presented by

Redfish UEFI and EDK2 Implementation

UEFI 2025 Developers Conference & Plugfest

October 10, 2025

Abner Chang and Igor Kulchytskyy

Meet the Presenters

Abner Chang

PMTS - AMD

Abner Chang, was a game developer before entering BIOS industry. About 25 years host firmware experience in the industry. 13-year in Phoenix, 8-year in HPE and 3-year until now in AMD as the AMD opensource host firmware and SoC feature architect. Industry standard contributor to UEFI/PI/SMBIOS/Gen-Z/Redfish/RISC-V Platform/PLIC, etc. Delivered RISC-V EDK2 port to industry in 2016 and was the vice chair of RISC-V Configuration Structure Task Force.

Meet the Presenters

Igor Kulchytskyy

PRMGENG-AMI

Igor Kulchytskyy worked for the company which manufactured e-Ink devices (Head of Firmware\Software R&D department) before entering BIOS industry. About 15 years host firmware experience in the industry. 15-year in AMI. Industry standard contributor to UEFI/ Redfish etc.

Contributors (List in the alphabet order)

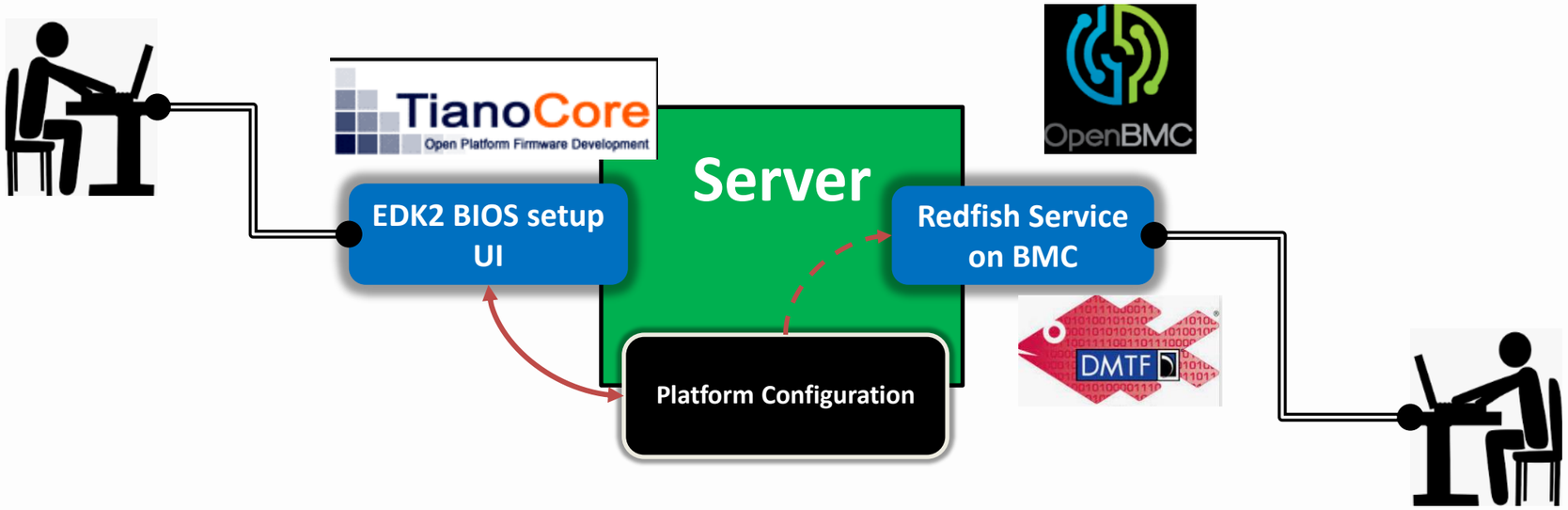
- **AMD**: Abner Chang
- **AMI**: Igor Kulchytskyy
- **Nvidia**: Nickle Wang
- **Special thanks to**:
  - **Qualcomm**: Aman Chhetry
  - Mike Maslenki

Agenda

- Where was Redfish UEFI in 2019?
- Current Redfish Firmware Solution
- Current Implementation of UEFI Redfish Specification
- UEFI Redfish Foundation on EDK2
- EDK2 Redfish Client Support Based on UEFI Redfish Foundation
- Advanced – Redfish Device Enablement
- Call to Action!

What is Redfish for the Host Firmware?

- Out-of-band server configuration management
- Server hardware inventory management



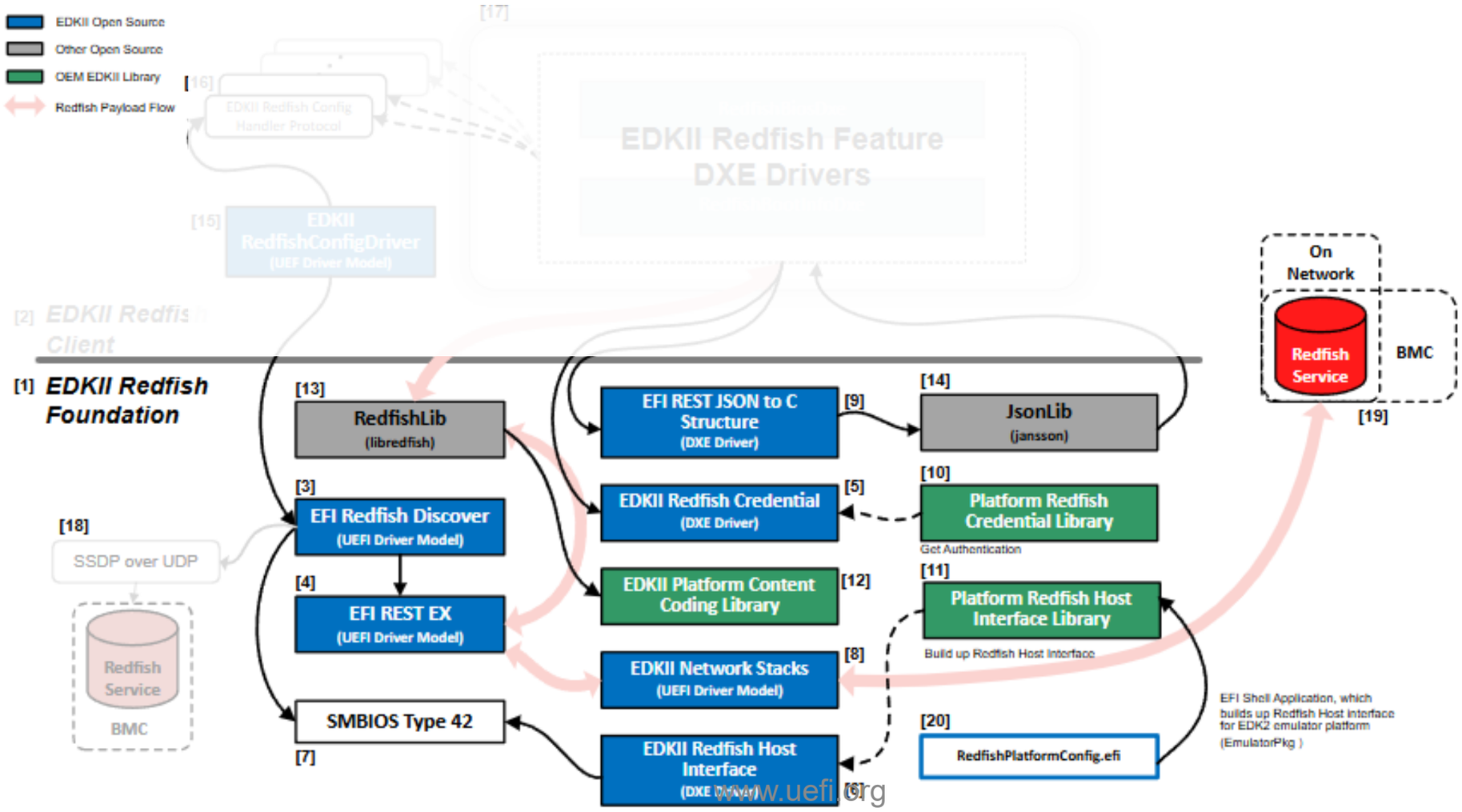
Where was Redfish UEFI in 2019?

- UEFI specification
  - REST EX Protocol
  - EFI Redfish Discovery Protocol
  - EFI Redfish JSON Structure Protocol
- Implementation
  - EDK2 staging as the PoC for the above protocols

Current Redfish Firmware Solution (Post 2019)

The Redfish Foundation - edk2/RedfishPkg

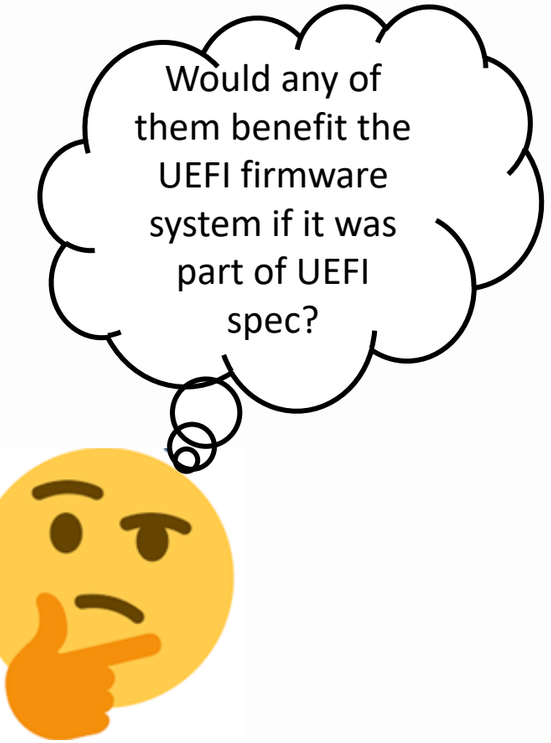
Design specification:
https://github.com/tianocore/edk2/tree/master/RedfishPkg



Current Redfish Firmware Solution

(continued)

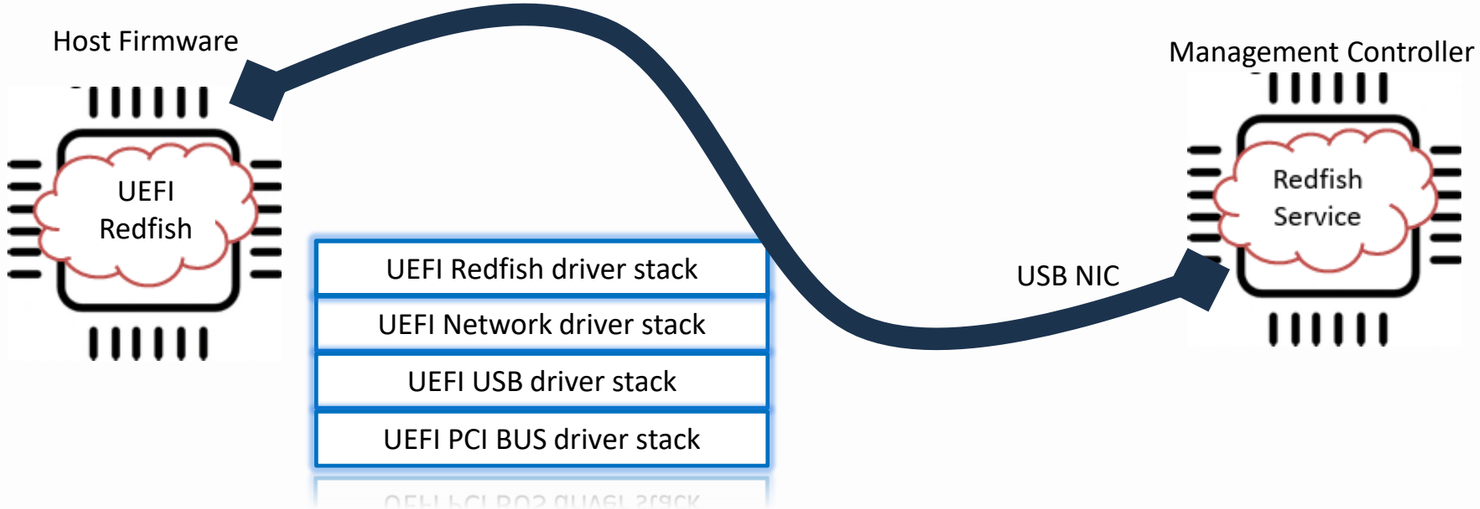
- Implementations of EFI Redfish Protocols
  - EFI Redfish Discovery Protocol
  - EFI REST EX Protocol for Redfish
  - EFI REST JSON Structure Protocol
- EDK2 Implementations
  - EDK2 Redfish Config Handler Protocol
  - EDK2 Redfish HTTP Protocol
  - EDK2 Redfish Platform Config Protocol
  - EDK2 Redfish Credential Protocol
  - EDK2 Redfish Host Interface Driver



Current Redfish Firmware Solution (continued)

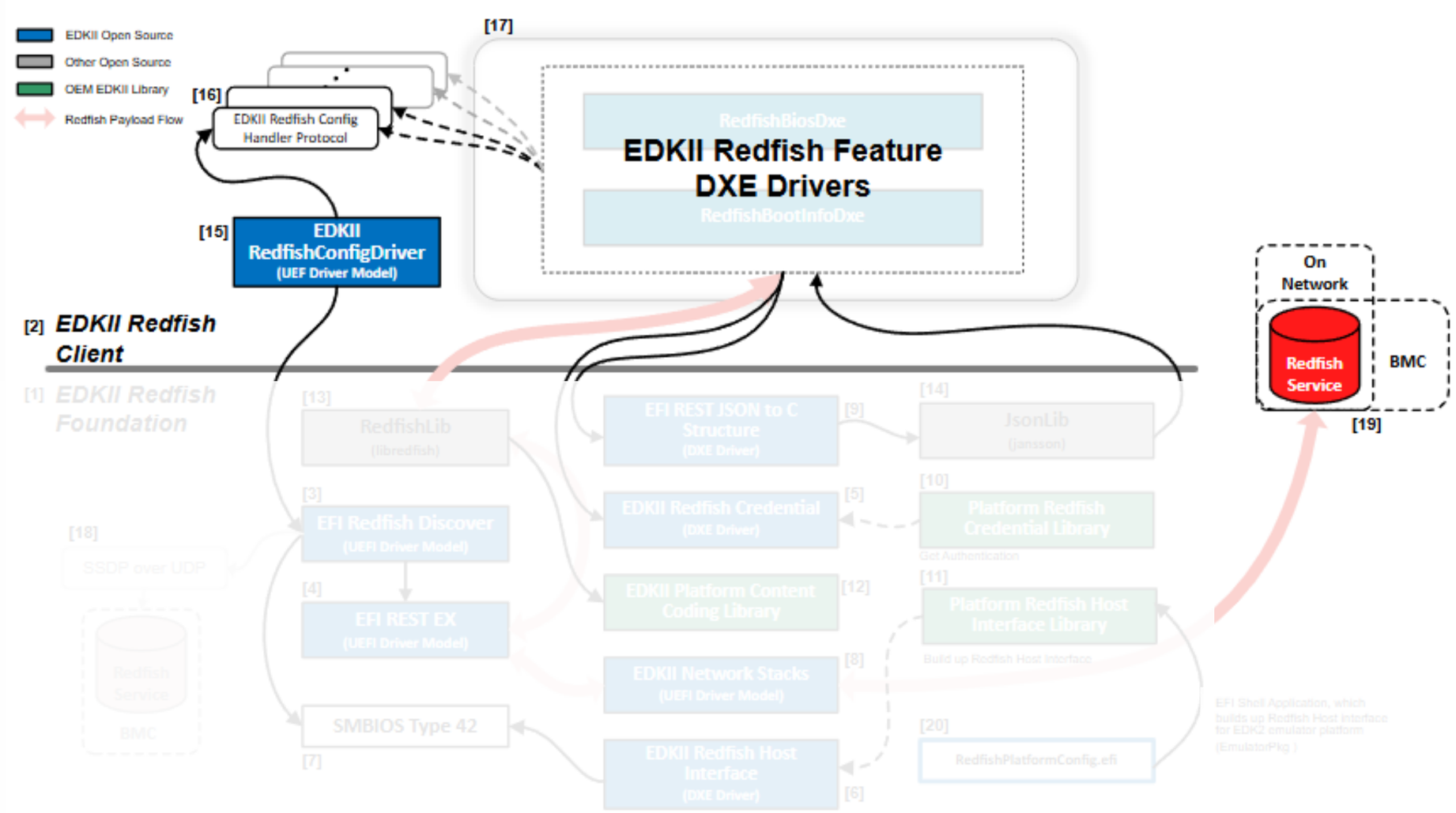
USB Network Interface

- AMI contributed the EDK2 USB NIC driver for USB Communication Device Class. The driver supports Ethernet Control Model, Network Control Model and Remote Network Driver Interface Specification interface.
- This EDK2 driver enables the USB NIC device exposed by Baseboard Management Control (BMC) to the host FW, which is the interface between host firmware and baseboard management controller
- EDK2 network stack is enabled on the USB NIC, that says host firmware can HTTP(s) to Redfish service hosted by baseboard management controller



Current Redfish Firmware Solution (continued)

The Redfish Client - tianocore EDK2-redfish-client repository



Current Redfish Firmware Solution (continued)

The Redfish Client - tianocore EDK2-redfish-client repository

- EDK2 Redfish ConfigLang Map Protocol
- EDK2 Redfish ETag Protocol
- EDK2 Redfish Feature Protocol
- EDK2 Redfish Resource Config Protocol

Current Redfish Firmware Solution (continued)

The Redfish Client - tianocore EDK2-redfish-client repository

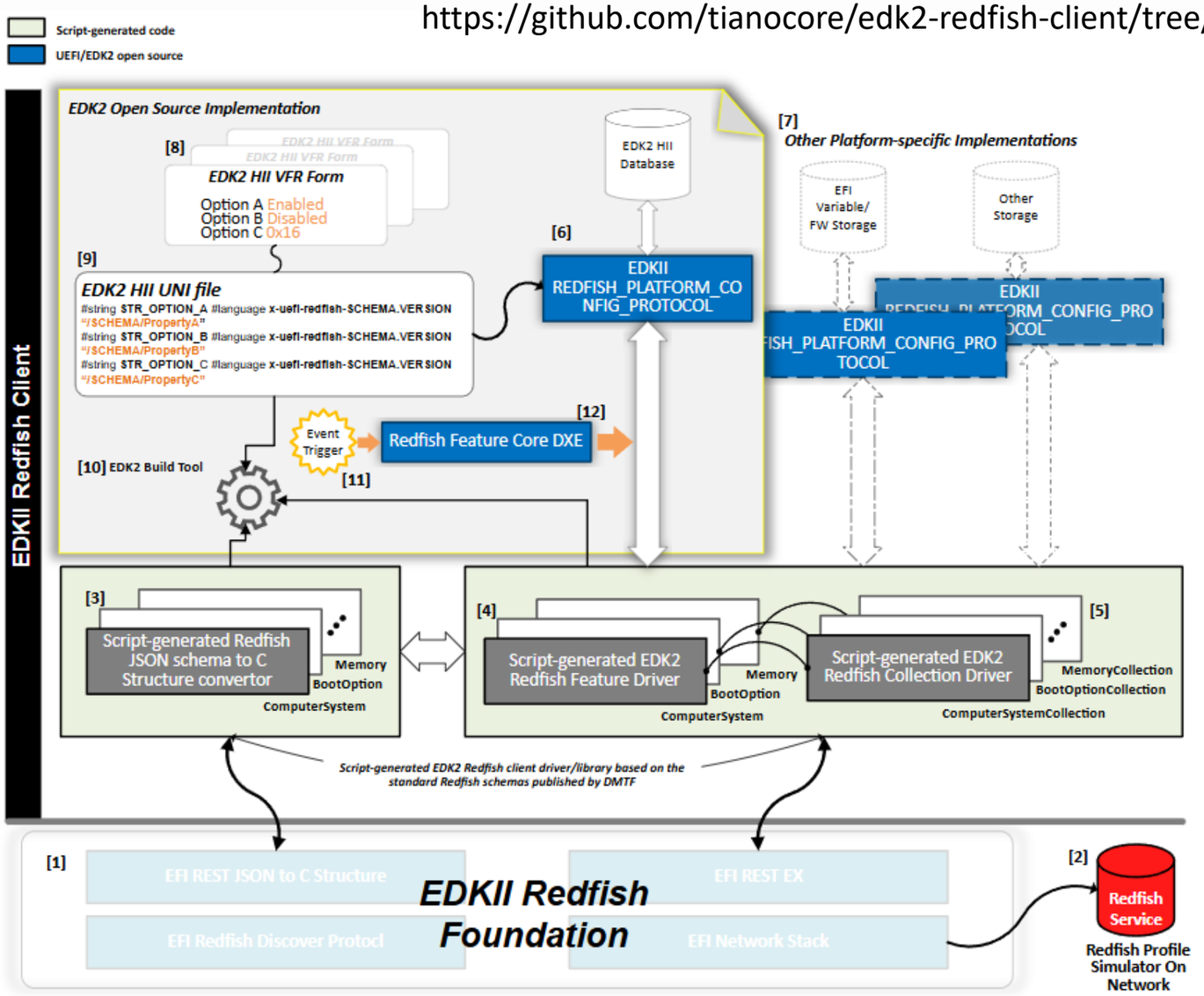
What does RedfishClientPkg provide?

- Out of band Platform Configuration Management
- Inventory Provisioning

Redfish Platform Configuration Management

tianocore EDK2-redfish-client repository

Design specification:
https://github.com/tianocore/edk2-redfish-client/tree/main/RedfishClientPkg



EDK2 Redfish Feature Drivers

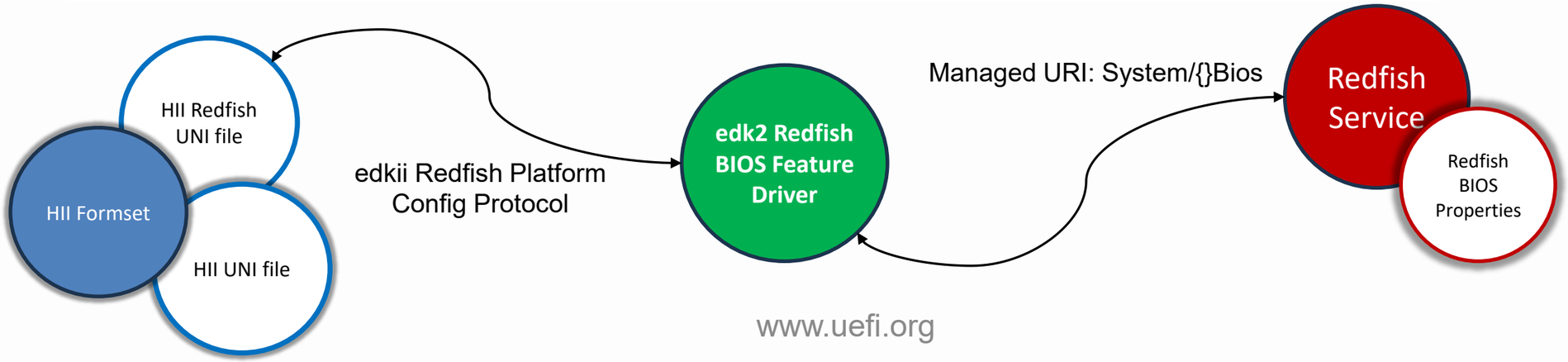
- Redfish Platform Configuration Management
  - BIOS — Systems/{}/Bios
  - BiosAttibuteRegistry (Not upstream yet) — Systems/{}/Bios/AttributeRegistry
  - BootOption — Systems/{}/BootOptions/{}/BootOption
  - BootOptionCollection — Systems/{}/BootOptions
  - ComputerSystem — Systems/{}/ComputerSystem
  - ComputerSystemCollection — Systems
- Redfish Inventory Feature Drivers (Not upstream yet)
  - Drive — Systems/{}/Storage/{}/Storage
  - DriveArray — Systems/{}/Storage
  - Memory — Systems/{}/Memory/{}/Memory
  - MemoryCollection — Systems/{}/Memory
  - PCIeDevice — Systems/{}/PCIeDevices/[]/PCIeDevice
  - PCIeDeviceCollection — Systems/{}/PCIeDevices

Redfish Platform Configuration Management

- The x-uefi keyword mapping
  https://uefi.org/configNamespace
- x-uefi-redfish is the HII language used for the HII option to Redfish property mapping. e.g., X-uefi-redfish-Bios.v1_0_9 is the Redfish BIOS schema HII language.

X-uefi-redfish-<SchemaName>.<SchemaVersion>

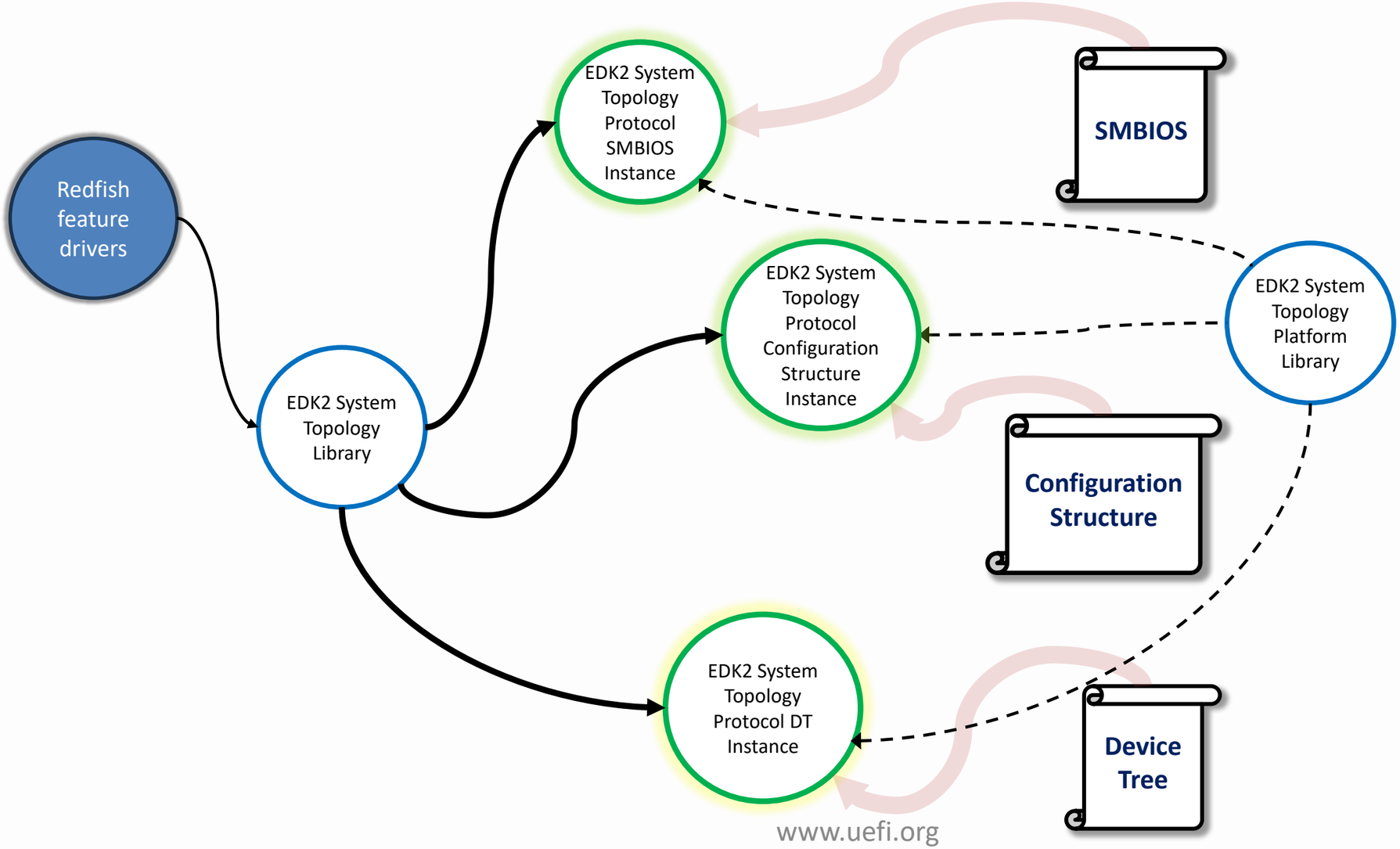
```
#langdef   x-UEFI-redfish-Bios.v1_0_9 "Bios.v1_0_9"

#string STR_BIOS_OPTION_1_PROMPT       #language x-UEFI-redfish-Bios.v1_0_9   "/Bios/Attributes/BiosOption1"
#string STR_BIOS_OPTION_2_PROMPT       #language x-UEFI-redfish-Bios.v1_0_9   "/Bios/Attributes/BiosOption2"
#string STR_BIOS_OPTION_3_PROMPT       #language x-UEFI-redfish-Bios.v1_0_9   "/Bios/Attributes/BiosOption3"
#string STR_BIOS_OPTION_4_PROMPT       #language x-UEFI-redfish-Bios.v1_0_9   "/Bios/Attributes/BiosOption4"

#string STR_BIOS_OPTION_1_ITEM_1       #language x-UEFI-redfish-Bios.v1_0_9   "Item #1"
#string STR_BIOS_OPTION_1_ITEM_2       #language x-UEFI-redfish-Bios.v1_0_9   "Item #2"
#string STR_BIOS_OPTION_1_ITEM_3       #language x-UEFI-redfish-Bios.v1_0_9   "Item #3"
```



Redfish Inventory Provisioning: EDK2 Redfish System Topology Protocol

- Why introduce EDK2 Redfish System Topology Protocol
  - SMBIOS, DeviceTree or Configuration Structure (OCP UPCI$_{\text{under development}}$) describe the hardware configuration of the platform. However, not all the Redfish properties have the corresponding definitions defined in the above spec
  - EDK2 Redfish System Topology Protocol abstracts the above hardware configuration structure.
  - This protocol abstracts the interface and method to retrieve the hardware configuration on the platform. For those Redfish property corresponding hardware configurations are not available in SMBIOS or DT, platform can still provide the implementation to retrieve the hardware configuration
  - It provides the flexibility to platform for the Redfish inventory provisioning
  - EDK2 Redfish feature driver can use this protocol to get the certain hardware configurations, such as PCIe® technology, Processor, memory, etc. and then fill up the information in the Redfish JSON resource. HTTP PUT or POST to the target Redfish resource URI for the inventory provisioning
  - Reduce the dependency with the specifications and increase the flexibility of platform implementation

Redfish Inventory Provisioning: EDK2 (UEFI)
Redfish System Topology Driver Stack



Redfish Inventory Provisioning: EDK2 Redfish System Topology Protocol

```
struct _EFI_SYS_TOPOLOGY_PROTOCOL {
   UINT16                                          Version;
   EFI_SYS_TOPOLOGY_PROTOCOL_CREATE_REPORT         CreateReport;
   EFI_SYS_TOPOLOGY_PROTOCOL_FREE_REPORT           FreeReport;
   EFI_SYS_TOPOLOGY_PROTOCOL_COLLECT_PCIE_DEVICES  CollectPcieDevices;
   EFI_SYS_TOPOLOGY_PROTOCOL_COLLECT_STORAGE_DEVICES CollectStorageDevices;
   EFI_SYS_TOPOLOGY_PROTOCOL_COLLECT_CXL_DEVICES   CollectCxlDevices;
   EFI_SYS_TOPOLOGY_PROTOCOL_COLLECT_MEMORY_DEVICES CollectMemoryDevices;
   EFI_SYS_TOPOLOGY_PROTOCOL_COLLECT_BIOS_Version  CollectBiosVersion;
};
```

Redfish Inventory Provisioning: EDK2 Redfish System Topology Library

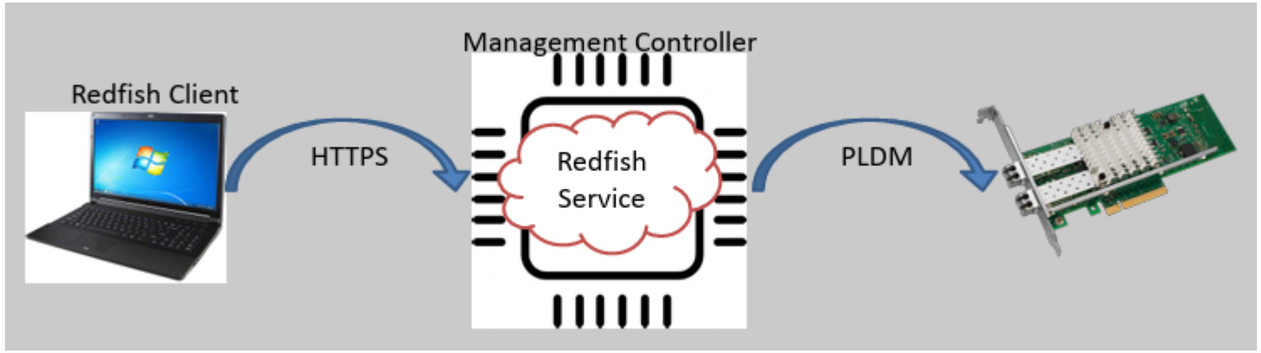
The library interface of EDK2 Redfish System Topology library is simple

```
/**  Get the count of specific REDFISH_SYSTEM_TOPOLOGY_TYPE.
 @param[in]    TopologyType    REDFISH_SYSTEM_TOPOLOGY_TYPE.
 @param[out]   Count           Count of this REDFISH_SYSTEM_TOPOLOGY_TYPE.
 @retval       EFI_STATUS      Some error happened.
**/
EFI_STATUS
RedfishSystemTopologyGetCount (
  IN     REDFISH_SYSTEM_TOPOLOGY_TYPE  TopologyType,
  OUT  UINTN                                    *Count
);

/**  Get the count of specific REDFISH_SYSTEM_TOPOLOGY_TYPE.
@param[in]    TopologyType    REDFISH_SYSTEM_TOPOLOGY_TYPE.
@param[out]   DeviceEntry     NULL to get the first entry of REDFISH_SYSTEM_TOPOLOGY_TYPE device.
                              Otherwise, returns the next entry of REDFISH_SYSTEM_TOPOLOGY_TYPE device.
@retval EFI_STATUS            Some error happened.
**/
EFI_STATUS
RedfishGetSystemTopologyGetEntry (
   IN  REDFISH_SYSTEM_TOPOLOGY_TYPE             TopologyType,
   IN  EDKII_REDFISH_SYSTEM_TOPOLOGY_DEVICE   **DeviceEntry
);
```

Redfish Device Enablement

The EDK2 Redfish feature drivers

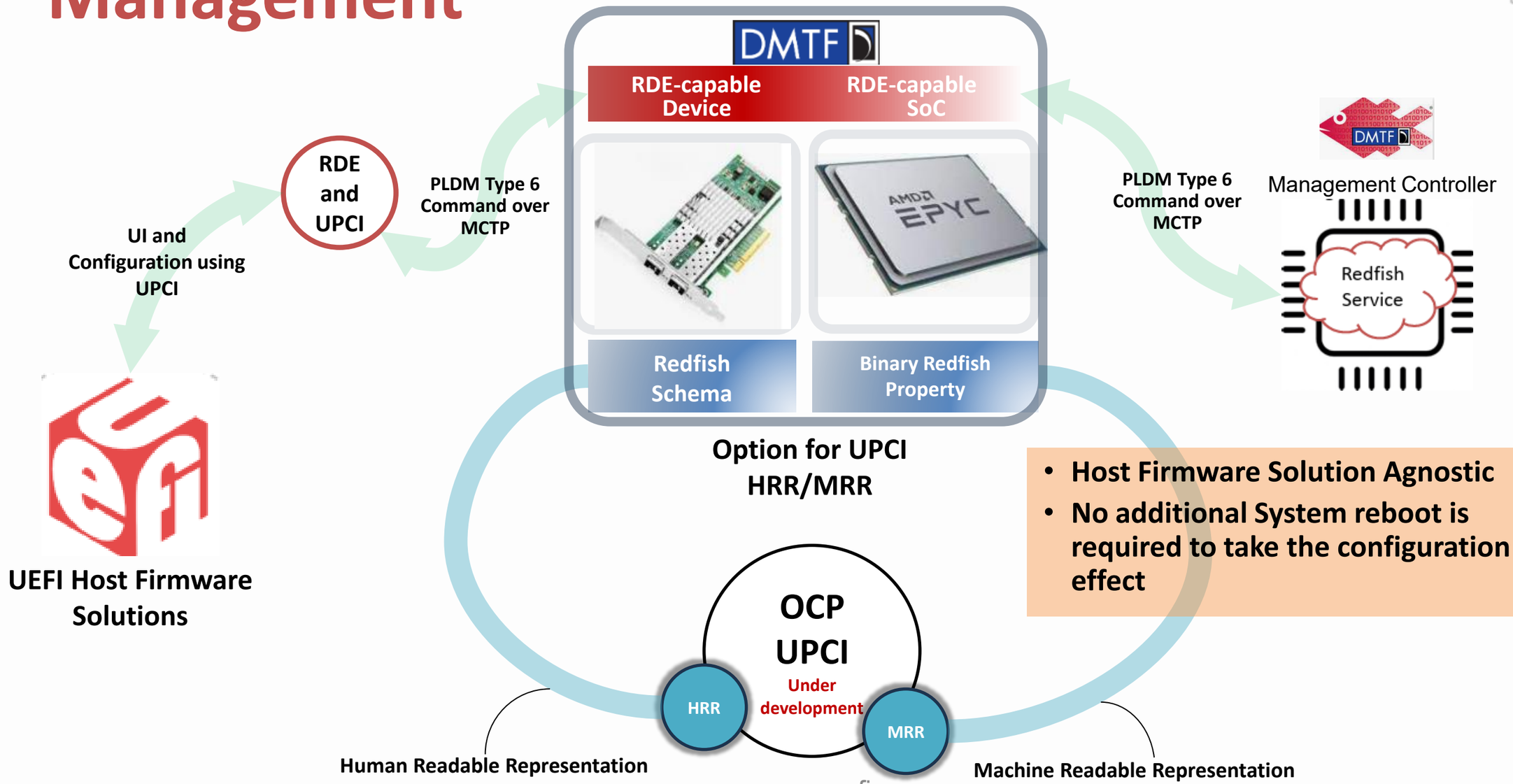
Out-of-band user manage/configure the PCIe® option card **without** host firmware involved



- **JSON format as the human readable configuration representation**
- **Binary encoded JSON as the machine-readable representation**

- OCP (Open Computing Project) is currently developing UPCI spec to define the Unified platform Configuration Format, accompanying with the unified configuration Interface (openSFI)
- Hardware Review Requirements (HRR) – Not determined yet
- Maximum Read Request (MRR) – Not determined yet

Redfish Device Enablement for the Server Management



- Host Firmware Solution Agnostic
- No additional System reboot is required to take the configuration effect

Call for Action

Evaluate the benefit to UEFI firmware system with below EDK2 protocol defined in UEFI spec:

- EDK2 Redfish Config Handler Protocol
- EDK2 Redfish HTTP Protocol
- EDK2 Redfish Platform Config Protocol
- EDK2 Redfish Credential Protocol
- EDK2 Redfish Host Interface Driver
- EDk2 Redfish System Topology Protocol

Connect UEFI HII with the Unified Platform Configuration Format

- Incorporate OCP UPCI with UEFI spec
- How does HII support OCP UPCI format?

Questions?