



# winhec

Shenzhen 2015



# AllJoyn Device System Bridge

Anders Liu  
Partner Enablement Team - Shenzhen

# Introduction and Agenda

## Session Agenda:

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

AllJoyn Support in Windows 10

# AllJoyn in Windows 10

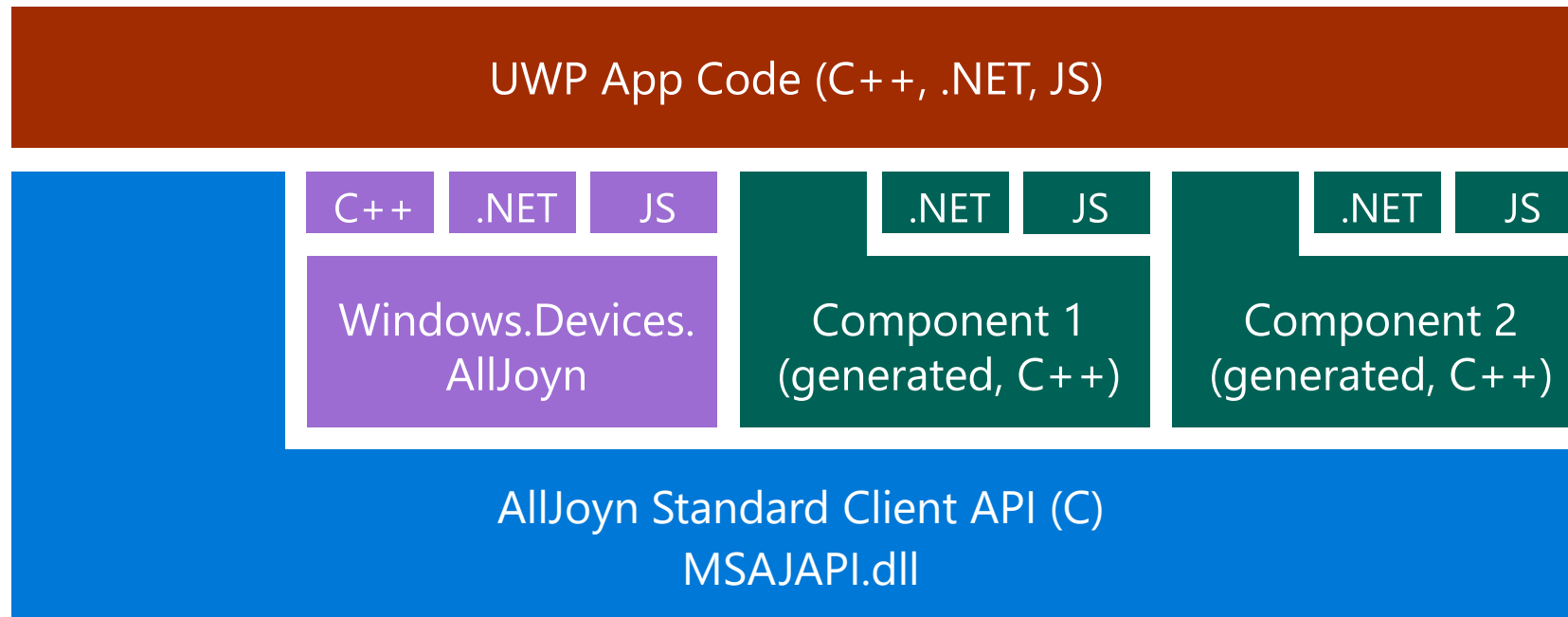
<b>Integrated AllJoyn Runtime</b>	Servicing, reduced code size
<b>Router Node Service</b>	Optimized performance, full integration
<b>C and UWP APIs</b>	Reduced code size, integrated with Windows SDK
<b>Visual Studio and SDK Integration</b>	AllJoyn Studio accelerates development for Windows
<b>Samples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

# Windows 10 AllJoyn UWP and OneCore

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



# AllJoyn UWP Apps

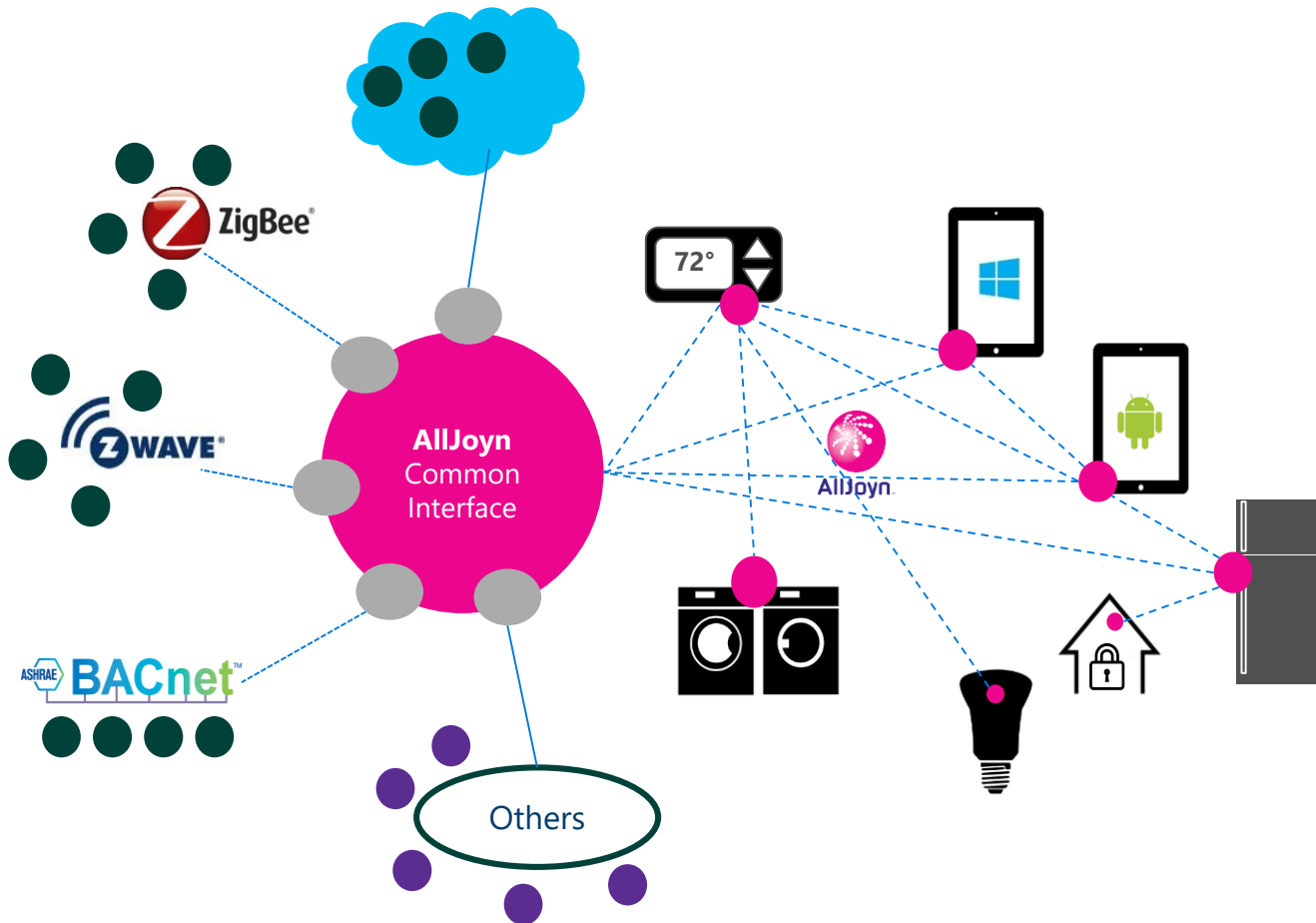


# AllJoyn Device System Bridge



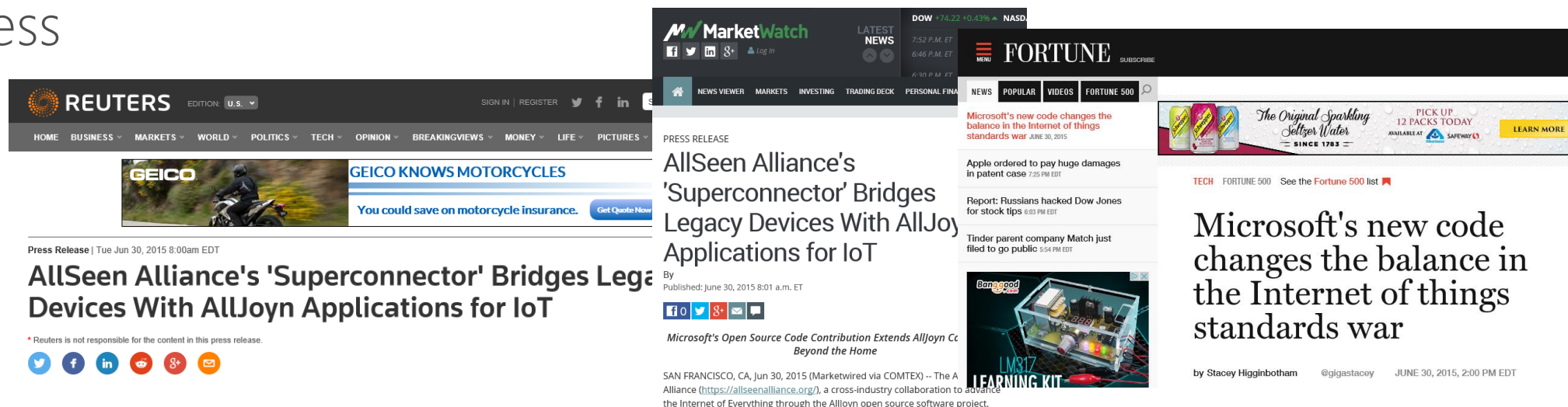
# AllJoyn as the Common Language

- AllJoyn Node ●
- AllJoyn Thin Client ●
- Other Proximal or Cloud Devices ●
- Device System Bridge** ●



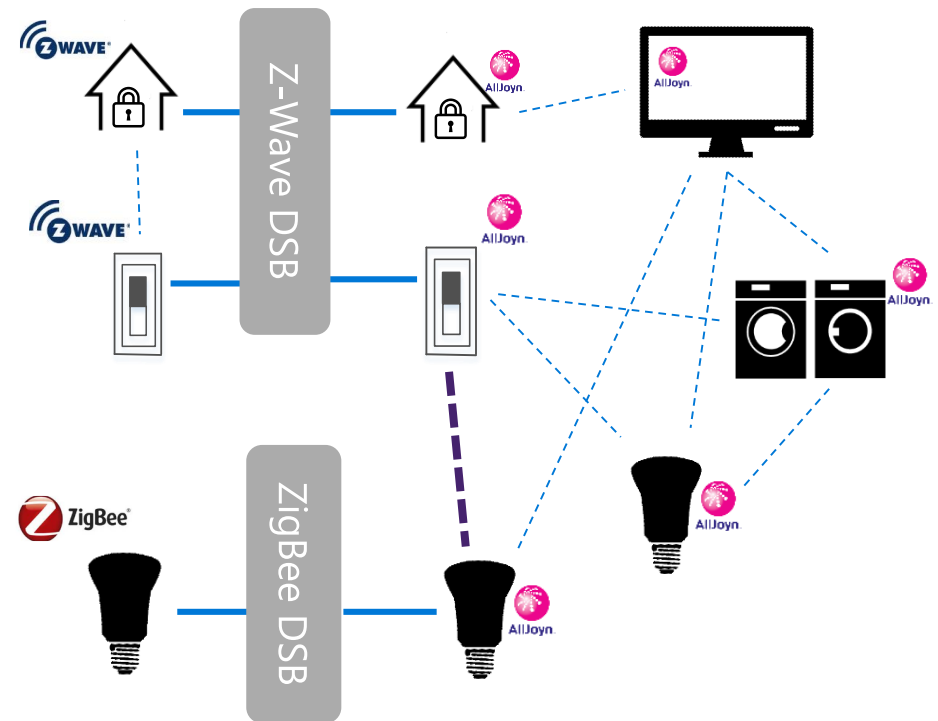
# Device System Bridge OSS Project

- **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



# Enabling non-AllJoyn devices

- Setup DSBs with access to both the AllJoyn and non-AllJoyn network
- DSB creates virtual devices 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

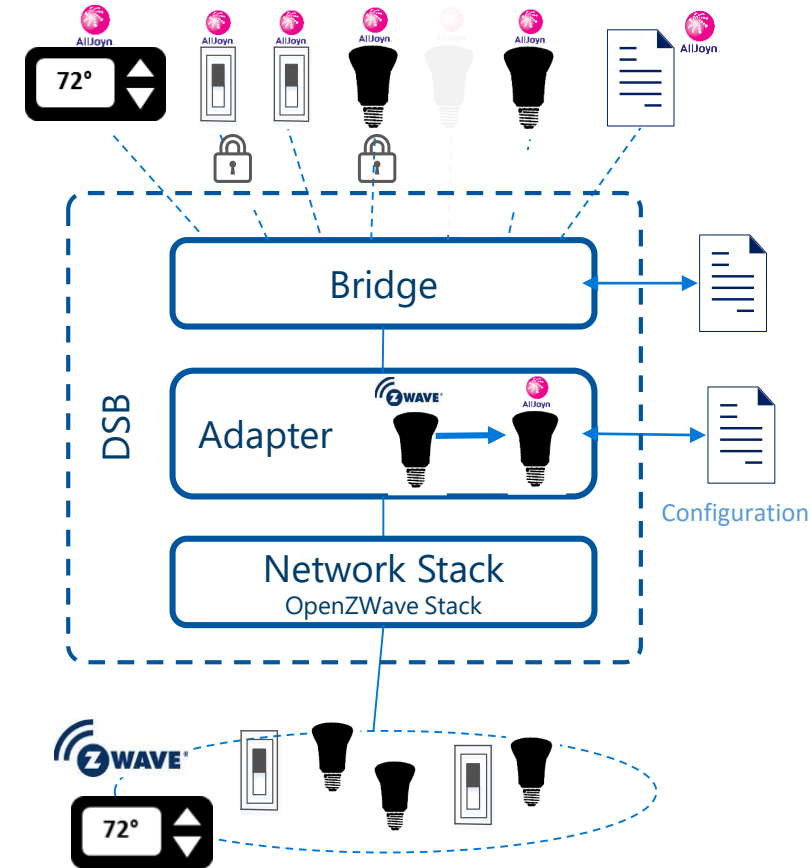
- Represents each internal device object as AllJoyn device, separate bus attachment for each device
- Devices are dynamically added to or removed from the AllJoyn bus
- Configuration manages device visibility and security
- Creates bus attachment for bridge and adapter configuration interface
- Bridge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

### Adapter

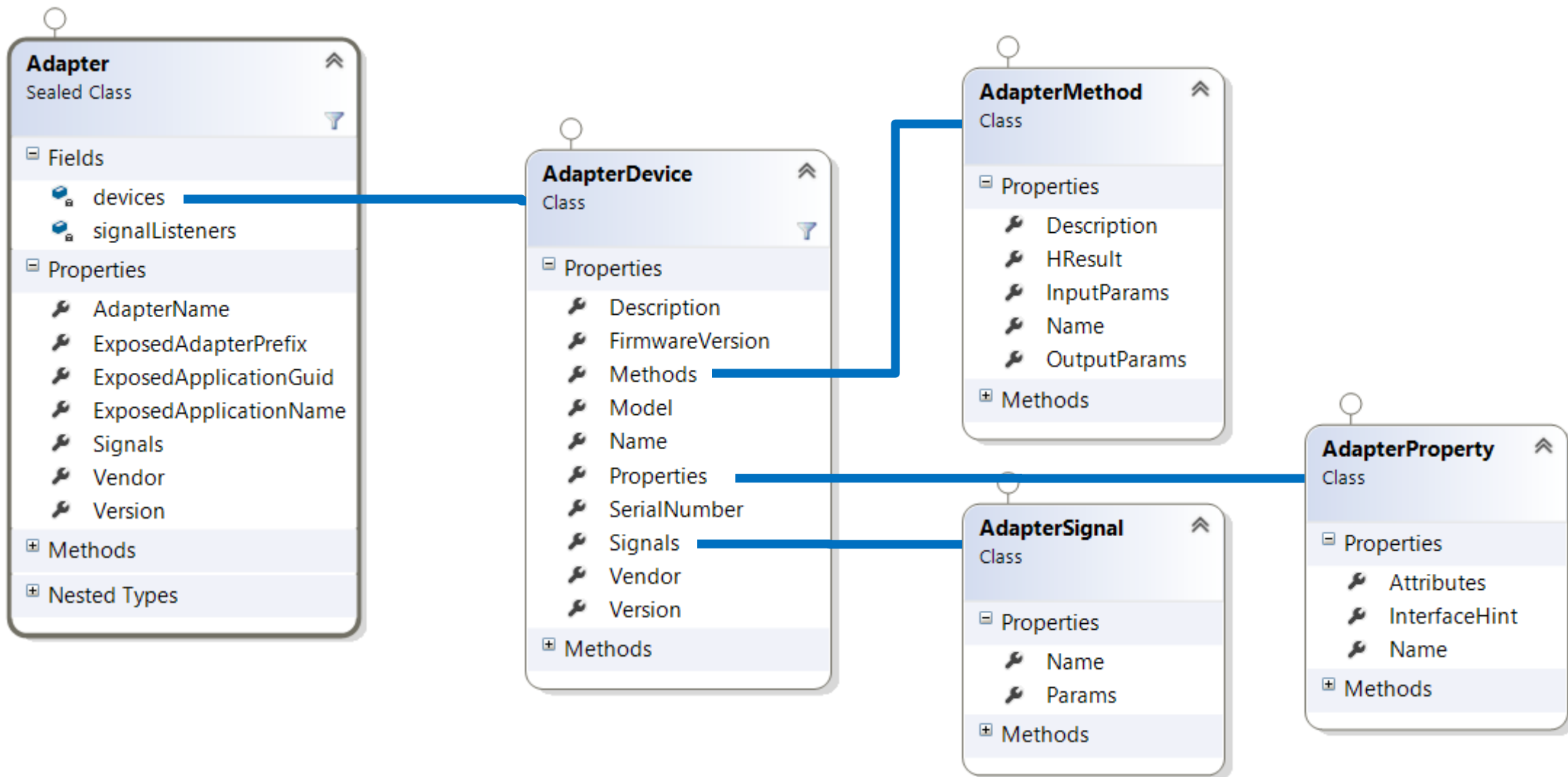
- Instantiates and manages virtual devices on behalf of each device from the non-AllJoyn network
- Translates device schemas into internal device objects
- Manages network resources, e.g. access keys, credentials

### Network Access Stack

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



# Adapter classes



# Interfaces

## Bus Interfaces

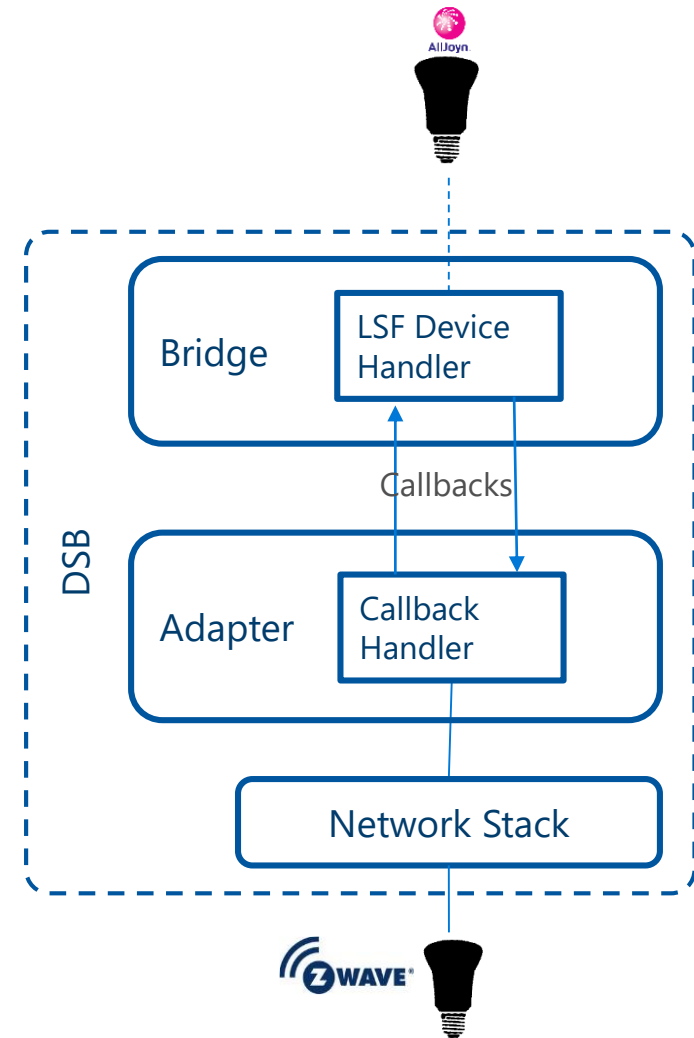
- Each device is a separate bus attachment and therefore has its own **About** and **Icon** interface. The content of both are coming from the Adapter
- Bridge will generate interface for each internal device object by mapping properties, attributes, methods and signals of the internal device object

## 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, or Control Panel. DSB can expose those with special handlers.
- LSF and Control Panel handler code is in the bridge and callbacks are provided for the adapter to handle requests



# Development Resources



# AllJoyn Studio

The screenshot shows the Visual Studio 'Extensions and Updates' window. The search results for 'AllJoyn' are displayed. Two extensions are listed:

- AllJoyn® Studio**: An AllJoyn® development toolkit to quick-start AllJoyn® development. Created by: Microsoft Corporation. Version: 1.0.1509.2. Downloads: 1318.
- AllJoyn Device System Bridge Template**: C++ and C# project templates to develop custom AllJoyn Device System Bridge applications. Rating: ★★★★★ (1 Vote). More Information. Report Extension to Microsoft.

Below the extensions window, a dialog titled 'Add/Remove AllJoyn interfaces' is open. It shows a list of interfaces for the current solution:

Interface	Action
<input checked="" type="checkbox"/> <b>org.alljoyn.example.Toaster</b> Example interface for controlling a toaster appliance	Add
<input checked="" type="checkbox"/> <b>com.example.Door.PrivateDoor</b> Examples for an AllJoyn-enabled door. Private interface that can only be used by the producer's owner.	Add, Rename
<input type="checkbox"/> <b>com.example.Door.PublicDoor</b> Examples for an AllJoyn-enabled door. Public interface that can be used by any consumer.	None

The 'Project Name' field at the bottom of the dialog is set to 'PrivateDoor'. Buttons for 'Browse...' and 'Ok' are visible at the bottom right.

- 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](http://git.allseenalliance.org/cgit/dsb.git)
- Supported platform: Standard client on Windows 10

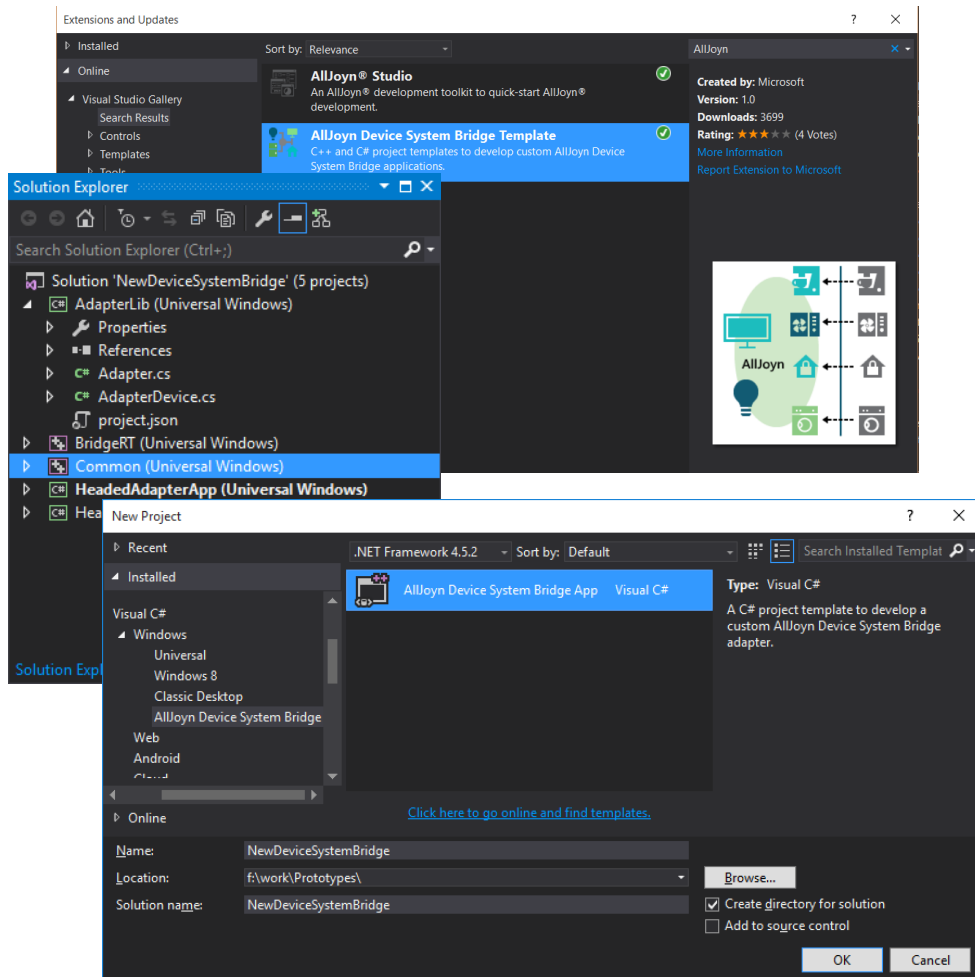
## Samples

- [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](#)

FAQ:

[msftziot@microsoft.com](mailto:msftziot@microsoft.com)

# Visual Studio DSB Template

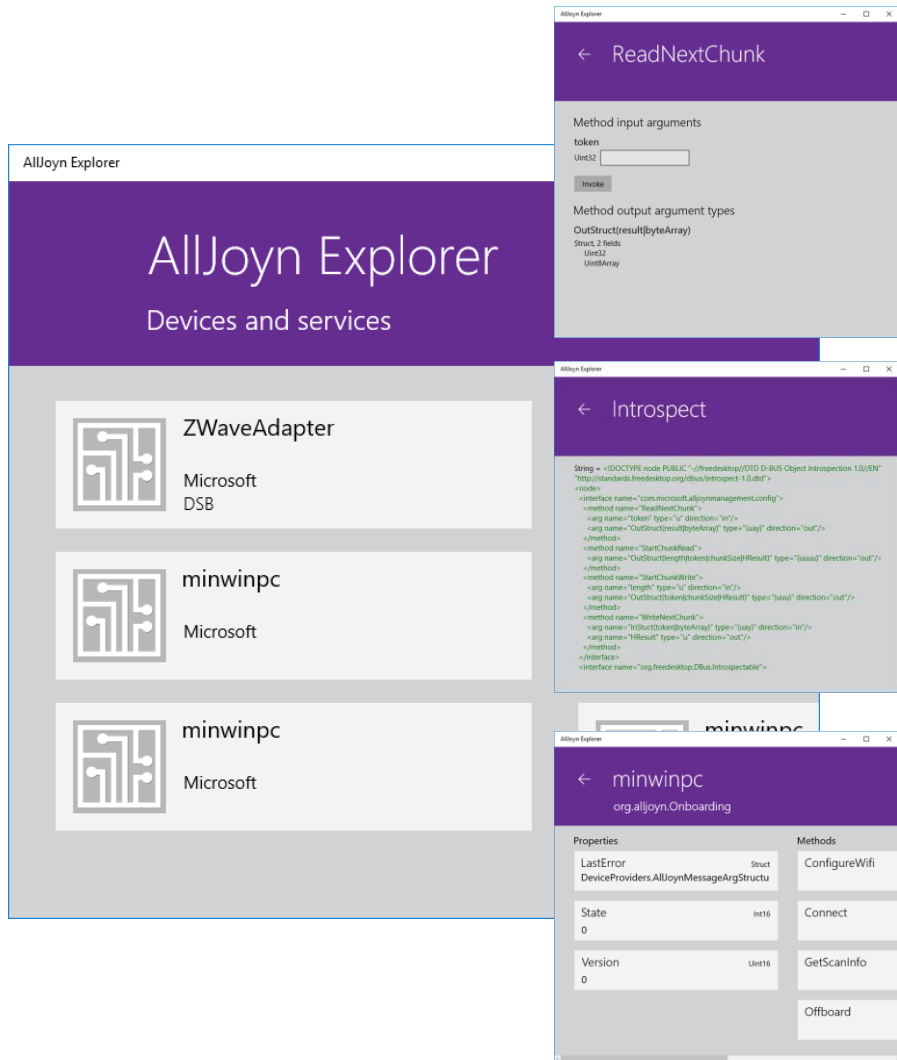


Visual Studio Extension to build Device System Bridge UWP Applications

- 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



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

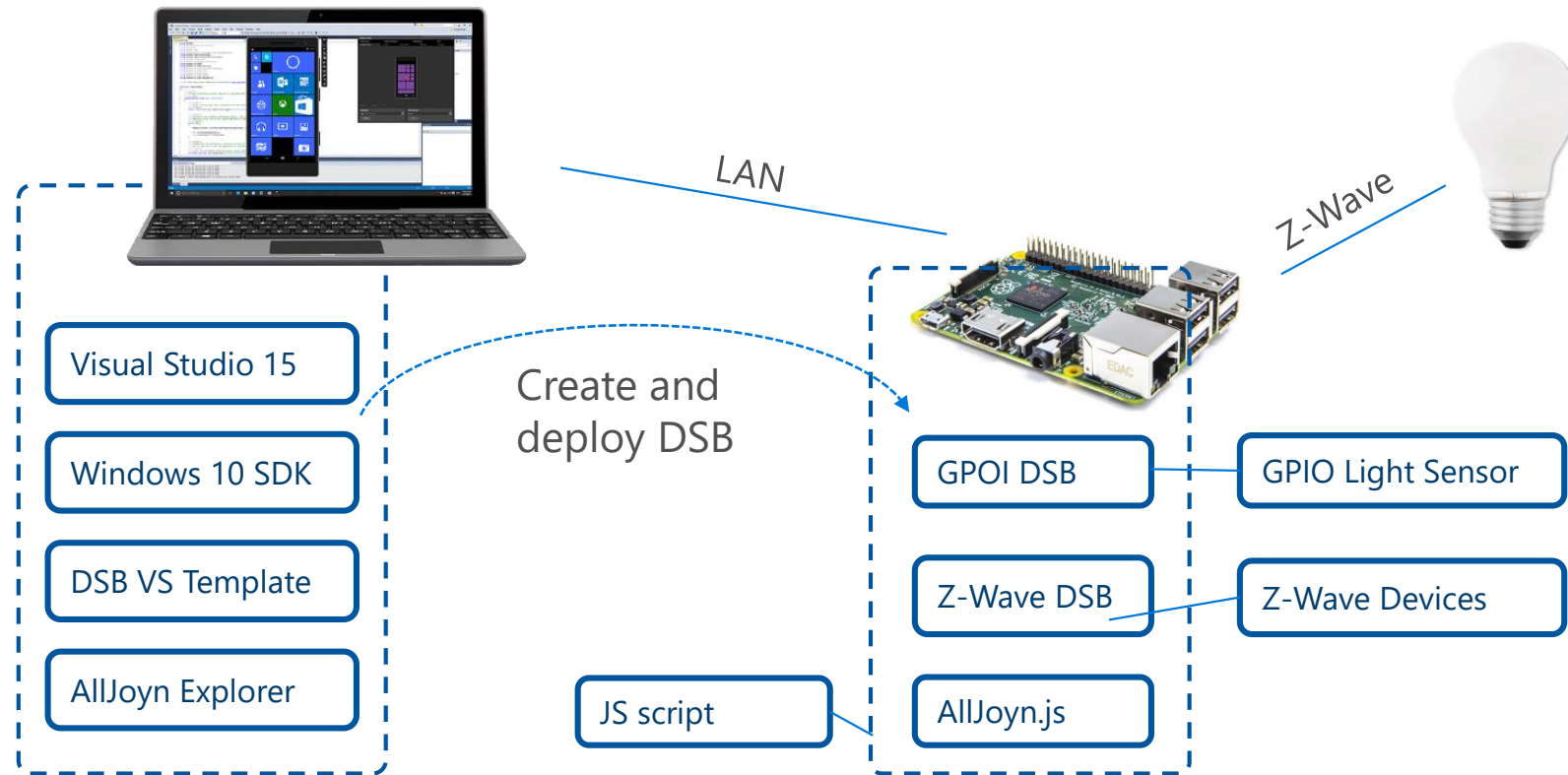
Available in Store soon, search for "AJX"

# Demo

Anders Liu

# Setup

Create a DSB in 5 min and use the device with other AllJoyn devices



# 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](mailto: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!

## Multiple ways to access Evaluation Forms:

1. CommNet stations located throughout conference venues
2. Via WinHEC app on your Windows Phone and Windows device
3. Via BYOD browser from any wired or wireless internet connection to [<link>](#)

