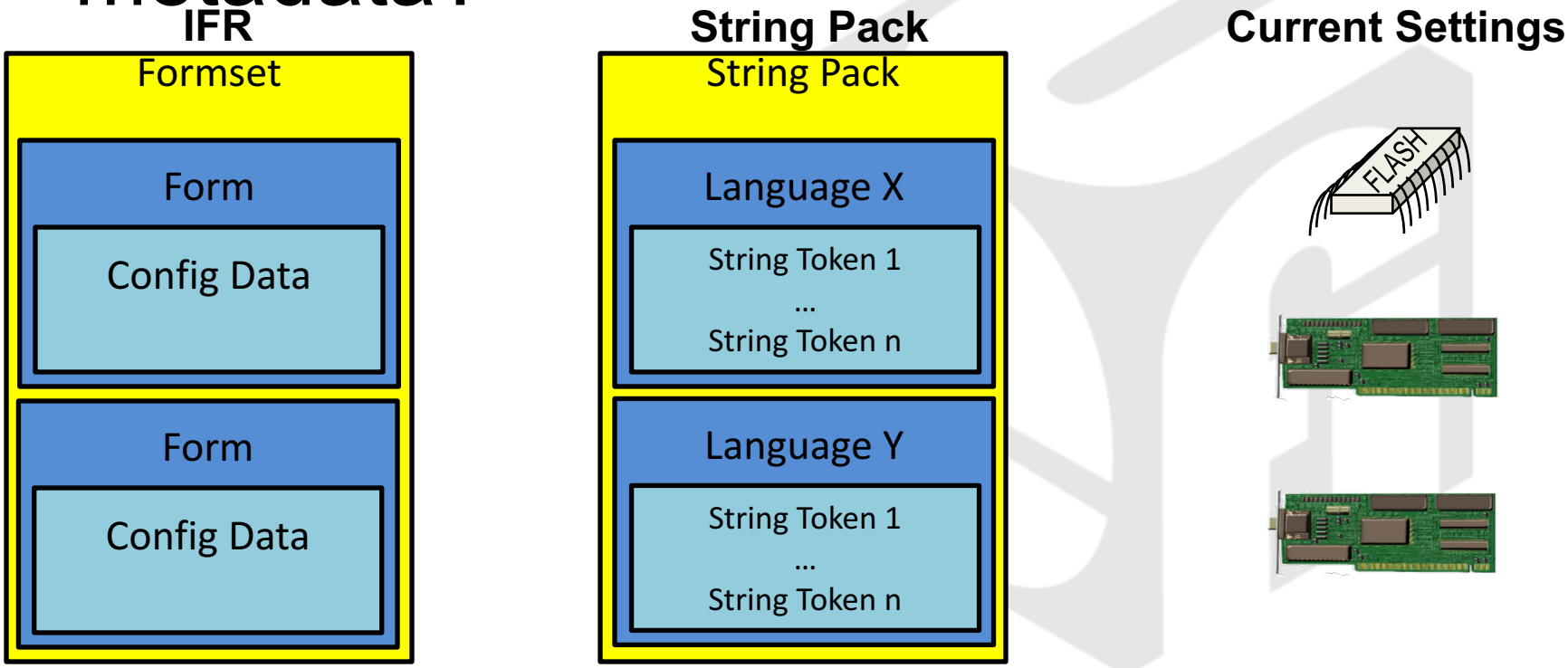
- Basic Software Interaction



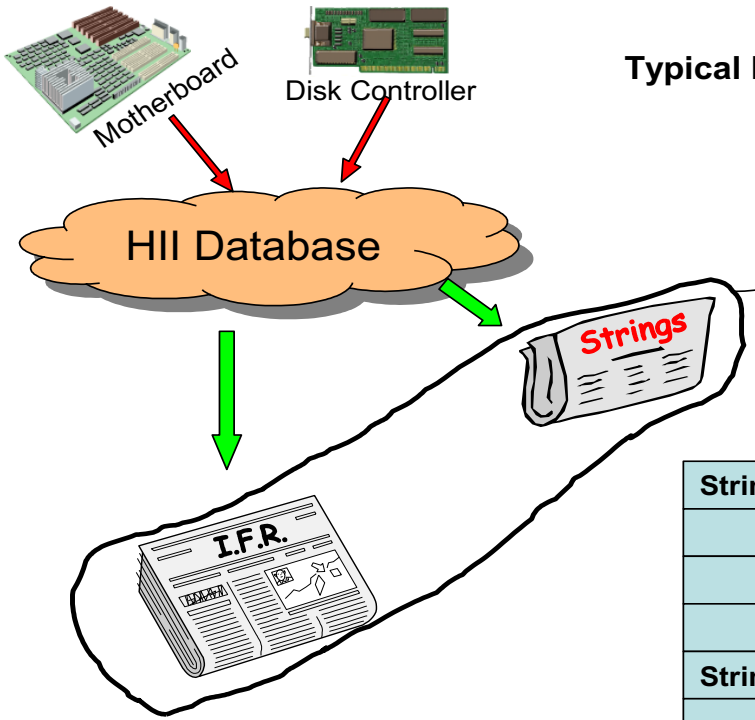
Overview of Configuration Metadata

Configuration Metadata

- What is UEFI's configuration metadata?



Configuration Metadata



Typical Data Contained in HII Database

	UINT 8	UINT 8	UINT 8	UINT 8
0	Op-Code	Length	Data Offset	
1	Prompt Token		Help Token	
2	Width	Flags		

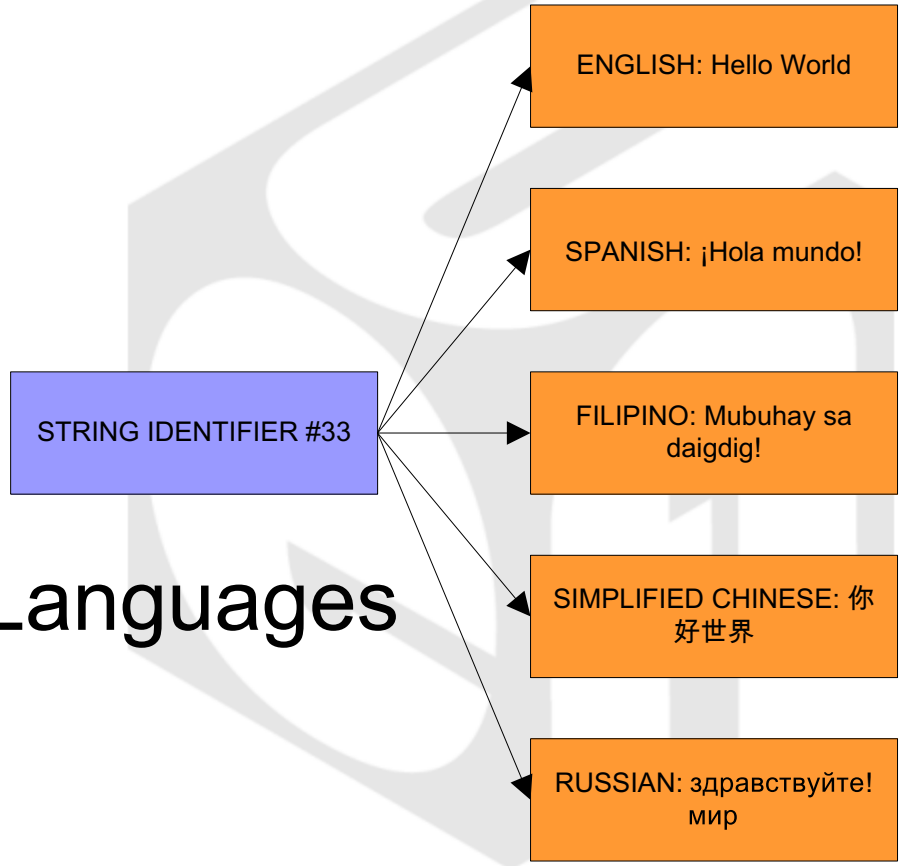
Sample IFR Op-code encoding

String Pack Language	ENG
String #1	“Hello World”
String #2	“Goodbye”
String #3	“Today”
String Pack Language	SPA
String #1	“Hola Mundo”
String #2	“Adios”
String #3	“Hoy”
String Pack Language	NULL

Sample String package encoding

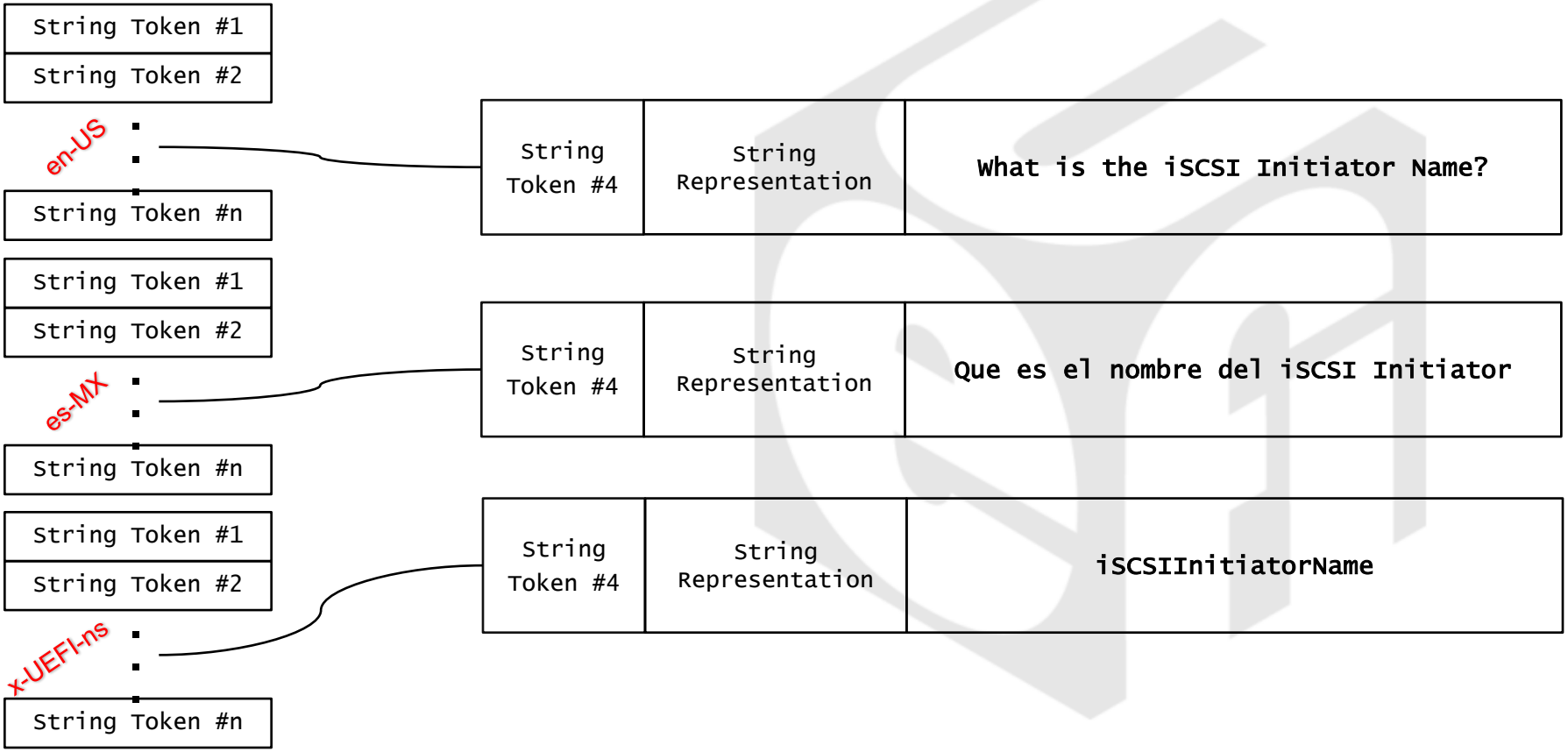
Configuration Metadata

- String
  - NewString
  - GetString
  - SetString
  - GetLanguages
  - GetSecondaryLanguages



Configuration Metadata

- A “platform language?”



Configuration Metadata

- Example Configuration Op-Code

```
#define EFI_IFR_NUMERIC_OP 0x07

typedef struct _EFI_IFR_NUMERIC {
  EFI_IFR_OP_HEADER          Header;
  EFI_IFR_QUESTION_HEADER    Question;
  UINT8                      Flags;
  UINT8                      MinValue;
  UINT8                      MaxValue;
  UINT8                      Step;
} EFI_IFR_NUMERIC;
```

String Token References

Where to retrieve/store data

Configuration Metadata

- IFR Question Header

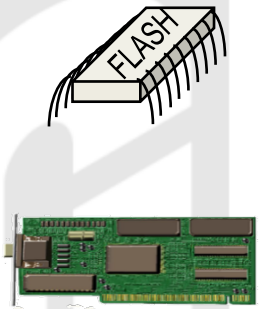
	Byte	Byte	Byte	Byte
Offset 0	Op-Code	Length	Prompt Token #	Help Token #
Offset 4	Question ID		VarStore ID	
Offset 8	VarStoreInfo		Flags	Op-Code Specific
Offset 12	Op-Code Specific	Op-Code Specific	Op-Code Specific	Op-Code Specific

Configuration Metadata

- VarStore Op-code

```
#define EFI_IFR_VARSTORE_OP 0x24

typedef struct {
  EFI_IFR_OP_HEADER       Header;
  EFI_GUID                Guid;
  EFI_VARSTORE_ID         VarStoreId;
  UINT16                  Size;
//UINT8                   Name[];
} EFI_IFR_VARSTORE;
```



Keywords and Namespaces

Keywords and Namespaces

- Example keywords

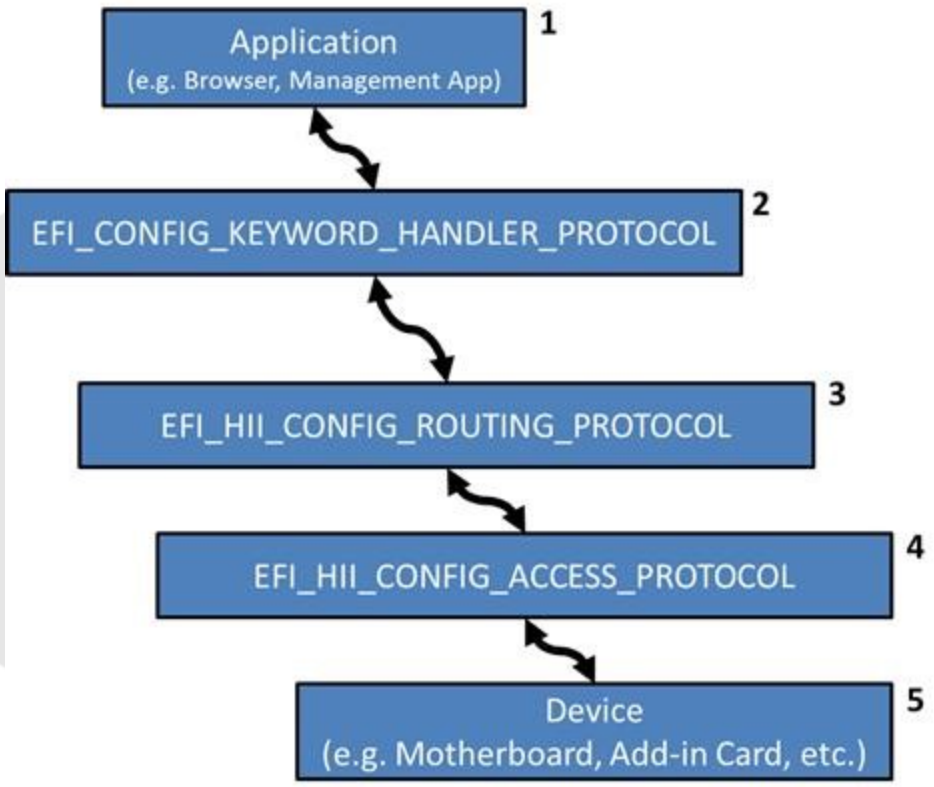
Keyword	DataType	UsageType	Description
iSCSIInitiatorName	Buffer	ReadWrite	The worldwide unique iSCSI Qualified Name (IQN) of this iSCSI Initiator. Only IQN format is accepted. EUI format is not supported. • CHAR8 InitiatorName[] Example: “iqn.1986-03.com.hp:init.sn-123456”
iSCSIAttemptName:#	Buffer	Read	Human readable descriptive name for this iSCSI boot attempt configuration • CHAR8 AttemptName[] • Example: “Attempt 1”
iSCSIBootEnable:#	Numeric:1	ReadWrite	Enables or Disables iSCSI Boot for a selected iSCSI boot attempt • 0 – iSCSI Boot Disabled • 1 – iSCSI Boot Enabled • 2 – iSCSI Boot Enabled for MPIO (Multi-path IO)

Keywords and Namespaces

- Namespaces
  - “x-UEFI-ns” is the UEFI Platform Configuration language.
  - “x-UEFI-*CompanyName*” not covered by the standard and intended to be used as an extensibility language for OEMs/IHVs.
- Configuration Strings
  - Think of this as a specially formatted string that forms the basis of how data is transported between APIs.
  - For example:
    - x-UEFI-ns&**PATH**=987654321ABCDEF&**KEYWORD**=iSCSIBootEnable:1&**VALUE**=1

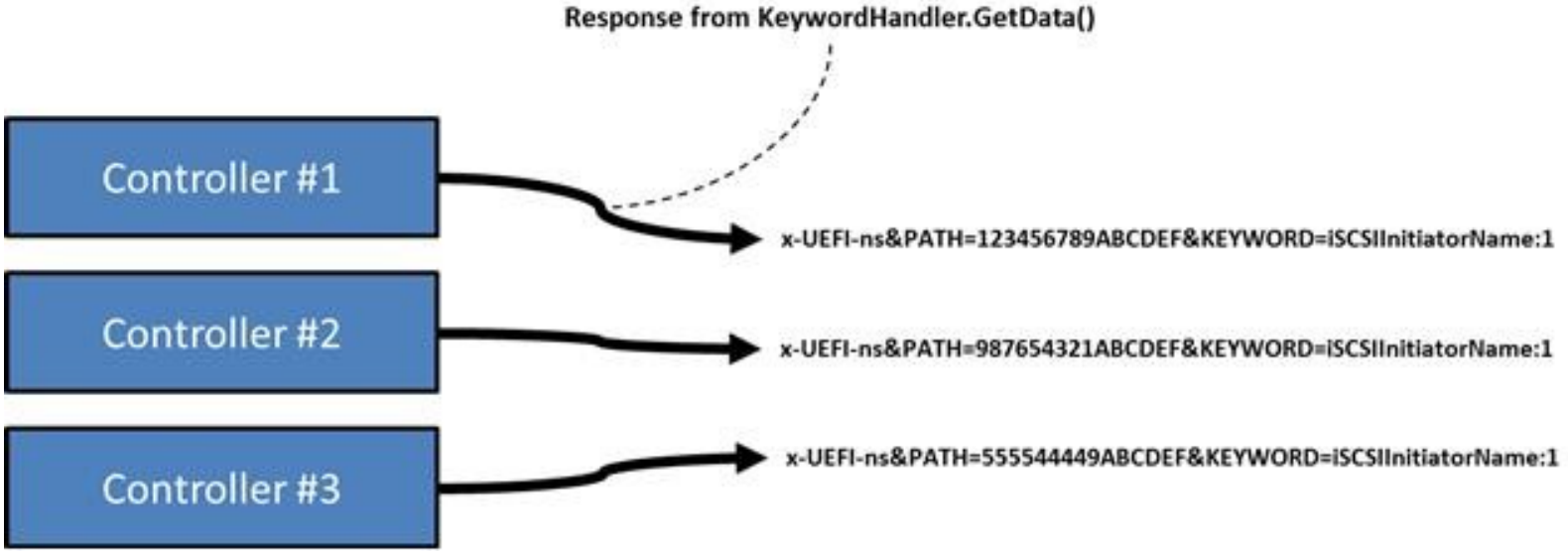
Keywords and Namespaces

1) Any application which wants to get or set any of the values abstracted by a keyword can interact with the API's that are defined within the UEFI specification. It would be the responsibility of this application to construct and interpret keyword strings that are passed or returned from the API's.
2) An agent within the system will expose the EFI_CONFIG_KEYWORD_HANDLER_PROTOCOL interface with its GetData() and SetData() functions. These services will interact both with the application that called it and the underlying routing routines within the system.
3) The EFI_HII_CONFIG_ROUTING_PROTOCOL is intended to act as a mechanism by configuration reading or writing directives are proxied to and from the appropriate underlying device(s) that have exposed configuration access abstractions.
4) Configurable items in the platform will expose an EFI_HII_CONFIG_ACCESS_PROTOCOL interface that allows the setting or retrieving of configuration data.
5) The component in the platform which has exposed configuration access abstractions.



Keywords and Namespaces

- Multiple Instances
  - Configurable agents may expose multiple instances of the same keyword, or there might be multiple agents exposing the same keyword.
  - How to differentiate?



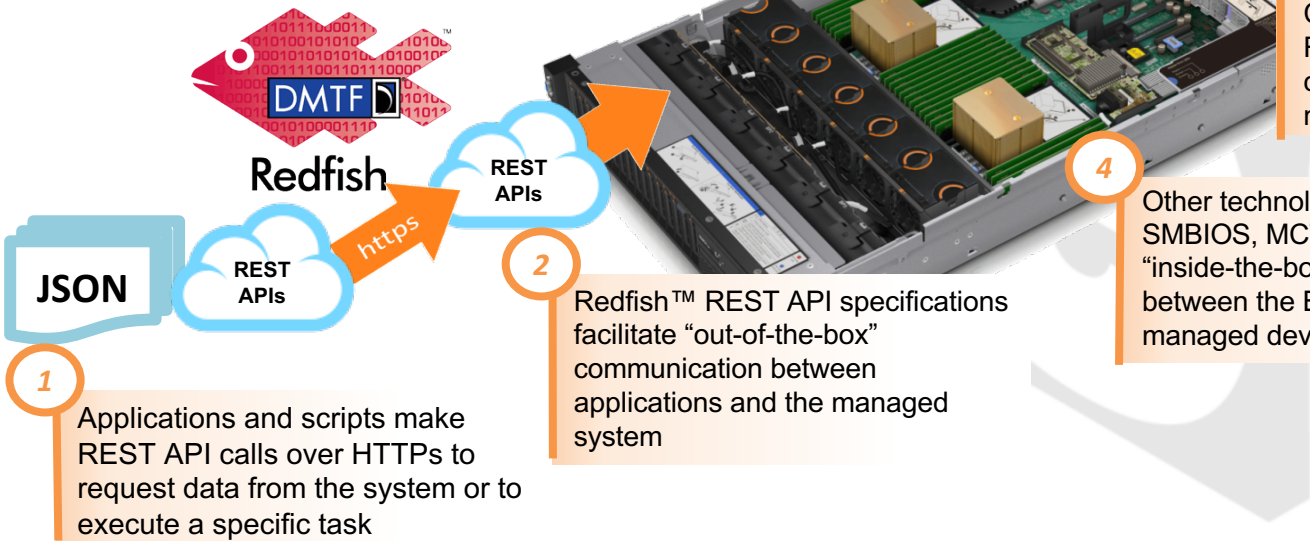
Keywords and Namespaces

- **Summary steps**
  1. Collect a list of all of the HII handles maintained by the HII database.
  2. For each of the registered HII database entries, look to see if any strings are registered within the x-UEFI-ns language name.
     a) If so, look for a string match of “iSCSIInitiatorName” in any of the strings for a particular HII handle
        i. If none are found, go to the next HII handle and execute 2a again.
        ii. If there are no more HII handles, then this platform doesn’t currently expose “iSCSIInitiatorName” as a programmatically manageable object.
  3. If a match is found, then note the String Token value (e.g. 4).
  4. Proceed to search through that HII handle’s registered IFR forms for a configuration op-code that has a matching Prompt Token value (e.g. 4).
  5. Once discovered, the configuration op-code contains all of the information needed to understand where that iSCSI Initiator Name information is stored.
     a) This allows a program to optionally extract the current settings as well as optionally set the current settings.

Redfish™ Overview

What is Redfish™?

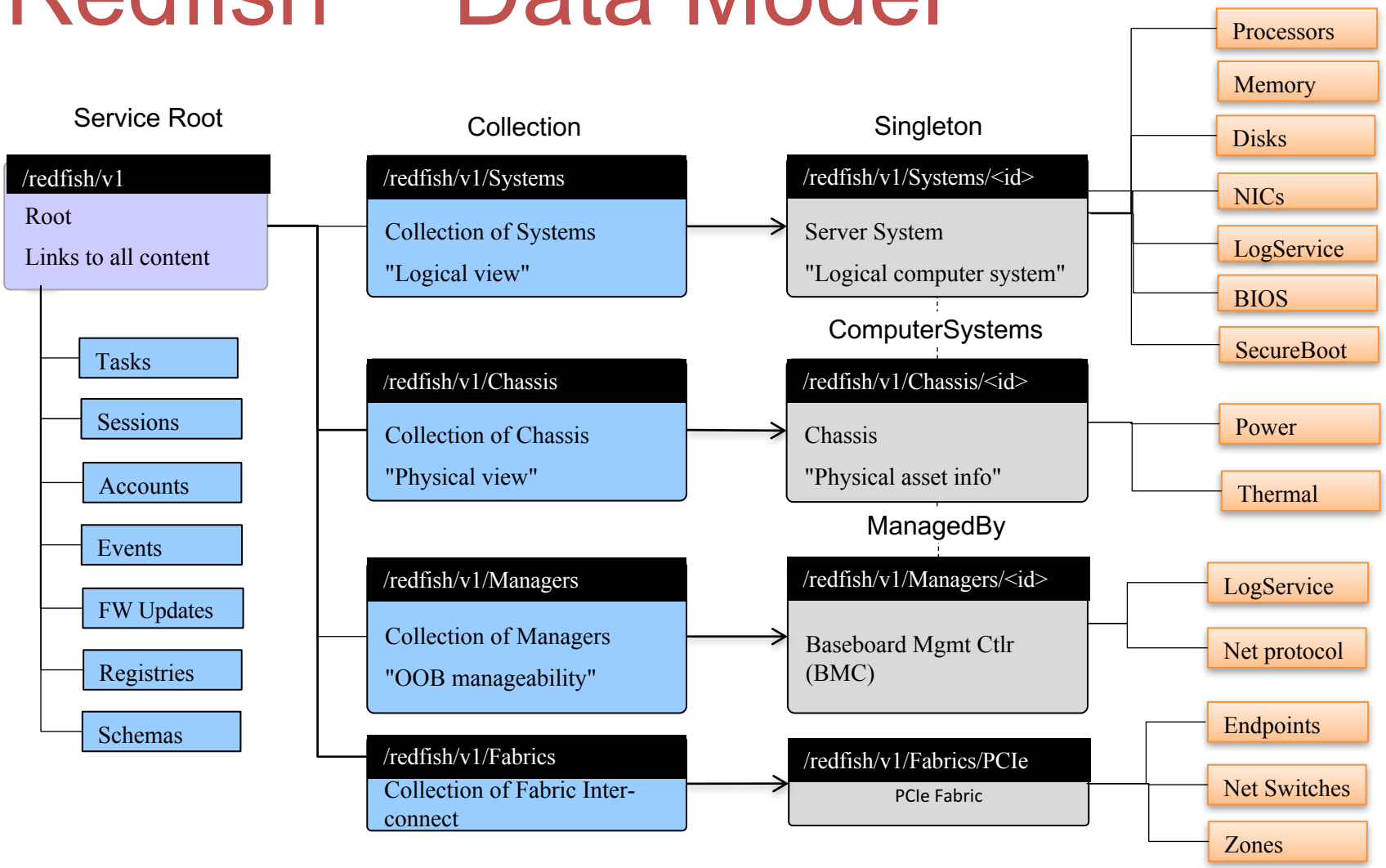
- **A DMTF industry standard**
  - RESTful interface for managing IT Infrastructure
  - Built on modern tool-chain (HTTPs/TLS, REST, JSON, OData)
  - Schema-backed, human readable output



What is REST and JSON?

- **REST = RE**presentational **S**tate **T**ransfer
  - Verbs: HTTP GET, POST, PUT, PATCH, HEAD and DELETE
  - Nouns: Resources uniquely identified by URIs
- **JSON = J**ava **S**cript **O**bject **N**otation
  - Lightweight human readable data-interchange format
  - Name/Value pairs

Redfish™ Data Model



Navigating Redfish™ Data

HTTP GET @ https://<ip_add>/redfish/v1/


```
{
  "@odata.id": "/redfish/v1/",
  "@odata.type": "#ServiceRoot.1.0.0.ServiceRoot",
  "@odata.context": "/redfish/v1/$metadata#ServiceRoot",
  "RedfishVersion": "1.0.0",
  "UUID": "00000000-0000-0000-0005-000000000001",
  "Chassis": {
    "@odata.id": "/redfish/v1/Chassis/",
  },
  "Tasks": {
    "@odata.id": "/redfish/v1/Tasks"
  },
  "Managers": {
    "@odata.id": "/redfish/v1/Managers/",
  },
  "Systems": {
    "@odata.id": "/redfish/v1/Systems/"
  },
  "SessionService": {
    "@odata.id": "/redfish/v1/SessionService/",
  },
  "Registries": {
    "@odata.id": "/redfish/v1/Registries/"
  },
  "JsonSchemas": {
    "@odata.id": "/redfish/v1/JsonSchemas/"
  }
}
```


Unique URI for each resource

Schema Type and Version

OData CSDL Namespace (XML Schema)

Properties

HTTP GET/PATCH (read/modify/write)
OR
HTTP GET (read only)

Links
Follow the HTTP URI to other linked resources (Tree crawling)

HII IFR to Attribute Registry

HII IFR	Redfish Attribute Registry
Question Prompt String (x-UEFI)	AttributeName
Question Prompt String (selected lang)	DisplayName
Question Help String	HelpText
EFI_IFR_WARNING_IF text	WarningText
• EFI_IFR_ONE_OF_OP • EFI_IFR_STRING_OP or EFI_IFR_TEXT_OP or EFI_IFR_DATE_OP or EFI_IFR_TIME_OP • EFI_IFR_NUMERIC_OP • EFI_IFR_CHECKBOX_OP • EFI_IFR_PASSWORD_OP	Type • Enumeration • String • Integer • Boolean • Pasword
EFI_IFR_LOCKED_OP EFI_IFR_GRAY_OUT_IF_OP EFI_IFR_DISABLE_IF_OP EFI_IFR_FLAG_READ_ONLY	ReadOnly GrayOut
EFI_IFR_SUPPRESS_IF_OP	Hidden

HII IFR to Attribute Registry

HII IFR	Redfish Attribute Registry
EFI_IFR_ONE_OF_OPTION_OP Prompt in x-UEFI	ValueName
EFI_IFR_ONE_OF_OPTION_OP Prompt in Lang	ValueDisplayName
Order relative to other Questions in the Form Pkg	DisplayOrder
Hierarchy of nest Form Titles in x-UEFI	MenuPath
EFI_IFR_PASSWORD_OP or EFI_IFR_STRING_OP MaxSize, MinSize	MaxLength, MinLength
EFI_IFR_NUMERIC MaxValue, MinValue, Step	UpperBound, LowerBound, ScalarIncrement
EFI_IFR_MATCH2 RegEx pattern	ValueExpression
HII Form Title in x-UEFI	MenuName
EFI_IFR_DEFAULT_OP EFI_IFR_DEFAULT Question specific default value Flag	DefaultValue

HII IFR to Attribute Registry

HII IFR	Redfish Attribute Registry
• EFI_IFR_EQUAL_OP or EFI_IFR_EQ_ID_VAL_OP or EFI_IFR_EQ_ID_ID_OP • EFI_IFR_NOT_EQUAL_OP • EFI_IFR_GREATER_THAN_OP • EFI_IFR_GREATER_EQUAL_OP • EFI_IFR_LESS_THAN_OP • EFI_IFR_LESS_EQUAL_OP	MapFromCondition: • EQU • NEQ • GTR • GEQ • LSS • LEQ
• EFI_IFR_AND_OP • EFI_IFR_OR_OP	MapTerms: • AND • OR
EFI_IFR_MAP_FORM_OP EFI_IFR_MAP_OP EFI_IFR_READ_OP EFI_IFR_WRITE	CurrentValue

HII IFR to Attribute Registry

HII IFR	Redfish Attribute Registry
Do not exist in HII • Require vendor IFR extensions • Or extending the UEFI Specification	• Immutable • WriteOnly • IsSystemUniquePropert

HII IFR to Attribute Registry

HII IFR	Redfish Attribute Registry
• SUBTITLE, IMAGE, ANIMATION, ACTION • REF / REF1/REF2/REF3 • NO_SUBMIT_IF , INCONSISTENT_IF • EFI_IFR_EQ_ID_VAL_LIST_OP • NOT, RULE • REFRESH • TO_LOWER, TO_UPPER • ORDERED_LIST • BITWISE_AND, OR, NOT • SHIFT_LEFT, SHIFT_RIGHT • ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULO • MID, FIND, TOKEN, SPAN, DUP, CATENATE	Do not exist in Redfish Attribute Registry • Require OEM specific extensions • Or Extending the Redfish Schema