

*presented by*



**Microsoft**



# UEFI Implications for Windows Server

Taipei UEFI Plugfest– March 18-22, 2013  
Presented by Arie van der Hoeven (Microsoft Corporation)

# Agenda



- Windows Server 2012 UEFI features
- Boot Flows
- Certification Basics on Windows Server 2012
- UEFI Challenges
- UEFI Driver Signing
- Resources
- Q&A





# Advantages of UEFI vs. BIOS

Interface	Legacy BIOS	UEFI
Architecture	x86 / X64 only	Agnostic
Mode	16 bit (real mode)	32/64 bit
Boot Partition	MBR (2.2 TB limit)	GPT (9.4 ZB* limit)
Runtime Services	No	Yes
Driver model	No	Yes
POST Graphics	VGA	Graphical Output Protocol (GOP)

\* A zettabyte is equal to 1B terabytes. The total amount of global data was expected to pass 1.2 ZB sometime during 2010.

# Multicast Support



- Traditional unicast image deployment methods require each system to set up an individual connection
- Windows systems that support UEFI can perform multicast image deployment
  - Image sent to multiple “listeners” at the same time
  - Any client that joins while the multicast is underway can receive the latter portion of the image, and then wait for the server to start another broadcast to fill in the first portion
  - Great for manufacturing - clients can simultaneously receive images without overwhelming the network
  - For Windows Server 2012, both IPV4 and IPV6 must be supported
  - Supported in Windows Server 2008

# Secure Boot



- Windows Server 2012 taps into UEFI's Secure Boot to ensure that the pre-OS environment is safe and secure.
- Secure Boot is a UEFI feature not a Windows Server 2012 feature

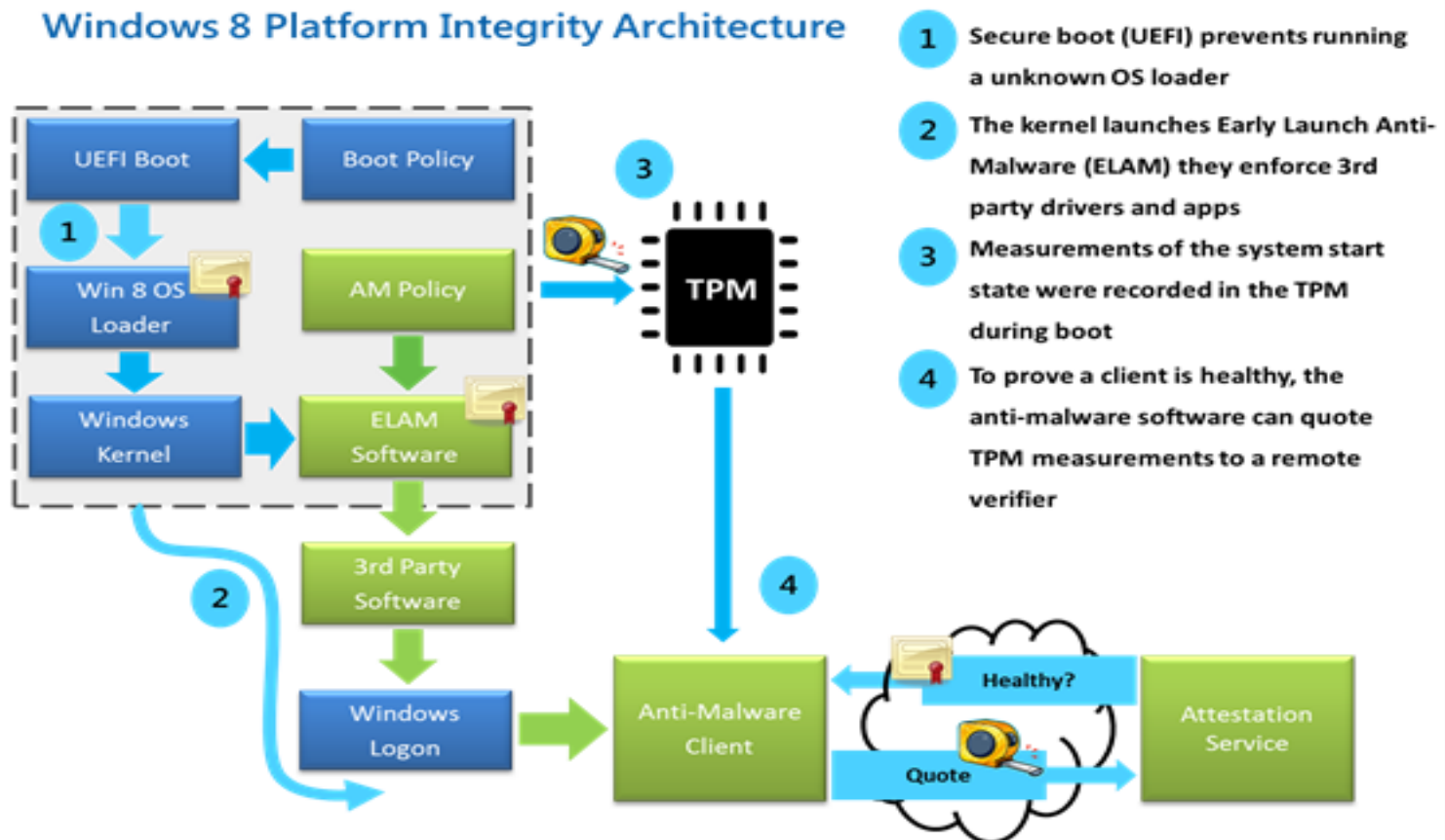


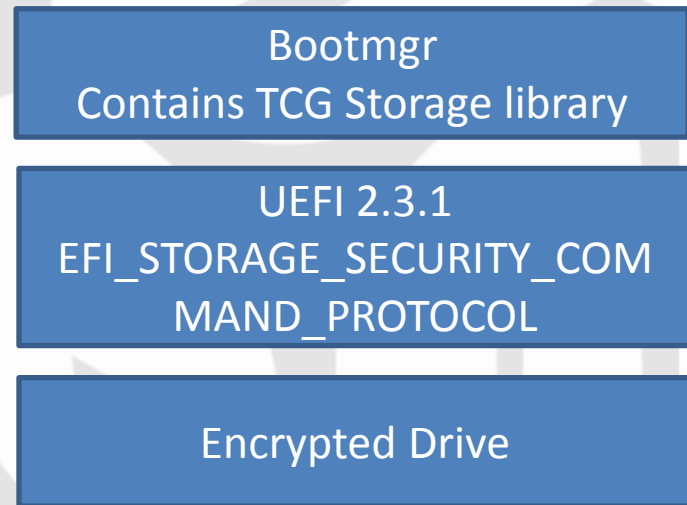
Figure 1 – Platform integrity architecture

# Encrypted Drive – Boot Support



- Offloads bulk encryption operations to the hard drive
- Improves boot time, runtime CPU usage and battery life (for non-server)
- Enables instant provisioning
- Requirements:
  - UEFI 2.3.1 `EFI_STORAGE_SECURITY_COMMAND_PROTOCOL`
  - Not compatible with legacy BIOS mode

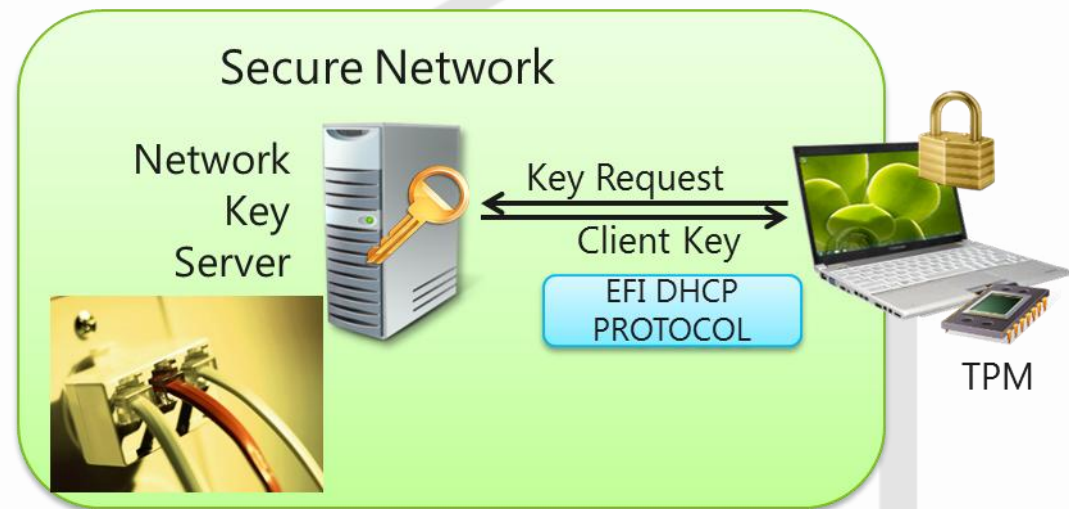
- Pre-boot Encrypted Drive Stack:



# Network Unlock for OS Volumes

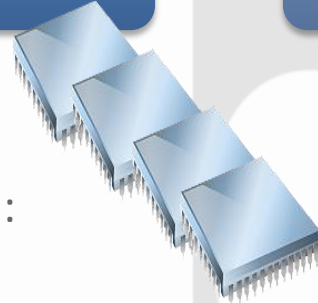


- Enables PC's connected to corporate network to boot without PIN
- Simplifies patch process for servers and desktops, wake on LAN, ease of use for end users



- Requirements:
  - UEFI 2.3.1 support for DHCPv4 and DHCPv6 protocols

# Optional Hybrid Boot Support in Windows Server 2012



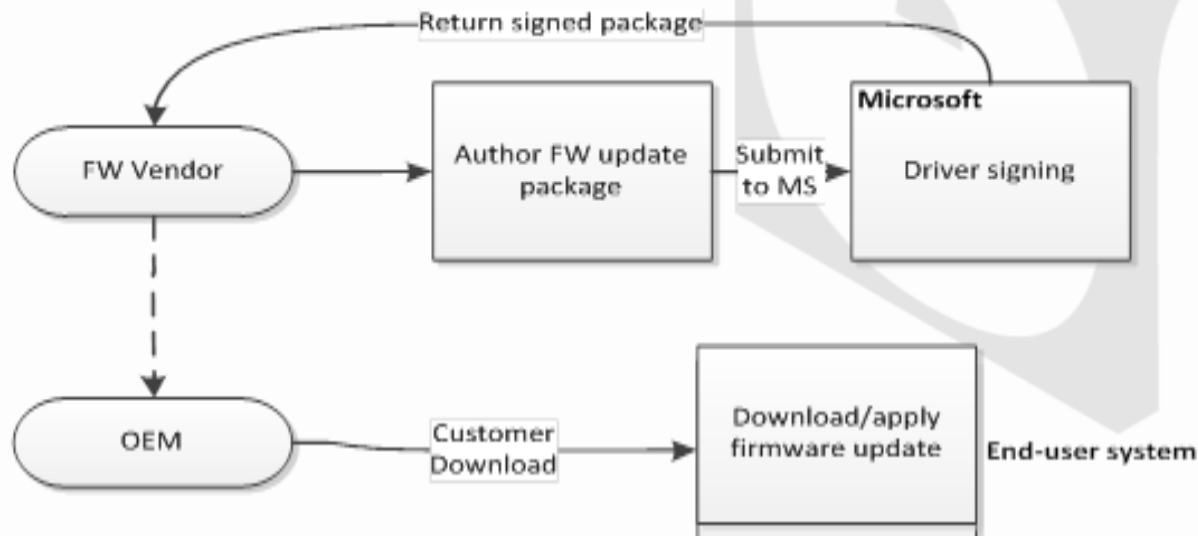
- System memory restoration:
  - Broken in 2 pieces
  - Enables the parallelization of decompression and data restoration during second phase of resume
  - Highly optimized, and dependent on system configuration
- Encryption/Decryption algorithms are right-sized for the platforms capabilities
- Optimized path used for both hibernate resume and hybrid boot



# UEFI Update Capsule Firmware Update



- Windows Server 2012 introduces support for UEFI *UpdateCapsule()*
  - *Generic* means for firmware update
  - Firmware provides versions through UEFI System Resource Table (ESRT)
  - Gets revised on successful security update; no rollback to earlier versions
  - Firmware must seamlessly recover from failed updates



# Server UEFI Drivers and Apps



- Remote management
- Security
- Custom UEFI Apps
- Use of runtime services using get/set UEFI variable APIs
- Rich UEFI/BIOS Menus

# UEFI Advantages with Windows Server 2012

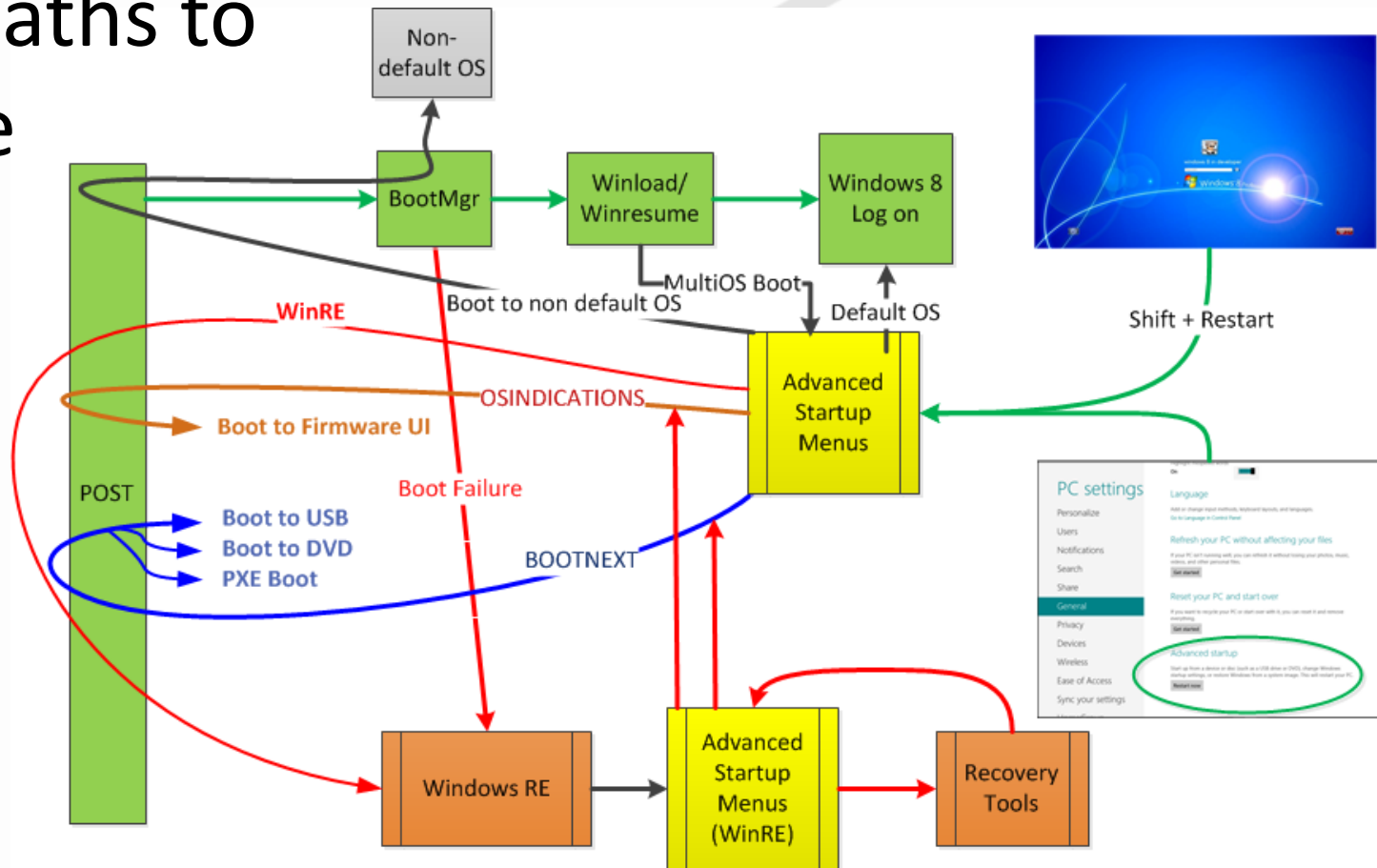


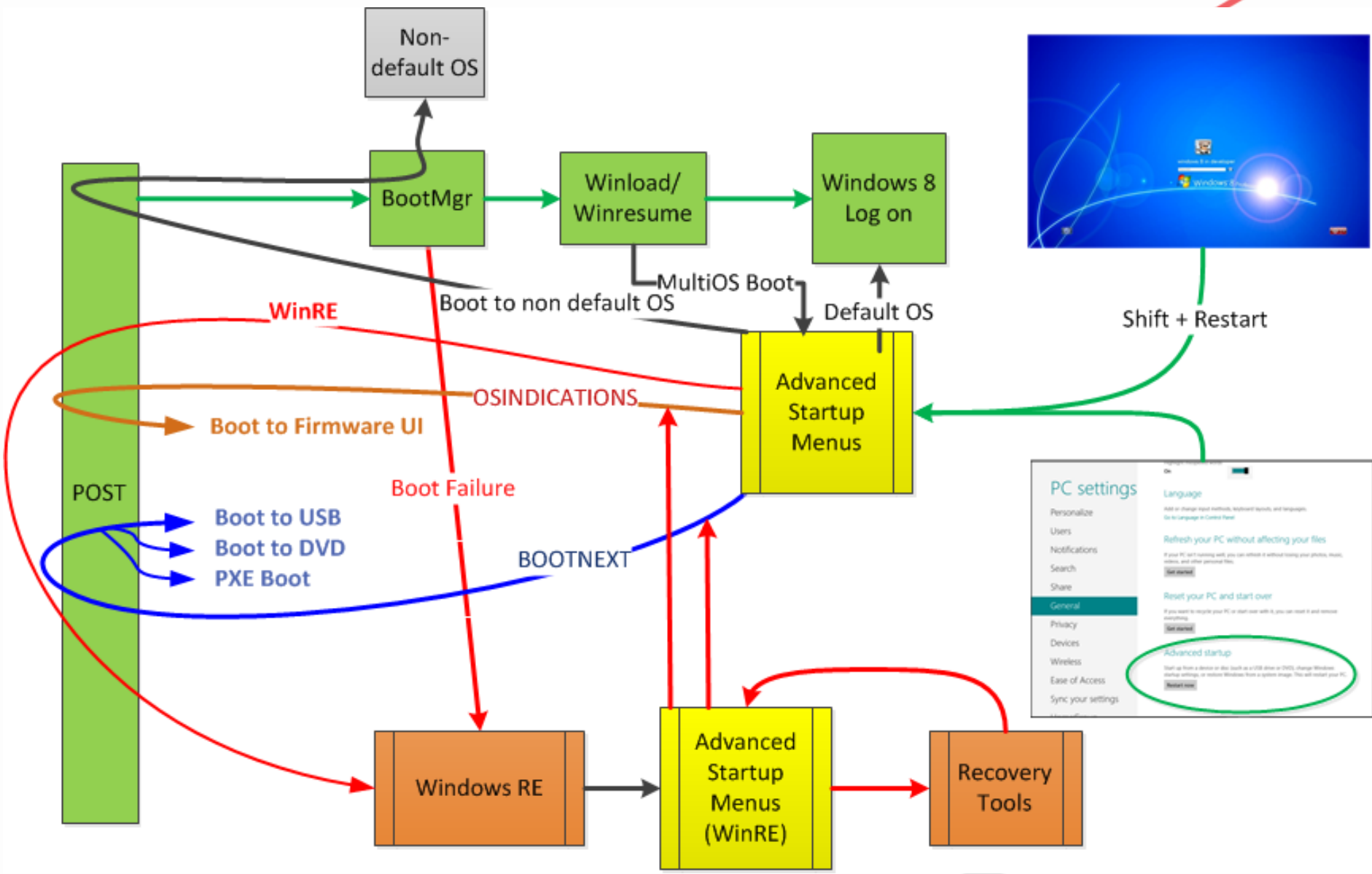
Windows OS and SKU >	WS 2012 UEFI mode	WS 2008 R2 UEFI mode	WS 2012 BIOS mode	WS 2008 R2 BIOS mode
GPT (>2.2TB boot disk)	Yes	Yes	No	No
WDS Multicast	Yes	Yes	No	No
Secure Boot (SB)	Yes	No	No	No
Native eDrive support	Yes	No	No	No
Bitlocker Network Unlock	Yes	No	No	No
Boot to Device from OS	Yes	No	No	No
TPM 2.0	Yes	No	No	No
Attestation	Yes	No	Yes	No
Measured Boot	Yes	No	Yes	No
Hybrid Boot	Optional	No	Yes	No
GOP support for Seamless Boot	Yes	Yes	No	No
64 bit UEFI drivers and Apps	Yes	Yes	No	No
Update Capsule() Support	Yes	No	No	No

# Windows Server 2012 Boot Flows

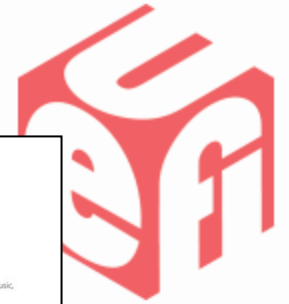


- Many paths to validate

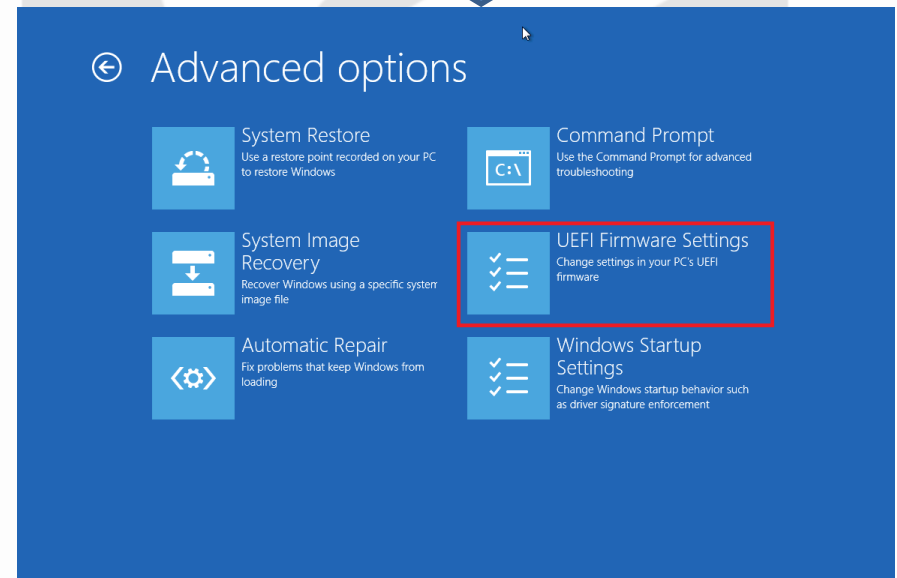
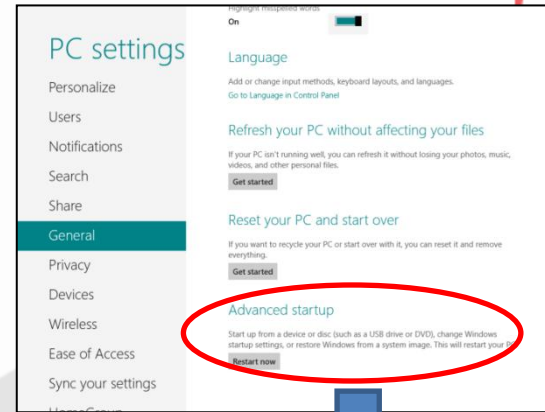




# Firmware Setup



- How it works
  - Displayed if firmware supports the UEFI variable for entering firmware setup
  - OS sets the UEFI variable and restarts the PC when option is selected by the customer
  - Firmware should display its own settings menu if variable is set at boot
- Uses UEFI “OSIndications” variable UEFI 2.3.1 Errata C



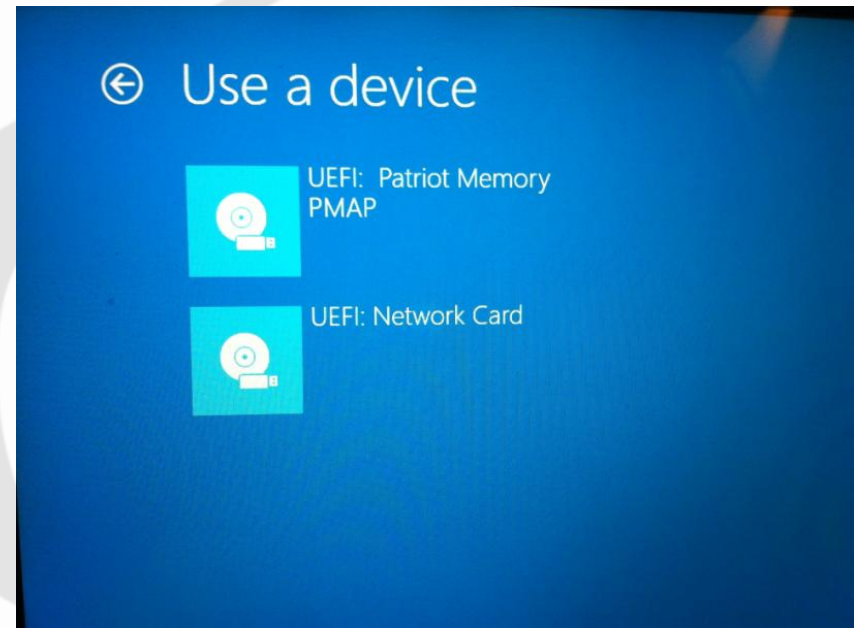
# Boot to Devices



- Recommended strings

Device	Description String
Generic USB Boot Entry	USB Drive
Hard Disk or Solid State Disk	Hard Drive
CD/DVD Device	CD/DVD
Network/ PXE boot	Network Adapter

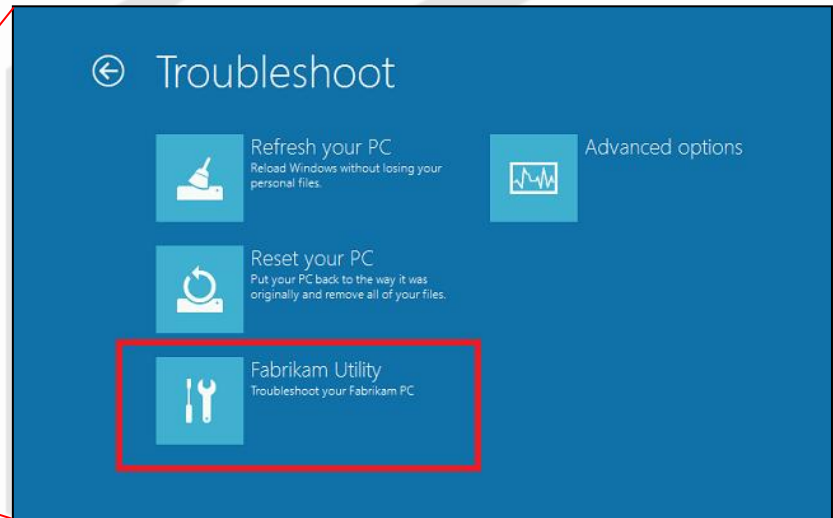
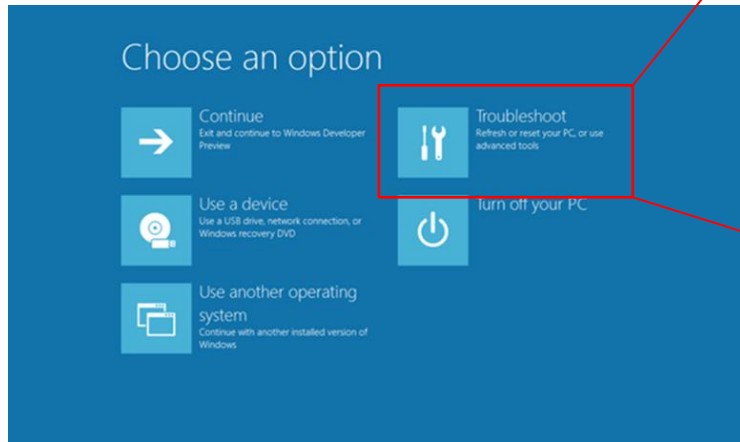
- Should include all possible boot device options
- See “Windows 8 Boot Experience Whitepaper”



# Custom Tool



- OEMs can add an extra link to launch their own diagnostic or troubleshooting tool in the “Troubleshoot” menu
  - Details on how to register the link available in ADK documentation





# Certification for UEFI Basics



- All Windows Server 2012 systems that implement UEFI 2.3.1 must support:
  - UEFI Graphics Output Protocol
  - Boot to USB, DVD, PXE
- **If Implemented**
  - BitLocker network key protector
  - BitLocker Encrypted Hard Drive support
  - TPM Requirements
  - Secure Boot
  - Secure firmware updates



# UEFI Driver Signing

- All UEFI Drivers, Applications, and OS Loaders Must be trusted
  - Trusted:
    - Signed by key or Certificate Authority in db
    - Hash of image is in db
- Does not apply to Platform Initialization (PI) phase or drivers in Core Firmware image
  - PI Phase is early firmware before the UEFI environment is launched
  - E.g. DXE drivers or UEFI drivers in the Core Firmware Image rather than loaded externally
  - Note: core firmware image must be integrity protected by the manufacturer

# UEFI Submission Review Process



- Submissions via Dev Center are reviewed twice a week
- Works for install on systems with the Windows Driver Signing CA 2011 in db (recommended, but not required)
- Remember when submitting to the UEFI signing portal to follow the package requirements:
  - Products must have production names, like "XYZ123 GOP Driver".
  - Modules must be ship-quality and should have already been tested using the [Secure Boot Windows HCK manual tests](#).
  - Modules must not allow untrusted code to execute.
  - Modules must not be licensed under GPLv3 or similar open source licenses
  - UEFI Secure Boot isn't supported by Windows for Itanium

# Resources



- Windows Server Certification Requirements <http://msdn.microsoft.com/en-US/library/windows/hardware/jj128256>
- Windows Dev Center <http://msdn.microsoft.com/en-us/windows/>
- MSDN: <http://msdn.microsoft.com/> Search on keywords like “UEFI”
- Microsoft Safety & Security Center <http://www.microsoft.com/security>
- UEFI 2.3.1. Specification errata C: <http://www.uefi.org/>
- Trusted Computing Group: <http://www.trustedcomputinggroup.org/>
- Tianocore: <http://www.tianocore.sourceforge.net>
- UEFI and Windows: <http://msdn.microsoft.com/en-us/windows/hardware/gg463149>
- SMBIOS HCT [http://msdn.microsoft.com/en-us/library/ff567493\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff567493(v=VS.85).aspx)
- New ACPI tables <http://msdn.microsoft.com/en-us/library/windows/hardware/gg463220.aspx>



# Q&A

