



## Desktop App Converter

*aka Convertire applicazioni Win32 in Universal Windows Platform con il Desktop App Converter*

Lorenzo Barbieri

@\_geniodelmale

19 LUGLIO

# Desktop apps

## Universal Windows apps

Uses the Windows 10 app model

AppX via Store or side-loaded

Capable of running on PCs, Phones, XBOX, etc.

## Classic Desktop apps ❤️

.NET Framework (WPF/WinForms) or Win32 apps

Installed using .msi technology

*Desktop to UWP converter to package as AppX*



# Getting your app to customers

- Last year we introduced .appX as the common installation package for UWP apps
  - PC, Phone, HoloLens, IOT all use the same technology
- Expanding the installation technology to be even more inclusive
  - Existing Desktop Apps
  - XBox
- Expanding where you can use .appX
  - November release enabled, by default, installing from anywhere not just the store



# modern desktop app installer



- Enables installation of .appx package without having to come from the store
- Built on Deployment platform
- Store updated so will iterate to refine experience
- **.appx must be signed with a cert from a trusted root**
  - The .ps1 script created by VS will install the certificate, this procedure don't
- Only in Windows 10 Anniversary Update

# DEMO



# Desktop App Converter



# What is Project Centennial?

## Improving the Windows platform for all developers

Convert desktop apps to the Universal Windows deployment technology

Keep your current Win32/.NET code running

Code targets Windows 10 and Windows 7 PCs

Add Universal Windows components over time

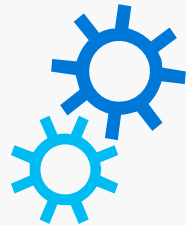
Monetize via Windows Store or other distribution of choice

# Developer Workflow

## Desktop App Converter

Converts existing desktop installer

Results in a packaged app with UWP manifest



\*Pending availability

## Test, Update and Evolve

Make compatible code changes that go into both MSI and app package

Evolve app package to take advantage of the Universal Windows Platform

## Publish and Distribute

Distribute the app package via sideloading, MDM or the Windows Store\*



DEMO  
DesktopConverter.cmd

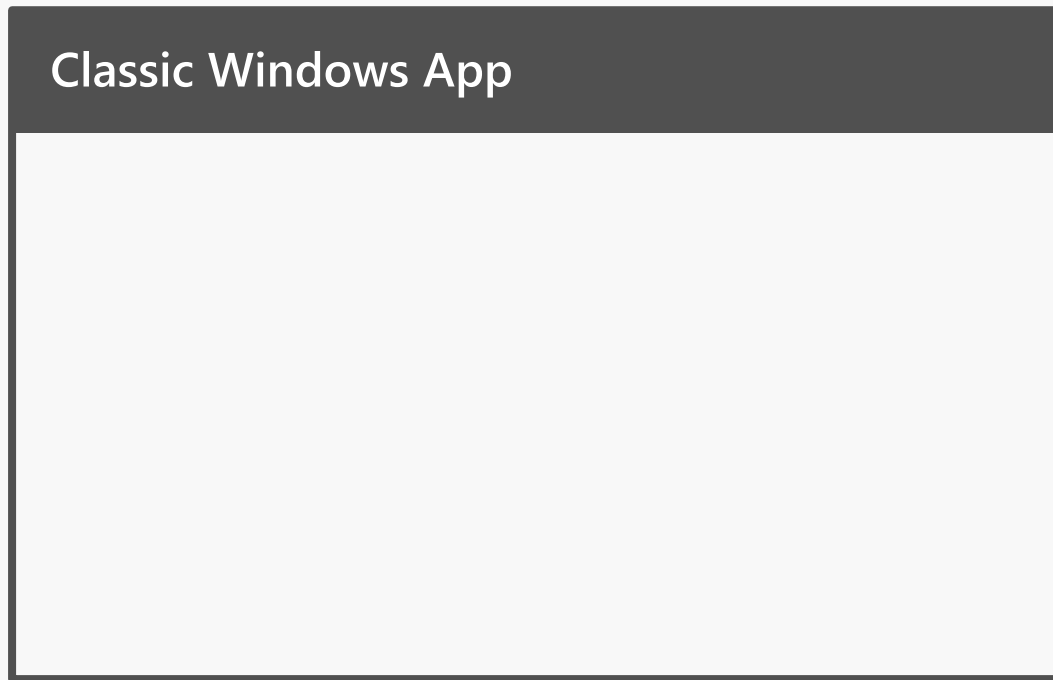
# Shopping list:

- Windows 10 AU build 14xxx+
  - 64 bit (x64) processor
  - Hardware-assisted virtualization
  - Second Level Address Translation (SLAT)
- BaselImage.wim (same build)
- Windows 10 UWP SDK 14xxx+
- DesktopConverter.cmd
- MakeAppx.exe (to iterate)
- MakeCert.exe (to create a test cert)
- SignTool.exe (to sign .appx)
- MMC (to trust the test cert)

<https://msdn.microsoft.com/en-us/windows/uwp/porting/desktop-to-uwp-run-desktop-app-converter>



# Step 0 – Win32 Desktop App



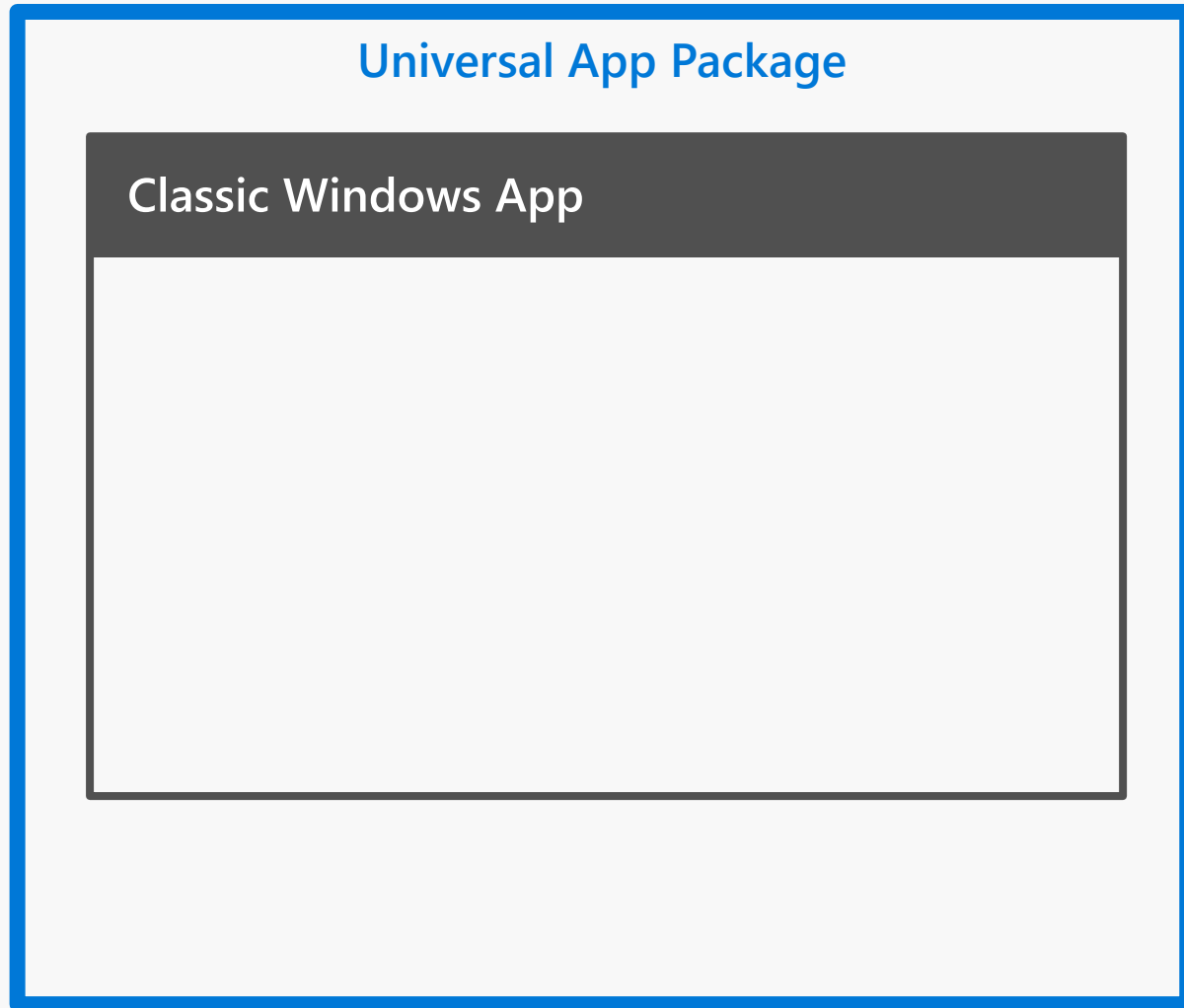
## Classic Desktop Application

Win32/.NET code base

MSI/EXE/CMD installer

Runs in a desktop process

# Step 1 – Convert the App



## Immediate Benefits:

Modern deployment technology for all apps

Seamless updates, clean uninstall

Monetize via Windows Store or other distribution of choice

Runs your current code

Small amount of code changes needed (if any)

# How does it work?

## Deployment

Uses Universal Windows Platform deployment stack

Extensions for required Win32 extension points

Shell extensions (e.g. file type associations)

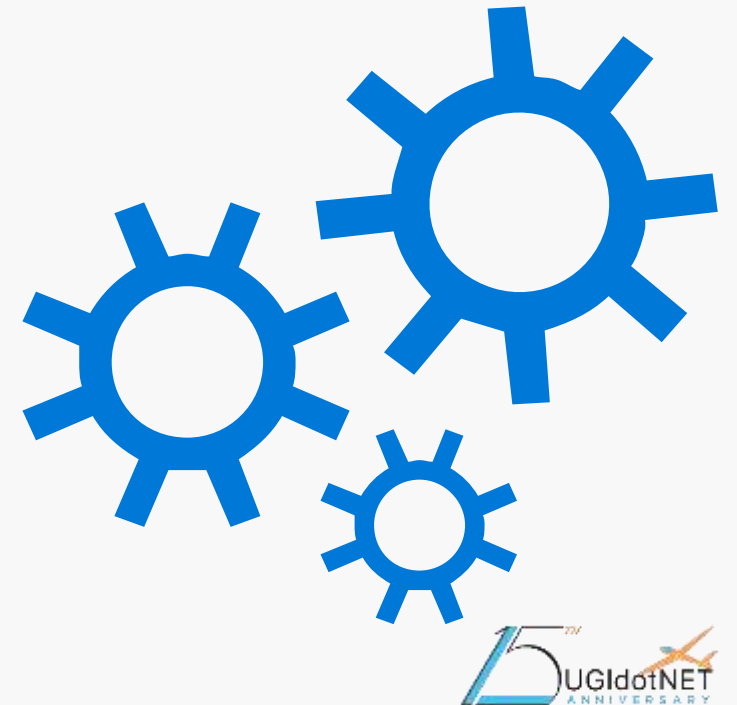
## Solving “DLL Hell” and “Registry Bloat”

Registry redirection / merging

Filesystem redirection / merging

Write redirect for certain directories

Same code will continue work on Windows 7, 8, etc.



# It's not «straightforward» at the moment

## It's a manual procedure

<http://aka.ms/converter> is your friend!

The conversion script only works when BaselImage.wim and Windows 10 have the same version:

- Disable automatic updates, updated BaselImage could take a week to appear
- Don't use it on production PC
- Installation requires a reboot to enable "Containers"

## CMD.Exe is your friend

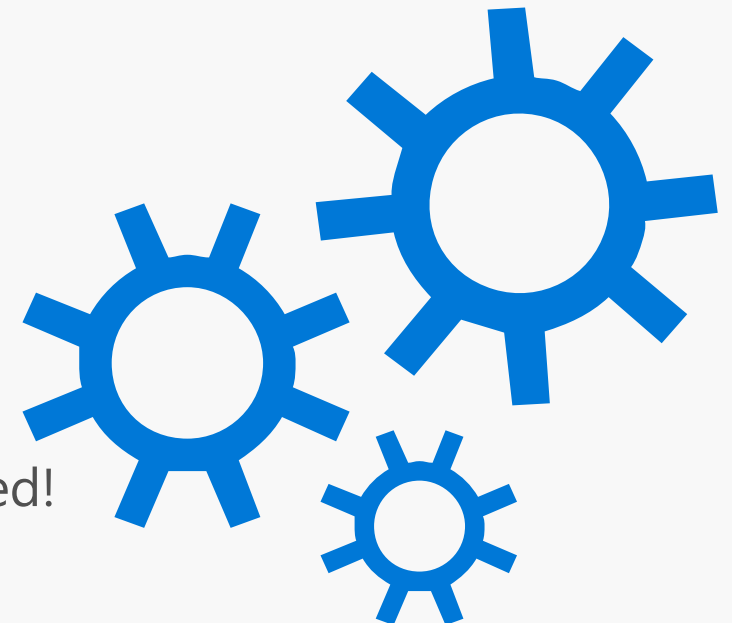
It's all script based

Application setup must run silently and logging must be redirected to a fixed folder

App should be manually signed and certificate should be manually trusted!

NO Windows NT Services, repeat with me, NO Windows NT Services,

NO drivers, only User Mode code



# Bridge to the Universal Windows Platform

At this point your app already has

Modern deployment

Activation via Tiles, file types or URI protocols – **should be added manually**

App Identity → access to more UWP API surface

UWP App Model

What else does this enable?

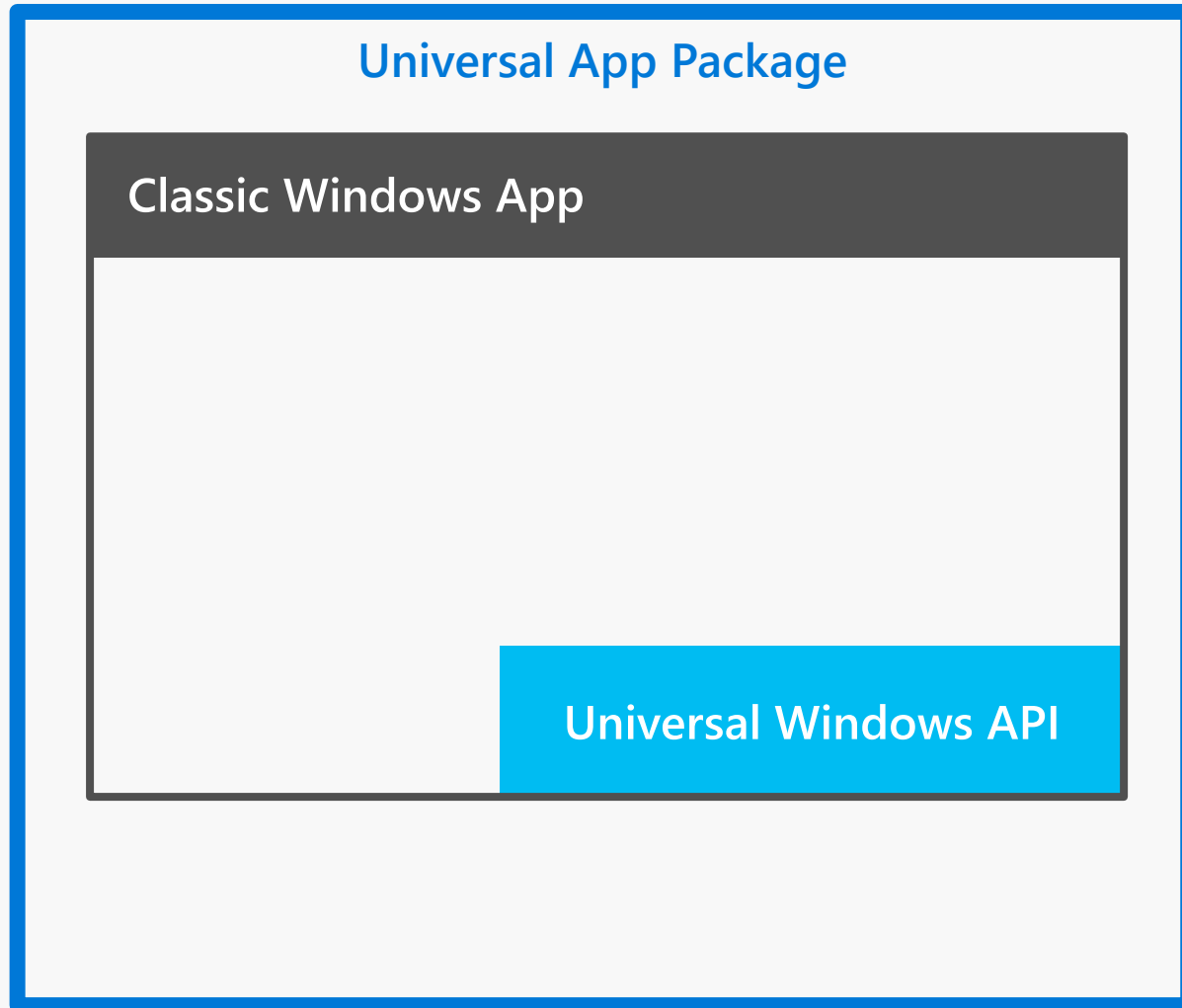
→ Short answer: Anything a UWP app can do !

DEMO

Look at manifest.xml



# Step 2 – Call UWP APIs



## Feature Examples:

Live Tile support  
Actionable Notifications  
Cortana  
etc.

# DEMO

## Toast notification in .NET

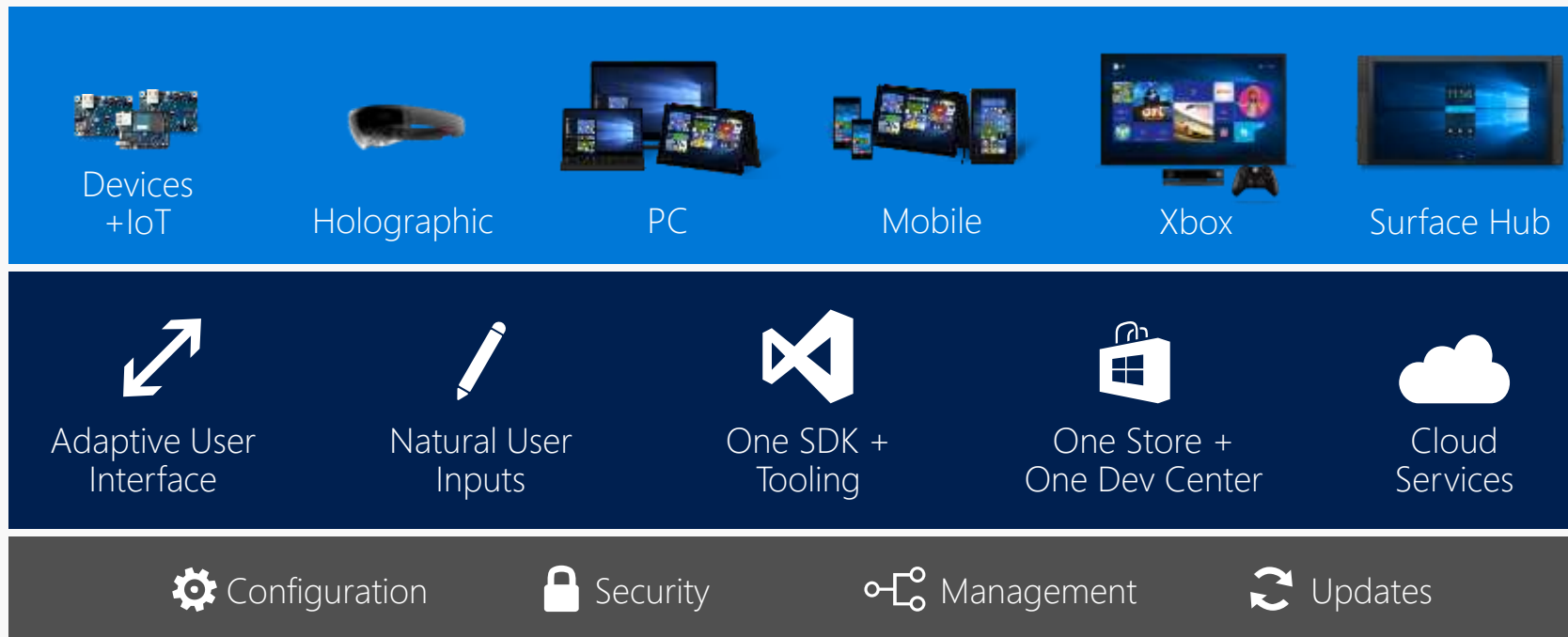
# Continue the Migration to UWP

## Do anything a UWP app can do

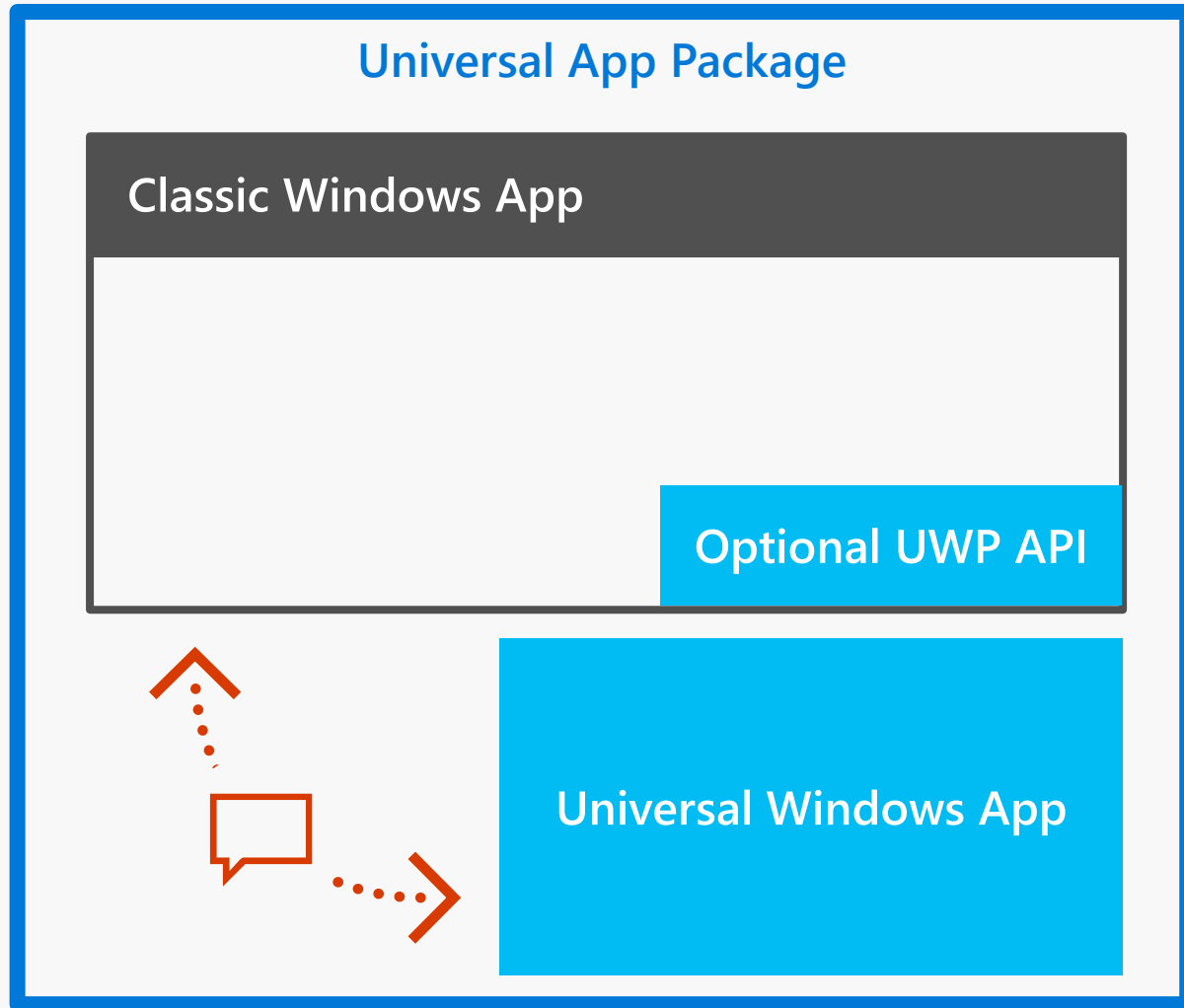
Take advantage of more UWP APIs

Take advantage of app services, background tasks & triggers

Migrate UI to XAML



# Step 3 – Add a UWP Component



## Feature Examples:

XAML UI front end

Background tasks

File Picker UI

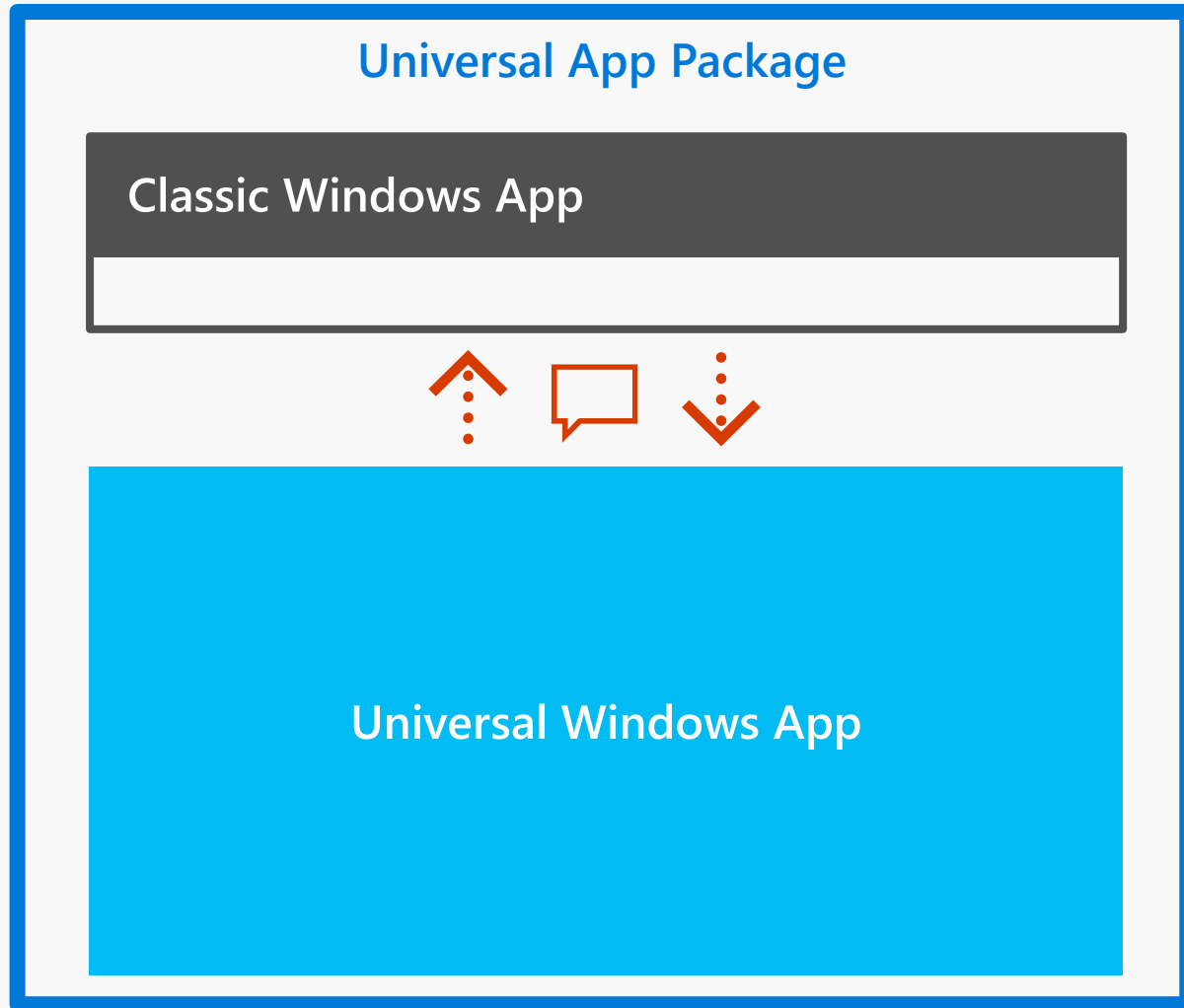
## Runs in the App Container

Communication via AppService

Each one could activate the other:

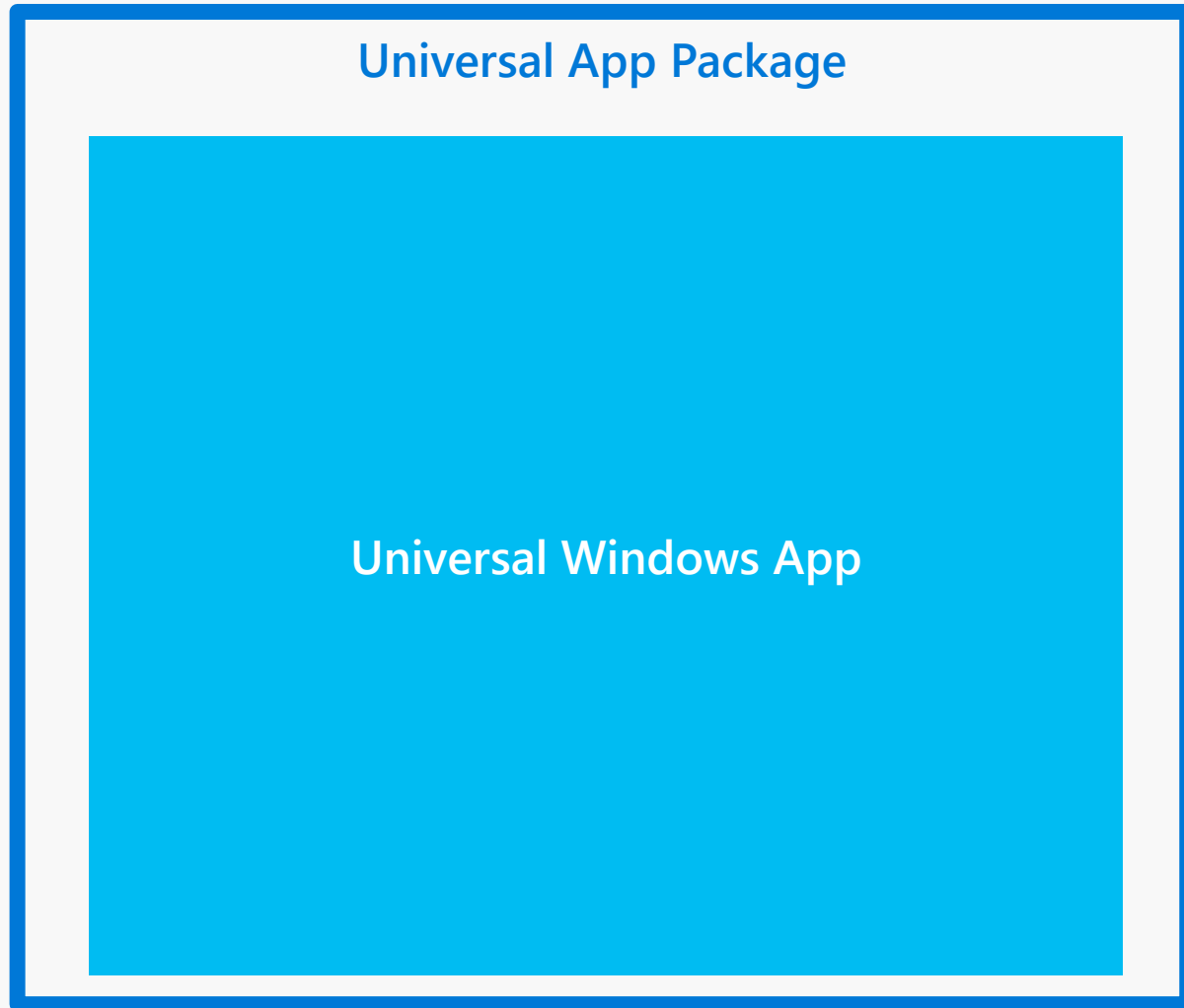
- i.e. Cortana activation
- Secondary tile

# Step 4 – Migrate code to UWP



Move UWP Compatible  
code/components into  
the App Container

# Step 5 – Fully UWP



Move remaining code  
into the App Container

Run everywhere

PC

Mobile

Xbox One

HoloLens

etc.

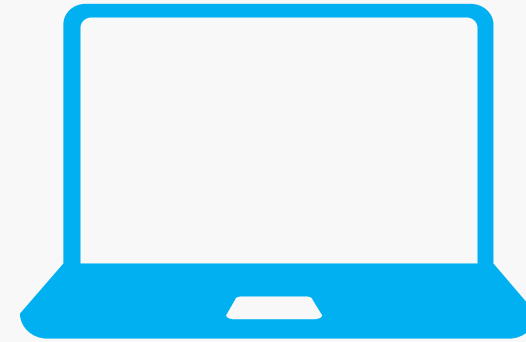
# Appendix A: Step 3 Alternatives



# There are other ways to integrate the two worlds



UWP App



Desktop App



# UWP <-> Desktop integration

## Enable Network Loopback

- + No new code to write
- need to expose data via HTTP/TCP

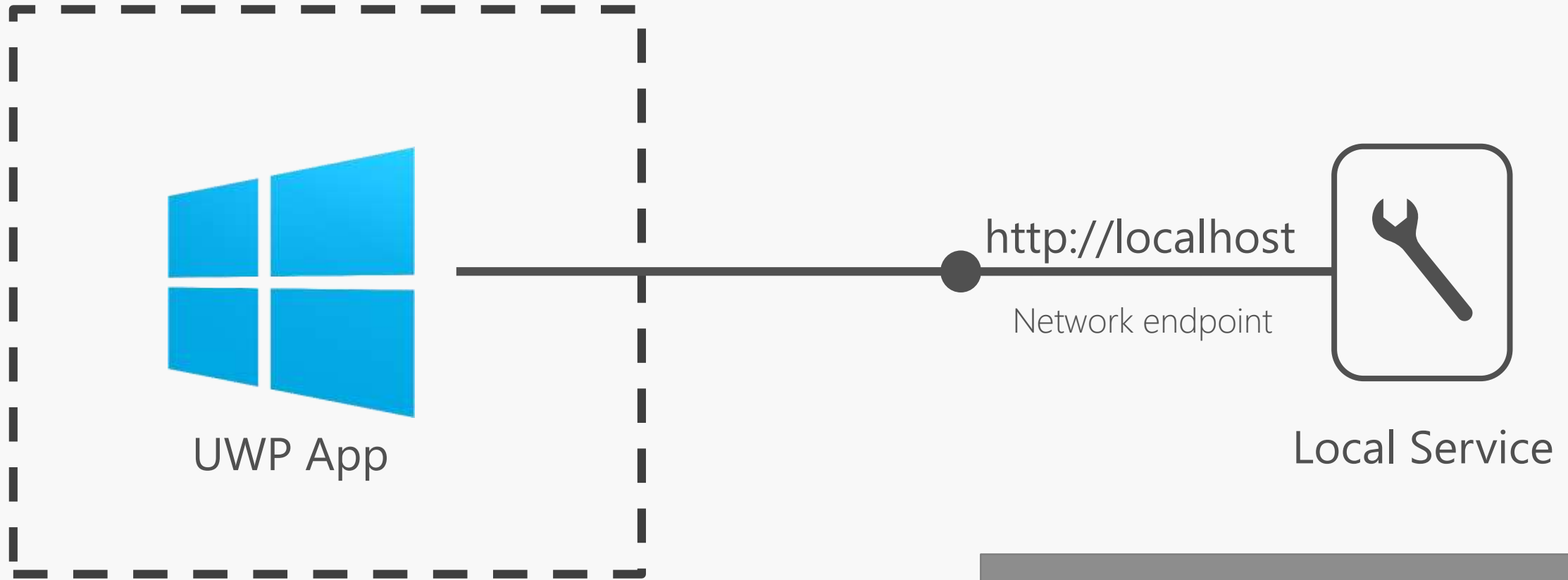
## Custom protocols

- + Small code to write
- limited scenarios

## Brokered WinRT Component

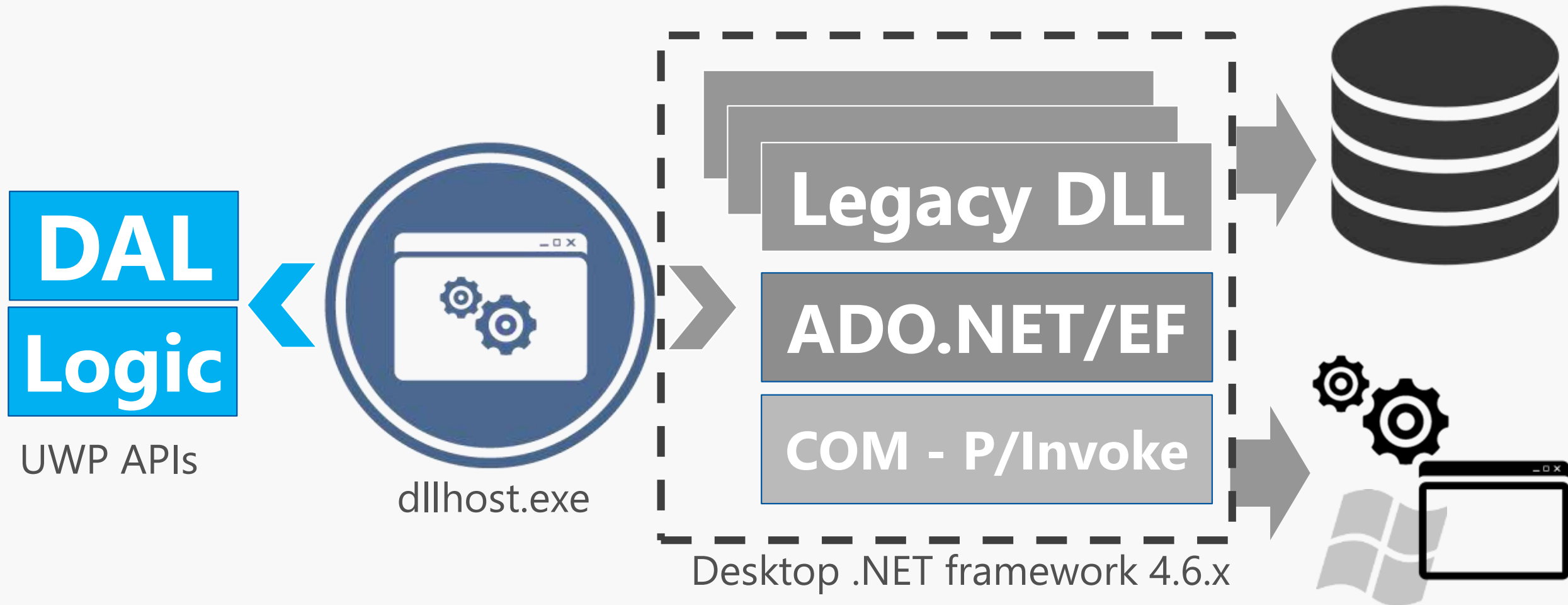
- + Allows More complex scenarios
- New code to write

# Network Loopback



**Checknetisolation.exe**

# BWC - Architecture

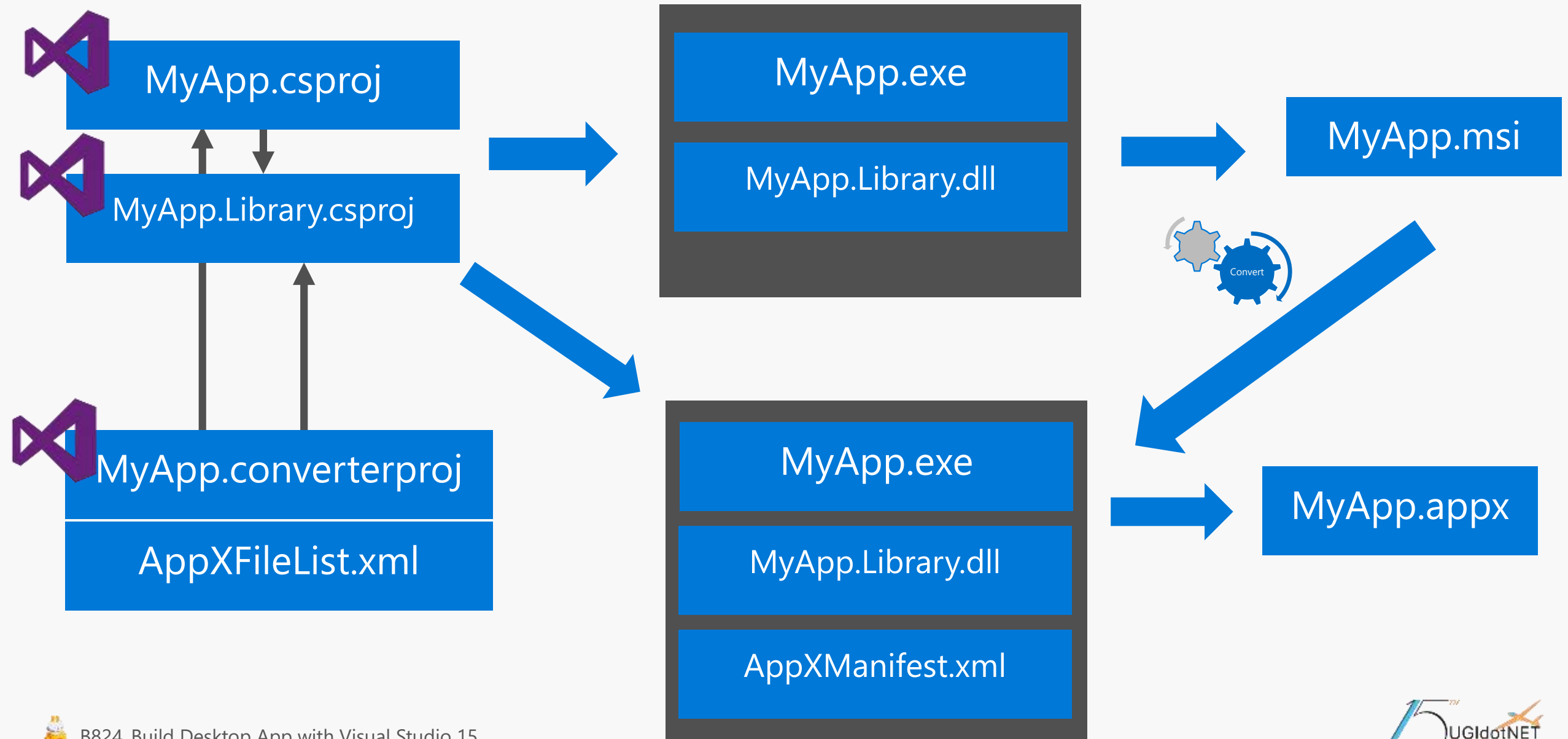


<https://msdn.microsoft.com/it-it/windows/uwp/winrt-components/brokered-windows-runtime-components-for-side-loaded-windows-store-apps>

# Appendix B: Visual Studio «15»



# Developer workflow in Visual Studio "15"



# Call to Action

- Visit [aka.ms/converter](https://aka.ms/converter) to get Desktop App Converter
- Visit [aka.ms/uwpextensions](https://aka.ms/uwpextensions) to learn more about UWP extensions – i.e. to register File Types without using registry
  - Documentation for new extensions coming soon!
- Watch Best of Build and other Desktop App Converter sessions on [Channel 9](#)
- [aka.ms/UserVoiceDesktopToUwp](https://aka.ms/UserVoiceDesktopToUwp) to provide your feedback

# Thank you! Questions?

<https://twitter.com/ugidotnet>

[lorenzo.barbieri@microsoft.com](mailto:lorenzo.barbieri@microsoft.com)

[@\\_geniodelmale](#)