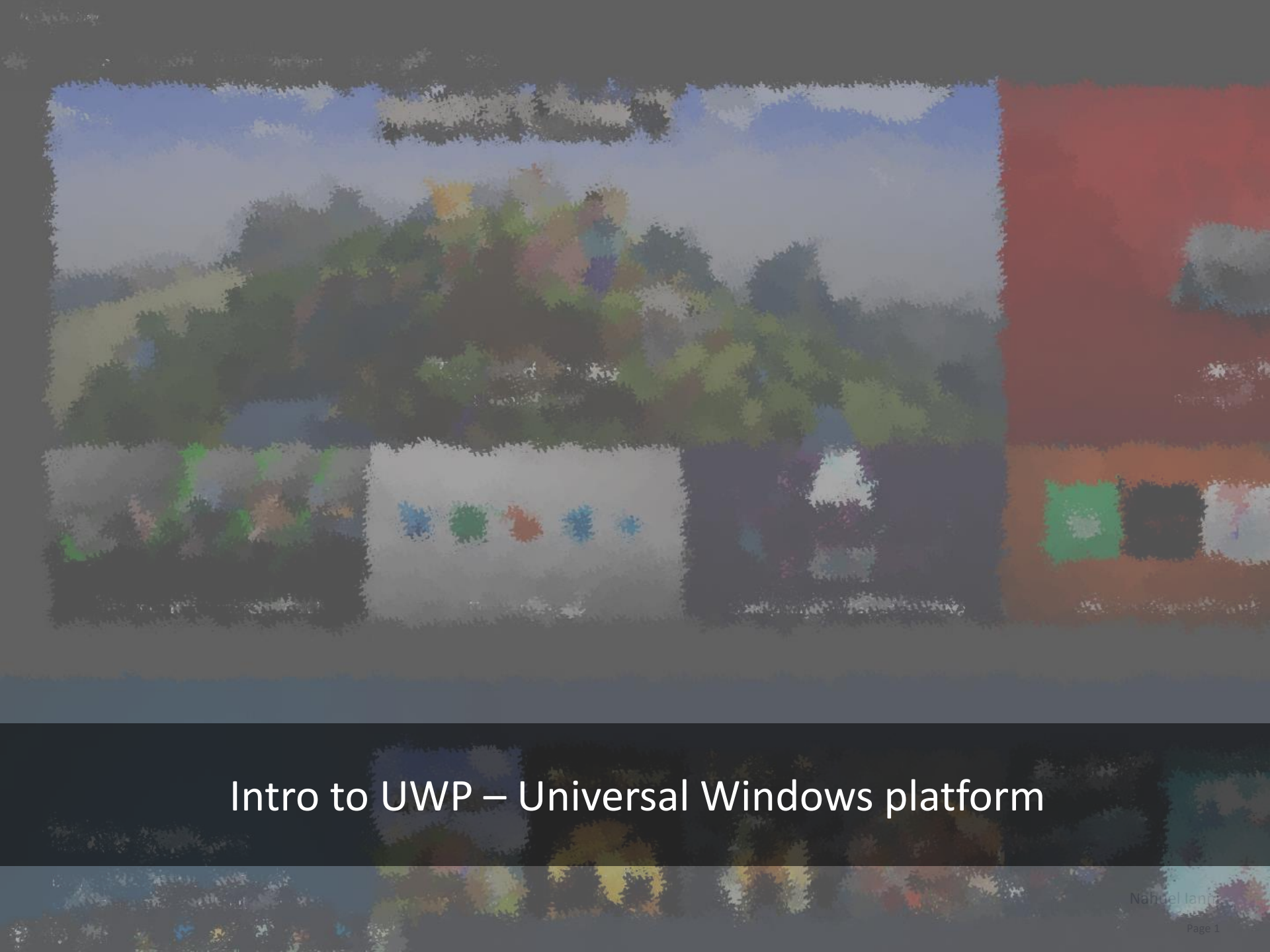


Intro to UWP – Universal Windows platform

Introduction

INTRODUCTION

Meet Nahuel

Nahuel, indigenous word for **tiger** in the long lost Araucanian's native tongue; is one of the oldest employees at ERNI Spain.

He has a wide range of experience in several different environments, technologies and roles. From software development, to testing, trainings and group leadership; he's been involved in all aspects of the application development lifecycle.

Technology enthusiastic, highly involved as of late in the video game industry and mobile platform development.

You can get in contact with him via LinkedIn: https://linkedin.com/in/nahuelianni

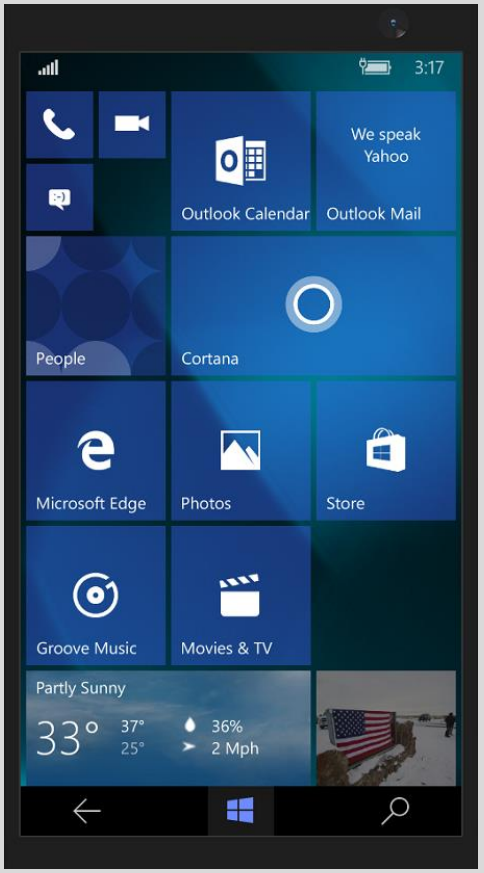
INTRODUCTION

Table of contents

Environment setup overview

INSTALLATION

Windows 10



Download links:

Windows 10:
https://www.microsoft.com/en-us/windows/windows-10-upgrade

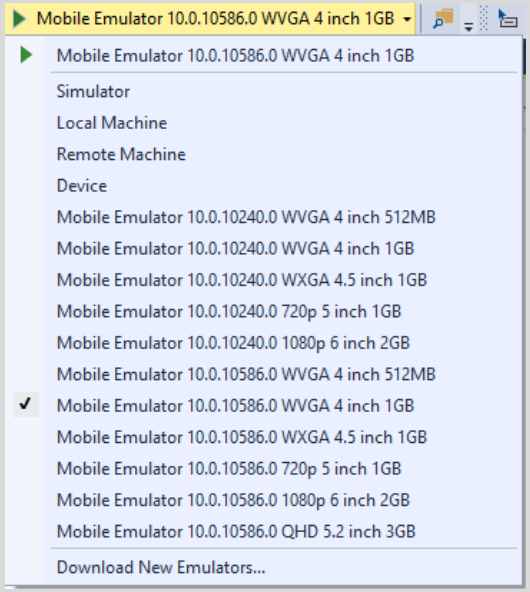
Visual Studio 2015 community edition:
https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx

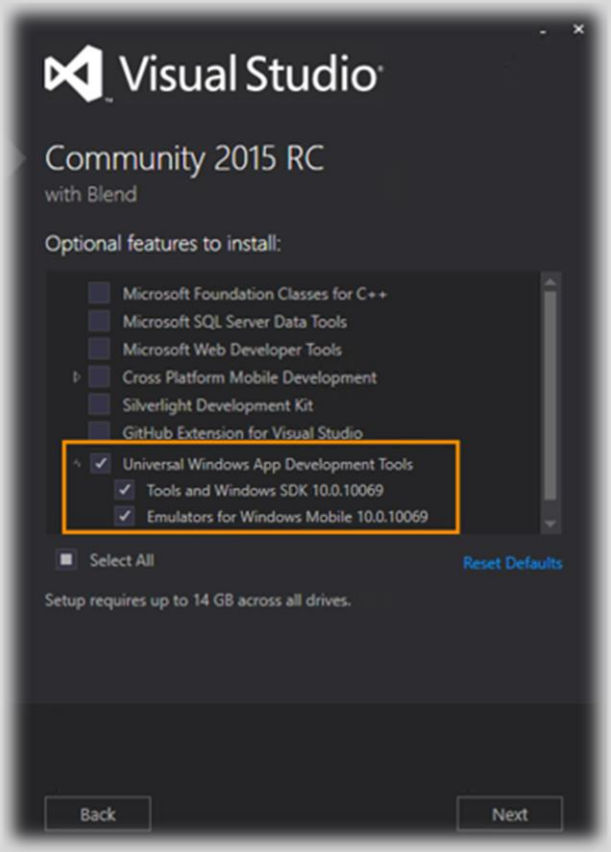
SDK and emulators:
https://dev.windows.com/en-us/downloads/sdk-archive

INSTALLATION

INSTALLATION PROCESS

- Install Windows 10.
- Install Visual Studio 2015.
- Activate Hyper-V via the system bios.
- Activate Hyper-V via the Windows menu.
- Install the UWP SDK and tools.
- Install the Windows Phone emulators update.





Demo: Windows Phone OS tour

Demo: Project template structure

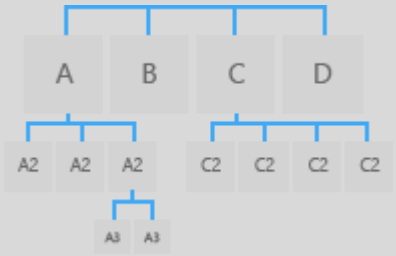
Demo: Application manifest and store intro

Application navigation

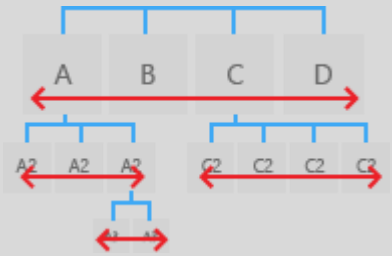
NAVIGATION

Basics

HIERARCHICAL NAVIGATION



PEERS NAVIGATION



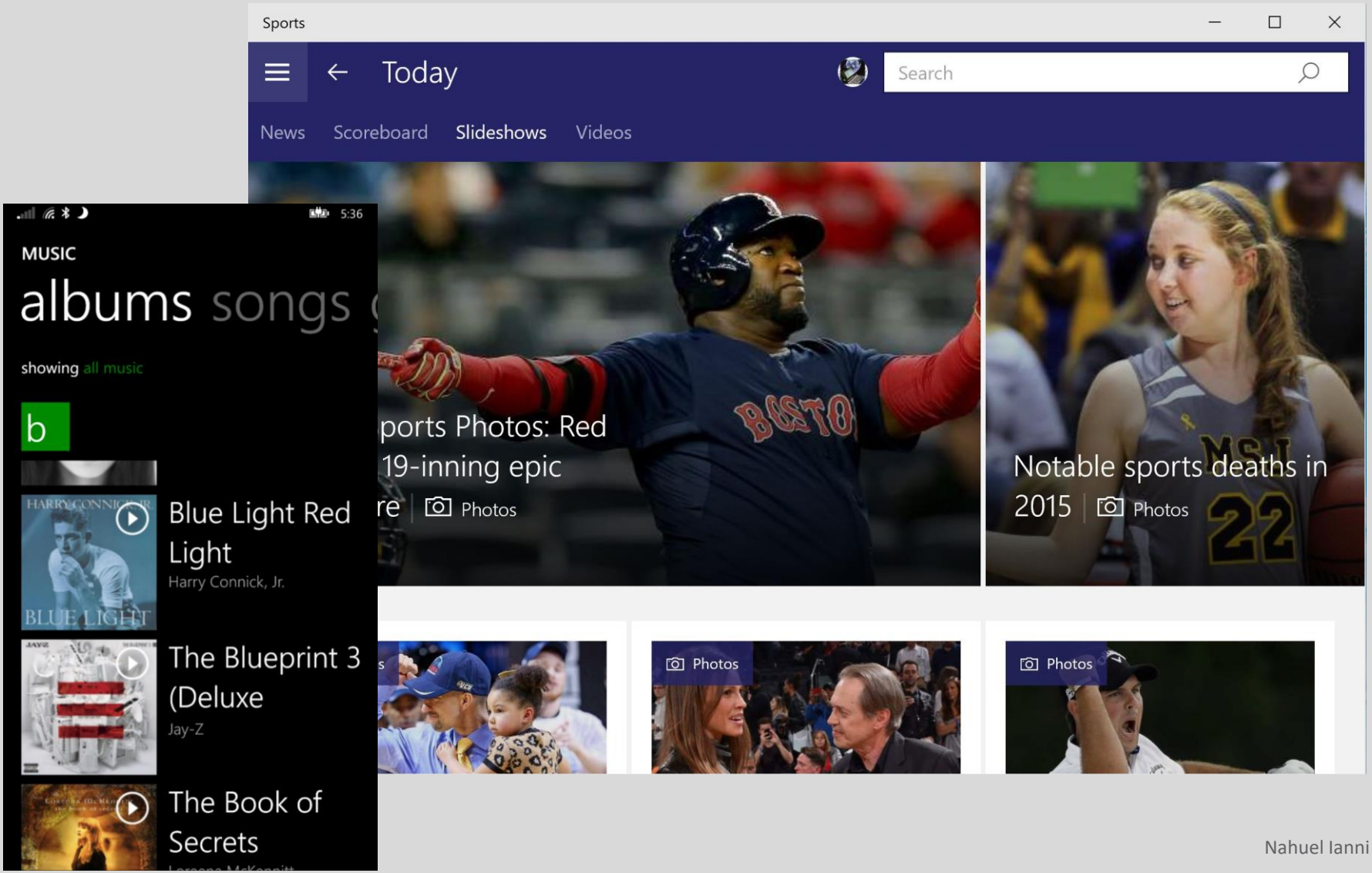
Remember the Lucky 7!

- Use a hierarchical version when there are more than 7 items.
- Use peer navigation when there are 7 or less pages to navigate to.

Source links
https://msdn.microsoft.com/en-us/library/windows/apps/dn958438.aspx

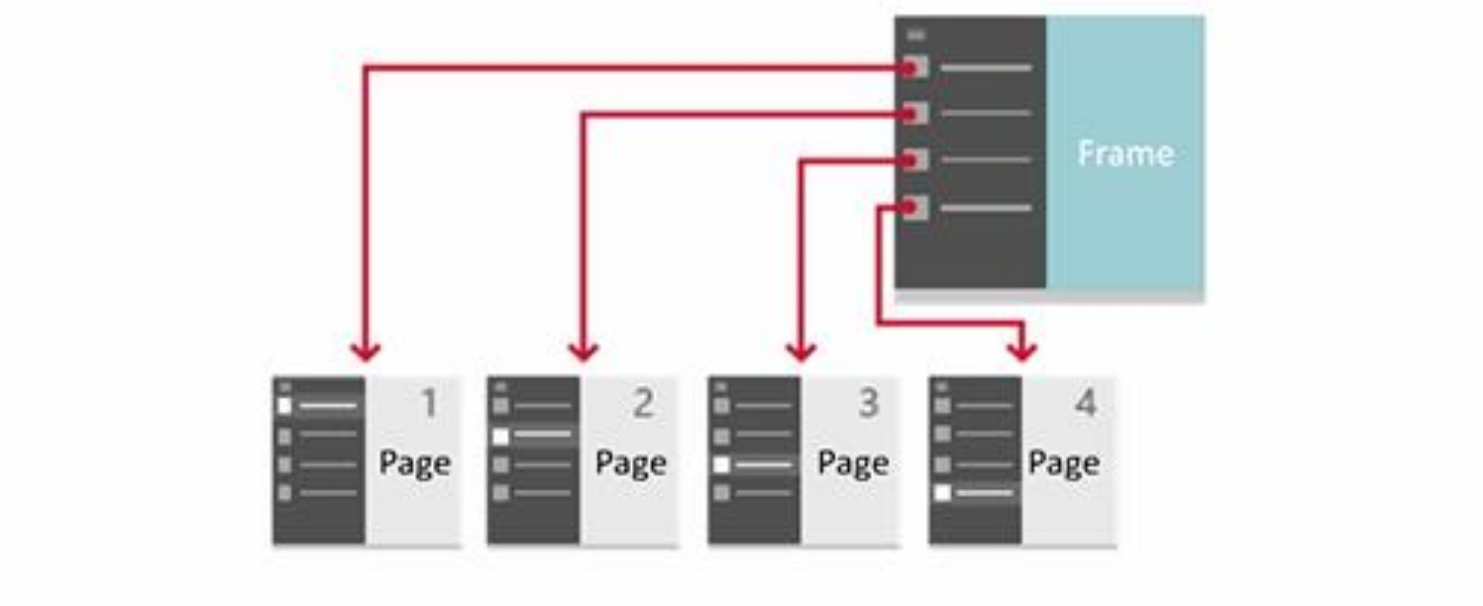
NAVIGATION

Microsoft apps example



NAVIGATION

Frame



- Frame is a container that resemble a browser window.
  - Stores information about the navigation context.
- A page, which contains the individual view; is placed inside the frame.
  - Has the information shown to the user.

NAVIGATION

Frame navigation

SIMPLE NAVIGATION

- Frame.Navigate(typeof(MainPage));

NAVIGATION WITH PARAMETERS

- Frame.Navigate(typeof(MainPage), new object());

Remarks

- Any object can be used as a parameter when navigating to a page, as long as it is serializable.
- Navigation to a page can be cached, meaning the page will remain in memory for faster loading.
- Keep in mind the memory footprint generated.

NAVIGATION

Page and frame relationship

PAGE CLASS

```
public class Page : UserControl, IPage, IPageOverrides
{
    public Page();

    public static DependencyProperty BottomAppBarProperty { get; }
    public static DependencyProperty FrameProperty { get; }
    public static DependencyProperty TopAppBarProperty { get; }
    public AppBar BottomAppBar { get; set; }
    //
    // Summary:
    //     Gets the controlling Frame for the Page content.
    //
    // Returns:
    //     The controlling Frame for the Page content.
    public Frame Frame { get; }
    public NavigationCacheMode NavigationCacheMode { get; set; }
    public AppBar TopAppBar { get; set; }

    protected virtual void OnNavigatedFrom(NavigationEventArgs e);
    protected virtual void OnNavigatedTo(NavigationEventArgs e);
    protected virtual void OnNavigatingFrom(NavigatingCancelEventArgs e);
}
```

NAVIGATION

Page caching

PAGE CLASS

```
public class Page : UserControl, IPage, IPageOverrides
{
    public Page();

    public static DependencyProperty BottomAppBarProperty { get; }
    public static DependencyProperty FrameProperty { get; }
    public static DependencyProperty TopAppBarProperty { get; }
    public AppBar BottomAppBar { get; set; }
    public Frame Frame { get; }
    //
    // Summary:
    //     Gets or sets the navigation mode that indicates whether this Page is cached,
    //     and the period of time that the cache entry should persist.
    //
    // Returns:
    //     A value of the enumeration. The default is Disabled.
    public NavigationCacheMode NavigationCacheMode { get; set; }
    public AppBar TopAppBar { get; set; }

    protected virtual void OnNavigatedFrom(NavigationEventArgs e);
    protected virtual void OnNavigatedTo(NavigationEventArgs e);
    protected virtual void OnNavigatingFrom(NavigatingCancelEventArgs e);
}
```

NAVIGATION

Page caching remarks

- To enable a page to be cached, set NavigationCacheMode to either Enabled or Required. The difference in behavior is that Enabled might not be cached if the frame's cache size limit (CacheSize) is exceeded, whereas Required always generates an entry no matter the size limit.

- If you want to change the value of NavigationCacheMode programmatically to Enabled or Required, you can only set these values in the constructor for the page.

- If you change the value of NavigationCacheMode from Required or Enabled to Disabled, the page is flushed from the cache. The page is not simply marked as available to be flushed when the configured CacheSize is exceeded.

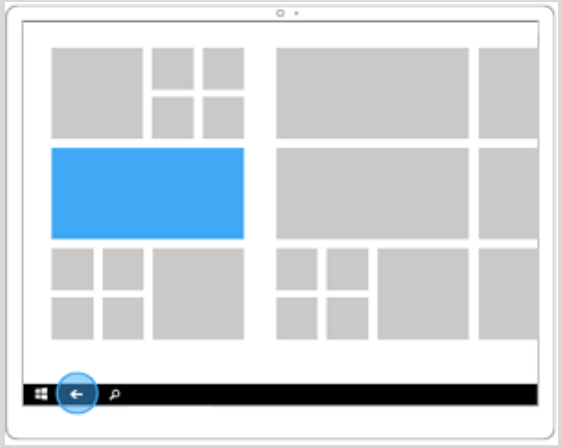
Source links
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.page.navigationcachemode.aspx

NAVIGATION

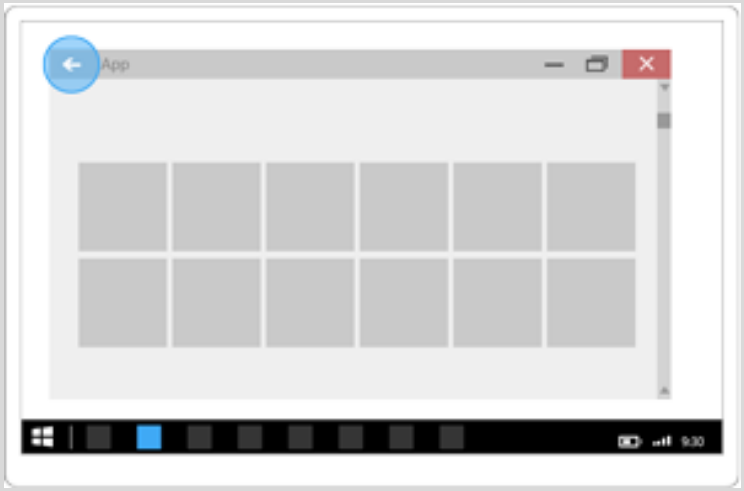
Handling back navigation

- Functionality now ready to be provided by the OS.
- Hardware/Software **may** differ greatly between devices.

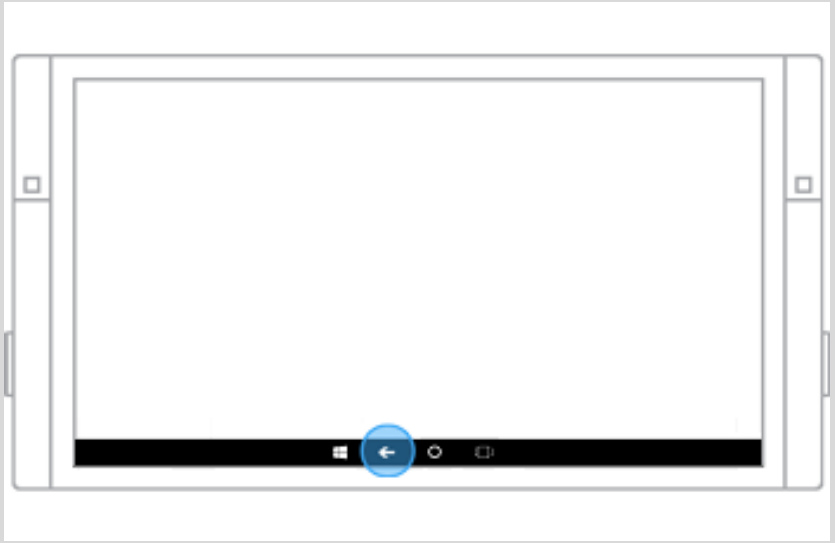
Phone



Tablet



Desktop



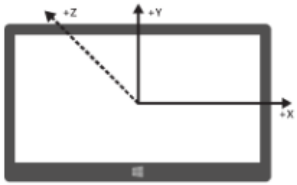
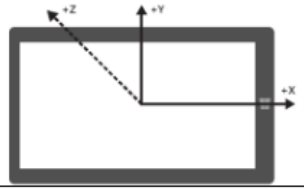
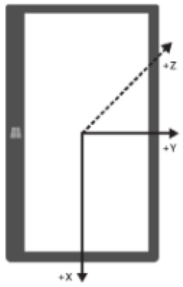
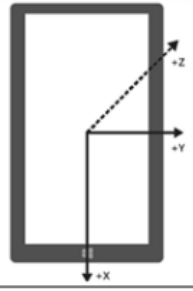
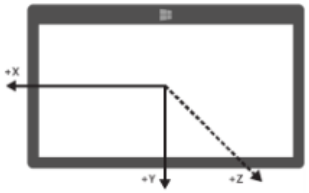
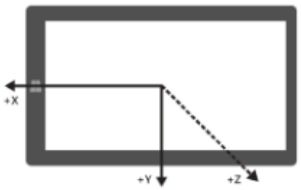
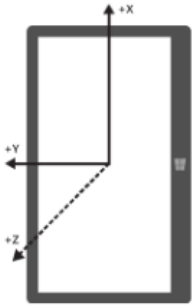
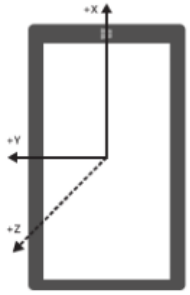
Surface Hub

Demo: Application navigation

View design

VIEW DESIGN

Device orientation

	Landscape-First	Portrait-First
Landscape		
Portrait		
LandscapeFlipped		
PortraitFlipped		

VIEW DESIGN

Adaptability and responsiveness

- The views, regardless of device; should always try to be **responsive** and **adaptive**.
- **Adaptive** layouts rearrange themselves by adding, removing or modifying the content they will display.
- **Responsive** layouts adapt themselves to the size of the screen where they are shown, such as grids.

More info can be found on the following links for adaptive and responisve layouts respectively:
https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt210475.aspx
https://msdn.microsoft.com/en-us/library/windows/apps/dn958435.aspx

VIEW DESIGN

Responsive (reposition)

Though it may seem adaptive, the controls reposition themselves based on the device.

VIEW DESIGN

Adaptive (change of content)

New content or functionality is shown depending on the device it runs on.
e.g. If menu 'E' was to make a call, it makes sense to have it on a phone but not so much on a desktop.

VIEW DESIGN

Adaptive (change of achitecture)

Depending on the device type, a change in the behavior of the application may be needed.
e.g. A "master-detail all in one" functionality changes to that of a "navigation" one.

VIEW DESIGN

Responsive (resize)

VIEW DESIGN

How to adapt

- The UI of the application can be design in two ways:
  - Custom XAML views per device family.
    - Complement with device API: *ApiInformation*.
  - Visual states and adaptive triggers.
    - Blend. Blend! BLEND!
    - Best reserved for simple, small cases.
    - Blend simplifies the use of states and triggers.
    - Can get messy really soon.
    - Did I mention you should be using Blend in this case?
    - Depending on the implementation of the controls, performance may suffer (visibility management).
    - *"Don't hate the Blend, hate the game".*

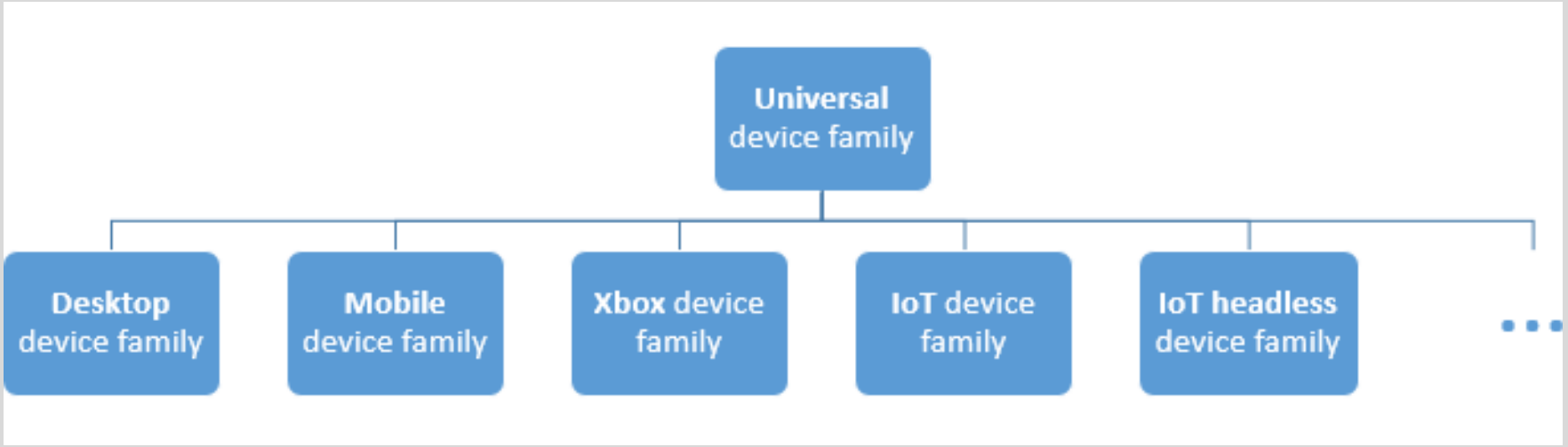
More info on the device family info and adaptive triggers respectively, can be found on the following links:
https://msdn.microsoft.com/en-us/library/windows/apps/dn958439.aspx
http://blogs.msdn.com/b/goutham/archive/2015/05/17/visual-state-manager-and-adaptive-triggers-for-universal-windows-app.aspx

VIEW DESIGN

All about the size...

VIEW DESIGN

… and the device!



Adaptive trigger	What is targeted
<AdaptiveTrigger MinWindowWidth="320"/>	Phone
<AdaptiveTrigger MinWindowWidth="720"/>	Tablet
<AdaptiveTrigger MinWindowWidth="1024"/>	Desktop
<AdaptiveTrigger MinWindowWidth="1280"/>	XBox
<AdaptiveTrigger MinWindowWidth="1920"/>	Surface Hub

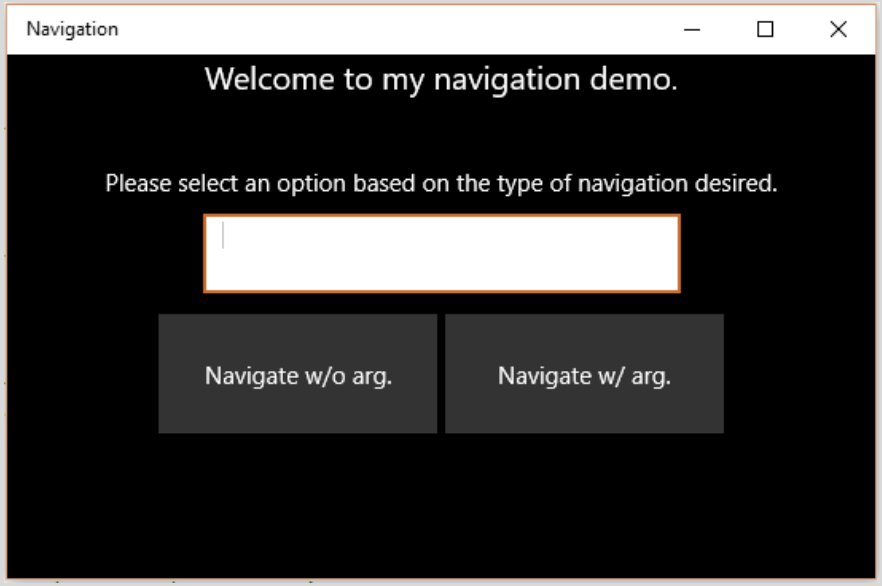
Demo: View designs

Application theme

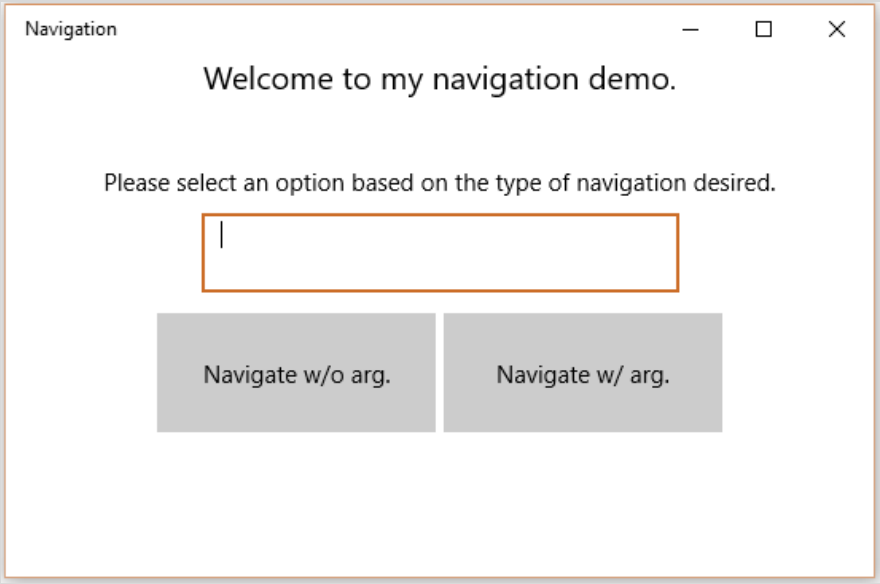
APPLICATION THEME

Available themes

DARK



LIGHT



APPLICATION THEME

System theming

BY CODE

```
this.RequestedTheme = Windows.UI.Xaml.ApplicationTheme.Dark;
this.RequestedTheme = Windows.UI.Xaml.ApplicationTheme.Light;
```

BY XAML

```
<Application
    x:Class="Navigation.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:Navigation"
    RequestedTheme="Dark">

</Application>
```

```
<Application
    x:Class="ApplicationTheme.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:ApplicationTheme"
    RequestedTheme="Light">

</Application>
```

- Themes can't be changed at runtime... Or so Microsoft says.
- Interesting links:
  https://github.com/Windows-XAML/Template10/wiki
  https://github.com/Windows-XAML/Template10/blob/master/Templates%20(Project)/Minimal/Services/SettingsServices/SettingsService.cs

Demo: Application theme

User interface design

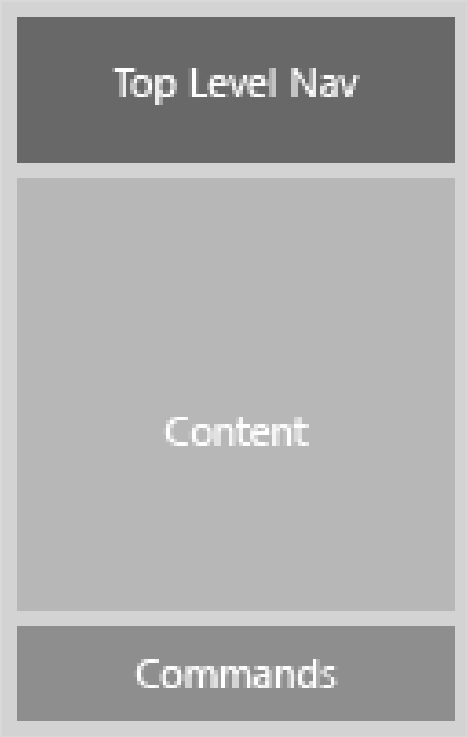
USER INTERFACE DESIGN

User controls

- More than 60 different controls out of the box.
- Highly customizable and extensible.
- Framework prepared for creating your own UI controls.
- Control list:
  - https://msdn.microsoft.com/en-us/library/windows/apps/mt185406.aspx

USER INTERFACE DESIGN

Application canvas



- Picture the app as having three different sections:
  - Navigation inside the app.
  - Actual content of the current view / app.
  - Contextual functionality for the specific state on the app and view.

Source links
https://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx

USER INTERFACE DESIGN

Command bar

- Can be placed on the top, bottom or both areas.
- Shows the icon of the buttons and can be expanded for more content.
- Contains a collection of AppBarButton
- In small devices should aim to have no more than 3/4 items.
- Group functionality into one button using flyouts and tooltips.

USER INTERFACE DESIGN

App bar

- Similar to the Command Bar control but it won't show it's content until it is expanded.
- Contains any type of content the user may want to introduce (grids, lists, buttons, etc.).

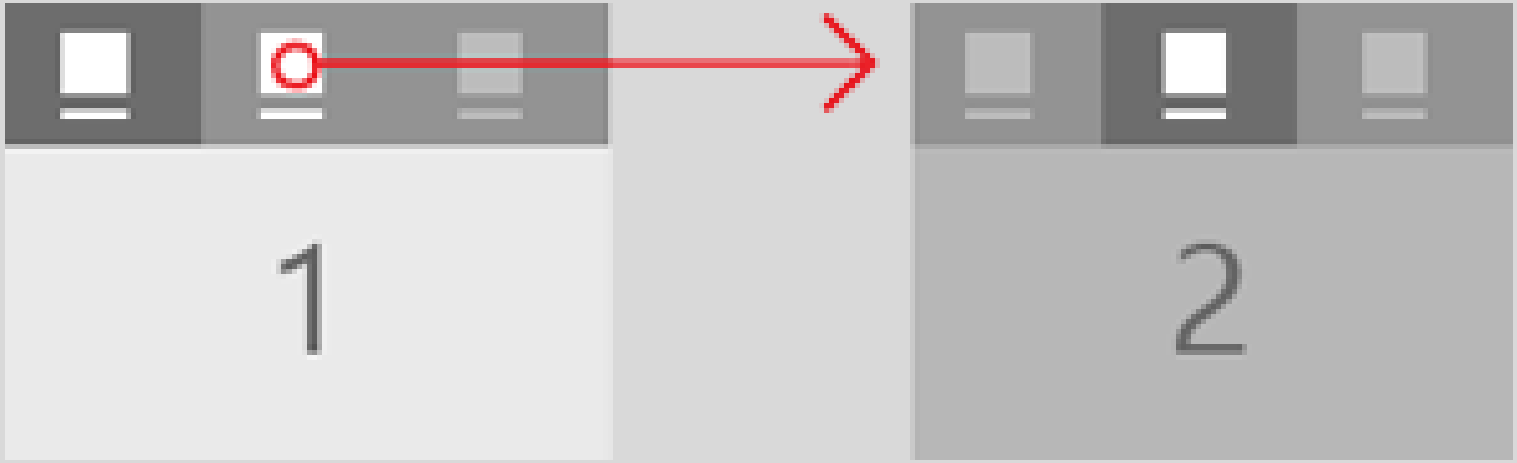
USER INTERFACE DESIGN

App bar item

- Buttons used inside App/Command bar.
- Have an Icon property with preset icons.
- Can be used with flyout menus for optimal functionality.

USER INTERFACE DESIGN

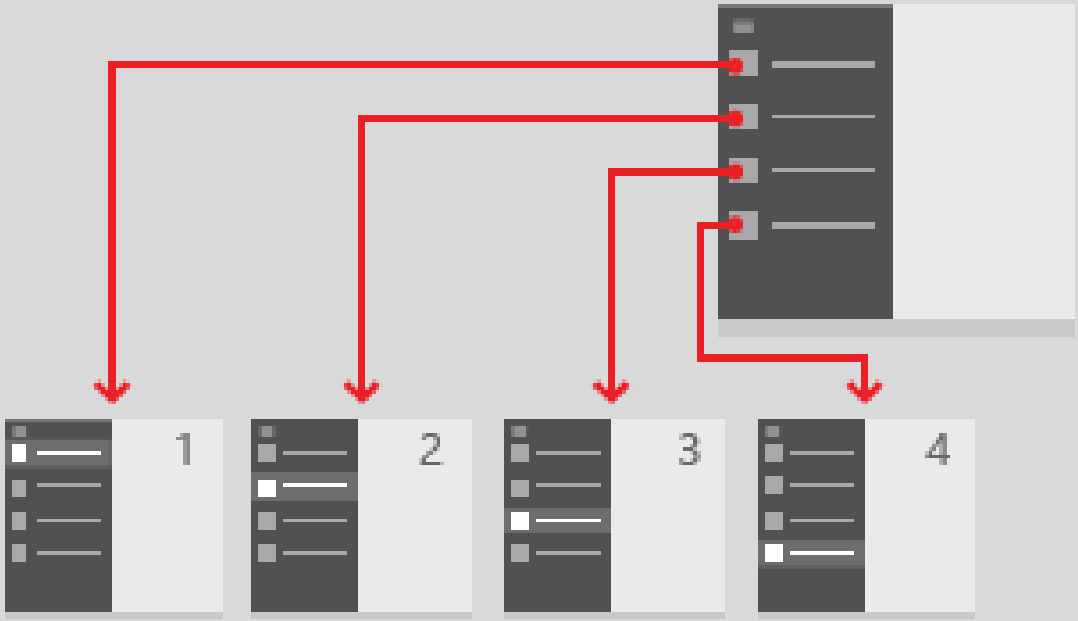
Tabs and pivot



- Use when:
  - There are 2-5 pages.
  - Users will be switching pages frequently.

USER INTERFACE DESIGN

Nav pane



- Use when:
  - Users won't be switching pages frequently.
  - The pages exist at the top level.
  - More space on the pages at the expense of slowing down navigation is not a problem.

Source link https://msdn.microsoft.com/en-us/library/windows/apps/dn997766.aspx
Nav pane template https://visualstudiogallery.msdn.microsoft.com/7284fa56-cd50-4d66-ba61-03ee77bddd19?SRC=Home

USER INTERFACE DESIGN

Hub

- Use when:
  - The user should get a preview/small taste of the content of the child page without having to navigate to it.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/dn449149.aspx

USER INTERFACE DESIGN

Master / detail

- Use when:
  - Users will be switching pages frequently.
  - The user needs to perform high-level operations (such as deleting or sorting) on individual and/or groups of items.
- Example: email inbox.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/dn997765.aspx

USER INTERFACE DESIGN

Master / detail - Side by side implementation

Available window width	Recommended style
320 epx-719 epx	Stacked
720 epx or wider	Side-by-side

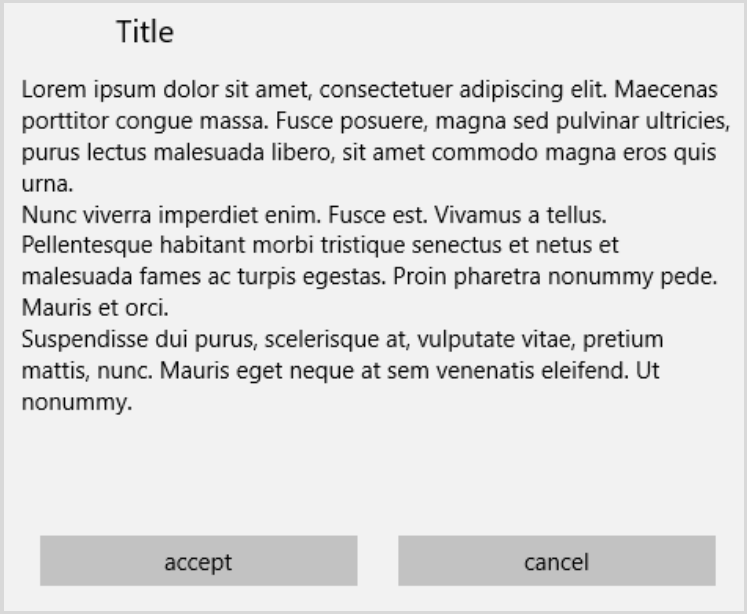
USER INTERFACE DESIGN

Master / detail - Stacked implementation

Available window width	Recommended style
320 epx-719 epx	Stacked
720 epx or wider	Side-by-side

USER INTERFACE DESIGN

- Content dialog

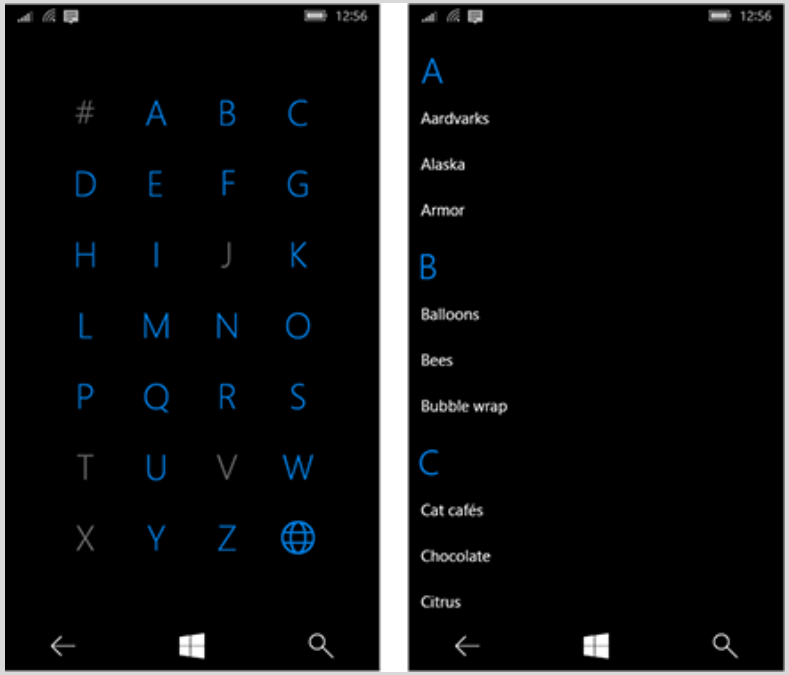


- Use when:
  - User's security has been compromised.
  - When the user is about to permanently alter or delete a valuable asset.
  - To confirm in-app purchase.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.contentdialog.aspx

USER INTERFACE DESIGN

Semantic zoom

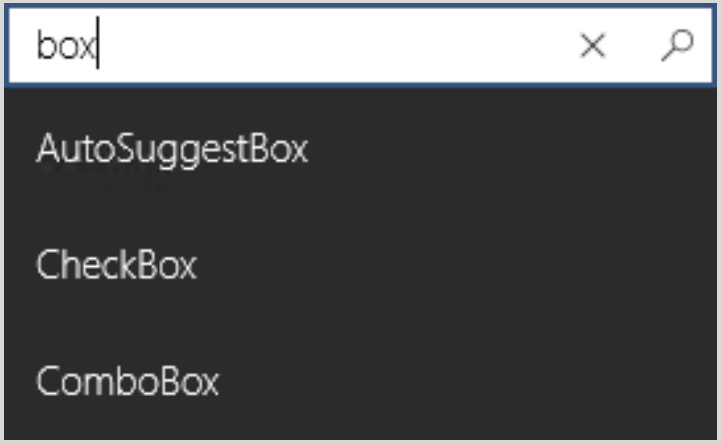


- Use when:
  - Facilitate the user to navigate inside a long list, grouping it logically.
- Never change the scope of the page as a result of the selection.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/hh465319.aspx

USER INTERFACE DESIGN

Autosuggest box



Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.autosuggestbox.aspx

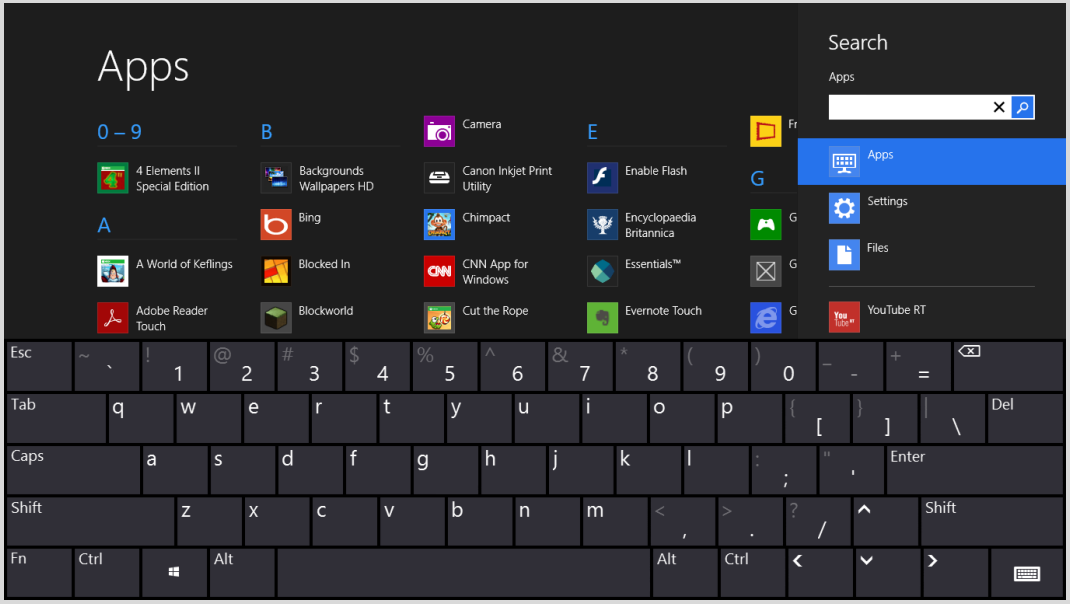
Demo: User interface design

Application interaction

APPLICATION INTERACTION

Input scope

- Dependency property.
- Indicates the expected input so that the XAML framework can display the appropriate touch keyboard.



Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.input.inputscope.aspx

APPLICATION INTERACTION

Input scope implementation

- It just a property in the input control.

```
<TextBlock x:Uid="UrlTextBlock" Grid.Column="0" Grid.Row="0"/>
<TextBox Grid.Column="1" Grid.Row="0" Height="35" MaxWidth="225" InputScope="Url"/>

<TextBlock x:Uid="EmailTextBlock" Grid.Column="0" Grid.Row="1"/>
<TextBox Grid.Column="1" Grid.Row="1" Height="35" MaxWidth="225" InputScope="EmailSmtpAddress"/>

<TextBlock x:Uid="NumberTextBlock" Grid.Column="0" Grid.Row="2"/
<TextBox Grid.Column="1" Grid.Row="2" Height="35" MaxWidth="225" InputScope="Number"/>

<TextBlock x:Uid="PasswordTextBlock" Grid.Column="0" Grid.Row="3" />
<TextBox Grid.Column="1" Grid.Row="3" Height="35" MaxWidth="225" InputScope="Password"/>

<TextBlock x:Uid="PhoneTextBlock" Grid.Column="0" Grid.Row="4"/>
<TextBox Grid.Column="1" Grid.Row="4" Height="35" MaxWidth="225" InputScope="TelephoneNumber"/>
```

APPLICATION INTERACTION

Input scope name values

Member	Value	Description
URL	1	Indicates a URI. This can include URL, File, or FTP formats.
EmailSmtpAddress	5	Input scope is intended for working with a SMTP form e-mail address (accountname@host).
Number	29	Input scope is intended for working with digits 0-9.
Password	31	Input scope is intended for working with an alphanumeric password, including other symbols, such as punctuation and mathematical symbols. Note Not supported in Windows 8 apps and Windows Phone apps.
TelephoneNumber	32	Input scope is intended for working with telephone numbers.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.input.inputscopenamevalue.aspx

APPLICATION INTERACTION

Input scope restriction

- No validation or data manipulation is performed by the input scope.

- No prevention of wrong data input is performed by the input scope.

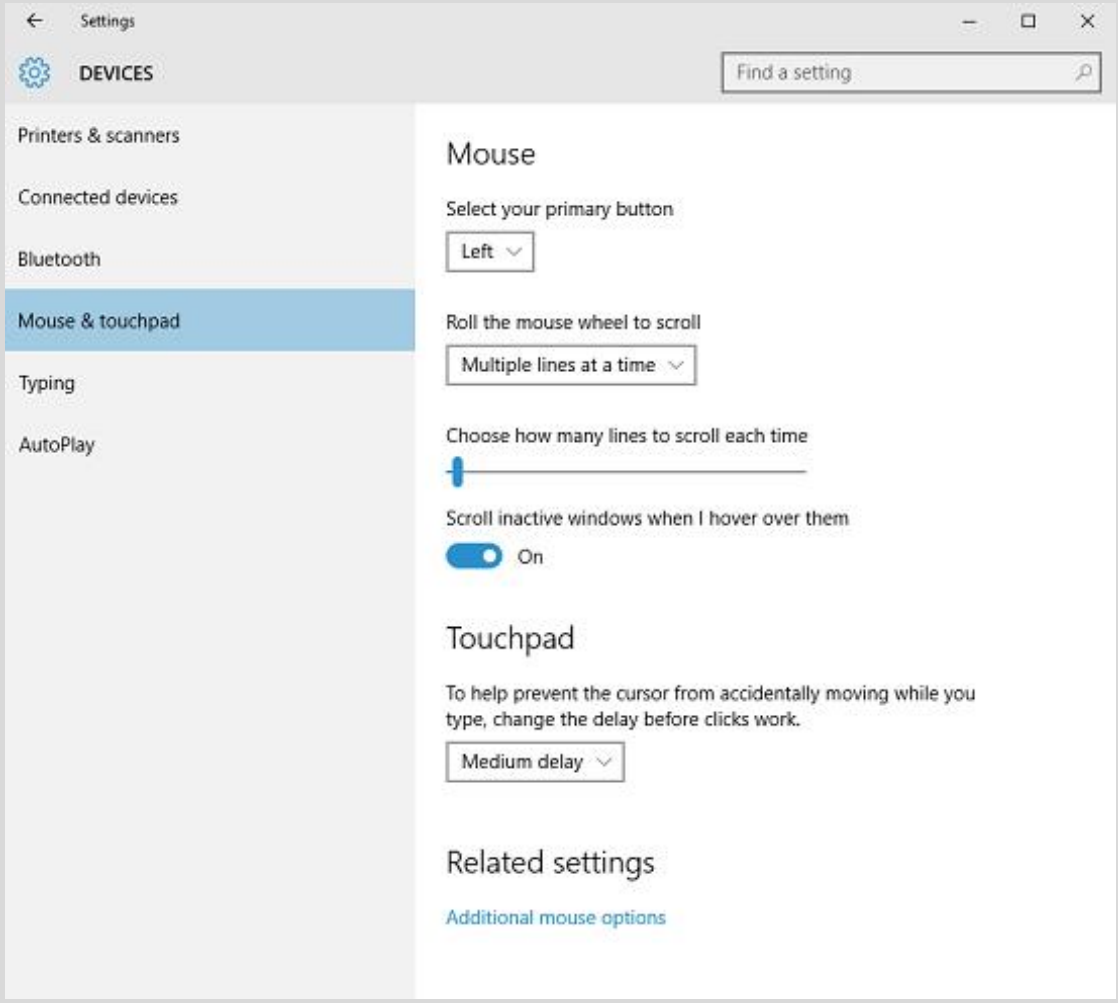
APPLICATION INTERACTION

Touchpad interaction

- Allows for user interaction by using hand gestures
- Two types available:
  - Standard touchpad
  - Windows precision touchpad
    - They enable the system to handle certain aspects of the touchpad experience natively, such as finger tracking and palm detection, for a more consistent experience across devices.
- The input data provided by the touchpad sensor can be:
  - Interpreted as a physical gesture for direct manipulation of one or more UI elements (such as panning, rotating, resizing, or moving). In contrast, interacting with an element through its properties window or other dialog box is considered indirect manipulation.
  - Recognized as an alternative input method, such as mouse or pen.
  - Used to complement or modify aspects of other input methods, such as smudging an ink stroke drawn with a pen.

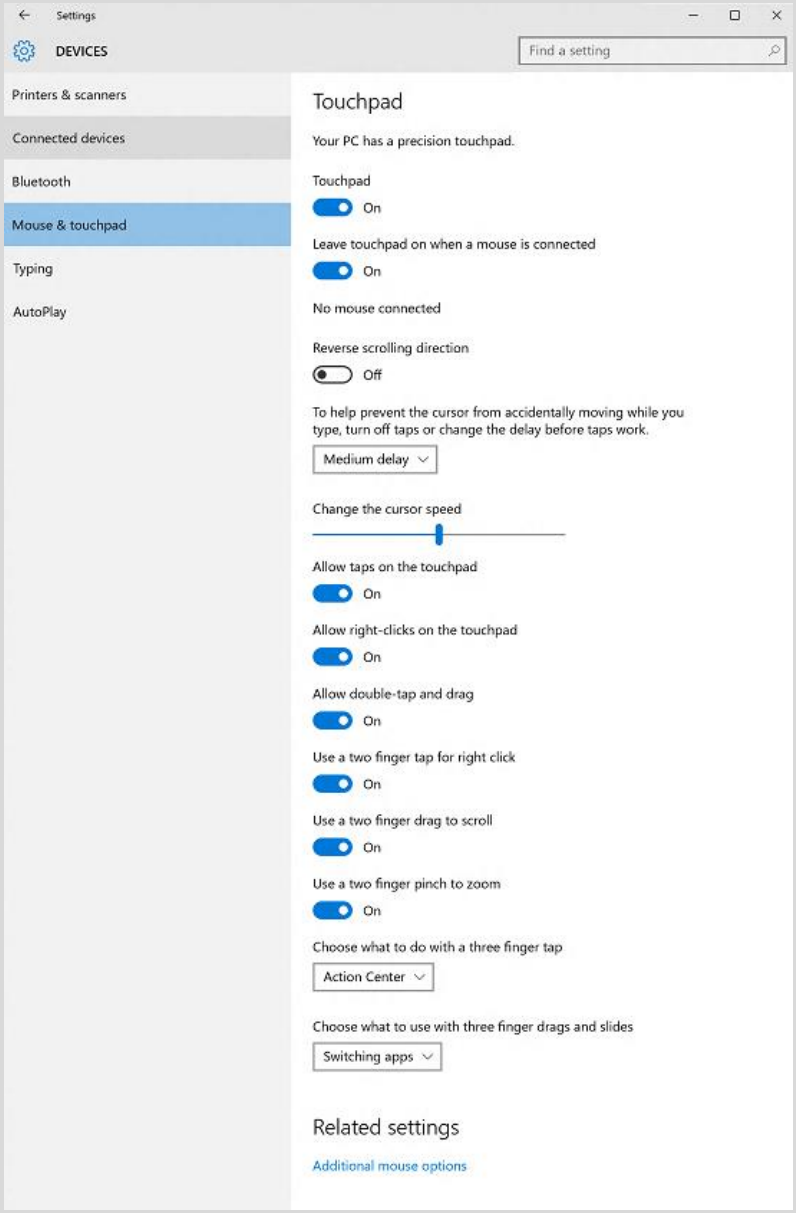
APPLICATION INTERACTION

Touchpad interaction - Standard



APPLICATION INTERACTION

Touchpad interaction - Precision



APPLICATION INTERACTION

Touchpad interaction - Precision

Term	Description
Three-finger tap	User preference to search with **Cortana** or show **Action Center**.
Three finger slide	User preference to open the virtual desktop Task View, show Desktop, or switch between open apps.
Single finger tap for primary action	Use a single finger to tap an element and invoke its primary action (such as launching an app or executing a command).
Two finger tap to right-click	Tap with two fingers simultaneously on an element to select it and display contextual commands.
Two finger slide to pan	Slide is used primarily for panning interactions but can also be used for moving, drawing, or writing.
Pinch and stretch to zoom	The pinch and stretch gestures are commonly used for resizing and Semantic Zoom.
Single finger press and slide to rearrange	Drag an element.
Single finger press and slide to select text	Press within selectable text and slide to select it. Double-tap to select a word.
Left and right click zone	Emulate the left and right button functionality of a mouse device.

APPLICATION INTERACTION

Resources

Win RT validation:

http://blog.jerrynixon.com/2014/07/lets-code-handling-validation-in-your.html

User interaction:

https://msdn.microsoft.com/en-us/library/windows/apps/mt185599.aspx

Basic input reference:

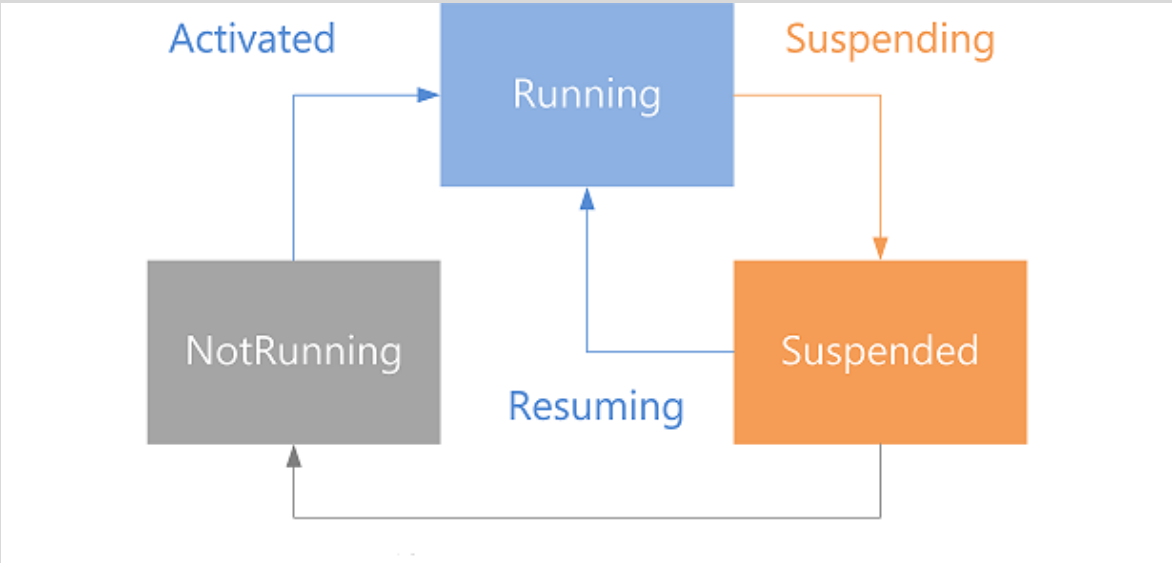
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.input.pointerpoint.aspx
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.devices.input.keyboardcapabilities.aspx
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.devices.input.mousecapabilities.aspx
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.devices.input.touchcapabilities.aspx
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.uielement.manipulationmode.aspx
- https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.input.gesturerecognizer.aspx

Demo: Application interaction

App lifecycle

APP LIFECYCLE

Lifecycle states

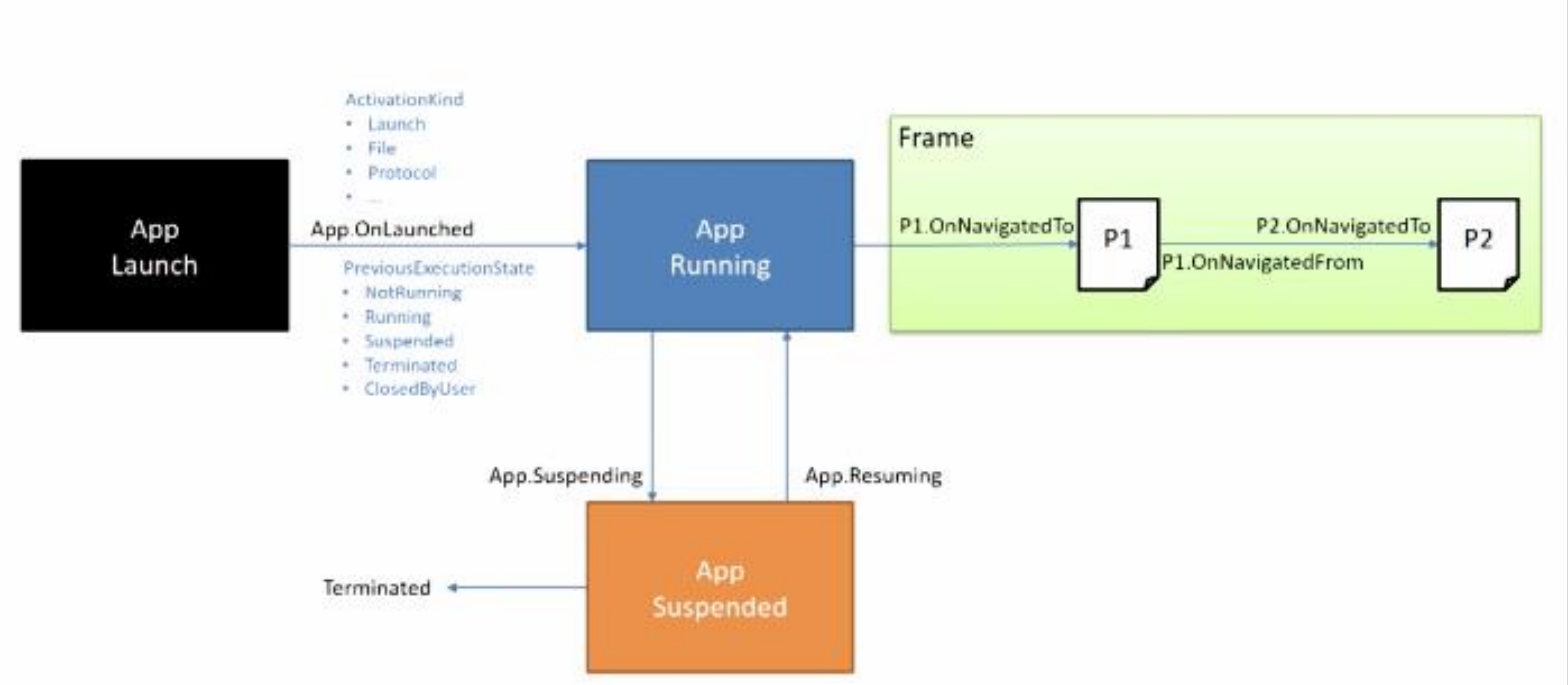


Source links:

https://msdn.microsoft.com/en-us/library/windows/apps/mt243287.aspx

APP LIFECYCLE

Cycle overview



APP LIFECYCLE

Activation kind

Member	Value	Description
Launch | launch	0	The user launched the app or tapped a content tile.
Search | search	1	The user wants to search with the app.
ShareTarget | shareTarget	2	The app is activated as a target for share operations.
File | file	3	An app launched a file whose file type this app is registered to handle.
Protocol | protocol	4	An app launched a URI whose scheme name this app is registered to handle.
FileOpenPicker | fileOpenPicker	5	The user wants to pick files that are provided by the app.
FileSavePicker | fileSavePicker	6	The user wants to save a file and selected the app as the location.
CachedFileUpdater | cachedFileUpdater	7	The user wants to save a file that the app provides content management for.
ContactPicker | contactPicker	8	The user wants to pick contacts.
Device | device	9	The app handles AutoPlay.

- There's a total of 31 activation kinds, allowing us to fully customize the launch processes of the app.
- Full list can be found on the following link:

https://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.activation.activationkind.aspx

APP LIFECYCLE

Suspension

- Suspending event
  - An event if fired when the OS needs resources and the app is not active (being used by the user).
  - Happens often on devices with low resources such as smartphones.
  - Plan for the future and always respond to the event accordingly (don't focus on one device).

- Application must answer within 5 seconds.
  - The app can request a deferral to the OS, which may be accepted or denied.

- Serialize state, options and data.
  - When suspended, the app may terminate at any time without notification (same for uninstall).
  - Serialize often, when suspended the app may not have enough time before being terminated!

APP LIFECYCLE

Restore state

The Microsoft guidelines dictate:

- The user expectation for the app is to have all the info/navigation it had when it was terminated.
  - The value of PreviousExecutionState property is *Terminated*.
- Do not restore data when the app loads for the first time or was closed by the user.
  - The value of PreviousExecutionState property is *NotRunning* or *ClosedByUser*.
- **Evaluate your own app and act accordingly!**

APP LIFECYCLE

Storage options

LOCAL

- Local application data storage
  - Default container assigned to every app
- Roaming data store
  - Replication between devices is not instant
  - Not all file types are allowed
- Temporary data store
  - Volatile
- Local database
  - Only Sqlite is supported in some devices

REMOTE

- OneDrive
- Azure storage
- Azure SQL
- Other non-Microsoft web services

APP LIFECYCLE

Local and roaming data highlights

- Local and roaming data are best used for state serialization.
- Always keep in mind there's a size limit for each one:
  - Local: up to 255 characters for the key, up to 8K bytes in size for the setting (composite settings up to 64K).
  - Roaming: up to 255 characters for the key, up to 8K bytes in size for the setting (composite settings up to 64K).

Local settings info: https://msdn.microsoft.com/en-us/library/windows/apps/windows.storage.applicationdata.localsettings.aspx

Roaming settings info: https://msdn.microsoft.com/en-us/library/windows/apps/windows.storage.applicationdata.roamingsettings.aspx

Demo: App lifecycle

Speech recognition

SPEECH RECOGNITION

Cortana integration

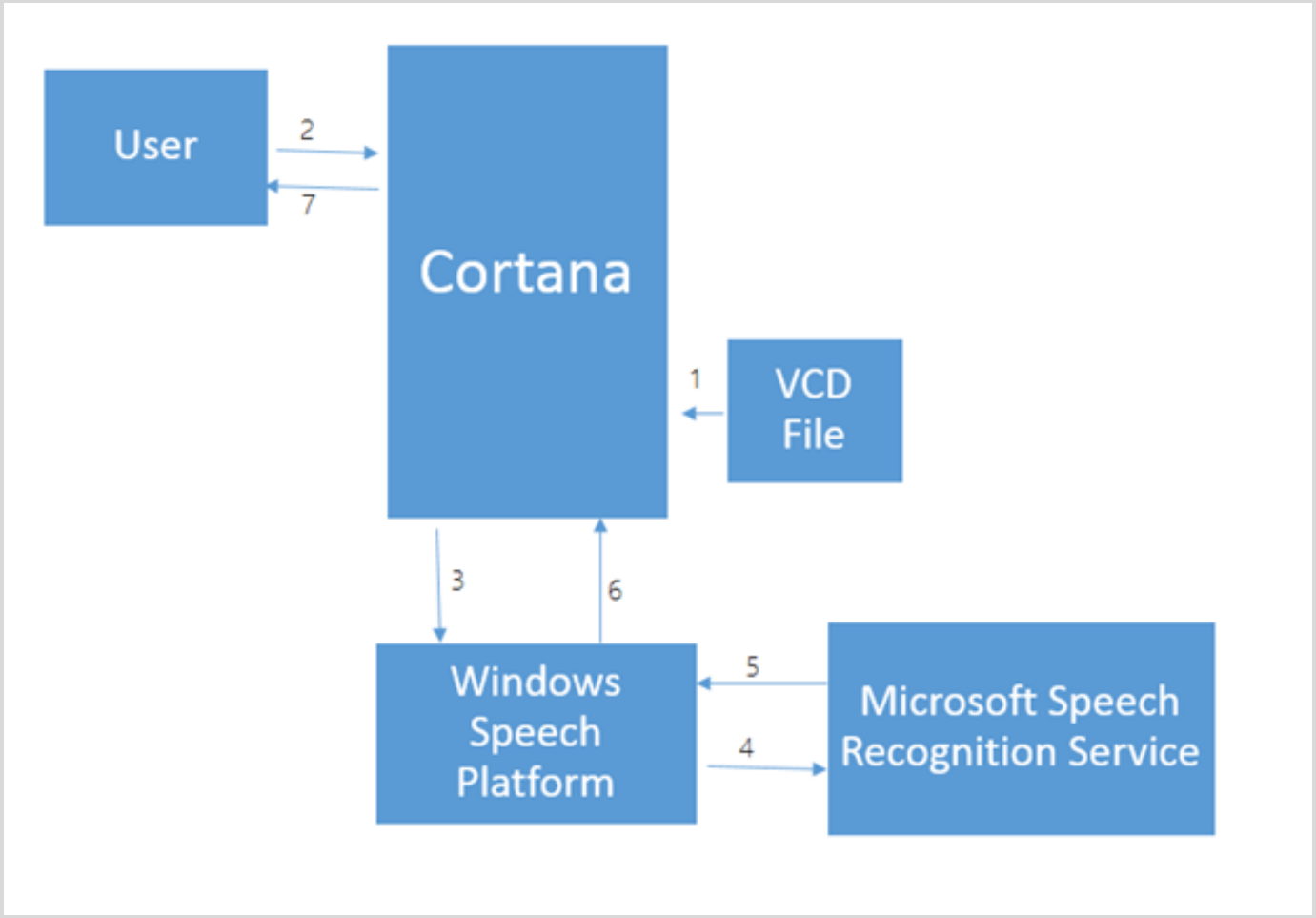
- Integrate the app with Cortana by registering the VoiceDefinitions XML file within the application:
  - Declare all voice commands in an XML file calles VoiceDefinitions.
  - Register the file within the application in the *VoiceCommandDefinitionManager*:
    - `await VoiceCommandDefinitionManager.InstallCommandDefinitionsFromStorageFileAsync(file);`

- After executing the application for the first time, we can start it up again with Cortana.

- Every voice command declared on the VoiceDefinitions will then be recognized by Cortana.

SPEECH RECOGNITION

Cortana workflow



SPEECH RECOGNITION

Voice definitions file - overview

```
<?xml version="1.0" encoding="utf-8" ?>
<VoiceCommands xmlns="http://schemas.microsoft.com/voicecommands/1.2">
  <CommandSet xml:lang="en-us" Name="SampleAppCommandSet_en-us">
    <CommandPrefix>Sample App,</CommandPrefix>
    <Example>Launch</Example>
    <Command Name="LaunchApp">
      <Example>launch</Example>
      <ListenFor>launch</ListenFor>
      <Feedback>Opening your speech recognition app</Feedback>
      <Navigate />
    </Command>
  </CommandSet>
</VoiceCommands>
```

SPEECH RECOGNITION

Listen for attribute

- Can have multiple values (declare more than one ListenFor attribute).

- Values RequireAppName:
  - BeforeOrAfterPhrase
  - BeforePhrase
  - AfterPhrase
  - ExplicitlySpecified

SPEECH RECOGNITION

Voice definitions file - attributes

- CommandSet
  - Declares the language of the file / command.
- CommandPrefix
  - Allows for Cortana to identify the app.
  - Every command must be given with this prefix.
- Example
  - Default command.
- Command
  - Process that Cortana may perform.
- Example (inside Command)
  - Word or phrase that responds to the command given to Cortana.
- ListensFor
  - Process that Cortana will recognize.
- Feedback
  - Feedback that Cortana gives the user when executing the command.
- Navigate
  - Process that Cortana will perform.

SPEECH RECOGNITION

Resources

Speech design guidelines
https://msdn.microsoft.com/en-us/library/windows/apps/dn596121.aspx

Cortana guidelines
https://msdn.microsoft.com/en-us/library/windows/apps/dn974233.aspx

Voice command design info
https://msdn.microsoft.com/en-us/library/windows/apps/dn722331.aspx

Demo: Speech recognition

Live tiles and notifications

LIVE TILES AND NOTIFICATIONS

Adaptive tiles

- New to Windows 10.
- Allow the developer to create their own templates via XML.
- Can be used on as both primary and secondary tiles.

LIVE TILES AND NOTIFICATIONS

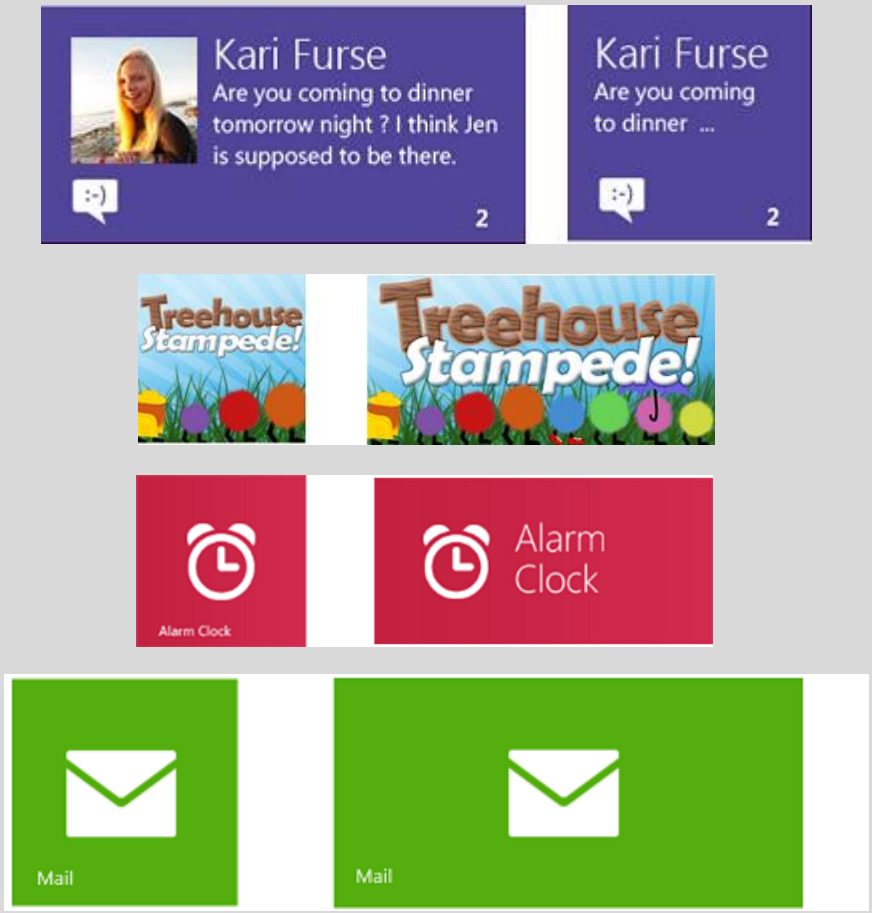
Secondary tiles

- Provide a deep linking into some of the app specific functions.
- Can pin as many as you like.
- Each secondary tile is unique and can show different content / content templates.
- Allows navigation to a specific functionality of the application.

LIVE TILES AND NOTIFICATIONS

Primary tiles

- Primary interaction with the app.
- Allow the app to show relevant info to the user.
- Throw the OnActivated event every time they are called.



LIVE TILES AND NOTIFICATIONS

Tile updates

- Tiles can get updated with custom content.

- Update process can be perform:
  - While the app is running (not suspended).
  - Background tasks (even when suspended).
  - Scheduled pull of URI (check a service and show data).

- One update every 30 minutes.

LIVE TILES AND NOTIFICATIONS

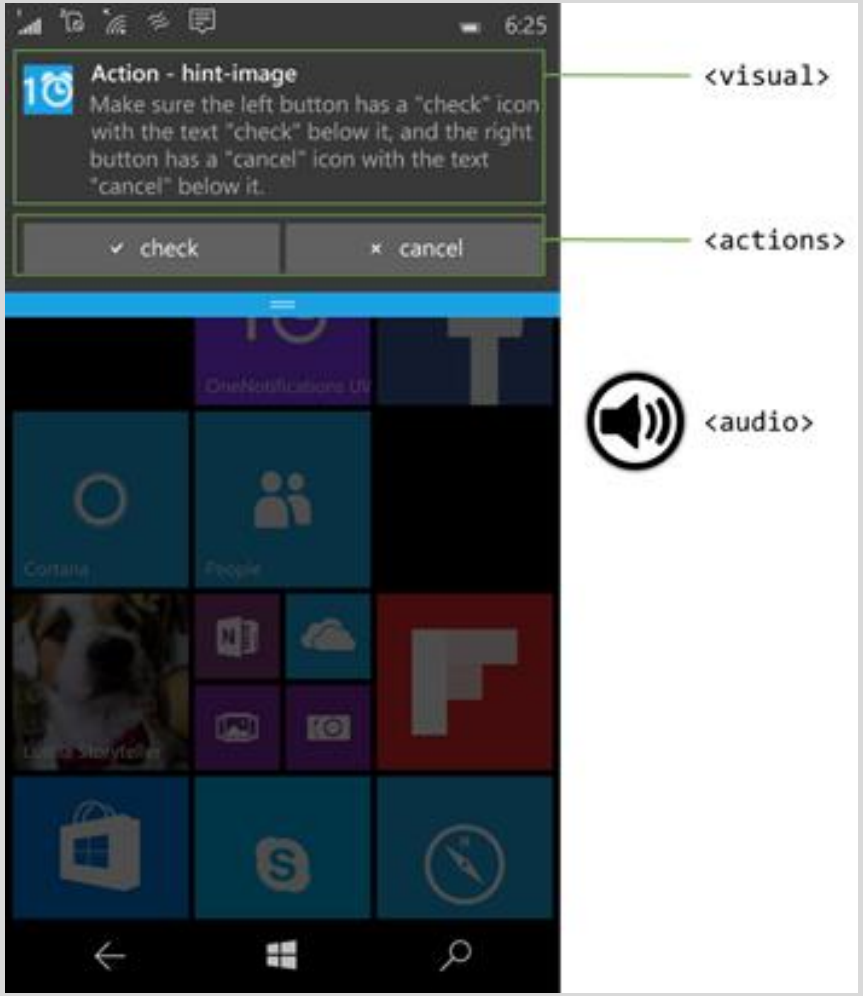
Badges and glyphs

- Badges are the numeric notification on the live tile.
  - A number of 0 clears it from the badge.
  - Numbers higher than 99 are displayed as 99+.
  - Keep it under 50!
- Glyphs are the little icons that can be shown.
- Both can be used in the tile and/or the lock screen.

LIVE TILES AND NOTIFICATIONS

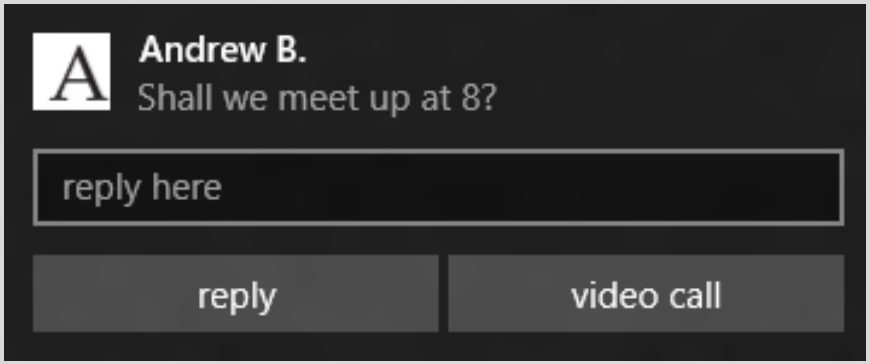
Toasts

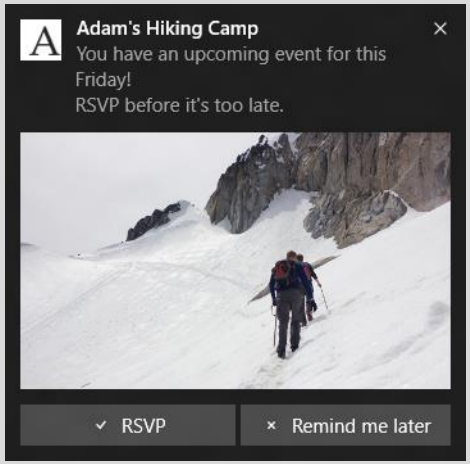
- Provide a different level of interaction between the app and the user:
  - Actions
  - Reminders
  - Information
  - …
- Based on XML.
- Extensions to create by using code.

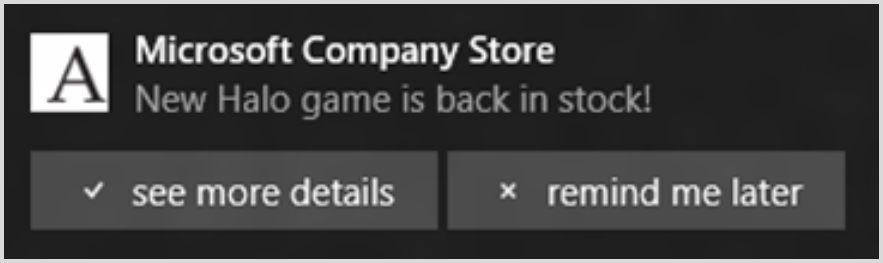


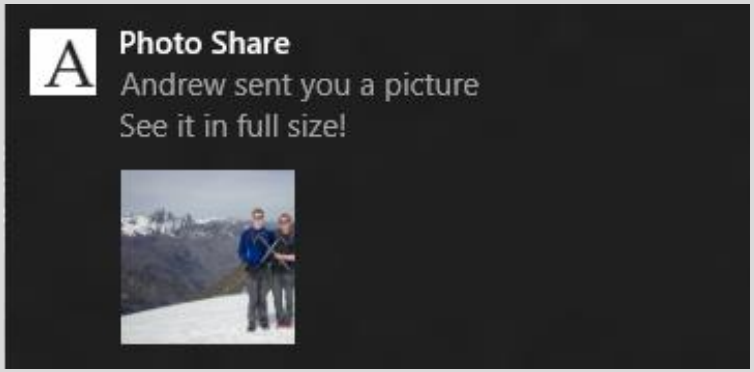
LIVE TILES AND NOTIFICATIONS

Toasts example









LIVE TILES AND NOTIFICATIONS

Useful resources

- Tile guidelines

https://msdn.microsoft.com/en-us/library/windows/apps/hh465403.aspx?f=255&MSPPError=-2147217396

- Badge overview

https://msdn.microsoft.com/en-us/library/windows/apps/hh779719.aspx?f=255&MSPPError=-2147217396

- Adaptive tiles

http://blogs.msdn.com/b/tiles_and_toasts/archive/2015/06/30/adaptive-tile-templates-schema-and-documentation.aspx

- Notifications visualizer

https://www.microsoft.com/en-us/store/apps/notifications-visualizer/9nblggh5xsl1

- Secondary tiles msdn

https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.startscreen.secondarytile.aspx

- Toast

http://blogs.msdn.com/b/tiles_and_toasts/archive/2015/09/07/10625426.aspx

Demo: Live tiles and notifications

Background agent

BACKGROUND AGENT

Intro

- Allow the app to run code without having the application running on the system.

- Can execute:
  - Synchronous processes
  - Asynchronous processes
    - Need to ask for a deferral first.

- Background tasks are public and sealed classes that implement the *IBackgroundTask* interface.

- Contain a unique entry point called *Run*.

BACKGROUND AGENT

Info

- Need to be registered before being executed.
- The same task can be registered more than once
  - Before registering, always check if the system already has it.
  - If it is registered more than once, it will run more than once every time it is requested!
- Before registration, the system has to allow for such a process to take place
  - Requesting registration does not guarantee access to the functionality.
- For all device families except desktop, if the device becomes low on memory, background tasks may be terminated. If an out of memory exception is not surfaced, or the app does not handle it, then the background task will be terminated without warning and without raising the OnCanceled event. This helps to ensure the user experience of the app in the foreground. Your background task should be designed to handle this scenario.

BACKGROUND AGENT

Advises

- Background tasks are limited to 30 seconds of wall-clock usage.
- Handle background task progress, completion, and cancellation.
- Do not rely on user interaction in background tasks.
- Do not display UI other than toasts, tiles, and badge updates from the background task.
- Use persistent storage to share data between the background task and the app.

BACKGROUND AGENT

Triggers

Trigger name	Description
InternetAvailable	The Internet becomes available.
NetworkStateChange	A network change such as a change in cost or connectivity occurs.
OnlineIdConnectedStateChange	Online ID associated with the account changes.
SmsReceived	A new SMS message is received by an installed mobile broadband device.
TimeZoneChange	The time zone changes on the device (for example, when the system adjusts the clock for daylight saving time).

BACKGROUND AGENT

Conditions

Condition name	Description
InternetAvailable	The Internet must be available.
InternetNotAvailable	The Internet must be unavailable.
SessionConnected	The session must be connected.
SessionDisconnected	The session must be disconnected.
UserNotPresent	The user must be away.
UserPresent	The user must be present.

BACKGROUND AGENT

Steps

- Create the task
  - Remember to inherit from *IBackgroundTask*.
  - The class needs to be public and sealed.
- Ask the OS permission to register.
- Register the task.
  - Make sure to use the correct trigger.
  - Remember to specify the condition if needed.
- Update the app manifest with the task list.
- Check that the registration was successful.

BACKGROUND AGENT

Useful resources

- Guidelines
  https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt187310.aspx

- Intro
  https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt299103.aspx

- Usage:
  https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt299100.aspx

- Debug:
  https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt299101.aspx

Demo: Background agent

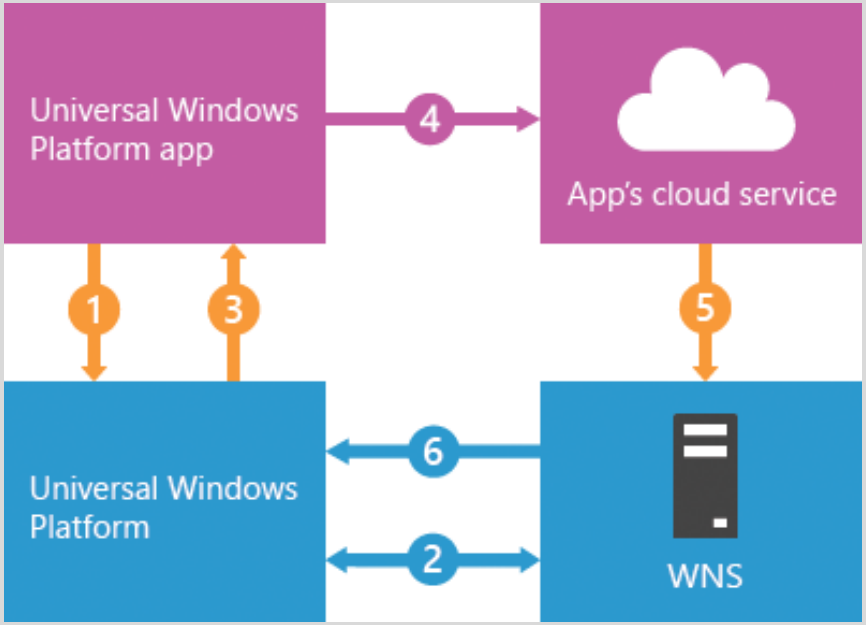
Demo: Intro to the push system

INTRO TO THE PUSH SYSTEM

Windows Notification Services

- The Windows Push Notification Services (WNS) enables third-party developers to send toast, tile, badge, and raw updates from their own cloud service.

- WNS provides a mechanism to deliver updates to your users in a dependable and power-efficient way, even when your app is not running.



INTRO TO THE PUSH SYSTEM

Usage

- The push delivery method allows users to receive notifications from your app at any time, even when the app isn't running.

- Push notifications are an excellent option if you want your app to share:
  - Real-time updates (like sports scores during the game)
  - Content that's generated at unpredictable times (such as breaking news, incoming emails, or social media updates)

- **Be sensitive of the resource cost!**

- There are four mechanisms that an app can use to deliver a notification:
  - Local
  - Scheduled
  - Periodic
  - Push

INTRO TO THE PUSH SYSTEM

Delivery method overview I

Delivery method	Use with	Description	Examples
Local	Tile, Badge, Toast	A set of API calls that send notifications while your app is running, directly updating the tile or badge, or sending a toast notification.	• A music app updates its tile to show what's "Now Playing". • A game app updates its tile with the user's high score when the user leaves the game. • A badge whose glyph indicates that there's new info int the app is cleared when the app is activated.
Scheduled	Tile, Toast	A set of API calls that schedule a notification in advance, to update at the time you specify.	• A calendar app sets a toast notification reminder for an upcoming meeting.

INTRO TO THE PUSH SYSTEM

Delivery method overview II

Periodic	Tile, Badge	Notifications that update tiles and badges regularly at a fixed time interval by polling a cloud service for new content.	• A weather app updates its tile, which shows the forecast, at 30-minute intervals. • A "daily deals" site updates its deal-of-the-day every morning. • A tile that displays the days until an event updates the displayed countdown each day at midnight.
Push	Tile, Badge, Toast, Raw	Notifications sent from a cloud server, even if your app isn't running.	• A shopping app sends a toast notification to let a user know about a sale on an item that they're watching. • A news app updates its tile with breaking news as it happens. • A sports app keeps its tile up-to-date during an ongoing game. • A communication app provides alerts about incoming messages or phone calls.

INTRO TO THE PUSH SYSTEM

Useful resources

- Guidelines
  https://msdn.microsoft.com/en-us/library/windows/apps/hh761462.aspx

- Intro
  https://msdn.microsoft.com/en-us/library/windows/apps/mt187203.aspx

- Implementing in XAML
  https://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh868244.aspx

Web view

WEB VIEW

Control

- A web view control embeds a view into your app that looks and behaves like Microsoft Edge. Hyperlinks can also appear and function in a web view control.

- Make sure that the website loaded is formatted correctly for the device and uses colors, typography, and navigation that are consistent with the rest of your app.

- Input fields should be appropriately sized. Users may not realize that they can zoom in to enter text.

- If a web view doesn't look like the rest of your app, consider alternative controls or ways to accomplish relevant tasks. If your web view matches the rest of your app, users will see it all as one seamless experience.

Source link:
https://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.webview.aspx

Demo: Web view

Vibration

VIBRATION

Presentation

- UWP functionality that targets a Windows Phone device.
- The phone can vibrate up to 5 seconds.
- The user can disable the functionality
  - Don't rely on it for critical notifications.
- Testable only on a physical device.

VIBRATION

Remarks

- The **Mobile** extension is needed in order to work.

- An app that's running in the background can't vibrate the phone. If your code tries to use vibration while the app is running in the background, nothing happens, but no exception is raised. If you want to vibrate the phone while your app is running in the background, you have to implement a toast notification.

- When called to the *Vibrate* function, an argument is asked representing the seconds to vibrate.
  - *Accepted values for the duration argument are from 0 to 5 seconds; any greater or lesser value raises an ArgumentException error.*

Demo: Vibration

Maps and location

MAPS AND LOCATION

Overview

- The UWP allows the developer to display maps, use map services, find a location and set up a geofence in your app.
- Implementation
  - Implement all this functionality inside your app
  - Use the Windows Maps app by calling it from inside your app

MAPS AND LOCATION

Guidelines

- Don't access the location functionality directly unless it is essential for the app
  - If not required, allow the user to decide when to start using it.
- Always display feedback
  - Progress bar or ring while waiting for the location data.
  - Appropriate error messages when location services are disabled or unavailable.
- Always clear the cache when the user disables location capabilities.
- Tell the user how location data will be used.
- Provide UI to enable users to manually refresh their location.
- Use a one time location request when needed (if your app does not need to continually know the location).

MAPS AND LOCATION

Geolocation troubleshooting

- Typical problems for which the geolocator may not work:
  - Requires access to the location capability
  - Location for this device is turned off (not applicable in Windows 10 Mobile)
  - The location services setting, Location, is turned off
  - Under Choose apps that can use your location, your app is set to off

MAPS AND LOCATION

Geolocator class

- Provides access to the current geographic location.
- Allows to easily react to changes in the user's location.
- In order to work, requires access to the geolocation system by the user
  - The user has to declare, specifically; that he allows access to the location system.

MAPS AND LOCATION

Useful resources

- Guidelines
  https://msdn.microsoft.com/en-us/library/windows/apps/hh465148.aspx

- Overview
  https://msdn.microsoft.com/en-us/library/windows/apps/mt219699.aspx

- Map launcher
  https://msdn.microsoft.com/en-us/library/windows/apps/mt228341.aspx

Demo: Maps and location

Launchers

LAUNCHERS

Overview

- Launchers allow to start (launch) an application from within our own app.

- Highly reworked on Windows 10.

- Allows to easily send and receive information between apps.

LAUNCHERS

Remarks

- Launchers use an *URI* parameter with a protocol name to launch the app
  - Protocols are not unique!

```
var testAppUri = new Uri("test-app2app:"); // The protocol handled by the launched app
```

  - The protocol for the app that **can be** launched needs to be declared on its AppXManifest file as a **Protocol** declaration.

  - If the system has two or more apps with the same protocol, a list showing each app is shown to the user so that he will choose which one to launch.

- Always check the reserved file and URI scheme names before registering yours
  - https://msdn.microsoft.com/en-us/library/windows/apps/mt210951.aspx

LAUNCHERS

Steps

- To prepare your app to be launch-able you need to:
  - Register a new **Protocol** on the Declarations of the appXManifest file.
  - Override the *OnActivated* method of the App class to respond accordingly.
  - Override the *OnNavigatedTo* method
    - The *NavigationEventArgs* contain the data passed to the app.
  - Create, if needed; a method to return to the caller app.

LAUNCHERS

Data sharing

- Allows to easily send and receive information between apps
  - Limited to 100KB.
  - If needed more, the token system provided by the *SharedStorageAccessManager* can be used
    - Enables an app to share a file with another app by passing a token as part of a Uri activation. The target app redeems the token to get the file shared by the source app.
    - https://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.datatransfer.sharedstorageaccessmanager.aspx

LAUNCHERS

Useful resources

- Overview
  https://msdn.microsoft.com/en-us/library/windows/apps/mt227652.aspx

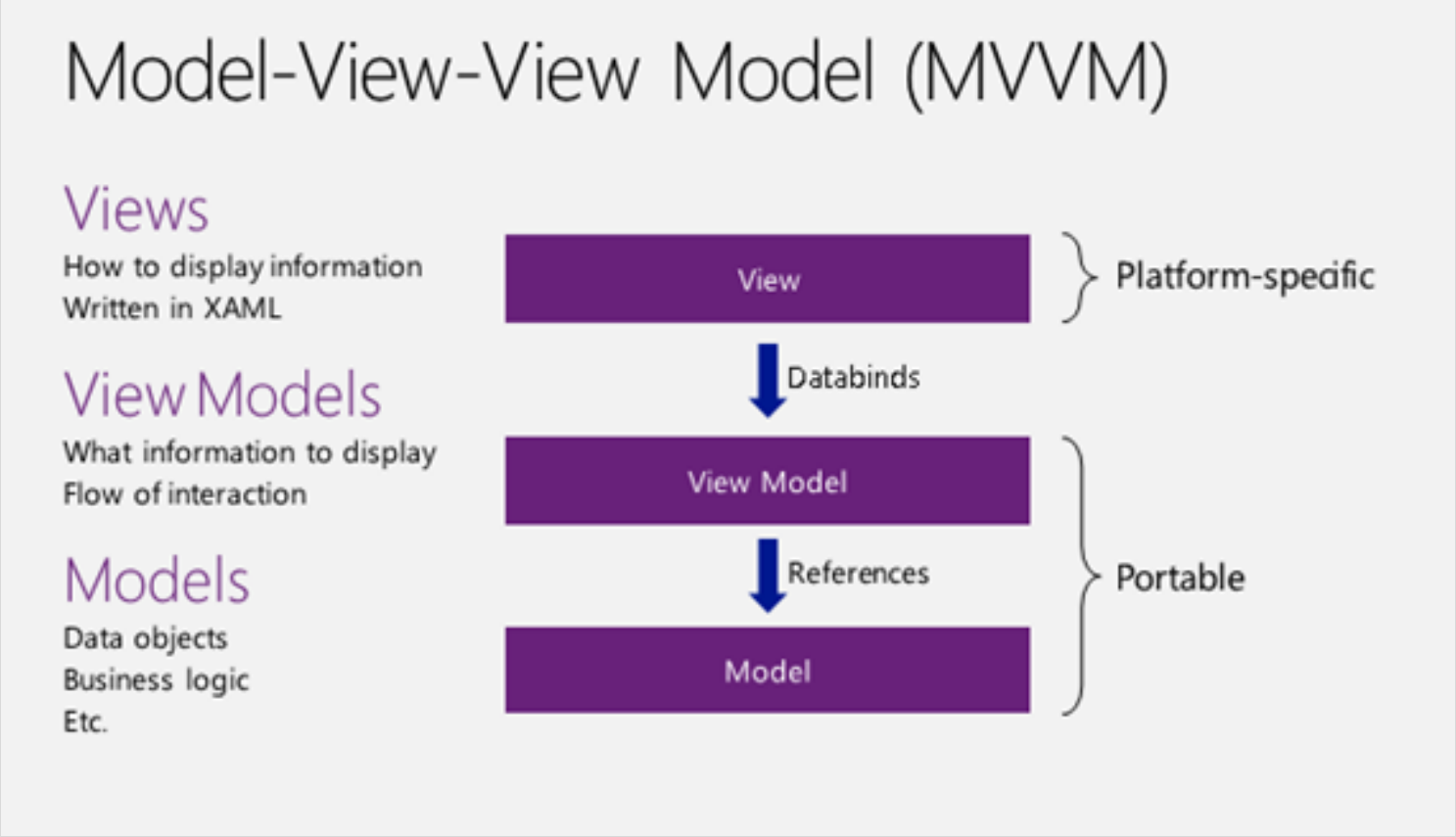
- Map launcher
  https://msdn.microsoft.com/en-us/library/windows/apps/mt228341.aspx

Demo: Launchers

Intro to MVVM

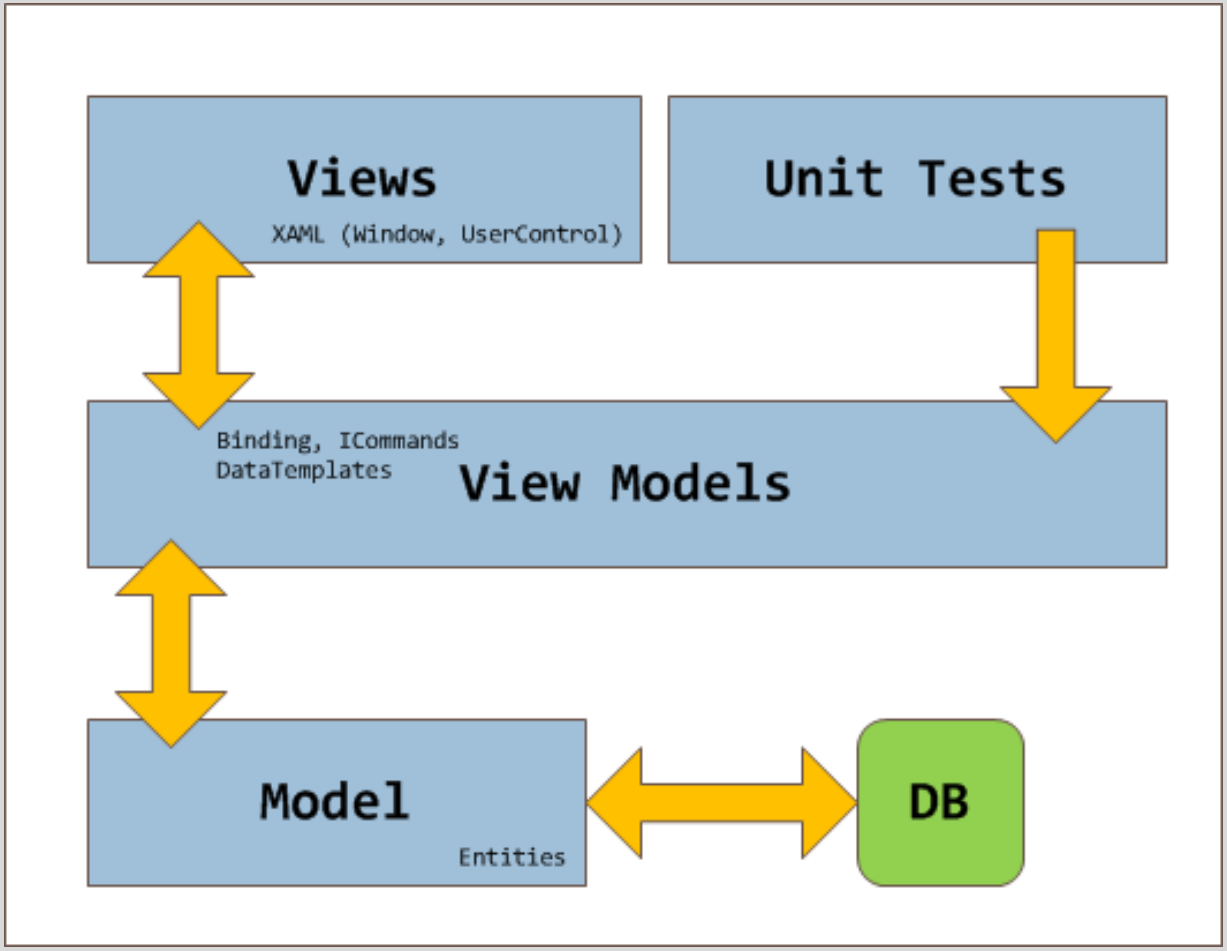
INTRO TO MVVM

Pattern



INTRO TO MVVM

Typical design



INTRO TO MVVM

Frameworks

- Usually aimed at complex LOB applications.
- There are many frameworks that help with the MVVM implementation:
  - Caliburn.Micro
  - MVVM Light
  - Prism
  - Template 10

INTRO TO MVVM

Frameworks - Caliburn micro

- Open source.
- Very lightweight.
- Has a learning curve.
- Supports WPF, SL, WinRT and UWP.

Source link
http://caliburnmicro.com/
https://github.com/Caliburn-Micro/Caliburn.Micro/commit/9406e4347cd5cccebfcf2e0f48a2f18c10fb608a

INTRO TO MVVM

Frameworks - Mvvm light

- Can be considered a toolkit.
  - Can use parts of the toolkit instead of modelling the whole app around it.
- Provides support for Xamarin.
- Simple to use.
- Supports WPF, SL, WinRT and UWP.
- Rich in documentation and examples.

Source link
http://www.mvvmlight.net/
https://mvvmlight.codeplex.com/

INTRO TO MVVM

Frameworks - Prism

- Originally owned by Microsoft pattern and practices.
- Now open source.
- Versions
  - 5 - WPF/Silverlight
  - Prism for Windows Runtime - Windows 8
  - 6 - UWP and Xamarin

Source link
http://pnpmvvm.codeplex.com/
https://github.com/PrismLibrary/Prism/releases

INTRO TO MVVM

Frameworks - Template 10

- Open source.
- Created by Microsoft evangelist Jerry Nixon.
- Not a framework but a UWP template.
- Built with UWP in mind
  - UWP conventions are present.
- Visual studio extension.
- Currently targets SDK v10240 only.

Source link
https://github.com/Windows-XAML/Template10/wiki

INTRO TO MVVM

Bindings

- Two types of data binding:
  - **Standard Binding**
    - Default XAML binding method.
    - One way by default.
    - Binding resolved at runtime.
    - Has to bind a DataContext value for the page or every control.
  - **Compiled Binding**

INTRO TO MVVM

Compile binding

- New in Windows 10.
- Syntax: *{x:Bind}*
- Bindings are resolved at compile time.
- Strongly typed.
- Limited support for **built-in** converters and intellisense.
- Scoped to Page level properties rather than Page.DataContext
- One time mode by default.
- Can't be created from code behind.

INTRO TO MVVM

Compile binding - Examples

Simple binding – OneTime is the default

```
– <TextBlock Text="{x:Bind ObservableString}" />
```

Two Way Binding

```
– <TextBox Text="{x:Bind ObservableString, Mode=TwoWay}" />
```

Collection Binding

```
– <GridView.ItemTemplate>
        <DataTemplate x:DataType="local:BookTemplate">
            <TextBlock Text="{x:Bind Title}"/>
        </DataTemplate>
</GridView.ItemTemplate>
```

INTRO TO MVVM

Property communication

- The app needs a specific communication between the model and UI.
- This communication allows for the values of each property to be updated accordingly.
- This communication is performed by implementing the *INotifyPropertyChanged* interface on the model.
  - The interface contains one event only called *PropertyChanged*

Source link
https://msdn.microsoft.com/en-us/library/system.componentmodel.inotifypropertychanged(v=vs.110).aspx

INTRO TO MVVM

INotifyPropertyChanged example

```
public class BindableClass : INotifyPropertyChanged
{
    public string ObservableString
    {
        get { return _observableString; }
        set { Set(ref _observableString, value); }
    }
    private string _observableString;


    //INotifyPropertyChanged Implementation
    public event PropertyChangedEventHandler PropertyChanged;

    public void RaisePropertyChanged([CallerMemberName]string propertyName = null)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
    }

    public bool Set<T>(ref T storage, T value, [CallerMemberName]string propertyName = null)
    {
        if (Equals(storage, value))
            return false;
        storage = value;
        RaisePropertyChanged(propertyName);
        return true;
    }
}
```

Demo: Intro to MVVM

Networking APIs

NETWORKING APIS

Overview

- The set of networking APIs available for developers in .NET Core 5 is an evolution from the set that was available for Windows Store app developers in Windows 8.1

- Although most of the networking API surface in .NET Core is the same as previous .NET Framework versions, the underlying implementation for some of these APIs has undergone a significant change as we move from the .NET Framework to .NET Core.

NETWORKING APIS

New features for UWP

- **System.Net.Sockets**
  - Includes types such as *System.Net.Sockets.Socket* and *System.Net.Sockets.SocketAsyncEventArgs*, which are used by developers for asynchronous socket communication.

- **System.Net.Http gets HTTP/2**
  - HTTP/2 is the latest version of the HTTP protocol and provides much lower latency in web access by minimizing the number of connections and round-trip messages. Adding this support into the `HttpClient` API means that server responses come back much faster, leading to an app that feels more fluid at the same network speed.
  - This feature on by default, so there is zero code change required to leverage this.
  - Example:

```
var myClient = new HttpClient();
var myRequest = new HttpRequestMessage(HttpMethod.Get, "http://www.contoso.com");
// This property represents the client preference for the HTTP protocol version.
// The default value for UWP apps is 2.0.
Debug.WriteLine(myRequest.Version.ToString());
var response = await myClient.SendAsync(myRequest);
// This tells if you if the client-server communication is actually using HTTP/2
Debug.WriteLine(response.Version.ToString());
```

NETWORKING APIS

Changes for UWP

- **System.Net.Http**
  - Completely new, lightweight wrapper on top of native Windows OS HTTP components such as `Windows.Web.Http`, which is based on WinINet.
  - Helps lower the memory consumption of .NET apps running on Windows 10, thereby giving the user a more fluid experience running multiple apps simultaneously.
  - Methods and properties remain the same as in Windows 8.1 (enabling porting code).

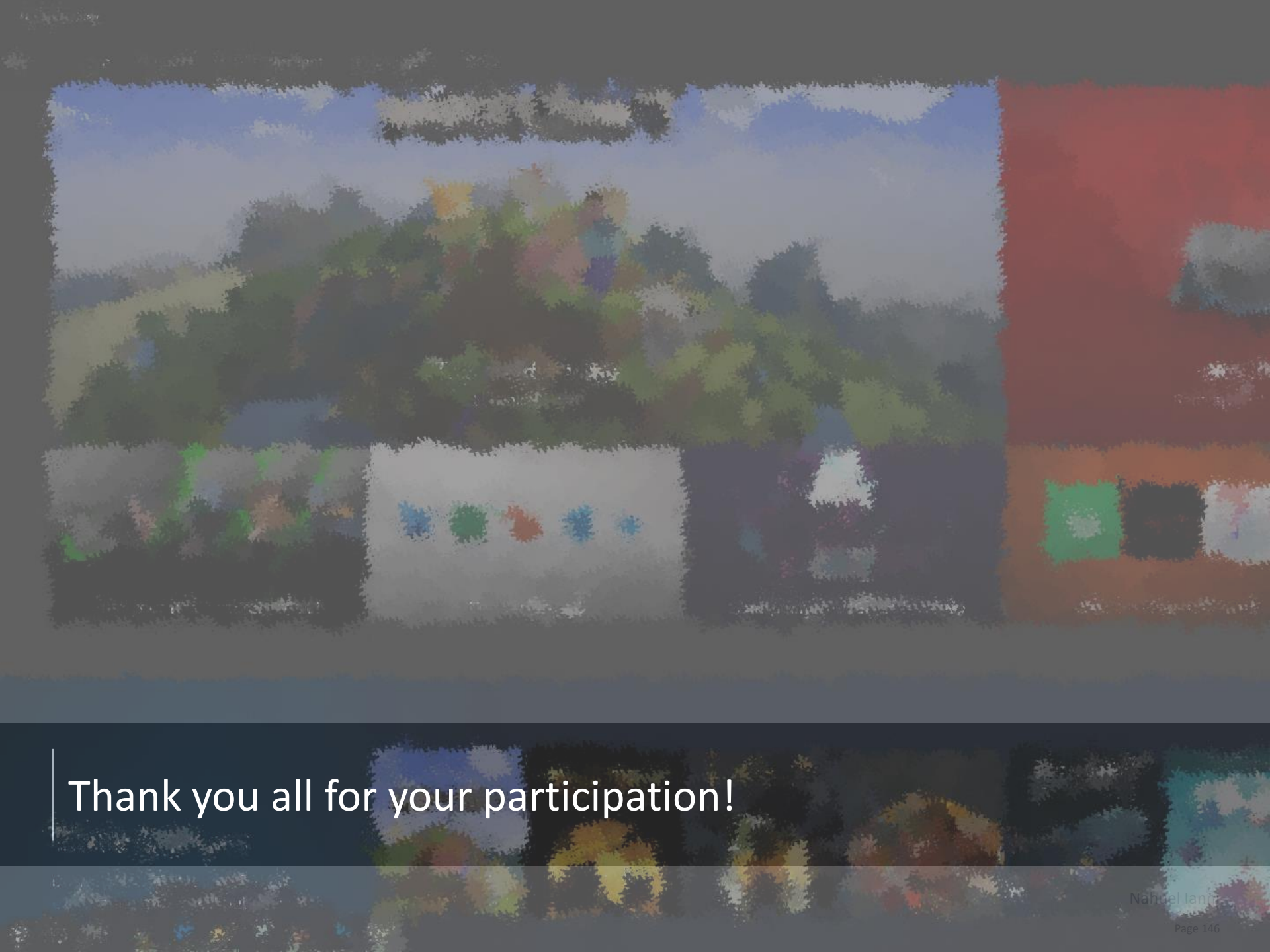
- **System.Net.Requests**
  - This library contains types related to `System.Net.HttpWebRequest` and `System.Net.HttpWebResponse` classes that allow developers to implement the client role of the HTTP protocol. The API surface for .NET Core 5 is the same as that available for Windows 8.1 apps and is very limited compared to the surface in the .NET Framework.
  - This library is provided purely for backward compatibility and to unblock usage of .NET libraries that use these older APIs.

NETWORKING APIS

Truth for UWP

- The networking functionality is ready but always changing.
- As newer versions of the SDK arrive, more changes are to be expected on the networking functionality.
- A big part of these changes come from the petitions made by the users:
- https://wpdev.uservoice.com/forums/253374-missing-platform-apis/category/82613--net-apis

Wrap up



Thank you all for your participation!