



## About

- Chief Architect, DevOps & Modern Software at BlueMetal
- Microsoft Regional Director
- Microsoft MVP
- Visual Studio ALM Ranger
  
- Blog: [//oren.codes](https://oren.codes)
- Twitter: [@onovotny](https://twitter.com/onovotny)



## What kind of apps?

- WPF?
- Windows Forms?



## Windows 10

- How many of your companies have moved to Windows 10?
- Remain on Windows 7?
- Plans to move by ...?



## Ways to enhance

- Distribution
- Add modern controls
- Enhancements to WPF and Windows Forms
- Bringing in cloud data with Graph
- Create touch-friendly companion apps



## DISTRIBUTION

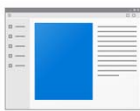


## Challenges today

- Installers put things everywhere
- Machine rot
- Clean uninstall virtually impossible
- Auto-updating
- Repackaging apps takes time
- .NET SxS



## Evolve to a modern deployment technology



Setup.exe



MSI



MSIX



Microsoft  
Application  
Virtualization



script.vb  
s



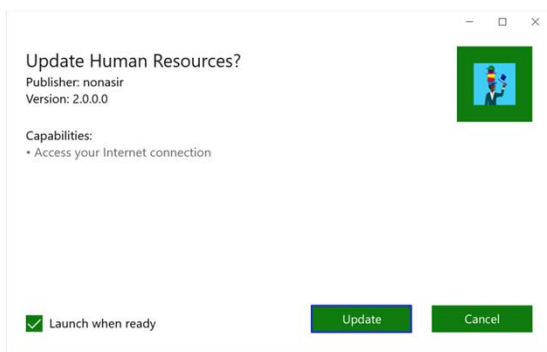
## MSIX Benefits

- Never regret installing an app: clean uninstall
- Uses a type of container
- Less disk space
- Faster updates: differential updates at block level
- OS Managed
- Integrity protection



## Automatic Updating

### User Prompted Updates



### Silent updates in background



Updates  
independently of  
app launch



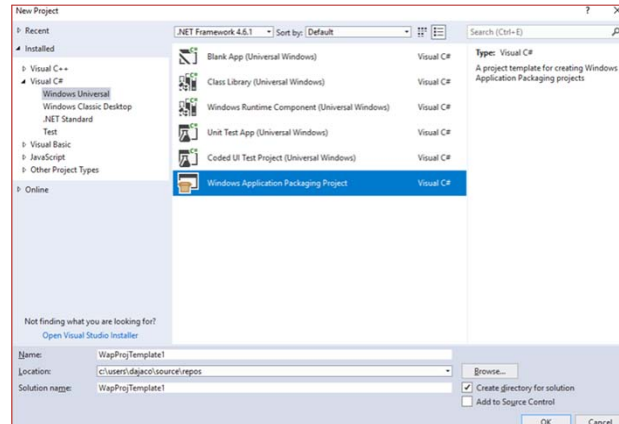
API to check for app  
updates while app is  
running



# Migrate existing projects

## New project template in VS 2017

- Package your Windows project directly in VS for streamline deployment
- You can reference your existing projects such as: WPF, Winforms, or MFC
- Leverage the advanced Universal Windows Platform tools such as: Packaging, Manifest editing, and Store association



Packaging project as MSIX

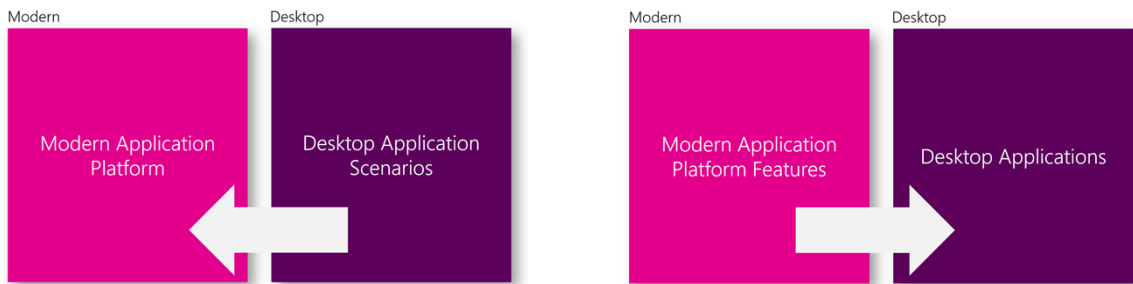
## DEMO



# MODERN UX



UWP XAML is LOB ready and hostable in your current apps



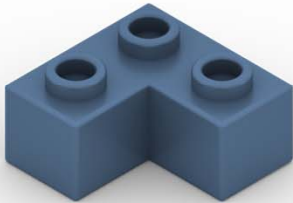
Improved Information Density  
LOB Controls & Framework  
Features  
Windowing APIs

XAML Islands



# Density

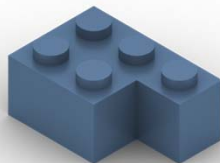
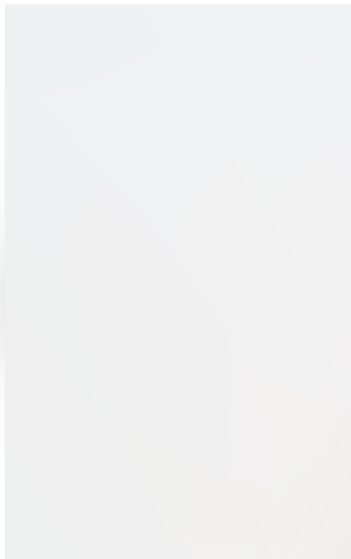
Today



# Density

New Microsoft Standard

New Compact Size





Density

Today

Control header

Level 1 peer item

Level 1 node

Level 2 peer item

Level 2 node

Level 1 node

Button

New Microsoft Standard

Control header

Level 1 peer item

Level 1 node

Level 2 peer item

Level 2 node

Level 1 node

Button

New Compact Size

Control header

Level 1 peer item

Level 1 node

Level 2 peer item

Level 2 node

Level 1 node

Button

Density

New Microsoft Standard

Control header

Level 1 peer item

Level 1 node

Level 2 peer item

Level 2 node

Level 1 node

Button

New Compact Size

Control header

Level 1 peer item

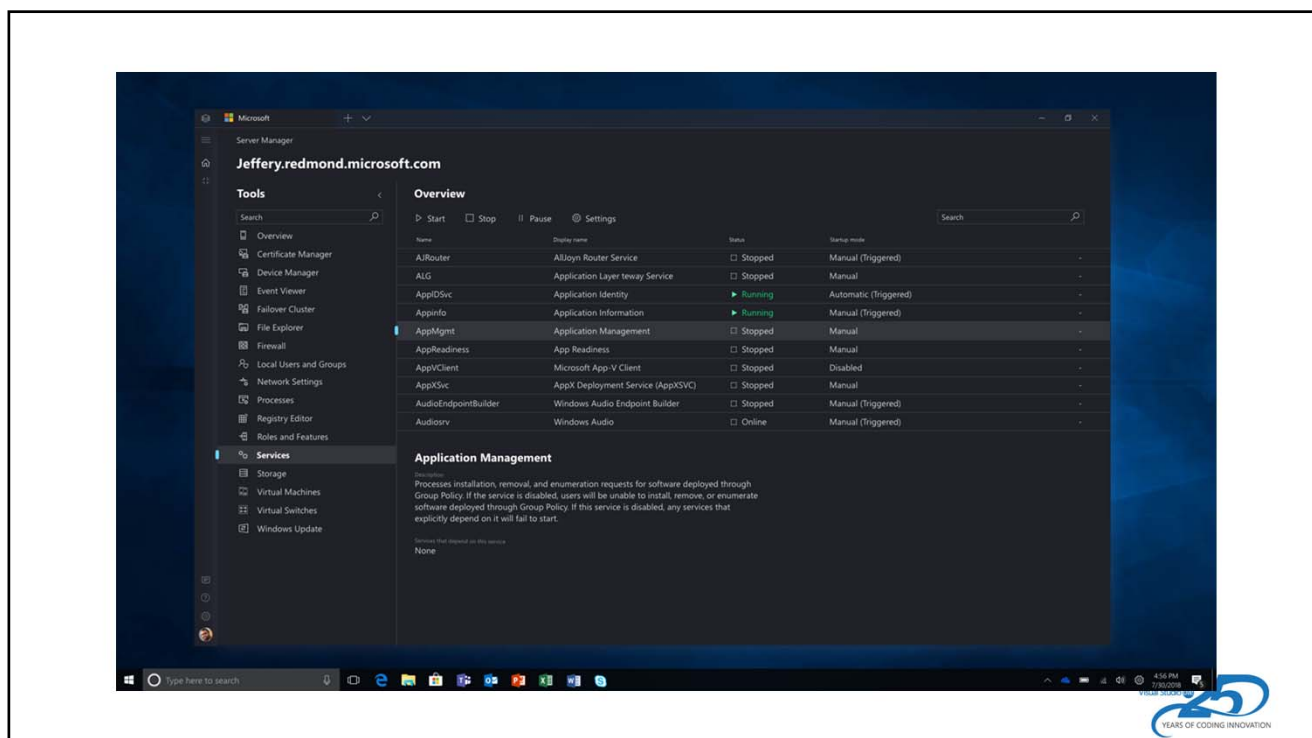
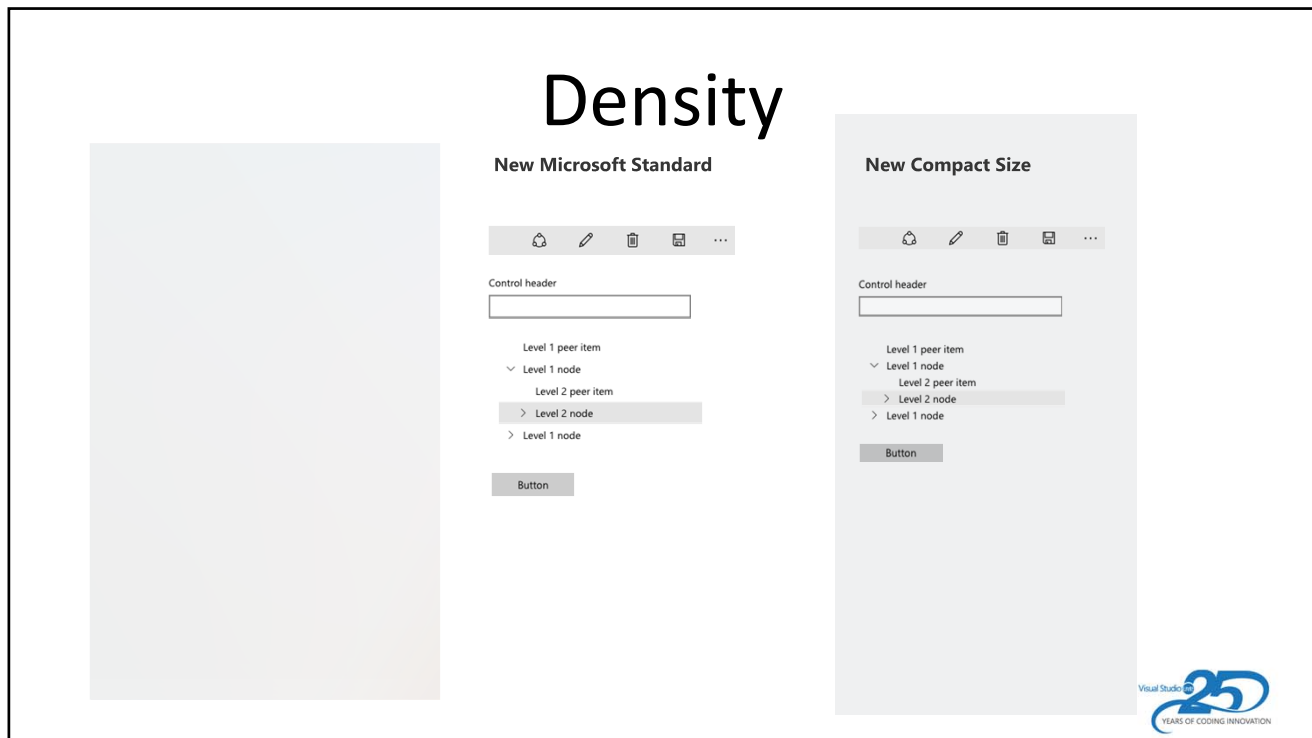
Level 1 node

Level 2 peer item

Level 2 node

Level 1 node

Button

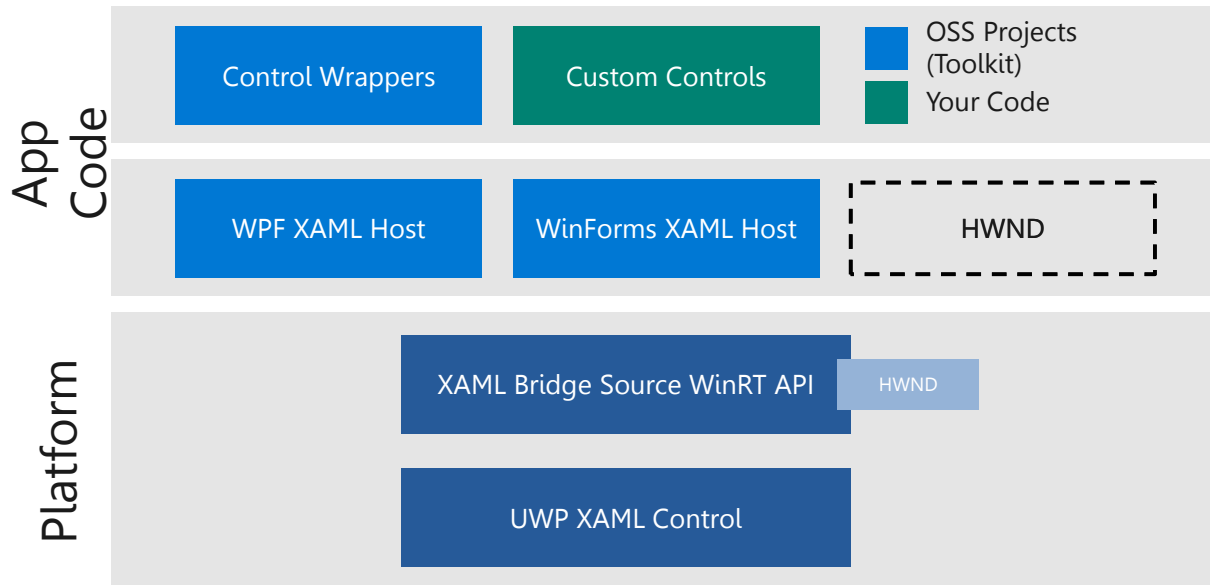


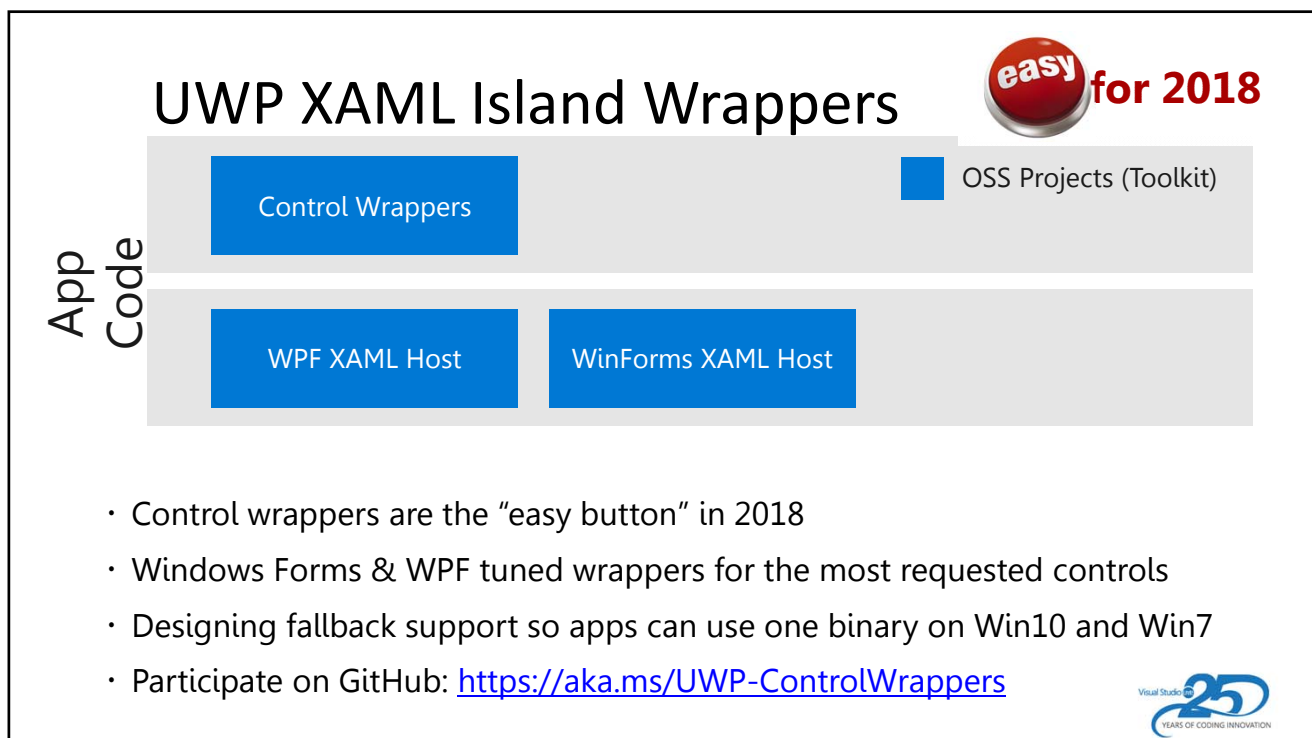
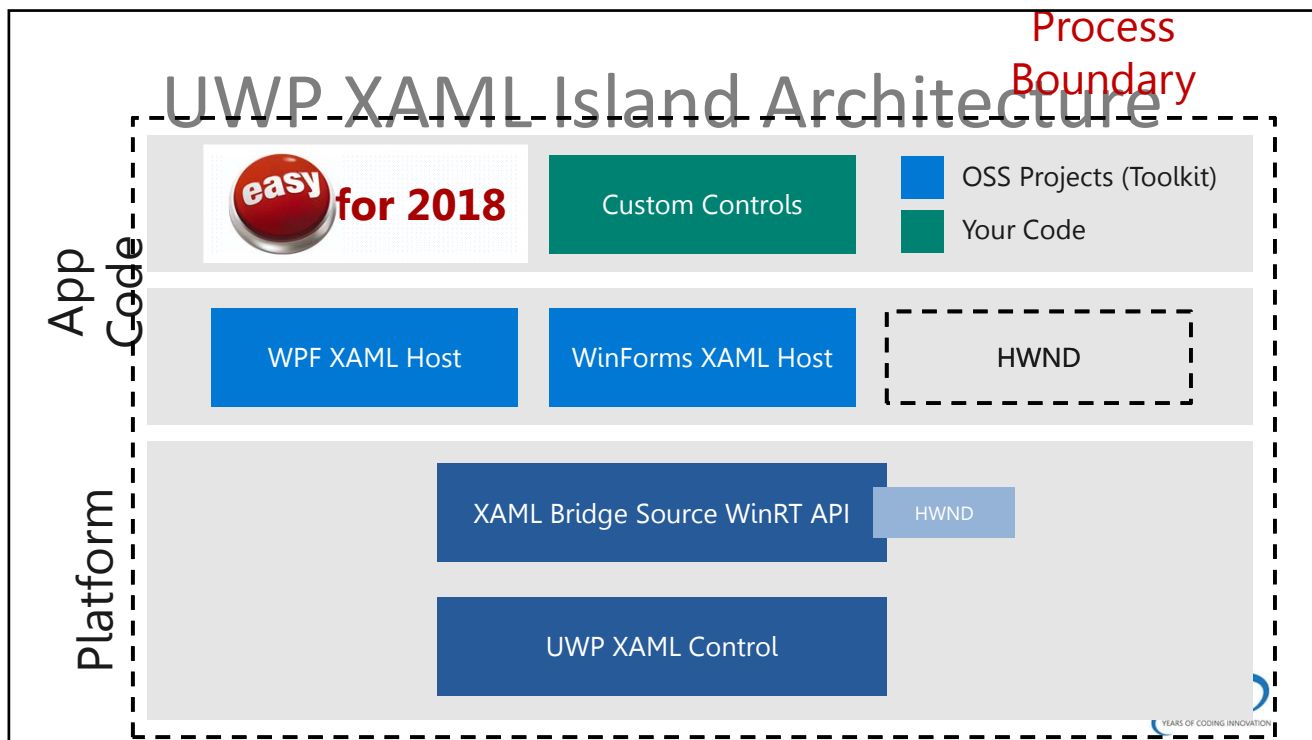
## Mix n' match

- Lots of existing applications
- Rewriting isn't an option (nor should you)
- Want to use some of the modern controls
  - WebView based on Edge instead of IE
  - Media Element
  - Ink



## UWP XAML Island Architecture





## UWP XAML Island Wrappers



App  
Code

Control Wrappers

OSS Projects  
(Toolkit)

WPF XAML Host

WinForms XAML Host

### Wrapped UWP Controls

WebView, MediaPlayer, InkCanvas & InkToolBar, Map, SwapChainPanel

Available as part of the Windows Community Toolkit

Participate on GitHub: <https://aka.ms/UWP-ControlWrappers>



## Web View

- How many of your apps use an embedded web view?
- WPF/WinForms use Trident (IE's rendering engine)
- Two new controls
  - WebView
    - Works only on Windows 10, exposes more API
  - WebViewCompatible
    - Works on Win 7 & Win 10, exposes API subset
    - Uses Trident on Win 7, Edge on Win 10



WebView & XAML Islands

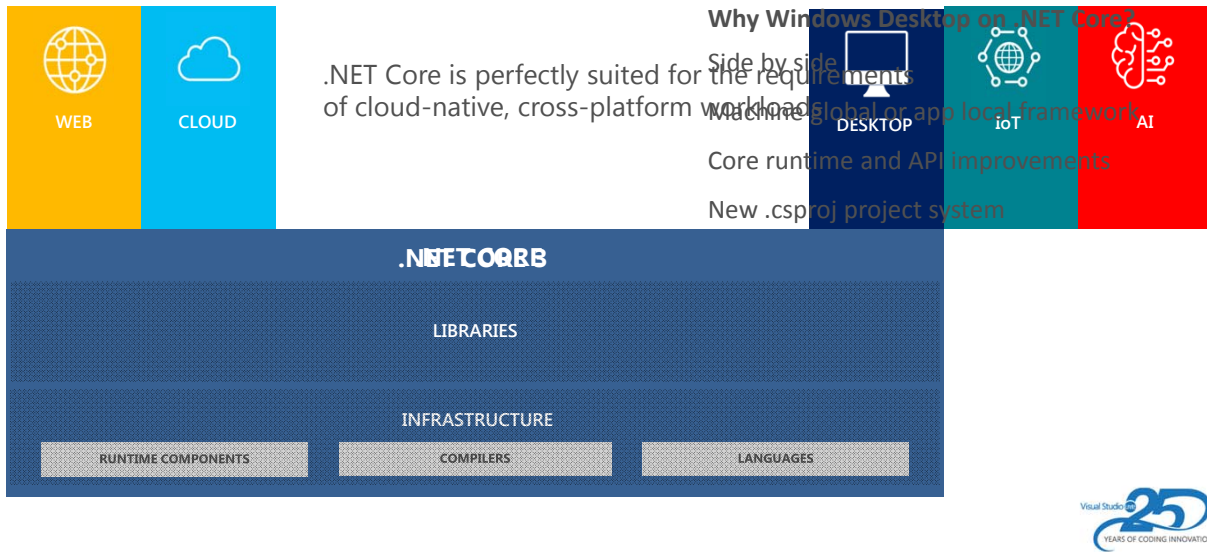
## DEMOS



## WPF & WINFORMS ROADMAP



## .NET Core 3



## .NET Core 3 Desktop Improvements

- Side-by-side
- .NET Core 3 supports WinForms and WPF frameworks
  - XAML Islands - WinForms & WPF can host UWP
  - XAML Controls – WinForms & WPF browser and media UWP controls
  - High DPI fixes for WinForms
- NET Core App Bundler
  - Precompiled, fast startup
  - Small apps by removing unused dependencies, link away unused IL

## Why Windows Desktop on .NET Core?

- Machine global or app local framework
- Core runtime and API improvements
- SDK based .csproj project system
- Access to all the Windows 10 API's



## .NET Core 3

- Next major release
- Adds WPF / WinForms
- App-local or shared framework
- Can be single exe
- Can use existing .NET Framework libraries as-is (control libraries)





.NET Core 3 app

**DEMO**



**CONNECTING TO THE CLOUD**

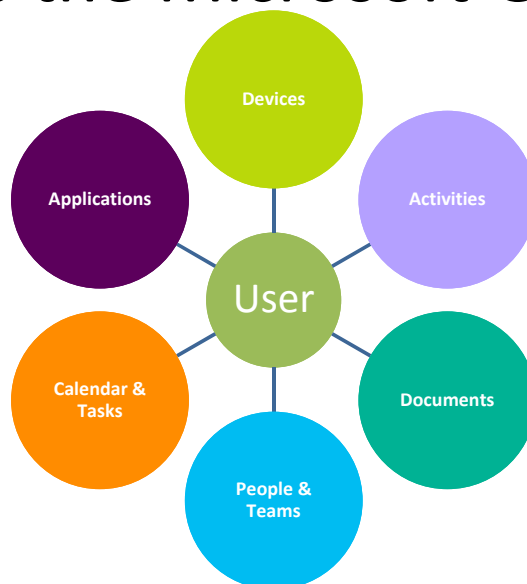


## M365 and the Graph

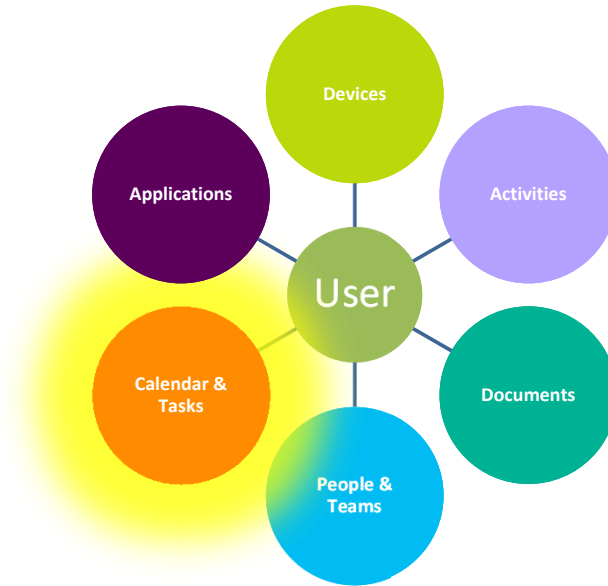
- Hype?
- Connectivity
- Future of office productivity



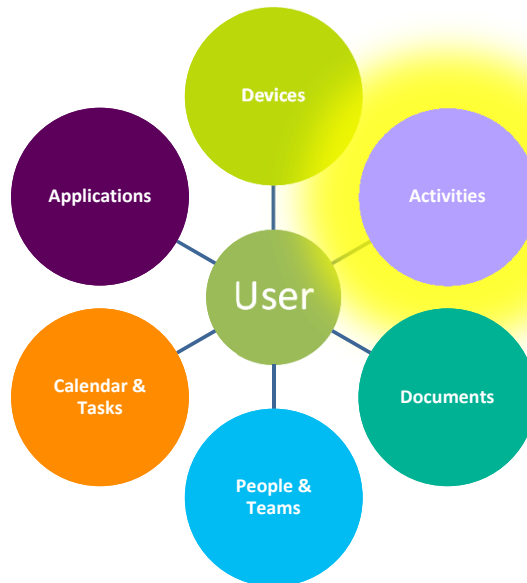
## What is the Microsoft Graph?



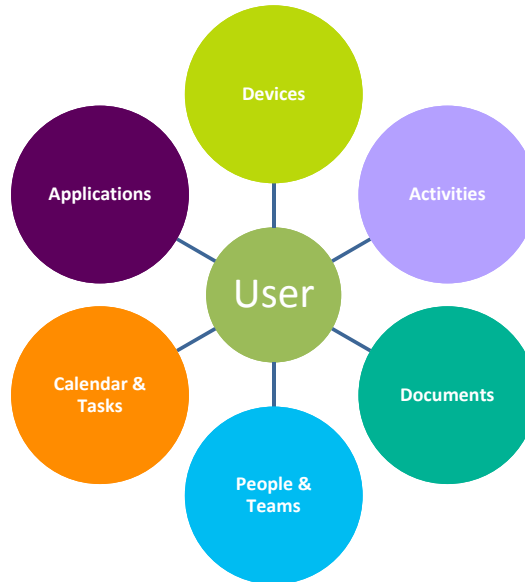
## What is the Microsoft Graph?



## What is the Microsoft Graph?



# What is the Microsoft Graph?



## Graph – Examples

- Adding “smarts” to a pick-list of users
- Accessing files from OneDrive
- Profile information
- Any data from SharePoint, Outlook, Planner, etc
- Sharing activities between devices
- Change notification (via webhooks)



Logging in with AAD

**DEMO**



**TOUCH FRIENDLY APPS**



## Touch screens increasing

- Many people are using Surface and other tablets
- Some kinds of apps are better with touch
  - Maps
  - Field service
- Embedded screens
  - Factory control systems
  - Medical devices



## Creating projects is hard

- Lots of plumbing code
- Learning curve with new concepts



## Windows Template Studio

- Accelerates creation of new UWP apps
- Wizard based
- Uses proven patterns and best practices
- Generated code has links to docs, SO and blogs
- Built with the community, for the community
- [aka.ms/mts](https://aka.ms/mts)



## Example

- Field service app
- Needs to remember settings
- Connects to intranet for certain web content
- Updates data in the background
- Reports telemetry



Windows Template Studio

**DEMO**



**THANK YOU!**

