Leveraging Windows Update to Distribute Firmware Updates

Model Based Servicing (MBS)

Presented by David Edfeldt,
Senior Program Manager
Windows PCE O3
(OEMs, ODMs, Operators)

presented by

Microsoft

UEFI Plugfest – May 2014

Agenda

- Model Based Servicing (MBS) Overview
  - What is MBS?
  - Why it matters
  - How it works
- Targeting + UEFI + PNP = MBS
  - Targeting
  - PNP interaction with UEFI
    - ESRT, Capsule
  - Driver details to consider
- Wrap up
- Questions

What is MBS?

A **system centric** method to **service** shipping systems

Blends a **flexible system targeting** with ability to **update both firmware and drivers**

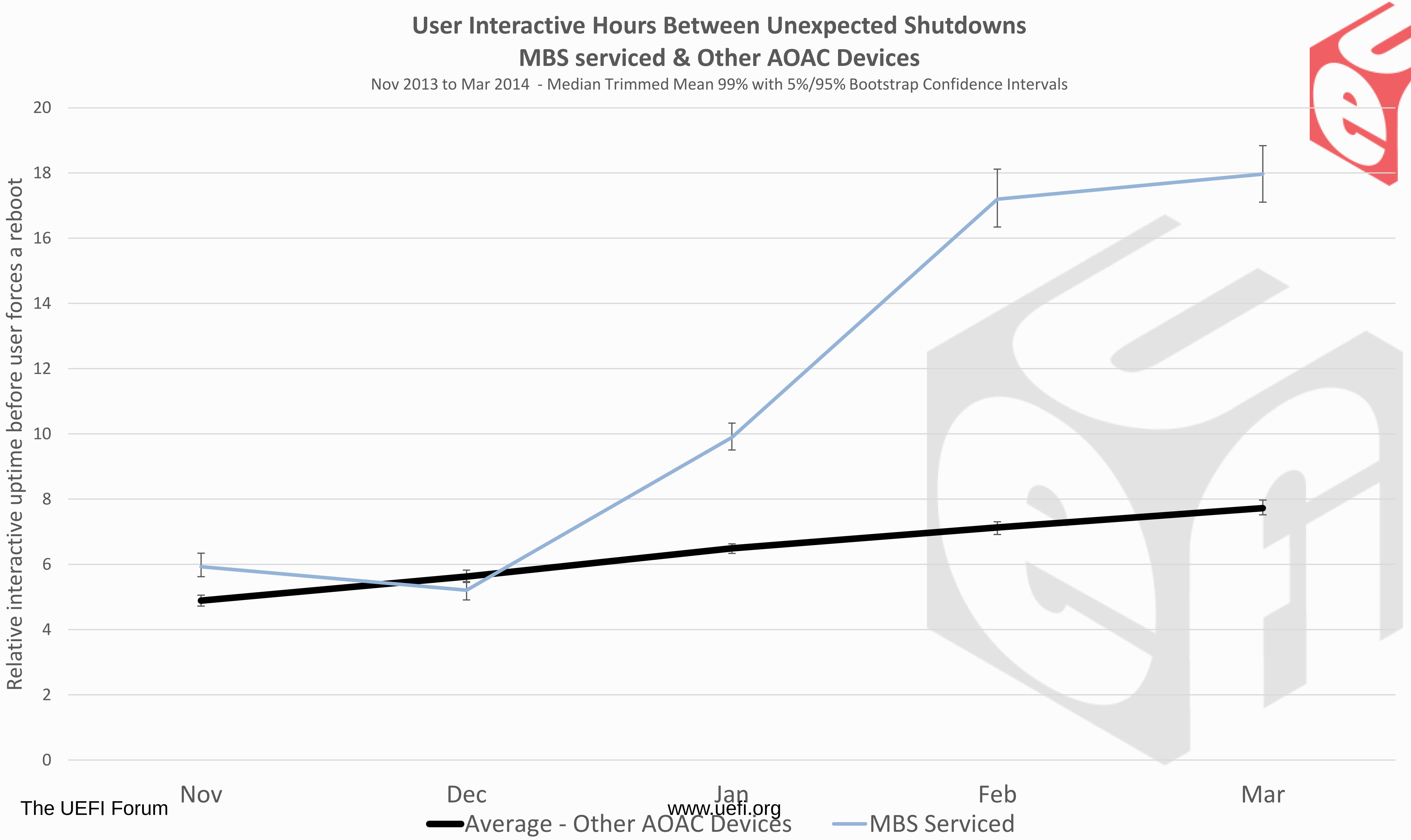
Installing **dependent** firmware and driver **sets possible**

Number of firmware/driver version mixes found in the field minimized, **simplifying support**

Telemetry data on installation success

Why MBS matters – Case study

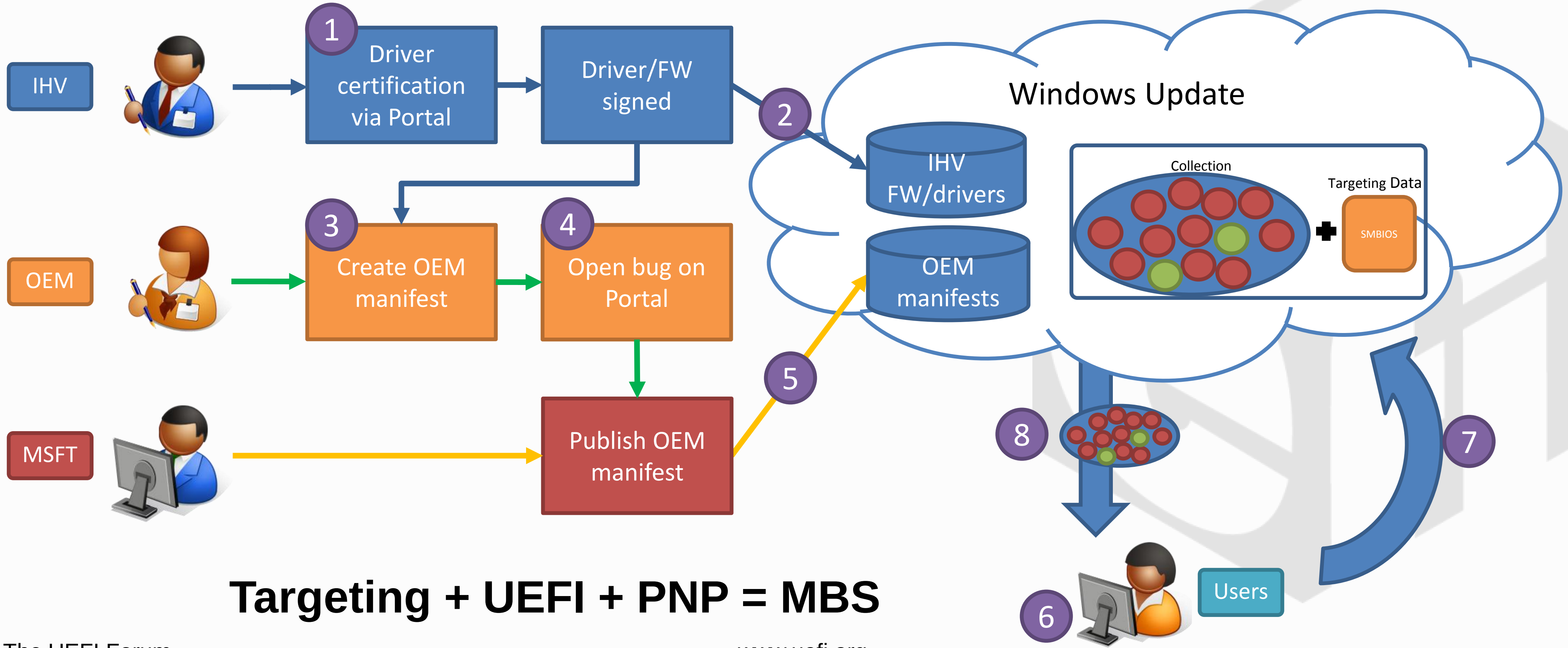
- First round of Windows 8.1 AOAC systems shared bugs with firmware and sleep states, resulting in user initiated hard reboots.
- Solution found – needed 2 firmware changes (UEFI BIOS and a Device) and drivers
- One OEM was able to use MBS to update
- Other OEMs used their proprietary services to make the change available



User Interactive Hours Between Unexpected Shutdowns
MBS serviced & Other AOAC Devices

Nov 2013 to Mar 2014 - Median Trimmed Mean 99% with 5%/95% Bootstrap Confidence Intervals

MBS set publishing workflow



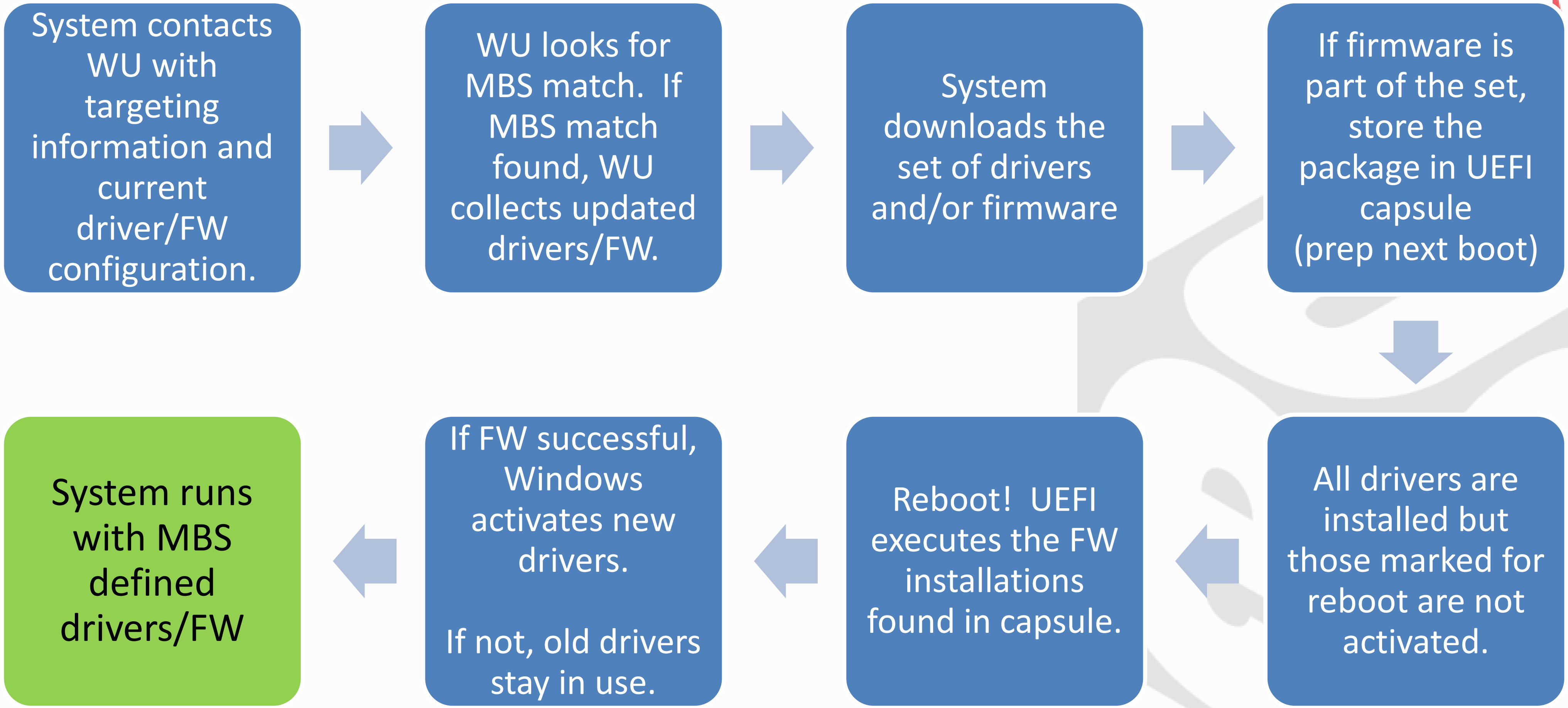
Targeting + UEFI + PNP = MBS

Targeting + UEFI + PNP = MBS

- OEMs determine **targeting** (via SMBIOS fields)
- Use **UEFI** to
  1. use EFI System Resource Table (ESRT) to enable FW update path
  2. Install FW through Capsule updating
  3. Determine if new drivers should be activated
- Use expanded **PNP** rules to
  1. Identify updatable FW
  2. Handle updating decisions with Windows Update
  3. Prep capsule for FW update
  4. Allow new driver to install beside old driver and delay activation

Targeting + UEFI + PNP = MBS

Details of MBS install



System contacts WU with targeting information and current driver/FW configuration.

WU looks for MBS match. If MBS match found, WU collects updated drivers/FW.

System downloads the set of drivers and/or firmware

If firmware is part of the set, store the package in UEFI capsule (prep next boot)

All drivers are installed but those marked for reboot are not activated.

Reboot! UEFI executes the FW installations found in capsule.

If FW successful, Windows activates new drivers.

If not, old drivers stay in use.

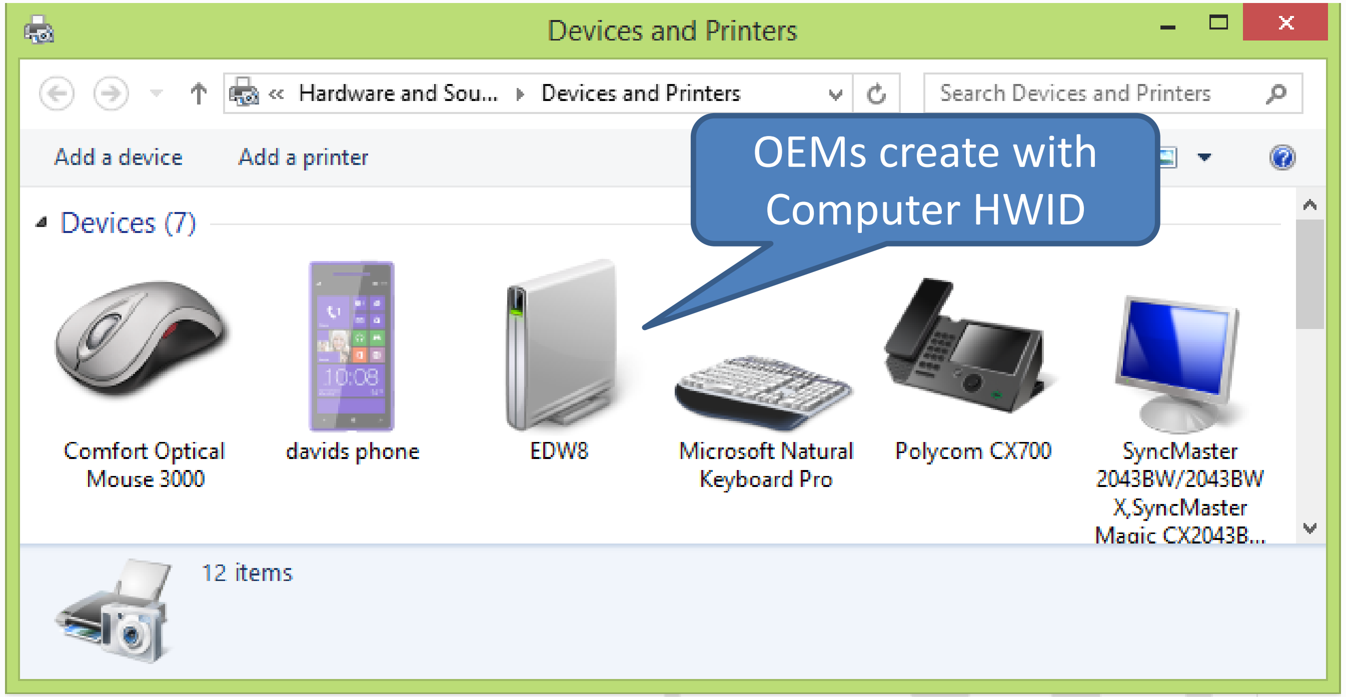
System runs with MBS defined drivers/FW

For details: **windows-uefi-firmware-update-platform.docx**.
Download at http://www.microsoft.com/en-sg/download/details.aspx?id=38405

Targeting + UEFI + PNP = MBS

Role of Computer Hardware IDs

- Computer Hardware IDs introduced in Windows 7 allows systems in device and printers view
- IDs are GUIDs generated from hashes of various model related BIOS fields
- Flexibility in which GUID to use, based on OEM's business plan for populating these fields


Devices and Printers
Hardware and Sou...
Devices and Printers
Search Devices and Printers
Add a device
Add a printer
Devices (7)
Comfort Optical Mouse 3000
davids phone
EDW8
Microsoft Natural Keyboard Pro
Polycom CX700
SyncMaster 2043BW/2043BW X,SyncMaster Magic CX2043B...
12 items

OEMs create with Computer HWID

Targeting + UEFI + PNP = MBS

Options leverage common fields

Order of Priority

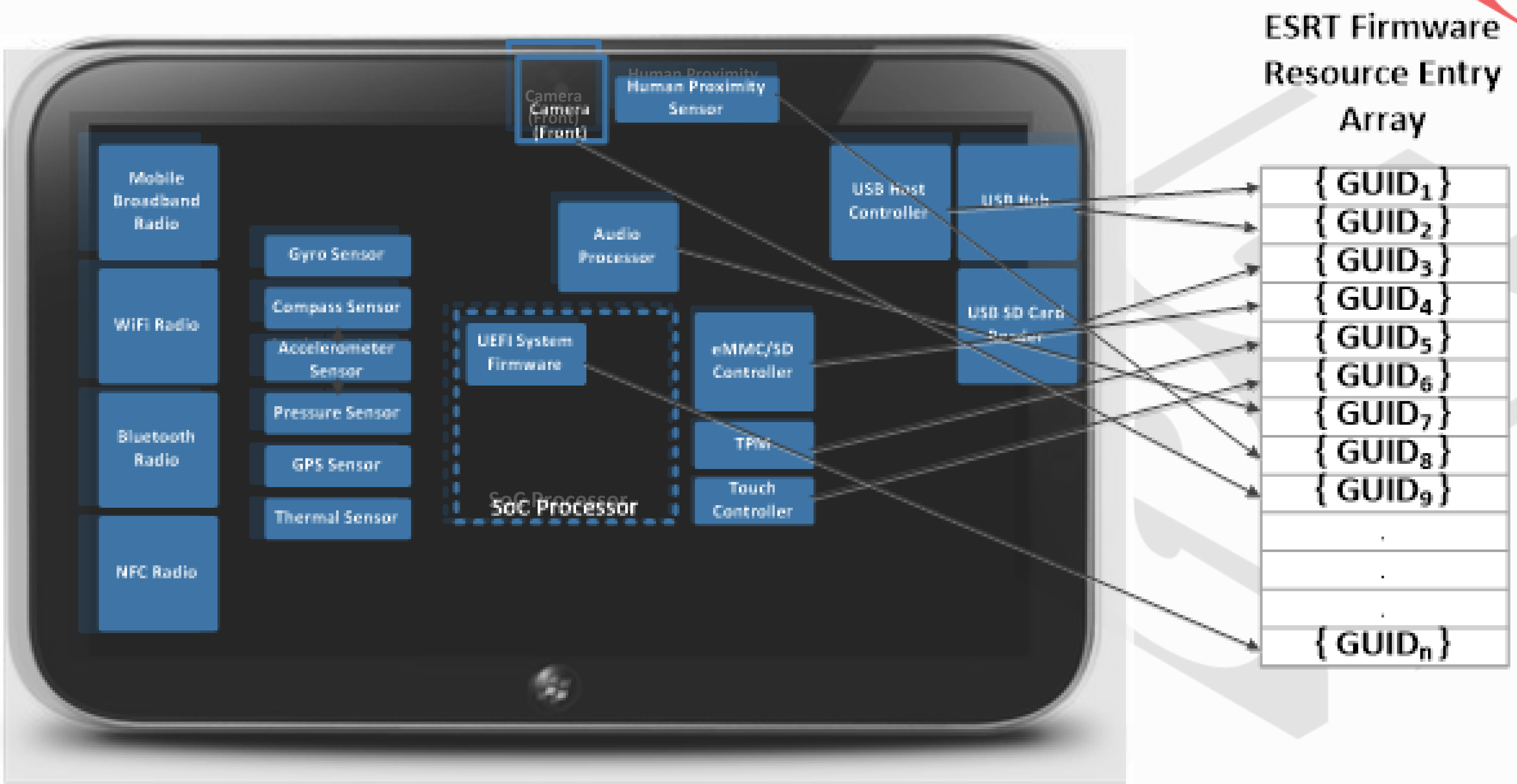
- Manufacturer + Family + ProductName + SKUNumber + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + Family + ProductName + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + ProductName + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + Family + ProductName + SKUNumber
- Manufacturer + Family + ProductName
- Manufacturer + SKUNumber
- Manufacturer + ProductName
- Manufacturer + Family
- Manufacturer + Enclosure Type
- Manufacturer

Sweet spot for most OEMs

These combinations are all usable for targeting. Which to use is OEM’s decision.

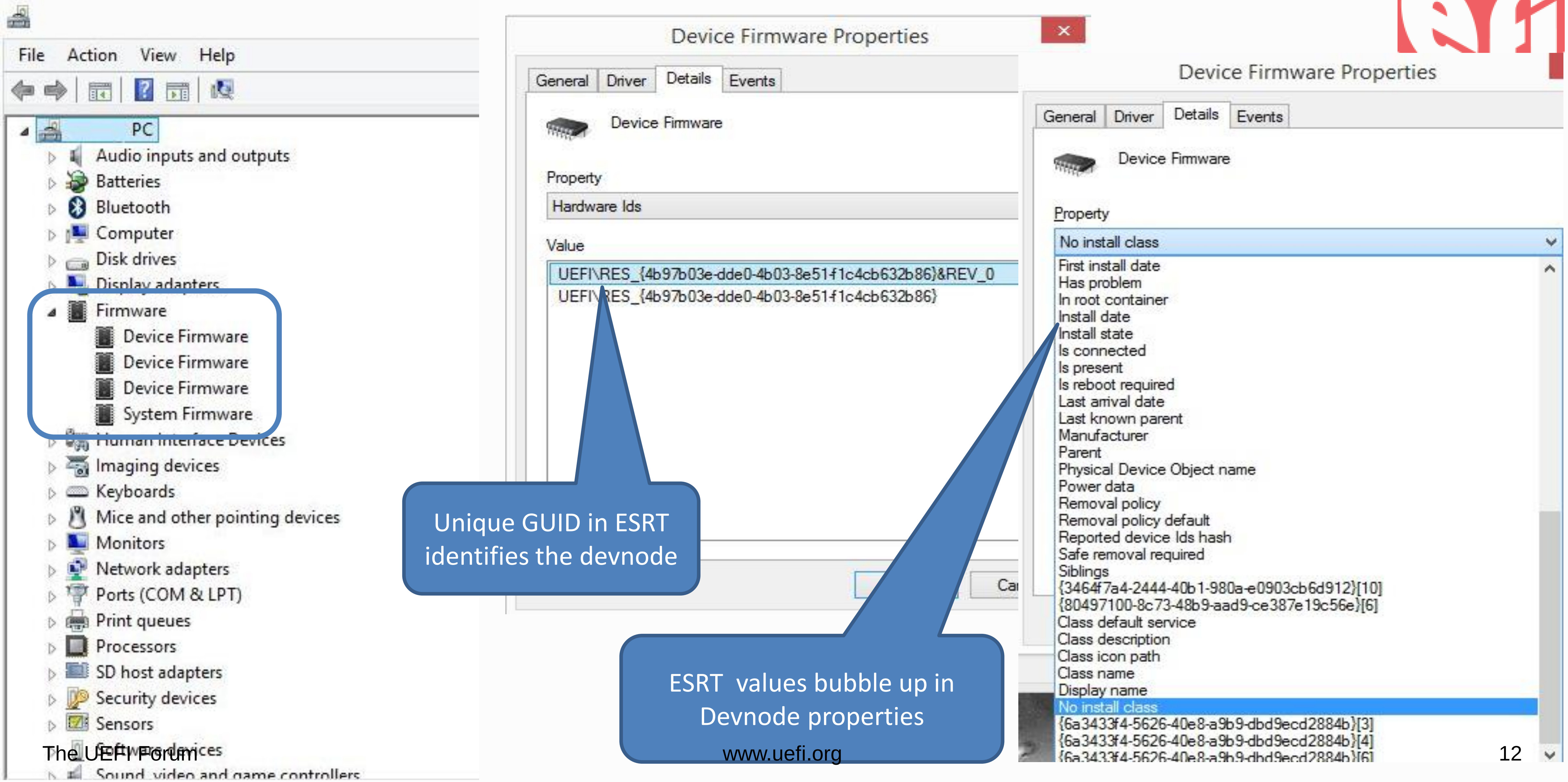
Targeting + UEFI + PNP = MBS

Starts with EFI System Resource Table (ESRT)



SoC system containing *n* firmware resources

ESRT adds Firmware in Device Manager



PC
- Audio inputs and outputs
- Batteries
- Bluetooth
- Computer
- Disk drives
- Display adapters
- Firmware
  - Device Firmware
  - Device Firmware
  - Device Firmware
  - System Firmware
- Human Interface Devices
- Imaging devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Print queues
- Processors
- SD host adapters
- Security devices
- Sensors
- Software devices
- Sound, video and game controllers

Device Firmware Properties

Device Firmware

Property: Hardware Ids

Value:

```
UEFI\RES_{4b97b03e-dde0-4b03-8e51-f1c4cb632b86}&REV_0
UEFI\RES_{4b97b03e-dde0-4b03-8e51-f1c4cb632b86}
```

Device Firmware Properties

Device Firmware

Property: No install class

```
First install date
Has problem
In root container
Install date
Install state
Is connected
Is present
Is reboot required
Last arrival date
Last known parent
Manufacturer
Parent
Physical Device Object name
Power data
Removal policy
Removal policy default
Reported device Ids hash
Safe removal required
Siblings
{3464f7a4-2444-40b1-980a-e0903cb6d912}[10]
{80497100-8c73-48b9-aad9-ce387e19c56e}[6]
Class default service
Class description
Class icon path
Class name
Display name
No install class
{6a3433f4-5626-40e8-a9b9-dbd9ecd2884b}[3]
{6a3433f4-5626-40e8-a9b9-dbd9ecd2884b}[4]
{6a3433f4-5626-40e8-a9b9-dbd9ecd2884b}[6]
```

Unique GUID in ESRT identifies the devnode

ESRT values bubble up in Devnode properties

Targeting + UEFI + PNP = MBS

PNP preps FW installation using Capsule

PNP mechanism sets up both driver and firmware installs

- PNPid identifies
- FW update packages in an INF, handled like a driver
- Must be signed by MS or an authority locally authenticated
- PNP places the Firmware in capsule

UEFI does the firmware install

- UpdateCapsule(), CapsuleHeaderArray, and QueryCapsuleCapabilities() used for setup, install
- A single firmware.bin for each capsule update, no dependency possible
- Multiple capsules possible
- Ordering no possible – if order important use a single package.
- Installation results recorded into ESRT so OS can determine if new driver should activate

windows-uefi-firmware-update-platform.docx
http://www.microsoft.com/en-sg/download/details.aspx?id=38405

Targeting + UEFI + PNP = MBS

Driver changes for updating with firmware

NO firmware dependencies for a driver?

- Driver does not need modification.
- Driver will install and be used immediately

Firmware dependencies for a driver?

- Driver activation waits on new firmware installed, ie, at reboot.
- Driver changes needed:
    1. Run drivers from the driver store (allows multiple driver version to be installed and kernel to pick correct version).
    2. Adjust references to HAL extensions since can no longer be assumed to be in same location.
    3. Include the reboot directive on any drivers that should not install until after the firmware is installed.

Documentation on writing drivers with activation on reboot is available.

Keys to success:

1. Thoughtful use of SMBIOS fields for targeting
2. ESRT completed so firmware can be targeted
3. Firmware updates designed to leverage capsule installations
4. Firmware updates packaged in INFs
   so they can use PNP to install.
5. Driver INFs configured so installation can occur with firmware

Status of MBS

- Limited to participating SoC systems H1/2014
- 8 OEMs either using or testing currently
- Looking to enroll up to 3 OEMs to expand across product lines during 2014
- Further targeted training in planning stages

Interested? Follow-up with david.edfeldt@microsoft.com

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

presented by

Microsoft