February 22, 2018

Out-Of-Band Management on UEFI System Firmware

Hui-Chin Cheng
Hewlett Packard Enterprise

Yun-Cheng Chang
Hewlett Packard Enterprise

Shang-Ching Hung
Hewlett Packard Enterprise

Out-Of-Band Management on UEFI System Firmware

1. Abstract and Introduction

The modern Redfish is a specification that utilize RESTful interface semantics to access data defined in model format to perform out-of-band (OOB) management through specific OOB software or hardware (such as Baseboard Management Controller, BMC). The OOB management allow users to configure system remotely when the system is in either power-off or power-on state. Industry can expect there are more and more pre-boot firmware drivers (like UEFI drivers) and system peripherals (such as PCI devices, PCI add-on-card and so on) support Redfish Schema/Configuration data model in the near future. This article describes the method to abstract the data communication/synchronization between UEFI drivers and OOB management on UEFI firmware environment. Furthermore, this article is not only restricted to single OOB management on system, the abstracts method described in this article is flexible and extensible to support multiple OOB management instances on one system simultaneously. Not only Redfish OOB management data model is supported, this article fulfills the requirements of any other data model of OOB managements such as OData XML/JSON data model, CIM-XML data model, 3rd party data model and etc.

2. The Firmware Driver Stacks of Out-Of-Band Management on UEFI System Firmware

Provide UEFI drivers having capabilities to manage the OOB manageable system firmware driver or system peripherals.

- UEFI OOB Management Storage Driver:
  This abstract level provides the mechanism to provision most of modern storage types (such as iSCSI/ FTP/ HTTP/ Physical USB/NAND devices) for storing OOB configurable properties.
  - With UEFI OOB Management Storage Driver installed on the UEFI environment, upstream UEFI drivers (such as UEFI OOB Management Driver) can directly pass configurable properties through UEFI OOB management storage driver without knowing how to configure and access to storage.
  - Store the OOB configurable properties into a variety of storages, such as BMC, non-volatile storage, network storage and etc. It's possible to have one or more than one UEFI OOB Management Storage Driver instances exist on one platform. OOB Configurable properties can be written to different storages through each UEFI OOB Management Storage Driver.
- UEFI OOB Management Driver:
  - This driver abstracts OOB configurable properties which is formatted in a variety of OOB management data models, such as Redfish data model, OData data model or vendor proprietary data model.
  - This driver writes the configurable properties to OOB management storage through UEFI OOB Management Storage Driver. This driver will locate all instances of UEFI OOB Management Storage Driver which installed on the UEFI environment and pass the configurable properties to each OOB Management Storage Driver instance.
  - Data synchronization: Handle the configurable property changes and responsible for updating the changes to upstream UEFI drivers if needed, and vice versa, this driver will also update the latest configurable property from

upstream UEFI driver to the OOB management storages. For example, when every time system is powered on. UEFI OOB Management Driver will check ETag (Like HTTP cache validation) of the configurations which stored in OOB management storage for each registered OOB manageable UEFI driver. This driver synchronizes the OOB-modified configurable properties to upstream UEFI driver or system peripherals.

3. Use Cases

Some use cases below are applied with the method mentioned in this article in reality. User can pre-configure the setting of PCI device through OOB management hardware frontend interface when platform is in power-off state. For example:

- Storage controller UEFI driver can register storage controller as an OOB manageable device through UEFI OOB Management Driver, after then, user can create a new RAID configuration via BMC restful API no matter system is in power-on or power-off state. The changes will take effect in the next boot.
- Pre-configure a bootable HTTP URL or PXE target for a network controller via OOB management frontend interface. System can boot to the desired network target in next boot. This instead of changing configurations in network controller option ROM or in-band configuration utility such as BIOS setup utility. The OOB manage UEFI drivers and system peripherals when platform is in power-off state can reduce system restart once.

UEFI driver with OOB configurable properties embedded in the UEFI image:

(Assume the implementation of UEFI driver image can accommodate OOB configurable properties in UEFI image's resource section).

If UEFI driver dispatcher recognizes OOB configurable property resource section in UEFI driver image, UEFI driver dispatcher can register this UEFI driver as an OOB manageable UEFI driver through UEFI OOB management. The data model and properties of UEFI driver is stored/retrieved through UEFI OOB Management Storage Driver. This is the on-demand UEFI driver OOB management.

PCI Add-on card with OOB configuration properties reported in PCI Option Read Only Memory (OPROM):

(Assume the implementation of PCI OPROM can accommodate OOB configurable properties as one of OPROM image code type).

If UEFI PCI bus driver recognizes this OOB configurable property code type in OPROM image, UEFI PCI bus driver can register this PCI device as an OOB manageable system peripherals through UEFI OOB management. The data model and properties of PCI device is stored/retrieved through UEFI OOB Management Storage Driver. This is the on-demand PCI device OOB management.

4. The Benefit of using Out-Of-Band Management on UEFI System Firmware
   - Convenience:

Being able to provide single or multiple UEFI OOB Management Driver instance(s) and OOB Management Storage Driver instances for configuring UEFI drivers and system peripherals in power-off or power-on state.

- Flexible and Extensible:
  System firmware provider can provide UEFI OOB Management Driver on UEFI environment for the specific or proprietary OOB management data model of UEFI driver/system peripherals. UEFI OOB Management Driver can determine if the data model of given schema is supported or not. For example, JSON format data model, OData XML/JSON, CIM XML or 3rd party specific format of OOB manageable UEFI driver/ system peripherals data models. System firmware or 3rd party provider provides the corresponding UEFI OOB Management Driver instance on system firmware for handling the specific format of data model. System firmware provider provides the UEFI OOB Management Storage Driver instance on UEFI environment for storing OOB configurable property on the specific or proprietary storage (such as iSCSI drive, NAND flash and etc.).

- No extra hardware cost:
  All of the implementations are based on UEFI system firmware environment which is firmware and software level programs, no extra hardware cost is required on server products.

5. Proposed Solution and the Implementation Details

Below is one of the possible solutions to support UEFI driver/system peripherals OOB management. We assume below functionalities are already implemented on UEFI driver image and system platform.

- The OOB data model of configurable properties (For example, Redfish properties) and the default values are stored in resource section of UEFI driver image.
- The platform has OOB management hardware facility (BMC).
- OOB management storage is behind OOB management hardware facility (Such as NAND flash and other non-volatile storages).

OOB manageable UEFI driver instance (UEFI image):

UEFI driver dispatcher loads UEFI driver image from firmware volume and analyzes the image's resource section to check if the configurable property schema and default value are available in the UEFI image's resource section. UEFI driver dispatcher then registers this UEFI driver as an OOB manageable UEFI driver through UEFI OOB Management Driver.

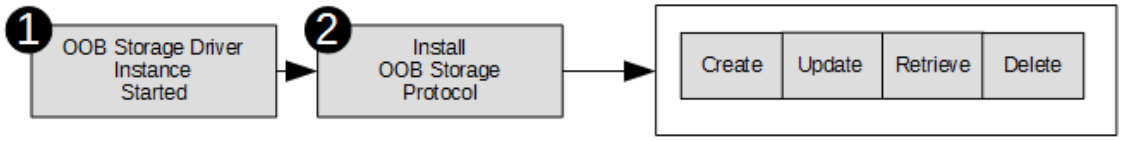
UEFI OOB Management Driver:

1. Maintain the database of registration of OOB manageable UEFI drivers
   Provide the interface to UEFI driver dispatcher for registering OOB manageable UEFI driver instance. UEFI OOB management must maintain the database in the non-volatile storage such as the NAND flash behind BMC, non-volatile ROM on system or etc.

2. Maintain the data model of configurable properties of OOB manageable UEFI drivers.
   UEFI OOB Management driver maintain the data model and default value of UEFI driver in the non-volatile storage such as the NAND flash behind BMC, non-volatile ROM on system or etc.
3. Synchronize the OOB-modified configuration properties to OOB manageable UEFI drivers.
   UEFI OOB Management Driver provides the protocol to UEFI driver dispatcher for synchronizing the OOB-modified configuration properties to UEFI driver instances.

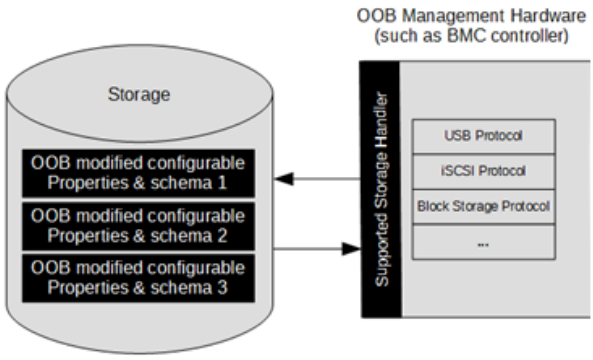
UEFI OOB Management Storage Driver:

UEFI OOB Management Storage Driver is designed to register the OOB management storage on UEFI system firmware environment. Each instance of UEFI OOB Management Storage Driver provides different kind of storage to UEFI OOB management. UEFI OOB Management Storage Driver is responsible for data CRUD (create/ retrieve/ update/ delete) actions. Upstream UEFI driver doesn't need to know how to access OOB management storage.



1. UEFI OOB Management Storage Driver Instance provides an interface for accessing to a non-volatile storage device which system provider intends to support. In this implementation example, the storage is behind BMC.
2. UEFI OOB Management Storage protocol is installed in UEFI system firmware environment during POST. This protocol provides the function entry points of Create/ Update/ Retrieve/ Delete. These functions are invoked for manipulating UEFI driver's configurable properties/data model which stored in OOB management storages. In this implementation sample, this OOB Management Storage Driver instance use the platform specific protocol between BMC and UEFI firmware to access to the storage behind BMC.

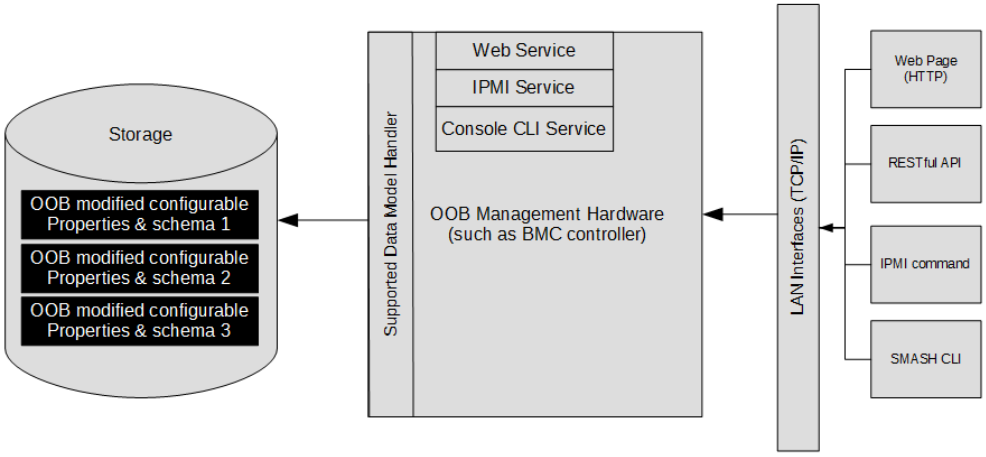
OOB Management Hardware Facility, BMC for this implementation sample:



The OOB management facility such as BMC controller is the mid-layer between UEFI system firmware and frontend user interface of OOB management facility. The OOB management facility is in charge of storing OOB configurable properties and converting the data model of configurable properties into the displayable content through frontend user interface. Such as BMC converts OOB configurable properties into HTML format. Then user can manipulate the properties from web browser through BMC frontend user interface

- Frontend User Interface:

  The OOB front end user interface provides a way to users for configuring the UEFI driver/System peripherals configurable properties. The interface to users could be a web page provided by OOB management hardware facility which always has power even when platform is shutdown. OOB management hardware facility such as BMC which has the dedicate network interface. User can connect to the Interface Protocol address (IP address) to browse or modify the UEFI driver configurable properties in out of band.



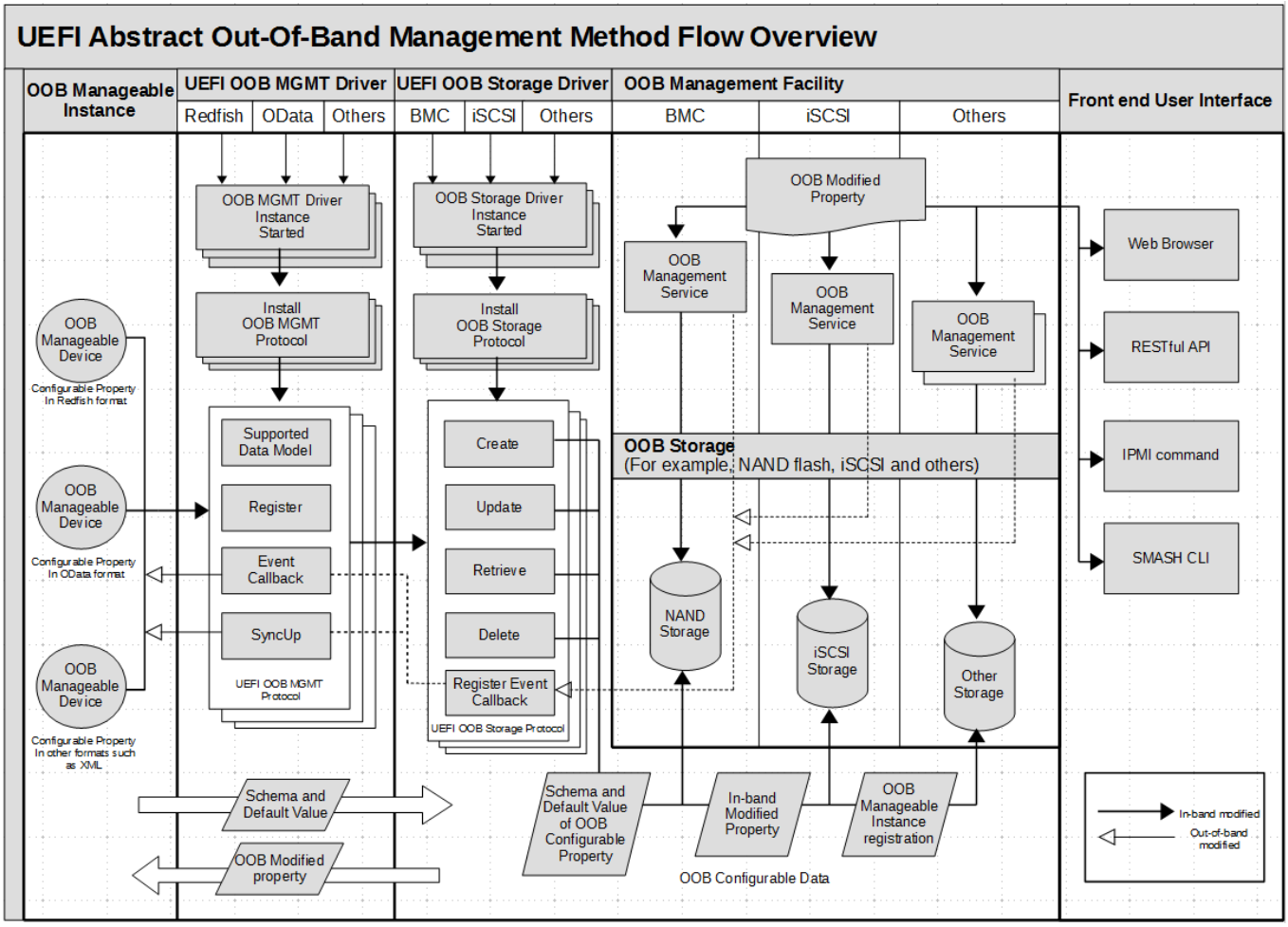
6. Flowchart

Below are the diagrams of abstract software layers of UEFI out-of-band management.

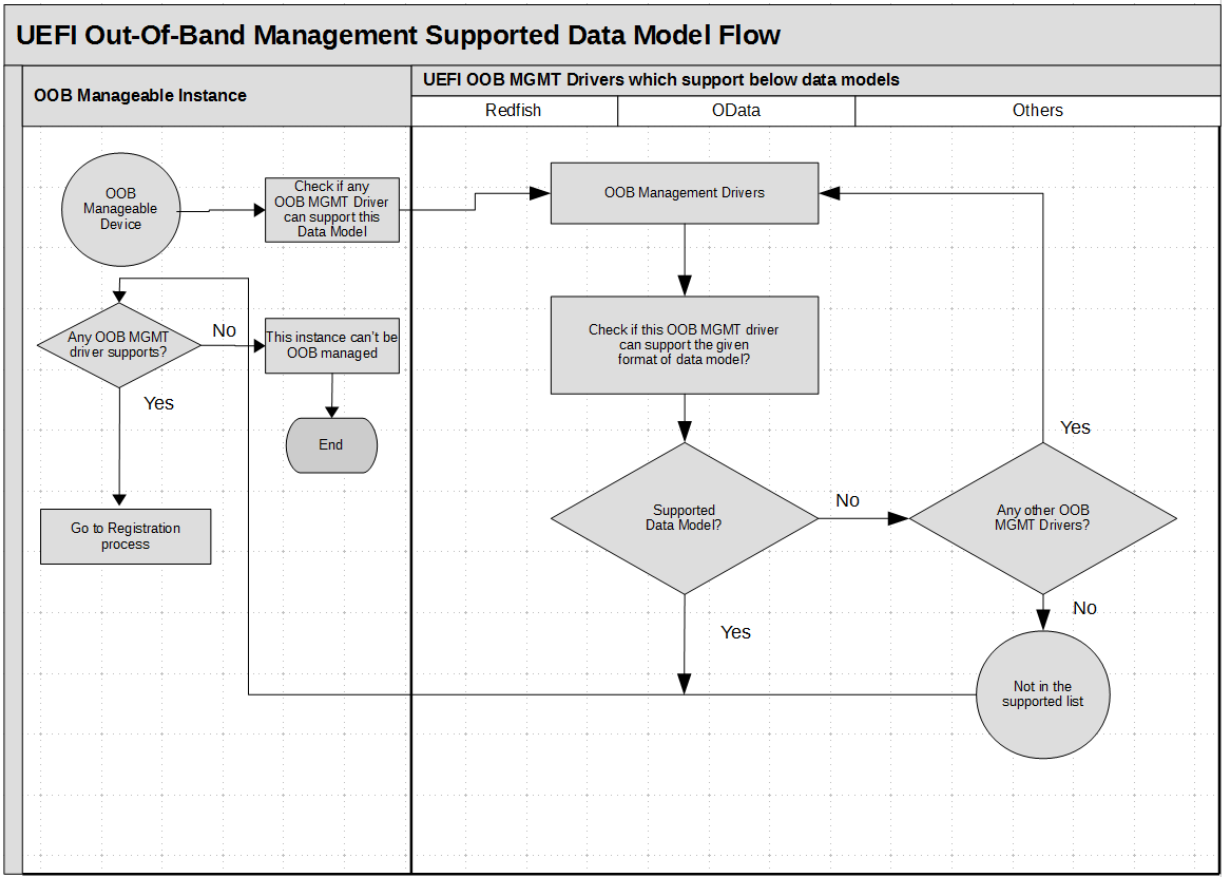
- OOB Manageable Instance
  This is the instance of UEFI driver or system peripherals which has the capability of OOB management.
- UEFI OOB Management Driver
  This is the instance of UEFI OOB management driver. Each instance has the capability to recognize different type of data models of OOB manageable instance, such as Redfish, OData, CIM-XML or others.
- UEFI OOB Management Storage Driver
  This is the driver instance of OOB management storage. Each instance provides different type of storage to accommodate data model, default properties and modified properties of OOB manageable instance.
- OOB Management Facility

This is the mechanism of OOB management facility which provide the hardware or software solution of OOB management.

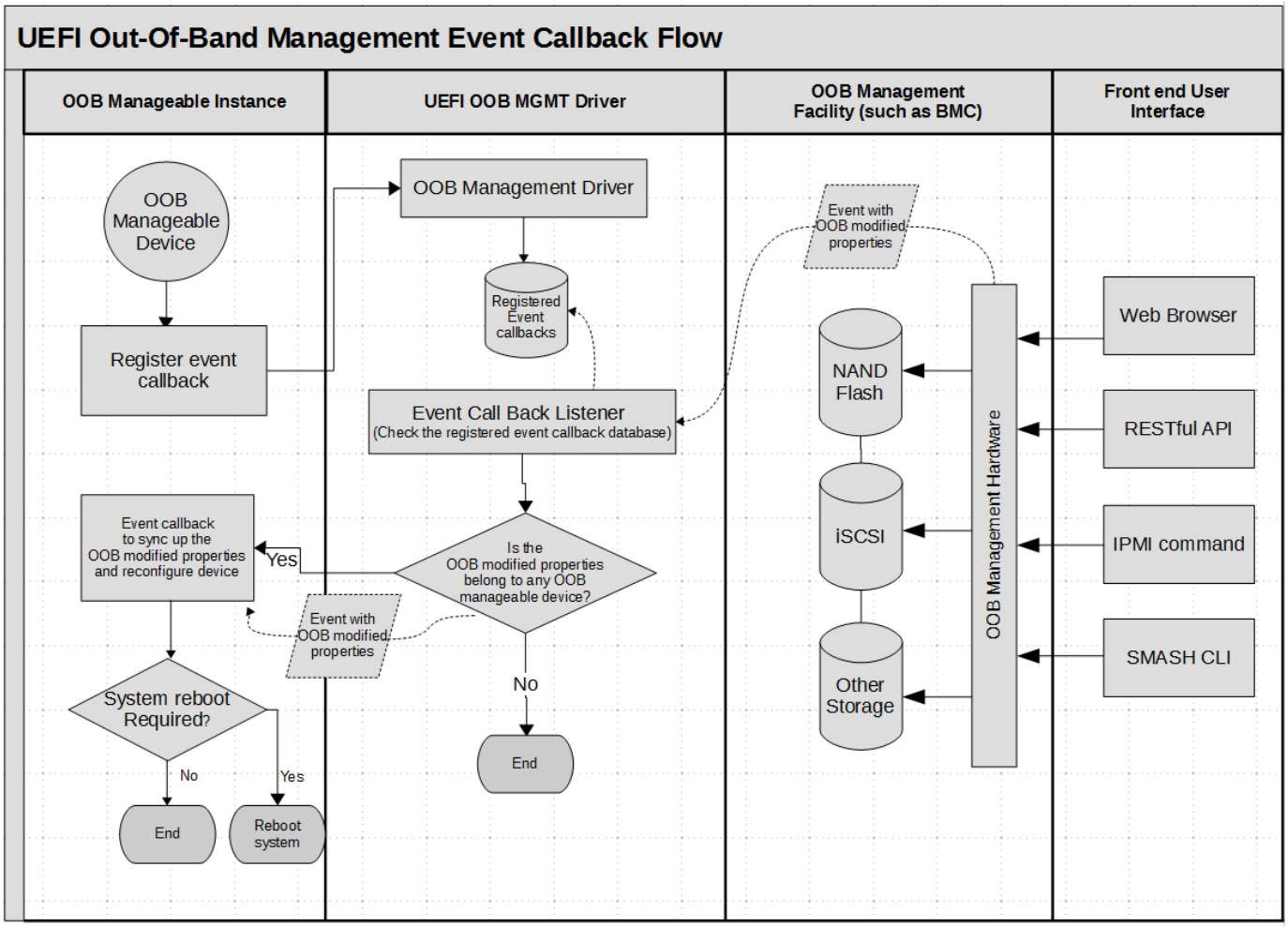
- Font end User Interface
  The font end user interface is provided by OOB management facility for configuring OOB management instance in out of band when system is in either power-on or power-off state.



Below is the flow chart of recognizing the OOB manageable instance data model,



Below is the flow chart of event handling when properties were modified in out of band through frontend user interface,



Below is the flow chart of synchronization of out of band modified properties,