Microsoft

winhec

Taipei 2015

Microsoft

Building and Servicing a Secure IoT Device

Mary Chau
Software Engineer

Partner Enablement Team

Agenda

Introduction

Device Image creation

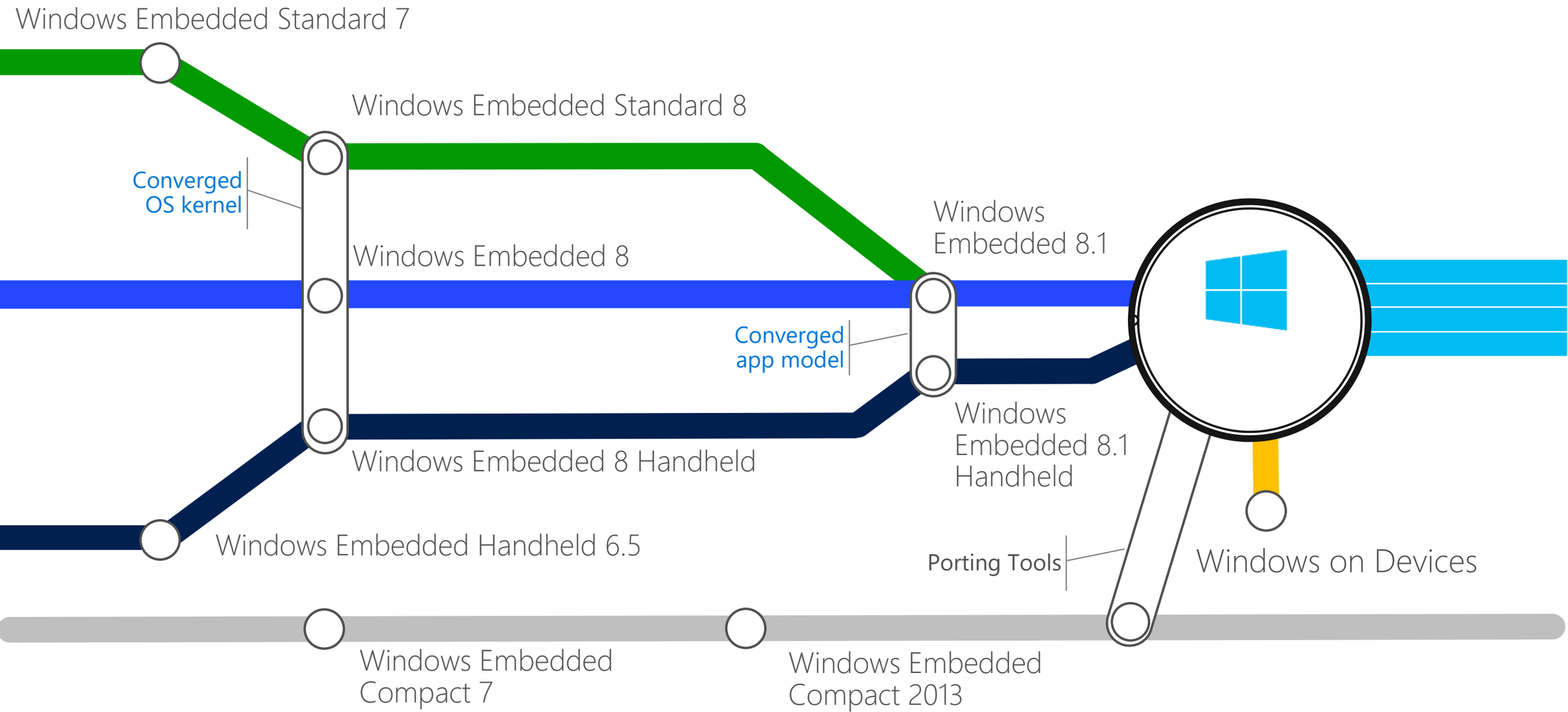
Certificates and Signing

Configuring & Provisioning device settings & apps

Securing the Device

Next Steps & Call to Action

Platform Convergence Journey



Windows 10 IoT Core

Small footprint version of Windows 10

- Provides capabilities customers already love about Windows: **security, update, tools, apps, manageability** targeting devices
- Uses the **same development tools** for all Windows 10 devices
- **Universal app support** across Windows client devices – PC, Phone, Xbox, IoT Clients
- **Purpose driven design** for custom device creation without the Windows shell; supports Headless devices.

Targeting a wide range of low cost IOT devices

- **Common image, common SDK**: no more costly & confusing custom images and custom SDKs
- Support for **wide range of devices** with different capabilities and price points

Windows 10 IoT Core

SoCs and developer boards widely available

- **Low BOM cost** and small footprint enables a range of modern low cost SoCs/boards
- **Rapid prototyping with low cost hobbyist boards and free tools**

Supports both x86 and ARM chipsets

- Ships initially with chipsets supported by **Intel, Broadcom and Qualcomm**.
- **WDK Driver Kit is available** for SV / IHVs / OEMs can bring up chipsets & systems supporting Windows 10 IoT Core.

Minimum Hardware Requirements

- X86 with PAE, NX and SSE2 support.
- ARM v7 with VFP3 and D32 (Coretex A)
- Headless - 256M RAM (128 MB free to OS) / 2 GB Storage
- Headed - 512 M RAM (256 MB free to OS) / 2 GB Storage

Universal Windows Platform (UWP)

- **Converged** APIs, write **ONE** Universal App (or Driver) and target all Windows 10 editions
- **Scale** and get **higher ROI** by selling same App to all Windows 10 editions OEMs/ODMSs
- Reuse **existing development skills**

Universal Windows Platform
Common & Consistent APIs

Languages
- C++ /CX
- C#, VB
- JS
- Python
- Node.js

UI Frameworks
- HTML
- Xaml
- DirectX

APIs
- WinRT
- Win32
- .NET

Deployment and Execution
- APPX
- App Isolation

Tools
- Visual Studio
- PowerShell
- SSH

Windows 10 IoT “Embedded Mode”

Headless mode available

APIs to access to system settings

Time zone, System Locale and Wi-Fi connection management

APIs to access busses

GPIO, I2C, SPI and easy access to custom hardware

http://channel9.msdn.com/Events/Build/BUILD2011/HW-747T

Background applications and long running tasks

Full control for your device, free of standard Process Lifecycle Management

Jump Start Dev with Low-Cost Developer Boards

Affordable, off the shelf boards:

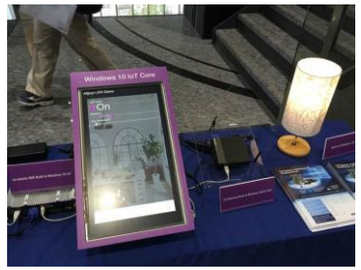
- www.windowsondevices.com

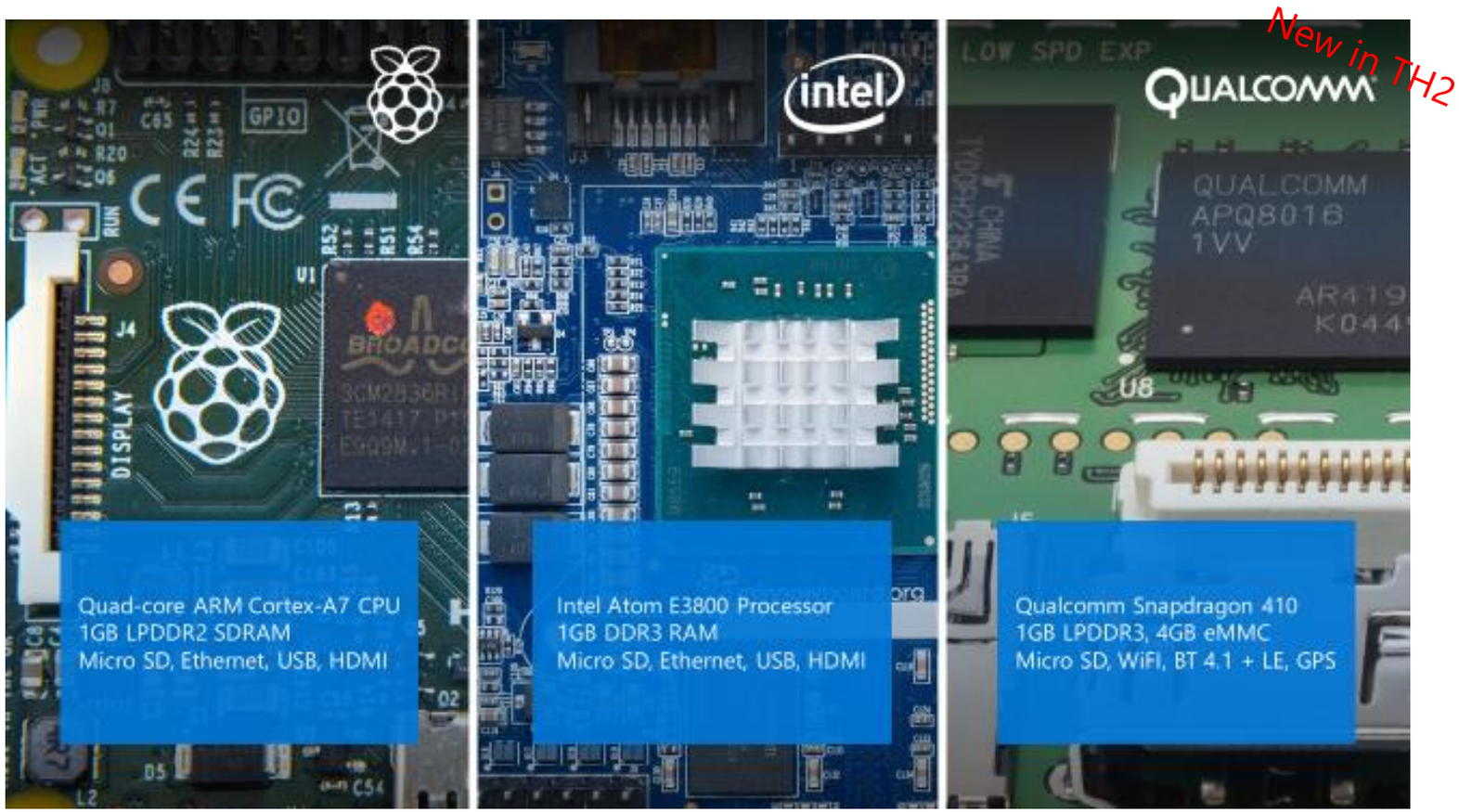
Boards Supported (blog here)

1. Raspberry Pi 2
2. Minnowboard Max
3. Qualcomm 8016-Dragonboard

Industry devices also available:

Example: Advantech Gateways





Creating an IoT Core device Image

Building, configuring and deploying Windows IoT to a device.

What's in an image?

Windows IoT Core OS, Device Drivers, Device Specific Apps and Default System Settings.

How is IoT Core image creation different from Windows Embedded

Windows Embedded Compact allowed a complex selection of OS components and settings to create an image.

- Too many options often meant that a default image, or slight variation was used
- Custom images broke cross platform compatibility as features could not be guaranteed to be included
- Limited ecosystem of BSPs, libraries and add-ons were available as a result

Windows 10 IoT Core has one complete OS image.

- Quick and easy process to get an image booted on a device
- UWP allows enables a broad ecosystem of drivers and software

Image Creation Requirements

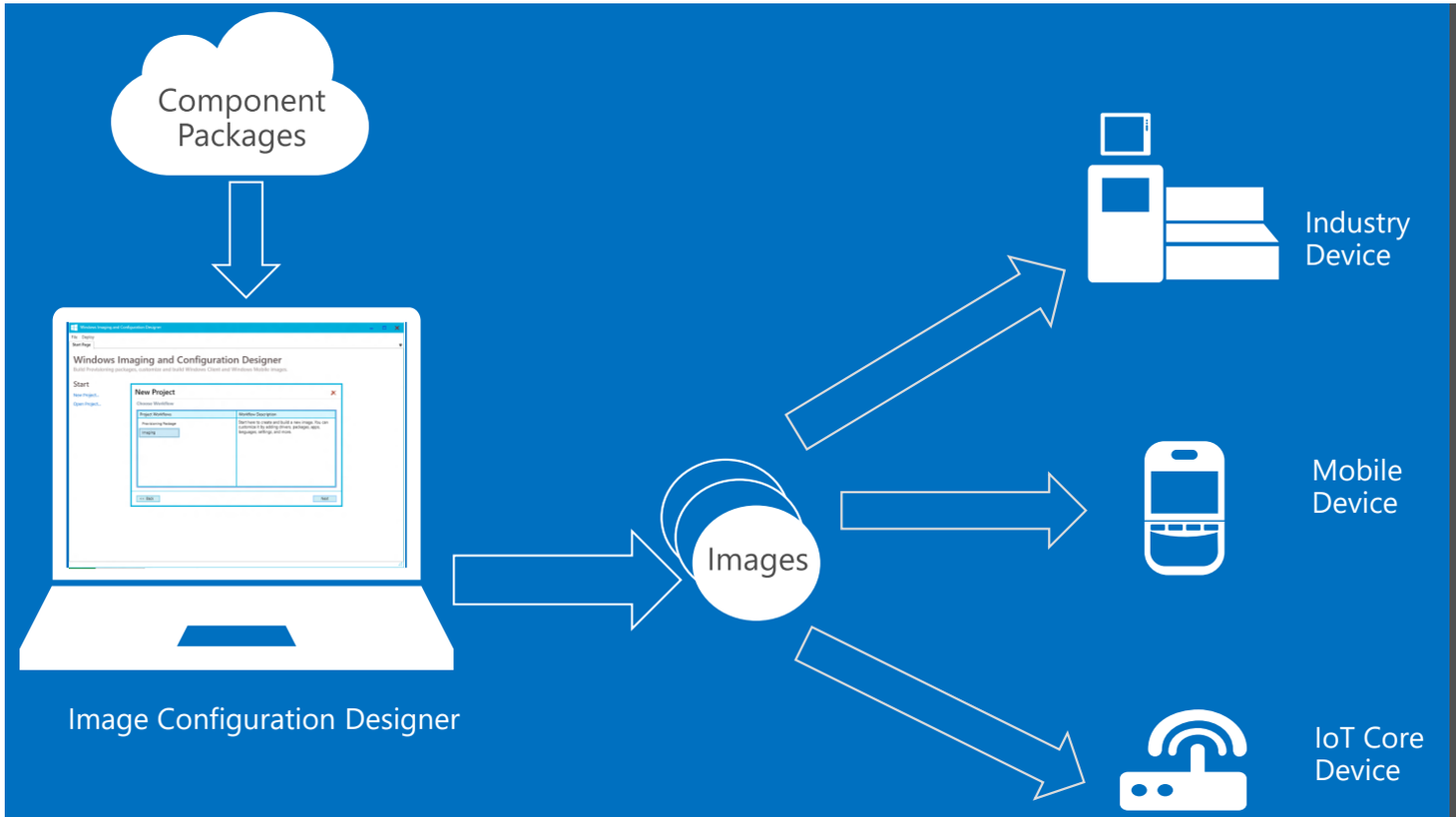
1. Visual Studio (Shell / App (and Driver) Development)
2. Windows Assessment and Deployment Kit (ADK)
   Contains image creation tools including Image Configuration Designer (ICD)
3. Latest Windows 10 IoT Core OS packages
4. Latest BSP/Drivers
   Sample BSPs for some boards. You must contact SV for latest BSP and drivers.
5. UAP app to provide UX, if Headed device
6. Image customization XLM files

https://msdn.microsoft.com/en-us/windows/hardware/dn913721

Prebuilt Developer Mode Reference Images available for Download on www.windowsondevices.com.

Configure OS for your device experience

Image Configuration Designer (ICD) making it easier for image creation and customization



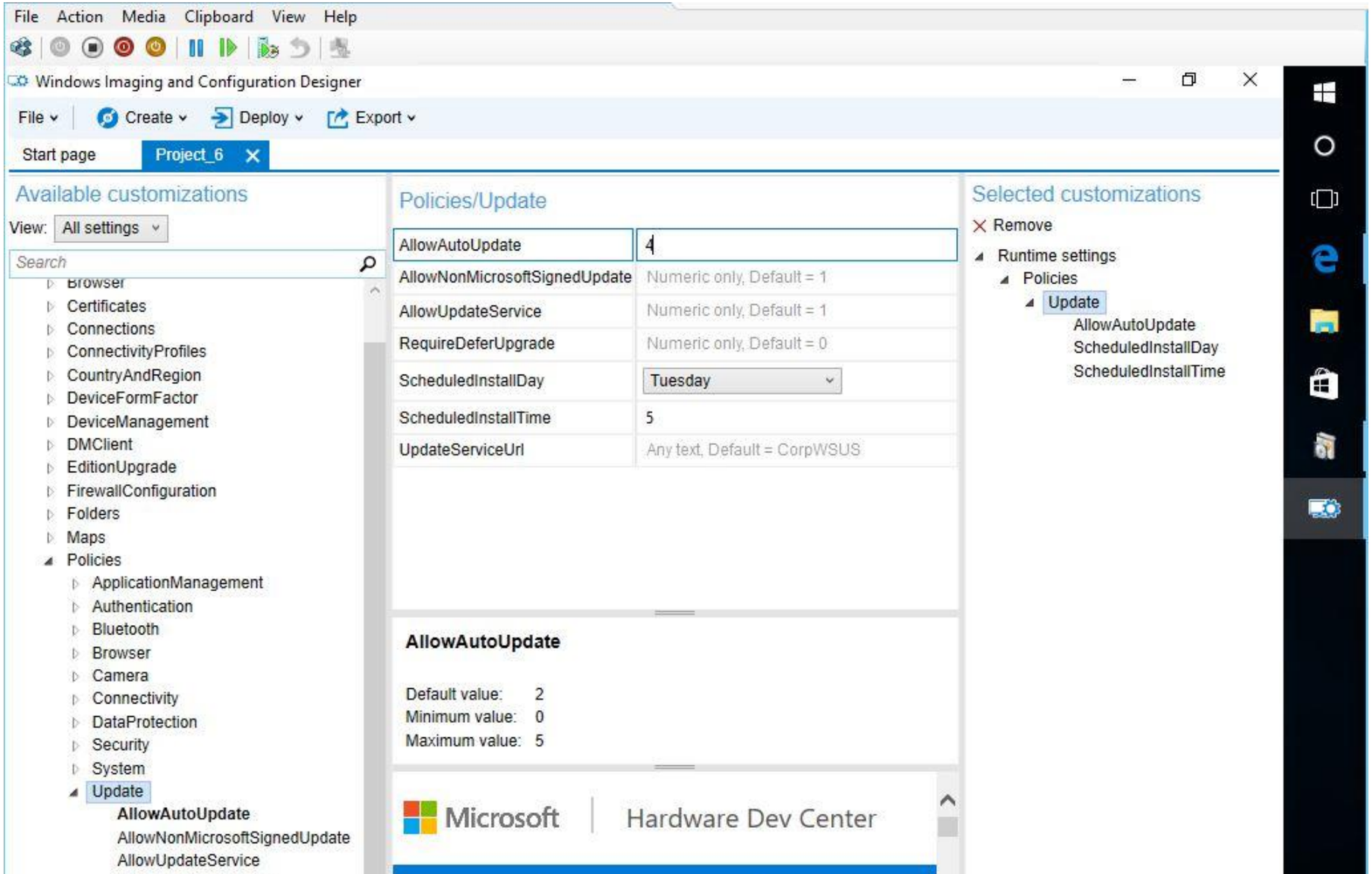
1. Customize device with Universal Applications, Drivers, Configuration settings
2. Customize experience including startup screens
3. Test experience on target device

Same tools for all OS configurations

Configure Device Update Behavior in ICD

Configure Maintenance Time other than default (Sun, 3am) to auto install updates and restart, suppress notification.
Set AllowAutoUpdate=4,
Then specify ScheduledInstallDay/Time.

Turn updates off
Set AllowAutoUpdate=5



Demo

Create FFU image through ICD

Creating Images

1. Use the ICD wizard based tool

   https://msdn.microsoft.com/en-us/library/windows/hardware/dn916113(v=vs.85).aspx

2. Use Command line tools

   https://msdn.microsoft.com/en-us/library/dn756630(v=vs.85).aspx

   Eg.

```
ImgGen.cmd OutputFile OEMInputXML MSPackageRoot [CPUType]
OEMCustomizationXML OEMCustomizationVer
```

```
imggen.cmd IoTCoreMBM.ffu
"%KITSROOT%OEMInputSamples\MBM\ProductionOEMInput.xml"
"%KITSROOT%MSPackages" x86
```

Deployment

Two deployment scenarios:

1. Deploying an *Application* to an imaged IoT Core Device

- Development scenario
- Via WEBB, FTP, Visual Studio F5

2. Deploying an *Image* to an IoT Core Device

- Production scenario
- Via device imaging / flashing

Where to get the image content packages

Microsoft Windows on Devices

FFUs are available for a set of supported devices

OS Packages to build custom images are available in full OS kit.

SV for new BSPs, Chipsets and Drivers

Some BSPs are included in the kit as a convenience. You must contact your Silicon Vendor for latest BSP and drivers if creating custom images.

1st or 3rd Party Apps

A sample UWP app is included with IoT Core to support rapid device bring up.

A custom UWP app should be developed to match your device scenario.

See https://msdn.microsoft.com/en-us/windows/hardware/dn913721 for more info and downloads.

Image definition and customization files

OS “kit” files used for image generation:

`C:\Program Files (x86)\Windows Kits\10\`

`OEMInputSamples` – OEM component and customization Files

C:\Program Files (x86)\Windows Kits\10\OEMInputSamples\MBM\CustomizationOEMInput.xml example contents

```
<BootUILanguage>en-us</BootUILanguage>
<BootLocale>en-us</BootLocale>
<BuildType>fre</BuildType>
<Resolutions>
<Resolution>1024x768</Resolution>
</Resolutions>
<Feature>IOT_ALLJOYN_APP</Feature>
<Feature>IOT_CRT140</Feature>
<Feature>IOT_BERTHA</Feature>
```

`FMFiles` – Feature Manifest Files

C:\Program Files (x86)\Windows Kits\10\FMFiles\x86\MBMFM.xml contains this reference:

```
<PackageFile Path="$(mspackageroot)\Retail\$(cputype)\$(buildtype)" Name="Intel.MBM.UART.cab" >
    <FeatureIDs>
      <FeatureID>MBM_DRIVERS</FeatureID>
    </FeatureIDs>
  </PackageFile>
```

`MSPackages` – OS component and layout Packages

C:\Program Files (x86)\Windows Kits\10\MSPackages\Retail\x86\fre**Intel.MBM.UART.cab**

Demo

Walkthrough of directory structure and key files

Signing

What is signing and why is it necessary?

Binary signing is to ensure the security of the platform by limiting the loading of binaries to those that are intact and trusted.

- App signing
- Driver signing

Developer Configuration

- Developer Package
- Developer Mode & Cert verification options

Signing options

- No Cert Verification
- Cross-Signing
- Microsoft Certification Signing

Developer Package & Mode

Why is there a developer package?

Package of development time tools and settings to make app development, device configuration, etc quick and easy when being used in a very dynamic way such as the development cycle
(Downloaded FFUs are preconfigured for VS “Developer Mode”)

What is included?

Remote Access is configured with simple preset user accounts and passwords
Command line tools to access with SSH & Powershell to quickly manage & configure device

What services are on in the image?

PowerShell, SSH, FTP, WEBB, Remote device pinger, netsh, IoTStartup, SetBootOption, SetDisplayResolution, … and more. (List is release dependent, check the release notes for complete list.)

For Development ONLY!

These tools would pose a security risk if shipped without appropriate configuration and access rights.
Do not include the developer package when building and testing your near-final / final images.

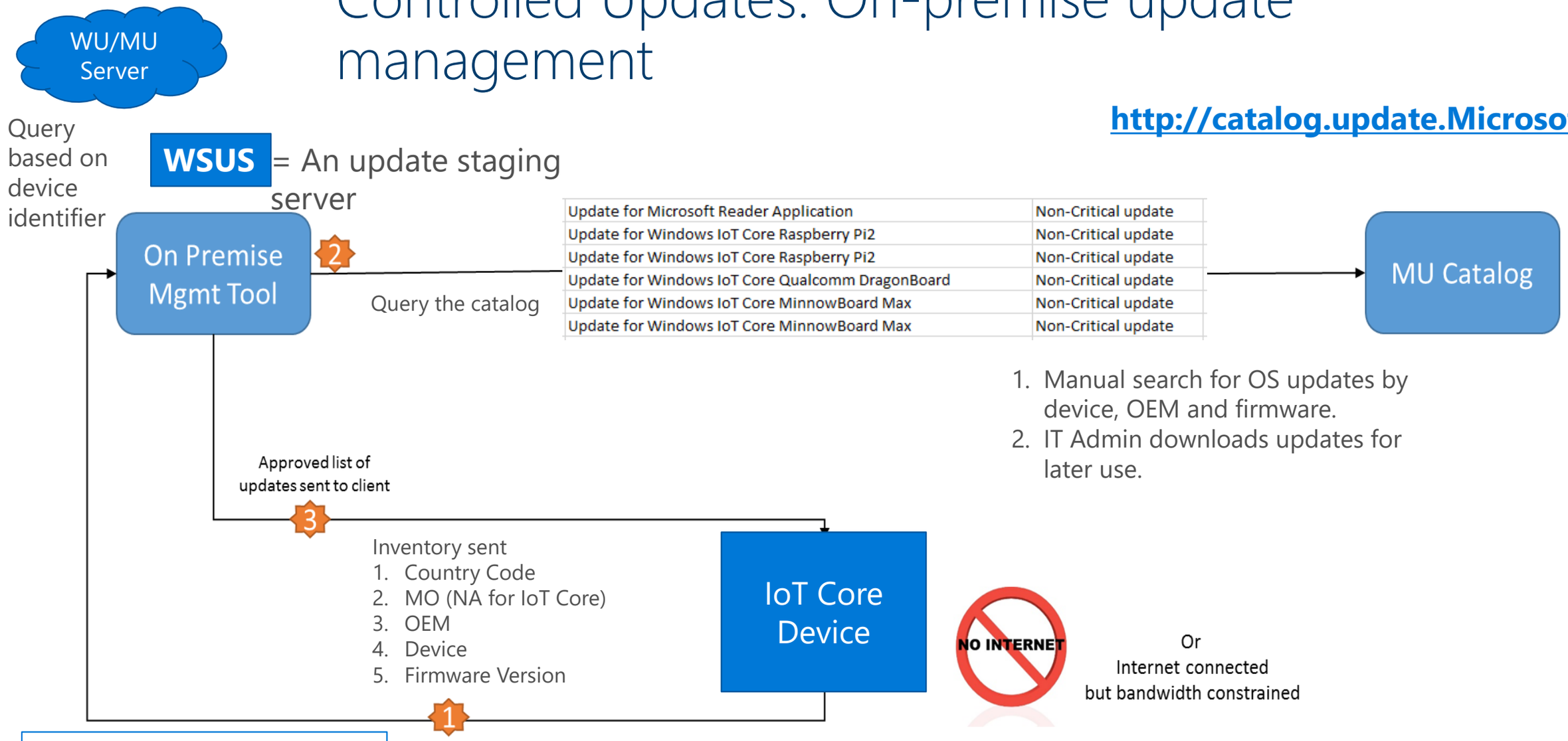
Provisioning and Device Management

Creating and applying custom settings and components.

Created with ICD

Can be used both at image creation time and via device management for devices in the field.

Controlled Updates: On-premise update management



http://catalog.update.Microsoft.com

1. Manual search for OS updates by device, OEM and firmware.
2. IT Admin downloads updates for later use.

- WSUS gets Device ID
- Sends Authorization cookie and maintains it.

- Device is preconfigured with link to WSUS server, maintenance time, auto-update/suppress notification.
- Trigger to ping server is suppressed

Demo

Provisioning AppInstall

Securing IoT Devices

- Add a TPM for best in class security with IoT Core!

  Can anyone afford an attack?

- Malware Protection

  "*Secure Boot*" and enable remote attestation with "*Measured Boot*"

- Protect customer data

  BitLocker – full *device encryption and secure key storage*

- Resist tampering

  Authenticity with a strong, hardware-bound device identity using Trusted Platform Modules (TPMs)

Win10 IoT TPM Options

- On-Chip
- Firmware
- Discrete (I2C or SPI)
- TPM Simulator-dev tool

Configuring Bitlocker

- Acquire necessary certificate (Test certs provided)
- Configure OS image settings
- Configure Security Policy settings
- Configure Firmware to enable Secure boot
- Schedule encryption
- Turn on Secure boot in image
- Verify

- More information will be available on http://www.WindowsOnDevices.com

Call to Action

- Download and explore Windows 10 IoT Core!
- Join the pre-release program to get early access to builds
- Give us feedback!

http://ms-iot.github.io/content/en-US/Community.htm#contact

Please Complete An Evaluation

Your input is important!

Microsoft

© 2015 Microsoft Corporation. All rights reserved.