Microsoft

winhec

Taipei 2015

Microsoft

Introduction to AllJoyn

Mary Chau
Software Engineer

Partner Enablement Team

Introduction and Agenda

Session Agenda:

- AllJoyn Support in Windows 10
- Device System Bridge – the “Superconnector”
- Device Management - AllJoyn Configuration Service Provider
- Development Tools and samples
- Demo

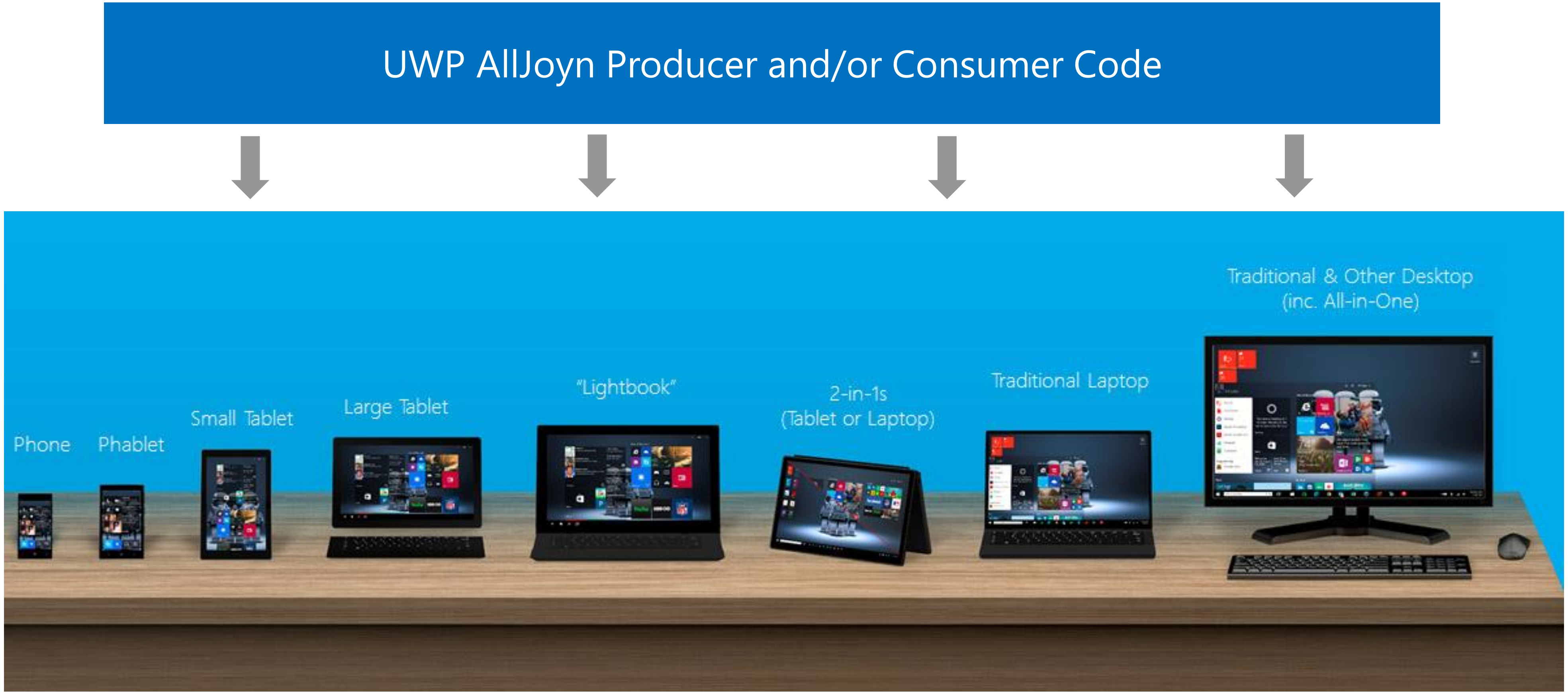
AllJoyn Support in Windows 10

AllJoyn in Windows 10

Integrated AllJoyn Runtime	Servicing, reduced code size
Router Node Service	Optimized performance, full integration
C and UWP APIs	Reduced code size, integrated with Windows SDK
Visual Studio and SDK Integration	AllJoyn Studio accelerates development for Windows
Samples	UWP Samples

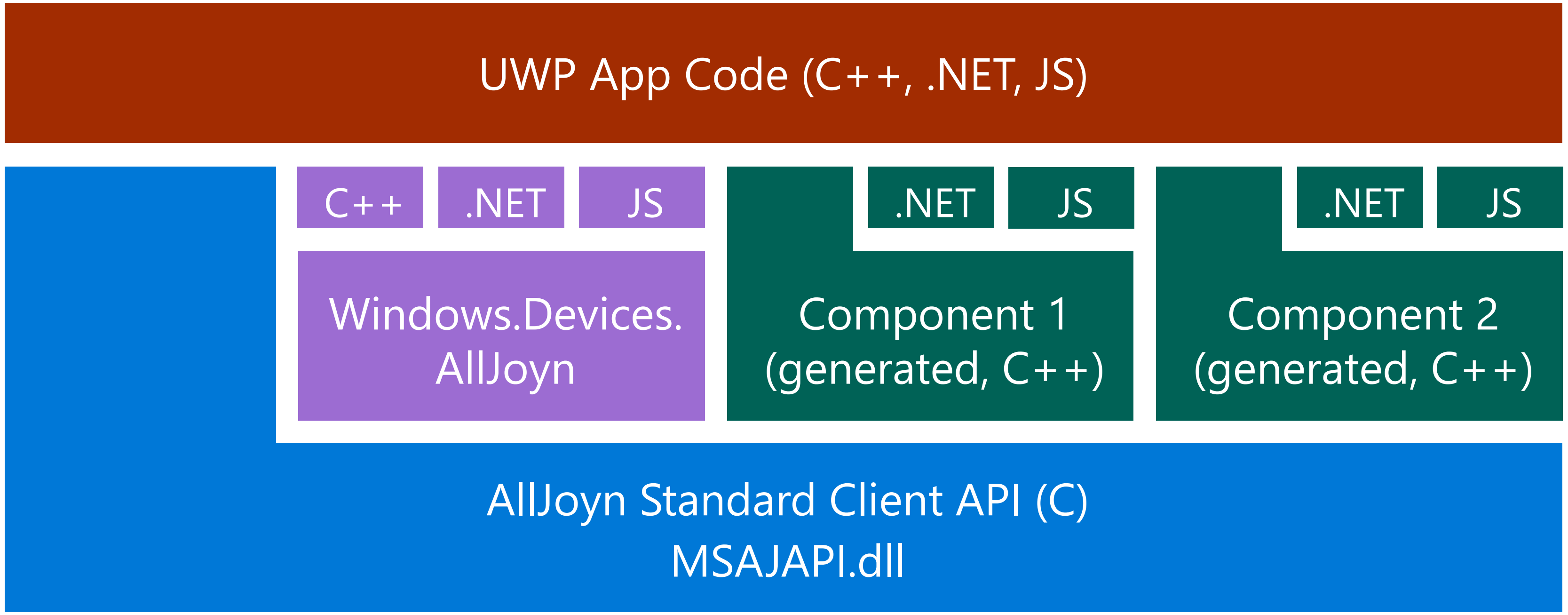
Windows 10 AllJoyn UWP and OneCore

Universal Windows Apps are built on the Universal Windows Platform (UWP), which exposes AllJoyn APIs.



UWP AllJoyn Producer and/or Consumer Code

AllJoyn UWP Apps



UWP App Code (C++, .NET, JS)

C++ .NET JS

Windows.Devices.AllJoyn

.NET JS

Component 1 (generated, C++)

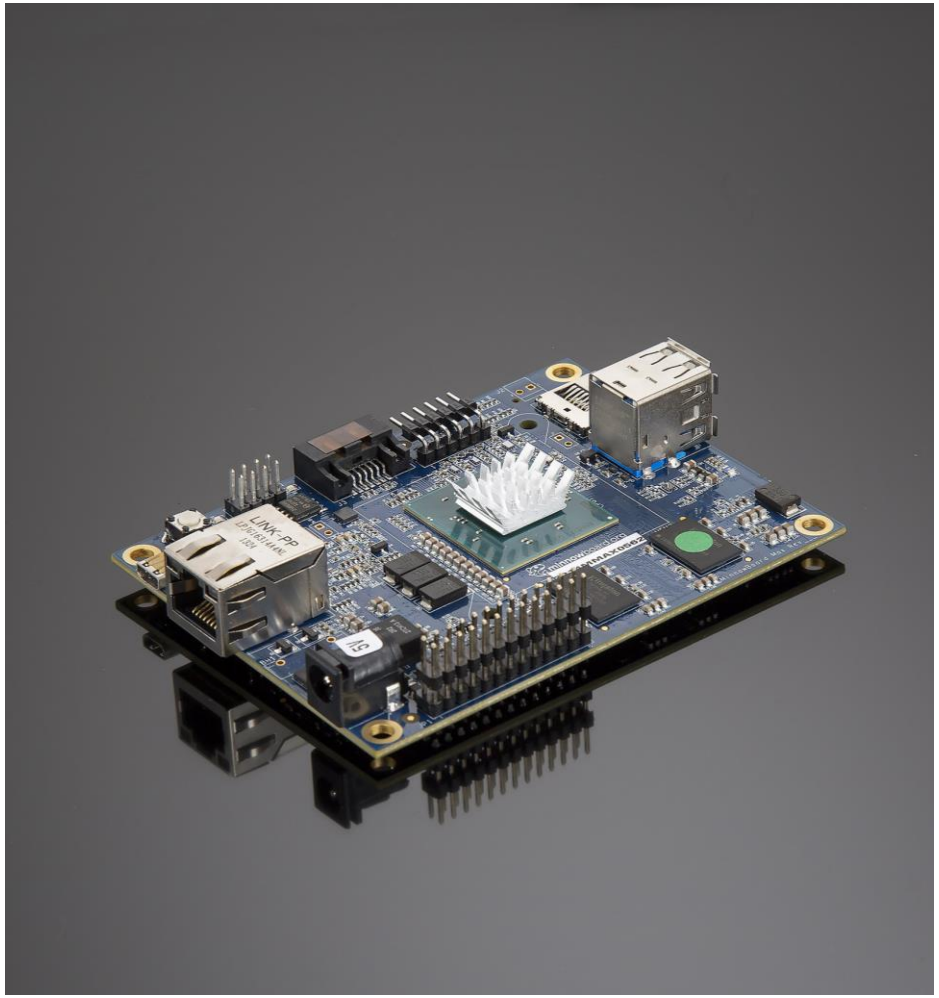
.NET JS

Component 2 (generated, C++)

AllJoyn Standard Client API (C)
MSAJAPI.dll

AllJoyn and Windows 10 IoT Core

- Router Node & UWP APIs
- Startup Tasks
- Busses & Interfaces
- Connectivity



AllJoyn Device System Bridge

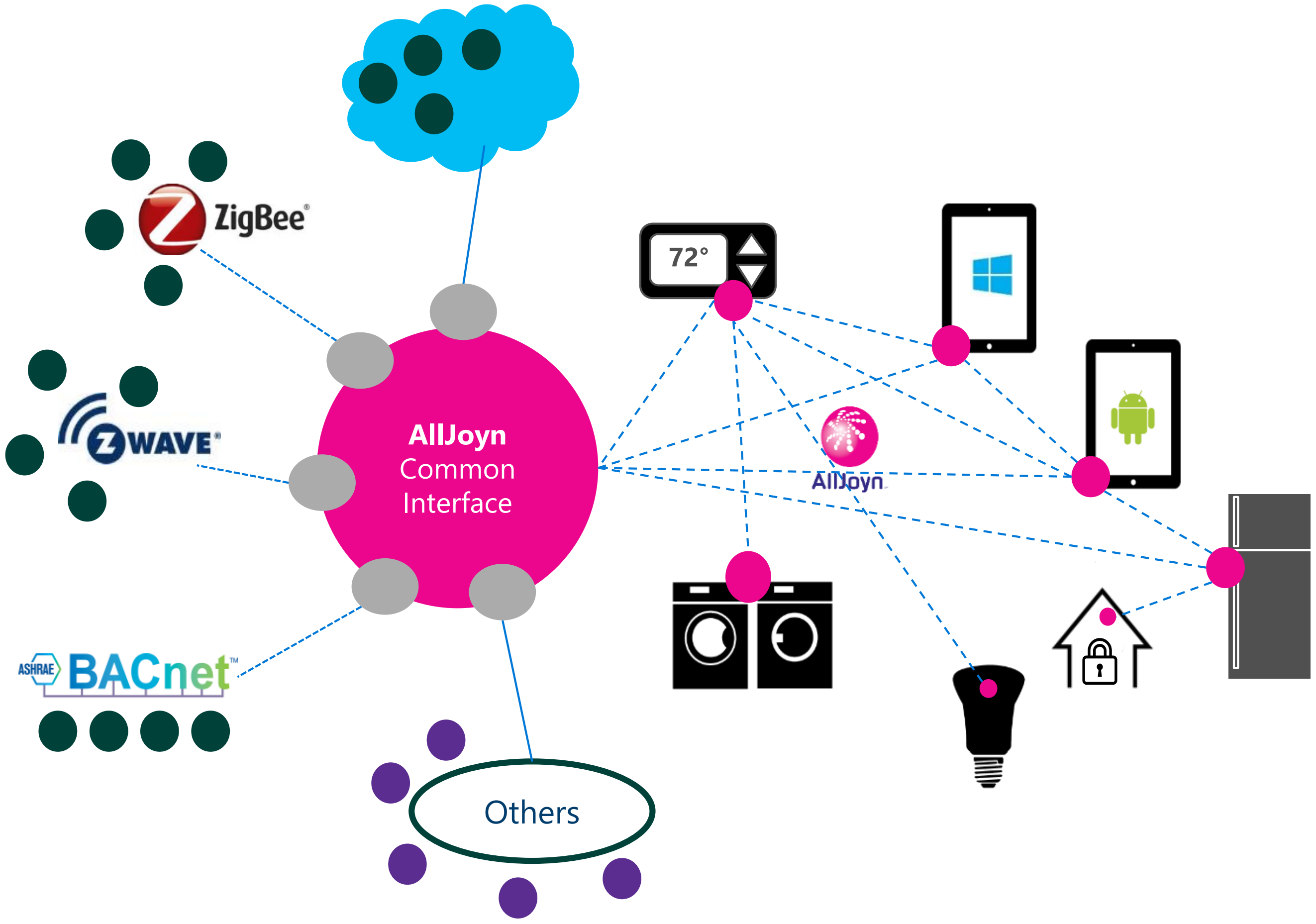
AllJoyn as the Common Language

AllJoyn Node ●

AllJoyn Thin Client •

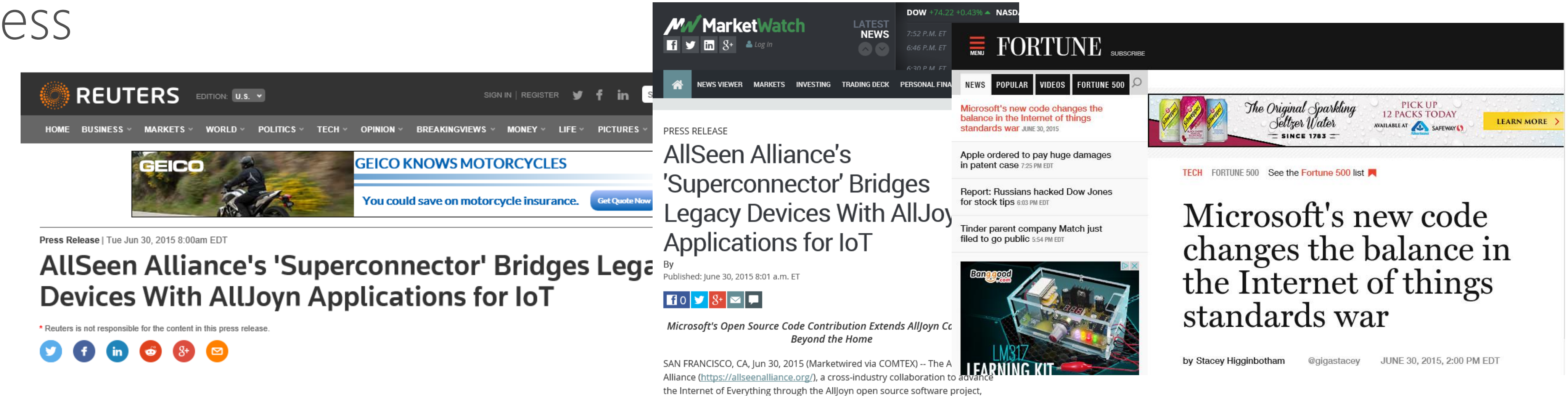
Other Proximal or Cloud Devices ●

Device System Bridge ●



Device System Bridge OSS Project

- March 2015: The Device System Bridge (DSB) project was introduced to the AllSeen Alliance. It was accepted as official project and hosted under the Gateway Working Group.
- April 2015: First code contribution from Microsoft. The code is managed as Open Source at the AllSeen Alliance.
  Sample code release and on-stage demo at //Build 2015.
- June 2015: Public announcement of the project was widely picked up by the press



Press Release | Tue Jun 30, 2015 8:00am EDT

AllSeen Alliance's 'Superconnector' Bridges Lega
Devices With AllJoyn Applications for IoT

* Reuters is not responsible for the content in this press release.

PRESS RELEASE

AllSeen Alliance's 'Superconnector' Bridges Legacy Devices With AllJoy Applications for IoT

By
Published: June 30, 2015 8:01 a.m. ET

Microsoft's Open Source Code Contribution Extends AllJoyn Co Beyond the Home

SAN FRANCISCO, CA, Jun 30, 2015 (Marketwired via COMTEX) -- The A Alliance (https://allseenalliance.org/), a cross-industry collaboration to advance the Internet of Everything through the AllJoyn open source software project,

Microsoft's new code changes the balance in the Internet of things standards war JUNE 30, 2015

Apple ordered to pay huge damages in patent case 7:25 PM EDT

Report: Russians hacked Dow Jones for stock tips 6:03 PM EDT

Tinder parent company Match just filed to go public 5:54 PM EDT

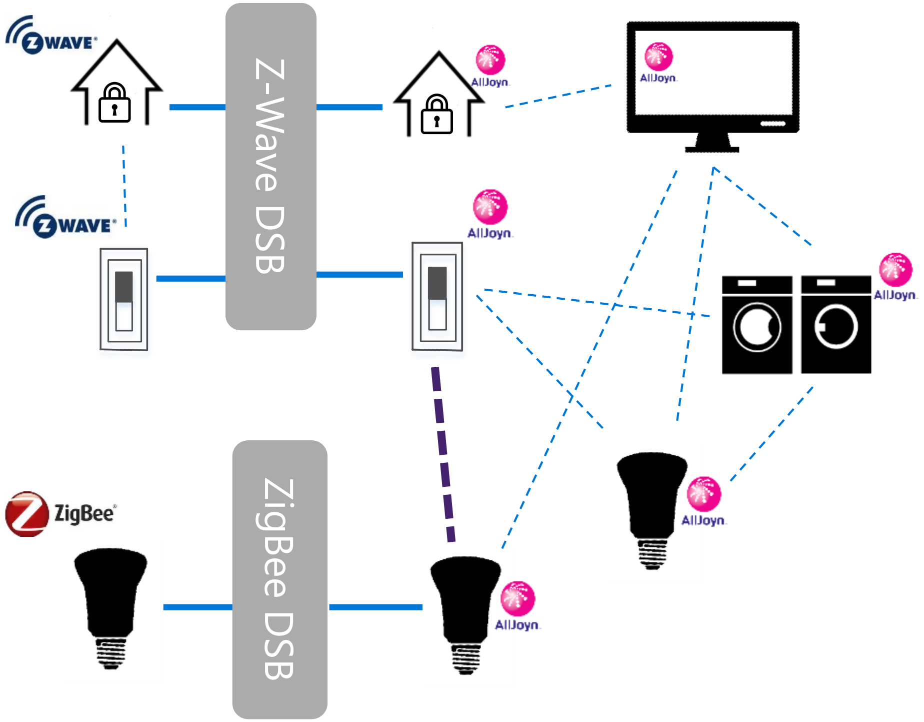
TECH FORTUNE 500 See the Fortune 500 list

Microsoft's new code changes the balance in the Internet of things standards war

by Stacey Higginbotham @gigastacey JUNE 30, 2015, 2:00 PM EDT

Enabling non-AllJoyn devices

- Setup DSBs with access to both the AllJoyn and non-AllJoyn network
- DSB creates virtual devices for each non-AllJoyn device on the AllJoyn bus
- Virtual devices can communicate with any AllJoyn
- Different non-AllJoyn systems can communicate with each other through AllJoyn
- No changes needed in AllJoyn or non-AllJoyn devices



DSB Architecture

OSS Z-Wave example

Bridge

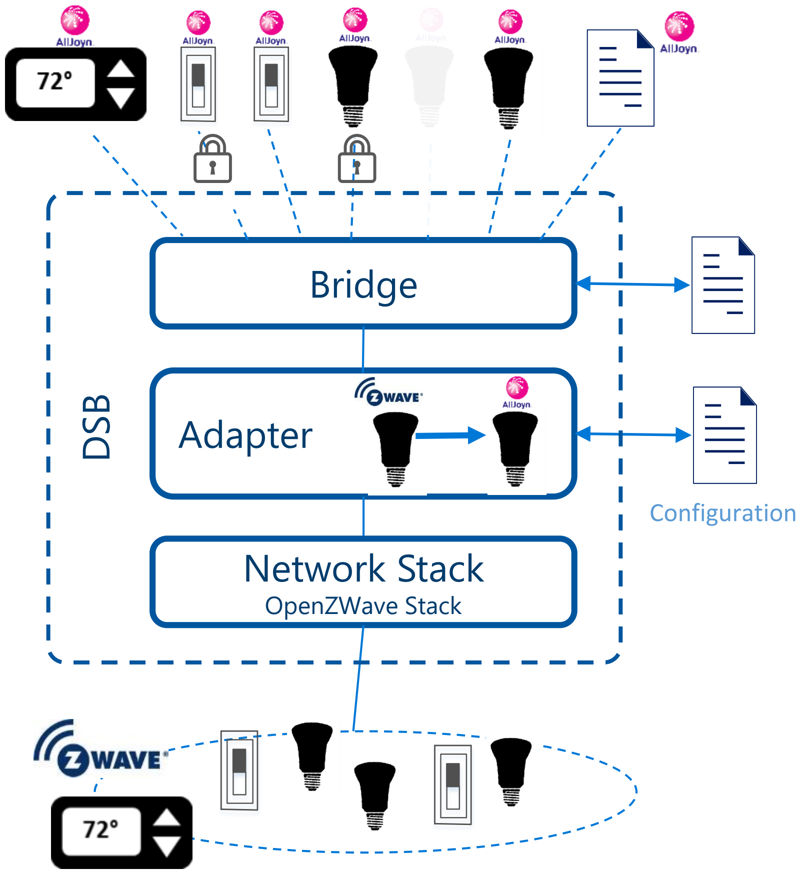
- Represents each internal device object as AllJoyn device, separate bus attachment for each device
- Devices are dynamically added to or removed from the AllJoyn bus
- Configuration manages device visibility and security
- Creates bus attachment for bridge and adapter configuration interface
- Bridge code is agnostic to internal device types and reusable for any type of DSB

Adapter

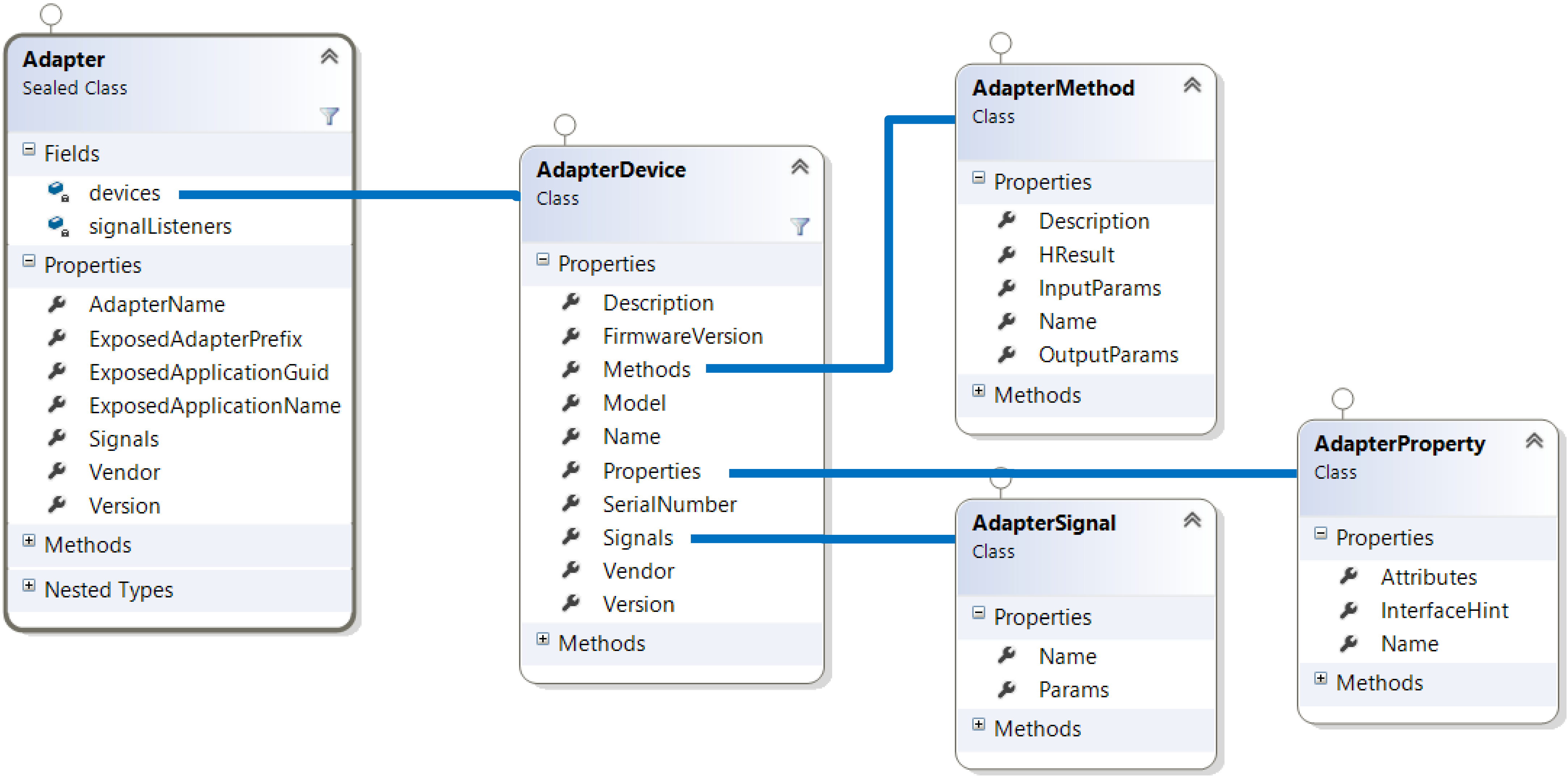
- Instantiates and manages virtual devices on behalf of each device from the non-AllJoyn network
- Translates device schemas into internal device objects
- Manages network resources, e.g. access keys, credentials

Network Access Stack

- Access to non-AllJoyn Network specific , e.g. Z-Wave stack



Adapter classes



Interfaces

Bus Interfaces

- Each device is a separate bus attachment and therefore has its own About and Icon interface. The content of both are coming from the Adapter
- Bridge will generate interface for each internal device object by mapping properties, attributes, methods and signals of the internal device object

Interface Names

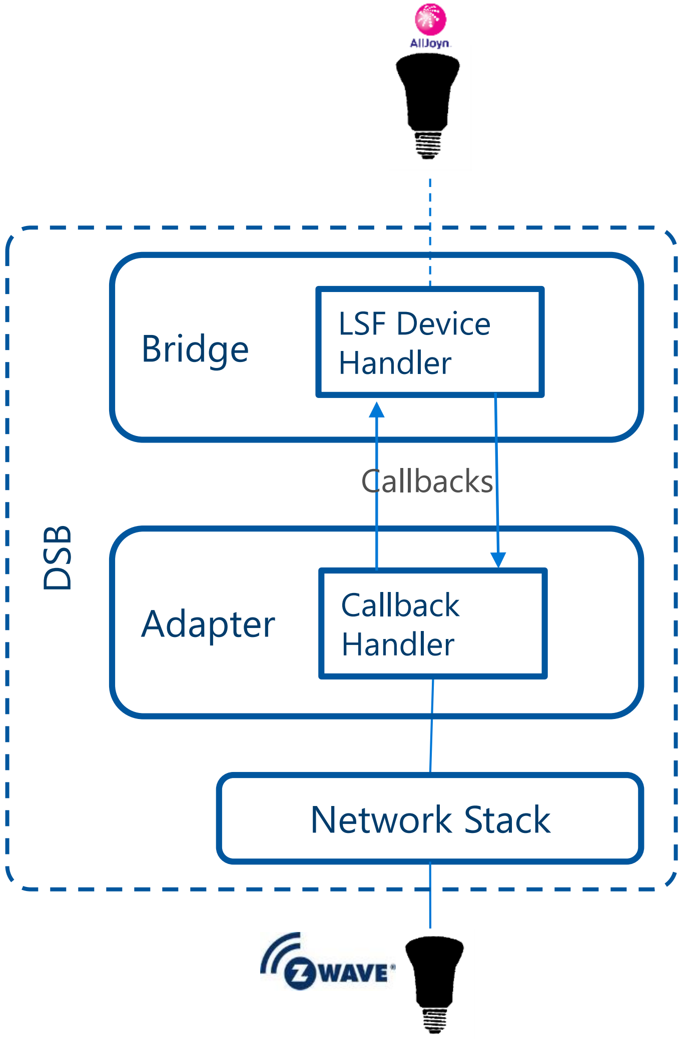
- The AllJoyn Interface name can be specified in the **`InterfaceHint`** property in the **`IAdapter`** interface
- If **`InterfaceHint`** is not specified then Interface names are created automatically from information in the **`IAdapter`**
- `<ExposedAdapterPrefix>.<AdapterName>.Interface_1`
- *e.g.* `com.microsoft.ZWaveAdapter.Interface_1`

Special handlers

- AllJoyn specifies several base services and standard interfaces frameworks such as **LSF, HAE or Control Panel**. DSB can exposes those with special handlers.

- LSF and Control Panel handler code is in the bridge

- Callback functions are provided for the adapter to serve requests from the special handlers in the Bridge



Device Management

AllJoyn Management CSP

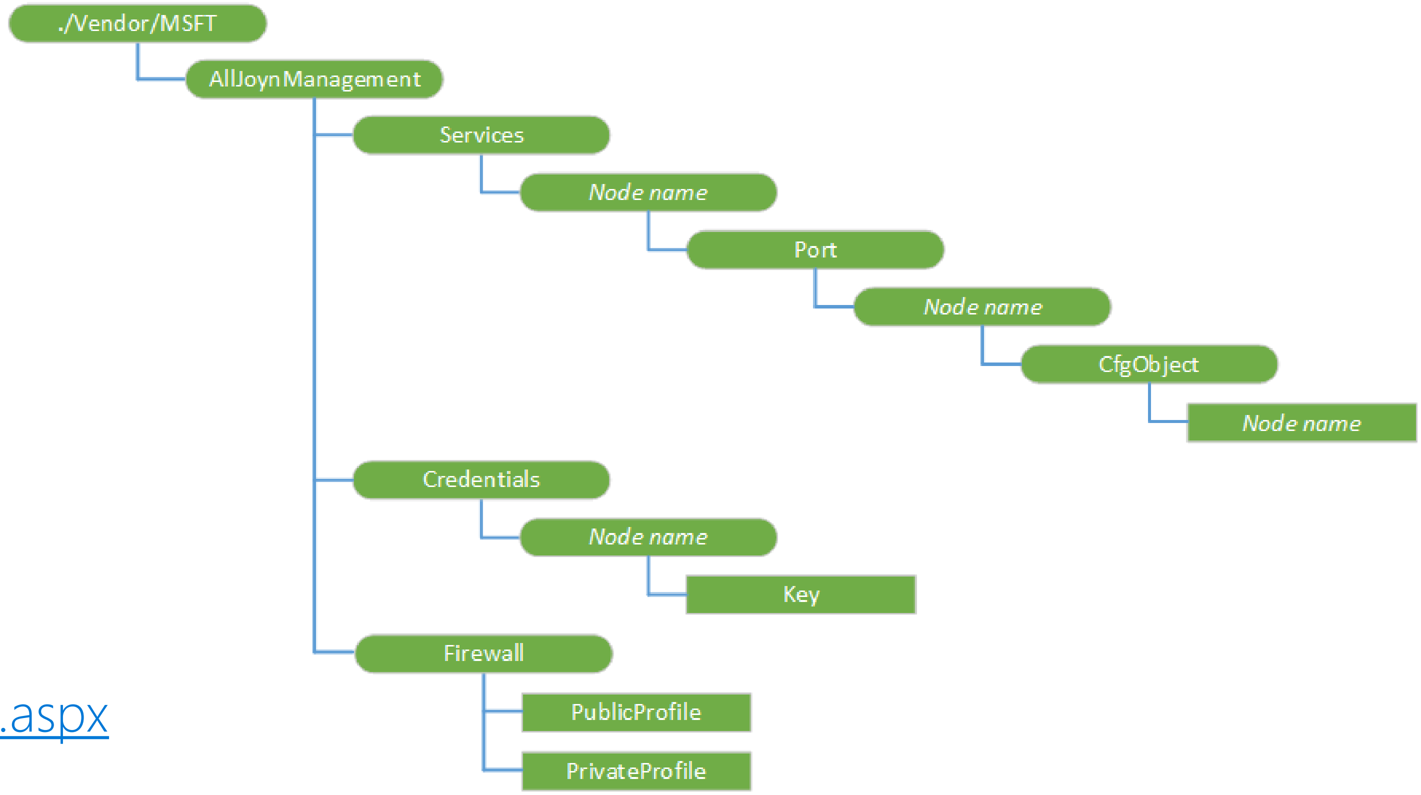
Windows 10 IOT Core allows to manage AllJoyn devices and configurations via `AllJoynManagement` CSP

CSP capabilities

- AllJoyn Device Information
- AllJoyn Device Configurations
- AllJoyn object Credential
- AllJoyn Firewall settings

Documentation

https://msdn.microsoft.com/en-us/library/windows/hardware/mt157022(v=vs.85).aspx



CSP: AllJoyn Device Information

Device Information

- Query the **`About`** interface of AllJoyn bus attachments, CSP exposes all data fields that are required by the About interface including the `HardwareVersion`
- Path `./Vendor/MSFT/AllJoynManagement/Services`

Device Configurations

- Read and write configurations to AllJoyn bus attachments that expose the `com.microsoft.alljoynmanagement.Config` interface
- Path`./Vendor/MSFT/AllJoynManagement/Services/{Service}/Port/{Port}/CfgObject/{ObjectPath}`

CSP: AllJoyn Credentials and Firewall Settings

Credentials

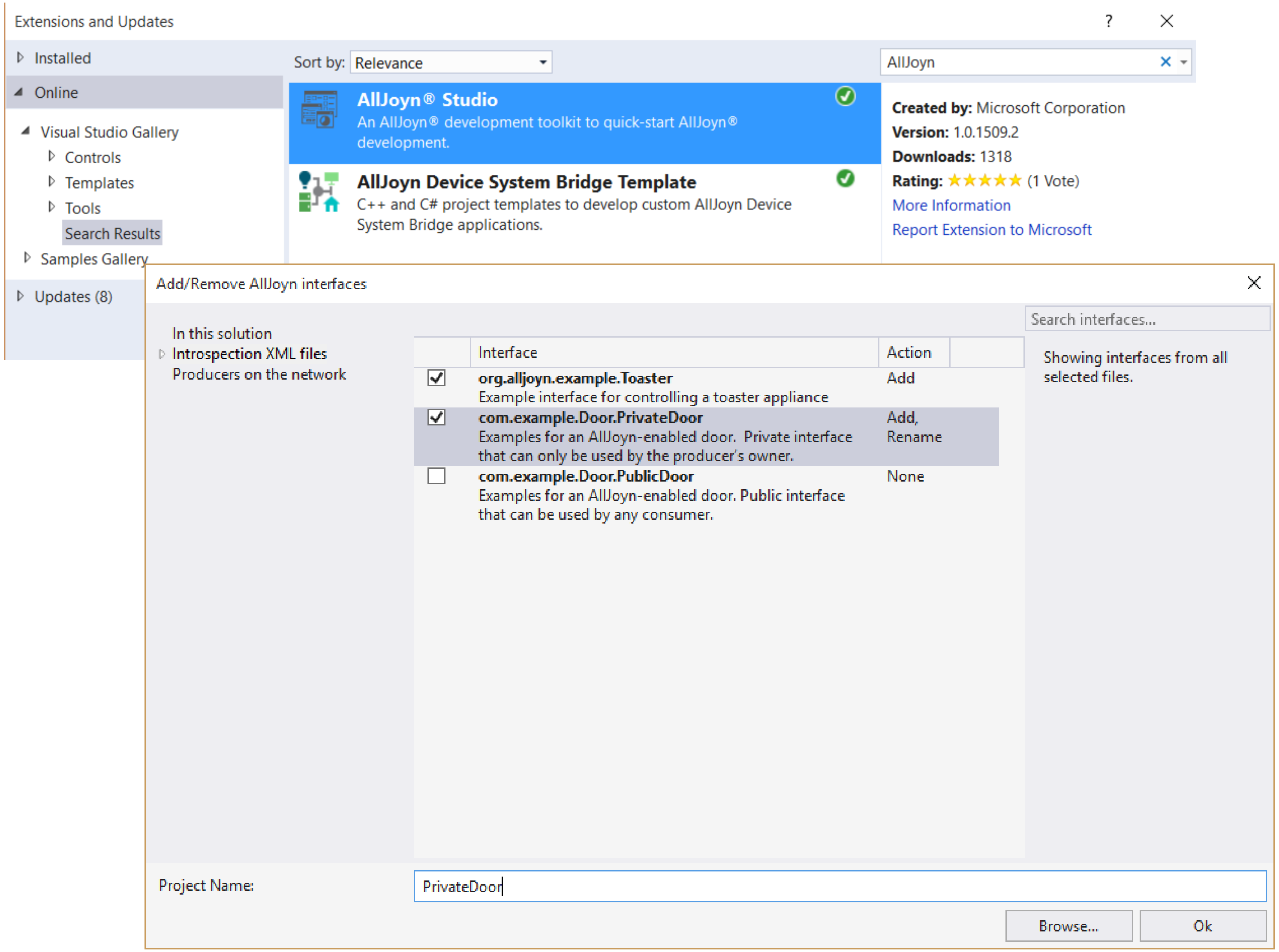
- Set credentials for each AllJoyn device that requires authentication
- Path `./Vendor/MSFT/AllJoynManagement/Credentials/Node name`

Firewall

- Enable or disable the AllJoyn router service (AJRouter.dll) for public network profile
- Path `./Vendor/MSFT/AllJoynManagement/Firewall/PublicProfile`

Development Resources

AllJoyn Studio



Extensions and Updates

Sort by: Relevance

AllJoyn

AllJoyn® Studio
An AllJoyn® development toolkit to quick-start AllJoyn® development.

AllJoyn Device System Bridge Template
C++ and C# project templates to develop custom AllJoyn Device System Bridge applications.

Created by: Microsoft Corporation
Version: 1.0.1509.2
Downloads: 1318
Rating: ★★★★★ (1 Vote)
More Information
Report Extension to Microsoft

Add/Remove AllJoyn interfaces

	Interface	Action
☑	**org.alljoyn.example.Toaster** Example interface for controlling a toaster appliance	Add
☑	**com.example.Door.PrivateDoor** Examples for an AllJoyn-enabled door. Private interface that can only be used by the producer's owner.	Add, Rename
□	**com.example.Door.PublicDoor** Examples for an AllJoyn-enabled door. Public interface that can be used by any consumer.	None

Showing interfaces from all selected files.

Project Name: PrivateDoor

- XML
  - Query devices on network
  - Upload from file

- Code Generation
- Project Management

Where to find …

Go to **Visual Studio Gallery** and search for **"AllJoyn"**

DSB

Getting started

- Go to WindowsOnDevcies.com and search for “Device System Bridge” or directly to http://ms-iot.github.io/content/en-US/win10/AllJoyn.htm

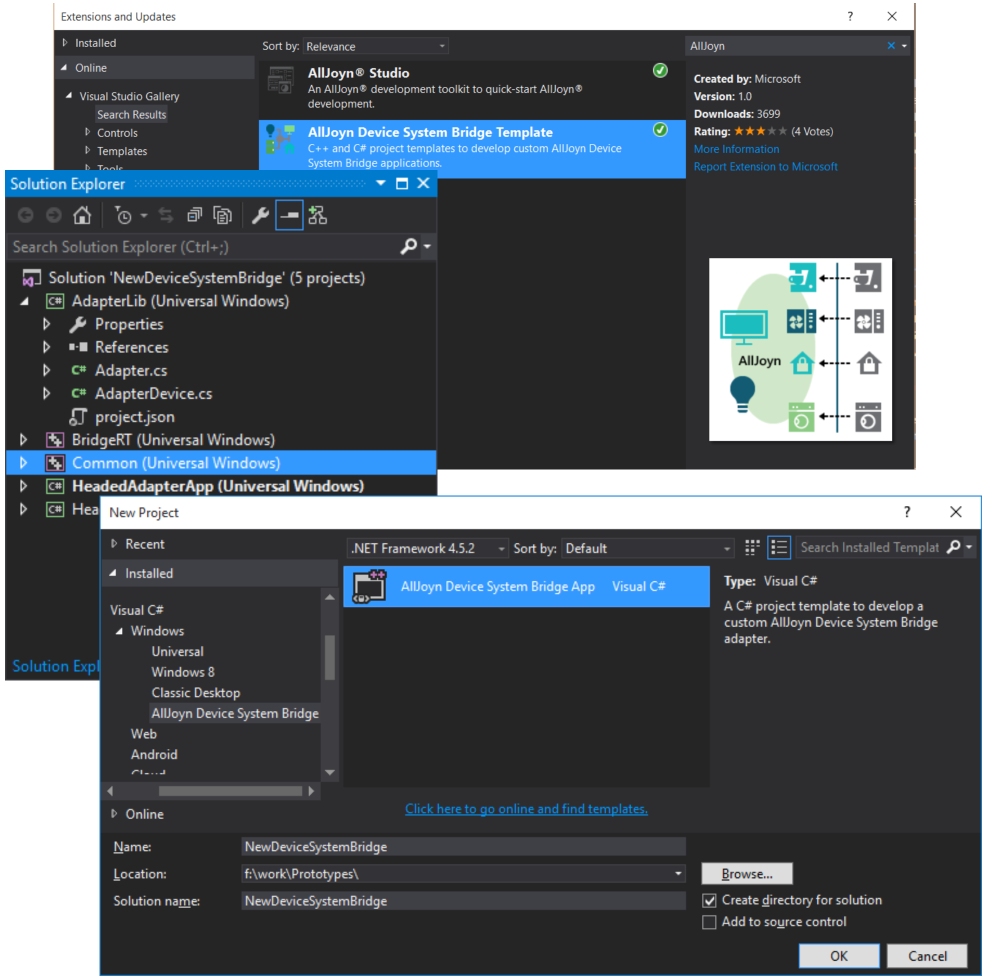
Repository

- All DSB code is available at the AllSeen Alliance GIT: git.allseenalliance.org/cgit/dsb.git
- Supported platform: Standard AllJoyn client on Windows 10

Samples

- Mock DSB Tutorial and Sample
- Z-Wave DSB Tutorial and Sample
- ZigBee DSB Tutorial and Sample (soon, keep checking)
- Nest DSB Tutorial and Sample (soon, keep checking)
- GPIO DSB Tutorial
- BACnet DSB Sample

Visual Studio DSB Template



AllJoyn® Studio
An AllJoyn® development toolkit to quick-start AllJoyn® development.

AllJoyn Device System Bridge Template
C++ and C# project templates to develop custom AllJoyn Device System Bridge applications.

Created by: Microsoft
Version: 1.0
Downloads: 3699
Rating: ★★★☆☆ (4 Votes)
More Information
Report Extension to Microsoft

Solution 'NewDeviceSystemBridge' (5 projects)
- AdapterLib (Universal Windows)
  - Properties
  - References
  - Adapter.cs
  - AdapterDevice.cs
  - project.json
- BridgeRT (Universal Windows)
- Common (Universal Windows)
- HeadedAdapterApp (Universal Windows)
- Hea

AllJoyn Device System Bridge App — Visual C#

Type: Visual C#

A C# project template to develop a custom AllJoyn Device System Bridge adapter.

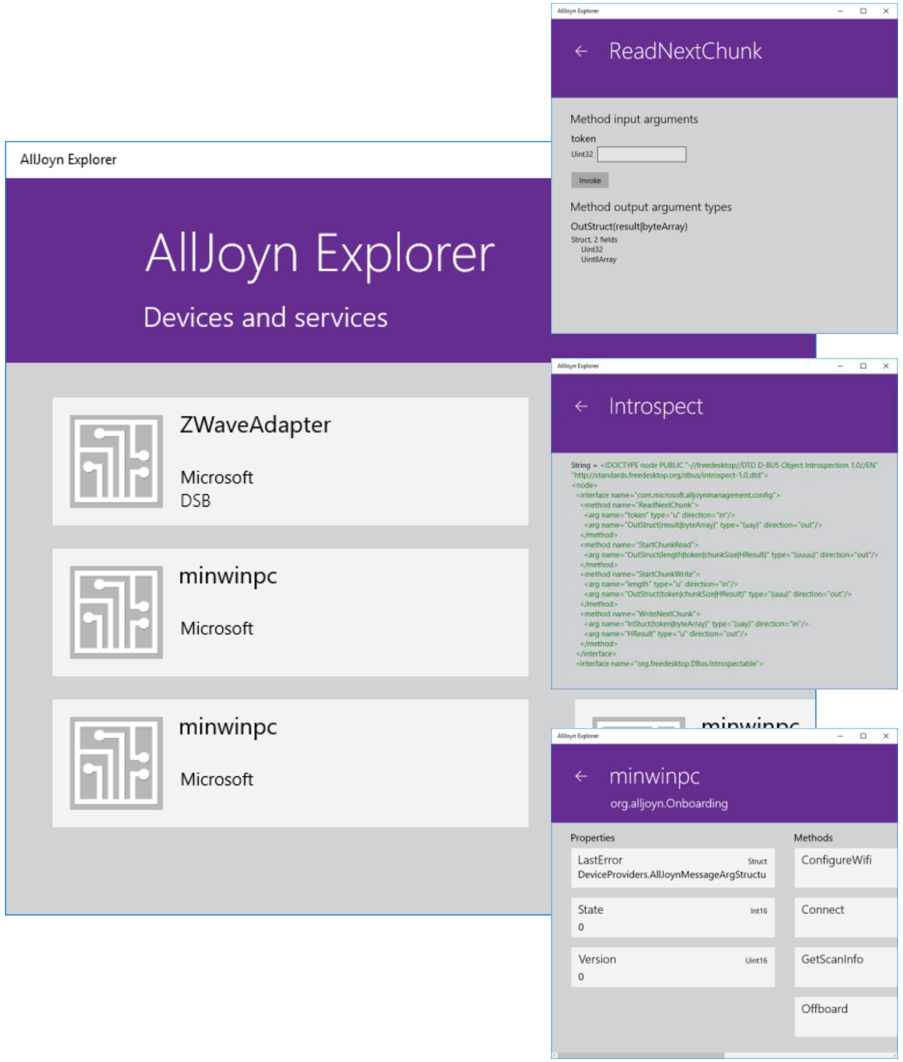
Name: NewDeviceSystemBridge
Location: f:\work\Prototypes\
Solution name: NewDeviceSystemBridge
☑ Create directory for solution
☐ Add to source control

Visual Studio Extension to build Device System Bridge UWP Applications

- Managed (C#) or Native (C++/CX)
- Headed or Headless UWP Application

Go to **Visual Studio Gallery** and search for **DSB** or Download here

AllJoyn Explorer



AllJoyn Explorer

Devices and services

ZWaveAdapter
Microsoft
DSB

minwinpc
Microsoft

minwinpc
Microsoft

ReadNextChunk

Method input arguments

token
Uint32

Method output argument types

OutStruct(result|byteArray)
Struct, 2 fields
Uint32
Uint8Array

Introspect

```
String = <!DOCTYPE node PUBLIC "-//freedesktop//DTD D-BUS Object Introspection 1.0//EN"
"http://standards.freedesktop.org/dbus/introspect-1.0.dtd">
<node>
 <interface name="com.microsoft.alljoynmanagement.config">
  <method name="ReadNextChunk">
   <arg name="token" type="u" direction="in"/>
   <arg name="OutStruct(result|byteArray)" type="(uay)" direction="out"/>
  </method>
  <method name="StartChunkRead">
   <arg name="OutStruct(length|token|chunkSize|HResult)" type="(uuuu)" direction="out"/>
  </method>
  <method name="StartChunkWrite">
   <arg name="length" type="u" direction="in"/>
   <arg name="OutStruct(token|chunkSize|HResult)" type="(uuu)" direction="out"/>
  </method>
  <method name="WriteNextChunk">
   <arg name="InStuct(token|byteArray)" type="(uay)" direction="in"/>
   <arg name="HResult" type="u" direction="out"/>
  </method>
 </interface>
 <interface name="org.freedesktop.DBus.Introspectable">
```

minwinpc
org.alljoyn.Onboarding

Properties		Methods
LastError DeviceProviders.AllJoynMessageArgStructu	Struct	ConfigureWifi
State 0	Int16	Connect
Version 0	Uint16	GetScanInfo
		Offboard

Windows Application to explore and interact with devices on the AllJoyn bus

- Enumerate servers, list interfaces and bus objects
- Read and write properties
- Call methods
- Subscribe to signals

Available in Store soon, search for “AJX”

Demo

Speaker name

Key Takeaways

AllJoyn is build into Windows

Windows 10 has AllJoyn build in it's core and provides powerful tools to create AllJoyn producers and consumers

DSBs enable non-AllJoyn devices

Non-AllJoyn device systems can be enabled for the AllJoyn ecosystem via DSBs. DSBs create virtual devices for each of the non-AllJoyn devices.

AllJoyn DM is build into Windows IoT Core

AllJoyn devices can be inventoried and configured via device management systems using the build in AllJoyn Configuration Service Provider

Call to action

- Join the AllSeen Alliance Gateway Mailing list for updates, suggestions and questions
  - ➢ allseen-gateway@lists.allseenalliance.org (Subscribe)
- Contribute new capabilities and fixes to the Device System Bridge Open Source Project at the AllSeen Alliance

Please Complete An Evaluation

Your input is important!

Microsoft